Random Tensor Theory for Tensor decomposition

Mohamed Ouerfelli[1,2]**, Mohamed Tamaazousti** [1]**, Vincent Rivasseau** [2]

[1]Université Paris-Saclay, CEA, List, F-91120 Palaiseau, France
[2]Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France
Mohamed-oumar.ouerfelli@cea.fr, Mohamed.tamaazousti@cea.fr, Vincent.rivasseau@th.u-psud.fr

Abstract

We propose a new framework for tensor decomposition based on trace invariants, which are particular cases of tensor networks. In general, tensor networks are diagrams/graphs that specify a way to "multiply" a collection of tensors together to produce another tensor, matrix or scalar. The particularity of trace invariants is that the operation of multiplying copies of a certain input tensor that produces a scalar obeys specific symmetry constraints. In other words, the scalar resulting from this multiplication is invariant under some specific transformations of the involved tensor. We focus our study on the O(n)-invariant graphs, i.e. invariant under orthogonal transformations of the input tensor. The proposed approach is novel and versatile since it allows to address different theoretical and practical aspects of both CANDECOMP/PARAFAC (CP) and Tucker decomposition models. In particular we obtain several results: (i) we generalize the computational limit of Tensor PCA (a rank-one tensor decomposition) to the asymmetric case (ii) we introduce new algorithms for both decomposition models (iii) we obtain theoretical guarantees for these algorithms and (iv) we show improvements with respect to state of the art on synthetic and real data which also highlights a promising potential for practical applications.

Introduction

Tensor PCA is a model introduced in (Richard and Montanari 2014) that is formally equivalent to the best rank-one approximation of a tensor. It consists in detecting and retrieving a spike $v_0^{\otimes k}$ associated to an unknown unit vector v_0 from noise-corrupted multi-linear measurements put in the form of a tensor T

$$\mathsf{T} = \beta v_0^{\otimes k} + \mathsf{Z}, \tag{1}$$

with $\mathsf{Z} \in (\mathbb{R}^n)^{\otimes k}$ a pure Gaussian noise tensor of order k and dimension n with identically independent distributed (iid) standard Gaussian entries: $\mathsf{Z}_{i_1,i_2,\ldots,i_k} \sim \mathcal{N}(0,1)$ and β is the signal-to-noise ratio.

Thus, it could be seen as a particular case (rank one) of both the CP and the Tucker tensor decomposition models, which are the two most notable generalizations of the matrix SVD to the tensor case and were recently successfully used to address important problems in unsupervised learning

(learning latent variable models, in particular latent Dirichlet allocation (Anandkumar et al. 2014, 2015)), supervised learning (training of two-layer neural networks, (Janzamin, Sedghi, and Anandkumar 2015)) and reinforcement learning (Azizzadenesheli, Lazaric, and Anandkumar 2016). However, despite this formal and well known correspondence between tensor PCA and Tensor decomposition, it is important to note that the respective literature of both subjects have quite different motivations.

Indeed, even if there are some practical applications of Tensor PCA (e.g. in telecommunications (Decurninge, Land, and Guillaud 2020)), the main motivation of its related literature is theoretical. In particular, these works concern the study of statistical limits when the signal-to-noise ratio (SNR) is low (Lesieur et al. 2017; Arous et al. 2019, 2020; Perry et al. 2020; Jagannath et al. 2020) and the study of the gradient-based exploration of high dimensional landscapes through Tensor PCA (Ros et al. 2019a) or its simplified version, the Matrix-Tensor PCA (Mannelli et al. 2019; Sarao Mannelli et al. 2019; Mannelli et al. 2020). In opposite, even if there are some theoretical results in the tensor decomposition (e.g. (Wang and Anandkumar 2016), (Zhang and Xia 2018)), the main motivation of its related literature is the practical applications (Tensor faces (Vasilescu and Terzopoulos 2002), Hyperspectral imagery (Lin and Bourennane 2013), DNN compression (Astrid and Lee 2017), chemical materials (Sun and Braatz 2020), multimodal data fusion (Lahat, Adali, and Jutten 2015), data mining (Papalexakis, Faloutsos, and Sidiropoulos 2016), etc.).

We argue that these point of views are complementary. For example, the statistical study of the performance of a practical tensor decomposition algorithm is important since it opens the possibility to apply this algorithm in high-risk and safety-critical applications where we need quantitative guarantees or certifications. Thus, it is important to have the possibility to find new algorithms to address a large set of problems (i.e. symmetric or asymmetric tensor), with the best possible precision/accuracy while also being able to compute the statistical performances of the algorithms regarding the SNR.

Recently, a fundamentally different set of mathematical tools organised in the framework of Random Tensor Theory (RTT) have been developed for tensors in the context of high energy physics. In this paper, leverage these novel and valu-

able tools to introduce a new and powerful framework based on RTT to design and study algorithms for this large family of problems including Tensor PCA and CP/Tucker Tensor decomposition. More specifically, RTT consists of a set of combinatorial tools dedicated to the study of some classes of tensor networks, i.e. graphs that can be interpreted as tensor multiplications.

Our contribution In this study, we show that we can use this sharply distinct and versatile framework in order to:

- Generate a matrix from a certain graph (which represents a way to multiply tensors) such that its eigenvalues and eigenvectors encodes the information contained in the tensor himself.

- Show that the two best existing algorithms (tensor unfolding and homotopy) appears to be related to the simplest graphs in the proposed framework.

- From a graph called tetrahedral, we derived new algorithms for the targeted Tensor decomposition problems.

- Experimentation with these new algorithms show improvements for Tensor PCA in the symmetric and asymmetric cases

- More generally, we find a new phase transition for Tensor PCA in the asymmetric case.

- We provide experimentation on real data for Tucker decomposition and CP decomposition.

