Polynomial-Time Tensor Decompositions with Sum-of-Squares

Tengyu Ma
Dept. of Computer Science
Princeton University
Princeton, NJ, USA
Email: tengyu@cs.princeton.edu

Jonathan Shi
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
Email: jshi@cs.cornell.edu

David Steurer
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
Email: dsteurer@cs.cornell.edu

***Abstract*—We give new algorithms based on the sum-of-squares method for tensor decomposition. Our results improve the best known running times from quasi-polynomial to polynomial for several problems, including decomposing random overcomplete 3-tensors and learning overcomplete dictionaries with constant relative sparsity. We also give the first robust analysis for decomposing overcomplete 4-tensors in the smoothed analysis model.**

A key ingredient of our analysis is to establish small spectral gaps in moment matrices derived from solutions to sum-of-squares relaxations. To enable this analysis we augment sum-of-squares relaxations with spectral analogs of maximum entropy constraints.

***Keywords*-tensor decomposition; smoothed analysis; FOOBI algorithm; sum-of-squares method; Lasserre hierarchy; simultaneous diagonalization**

I. Introduction

Tensors are arrays of (real) numbers with multiple indices—generalizing matrices (two indices) and vectors (one index) in a natural way. They arise in many different contexts, e.g., moments of multivariate distributions, higher-order derivatives of multivariable functions, and coefficients of multivariate polynomials. An important ongoing research effort aims to extend algorithmic techniques for vectors and matrices to more general tensors. A key challenge is that many tractable matrix computations (like rank and spectral norm) become NP-hard in the tensor setting (even for just three indices) [1], [2]. However, recent work gives evidence that it is possible to avoid this computational intractability and develop provably efficient algorithms, especially for low-rank tensor decompositions, by making suitable assumptions about the input and allowing for approximations [3]–[7]. These algorithms lead to the best known provable guarantees for a wide range of unsupervised learning problems [8]–[11], including learning mixtures of Gaussians [12], Latent Dirichlet topic modeling [13], and dictionary learning [14]. Low-rank tensor decompositions are useful for these learning problems because they are often unique up to permuting the factors—in contrast, low-rank matrix factorizations are unique only up to unitary transformation. In fact, as far as we are aware, in all natural situations where finding low-rank tensor decompositions is tractable, the decompositions are also unique.

We consider the following (symmetric) version of the tensor decomposition problem: Let $a_1, \ldots, a_n \in \mathbb{R}^d$ be d-dimensional unit vectors. We are given (approximate) access to the first k moments $\mathcal{M}_1, \ldots, \mathcal{M}_k$ of the uniform distribution over $a_1, \ldots, a_n$, that is,

$$\mathcal{M}_t = \tfrac{1}{n} \sum_{i=1}^{n} a_i^{\otimes t} \quad \text{for } t \in \{1, \ldots, k\}\,. \tag{1}$$

The goal is to approximately recover the vectors $a_1, \ldots, a_n$. What conditions on the vectors $a_1, \ldots, a_n$ and the number of moments k allow us to efficiently and robustly solve this problem?

A classical algorithm based on (simultaneous) matrix diagonalization [15], [16, attributed to Jennrich] shows that whenever the vectors $a_1, \ldots, a_n$ are linearly independent, $k = 3$ moments suffice to recover the vectors in polynomial time. (This algorithm is also robust against polynomially small errors in the input moment tensors [9], [10], [17].) Therefore an important remaining algorithmic challenge for tensor decomposition is the *overcomplete* case, when the number of vectors (significantly) exceeds their dimension. Several recent works studied this case with different assumptions on the vectors and the number of moments. In this work, we give a unified algorithmic framework for overcomplete tensor decomposition that achieves—and in many cases surpasses—the previous best guarantees for polynomial-time algorithms.

In particular, some decompositions that previously required quasi-polynomial time to find are reduced to polynomial time in our framework, including the case of general tensors with order logarithmically large in its overcompleteness n/d [14] and random order-3 tensors with rank $n \leq d^{3/2}/\log^{O(1)}(d)$ [5]. Iterative methods may also achieve fast local convergence guarantees for incoherent order-3 tensors with rank $o(d^{3/2})$, which become global convergence guarantees under no more than constant overcompleteness [8]. In the smoothed analysis model, where each vector of the desired decomposition is assumed to have been randomly perturbed by an inverse polynomial amount, polynomial-time decomposition was achieved for order-5 tensors of rank up to $d^2/2$ [9]. Our framework extends this result to order-4 tensors, for which the corresponding analysis was previously unknown for any superconstant overcompleteness.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.54

The starting point of our work is a new analysis of the aforementioned matrix diagonalization algorithm that works for the case when $a_1, \ldots, a_n$ are linearly independent. A key ingredient of our analysis is a powerful and by now standard concentration bound for Gaussian matrix series [18], [19]. An important feature of our analysis is that it is captured by the sum-of-squares (SoS) proof system in a robust way. This fact allows us to use Jennrich's algorithm as a rounding procedure for sum-of-squares relaxations of tensor decomposition, which is the key idea behind improving previous quasi-polynomial time algorithms based on these relaxations [5], [14].

The main advantage that sum-of-squares relaxations afford for tensor decomposition is that they allow us to efficiently hallucinate faithful *higher-degree moments* for a distribution given only its lower-degree moments. We can now run classical tensor decomposition algorithms like Jennrich's on these hallucinated higher-degree moments (akin to *rounding*). The goal is to show that those algorithms work as well as they would on the true higher moments. What is challenging about it is that the analysis of Jennrich's algorithm relies on small spectral gaps that are difficult to reason about in the sum-of-squares setting. (Previous sum-of-squares based methods for tensor decomposition also followed this outline but used simpler, more robust rounding algorithms which required quasi-polynomial time.)

To this end, we view solutions to sum-of-squares relaxations as *pseudo-distributions*, which generalize classical probability distributions in a way that takes computational efficiency into account.[1] More concretely, pseudo-distributions are indistinguishable from actual distributions with respect to tests captured by a restricted system of proofs, called *sum-of-squares proofs*.

