Is Rank Minimization of the Essence to Learn Tensor Network Structure?

Chao Li
RIKEN-AIP
chao.li@riken.jp

Qibin Zhao*
RIKEN-AIP
qibin.zhao@riken.jp

Abstract

Structure learning of tensor network (TN) is to select the optimal network for TN contraction to fit a tensor. In past literature and the view of many tensor researchers, the task is widely considered as the same as learning TN (model) ranks. In this manuscript, we briefly analyze the relation of these two critical tasks, stating that rank minimization is actually a subtopic of the structure learning for TN, with ignoration of the graph essence of TN structures. On one hand, we agree that the vanilla structure learning task *amounts to* rank minimization of TN associated to a complete graph. On the other hand, we propose an open problem referred as to *permutation learning*, a variant of the structure learning constrained by a graph, to point out that rank minimization would fail in this case, due to its limitation of exploring graph spaces. We last focus on the permutation learning and offer several preliminary results to help understand this unexplored task.

1 Introduction

Tensor network (TN) is recently paid much attention by researchers in various scientific fields, including computer science, signal processing, physics and applied mathematics. In these fields many problems involve constructing low-dimensional representations for extremely-high-dimensional functions or data. The framework of TN exhibits the potential from an algebraic perspective to satisfy this requirement. And more importantly there have been rich results as to both algorithms and theory to demonstrate the efficiency of TN in real-world applications. Thanks to the flexibility of modeling language like the diagram notation [9], TN is also expected to achieve the similar express capability as the well-known deep neural networks [11].

The flexibility of TN, on the other hand, leads to a severe challenge on structure learning, *i.e.*, the model selection for TN, in practice. It has mentioned in our previous work [8] that there would be more *68 billion* possible structures only for fitting an order-9 tensor, even though the (model) ranks (also known as bond dimension) are fixed. As a consequence, it yields an intractable task that learns the optimal structure from the vast number of candidates.

Rank minimization of TN is reckoned the essence to tackle the structure learning task by some existing works to our knowledge. Indeed, the structure learning can be addressed by minimizing the (model) ranks from a "fully-connected" TN [16] associated to a complete graph. Since the edges — of a simple graph used to formulate TN — can be harmlessly discarded if the associated ranks equal one [8, 13], the TN structure is thus learned by such operations.

Are rank minimization and structure learning therefore two sides of the same coin? Maybe not in some cases. In this manuscript, state that we cannot handle all possible structure learning problem of TN by only rank minimization, particularly under graph constraint. We support the statement

*corresponding author

Second Workshop on Quantum Tensor Networks in Machine Learning (QTNML), 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

by proposing an open problem called *permutation learning* of TN. In the new problem we learn the optimal mapping from tensor modes [6] to vertices (also known as core tensors in literature [1, 2, 3]). We point out that rank minimization fails to solve this problem in general since it ignores the inherent graph nature of TN structures, which contain the important information in respect of relations among vertices.

In the rest of the manuscript, we first review the notions of tensor network and its rank minimization task. After that, we attempt to analyze the relation between rank minimization and the structure learning of TN by answering the questions: *when and why* the structure learning can be tackled by rank minimization? and why such the fashion gets less effective on permutation learning? Last but not least, we focus on the permutation learning and offer several results to shed brief light on a promising direction to address the issue.

2 Tensor network and its rank minimization

We consider an *order-N tensor* as a multi-dimensional array of real numbers represented by $\mathcal{X}_{i_1,i_2,\ldots,i_N} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$, where $\otimes$ denotes the tensor product, the indices i_m, $m \in [N]$ corresponds to mth (tensor) *mode* associated to the vector space $\mathbb{R}^{I_m}$, and $[N]$ denotes a set of integers from 1 to N. Sometimes we also use $\mathcal{X}$ by ignoring those indices to denote the same tensor for brevity. *Tensor contraction*, a binary operation on tensors, is defined as a matrix-like multiplication of two tensors under given indices. The operation details refer to the work [9].

