Practical Alternating Least Squares for Tensor Ring Decomposition[1]

Yajie Yu

Chongqing University, Chongqing, P.R. China

CQSIAM2024, 25th May 2024

[1] A joint work with Hanyu Li
E-mail: zqyu@cqu.equ.cn

Presentation Outline

1 Introduction

2 Proposed Method

3 Numerical Results

4 Conclusions

A Tensor is an Multi-Way Array

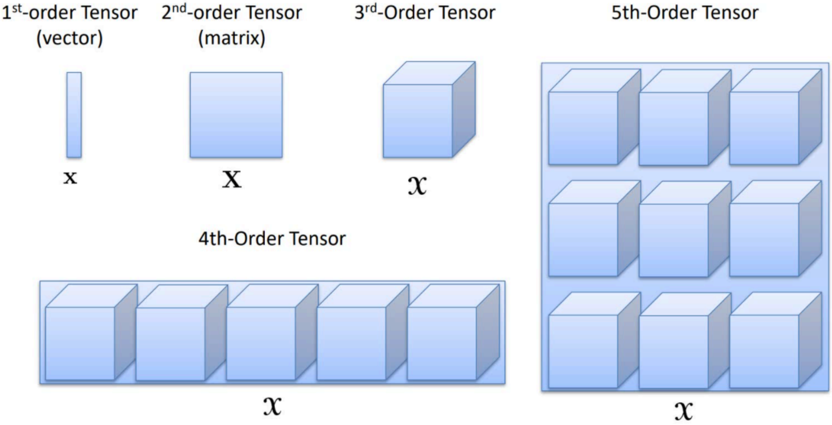
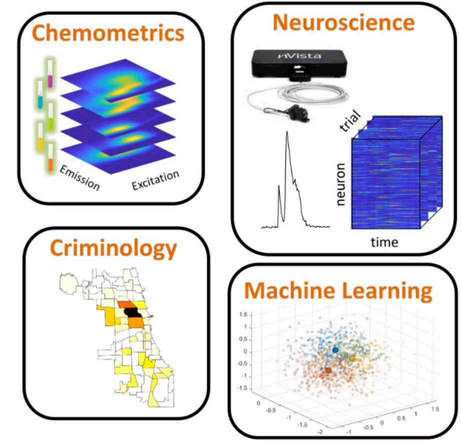


Figure: Graphical representation of multiway array (tensor) data.

Tensors Come From Many Applications

Tensor Decomposition Finds Patterns in Massive Data (Unsupervised Learning)



- Chemometrics: Emission x Excitation x Samples (Fluorescence Spectroscopy)
- Neuroscience: Neuron x Time x Trial
- Criminology: Day x Hour x Location x Crime (Chicago Crime Reports)
- Machine Learning: Multivariate Gaussian Mixture Models Higher-Order Moments
- Transportation: Pickup x Dropoff x Time (Taxis)
- Sports: Player x Statistic x Season (Basketball)
- Cyber-Traffic: IP x IP x Port x Time
- Social Network: Person x Person x Time x Interaction-Type
- Signal Processing: Sensor x Frequency x Time
- Trending Co-occurrence: Term A x Term B x Time

Some Popular Tensor Decompositions

- CANDECOMP/PARAFAC (CP) decomposition.
  - The CP tensor decomposition aims to approximate an Nth-order tensor as a sum of R rank-one tensors;
  - $\mathcal{X} \approx \tilde{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)} = [[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}]]$;
  - $\mathcal{O}\left(NIR\right)$ parameters: is linear to the tensor order N.
- Tucker decomposition
  - The Tucker decomposition decomposes a tensor into a core tensor multiplied (or transformed) by a matrix along each mode;
  - $\mathcal{X} \approx \tilde{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \cdots \times_N \mathbf{A}^{(N)} = [[\mathcal{G}; \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}]]$;
  - $\mathcal{O}\left(NIR + R^N\right)$ parameters: is exponential to the tensor order N.
- Tensor train (TT) decomposition
  - The TT decomposition decomposes a tensor into two matrices and $N-2$ core tensors by contracting auxiliary indices;
  - $\mathcal{X} \approx \tilde{\mathcal{X}} = \mathbf{G}_1 \times^1 \mathcal{G}_2 \times^1 \cdots \times^1 \mathbf{G}_N = [[\mathbf{G}^{(1)}, \mathcal{G}^{(2)}, \cdots, \mathbf{G}^{(N)}]]$;
  - $\mathcal{O}\left((N-2)IR^2 + 2IR\right)$ parameters: is linear to the tensor order N.

These Popular Tensor Decompositions Have Some Limitions and Tensor Ring (TR) Decomposition Can Overcome Them.

- Limitions

  CP Its optimization problem is difficult; it is difficult to find the optimal solution and CP-rank (NP-hard);

  Tucker Its number of parameters is exponential to tensor order. (Curse of Dimensionality)

  TT
  - The constraint on TT-ranks, i.e., $R_1 = R_{N+1} = 1$, leads to the limited representation ability and flexibility;
  - TT-ranks always have a fixed pattern, i.e., smaller for the border cores and larger for the middle cores, which might not be the optimum for specific data tensor;
  - The multilinear products of cores in TT decomposition must follow a strict order such that the optimized TT cores highly depend on the permutation of tensor dimensions. Hence, finding the optimal permutation remains a challenging problem.
- TR decomposition: $\mathcal{X} \approx \tilde{\mathcal{X}} = \mathrm{Trace}\left(\mathbf{G}_1 \times^1 \mathbf{G}_2 \times^1 \cdots \times^1 \mathbf{G}_N\right)$. ($\mathcal{O}\left(NIR^2\right)$ parameters: is linear to the tensor order N.)
- Advantages: more generalized/ powerful representation; more flexible; circular dimensional permutation invariance; TR-ranks are usually smaller than TT-ranks.