An interesting feature of how we use pseudo-distributions is that our relaxations search for pseudo-distributions of large *entropy* (via an appropriate surrogate). This objective is surprising, because when we consider convex relaxations of NP-hard search problems, the intended solutions typically correspond to atomic distributions which have entropy 0. Here, high entropy in the pseudo-distribution allows us to ensure that rounding results in a useful solution. This appears to be related to the way in which many randomized rounding procedures use maximum-entropy distributions [20], but differs in that the aforementioned rounding procedures focus on the entropy of the rounding process rather than the entropy (surrogate) of the solution to the convex relaxation. A measure of "entropy" has also been directly ascribed to pseudo-distributions previously [21], and the principle of maximum entropy has been applied to pseudo-distributions as well [22], but these have previously occurred separately, and our application is the first to encode a surrogate notion of entropy directly into the sum-of-squares proof system.

Our work also takes inspiration from a recent work that uses sum-of-squares techniques to design fast spectral algorithms for a range of problems including tensor decomposition [7]. Their algorithm also proceeds by constructing surrogates for higher moments and applying a classical tensor decomposition algorithm on these surrogates. The difference is that the surrogates in [7] are explicitly constructed as low-degree polynomial of the input tensor, whereas our surrogates are computed by sum-of-squares relaxations. The explicit surrogates of [7] allow for a direct (but involved) analysis through concentration bounds for matrix polynomials. In our case, a direct analysis is not possible because we have very little control over the surrogates computed by sum-of-squares relaxations. Therefore, the challenge for us is to understand to what extent classical tensor decomposition algorithms are compatible with the sum-of-squares proof system. Our analysis ends up being less technically involved compared to [7] (using the language of pseudo-distributions and sum-of-squares proofs).

This version of the paper is an abbreviated version containing only rough overviews and proof sketches. See the full version for the complete arguments.

A. Results for tensor decomposition

Let $\{a_1, \ldots, a_n\} \subseteq \mathbb{R}^d$ be a set of unit vectors. We study the task of approximately recovering this set of vectors given (noisy) access to its first k moments (1). We organize this overview of our results based on different kinds of assumptions imposed on the set $\{a_1, \ldots, a_n\}$ and the order of tensor/moments that we have access to. All of our algorithms are randomized and may fail with some small probability over their internal randomness, say probability at most 0.01. (Standard arguments allow us to amplify this probability at the cost of a small increase in running time.)

Orthogonal vectors.: This scenario often captures the case of general linearly independent vectors because knowledge of the second moments of $a_1, \ldots, a_n$ allows us to orthonormalize the vectors (this process is sometimes called "whitening"). Many efficient algorithms are known in this case. Our contribution here is in improving the error tolerance. For a symmetric 3-tensor $E \in (\mathbb{R}^d)^{\otimes 3}$, we use $\|E\|_{\{1\},\{2,3\}}$ to denote the spectral norm of E as a d-by-d^2 matrix (using the first mode of E to index rows and the last two modes of E to index the columns). This norm is at most $\sqrt{d}$ times the injective norm $\|E\|_{\{1\},\{2\},\{3\}}$ (the maximum of $\langle E, x \otimes y \otimes z \rangle$ over all unit vectors $x, y, z \in \mathbb{R}^d$). The previous best error tolerance for this problem required the error tensor $E = T - \sum_{i=1}^n a_i^{\otimes 3}$ to have injective norm $\|E\|_{\{1\},\{2\},\{3\}} \ll 1/d$. Our algorithm requires only $\|E\|_{\{1\},\{2,3\}} \ll 1$, which is satisfied in particular when $\|E\|_{\{1\},\{2\},\{3\}} \ll 1/\sqrt{d}$.

[1] In particular, the set of constant-degree moments of n-variate pseudo-distributions admits an $n^{O(1)}$-time separation oracle based on computing eigenvectors.

Theorem 1. *There exists a polynomial-time algorithm that given a symmetric* 3*-tensor* $T \in (\mathbb{R}^d)^{\otimes 3}$ *outputs a set of vectors* $\{a'_1, \dots, a'_{n'}\} \subseteq \mathbb{R}^d$ *such that for every orthonormal set* $\{a_1, \dots, a_n\} \subseteq \mathbb{R}^d$*, the Hausdorff distance*[2] *between the two sets is at most*

$$\begin{aligned}\operatorname{dist}_H\left(\{a_1, \dots, a_n\}, \{a'_1, \dots, a'_{n'}\}\right)^2 \\ \le O(1) \cdot \left\| T - \sum\nolimits_{i=1}^n a_i^{\otimes 3} \right\|_{\{1\},\{2,3\}} .\end{aligned} \tag{2}$$

With an additional assumption $\|T - \sum_{i=1}^n a_i^{\otimes 3}\|_{\{1\},\{2,3\}} \le 1/\log d$, the running time of the algorithm can be improved to $O(d^{1+\omega}) \le d^{3.33}$ using fast matrix multiplication, where ω is the number such that two $n \times n$ matrices can be multiplied together in time n^ω (See full version of this paper for precise statement and proof).

It is also possible to replace the spectral norm $\|\cdot\|_{\{1\},\{2,3\}}$ in the above theorem statement by constant-degree sum-of-squares relaxations of the injective norm of 3-tensors. If the error E has Gaussian distribution $\mathcal{N}(0, \sigma^2 \cdot \operatorname{Id}_d^{\otimes 3})$, then this norm is w.h.p. bounded by $\sigma \cdot d^{3/4} (\log d)^{O(1)}$ [6], whereas the norm $\|\cdot\|_{\{1\},\{2,3\}}$ has magnitude $\Omega(\sigma \cdot d)$.

Random vectors.: We consider the case that $a_1, \dots, a_n$ are chosen independently at random from the unit sphere of $\mathbb{R}^d$. For $n \le d$, this case is roughly equivalent to the case of orthonormal vectors. Thus, we are interested in the "overcomplete" case $n \gg d$, when the rank is larger than the dimension. Previous work found the decomposition in quasi-polynomial time when $n \le d^{3/2}/\log^{O(1)} d$ [5], or in time subquadratic in the input size when $n \le d^{4/3}/\log^{O(1)} d$ [7]. Our polynomial-time algorithm therefore is an improvement when n is between $d^{4/3}$ and $d^{3/2}$ (up to logarithmic factors).

