Tensor Decomposition and Applications

Hrriday Viraj Ruparel | Prof. Anirban Dasgupta

Computer Science and Engineering, IIT Gandhinagar

Abstract

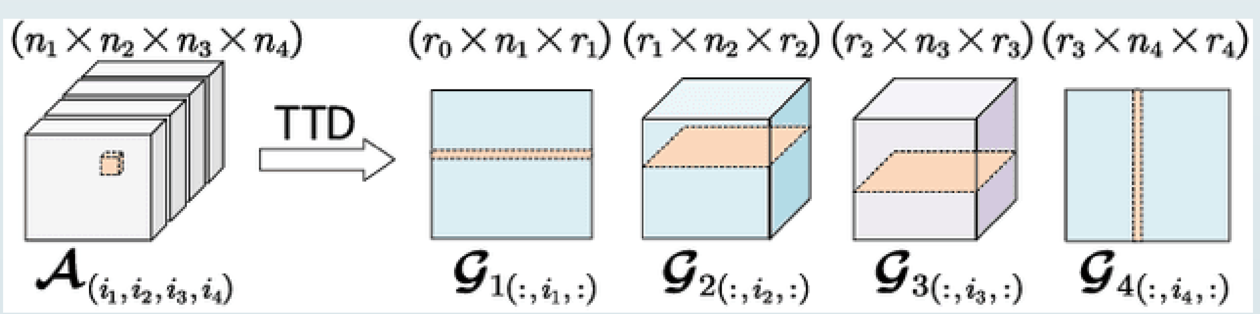
This project explores efficient algorithms for and potential novel applications of tensor decomposition, with special focus on a type of tensor decomposition called **Tensor Train Decomposition (TTD)** utilized for large-scale multi-way tensor analysis and computations. While existing literature models TTD as a reconstruction problem under F-norm, this project analyzes properties of **TTD under L_1-norm**. Modifying the optimization problem, thus brings new challenges, provides key insights and opens avenues for further research.

Introduction

Analogous to matrix decomposition/factorization techniques, tensor decomposition breaks down high-order tensors into meaningful factors or smaller tensors, enabling multi-way factor analysis and compression.

Among the many techniques for tensor decomposition, our main focus would be on a special case, namely Tensor Train Decomposition (TTD).

Tensor Train Decomposition (TTD)



- Inspired from the **Matrix Product State (MPS)** representation, TTD decomposes any Nth-order tensor to smaller multiple 3rd-order tensors via a sequential **Singular Value Decomposition (SVD)** approach.
- Original tensor can be **arbitrarily approximated** using TTD, and the representation elegantly overcomes the **curse of dimensionality**.
- Governing optimization problem:

$$\min_{\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_N} \left\|\mathcal{A} - \hat{\mathcal{A}}\right\|_F, \qquad \mathcal{A}, \hat{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$$

$$\hat{\mathcal{A}} = \mathcal{G}_1 \times_3^1 \mathcal{G}_2 \times_3^1 \ldots \times_3^1 \mathcal{G}_N, \qquad \mathcal{G}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

Problem Formulation

New optimization problem

$$\min_{\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_N} \left\|\mathcal{A} - \hat{\mathcal{A}}\right\|_{L_1}, \qquad \mathcal{A}, \hat{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$$