Some Algorithms for Tenor Ring (TR) Decomposition

- TR-SVD and TR-ALS [Zha+16][2].
- Randomized algorithms
  - Randomized SVD [Ahm+20][3];
  - Randomized sampling [MB21][4], [Mal22][5];
  - Randomized projection [YL24][6];
  - Others[Yua+19][7];
- Etc.

[2] Qibin Zhao et al. "Tensor Ring Decomposition". In: *arXiv preprint arXiv:1606.05535* (2016).

[3] Salman Ahmadi-Asl et al. "Randomized Algorithms for Fast Computation of Low Rank Tensor Ring Model". In: *Mach. Learn.: Sci. Technol.* 2.1 (2020), p. 011001. DOI: 10.1088/2632-2153/abad87.

[4] Osman Asif Malik and Stephen Becker. "A Sampling-Based Method for Tensor Ring Decomposition". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Virtual Event: PMLR, 2021, pp. 7400–7411.

[5] 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. Virtual Event: PMLR, 2022, pp. 14887–14917.

[6] Yajie Yu and Hanyu Li. "Practical sketching-based randomized tensor ring decomposition". In: *Numer. Linear Algebra Appl.* (2024), e2548. DOI: 10.1002/nla.2548.

[7] Longhao Yuan et al. "Randomized Tensor Ring Decomposition and Its Application to Large-Scale Data Reconstruction". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton Conference Centre Brighton, U.K.: IEEE, 2019, pp. 2127–2131.

Presentation Outline

1 Introduction

2 Proposed Method

- TR-ALS Based on Normal Equation
- TR-ALS Based on QR Factorization
- TR-ALS Based on QR Factorization and Normal Equation

3 Numerical Results

4 Conclusions

Some Motivations of Our Algorithms

- LS solution
  - Via normal equations;
  - Via QR factorization.
- Some techniques in CP

$$\arg\min_{\mathbf{A}_n} \|\mathbf{Z}^{(n)}\mathbf{A}_n^\intercal - \mathbf{X}_{(n)}^\intercal\|_F,$$

where $\mathbf{Z}^{(n)} = \mathbf{A}_N \odot \cdots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \cdots \odot \mathbf{A}_1$

  - Normal equations + property of KR product [KB09];

$$\begin{aligned}\mathbf{X}_{(n)}\mathbf{Z}^{(n)} &= \mathbf{A}_n \left((\mathbf{Z}^{(n)})^\intercal \mathbf{Z}^{(n)}\right)\\ &= \mathbf{A}_n \left((\mathbf{A}_N^\intercal \mathbf{A}_N) \circledast \cdots \circledast (\mathbf{A}_{n+1}^\intercal \mathbf{A}_{n+1}) \circledast \cdots \circledast (\mathbf{A}_1^\intercal \mathbf{A}_1)\right)\end{aligned}$$

  - QR factorization + property of KR product [Min+23];

$$\begin{aligned}\mathbf{Z}^{(n)} &= \mathbf{Q}_N\mathbf{R}_N \cdots \odot \mathbf{Q}_{n+1}\mathbf{R}_{n+1} \odot \mathbf{Q}_{n-1}\mathbf{R}_{n-1} \cdots \odot \mathbf{Q}_1\mathbf{R}_1\\ &= (\mathbf{Q}_N \cdots \otimes \mathbf{Q}_{n+1} \otimes \mathbf{Q}_{n-1} \cdots \otimes \mathbf{Q}_1)\underbrace{(\mathbf{R}_N \cdots \odot \mathbf{R}_{n+1} \odot \mathbf{R}_{n-1} \cdots \odot \mathbf{R}_1)}_{\mathbf{V}_n = \mathbf{Q}_0\mathbf{R}_0}\\ &= \underbrace{(\mathbf{Q}_N \cdots \otimes \mathbf{Q}_{n+1} \otimes \mathbf{Q}_{n-1} \cdots \otimes \mathbf{Q}_1)\mathbf{Q}_0}_{\mathbf{Q}}\ \underbrace{\mathbf{R}_0}_{\mathbf{R}}\end{aligned}$$

Write the Coefficient Matrix Explicitly with Subchain Product

- TR-ALS:

$$\arg\min_{\mathbf{G}_{n(2)}} \left\|\mathbf{G}_{[2]}^{\neq n}\mathbf{G}_{n(2)}^{\intercal} - \mathbf{X}_{[n]}^{\intercal}\right\|_F$$

- The normal equation:

$$\mathbf{X}_{[n]}\mathbf{G}_{[2]}^{\neq n} = \mathbf{G}_{n(2)}\left((\mathbf{G}_{[2]}^{\neq n})^{\intercal}\mathbf{G}_{[2]}^{\neq n}\right).$$

Definition 2.1 (Subchain Product [YL24])