Notations We use bold characters $\mathbf{T}, M, v$ for tensors, matrices and vectors and T_{ijk}, M_{ij}, v_i for their components. $[p]$ denotes the set $\{1, \ldots, n\}$. A real k−th order tensor is of order k if it is a member of the tensor product of $\mathbb{R}^{n_i}, i \in [k]$: $\mathbf{T} \in \bigotimes_{i=1}^{k} \mathbb{R}^{n_i}$. It is symmetric if $T_{i_1 \ldots i_k} = T_{\tau(i_1) \ldots \tau(i_k)} \ \forall \tau \in \mathfrak{S}_k$ where $\mathfrak{S}_k$ is the symmetric group (more details are provided in Supplementary Material). $A^\top$ is the transpose of A and I_n the identity matrix. $O(n)$ is the set of orthogonal matrices (that verifies $AA^\top = I_n$). For a vector $v \in \mathbb{R}^n$, we use $v^{\otimes p} \equiv v \otimes v \otimes \cdots \otimes v \in \bigotimes^p \mathbb{R}^n$ to denote its p-th tensor power. $\langle v, w \rangle$ denotes the scalar product of v and w. The operator norm is equivalent to the highest eigenvalue of a tensor of any order: $\|\mathbf{T}\|_{\mathrm{op}} \equiv \max\{\mathbf{T}_{i_1, \ldots, i_k}(w_1)_{i_1} \ldots (w_k)_{i_k}, \forall i \in \{1, \ldots, n\}, \|w_i\| \le 1\}$. The trace of M is denoted $\mathrm{Tr}(M)$. We denote the expectation of a variable X by $\mathbb{E}(X)$ and its variance by $\sigma(X)$. We say that a function f is negligible compared to a positive function g and we write $f = o(g)$ if $\lim_{n \to \infty} f/g \to 0$.

Related work and positioning

Tensor decomposition: From the various generalizations of matrix SVD to tensors, there is two main tensor decompositions that have been successfully used in numerous applications: (i) CANDECOMP/PARAFAC (CP) decomposition that consists in approximating a tensor with a sum of rank-one tensors. (Anandkumar et al. 2014; Anandkumar, Ge, and Janzamin 2015; Wang and Anandkumar 2016). One of the most used algorithms is Alternative Least Squares (ALS)

(Comon, Luciani, and De Almeida 2009). (ii) Tucker decomposition that approximates the initial tensor with one small core tensor and a set of matrices (Zhang and Xia 2018). HOSVD (De Lathauwer, De Moor, and Vandewalle 2000a) and HOOI (De Lathauwer, De Moor, and Vandewalle 2000b) are the most popular algorithms for this model and their statistical limits have been studied in (Zhang and Xia 2018).

Tensor PCA: Tensor PCA was introduced by (Richard and Montanari 2014) where the authors suggested and analyzed different methods to recover the signal vector like tensor unfolding and power iteration. Since then, various other methods were proposed. Hopkins, Shi, and Steurer (2015) introduced algorithms based on the sum of squares hierarchy with the first proven algorithmic threshold of $n^{k/4}$. However this class of algorithm generally requires high computing resources and relies on complex mathematical tools (which makes its algorithmic optimization difficult). Other studied methods have been inspired by different perspectives like homotopy in Anandkumar et al. (2017), statistical physics (Arous et al. 2020; Ros et al. 2019b; Wein, El Alaoui, and Moore 2019; Biroli, Cammarota, and Ricci-Tersenghi 2020), quantum computing (Hastings 2020) as well as statistical query (Dudeja and Hsu 2020).

Moitra and Wein (2019) briefly mentioned in their paper that some existing methods like Homotopy and Unfolding can be viewed through tensor Networks, as spectral methods on matrices built from contracting (multiplying) the tensor. However, their contribution to Tensor PCA is summarized in that remark. By contrast, in our work we use the tensor invariant concept (the backbone of our framework) to select specific matrices obtained from restricted class of graphs and justify theoretically their relevance. This is reflected in the fact that our framework successfully introduced new and more accurate algorithms. Moreover the new theoretical results were built on combinatorial tools associated to tensor invariants and cannot be used in general tensor network.

On another side, Random Tensor Theory have been used in (Evnin 2020) to study the highest eigenvalue of a real symmetric Gaussian tensor. Subsequently, (Gurau 2020) provided a theoretical study on a function based on an infinite sum of these invariants. Their results suggest a transition phase for the highest eigenvalue of a tensor for β around $n^{1/2}$ in a similar way to the BBP transition in the matrix case (Baik et al. 2005). Thus, this function allows the detection of a spike. However evaluating it involves computing an integral over a n-dimensional space, which may not be possible in a polynomial time. Furthermore, the question of signal recovery was not studied.

The contribution of this paper is the use of these invariant tools to build tractable algorithms with polynomial complexity. In contrast to (Gurau 2020), instead of using a sum of an infinite number of invariants, we select one trace invariant with convenient properties to build our algorithm. In order to recover the signal vector besides simply detecting it, we introduce new tools in the form of matrices associated to this specific invariant.

Random Tensor Theory

Random Tensor Theory (RTT) provides a set of combinatorial tools dedicated to the study of trace invariant graphs (Gurau 2017). Trace invariants of a tensor $\mathbf{T} \in \bigotimes_{i=1}^{k} \mathbb{R}^{n_i}$ are tensor networks scalars that are invariant under the following $O(n_1) \times \cdots \times O(n_k)$ transformations:

$$\mathbf{T}_{i_1 \ldots i_k} \longrightarrow \mathbf{T}'_{i_1 \ldots i_k} = \sum_{j_1 \ldots j_k} O^{(1)}_{i_1 j_1} \ldots O^{(k)}_{i_k j_k} \mathbf{T}_{j_1 \ldots j_k}$$

RTT allows to obtain important probabilistic results on trace invariants by using simple enumerative combinatorics. In particular, it gives a simple way to compute the moments (expected value, variance, etc.) of the distribution of these scalars for random tensors. In the following, it should be understood from the context that

Einstein summation convention: It is important to keep in mind throughout the paper that we will follow the Einstein summation convention: when an index variable appears twice in a single term and is not otherwise defined, it implies summation of that term over all the values of the index. For example: $T_{ijk} T_{ijk} \equiv \sum_{ijk} T_{ijk} T_{ijk}$. It is a common convention when addressing tensor problems that helps to make the equations more comprehensible.

