Exploring quasi-optimal contraction strategies for fast scalar product in Tensor-Train format

Przemyslaw Dominikowski[0009−0006−0984−9213]

Laboratoire Interdisciplinaire des Sciences du Numérique
Université Paris-Saclay, Gif-sur-Yvette, France
przemyslaw.dominikowski@universite-paris-saclay.fr

Abstract. The tensor-train decomposition has gained particular interest in recent years in tensor algebra due to its space- and compute-efficiency, as well as its ability to extend vector and matrix products into networks of tensor contractions. However, the tensor-train scalar product introduces the challenge of finding a suitable contraction order.
In this work, we introduce quasi-optimal contraction ordering algorithms tailored for the tensor-train scalar product. Experimental results demonstrate that our algorithms surpass general tensor network solvers, while having shorter execution times than a state-of-the-art optimal solver, offering a promising alternative to both.

Keywords: numerical linear algebra · multilinear algebra · tensor decomposition · tensor-train product · contraction strategies

1 Motivation

In multilinear algebra, a tensor is an object representing multidimensional data. Tensors have become a common tool in a multitude of fields, such as machine learning and signal processing [8], quantum computing [4], and many more [5]. Thus, accelerating tensor operations is crucial for enhancing performance and efficiency, making it a **core building block** for advancing scientific research.

The usefulness of tensors is, however, limited by the exponential growth of memory requirements, which is often called **the curse of dimensionality**. To overcome this limitation and to improve computational performance, tensors can be decomposed into memory-efficient approximations [5], one technique of particular interest in recent years being **tensor-train decomposition** [7].

The definition of tensor-trains permits one to extend vector- and matrix-products into the TT form, representing them as a network of contractions between tensor-trains. Immediately, a problem arises: what is the best order to compute the contractions in tensor-train products?

In our experience, state-of-the solutions for general tensor networks [3] are good on small TT and in many cases give progressively worse orderings when TT get longer, even surpassing the naive cost, which motivated us to design **the contraction ordering** algorithms tailored to TT scalar product structure.

2 Contributions

As finding an optimal ordering of the contractions for the general tensor network is a well-known **NP-complete** problem [6], in this work we present multiple quasi-optimal algorithms to solve the tensor-train scalar product, each variant is either faster or gives a better solution. While focused, our algorithms are generalized to any type of tensor train scalar-product, i.e. products of type vector-matrix-...-matrix-vector.

1-sided 1-dim algorithm

The *1-sided 1-dim* algorithm is an iterative algorithm, which finds a solution for a window consisting of the left-most mode contraction and rank contractions. At each step of the algorithm, only two out of the three contractions are performed, generating multiple sub-problems with different mode contraction costs.

2-sided Δ-dim algorithm

The *2-sided Δ-dim algorithm* is a dynamic programming algorithm, which finds the solutions for each window in the network using either optimal solver for windows up to Δ, or by building quasi-optimal orders based on already calculated smaller windows for windows larger than Δ.

3 Results

We have compared our algorithms in terms of the computational cost of yielded contraction order and the execution time with the *Optimal* solver, state-of-the-art hypergraph partitioning tools *Hyper-Greedy*, *CGreedy*, *Hyper-Kahypar* [3], *QuickBB* [2] and *FlowCutter* [1] algorithms, and the naive approach. The benchmark consisted of various TT networks and scalar product problems.

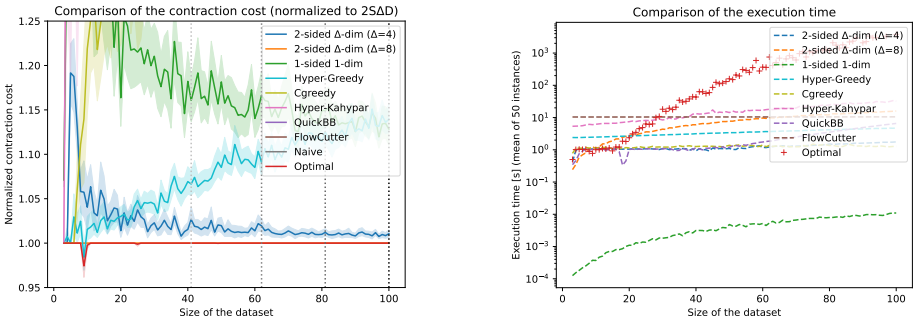

Fig. 1. Comparison of the characteristics of our contraction ordering algorithms with state-of-the-art solutions for vector-vector scalar product xx^T

The comparison of the algorithms for the vector-vector product case shows that our algorithms provide best-in-class quasi-optimal contraction orders, converging towards the optimal contraction cost. Notably, *2-sided Δ-dim* is **nearly indistinguishable** from the optimal solution, even for small Δ, while the best competing algorithms *Hyper-Greedy* and *CGreedy* diverge from optimal, reaching 10-15% and 15-50% worse results respectively, for TT of length 100.

Additionally, an important advantage of our algorithms is in terms of execution time. The optimal solver increases in execution time much faster than all other algorithms, easily reaching the threshold of 1 hour or more to find a solution, while *1-sided 1-dim* algorithm has an unbeatable **competitive edge**.

4 Conclusions

We have introduced quasi-optimal algorithms to solve the contraction ordering problem in tensor-train scalar products, that show **significant improvement** when compared to algorithms designed for a general tensor network, while necessitating only a fraction of the time required by the exact solver.

Thanks to the ubiquitous usage of tensors in a multitude of domains, our algorithms offer a **promising alternative** to solve the contraction ordering problem on tensor-train scalar products and can thus bring a significant gain in terms of the performance and efficiency of the computations in these fields.

Acknowledgments. This work was partially funded by Agence Nationale de Recherche (ANR-20-CE46-0008-01) and Île de France DIM-RFSI (DIM-RFSI-NO-2021-05).

This work was performed using computational resources from the "Mésocentre" computing center of Université Paris-Saclay, CentraleSupélec and ENS Paris-Saclay supported by CNRS and Région Île-de-France.

References

1. Dudek, J.M., et al.: Efficient contraction of large tensor networks for weighted model counting through graph decompositions (2020), https://arxiv.org/abs/1908.04381
2. Gogate, V., Dechter, R.: A complete anytime algorithm for treewidth (2012), https://arxiv.org/abs/1207.4109
3. Gray, J., Kourtis, S.: Hyper-optimized tensor network contraction. Quantum **5**, 410 (Mar 2021). https://doi.org/10.22331/q-2021-03-15-410
4. Huang, C., et al.: Efficient parallelization of tensor network contraction for simulating quantum computation. Nature Computational Science **1**(9), 578–587 (2021)
5. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review **51**(3), 455–500 (2009), https://doi.org/10.1137/07070111X
6. Lam, C.C., et al.: On optimizing a class of multi-dimensional loops with reductions for parallel execution. Parallel Process. Lett. **7**, 157–168 (1997), https://api.semanticscholar.org/CorpusID:9440379
7. Oseledets, I.: Tensor-train decomposition. SIAM J. Scientific Computing **33**, 2295–2317 (01 2011). https://doi.org/10.1137/090752286
8. Sidiropoulos, N.D., et al.: Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing **65**(13), 3551–3582 (2017). https://doi.org/10.1109/TSP.2017.2690524