Let $\mathcal{A} \in \mathbb{R}^{I_1 \times J_1 \times K}$ and $\mathcal{B} \in \mathbb{R}^{K \times J_2 \times I_2}$ be two 3rd-order tensors, and $\mathbf{A}(j_1)$ and $\mathbf{B}(j_2)$ be the j_1-th and j_2-th lateral slices of $\mathcal{A}$ and $\mathcal{B}$, respectively. The mode-2 **subchain product** of $\mathcal{A}$ and $\mathcal{B}$ is a tensor of size $I_1 \times J_1 J_2 \times I_2$ denoted by $\mathcal{A} \boxtimes_2 \mathcal{B}$ and defined as

$$(\mathcal{A} \boxtimes_2 \mathcal{B})(\overline{j_1 j_2}) = \mathcal{A}(j_1)\mathcal{B}(j_2).$$

The mode-1 and mode-3 subchain products can be defined similarly.

$$\mathcal{G}^{\neq n} = \mathcal{G}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_N \boxtimes_2 \mathcal{G}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_{n-1}.$$

Develop the Property of Subchain Product I

Definition 2.2 (Outer Product)

For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, their **outer product** is a tensor of size $I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M$ denoted by $\boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{B}}$ and defined element-wise via

$$(\boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{B}})(i_1, \cdots, i_N, j_1, \cdots, j_M) = \boldsymbol{\mathcal{A}}(i_1, \cdots, i_N)\boldsymbol{\mathcal{B}}(j_1, \cdots, j_M).$$

Definition 2.3 (General Contracted Tensor Product)

For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times J \times R_1 \times K}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{J \times I_2 \times K \times R_2}$, their **general contracted tensor product** is a 4th-order tensor of size $I_1 \times I_2 \times R_1 \times R_2$ defined as

$$(\boldsymbol{\mathcal{A}} \times_{2,4}^{1,3} \boldsymbol{\mathcal{B}})(i_1, i_2, r_1, r_2) = \sum_{j,k} \boldsymbol{\mathcal{A}}(i_1, j, r_1, k)\boldsymbol{\mathcal{B}}(j, i_2, k, r_2).$$

Develop the Property of Subchain Product II

Proposition 2.4 (Main)

Let $\mathcal{A} \in \mathbb{R}^{I_1 \times J \times K_1}$, $\mathcal{B} \in \mathbb{R}^{K_1 \times R \times L_1}$, $\mathcal{C} \in \mathbb{R}^{I_2 \times J \times K_2}$ *and* $\mathcal{D} \in \mathbb{R}^{K_2 \times R \times L_2}$ *be 3rd-order tensors. Then*

$$(\mathcal{A} \boxtimes_2 \mathcal{B})_{[2]}^{\intercal} (\mathcal{C} \boxtimes_2 \mathcal{D})_{[2]} = \left((\sum_{r=1}^{R} \mathcal{B}(r)^{\intercal} \circ \mathcal{D}(r)^{\intercal}) \times_{2,4}^{1,3} (\sum_{j=1}^{J} \mathcal{A}(j)^{\intercal} \circ \mathcal{C}(j)^{\intercal}) \right)_{<2>} .$$

$$\begin{aligned}(\mathbf{G}_{[2]}^{\neq n})^{\intercal} \mathbf{G}_{[2]}^{\neq n} &= (\mathcal{G}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_N \boxtimes_2 \mathcal{G}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_{n-1})_{[2]}^{\intercal} \times \\ &\qquad (\mathcal{G}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_N \boxtimes_2 \mathcal{G}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_{n-1})_{[2]} \\ &= \left(\mathcal{P}_{n-1} \times_{2,4}^{1,3} \cdots \times_{2,4}^{1,3} \mathcal{P}_1 \times_{2,4}^{1,3} \mathcal{P}_N \times_{2,4}^{1,3} \cdots \times_{2,4}^{1,3} \mathcal{P}_{n+1} \right)_{n<2>} ,\end{aligned}$$

where $\mathcal{P}_j = \sum_{i_j=1}^{I_j} \mathbf{G}_j(i_j)^{\intercal} \circ \mathbf{G}_j(i_j)^{\intercal}$ for $j \neq n$.

Algorithm 1 TR-ALS-NE (Proposal)