Trace invariants and their associated graphs

An important concept in problems involving matrices is the spectral theory. It refers to the study of eigenvalues and eigenvectors of a matrix and it is of fundamental importance in numerous areas. Equivalently, the traces of the n first matrix powers $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{A}^\top), \mathrm{Tr}((\boldsymbol{A}\boldsymbol{A}^\top)^2), \ldots, \mathrm{Tr}((\boldsymbol{A}\boldsymbol{A}^\top)^n)$ contain the same information as the eigenvalues (in absolute value) since each set can be inferred from the other through some basic algebraic operations.

In the tensor case, the concept of eigenvalue and eigenvector is ill-defined and not practical giving that the number of eigenvalues is exponential with the dimension n (Qi 2005; Cartwright and Sturmfels 2013) and computing them is very complicated. In contrast, we have a very convenient generalization of the traces of the power matrices for the tensors that we call trace invariants.

We first give a more formal definition of trace invariants. Let $\mathbf{T}$ be a tensor whose entries are $T_{i_1, \ldots, i_k}$. Let's define a contraction of a pair of indices as setting them equal to each other and summing over them, as in calculating the trace of a matrix ($\boldsymbol{A}_{ij} \to \sum_{i=1}^{n} \boldsymbol{A}_{ii}$). The trace invariants of the tensor $\mathbf{T}$ correspond to the different ways to contract pairs of indices in a product of an even number of copies of $\mathbf{T}$. The degree of the trace invariants consists in the number of copies of $\mathbf{T}$ contracted. For example, $\sum_{i_1, i_2, i_3} T_{i_1 i_2 i_3} T_{i_1 i_2 i_3}$ and $\sum_{i_1, i_2, i_3} T_{i_1 i_2 i_2} T_{i_1 i_3 i_3}$ are trace invariants of degree 2.

A trace invariant of degree d of a tensor $\mathbf{T}$ of order k admits a practical graphical representation as an edge colored graph $\mathcal{G}$ obtained by following two steps: we first draw d vertices representing the d different copies of $\mathbf{T}$. The indices of each copy is represented by k half-edges with a different color in $\{1, \ldots, k\}$ for each index position as shown in Figure 1a. Then, when two different indices are contracted

in the tensor invariant, we connect their corresponding half-edges in $\mathcal{G}$. Reciprocally, to obtain the tensor invariant associated to a graph $\mathcal{G}$ with d vertices, we take d copies of $\mathbf{T}$ (one for each vertex), we associate a color for each index position $\{1, \ldots, k\}$, and we contract the indices of the d copies of $\mathbf{T}$ following the coloring of the edges connecting the vertices. We denote this invariant $I_{\mathcal{G}}(\mathbf{T})$. Three important examples of trace invariants worth mentioning are: the melon diagram (Figure 1b), the tetrahedral (1c) and the tadpole (1d). (Avohou, Ben Geloun, and Dub 2020) provides a thorough study about the number of trace invariants for a given degree d.

Combinatorial tools

In order to be able to compute the moments of trace invariants in a simple way, we introduce the concept of covering graph used in (Gurau 2014): a covering graph of $\mathcal{G}$ consists in adding $d/2$ new edges of color 0 (also called propagators) relying pairwise the vertices of $\mathcal{G}$. If we denote $\mathcal{E}^0(\mathcal{G})$ the edges of color 0 of a graph $\mathcal{G}$, then $\{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}\}$ denotes the graphs which restrict to the graph $\mathcal{G}$ when we remove their edges of color 0. These are by the definition the covering graphs.

Let $c_1, c_2 \in \{0, \ldots, d\}$ be two different colors of edges. We denote $F^{c_1, c_2}(\mathcal{G})$ the number of closed cycles (that we also call faces) of 2 colors of $\mathcal{G}$. More explicitly, it consists of the number of connected subgraphs left when we keep in $\mathcal{G}$ only the edges of colors c_1, c_2.

Then, we have the formula:

$$\mathbb{E}(I_{\mathcal{G}}(\mathbf{T})) = \sum_{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}} n^{\sum_c F^{0,c}(\mathcal{G}')}. \qquad (2)$$

This will be the expression (of enumerative combinatorial nature) that we will use to calculate the expectations of our graphs. The details of its derivation are given in the supplementary material.

Random Tensor Theory for Tensor Decomposition

In this section, we will first demonstrate how to recover the signal in the Tensor PCA model using trace invariants, then we will indicate how to generalize the method to address the CP and Tucker Tensor decomposition models. But first we give the general idea of the proposed framework. As we previously explained, generalizing eigenvalues and eigenvectors to the tensor case is not convenient. Thus, our approach is to associate a matrix to the tensor of interest in order to recover the signal by exploiting the well mastered spectral theory of matrices. However, there is two main important characteristics for these matrices that are required for them to be of interest: they have to be relevant, in the sense that they should reveal the information/signal hidden in the tensor even in low signal regime, and they also have to be easy to study from a probabilistic point of view in order to provide theoretical guarantees. Conveniently, RTT allows us to select matrices that meet these requirements. Indeed, we provide matrices that are able to obtain the signal in the

(a) T_{ijk} (b) $T_{ijk}T_{ijk}$ (c) $T_{ijk}T_{ij'k'}$ $T_{i'jk'}T_{i'j'k}$ (d) $T_{ijj}T_{ikk}$
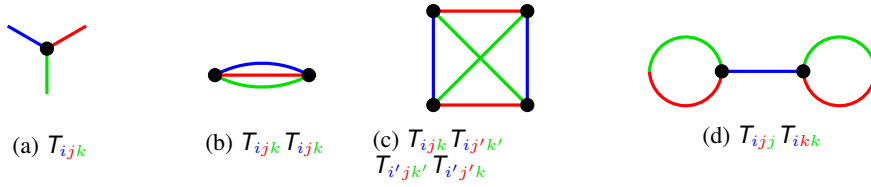
Figure 1: Example of graphs and their associated invariants.

high noise regime, and we have access to simple enumerative combinatorial tools in order to have theoretical guarantees for their performance.