Theorem 2. *There exists a polynomial-time algorithm* A *such that with probability* $1 - d^{-\omega(1)}$ *over the choice of random unit vectors* $a_1, \dots, a_n \in \mathbb{R}^d$*, every symmetric* 3*-tensor* $T \in (\mathbb{R}^d)^{\otimes 3}$ *satisfies*

$$\begin{aligned}\operatorname{dist}_H\Big(A(T), \{a_1, \dots, a_n\}\Big)^2 \\ \le O\left(\left(\frac{n}{d^{1.5}}\right)^{\Omega(1)} + \left\| T - \sum\nolimits_{i=1}^n a_i^{\otimes 3} \right\|_{\{1\},\{2,3\}} \right) .\end{aligned} \tag{3}$$

Again we may replace the spectral norm $\|\cdot\|_{\{1\},\{2,3\}}$ in the above theorem statement by constant-degree sum-of-squares relaxations of the injective norm of 3-tensors, which as mentioned before give better bounds for Gaussian error tensors.

Smoothed vectors.: Next, we consider a more general setup where the vectors $a_1, \dots, a_n \in \mathbb{R}^d$ are smoothed, i.e., randomly perturbed. This scenario is significantly more general than random vectors. Again we are interested in the overcomplete case $n \gg d$. The previous best work [9] showed that the fifth moment of smoothed vectors $a_1, \dots, a_n$ with $n \le d^2/2$ is enough to approximately recover the vectors even in the presence of a polynomial amount of error. For fourth moments of smoothed vectors, no such result was known even for lower overcompleteness, say $n = d^{1.01}$.

We give an interpretation of the 4-tensor decomposition algorithm FOOBI[3] [23] as a special case of a sum-of-squares based decomposition algorithm. We show that the sum-of-squares based algorithm works in the smoothed setting even in the presence of a polynomial amount of error. We define a condition number $\kappa(\cdot)$ for sets of vectors $a_1, \dots, a_n \in \mathbb{R}^d$ (a polynomial in the condition number of two matrices, one with columns $\{a_i^{\otimes 2} \mid i \in [n]\}$ and one with columns $\{a_i \otimes (a_i \otimes a_j - a_j \otimes a_i) \otimes a_j \mid i \neq j \in [n]\}$). First, we show that the algorithm can tolerate error $\ll 1/\kappa$ which could be independent of the dimension. Concretely, our algorithm will output a set of vectors $\hat{a}_1, \dots, \hat{a}_n$ which will be close to $\{a_1, \dots, a_n\}$ up to permutations and sign flip with a relative error that scales linearly in the error of the input and the condition number κ. Second, we show that for smoothed vectors this condition number is at least inverse polynomial with probability exponentially close to 1.

Theorem 3. *There exists a polynomial-time algorithm such that for every symmetric* 4*-tensor* $T \in (\mathbb{R}^d)^{\otimes 4}$ *and every set* $\{a_1, \dots, a_n\} \subseteq \mathbb{R}^d$ *(not necessarily unit length), the output* $\{a'_1, \dots, a'_n\}$ *of the algorithm on input* T *satisfies*

$$\begin{aligned}\min_{\pi\colon [n] \xrightarrow{bij.} [n]} \max_{i \in [n]} \frac{\left\| a_i^{\otimes 2} - a_i'^{\otimes 2} \right\|^2}{\left\| a_i^{\otimes 2} \right\|^2} \\ \le O(1) \cdot \left\| T - \sum\nolimits_{i=1}^n a_i^{\otimes 4} \right\|_{\{1,2\},\{3,4\}} \cdot \kappa(a_1, \dots, a_n) .\end{aligned} \tag{4}$$

We say that a distribution over vectors $a_1, \dots, a_n \in \mathbb{R}^d$ is γ-smoothed if $a_i = a_i^0 + \gamma \cdot g_i$, where $a_1^0, \dots, a_n^0$ are fixed vectors and $g_1, \dots, g_n$ are independent Gaussian vectors from $\mathcal{N}(0, \frac{1}{d} \operatorname{Id}_d)$.

Theorem 4. *Let* $\varepsilon > 0$ *and* $n, d \in \mathbb{N}$ *with* $n \le d^2/10$*. Then, for any* γ*-smoothed distribution over vectors* $a_1, \dots, a_n$ *in* $\mathbb{R}^d$*,*

$$\mathbb{P}\Big\{ \kappa(a_1, \dots, a_n) \le \operatorname{poly}(d, \gamma) \Big\} \ge 1 - \exp(-d^{\Omega(1)}) .$$

The above theorems together imply a polynomial-time algorithm for approximately decomposing overcomplete smoothed 4-tensors even if the input error is polynomially large. The error probability of the algorithm is exponentially small over the choice of the smoothing. It is an interesting

[2] The Hausdorff distance $\operatorname{dist}_H(X, Y)$ between two finite sets X and Y measures the length of the largest gap between the two sets. Formally, $\operatorname{dist}_H(X, Y)$ is the maximum of $\max_{x \in X} \min_{y \in Y} \|x - y\|$ and $\max_{y \in Y} \min_{x \in X} \|x - y\|$.

[3] The FOOBI algorithm is known to work for overcomplete 4-tensors when there is no error in the input. Researchers [9] asked if this algorithm tolerates a polynomial amount of error. Our work answers this question affirmatively for a variant of FOOBI (based on sum-of-squares).

open problem to extend this result to overcomplete smoothed 3-tensors, even for lower overcompleteness $n = d^{1.01}$.

Separated unit vectors.: In the scenario, when inner products among the vectors $a_1, \ldots, a_n \in \mathbb{R}^d$ are bounded by $\rho < 1$ in absolute value, the previous best decomposition algorithm shows that moments of order $(\log n)/\log \rho$ suffice [24]. Our algorithm requires moments of higher order (by a factor logarithmic in the desired accuracy) but in return tolerates up to constant spectral error. This increased error tolerance also allows us to apply this result for dictionary learning with up to constant sparsity (see Section I-B).