Intuitively, a *tensor network (TN)* is a set of tensors, each of which can be represented as a sequence of tensor contraction of atomic tensors, dubbed vertices in the manuscript, associated to vector spaces $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \ldots, \mathbb{R}^{I_N}$ and a labelled simple graph $(G, f_r) = (V, E, f_r)$, where V, E, f_r denote the set of vertices, edges and the labelling function on E, respectively. Note that the function $f_r : E \to \mathbb{Z}_+$ labels each edge of the graph G, called (network) *topology*, with a positive integer, of which a collection over all edges is referred to as the *(model) ranks* of TN. Formally, we apply the well-defined expression $TNS(G, f_r, \mathbb{R}^{I_1}, \ldots, \mathbb{R}^{I_N})$ in the work [13] to representing such a tensor network. If the vector spaces $\mathbb{R}^{I_n}, n \in [N]$ are unimportant in the task, we also simplify the expression by $TNS(G, f_r)$ without confusion.

Note that the definition of the (model) ranks of a TN differs from those introduced by Ye and Lim [13], including G-ranks, border ranks and generic G-ranks, although they are strongly close to each other. In contrast to them, the model ranks reflect the *degree of freedom* of TN as a function, rather than an algebraic property of a specific tensor itself. Letting a tensor $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$, the *rank minimization* for TN associated to graph G is defined as solving a multi-objective optimization problem as follows:

$$\min_{f_r \in \mathbb{F}_r} f_r(E), \quad s.t.\, \mathcal{X} \in TNS(G, f_r), \tag{1}$$

where $\mathbb{F}_r$ denotes the set of all feasible labelling functions in our scenario. Note that the optimization (1) is actually to learn the G-ranks of TN defined in the work [13], and from the machine learning perspective the aim of (1) is to select the smallest degree of freedom for TN, in turn yielding much bias yet less variance for prediction.

3 Rank minimization loses the graph information of TN structures

Structure learning of tensor network (TN), falling under the general heading *model selection*, is to learn the most suitable TN models, composed of a graph G and the rank-related labelling function f_r, which performs well in practice. As aforementioned that a model with smaller degree of freedom is more preferable (*i.e.*, suitable) in general because of the "bias–variance" balance. The structure learning is therefore connected from the rank minimization of TN in a natural manner.

For convenience, we focus on a vanilla tensor fitting problem in the manuscript. Supposing a tensor $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$, the associated structure learning is to solve the following optimization problem [8]:

$$\min_{G \in \mathbb{G}, f_r \in \mathbb{F}_r} \phi(G, f_r), \quad s.t.\, \mathcal{X} \in TNS(G, f_r), \tag{2}$$

where $\mathbb{G}$ denotes the set consisting of all possible simple graphs with N vertices and $\phi(\,\cdot\,)$ is any fitness function over TN structures, but generally the compression ratio, number of parameters or

even directly the ranks. We can observe from (1) and (2) that they are "visually" close to each other especially when the fitness function is supposed to be the model ranks. Meanwhile, Their difference is also readily apparent in terms of formulas. Searching the graph G from space $\mathbb{G}$ is taken into account in structure learning, while only the ranks are considered in the rank minimization task.

What does the discrepancy mean when comparing the two tasks? We will see, if there is no structural constraint on $\mathbb{G}$, the discrepancy offers no additional information since

Proposition 1. *Let the fitness function in* (2) *be the same as in* (1)*, i.e.,* $\phi(G, f_r) = f_r(E)$*, then the optimization problem* (2) *is equivalent to* (1) *if the graph* G *in* (1)*, or its complement, is complete.*

It is proved by the fact that the edges labelling as "1" can be harmlessly discarded from G [13]. The equivalence between the two tasks motivates most recent works like [4, 5] to learning TN structures by ranks. Indeed as such, various algorithms for matrix rank minimization— such as *forward/backward-stepwise searching, shrinkage methods* and *Bayesian inference*— can be naturally extended into their tensor forms.