Matrices associated trace invariants

Given that our the objective of Tensor PCA is to recover the signal, we should find mathematical objects that are able to provide a vector. To this effect, we introduce in this paper a new set of tools in the form of matrices. We denote by $M_{\mathcal{G},e}$ the matrix obtained by cutting an edge e of a graph $\mathcal{G}$ in two half edges (see Figure 2 for an example). Indeed, this cut amounts to not summing over the two indices i_1 and i_2 associated to these two half-edges and using them to index the matrix instead. We will drop the index $\mathcal{G}, e$ of the matrix when it is clear. Advantageously, we can compute the operator norms of these matrices using the same tools described above.
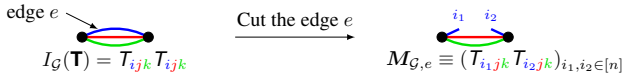

Figure 2: Obtaining a matrix by cutting the edge of a trace invariant graph $\mathcal{G}$.

Retrieving the signal using Random Tensor Theory

In Tensor PCA, we can represent the tensor from which we hope to extract the signal graphically as:

$$T_{ij_1\ldots j_{k-1}} = \beta\, v_i v_{j_1}\ldots v_{j_{k-1}} + Z_{ij_1\ldots j_{k-1}}$$

An identical decomposition can be carried out for the matrix based on this. Let's consider a tensor $\mathbf{T}$, a graph $\mathcal{G}$ and its associated trace invariant $I_{\mathcal{G}}(\mathbf{T})$. Let's denote $I'_{\mathcal{G}}(\mathbf{T})$ the invariant associated to the subgraph obtained by removing from $\mathcal{G}$ the edge e and its two vertices. We can distinguish three kind of contributions to the matrix $M_{\mathcal{G},e}$ that we denote $M_{\mathcal{G},e}^{(N)}$, $M_{\mathcal{G},e}^{(D)}$ and $M_{\mathcal{G},e}^{(R)}$, illustrated in Figure 3 (where we denoted the invariant $I'_{\mathcal{G}}(\mathbf{T})$ by I' and dropped the index $\mathcal{G}, e$ for simplicity).

Lemma 1 $\mathbb{E}(M^{(N)}) = \frac{\mathbb{E}(I_{\mathcal{G}}^{(N)})}{n}I_n.$

Using the lemma 1, we identify two possible phases depending on which matrix operator norm is much larger than the others: **(i) No recovery:** If $\left\|M^{(N)} - \mathbb{E}(M^{(N)})\right\|_{\text{op}} \gg \left\|M^{(D)}\right\|_{\text{op}}, \left\|M^{(R)}\right\|_{\text{op}}$ then we can't distinguish if there is

a signal. It is for example the phase for $\beta \to 0$. **(ii) Recovery:** $\left\|M^{(R)}\right\|_{\text{op}} \gg \left\|M^{(N)} - \mathbb{E}(M^{(N)})\right\|_{\text{op}}, \left\|M^{(D)}\right\|_{\text{op}}$. We recover the signal vector. It is for example the phase for $\beta \to \infty$.

Algorithmic Threshold for a General Graph

We can now state the important algorithms that will be essential for this paper. It is important to keep in mind that the following claims concern the large n limit. Empirically, the approximation of large n limit seems valid for $n > 25$.

The proposed algorithm is able to recover the spike in a tensor $\mathbf{T}$ through the construction of the matrix of size $n \times n$ $M_{\mathcal{G},e}(\mathbf{T})$ associated to a given graph $\mathcal{G}$ and edge e.

Algorithm 1 Recovery algorithm associated to the graph $\mathcal{G}$ and edge e

---

Input: The tensor $\mathbf{T} = \beta\mathbf{v}^{\otimes k} + \mathbf{Z}$
Goal: Estimate $\mathbf{v}_0$.
Calculate the matrix $M_{\mathcal{G},e}(\mathbf{T})$
Compute its top eigenvector by matrix power iteration (repeat $v_i \leftarrow M_{ij}v_j$).
Output: Obtaining an estimated vector $\mathbf{v}$

Theorem 2 *Let $\mathcal{G}$ be a graph of degree d, $\exists \;\; \beta_{rec} > 0$ so that Algorithm 1 gives an estimator $\mathbf{v}$ so that $\mathbf{v}$ is strongly correlated to $\mathbf{v}_0$ ($\langle\mathbf{v},\mathbf{v}_0\rangle > 0.9$) for $\beta \geq \beta_{rec}$.*

Since the algorithm 1 consists in algebraic operations on the tensors entries, it is very suitable for a parallel architecture (for example by computing independently each entry of the matrix $M_{\mathcal{G},e}(\mathbf{T})$). The Theorem 3 gives a lower bound to the threshold above which we can recover a spike using a single graph. Interestingly, this threshold which appears naturally in our framework, matches the threshold below which there is no known algorithm that is able to recover the spike in polynomial time. We call the Gaussian variance of a graph $\mathcal{G}$, the variance of the invariant $I_{\mathcal{G}}(\mathbf{B})$ where B_{ijk} are Gaussian random.