Theorem 5. *There exists an algorithm A with polynomial running time (in the size of its input) such that for all $\eta, \rho \in (0,1), \sigma \geq 1$, for every symmetric $2k$-tensor $T \in (\mathbb{R}^d)^{\otimes 2k}$ with $k \geq O\left(\frac{1+\log\sigma}{\log\rho}\right) \cdot \log(1/\eta)$ and every set of unit vectors $\{a_1, \ldots, a_n\} \subseteq \mathbb{R}^d$ with $\|\sum_{i=1}^n a_i {a_i}^{\mathsf{T}}\| \leq \sigma$ and $\max_{i \neq j} \langle a_i, a_j \rangle^2 \leq \rho$,*

$$\begin{aligned} &\operatorname{dist}_H \left(A(T), \{a_1^{\otimes 2}, \ldots, a_n^{\otimes 2}\}\right)^2 \\ &\leq O\left(\eta + \left\|T - \sum\nolimits_{i=1}^n a_i^{\otimes 2k}\right\|_{\{1,\ldots,k\},\{k+1,\ldots,2k\}}\right) \end{aligned} \quad (5)$$

We also show (in the full version of this paper) that a simple spectral algorithm with running time close to d^{2k} (the size of the input) achieves similar guarantees. However, the error tolerance of this algorithm is in terms of an *unbalanced* spectral norm: $\|T - \sum_{i=1}^n a_i^{\otimes 2k}\|_{\{1,\ldots,2k/3\},\{2k/3+1,\ldots,2k\}}$ (the spectral norm of the tensor viewed as a $d^{2k/3}$-by-$d^{4k/3}$ matrix). This norm is always larger than the balanced spectral norm in the theorem statement. In particular, for dictionary learning applications, this norm is larger than 1, which renders the guarantee of the simpler spectral algorithm vacuous in this case.

General unit vectors.: In this scenario, the number of moments that our algorithm requires is constant as long as $\sum_i a_i {a_i}^{\mathsf{T}}$ has constant spectral norm and the desired accuracy is constant.

Theorem 6. *There exists an algorithm A (see full version of this paper) with polynomial running time (in the size of its input) such that for all $\varepsilon \in (0,1), \sigma \geq 1$, for every set of unit vectors $\{a_1, \ldots, a_n\} \subseteq \mathbb{R}^d$ with $\|\sum_{i=1}^n a_i {a_i}^{\mathsf{T}}\| \leq \sigma$ and every symmetric $2k$-tensor $T \in (\mathbb{R}^d)^{\otimes 2k}$ with $k \geq (1/\varepsilon)^{O(1)} \cdot \log(\sigma)$ and $\|T - \sum_i a_i^{\otimes 2k}\|_{\{1,\ldots,k\},\{k+1,\ldots,2k\}} \leq 1/3$, we have*

$$\operatorname{dist}_H \left(A(T), \{a_1^{\otimes 2}, \ldots, a_n^{\otimes 2}\}\right)^2 \leq O(\varepsilon) \,.$$

The previous best algorithm for this problem needed tensors of order $(\log\sigma)/\varepsilon$ and required running time $d^{O((\log\sigma)/\varepsilon^{O(1)} + \log n)}$ [14, Theorem 4.3]. We require the same order of the tensor and the runtime is improved to be polynomial in the size of the inputs (that is, $n^{\operatorname{poly}((\log\sigma)/\varepsilon)}$).

B. Applications of tensor decomposition

Tensor decomposition has a wide range of applications. We focus here on learning sparse dictionaries, which is an example of the more general phenomenon of using tensor decomposition to learn latent variable models. Here, we obtain the first polynomial-time algorithms that work in the overcomplete regime up to constant sparsity.

Dictionary learning is an important problem in multiple areas, ranging from computational neuroscience [25]–[27], machine learning [28], [29], to computer vision and image processing [30]–[32]. The general goal is to find a good basis for given data. More formally, in the dictionary learning problem, also known as sparse coding, we are given samples of a random vector $y \in \mathbb{R}^n$, of the form $y = Ax$ where A is some unknown matrix in $\mathbb{R}^{n \times m}$, called *dictionary*, and x is sampled from an unknown distribution over sparse vectors. The goal is to approximately recover the dictionary A.

We consider the same class of distributions over sparse vectors $\{x\}$ as [14], which as discussed in [14] admits a wide-range of non-product distributions over sparse vectors. (The case of product distributions reduces to the significantly easier problem of independent component analysis.) We say that $\{x\}$ is (k,τ)-*nice* if $\mathbb{E}\, x_i^k = 1$ for every $i \in [m]$, $\mathbb{E}\, x_i^{k/2} x_j^{k/2} \leq \tau$ for all $i \neq j \in [m]$, and $\mathbb{E}\, x^\alpha = 0$ for every non-square degree-k monomial x^α. Here, τ is a measure of the relative sparsity of the vectors $\{x\}$.

We give an algorithm that for nice distributions solves the dictionary learning problem in polynomial time when the desired accuracy is constant, the overcompleteness of the dictionary is constant (measured by the spectral norm $\|A\|$), and the sparsity parameter τ is a sufficiently small constant (depending only on the desired accuracy and $\|A\|$). The previous best algorithm [14] requires quasi-polynomial time in this setup (but works in polynomial-time for polynomial sparsity $\tau \leq n^{-\Omega(1)}$).

Theorem 7. *There exists an algorithm $\mathcal{R}$ parameterized by $\sigma \geq 1, \eta \in (0,1)$, such that for every dictionary $A \in \mathbb{R}^{n \times m}$ with $\|A\| \leq \sigma$ and every (k,τ)-nice distribution $\{x\}$ over $\mathbb{R}^m$ with $k \geq k(\eta,\sigma) = O((\log\sigma)/\eta)$ and $\tau \leq \tau(k) = k^{-O(k)}$, the algorithm given $n^{O(k)}$ samples from $\{y = Ax\}$ outputs in time $n^{O(k)}$ vectors $a'_1, \ldots, a'_m$ that are $O(\eta)^{1/2}$-close to the columns of A.*

Since previous work [14] provides a black box reduction from dictionary learning to tensor decomposition, the theorem above follows from Theorem 6. Our Theorem 5 implies a dictionary learning algorithm with better parameters for the case that the columns of A are separated.

C. Polynomial optimization with few global optima

Underlying our algorithms for the tensor decomposition is an algorithm for solving general systems of polynomial constraints with the property that the total number of different solutions is small and that there exists a short certificate for that fact in form of a sum-of-squares proof.

Let $\mathcal{A}$ be a system of polynomial constraints over real variables $x = (x_1, \ldots, x_d)$ and let $P \colon \mathbb{R}^d \to \mathbb{R}^{d^\ell}$ be a polynomial map of degree at most ℓ—for example, $P(x) = x^{\otimes \ell}$. We say that solutions $a_1, \ldots, a_n \in \mathbb{R}^d$ to $\mathcal{A}$ are *unique* under the map P if the vectors $P(a_1), \ldots, P(a_n)$ are orthonormal up to error 0.01 (in spectral norm) and every solution a to $\mathcal{A}$ satisfies $P(a) \approx P(a_i)$ for some $i \in [n]$. We encode this property algebraically by requiring that the constraints in $\mathcal{A}$ imply the constraint $\sum_{i=1}^n \langle P(a_i), P(x)\rangle^4 \geq 0.99 \cdot \|P(x)\|^4$. We say that the solutions $a_1, \ldots, a_n$ are *ℓ-certifiably unique* if in addition this implication has a degree-ℓ sum-of-squares proof.