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, TR-ranks $R_1, \cdots, R_N$
Output: TR-cores $\{\mathcal{G}_n \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}\}_{n=1}^{N}$
Initialize TR-cores $\mathcal{G}_1, \cdots, \mathcal{G}_N$
Compute $\mathcal{P}_1 = \sum_{i_1=1}^{I_1} \mathbf{G}_1(i_1)^\intercal \circ \mathbf{G}_1(i_1)^\intercal, \cdots, \mathcal{P}_N = \sum_{i_N=1}^{I_N} \mathbf{G}_N(i_N)^\intercal \circ \mathbf{G}_N(i_N)^\intercal$
repeat
    for $n = 1, \cdots, N$ do
        $\mathcal{S}_n \leftarrow \mathcal{P}_{n-1} \times_{2,4}^{1,3} \cdots \times_{2,4}^{1,3} \mathcal{P}_1 \times_{2,4}^{1,3} \mathcal{P}_N \times_{2,4}^{1,3} \cdots \times_{2,4}^{1,3} \mathcal{P}_{n+1}$
        $\mathcal{G}^{\neq n} \leftarrow \mathcal{G}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_N \boxtimes_2 \mathcal{G}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_{n-1}$
        $\mathbf{M}_n \leftarrow \mathbf{X}_{[n]} \mathbf{G}^{\neq n}_{[2]}$    ▷ matricized-tensor times subchain product (MTTSP)
        Solve $\mathbf{G}_{n(2)} \mathbf{S}_{n<2>} = \mathbf{M}_n$    ▷ normal equation
        Recompute $\mathcal{P}_n = \sum_{i_n=1}^{I_n} \mathbf{G}_n(i_n)^\intercal \circ \mathbf{G}_n(i_n)^\intercal$ for the updated TR-core $\mathcal{G}_n$
    end for
until termination criteria met
```

Motivation of TR-ALS Based on QR Factorization

- The accuracy of the solutions of TR-ALS-NE depends on the square of the condition number. Thus, TR-ALS-NE can break down on matrices that are not particularly close to being numerically rank deficient.
- $\mathbf{G}_{[2]}^{\neq n} = \mathbf{QR}_{[2]}$.
- $\|\mathbf{G}_{[2]}^{\neq n}\mathbf{G}_{n(2)}^{\intercal} - \mathbf{X}_{[n]}^{\intercal}\|_F = \|\mathbf{Q}^{\intercal}\mathbf{G}_{[2]}^{\neq n}\mathbf{G}_{n(2)}^{\intercal} - \mathbf{Q}^{\intercal}\mathbf{X}_{[n]}^{\intercal}\|_F = \|\mathbf{R}_{[2]}\mathbf{G}_{n(2)}^{\intercal} - \mathbf{W}_{[n]}^{\intercal}\|_F$.

Redefine the QR Factorization for the 3rd-Order TR-core Tensor

It is Equivalent to the QR Factorization of the Mode-n unfolding of the TR-core

Definition 2.5 (Mode-n QR Factorization)

For $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its **mode-$n$ QR factorization** is defined as follows:
(1) If $I_n \geq \prod_{j \neq n} I_j$, $n = 1, 2, 3$,

$$\mathcal{A} = \mathcal{R} \times_n \mathbf{Q}, \quad n = 1, 2, 3,$$

where $\mathbf{Q} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ is an orthogonal matrix, and $\mathcal{R}$ is a 3rd-order tensor whose mode-n unfolding matrix is a upper triangular matrix of size $\prod_{j \neq n} I_j \times \prod_{j \neq n} I_j$.
(2) If $I_n \leq \prod_{j \neq n} I_j$, $n = 1, 2, 3$,

$$\mathcal{A} = \mathcal{R} \times_n \mathbf{Q}, \quad n = 1, 2, 3,$$

where $\mathbf{Q} \in \mathbb{R}^{I_n \times I_n}$ is an orthogonal matrix, and $\mathcal{R}$ is a 3rd-order tensor whose mode-n unfolding matrix is a upper triangular matrix of size $I_n \times \prod_{j \neq n} I_j$.

Efficient QR Factorization of the Coefficient Matrix $\mathbf{G}_{[2]}^{\neq n}$

Proposition 2.6 ([YL24])

Let $\mathcal{A} \in \mathbb{R}^{I_1 \times J_1 \times K}$ and $\mathcal{B} \in \mathbb{R}^{K \times J_2 \times I_2}$ be two 3rd-order tensors, and $\mathbf{A} \in \mathbb{R}^{R_1 \times J_1}$ and $\mathbf{B} \in \mathbb{R}^{R_2 \times J_2}$ be two matrices. Then

$$(\mathcal{A} \times_2 \mathbf{A}) \boxtimes_2 (\mathcal{B} \times_2 \mathbf{B}) = (\mathcal{A} \boxtimes_2 \mathcal{B}) \times_2 (\mathbf{B} \otimes \mathbf{A}).$$

$$\begin{aligned}
\mathbf{G}_{[2]}^{\neq n} &= (\mathcal{G}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_N \boxtimes_2 \mathcal{G}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_{n-1})_{[2]} \\
&= ((\mathcal{R}_{n+1} \times_2 \mathbf{Q}_{n+1}) \boxtimes_2 \cdots \boxtimes_2 (\mathcal{R}_N \times_2 \mathbf{Q}_N) \boxtimes_2 (\mathcal{R}_1 \times_2 \mathbf{Q}_1) \boxtimes_2 \cdots \boxtimes_2 (\mathcal{R}_{n-1} \times_2 \mathbf{Q}_{n-1}))_{[2]} \\
&= (\underbrace{(\mathcal{R}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_N \boxtimes_2 \mathcal{R}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_{n-1})}_{\mathcal{V}_n = \mathcal{R} \times_2 \mathbf{Q}_0} \times_2 (\mathbf{Q}_{n-1} \otimes \cdots \otimes \mathbf{Q}_1 \otimes \mathbf{Q}_N \otimes \cdots \otimes \mathbf{Q}_{n+1}))_{[2]} \\
&= ((\mathcal{R} \times_2 \mathbf{Q}_0) \times_2 (\mathbf{Q}_{n-1} \otimes \cdots \otimes \mathbf{Q}_1 \otimes \mathbf{Q}_N \otimes \cdots \otimes \mathbf{Q}_{n+1}))_{[2]} \\
&= (\mathcal{R} \times_2 \underbrace{((\mathbf{Q}_{n-1} \otimes \cdots \otimes \mathbf{Q}_1 \otimes \mathbf{Q}_N \otimes \cdots \otimes \mathbf{Q}_{n+1})\mathbf{Q}_0)}_{\mathbf{Q}})_{[2]} \\
&= (\mathcal{R} \times_2 \mathbf{Q})_{[2]} \\
&= \mathbf{Q}\mathbf{R}_{[2]}
\end{aligned}$$

Algorithm 2 TR-ALS-QR (Proposal)

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, TR-ranks $R_1, \cdots, R_N$
Output: TR-cores $\{\mathcal{G}_n \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}\}_{n=1}^{N}$
Initialize TR-cores $\mathcal{G}_1, \cdots, \mathcal{G}_N$
Compute the mode-2 QR factorizations $\mathcal{R}_1 \times_2 \mathbf{Q}_1, \cdots, \mathcal{R}_N \times_2 \mathbf{Q}_N$ of TR-cores
repeat
    for $n = 1, \cdots, N$ do
        $\mathcal{V}_n \leftarrow \mathcal{R}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_N \boxtimes_2 \mathcal{R}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_{n-1}$
        Compute mode-2 QR factorization $\mathcal{V}_n = \mathcal{R} \times_2 \mathbf{Q}_0$
        $\mathcal{Y} \leftarrow \mathcal{X} \times_1 \mathbf{Q}_1^\top \times_2 \cdots \times_{n-1} \mathbf{Q}_{n-1}^\top \times_{n+1} \mathbf{Q}_{n+1}^\top \times_{n+2} \cdots \times_N \mathbf{Q}_N^\top$
        $\mathbf{W}_n \leftarrow \mathbf{Y}_{[n]} \mathbf{Q}_0$
        Solve $\mathbf{G}_{n(2)} \mathbf{R}_{[2]}^\top = \mathbf{W}_n$ by substitution
        Recompute the mode-2 QR factorization $\mathcal{R}_n \times_2 \mathbf{Q}_n$ for the updated TR-core $\mathcal{G}_n$
    end for
until termination criteria met
```