Theorem 3 *Let $k \geq 3$. It is impossible to detect or recover the signal using a single graph below the threshold $\beta \leq n^{k/4}$ which is the minimal Gaussian variance of any graph $\mathcal{G}$.*

Using these algorithms, we are now able to investigate the performance of our framework in various theoretical settings. In the first two subsections, we study the algorithms associated to two trace invariants of degree 2. They consist of the melonic diagram and the tadpole diagram.

$$M_{i_1 i_2} = M^{(N)}_{i_1 i_2} + M^{(D)}_{i_1 i_2} + M^{(R)}_{i_1 i_2}$$

$$= \mathbf{Z}_{i_1..}\mathbf{Z}_{i_2..}\ldots + \mathbf{Z}_{i_1..}\mathbf{Z}_{i_2...}\ldots + (v_{i_1}\mathbf{Z}_{i_2..}\ldots + v_{i_2}\mathbf{Z}_{i_1..}\ldots) + v_{i_1}v_{i_2}\ldots$$
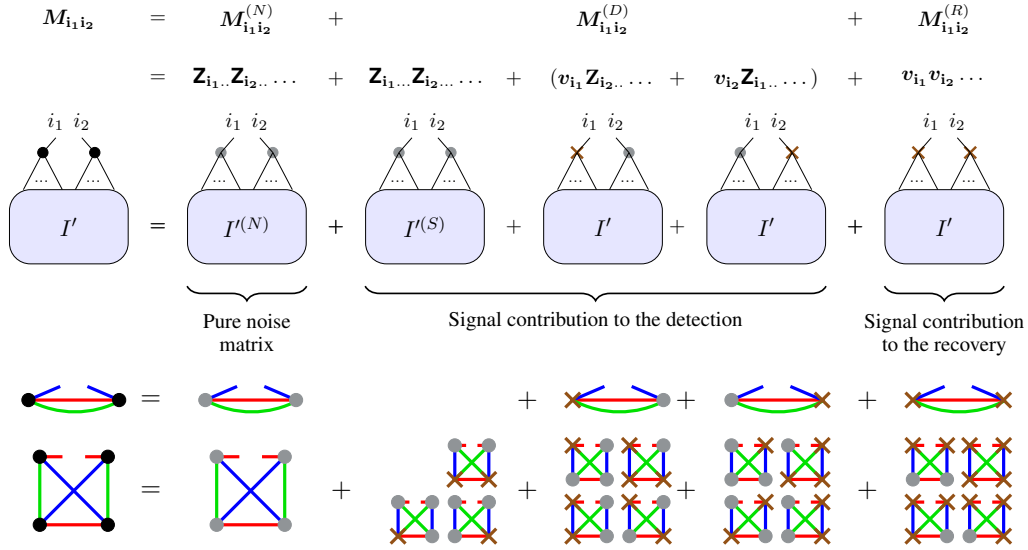

Figure 3: Decomposition of a matrix graph and the melon example

The third subsection goes further in terms of graph's degree and investigate the algorithms associated to the perfect one-factorization graphs (consisting in the tetrahedral when $k = 3$). The last two sections are an illustration of the versatility of this framework. We study the case where the dimensions n_i of the tensor $\mathbf{T}$ ($\mathbf{T} \in \bigotimes_{i=1}^{k} \mathbb{R}^{n_i}$) are not necessarily equal, which is important for practical applications where the dimensions are naturally asymmetric. Our methods allows us to derive a new algorithmic threshold for this case and we provide generalizations of this method for tensor decomposition.

The melon graph and the tadpole graph corresponds to existent algorithms

Let's consider the invariant $T_{i_1\ldots i_n}T_{i_1\ldots i_n}$ (illustrated by the graph in Figure 1b when $k = 3$). Its recovery algorithm (with the matrix obtained by cutting any of the edges) is similar to the tensor unfolding method presented in Richard and Montanari (2014).

Theorem 4 *The algorithms 1 work for the melon graph with $\beta_{rec} = O(n^{k/4})$ in linear time and $O(n^2)$ memory requirement.*

Figure 1d has a special characteristic: we can obtain two disconnected parts by cutting only one line. Therefore, the matrix obtained by cutting that edge is of rank one (in the form of $\boldsymbol{vv}^\top$). Thus, the vector $\boldsymbol{v}$ has a weak correlation with the signal $\boldsymbol{v}_0$, which allow the tensor power iteration ($v_i \leftarrow T_{ijk}v_jv_k$) to empirically recover it (formal proofs require to consider some more sophisticated variants of power iteration like in (Anandkumar et al. 2017) and (Biroli, Cammarota, and Ricci-Tersenghi 2020)). This algorithm is a variant of the already existent homotopy algorithm.

Theorem 5 *The tadpole graph allows to recover the signal vector for $k \geq 3$ and $\beta = O(n^{k/4})$ by using local algorithms to enhance the signal contribution of the vector T_{ijj}.*

Tensor invariant based on the perfect one-factorization graphs

The theoretical physics community that had developed the theory of trace invariants for tensor have made a particular focus on a family of graphs called the perfect one-factorization graphs (Ferrari, Rivasseau, and Valette 2019) (more details in supplementary material). This focus is motivated by their nice combinatorial properties due to their symmetries. It was then natural for us to explore the algorithmic potential of these graphs in our tensor decomposition context. Our first candidate was the simplest next graph which is of degree strictly superior to two named tetrahedral graph. Our investigation through the tetrahedral shows that for $k = 3$, the algorithms based on the tetrahedral graph shows a very interesting improvement of empirical results and thus highlights the richness of the proposed framework. Moreover, the properties of this family of graphs also simplify the proofs for recovery theorems. Therefore, the standard methods involved in the demonstrations of these theorems are instructive for the study of more general graphs.

Theorem 6 *The algorithm associated to a perfect one-factorization graph is able to recover the signal vector for $\beta = O(n^{k/4})$.*

Handling a spike with different dimensions

We consider the more general case where the tensor $\mathbf{T}$ has axes of different dimensions n_i ($\mathbf{T} \in \bigotimes_{i=1}^{k} \mathbb{R}^{n_i}$). We can assume without any loss of generality that $n_1 \geq n_2 \geq \cdots \geq n_k$.

$$\mathbf{T} = \beta \boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_k + \mathbf{Z} \quad \text{where} \quad \boldsymbol{v}_i \in \mathbb{R}^{n_i}, \quad n_i \in \mathbb{N}. \tag{3}$$

Our framework naturally handles this case and allows us to derive a new algorithmic threshold. It is, to the best of our knowledge, the first generalization of the threshold $\beta =$

$n^{k/4}$ derived in (Richard and Montanari 2014) when $n_i = n \ \forall i \in [k]$ and appears easily using our framework.