The following theorem shows that if polynomial constraints have certifiably unique solutions (under a given map P), then we can find them efficiently (under the map P).

Theorem 8 (Informal statement: see full version of this paper for precise formulation and proof)**.** *Given a system of polynomial constraints $\mathcal{A}$ and a polynomial map P such that there exists ℓ-certifiably unique solutions $a_1, \ldots, a_n$ for $\mathcal{A}$, we can find in time $d^{O(\ell)}$ vectors 0.1-close to $P(a_1), \ldots, P(a_n)$ in Hausdorff distance.*

II. Techniques

Here is the basic idea behind using sum-of-squares for tensor decomposition: Let $a_1, \ldots, a_n \in \mathbb{R}^d$ be unit vectors and suppose we have access to their first three moments $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ as in (1). Since the task of recovering $a_1, \ldots, a_n$ is easier the more moments we know, we would make a lot of progress if we could compute higher moments of $a_1, \ldots, a_n$, say the fourth moment $\mathcal{M}_4$. A natural approach toward that goal is to compute a probability distribution D over the sphere of $\mathbb{R}^d$ such that D matches the moments of $a_1, \ldots, a_k$ that we know, i.e., $\mathbb{E}_{D(u)} u = \mathcal{M}_1$, $\mathbb{E}_{D(u)} u^{\otimes 2} = \mathcal{M}_2$, $\mathbb{E}_{D(u)} u^{\otimes 3} = \mathcal{M}_3$, and then use the fourth moment $\mathbb{E}_D u^{\otimes 4}$ as an estimate for $\mathcal{M}_4$.

There are two issues with this approach: (1) computing such a distribution D is intractable and (2) even if we could compute such a distribution it is not clear if its fourth moment will be close to the fourth moments $\mathcal{M}_4$ we are interested in.

We address issue (1) by relaxing D to be a pseudo-distribution (solution to sum-of-squares relaxations). Then, we can match the given moments efficiently.

Issue (2) is related to the uniqueness of the tensor decomposition, which relies on properties of the vectors $a_1, \ldots, a_n$. Here, the general strategy is to first prove that this uniqueness holds for actual distributions and then transfer the uniqueness proof to the sum-of-squares proof system, which would imply that uniqueness also holds for pseudo-distributions.

In subsection II-A below, we demonstrate our key rounding idea on the (nearly) orthogonal tensor decomposition problem. Then in subsection II-B we discuss the high level insight for the robust 4th-order tensor decomposition algorithm and in subsection II-C the techniques for random 3rd-order tensor decomposition.

A. Rounding pseudo-distributions by small spectral gaps

Our main departure from previous tensor decomposition algorithms based on sum-of-squares [5], [14] lies in *rounding*: the procedure to extract an actual solution from a pseudo-distribution over solutions. The previous algorithms rounded a pseudo-distribution D by directly using the first moments (or the mean) $\mathbb{E}_{D(u)} u$, which requires D to concentrate strongly around the desired solution. Our approach here instead uses Jennrich's (simultaneous) matrix diagonalization [15], [16], to extract the desired solution as a *singular vector* of a matrix of the form $\mathbb{E}_{D(u)} \langle g, u\rangle uu^{\mathsf{T}}$, for a random vector g. [4] This serves to permit much weaker conditions on D to be required.

For the rest of this subsection, we assume that we have an actual distribution D that is supported on vectors close to some orthonormal basis $a_1, \ldots, a_d$ of $\mathbb{R}^d$, and we will design a rounding algorithm that extracts the vectors a_i from the low-degree moments of D. This is a much simpler task than rounding from a pseudo-distribution, though it captures most of the essential difficulties. Since pseudo-distributions behave similarly to actual distributions on the low-degree moments, the techniques involved in rounding from actual distributions will turn out to be easily generalizable to the case of pseudo-distributions.

Let D be a distribution over the unit sphere in $\mathbb{R}^d$. Suppose that this distribution is supported on vectors close to some orthonormal basis $a_1, \ldots, a_d$ of $\mathbb{R}^d$, in the sense that the distribution satisfies the constraint

$$\left\{ \sum_{i=1}^d \langle a_i, u\rangle^3 \geq 1 - \varepsilon \right\}_{D(u)} . \tag{6}$$

(This constraint implies $\{\max_{i \in [d]} \langle a_i, u\rangle \geq 1 - \varepsilon\}_{D(u)}$ because $\sum_{i=1}^d \langle a_i, u\rangle^3 \leq \max_{i\in[d]} \langle a_i, u\rangle$ by orthonormality.) The analysis of [14] shows that reweighing the distribution D by a function of the form $u \mapsto \langle g, u\rangle^{2k}$ for $g \sim \mathcal{N}(0, \mathrm{Id}_d)$ and some $k \leq O(\log d)$ creates, with significant probability, a distribution D' such that for one of the basis vectors a_i, almost all of the probability mass of D' is on vectors close to a_i, in the sense that

$$\max_{i \in [d]} \mathop{\mathbb{E}}_{D'(u)} \langle a_i, u\rangle \geq 1 - O(\varepsilon)\,, \text{where } D'(u) \propto \langle g, u\rangle^{2k} D(u).$$

In this case, we can extract a vector close to one of the vectors a_i by computing the mean $\mathbb{E}_{D'(u)} u$ of the

[4] In previous treatments of simultaneous diagonalization, multiple matrices would be used for noise tolerance—increasing the confidence in the solution when more than one matrix agrees on a particular singular vector. This is unnecessary in our setting, since as we'll see, the SoS framework itself suffices to certify the correctness of a solution.

reweighted distribution. This rounding procedure takes quasi-polynomial time because it requires access to logarithmic-degree moments of the original pseudo-distribution D.

To avoid this quasi-polynomial running time, our strategy is to instead modify the original distribution D in order to create a small bias in one of the directions a_i such that a modified moment matrix of D has a one-dimensional eigenspace close to a_i. (This kind of modification is much less drastic than the kind of modification in previous works. Indeed, reweighing a distribution such that it concentrates around a particular vector seems to require logarithmic degree.)