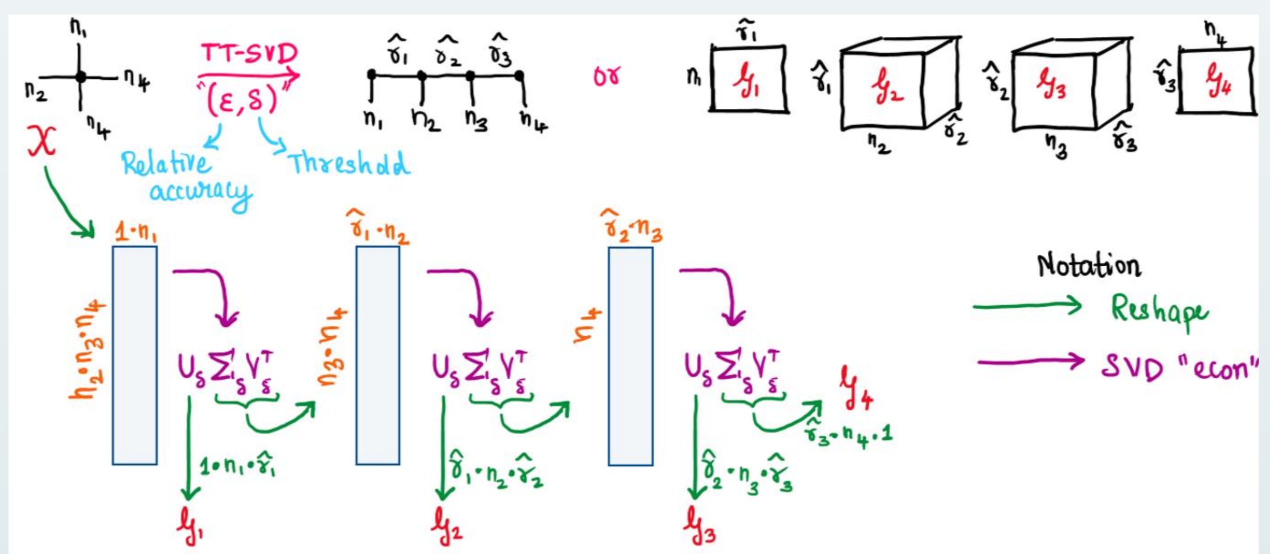
$$\hat{\mathcal{A}} = \mathcal{G}_1 \times_3^1 \mathcal{G}_2 \times_3^1 \ldots \times_3^1 \mathcal{G}_N, \qquad \mathcal{G}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$$

Motivation

- To analyze robustness to outliers.
- To investigate low tensor-rank structure in data
- Similar work done for Tucker Decomposition

Methodology

TT-SVD (Sequential SVDs)



Alternating Convex Programming (AltConvPro)

Inspired by TT-SVD, this sequential algorithm solves an **L_1-norm based matrix factorization** problem using simpler alternating linear programs.

$$\min_{U,V} \|X - UV^T\|_{L_1}$$

$$\downarrow$$

$$\min_{U} \|X - UV^T\|_{L_1} \;\; \& \;\; \min_{V} \|X - UV^T\|_{L_1}$$

$$X \in \mathbb{R}^{m \times n}, U \in \mathbb{R}^{m \times r}, V = \mathbb{R}^{n \times r}, r \ll m, n$$

$$\downarrow$$

Linear Programs

In the matrix case, the following properties hold:

- Convex Problem ⇒ **Global Minima**
- Can handle missing data with small modification to optimization problem (weighted optimization)
- Global convergence guaranteed
- **Polynomial-time (but very high!)**

Experiments

Setup

- Fix a tensor size and generate a random tensor
- Compute TT-ranks and optionally, reduce the TT-ranks by varying compression appropriately.
- For different fractions of outliers and noise level added uniformly at random, perform:
  1. L_1-norm based TTD via AltConvPro
  2. L_2-norm based TTD via TT-SVD
- Measure normalized reconstruction errors wrt. uncorrupted tensor, time taken and number of close-to-zero values in the tensor cores.

$$\text{NRE} = \frac{\left\|\mathcal{A} - \hat{\mathcal{A}}\right\|_F}{\|\mathcal{A}\|_F}$$