Theorem 7 *Using the melon graph, the threshold for $\boldsymbol{v}_1$ is given by* $\max\left((\prod_{i=1}^{k} n_i)^{1/4}, n_1^{1/2}\right)$ *while the thresholds for $\boldsymbol{v}_j$, $j \geq 2$ are equal to $(\prod_{i=1}^{k} n_i)^{1/4}$.*

Tensor decomposition

Adaptation to low-rank CP decomposition: We consider a symmetric tensor with multiple orthogonal spikes.

$$\mathsf{T} = \sum_{i=1}^{p} \beta_i \boldsymbol{v}_i^{\otimes k} + \mathsf{Z} \quad \text{where} \quad \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \quad \forall i \neq j. \quad (4)$$

Theorem 8 *If we have a number of spikes p that is constant in respect to n, we can recover the p spikes by an alternating the use of melonic diagram, power iteration and deflation.*

We first use the tetrahedral diagram to obtain a vector correlated with the signal vectors, then we follow by power iteration to obtain a normalized vector $\boldsymbol{v}$ highly correlated to one of the spikes and then deflation which consists in replacing the tensor T by $\mathsf{T} - \alpha \boldsymbol{v}^{\otimes 3}$ where $\alpha = \mathsf{T}.\boldsymbol{v}^{\otimes 3}$. The experimental results suggest that it works not only for small values of p but also for a number of spikes up to n.

Adaptation to Tucker decomposition: We consider the decomposition of a tensor into a set of matrices (loadings) with orthogonal columns and a core tensor. Thus, we adapted in a simple way our framework to this other decomposition scheme. This highlights how generic and important this new framework is. We compared the principal methods of Tucker decomposition with a straightforward adaptation of these methods where we use the matrix associated to the tetrahedral instead of the melonic (tensor unfolding matrices) that are initially used in these methods (as loadings in the initialization as well as in the power iteration).

The case r=1 It is straightforward to see that when $r_1 = r_2 = r_3 = 1$, HOOI is exactly equivalent to Tensor Unfolding with power iteration. From a theoretical point of view, not only our algorithm also achieve the optimal estimation error rate but we also unveiled a new phase transition appearing in the asymmetric case when the tensor dimensions are of the form $n1 > n2 * n3$. This was not studied in (Zhang and Xia 2018) as they considered only the cases where $\exists C_0$ an universal constant such that $n_1, n_2, n_3 \leq C_0 \min(n_1, n_2, n_3)$.

The case r>1 We adapted the framework to the Tucker decomposition case to compare to HOOI. Indeed one straightforward application of our new results is to replace the matricization $\mathcal{M}_k$ which correspond to the melonic graph (since SVD($\mathbf{M}$)=Eigenvalues($\mathbf{MM}^\top$)) with a new matricization corresponding to the tetrahedral graph. $\Rightarrow$ We perform SVD on the tetrahedral matrices.

Numerical experiments

In this section we will investigate the empirical results of the previously mentioned applications in order to see if they match with our theoretical results. We restrict to the dimension $k = 3$ for simplicity. More experiments, a discussion

about the computational complexity of the algorithms, and details about the settings could be found in the Supplementary Material.

Comparison of recovery methods for Tensor PCA

For the recovery algorithm, we focus in the symmetric case and, as in (Richard and Montanari 2014), for every algorithm we use two variants: the simple algorithm outputting $\boldsymbol{v}$ and an algorithm where we apply 100 power iterations on $\boldsymbol{v}$: $v_i \leftarrow T_{ijk} v_j v_k$, distinguishable by a prefix "p-". In Figure 4, we run 200 experiments for each value of β and plot the 95% confidence interval of the correlation of the vector recovered with the signal vector. We will compare our method (tetrahedral) to other algorithmic methods: the melonic (tensor unfolding) (Richard and Montanari 2014) and the homotopy (Anandkumar et al. 2017). To the best of our knowledge, they give the state of art respectively for the symmetric and asymmetric tensor (Biroli, Cammarota, and Ricci-Tersenghi 2020). Other methods exist but are either too computationally expensive (sum of squares) or are variants of these algorithms.
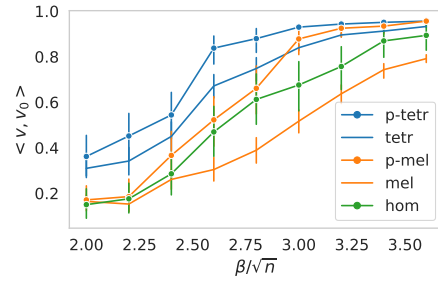

Figure 4: Comparison of different methods for symmetric recovery. n=150.

Spike with different dimensions: We aim to recover the three vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ and $\boldsymbol{v}_3$ from a tensor $\mathsf{Z} + \beta \boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{v}_3$. We repeat 100 times each instance consisting in choosing randomly $\boldsymbol{v}_1, \boldsymbol{v}_2$ and $\boldsymbol{v}_3$ and the Gaussian random tensor Z and we plot the correlation of the signal vectors with the vectors recovered using the tetrahedral. We see in the Figure 5 that the threshold $(n_1 n_2 n_3)^{1/4}$ for the three vectors matches perfectly with the experiences when $n_3 < n_1.n_2$. We also see that when $n_3 > n_1.n_2$ the recovery of v_3 (in green and with the diamond and square markers) have a different asymptotic behavior than v_1 and v_2 (it becomes $n_3^{1/2}$ since $n_3^{1/2} \geq (n_1 n_2 n_3)^{1/4}$), corresponding to what our theoretical study predicted.