Concretely, we will study the spectrum of matrices of the following form, for $g \sim \mathcal{N}(0, \mathrm{Id}_d)$:

$$M_g = \mathop{\mathbb{E}}_{D(u)} \langle g, u\rangle \cdot uu^{\mathsf{T}}.$$

Our goal is to show that with good probability, M_g has a one-dimensional eigenspace close to one of the vectors a_i.

However, this is not actually true for a naïve distribution: although we have encoded the basis vectors a_i into the distribution D by means of (6), we cannot yet conclude that the eigenspaces of M_g have anything to do with them. We can understand this as the error allowed in (6) being highly under-constrained. For example, the distribution could be a uniform mixture of vectors of the form $a_i + \varepsilon w$ for some fixed vector w, which causes w to become by far the most significant contribution to the spectrum of M_g. More generally, an arbitrary spectrally small error could still completely displace all of the eigenspaces of M_g.

An interpretation of this situation is that we have permitted D itself to contain a large amount of information that we do not actually possess. Equation (6) is consistent with a wide range of possible solutions, yet in the pathological example above, the distribution does not at all reflect this uncertainty, instead settling arbitrarily on some particular biased solution: it is this bias that disrupts the usefulness of the rounding procedure.

A similar situation has previously arisen in strategies for rounding convex relaxations—specifically, when the variables of the relaxations were interpreted as the marginals of some probability distribution over solutions, then actual solutions were constructed by sampling from that distribution. In that context, a workaround was to sample those solutions from the maximum-entropy distributions consistent with those marginals [20], to ensure that the distribution faithfully reflected the ignorance inherent in the relaxation solution rather than incorporating arbitrary information. Our situation differs in that it is the solution to the convex relaxation itself which is misbehaving, rather than some aspect of the rounding process, but the same approach carries over here as well.

Therefore, suppose that D satisfies the maximum-entropy constraint $\|\mathbb{E}_{D(u)}\, uu^{\mathsf{T}}\| \leq 1/n$. This essentially enforces D to be a uniform distribution over vectors close to $a_1, \ldots, a_n$. For the sake of demonstration, we assume that D is a uniform distribution over $a_1, \ldots, a_n$. Moreover, since our algorithm is invariant under linear transformations, we may assume that the components $a_1, \ldots, a_n$ are the standard basis vectors $e_1, \ldots, e_n \in \mathbb{R}^d$. We first decompose M_g along the coordinate g_1,

$$M_g = g_1 \cdot M_{e_1} + M_{g'}, \quad \text{where } g' = g - g_1 \cdot e_1 .$$

Note that under our simplified assumption for D, by simple algebraic manipulation we have $M_{e_1} = \mathbb{E}_{D(u)}\, u_1 uu^{\mathsf{T}} = e_1 e_1{}^{\mathsf{T}}$. Moreover, by definition, g_1 and g' are independent. It turns out that the entropy constraint implies $\mathbb{E}_{g'} \|M_{g'}\| \lesssim \sqrt{\log d} \cdot 1/n$ (using concentration bounds for Gaussian matrix series [18]). Therefore, if we condition on the event $g_1 > \eta^{-1}\sqrt{\log d}$, we have that $M_g = g_1 e_1 e_1{}^{\mathsf{T}} + M_{g'}$ consists of two parts: a rank-1 single part $g_1 e_1 e_1{}^{\mathsf{T}}$ with with eigenvalue larger than $\eta^{-1}\sqrt{\log d}$, and a noise part which has spectral norm at most $\lesssim \sqrt{\log d}$. Hence, by the eigenvector perturbation theorem we have that the top eigenvector is $O(\eta^{1/2})$-close to e_1 as desired.

Taking $\eta = 0.1$, we see with $1/\operatorname{poly}(d)$ probability the event $g_1 > \eta^{-1}\sqrt{\log d}$ will happen, and therefore by repeating this procedure $\operatorname{poly}(d)$ times, we obtain a vector that is $O(\eta^{1/2})$-close to e_1. We can find other vectors similarly by repeating the process (in a slightly more delicate way), and the accuracy can also be boosted (see full version of this paper for details).

B. Overcomplete fourth-order tensor

In this section, we give a high-level description of a robust sum-of-squares version of the tensor decomposition algorithm FOOBI [23]. For simplicity of the demonstration, we first work with the noiseless case where we are given a tensor $T \in (\mathbb{R}^d)^{\otimes 4}$ of the form

$$T = \sum_{i=1}^{n} a_i^{\otimes 4} . \tag{7}$$

We will first review the key step of FOOBI algorithm and then show how to convert it into a sum-of-squares algorithm that will naturally be robust to noise.

To begin with, we observe that by viewing T as a $d^2 \times d^2$ matrix of rank n, we can easily find the span of the $a_i^{\otimes 2}$'s by low-rank matrix factorization. However, since the low rank matrix factorization is only unique up to unitary transformation, we are not able to recover the $a_i^{\otimes 2}$'s from the subspace that they live in. The key observation of [23] is that the $a_i^{\otimes 2}$'s are actually the only "rank-1" vectors in the span, under a mild algebraic independence condition. Here, a d^2-dimensional vector is called "rank-1" if it is a tensor product of two vectors of dimension d.

Lemma 9 ([23]). *Suppose the following set of vectors is linearly independent,*

$$\left\{ a_i^{\otimes 2} \otimes a_j^{\otimes 2} - (a_i \otimes a_j)^{\otimes 2} \;\middle|\; i \neq j \right\} . \tag{8}$$

Then every vector $x^{\otimes 2}$ in the linear span of $a_1^{\otimes 2}, \ldots, a_n^{\otimes 2}$ is a multiple of one of the vectors $a_i^{\otimes 2}$.

This observation leads to the algorithm FOOBI, which essentially looks for rank-1 vectors in the span of $a_i^{\otimes 2}$'s. The main drawback is that it uses simultaneous diagonalization as a sub-procedure, which is unlikely to tolerate noise better than inverse polynomial in d, and in fact no noise tolerance guarantee has been explicitly shown for it before.

Our approach starts with rephrasing the original proof of Lemma 9 into the following SoS proof (which only uses polynomial inequalities that can be proved by SoS).