The QR Factorization $\mathcal{V}_n$ is Expensive even though it is a Highly Structured Sparse Tensor

TR-ALS-QRNE: Balance the Computational Efficiency and Numerical Stability

$$
\begin{aligned}
\mathbf{G}^{\neq n}_{[2]} &= \left(\mathcal{G}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_N \boxtimes_2 \mathcal{G}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{G}_{n-1}\right)_{[2]} \\
&= \left((\mathcal{R}_{n+1} \times_2 \mathbf{Q}_{n+1}) \boxtimes_2 \cdots \boxtimes_2 (\mathcal{R}_N \times_2 \mathbf{Q}_N) \boxtimes_2 (\mathcal{R}_1 \times_2 \mathbf{Q}_1) \boxtimes_2 \cdots \boxtimes_2 (\mathcal{R}_{n-1} \times_2 \mathbf{Q}_{n-1})\right)_{[2]} \\
&= \Big(\underbrace{(\mathcal{R}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_N \boxtimes_2 \mathcal{R}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_{n-1})}_{\mathcal{V}_n} \times_2 (\mathbf{Q}_{n-1} \otimes \cdots \otimes \mathbf{Q}_1 \otimes \mathbf{Q}_N \otimes \cdots \otimes \mathbf{Q}_{n+1})\Big)_{[2]}
\end{aligned}
$$

Algorithm 3 TR-ALS-QRNE (Proposal)