3.1 Learning TN structure under a graph constraint

However such the equivalence would be invalid if the graph space $\mathbb{G}$ in (2) is constrained — leading to a *graph-constrained structure learning* task of TN. We explain it by a specific yet practically important example: suppose an order-4 tensor without loss of generality, and we apply the tensor ring (TR, [14]) to the approximation by the four vertices, as illustrated in Figure 1. Obverse that, not only TR-ranks, we also need to select a suitable mapping from the tensor modes, *i.e.,*, A, B, C, and D, to vertices of the TR to finally determine the structure. In this case, we will have three different mappings, including "A->B->C->D", "A->C->B->D", "A->B->D->C" three different sequences. We call the problem to select the optimal from those mapping the "*permutation learning*" of TN. The word "permutation" reveals that those mappings essentially amount to permutations of vertices.

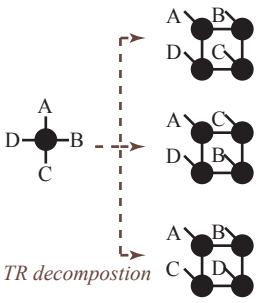

Figure 1: Which tensor ring (TR) is the optimal?

The vanilla rank minimization of TN would *fail* in the above example. Due to the Fixed the graph G in (1) it bonds the relation between tensor modes and vertices in TN. In turn, rank minimization is only able to search the optimal label for each edge or to check whether the edges can be discarded or not. No permutation can be learned in this form. One may argue that the permutation learning can be tackled by trivially solving (2) or equivalently by minimizing the ranks from a TN with a complete graph K_4, as aforementioned. It is reasonable if the tensor owns an inherent low-rank TR structure. But this assumption is generally *not true* for most of real-world data. In practice there is no guarantee that the solutions of (1) or (2) obey the desired TR format, in turn failing in permutation learning of TN.

Remark. One may also argue, from a algorithmic view, that the existing methods for learning TN structures without constraints would be tackle the task of permutation learning within slight modification. "The greedy method like [4] can preserve the topology of the solution in a straightforward manner by limiting the search space of the "find-best-edge" subroutine", one reviewer said in the comment. We agree if the "template" graph is as simple as a cycle graph mentioned in the example. But the first author of the manuscript reckon that there are still two issues that are deserved to discuss: first, how to efficiently restrict the greedy trajectory into a given complex template graphs, which would be non-symmetric and disconnected almost everywhere? second, how to ensure the algorithm avoiding those solutions, which would be even not stationary points? It is also worthwhile to note that limiting the greedy trajectory has been a way to take the graph information into account, which gets different from the trivial rank minimization defined in (1). All we agree that the greedy algorithms would be a promising way for permutation learning, although this two questions seem unanswered to date.

Although it remains open on theory, the above example partially reflects the limitation of solving TN structure learning only by rank minimization. The insight hiding behind this example is that *only concerning the ranks would lose the important graph information of TN structures*. In other words, the irregularity imposed by the graph constraint cannot be trivially modelled by the problem (1)

with a complete, unless further constructing a sophisticated constraint on $\mathbb{F}_r$ associated to the graph constraint.

4 Preliminary results on the permutation learning for TN

we last give several elementary results for analyzing the proposed permutation learning problem, where we focus on a question that how the network topology, as the cycle graph C_4 in the preceding example, impacts the search space of the optimization (2).

We first shed light on the relationship between permutation learning and a general structure learning for TN. We see that learning the permutation is a variant of the structure learning under a graph constraint. As known from the example, the tensor ring (TR) assumption limits the original search space $\mathbb{G}$ of (2) into its subset encompassing all *automorphisms* from the cycle graph C_4. Thus we have an explicit form for the *permutation (and rank) learning* associated to any network topology G_0 as

$$\min_{G\in\mathbb{G}_0, f_r\in\mathbb{F}_r} \phi(G, f_r), \quad s.t.\, \mathcal{X} \in TN(G, f_r),\, \mathbb{G}_0 = \{G \in \mathbb{G} | G \cong G_0\} \tag{3}$$