Comparison for CP and Tucker decomposition

In Figure 6 and 7, we compare the proposed tensor decomposition algorithms that we derived from our framework with the state of the art for both CP and Tucker decompositions. The comparisons are done for $n = 100$ and for different β over 20 independent runs. The figure 6 concerns the CP decomposition and suggests that the proposed Tetrahedral

(Tetra) method is more robust to noise than the commonly used algorithm of the TensorLy package (based on ALS), as well as the power iteration method. For Tucker decomposition, the results show that the proposed method (Tetra HOOI) provides better results than HOOI and HOSVD in the symmetric case with $r_1 = r_2 = r_3 = 20$ in the large noise regime. Furthermore, we carried out experiments on structured real data, the Yale Face Database B (Lee, Ho, and Kriegman 2005). In more details, we considered a set of stacked face images that form this database as an initial tensor to which we added Gaussian noise. First, we compare HOOI and the proposed Tetra-HOOI algorithms for a fixed value of the rank of the matrices involved in this type of methods $(r_1, r_2, r_3) = (10, 10, 10)$ and for different values of the noise intensity (λ). To evaluate the denoising performance of those methods, we compute the average and standard deviation (over 5 runs) of the Frobenius norm $\frac{\|\mathbf{X}-\mathbf{V}\|}{\|\mathbf{V}\|}$ where $\mathbf{V}$ is the input tensor and $\mathbf{X}$ is the output of the algorithm. The results which are reported in the table 1 show that Tetra HOOI again outperforms HOOI even on real data. Note that, we obtained similar results (reported in supplementary materials) for a fixed ($\lambda = 1000$) and for different values of (r_1, r_2, r_3). Note that the Frobenius norm of the difference between two completely uncorrelated normalized vectors is equal to 2.
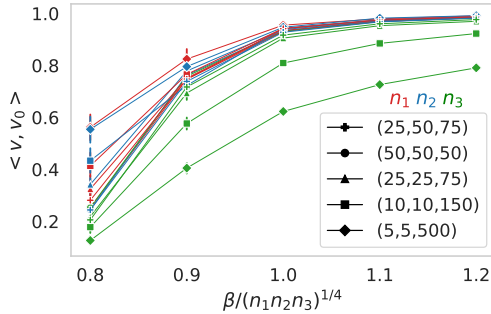

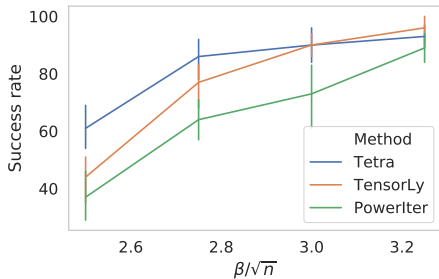Figure 5: Recovery of a spike with different dimensions.

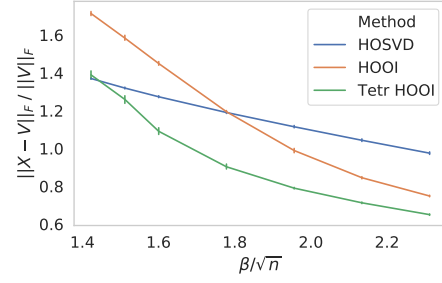Figure 6: Comparison CP decomposition methods for $n = 100$ and $n_{\text{spikes}} = 20$.

Figure 7: Comparison Tucker decomposition methods for $n = 100$ and $r_1 = r_2 = r_3 = 20$

λ	900	1100	1300	1500
HOOI	0.696 ± 0.008	0.845 ± 0.006	0.993 ± 0.008	1.125 ± 0.007
Tetra-HOOI	0.610 ± 0.017	0.648 ± 0.010	0.707 ± 0.019	0.754 ± 0.013

Table 1: We compare HOOI and the proposed Tetra-HOOI algorithms for a fixed value of the rank of the matrices involved in this type of methods $(p_1, p_2, p_3) = (10, 10, 10)$. We compute the average and standard deviation (over 5 runs) of the Frobenius norm for different values of the noise intensity (λ).

Conclusion

In this paper we introduced a novel framework for the tensor decomposition based on trace invariants. Within this framework, we provide different algorithms to recover a signal vector it with theoretical guarantees. These algorithms use tensor contractions that has a high potential of parallelization and computing optimization. We illustrate the practical pertinence of our framework by presenting some examples of algorithms as well as generalizations of these algorithms for CP decomposition and Tucker decomposition methods. Our experimental results show that the tetrahedral graph performs better than the the state of the art for Tensor PCA, and that its tensor decomposition generalizations shows a better robustness to noise comparing to existent algorithms. Interestingly, our framework is also able to extend the theoretical and practical study of tensor PCA to new and less restrictive situations like data where the dimensions of the axes are different. Important directions of future research is to explore the potential of more general graphs, as well as investigate the new proposed theoretical threshold for different dimensions in the context of tensor decomposition.

Acknowledgments

References

Anandkumar, A.; Deng, Y.; Ge, R.; and Mobahi, H. 2017. Homotopy analysis for tensor PCA. In *Conference on*

Learning Theory, 79–104. PMLR.

Anandkumar, A.; Foster, D. P.; Hsu, D.; Kakade, S. M.; and Liu, Y.-K. 2015. A spectral algorithm for latent dirichlet allocation. *Algorithmica* 72(1): 193–214.

Anandkumar, A.; Ge, R.; Hsu, D.; Kakade, S. M.; and Telgarsky, M. 2014. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research* 15: 2773–2832.

Anandkumar, A.; Ge, R.; and Janzamin, M. 2015. Learning overcomplete latent variable models through tensor methods. In *Conference on Learning Theory*, 36–112. PMLR.

Arous, G. B.; Gheissari, R.; Jagannath, A.; et al. 2020. Algorithmic thresholds for tensor PCA. *Annals of Probability* 48(4): 2052–2087.

Arous, G. B.; Mei, S.; Montanari, A.; and Nica, M. 2019. The landscape of the spiked tensor model. *Communications on Pure and Applied Mathematics* 72(11): 2282–2330.

Astrid, M.; and Lee, S.-I. 2017. Cp-decomposition with tensor power method for convolutional neural networks compression. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 115–118. IEEE.

Avohou, R. C.; Ben Geloun, J.; and Dub, N. 2020. On the counting of $O(N)$ tensor invariants. *Adv. Theor. Math. Phys.* 24(4): 821–878. doi:10.4310/ATMP.2020.v24.n4.a1.

Azizzadenesheli, K.; Lazaric, A.; and Anandkumar, A. 2016. Reinforcement Learning of POMDPs using Spectral Methods. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research, publisher = PMLR*, 193–256.