Proof of Lemma 9: Let $\alpha_1, \ldots, \alpha_n$ be multipliers such that $x^{\otimes 2} = \sum_{i=1}^n \alpha_i \cdot a_i^{\otimes 2}$.[5] Then, these multipliers satisfy the following quadratic equations:

$$x^{\otimes 4} = \sum\nolimits_{i,j} \alpha_i \alpha_j \cdot a_i^{\otimes 2} \otimes a_j^{\otimes 2} ,$$
$$x^{\otimes 4} = \sum\nolimits_{i,j} \alpha_i \alpha_j \cdot (a_i \otimes a_j)^{\otimes 2} .$$

Together, the two equations imply that

$$0 = \sum\nolimits_{i \neq j} \alpha_i \alpha_j \cdot \left(a_i^{\otimes 2} \otimes a_j^{\otimes 2} - (a_i \otimes a_j)^{\otimes 2} \right) .$$

By assumption, the vectors $a_i^{\otimes 2} \otimes a_j^{\otimes 2} - (a_i \otimes a_j)^{\otimes 2}$ are linearly independent for $i \neq j$. Therefore, from the equation above, we conclude $\sum_{i \neq j} \alpha_i^2 \alpha_j^2 = 0$, meaning that at most one of α_i can be non-zero. We note that the argument here is indeed a SoS proof since for any matrix A with linearly independent columns, the inequality $\|x\|^2 \leq \frac{1}{\sigma_{\min}(A)^2} \|Ax\|^2$ can be proved by SoS (here $\sigma_{\min}(A)$ denotes the least singular value of matrix A). ■

When there is noise present, we cannot find the true subspace of the $a_i^{\otimes 2}$'s and instead we only have an approximation, denoted by V, of that subspace. We will modify the proof above by starting with a polynomial inequality

$$\| \mathrm{Id}_V \, x^{\otimes 2} \|^2 \geq (1 - \delta) \| x^{\otimes 2} \|^2 , \qquad (9)$$

which constrains $x^{\otimes 2}$ to be close to the estimated subspace V (where δ is a small number that depends on error and condition number). Then an extension of the proof of Lemma 9 will show that (9) implies (via a SoS proof) that for some small enough δ,

$$\sum_{i \neq j} \alpha_i^2 \alpha_j^2 \leq o(1) . \qquad (10)$$

Note that $\alpha = K x^{\otimes 2}$ is a linear transformation of $x^{\otimes 2}$, and furthermore K is the pseudo-inverse of the matrix with columns $a_i^{\otimes 2}$. Moreover, if we assume for a moment that α has 2-norm 1 (which is not true in general), then the equation above further implies that

$$\sum_{i=1}^{n} \langle K_i, x^{\otimes 2} \rangle^4 = \|\alpha\|_4^4 \geq 1 - o(1) , \qquad (11)$$

where $K_i \in \mathbb{R}^{d^2}$ is the i-th row of K. This effectively gives us access to the 4-tensor $\sum_i K_i^{\otimes 4}$ (which has ambient dimension d^2 when flattened into a matrix), since (11) is anyway the constraint that would have been used by the SoS algorithm if given the tensor $\sum_i K_i^{\otimes 4}$ as input. Note that because the K_i are not necessarily (close to) orthogonal, we cannot apply the SoS orthogonal tensor decomposition algorithm directly. However, since we are working with a 4-tensor whose matrix flattening has higher dimension d^2, we can whiten K_i effectively in the SoS framework and then use the orthogonal SoS tensor decomposition algorithm to find the K_i's, which will in turn yield the a_i's.

Many details were omitted in the heuristic argument above (for example, we assumed α to have norm 1). The complete argument can be found in the full version of this paper.

C. Random overcomplete third-order tensor

In the random overcomplete setting, the input tensor is of the form

$$T = \sum_{i=1}^{n} a_i^{\otimes 3} + E ,$$

where each a_i is drawn uniformly at random from the Euclidean unit sphere, we have $d < n \leq d^{1.5} / (\log d)^{O(1)}$, and E is some noise tensor such that $\|E\|_{\{1\},\{2,3\}} < \varepsilon$ or alternatively such that a constant-degree sum-of-squares relaxation of the injective norm of E is at most ε.

Our original rounding approach depends on the target vectors a_i being orthonormal or nearly so. But when $n \gg d$ in this overcomplete setting, orthonormality fails badly: the vectors a_i are not even linearly independent.

We circumvent this problem by embedding the vectors a_i in a larger ambient space—specifically by taking the tensor powers $a'_1 = a_1^{\otimes 2}, \ldots, a'_n = a_n^{\otimes 2}$. Now the vectors $a'_1, \ldots, a'_n$ are linearly independent (with probability 1) and actually close to orthonormal with high probability. Therefore, if we had access to the order-6 tensor $\sum (a'_i)^{\otimes 3} = \sum a_i^{\otimes 6}$, then we could (almost) apply our rounding method to recover the vectors a'_i.

The key here will be to use the sum-of-squares method to generate a pseudo-distribution over the unit sphere having T as its third-order moments tensor, and then to extract from it the set of order-6 pseudo-moments estimating the moment tensor $\sum_i a_i^{\otimes 6}$. This pseudo-distribution would obey the constraint $\{ (u \otimes u \otimes u)^{\mathsf{T}} T \geq 1 - \varepsilon \}$, which implies the constraint $\{ \sum_i \langle a_i, u \rangle^3 \geq 1 - \varepsilon \}$, saying, informally, that our pseudo-distribution is close to the actual uniform distribution over $\{a_i\}$. Substituting $v = u^{\otimes 2}$, we obtain an implied pseudo-distribution in v which therefore ought to be close to the uniform distribution over $\{a'_i\}$, and we should therefore be able to round the order-3 pseudo-moments of v to recover $\{a'_i\}$.

Only two preconditions need to be checked: first that $\sum_i (a'_i)(a'_i)^{\mathsf{T}}$ is not too large in spectral norm, and second

[5] technically, $\alpha_1, \ldots, \alpha_n$ are polynomials in x so that $x^{\otimes 2} = \sum_{i=1}^n \alpha_i \cdot a_i^{\otimes 2}$ holds

that our pseudo-distribution in v satisfies the constraint $\{\sum_i \langle a_i', v\rangle^3 \geq 1 - O(\varepsilon)\}$. The first precondition is true (except for a spurious eigenspace which can harmlessly be projected away) and is essentially equivalent to a line of matrix concentration arguments previously made in [7]. The second precondition follows from a line of constant-degree sum-of-squares proofs, notably extending arguments made in [5] stating that the constraints $\{\sum_i \langle a_i, u\rangle^3 \geq 1 - \varepsilon, \|u\|^2 = 1\}$ imply with constant-degree sum-of-squares proofs that $\{\sum_i \langle a_i, u\rangle^k \geq 1 - O(\varepsilon) - \tilde{O}(n/d^{3/2})\}$ for some higher powers k. The rigorous verification of these conditions is given in the full version of this paper.