where $\cong$ denotes the graph isomorphism. We see that if $G_0 = K_N$ then the permutation learning is identical to the structure learning (2), and as well to the rank minimization with $G = K_N$ in (1). In nontrivial cases such as $G_0 \neq K_N$, however, G_0 constrains the search space into a proper subset. It is therefore interesting to know how G_0 impacts the scale of the search space? In doing so, we suppose a graph $G_0 = (V, E_0)$ with N vertices and the labelling functions $\mathbb{F}_r$, related to the ranks, which is bounded by $R > 0$. Denote the search space of (3) by $\mathbb{L}_{G_0,R}$, *i.e.*, $\mathbb{L}_{G_0,R} := \mathbb{G}_0 \times \mathbb{F}_R$, where $\times$ is the direct product. The scale, also known as cardinality, of $\mathbb{L}_{G_0,R}$ is thus given as

$$\log |\mathbb{L}_{G_0,R}| = |E_0| \log R + \log N! - \log |Aut(G_0)|, \tag{4}$$

where $|\cdot|$ denotes the cardinality of a set, $\log(\cdot)$ denotes the natural logarithm and $Aut(G_0)$ denotes the *graph automorphisms* of G_0. Specifically with typologies for those practical TNs, we have

1. **Tensor train** [10]: $\log |\mathbb{L}_{P_N,R}| = (N-1)\log R + \log(N!) - \log 2$
2. **Tensor ring** [15]: $\log |\mathbb{L}_{C_N,R}| = N\log(R) + \log(N-1)! - \log 2$
3. **Tensor tree** [13]: $(N-1)\log R + \log N \leq \log |\mathbb{L}_{T_N,R}| \leq \log |\mathbb{L}_{P_N,R}|$
4. **PEPS** [12] : $\log |\mathbb{L}_{L_{m,n}}| \leq (2mn - m - n)\log R + \log(mn)! - \log 4$,

where in PEPS, Projected Entangled Pair States [12], m, n denotes the number of vertices of rows and columns for a two-dimensional lattice and $N = mn$ in this case. We have two observations from these results. First, Fixed the number of vertices N, the scale of the search space is determined by not only $|E_0|$, the number of edges, but also the automorphism term $|Aut(G_0)|$, which reflects the symmetry of G_0. Second, in those practical TNs the scale is mainly dominated by two terms, $N\log(R)$ and $\log(N!)$, which correspond to the first two terms on the right-hand side of Eq. (4), respectively. Surprisingly, the impact by the third term of Eq. (4) *w.r.t.* the automorphism of G_0 seems relatively *weak* in these cases! Does this fact universally exist in all TNs? We use the following proposition to rigorously answer the question that the affect by the automorphism of G_0 is generally ignorable for TNs, of which the topology G_0 is degree-bounded.

Proposition 2. *Assume that $G_0 = (V, E_0)$ with N vertices is connected graph and its maximum degree Δ_{G_0} is a constant that is far less than N, then we have*

$$\log |\mathbb{L}_{G_0,R}| = \Omega\left(N\log R + N\log N\right), \tag{5}$$

where $\Omega(\cdot)$ denotes the big-Ω notation, which means that there exist $N_0, R_0 > 0$ and a constant $M > 0$, such that for any $N > N_0$ and $R > R_0$ we have the inequality $\log(|\mathbb{L}_{G_0,R}|) \geq M \cdot (N\log R + N\log N)$

The proof is given as the appendix. As a sketch, the result is proved by bounding the both $|E_0|$ and $|Aut(G_0)|$ in Eq. (4) by the maximum degree Δ_{G_0} using the Handshaking lemma known in graph theory and Theorem 2 given in [7], respectively.The assumption of the bounded graph degree can be satisfied in most of practical TNs since those vertices are expected to be bounded by a constant in general.

The ignorablity of the automorphism of G_0 reveals an important fact: a symmetric group $\mathbb{S}_N$ of order N would be a "good" alternative to $\mathbb{G}_0$, such that the overall search space $\mathbb{L}_{G_0,R}$ can be embedded into a algebraic group, of which we have a clearly defined operation and rich theoretical tools. As a potential consequence, they would help develop optimization algorithms to computationally address the permutation learning task of TN.

5 Conclusion

We use the permutation learning, a specific yet practically important variant of the structure learning of tensor network (TN), to illustrate why and when learning TN structure differs from the rank minimization task. In the manuscript, we have seen that the the structure learning is equivalent to rank minimization with a complete graph, up to the objective, if there is no structural assumption on the graph space $\mathbb{G}$. Otherwise, the rank minimization model would fail due to the irregular essence of the search space by the graph constraint. We also briefly discuss the properties of the permutation learning, suggesting that in many practical TNs a symmetric group has been sufficiently good as an alternative to the graph part of the search space for the computational purpose. We expect these results can help solve the permutation learning problem in a near future.