Baik, J.; Arous, G. B.; Péché, S.; et al. 2005. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability* 33(5): 1643–1697.

Biroli, G.; Cammarota, C.; and Ricci-Tersenghi, F. 2020. How to iron out rough landscapes and get optimal performances: averaged gradient descent and its application to tensor PCA. *Journal of Physics A Mathematical General* 53(17): 174003.

Biroli, G.; Cammarota, C.; and Ricci-Tersenghi, F. 2020. How to iron out rough landscapes and get optimal performances: averaged gradient descent and its application to tensor PCA. *Journal of Physics A: Mathematical and Theoretical* 53(17): 174003.

Cartwright, D.; and Sturmfels, B. 2013. The number of eigenvalues of a tensor. *Linear algebra and its applications* 438(2): 942–952.

Comon, P.; Luciani, X.; and De Almeida, A. L. 2009. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics: A Journal of the Chemometrics Society* 23(7-8): 393–405.

De Lathauwer, L.; De Moor, B.; and Vandewalle, J. 2000a. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications* 21(4): 1253–1278.

De Lathauwer, L.; De Moor, B.; and Vandewalle, J. 2000b. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications* 21(4): 1324–1342.

Decurninge, A.; Land, I.; and Guillaud, M. 2020. Tensor-based modulation for unsourced massive random access. *IEEE Wireless Communications Letters* 10(3): 552–556.

Dudeja, R.; and Hsu, D. 2020. Statistical Query Lower Bounds for Tensor PCA. *arXiv e-prints* arXiv:2008.04101.

Evnin, O. 2020. Melonic dominance and the largest eigenvalue of a large random tensor. *arXiv preprint arXiv:2003.11220* .

Ferrari, F.; Rivasseau, V.; and Valette, G. 2019. A new large N expansion for general matrix–tensor models. *Communications in Mathematical Physics* 370(2): 403–448.

Gurau, R. 2014. Universality for Random Tensors. *Ann. Inst. H. Poincare Probab. Statist.* 50(4): 1474–1525.

Gurau, R. 2017. *Random Tensors*. Oxford University Press.

Gurau, R. 2020. On the generalization of the Wigner semicircle law to real symmetric tensors. *arXiv preprint arXiv:2004.02660* .

Hastings, M. B. 2020. Classical and Quantum Algorithms for Tensor Principal Component Analysis. *Quantum* 4: 237.

Hopkins, S. B.; Shi, J.; and Steurer, D. 2015. Tensor principal component analysis via sum-of-squares proofs. *CoRR* abs/1507.03269.

Jagannath, A.; Lopatto, P.; Miolane, L.; et al. 2020. Statistical thresholds for tensor PCA. *Annals of Applied Probability* 30(4): 1910–1933.

Janzamin, M.; Sedghi, H.; and Anandkumar, A. 2015. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473* .

Lahat, D.; Adali, T.; and Jutten, C. 2015. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE* 103(9): 1449–1477.

Lee, K.-C.; Ho, J.; and Kriegman, D. J. 2005. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on pattern analysis and machine intelligence* 27(5): 684–698.

Lesieur, T.; Miolane, L.; Lelarge, M.; Krzakala, F.; and Zdeborová, L. 2017. Statistical and computational phase transitions in spiked tensor estimation. In *2017 IEEE International Symposium on Information Theory (ISIT)*, 511–515. IEEE.

Lin, T.; and Bourennane, S. 2013. Survey of hyperspectral image denoising methods based on tensor decompositions. *EURASIP journal on Advances in Signal Processing* 2013(1): 1–11.

Mannelli, S. S.; Biroli, G.; Cammarota, C.; Krzakala, F.; Urbani, P.; and Zdeborová, L. 2020. Marvels and pitfalls of the langevin algorithm in noisy high-dimensional inference. *Physical Review X* 10(1): 011057.

Mannelli, S. S.; Krzakala, F.; Urbani, P.; and Zdeborova, L. 2019. Passed & spurious: Descent algorithms and local minima in spiked matrix-tensor models. In *international conference on machine learning*, 4333–4342. PMLR.

Moitra, A.; and Wein, A. S. 2019. Spectral methods from tensor networks. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 926–937.

Papalexakis, E. E.; Faloutsos, C.; and Sidiropoulos, N. D. 2016. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8(2): 1–44.

Perry, A.; Wein, A. S.; Bandeira, A. S.; et al. 2020. Statistical limits of spiked tensor models. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 56, 230–264. Institut Henri Poincaré.

Qi, L. 2005. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation* 40(6): 1302 – 1324.

Richard, E.; and Montanari, A. 2014. A statistical model for tensor PCA. In *Advances in Neural Information Processing Systems*, 2897–2905.

Ros, V.; Arous, G. B.; Biroli, G.; and Cammarota, C. 2019a. Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions. *Physical Review X* 9(1): 011003.

Ros, V.; Arous, G. B.; Biroli, G.; and Cammarota, C. 2019b. Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions. *Physical Review X* 9(1): 011003.

Sarao Mannelli, S.; Biroli, G.; Cammarota, C.; Krzakala, F.; and Zdeborová, L. 2019. Who is afraid of big bad minima? analysis of gradient-flow in spiked matrix-tensor models. *Advances in Neural Information Processing Systems* 32: 8679–8689.

Sun, W.; and Braatz, R. D. 2020. Opportunities in tensorial data analytics for chemical and biological manufacturing processes. *Computers & Chemical Engineering* 107099.

Vasilescu, M. A. O.; and Terzopoulos, D. 2002. Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision*, 447–460. Springer.

Wang, Y.; and Anandkumar, A. 2016. Online and differentially-private tensor decomposition. *arXiv preprint arXiv:1606.06237* .

Wein, A. S.; El Alaoui, A.; and Moore, C. 2019. The kikuchi hierarchy and tensor pca. In *Annual Symposium on Foundations of Computer Science (FOCS)*, 1446–1468. IEEE.

Zhang, A.; and Xia, D. 2018. Tensor SVD: Statistical and computational limits. *IEEE Transactions on Information Theory* 64(11): 7311–7338.