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, TR-ranks $R_1, \cdots, R_N$
Output: TR-cores $\{\mathcal{G}_n \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}\}_{n=1}^{N}$
Initialize TR-cores $\mathcal{G}_1, \cdots, \mathcal{G}_N$
Compute the mode-2 QR factorizations $\mathcal{R}_1 \times_2 \mathbf{Q}_1, \cdots, \mathcal{R}_N \times_2 \mathbf{Q}_N$ of TR-cores
Compute $\mathcal{P}_1 = \sum_{i_1=1}^{I_1} \mathbf{R}_1(i_1)^\top \circ \mathbf{R}_1(i_1)^\top, \cdots, \mathcal{P}_N = \sum_{i_N=1}^{I_N} \mathbf{R}_N(i_N)^\top \circ \mathbf{R}_N(i_N)^\top$
repeat
    for $n = 1, \cdots, N$ do
        $\mathcal{S}_n \leftarrow \mathcal{P}_{n-1} \times_{2,4}^{1,3} \cdots \times_{2,4}^{1,3} \mathcal{P}_1 \times_{2,4}^{1,3} \mathcal{P}_N \times_{2,4}^{1,3} \cdots \times_{2,4}^{1,3} \mathcal{P}_{n+1}$
        $\mathcal{V}_n \leftarrow \mathcal{R}_{n+1} \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_N \boxtimes_2 \mathcal{R}_1 \boxtimes_2 \cdots \boxtimes_2 \mathcal{R}_{n-1}$
        $\mathcal{Y} \leftarrow \mathcal{X} \times_1 \mathbf{Q}_1^\top \cdots \times_{n-1} \mathbf{Q}_{n-1}^\top \times_{n+1} \mathbf{Q}_{n+1}^\top \cdots \times_N \mathbf{Q}_N^\top$
        $\mathbf{M}_n \leftarrow \mathbf{Y}_{[n]} \mathbf{V}_{n[2]}$
        Solve $\mathbf{G}_{n(2)} \mathbf{S}_{n<2>} = \mathbf{M}_n$
        Recompute the mode-2 QR factorization $\mathcal{R}_n \times_2 \mathbf{Q}_n$ for the updated TR-core $\mathcal{G}_n$
        Recompute $\mathcal{P}_n = \sum_{i_n=1}^{I_n} \mathbf{R}_n(i_n)^\top \circ \mathbf{R}_n(i_n)^\top$ for the updated $\mathcal{R}_n$
    end for
until termination criteria met
```

Presentation Outline

1 Introduction

2 Proposed Method

3 Numerical Results

- Efficiency of TR-ALS-NE
- Stability of TR-ALS-QR and TR-ALS-QRNE
- Performance on Real Datasets

4 Conclusions

Data Generation

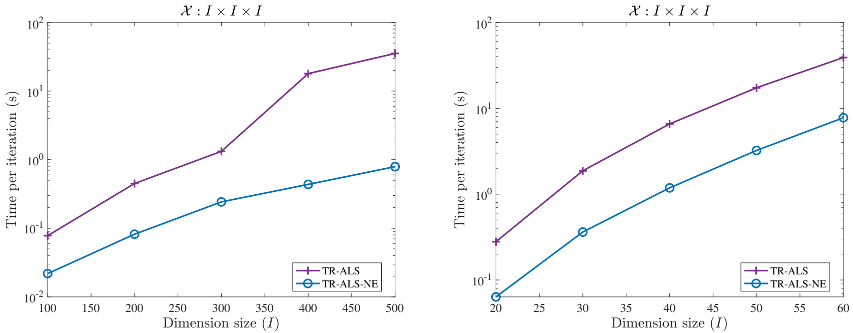
- $$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{true} + \eta \left(\frac{\|\boldsymbol{\mathcal{X}}_{true}\|_F}{\|\boldsymbol{\mathcal{N}}\|_F} \right) \boldsymbol{\mathcal{N}},$$

  where $\boldsymbol{\mathcal{X}}_{true} = \mathrm{TR}\left(\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^{N}\right)$, the entries of $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{I \times \cdots \times I}$ are drawn from a standard normal distribution, and the parameter η is the amount of noise.

  - Generate_Tensor(I, R_{true}, η)
    Each TR-core is generated by a random Gaussian tensor with entries drawn independently from a standard normal distribution.
  - Generate_Collinear_Tensor$(I, R_{true}, \eta, \gamma)$
    Using Matrandcong(I, R_{true}^2, γ) to generate the TR-cores, The parameter γ is used to control the congruence of matrices and hence the collinearity of TR-cores.
  - Generate_Multivariate_T-Distribution_Tensor$(I, R_{true}, \eta, \theta)$
    Using Mvtrnd$(\mathbf{C}, d, I)$ togenerate the TR-cores, where $\mathbf{C} \in \mathbb{R}^{R_{true} \times R_{true}}$ is a correlation matrix whose (i, j)-th element is equal to $\theta^{|i-j|}$ with θ describing the correlation level, and d is the degrees of freedom. We always set $d = 1$ in our specific experiment.