Acknowledgments and Disclosure of Funding

We thank the anonymous (meta-)reviewers of QTNML 2021 for their insightful comments. The work is partially supported by JSPS KAKENHI (Grant No. 20K19875, 20H04249, 20H04208) and the National Natural Science Foundation of China (Grant No. 62006045).

References

[1] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[2] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[3] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9(6):431–673, 2017.

[4] Meraj Hashemizadeh, Michelle Liu, Jacob Miller, and Guillaume Rabusseau. Adaptive tensor learning with tensor networks. *arXiv preprint arXiv:2008.05437*, 2020.

[5] Maxim Kodryan, Dmitry Kropotov, and Dmitry Vetrov. Mars: Masked automatic ranks selection in tensor decompositions. *arXiv preprint arXiv:2006.10859*, 2020.

[6] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[7] I Krasikov, A Lev, and BD Thatte. Upper bounds on the automorphism group of a graph. *Discrete Math.*, 256(math. CO/0609425):489–493, 2006.

[8] Chao Li and Zhun Sun. Evolutionary topology search for tensor network decomposition. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[9] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[10] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[11] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[12] Frank Verstraete and J Ignacio Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.

[13] Ke Ye and Lek-Heng Lim. Tensor network ranks. *arXiv preprint arXiv:1801.02662*, 2019.

[14] Qibin Zhao, Masashi Sugiyama, Longhao Yuan, and Andrzej Cichocki. Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8608–8612. IEEE, 2019.

[15] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[16] Yu-Bang Zheng, Ting-Zhu Huang, Xi-Le Zhao, Qibin Zhao, and Tai-Xiang Jiang. Fully-connected tensor network decomposition and its application to higher-order tensor completion. 2021.

Appendix

A Proof of Proposition 2

Proof. According to Eq. (4) we have the following lower bound of the cardinality of $\mathbb{L}_{G_0,R}$, that is,

$$\log|\mathbb{L}_{G_0,R}| \geq |E_0|\log R + \log(N-1)! - \log\Delta_{G_0}! - (N-\Delta_{G_0}-1)\log(\Delta_{G_0}-1), \tag{6}$$

The inequality is held by the relation in Theorem 2 of the work [7]:

$$\log(Aut(G_0)) \leq \log N + \log\Delta_{G_0}! + (N-\Delta_{G_0}-1)\log(\Delta_{G_0}-1), \tag{7}$$

if G_0 is connected. Meanwhile, we use the handshaking lemma in graph theory to bound $|E_0|$ by

$$|E_0| \geq \frac{1}{2}\sum_{v\in V} deg(v) \geq \frac{1}{2}N\delta_{G_0}, \tag{8}$$

where deg denotes the degree of a vertex and $\delta_{G_0} \leq \Delta_{G_0}$ is the minimum degree of G_0. We also bound the factorial terms in (6) by the Stirling' approximation as

$$\begin{aligned}\log\Delta_{G_0}! &\leq 0.5\log 2\pi + (\Delta_{G_0}+1/2)\log\Delta_{G_0} - \Delta_{G_0} + \frac{1}{12\Delta_{G_0}},\\ \log N! &\geq 0.5\log 2\pi + (N+1/2)\log N - N + \frac{1}{12N+1}.\end{aligned} \tag{9}$$

Substitute (8) and (9) in to (6), then we have

$$\log|\mathbb{L}_{G_0,R}| \geq \frac{1}{2}\delta_{G_0}N\log R + (N-0.5)\log(N-1) - \alpha N + \beta, \tag{10}$$

where α, β denotes two constant with respect to Δ_{G_0}. Since δ_{G_0} is upper bounded by the constant Δ_{G_0}, we easily have

$$\log|\mathbb{L}_{G_0,R}| = \Omega\left(N\log R + N\log N\right), \tag{11}$$

by the definition of the big-Ω notation. □