Acknowledgments

T.M. was supported by the Simons Award in Theoretical Computer Science and an IBM Ph. D. Fellowship. D.S. was supported by a Microsoft Research Fellowship, an Alfred P. Sloan Fellowship, an NSF CAREER award, and the Simons Collaboration for Algorithms and Geometry. J.S. also acknowledges the support of D.S.'s NSF CAREER award.

References

[1] J. Håstad, "Tensor rank is np-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990. [Online]. Available: http://dx.doi.org/10.1016/0196-6774(90)90014-6 1

[2] C. J. Hillar and L. Lim, "Most tensor problems are np-hard," *J. ACM*, vol. 60, no. 6, p. 45, 2013. [Online]. Available: http://doi.acm.org/10.1145/2512329 1

[3] A. Anandkumar, R. Ge, and M. Janzamin, "Learning overcomplete latent variable models through tensor methods," in *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6*, 2015, pp. 36–112. [Online]. Available: http://jmlr.org/proceedings/papers/v40/Anandkumar15.html 1

[4] ——, "Analyzing tensor power method dynamics: Applications to learning overcomplete latent variable models," *CoRR*, vol. abs/1411.1488, 2014. [Online]. Available: http://arxiv.org/abs/1411.1488 1

[5] R. Ge and T. Ma, "Decomposing overcomplete 3rd order tensors using sum-of-squares algorithms," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, Princeton, NJ, USA*, 2015, pp. 829–849. [Online]. Available: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2015.829 1, 2, 3, 5, 8

[6] S. B. Hopkins, J. Shi, and D. Steurer, "Tensor principal component analysis via sum-of-square proofs," in *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6*, 2015, pp. 956–1006. [Online]. Available: http://jmlr.org/proceedings/papers/v40/Hopkins15.html 1, 3

[7] S. B. Hopkins, T. Schramm, J. Shi, and D. Steurer, "Fast spectral algorithms from sum-of-squares proofs: Tensor decomposition and planted sparse vectors," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2016. New York, NY, USA: ACM, 2016, pp. 178–191. [Online]. Available: http://doi.acm.org/10.1145/2897518.2897529 1, 2, 3, 8

[8] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2773–2832, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2697055 1

[9] A. Bhaskara, M. Charikar, A. Moitra, and A. Vijayaraghavan, "Smoothed analysis of tensor decompositions," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03*, 2014, pp. 594–603. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591881 1, 3

[10] N. Goyal, S. Vempala, and Y. Xiao, "Fourier PCA and robust tensor decomposition," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03*, 2014, pp. 584–593. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591875 1

[11] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade, "A tensor approach to learning mixed membership community models," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2239–2312, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2670323 1

[12] R. Ge, Q. Huang, and S. M. Kakade, "Learning mixtures of gaussians in high dimensions," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17*, 2015, pp. 761–770. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746616 1

[13] A. Anandkumar, D. P. Foster, D. Hsu, S. M. Kakade, and Y. Liu, "A spectral algorithm for latent dirichlet allocation," *Algorithmica*, vol. 72, no. 1, pp. 193–214, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00453-014-9909-1 1

[14] B. Barak, J. A. Kelner, and D. Steurer, "Dictionary learning and tensor decomposition via the sum-of-squares method," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17*, 2015, pp. 143–151. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746605 1, 2, 4, 5

[15] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis," 1970. 1, 5

[16] S. Leurgans, R. Ross, and R. Abel, "A decomposition for three-way arrays." *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 4, pp. 1064–1083, 1993. 1, 5

[17] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models (A survey for ALT)," in *Algorithmic Learning Theory - 26th International Conference, ALT 2015, Banff, AB, Canada, October 4-6, Proceedings*, 2015, pp. 19–38. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24486-0_2 1

[18] R. I. Oliveira, “Sums of random Hermitian matrices and an inequality by Rudelson.” *Electron. Commun. Probab.*, vol. 15, pp. 203–212, 2010. 2, 6

[19] J. A. Tropp, “User-friendly tail bounds for sums of random matrices,” *Foundations of Computational Mathematics*, vol. 12, no. 4, pp. 389–434, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10208-011-9099-z 2

[20] S. O. Gharan, “New rounding techniques for the design and analysis of approximation algorithms,” Ph.D. dissertation, STANFORD UNIVERSITY, 2014. 2, 6

[21] J. R. Lee, P. Raghavendra, and D. Steurer, “Lower bounds on the size of semidefinite programming relaxations,” in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, ser. STOC ’15. New York, NY, USA: ACM, 2015, pp. 567–576. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746599 2

[22] B. Barak, S. B. Hopkins, J. Kelner, P. K. Kothari, A. Moitra, and A. Potechin, “A nearly tight sum-of-squares lower bound for the planted clique problem,” in *FOCS*. IEEE Computer Society, 2016. 2

[23] L. D. Lathauwer, J. Castaing, and J. Cardoso, “Fourth-order cumulant-based blind identification of underdetermined mixtures,” *IEEE Transactions on Signal Processing*, vol. 55, no. 6-2, pp. 2965–2973, 2007. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TSP.2007.893943 3, 6

[24] T. Schramm and B. Weitz, “Low-rank matrix completion with adversarial missing entries,” *CoRR*, vol. abs/1506.03137, 2015. [Online]. Available: http://arxiv.org/abs/1506.03137 4

[25] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?” *Vision Research*, vol. 37, no. 23, pp. 3311 – 3325, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698997001697 4

[26] ——, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, pp. 607–609, 1996. 4

[27] ——, “Natural image statistics and efficient coding*,” *Network: computation in neural systems*, vol. 7, no. 2, pp. 333–339, 1996. 4

[28] A. A. T. Evgeniou and M. Pontil, “Multi-task feature learning,” in *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, vol. 19. MIT Press, 2007, pp. 41–48. 4

[29] Y. Marc’Aurelio Ranzato, L. Boureau, and Y. LeCun, “Sparse feature learning for deep belief networks,” *Advances in neural information processing systems*, vol. 20, pp. 1185–1192, 2007. 4

[30] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006. 4

[31] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, “Discriminative sparse image models for class-specific edge detection and image interpretation,” in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 43–56. 4

[32] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. 4