Efficiency of TR-ALS-NE: Experiment A-I

Data: Generate_Tensor$(I, R_{true}, 0)$



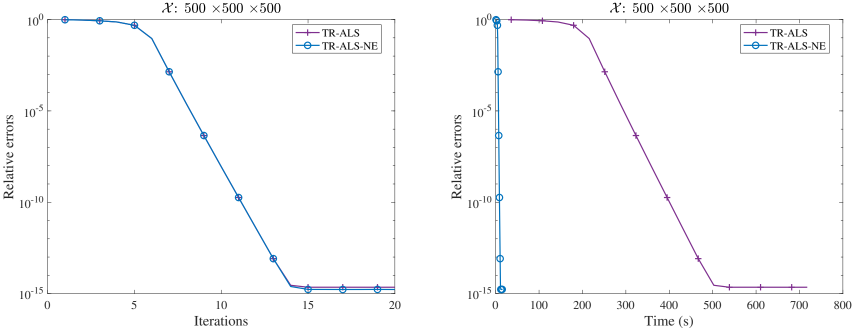
(a) 3rd-order tensor, $R_{true} = R = 10$

(b) 5th-order tensor, $R_{true} = R = 4$

Figure: Mean time per iteration of TR-ALS and TR-ALS-NE for 3rd- and 5th-order tensors. Each dot represents the mean iteration time over 20 iterations (no checks for convergence).

Efficiency of TR-ALS-NE: Experiment A-II

Data: Generate_Tensor$(I, R_{true}, 0)$

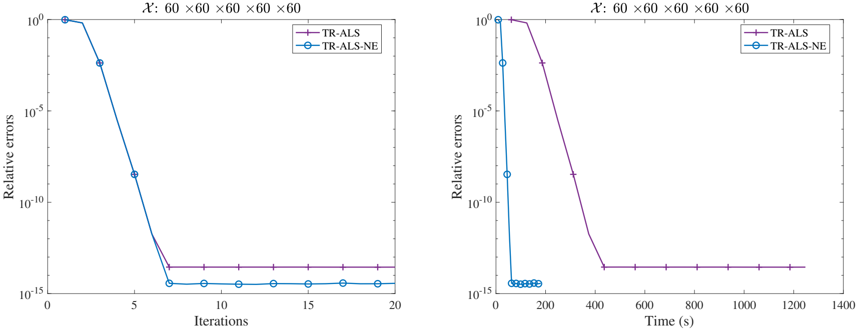


(a) Number of iterations v.s. Relative errors **(b)** Time v.s. Relative errors

Figure: Results output by TR-ALS and TR-ALS-NE for $\mathfrak{X}: 500 \times 500 \times 500$, $R_{true} = R = 10$.

Efficiency of TR-ALS-NE: Experiment A-II

Data: Generate_Tensor$(I, R_{true}, 0)$

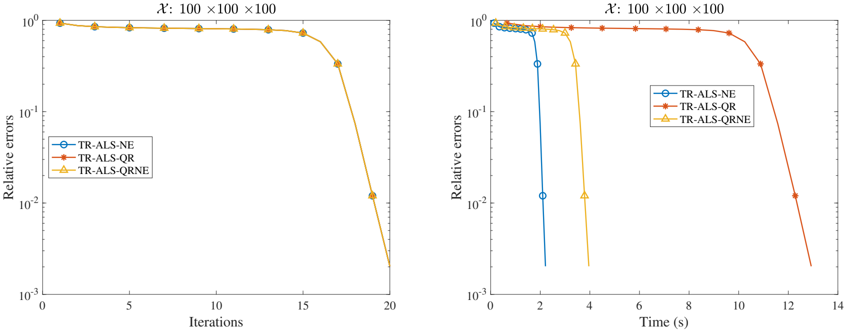


(a) Number of iterations v.s. Relative errors **(b)** Time v.s. Relative errors

Figure: Results output by TR-ALS and TR-ALS-NE for $\mathfrak{X}$: $60 \times 60 \times 60 \times 60 \times 60$, $R_{true} = R = 5$.

Stability of TR-ALS-QR and TR-ALS-QRNE: Experiment B-I

Data: Generate_Tensor$(I, R_{true}, 0)$

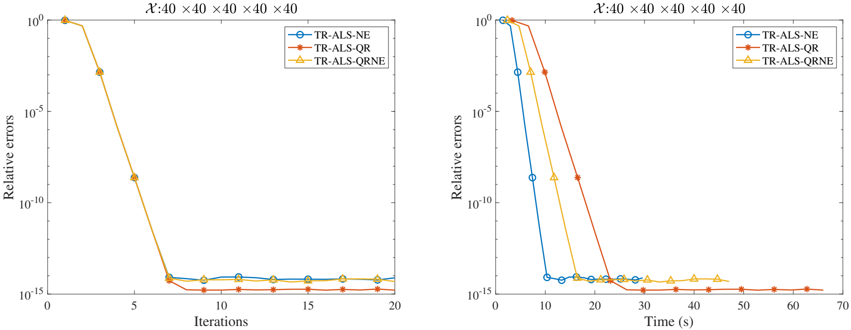


(a) Number of iterations v.s. Relative errors (b) Time v.s. Relative errors

Figure: Results output by TR-ALS-NE, TR-ALS-QR and TR-ALS-QRNE for $\mathfrak{X} : 100 \times 100 \times 100$, $R_{true} = R = 15$.

Stability of TR-ALS-QR and TR-ALS-QRNE: Experiment B-I

Data: Generate_Tensor(I, R_{true}, 0)



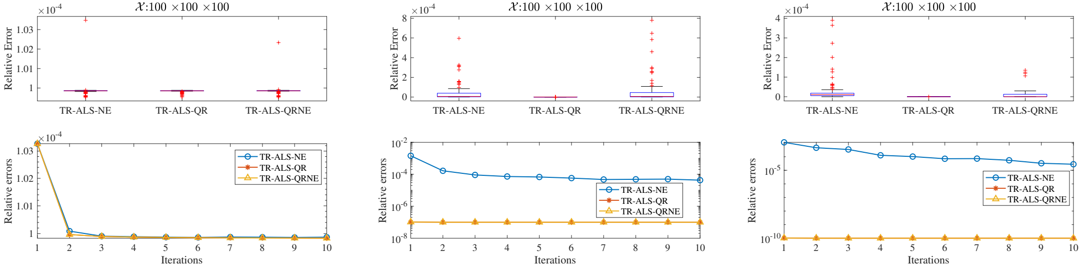
(a) Number of iterations v.s. Relative errors

(b) Time v.s. Relative errors

Figure: Results output by TR-ALS-NE, TR-ALS-QR and TR-ALS-QRNE for $\mathfrak{X} : 40 \times 40 \times 40 \times 40 \times 40$, $R_{true} = R = 5$.

Stability of TR-ALS-QR and TR-ALS-QRNE: Experiment B-II

Data: Generate_Collinear_Tensor$(I, R_{true}, \eta, \gamma)$

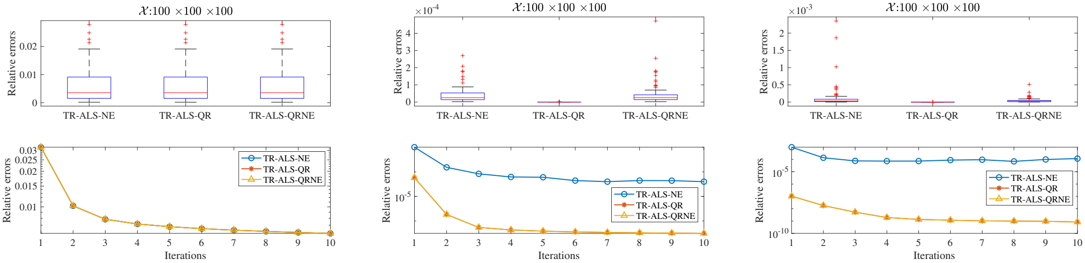


(a) $\eta = 10^{-4}, \gamma = 1 - 10^{-4}$ **(b)** $\eta = 10^{-7}, \gamma = 1 - 10^{-7}$ **(c)** $\eta = 10^{-10}, \gamma = 1 - 10^{-10}$

Figure: Boxplots and line chart of relative errors for TR-ALS-NE, TR-ALS-QR and TR-ALS-QRNE on a $100 \times 100 \times 100$ synthetic tensor of rank 5 with three different levels of collinearity for the true TR-cores and three different levels of Gaussian noise added. Each algorithm is run 100 trials.

STABILITY OF TR-ALS-QR AND TR-ALS-QRNE: EXPERIMENT B-III

DATA: GENERATE_MULTIVARIATE_T-DISTRIBUTION_TENSOR$(I, R_{true}, \eta, \theta)$



(a) $\eta = 10^{-4}, \theta = 1 - 10^{-1}$ **(b)** $\eta = 10^{-7}, \theta = 1 - 10^{-4}$ **(c)** $\eta = 10^{-10}, \theta = 1 - 10^{-7}$

Figure: Boxplots and line chart of relative errors for TR-ALS-NE, TR-ALS-QR and TR-ALS-QRNE on a $100 \times 100 \times 100$ synthetic tensor of rank 5 with three different kinds of the correlation matrices for the true TR-cores and three different levels of Gaussian noise added. Each algorithm is run 100 trials.

Real Datasets

Table: Size and type of real datasets.

Dataset	Size	Type