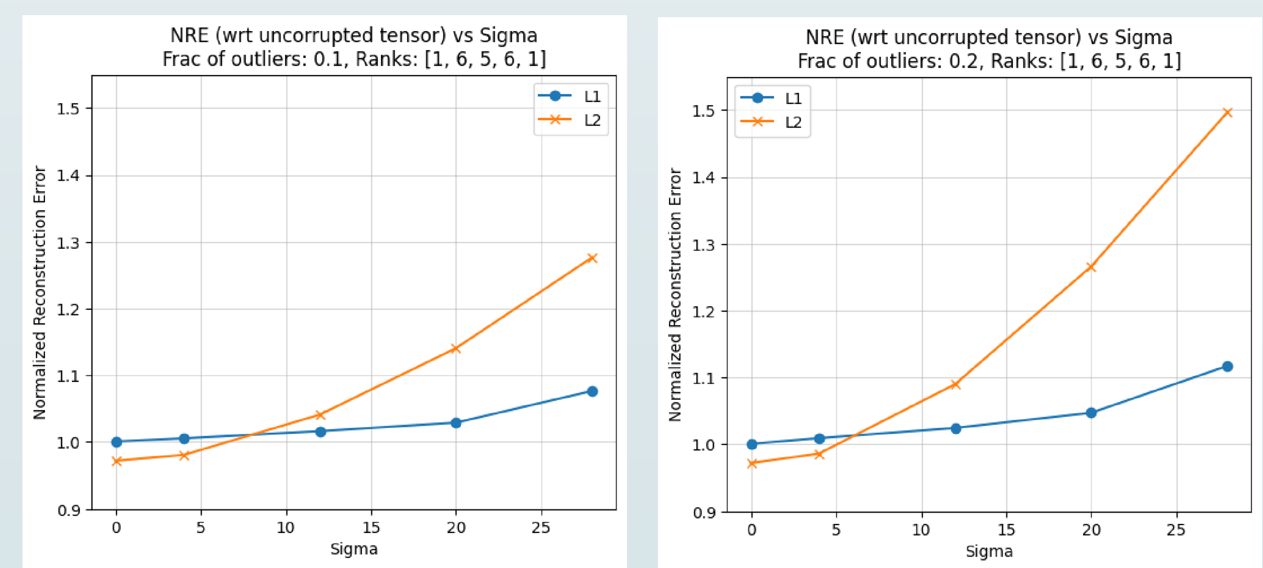
Parameters

- Tensor shape: (10, 15, 10, 15)
- Frac of Outliers: [0, 0.05, 0.1, 0.15, 0.2]
- Sigma (AWGN Outlier Noise): [0, 4, 12, 20, 28]
- TT-Ranks: {[1, 4, 3, 4, 1], [1, 6, 5, 6 ,1], [1, 8, 9, 8, 1]}

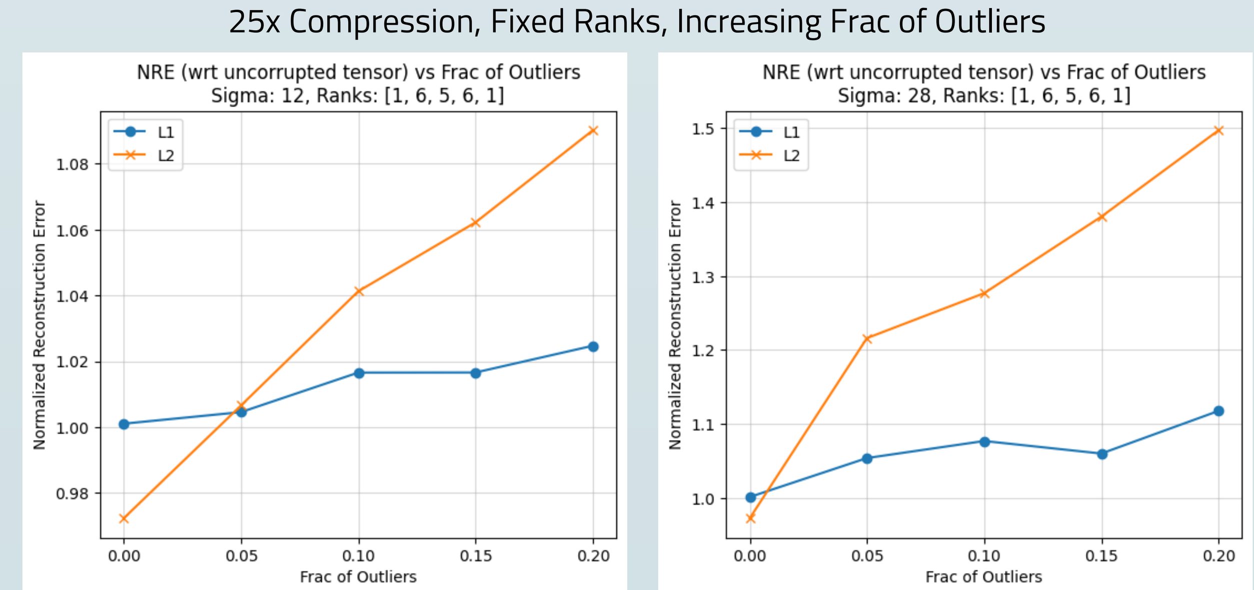
$$\mathcal{A}_{corr} = \mathcal{A} + \mathcal{O}$$

$$[\mathcal{A}]_{i,j,k,l} \sim \mathcal{N}(0, 9), \qquad [\mathcal{O}]_{i,j,k,l} \sim \mathcal{N}(0, Sigma^2)$$