DC Mall	$1280 \times 307 \times 191$	Hyperspectral image
Park Bench	$1080 \times 1920 \times 364$	Video
Tabby Cat	$720 \times 1280 \times 286$	Video

Table: Size of truncated and reshaped real datasets.

Dataset	Size
DC Mall (reshaped)	$32 \times 40 \times 18 \times 17 \times 10 \times 19$
Park Bench (reshaped)	$24 \times 45 \times 32 \times 60 \times 28 \times 13$
Tabby Cat (reshaped)	$16 \times 45 \times 32 \times 40 \times 13 \times 22$

Performance

Table: Decompositions for real datasets with target rank $R = 3$.

	DC Mall		Park Bench		Tabby Cat	
Method	Error	Time (s)	Error	Time (s)	Error	Time (s)
TR-ALS	0.331	54.6	0.183	519.2	0.189	188.7
TR-ALS-NE	0.331	5.6	0.183	47.9	0.189	19.8
TR-ALS-QR	0.331	2.8	0.183	33.3	0.189	12.2
TR-ALS-QRNE	0.331	2.8	0.183	33.3	0.189	12.1

Table: Decompositions for real datasets with target rank $R = 3$.

	DC Mall (reshaped)		Park Bench (reshaped)		Tabby Cat (reshaped)	
Method	Error	Time (s)	Error	Time (s)	Error	Time (s)
TR-ALS	0.384	114.9	0.214	914.7	0.197	351.1
TR-ALS-NE	0.384	22.2	0.214	189.0	0.197	74.3
TR-ALS-QR	0.384	13.3	0.214	107.4	0.197	53.7
TR-ALS-QRNE	0.384	12.9	0.214	107.0	0.197	53.4

Presentation Outline

1 Introduction

2 Proposed Method

3 Numerical Results

4 Conclusions

Conclusions

- Conclusions
  - We propose three efficient algorithms for TR decomposition by fully using its structure: TR-ALS-NE, TR-ALS-QR, and TR-ALS-QRNE
  - We provide a property of the subchain product and the mode-n QR factorization of 3rd-order tensor.
  - Numerical experiments are provided to test the proposed methods.
- Future works
  - MTTSP.
  - Special tensors.
  - Other tensor decompositions.

Thanks!

Yu, Y., & Li, H. (2024). Practical alternating least squares for tensor ring decomposition. Numerical Linear Algebra with Applications, 31(3), e2542.

References I

[Ahm+20] Salman Ahmadi-Asl et al. "Randomized Algorithms for Fast Computation of Low Rank Tensor Ring Model". In: *Mach. Learn.: Sci. Technol.* 2.1 (2020), p. 011001. doi: 10.1088/2632-2153/abad87.

[KB09] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Rev.* 51.3 (2009), pp. 455–500. doi: 10.1137/07070111X.

[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. Virtual Event: PMLR, 2022, pp. 14887–14917.

[MB21] Osman Asif Malik and Stephen Becker. "A Sampling-Based Method for Tensor Ring Decomposition". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Virtual Event: PMLR, 2021, pp. 7400–7411.

References II

[Min+23] Rachel Minster et al. “CP Decomposition for Tensors via Alternating Least Squares with QR Decomposition”. In: *Numer. Linear Algebra Appl.* (2023), e2511. DOI: 10.1002/nla.2511.

[YL24] Yajie Yu and Hanyu Li. “Practical sketching-based randomized tensor ring decomposition”. In: *Numer. Linear Algebra Appl.* (2024), e2548. DOI: 10.1002/nla.2548.

[Yua+19] Longhao Yuan et al. “Randomized Tensor Ring Decomposition and Its Application to Large-Scale Data Reconstruction”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton Conference Centre Brighton, U.K.: IEEE, 2019, pp. 2127–2131.

[Zha+16] Qibin Zhao et al. “Tensor Ring Decomposition”. In: *arXiv preprint arXiv:1606.05535* (2016).