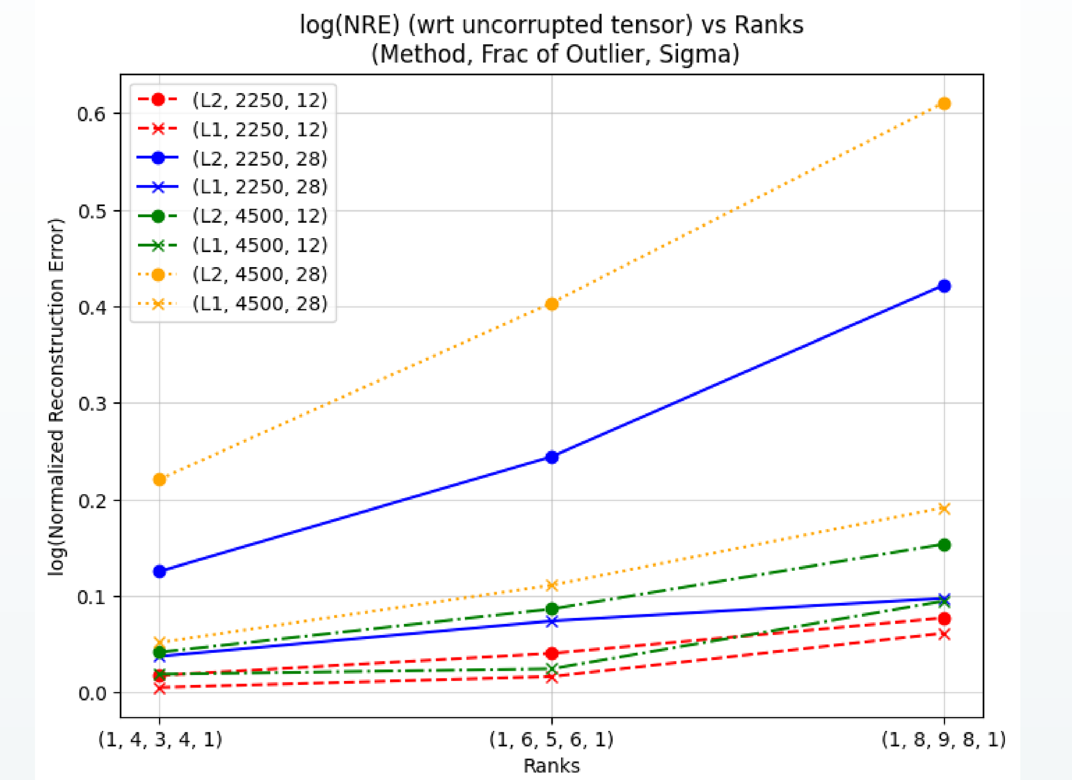
Results



25x Compression, Fixed Ranks, Increasing Frac of Outliers



25x Compression, Fixed Ranks, Increasing Sigma



Variation of log(NRE) with TT-ranks at varying number of outliers and noise levels

Observations

- As expected, the proposed L_1-norm based TTD via AltConvPro achieves **robust results** as compared to L_2-norm based TTD via TT-SVD.
- The proposed algorithm *significantly* reduces NRE wrt. uncorrupted tensor under **high-noise conditions**, **large number of outliers** and **large TT-ranks**.
- Unlike TT-SVD, AltConvPro was able to set some entries of the TT-cores to zero.
- While TT-SVD decomposed the tensor in fractions of seconds, AltConvPro took nearly **40-80s** (40s at small TT-ranks, 80s at large TT-ranks).

Future Work

- Explore probabilistic and other regimes of optimization techniques for tackling both time and space complexity issues
- Provide theoretical guarantees about the hardness of the problem
- Apply algorithm to real-world applications

References

- Le, Duc H., and Panos P. Markopoulos. "**Robust Singular Values based on L1-norm PCA**." 2022 IEEE Workshop on Signal Processing Systems (SiPS). IEEE, 2022.
- Chachlakis, Dimitris G., Ashley Prater-Bennette, and Panos P. Markopoulos. "**L1-norm Tucker tensor decomposition**." IEEE Access 7 (2019): 178454-178465.
- Ke, Qifa, and Takeo Kanade. "**Robust L1/norm factorization in the presence of outliers and missing data by alternative convex programming**." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

I would like to thank Prof. Anirban Dasgupta for his unwavering support and guidance throughout the project.