# Dictionary Learning and Tensor Decomposition via the Sum-of-Squares Method

Boaz Barak* Jonathan A. Kelner† David Steurer‡

July 6, 2014

**Abstract**

We give a new approach to the dictionary learning (also known as "sparse coding") problem of recovering an unknown $n \times m$ matrix $A$ (for $m \geqslant n$) from examples of the form

$$y = Ax + e,$$

where $x$ is a random vector in $\mathbb{R}^m$ with at most $\tau m$ nonzero coordinates, and $e$ is a random noise vector in $\mathbb{R}^n$ with bounded magnitude. For the case $m = O(n)$, our algorithm recovers every column of $A$ within arbitrarily good constant accuracy in time $m^{O(\log m/\log(\tau^{-1}))}$, in particular achieving polynomial time if $\tau = m^{-\delta}$ for any $\delta > 0$, and time $m^{O(\log m)}$ if $\tau$ is (a sufficiently small) constant. Prior algorithms with comparable assumptions on the distribution required the vector $x$ to be much sparser—at most $\sqrt{n}$ nonzero coordinates—and there were intrinsic barriers preventing these algorithms from applying for denser $x$.

We achieve this by designing an algorithm for *noisy tensor decomposition* that can recover, under quite general conditions, an approximate rank-one decomposition of a tensor $T$, given access to a tensor $T'$ that is $\tau$-close to $T$ in the spectral norm (when considered as a matrix). To our knowledge, this is the first algorithm for tensor decomposition that works in the constant spectral-norm noise regime, where there is no guarantee that the local optima of $T$ and $T'$ have similar structures.

Our algorithm is based on a novel approach to using and analyzing the *Sum of Squares* semidefinite programming hierarchy (Parrilo 2000, Lasserre 2001), and it can be viewed as an indication of the utility of this very general and powerful tool for unsupervised learning problems.

**Keywords:** sparse coding, dictionary learning, sum-of-squares method, semidefinite programming, machine learning, unsupervised learning, statistical recovery, approximation algorithms, tensor optimization, polynomial optimization.

*Microsoft Research.

†Department of Mathematics, Massachusetts Institute of Technology.

‡Department of Computer Science, Cornell University.

# Contents

# 1 Introduction

The *dictionary learning* (also known as "sparse coding") problem is to recover an unknown $m \times n$ matrix $A$ (known as a "dictionary") from examples of the form

$$y = Ax + e \,, \tag{1.1}$$

where $x$ is sampled from a distribution over *sparse* vectors in $\mathbb{R}^n$ (i.e., with much fewer than $n$ nonzero coordinates), and $e$ is sampled from a distribution over noise vectors in $\mathbb{R}^m$ of some bounded magnitude.

This problem has found applications in multiple areas, including computational neuroscience [OF97, OF96a, OF96b], machine learning [EP07, MRBL07], and computer vision and image processing [EA06, MLB+08, YWHM08]. The appeal of this problem is that, intuitively, data should be sparse in the "right" representation (where every coordinate corresponds to a meaningful feature), and finding this representation can be a useful first step for further processing, just as representing sound or image data in the Fourier or wavelet bases is often a very useful preprocessing step in signal or image processing. See [SWW12, AAJ+13, AGM13, ABGM14] and the references therein for further discussion of the history and motivation of this problem.

This is a non-linear problem, as both $A$ and $x$ are unknown, and dictionary learning is a computationally challenging task even in the noiseless case. When $A$ is known, recovering $x$ from $y$ constitutes the *sparse recovery / compressed sensing* problem, which has efficient algorithms [Don06, CRT06]. Hence, a common heuristic for dictionary learning is to use alternating minimization, using sparse recovery to obtain a guess for $x$ based on a guess of $A$, and vice versa.

Recently there have been several works giving dictionary learning algorithms with rigorous guarantees on their performance [SWW12, AAJ+13, AAN13, AGM13, ABGM14]. These works differ in various aspects, but they all share a common feature: they give no guarantee of recovery unless the distribution $\{x\}$ is over *extremely sparse* vectors, namely having less than $\sqrt{n}$ (as opposed to merely $o(n)$) nonzero coordinates. (There have been other works dealing with the less sparse case, but only at the expense of making strong assumptions on $x$ and/or $A$; see Section 1.3 for more discussion of related works.)

In this work we give a different algorithm that can be proven to approximately recover the matrix $A$ even when $x$ is much denser (up to $\tau n$ coordinates for some small constant $\tau > 0$ in some settings). The algorithm works even with noise, and in the so-called overcomplete case (where $m > n$). We note that it only recovers the vectors approximately, as exact recovery is impossible (even from an identifiability standpoint) unless one makes further assumptions.

Our algorithm is based on the *Sum of Squares* (SOS) semidefinite programming hierarchy [Sho87, Nes00, Par00, Las01]. The SOS algorithm is a very natural method for solving non-convex optimization problems, that had found applications in a variety of scientific fields, including control theory [HG05], quantum information theory [DPS02], game theory [Par06], formal verification [Har07],

and more. Nevertheless, to our knowledge this work provides the first rigorous bounds on the SOS algorithm's running time for a natural unsupervised learning problem.

Our algorithm's running running time (which ranges from a large polynomial to quasipolynomial time, depending on various parameters) leaves much room for improvement. Indeed, the way it is described, our algorithm is strictly theoretical, as general-purpose Sum-of-Square solvers currently work only for a fairly small number of variables. Nonetheless we hope that, as was the case with other algorithms based on semidefinite programming or large linear programming relaxations [GP04, She09, LRS+10, AGH+13], the ideas behind this algorithm and its analysis can form the basis for more efficient algorithms, ideally ones that can be implemented in practice.

### 1.1 Problem definition and conditions on coefficient distribution

In this section we formally define the dictionary problem and state our result. We define a *σ-dictionary* to be an $m \times n$ matrix $A = (a^1|\cdots|a^m)$ such that $\|a^i\| = 1$ for all $i$, and $A^\top A \preceq \sigma I$ (where $I$ is the identity matrix). The parameter $\sigma$ is an analytical proxy for the overcompleteness $m/n$ of the dictionary $A$. In particular, if the columns of $A$ are in isotropic position (i.e., $A^\top A$ is proportional to the identity), then the top eigenvalue of $A^\top A$ is its trace divided by $n$, which equals $(1/n)\sum_i \|a^i\|^2 = m/n$ because all of the $a^i$'s have unit norm.[1] In this work we are mostly interested in the case $m = O(n)$, which corresponds to $\sigma = O(1)$.

**Nice distributions.** Let $\{x\}$ be some distribution over the coefficients in (1.1). We will posit some conditions on low order moments of $\{x\}$ to allow recovery. Let $d$ be some even constant that we will use as a paramter (think of $d = O(1)$). Consider a 0/1 vector $x \in \mathbb{R}^m$ with $\tau m$ nonzero coordinates. Then $\frac{1}{m}\sum_{k\in[m]} x_i^d = \tau$ and $(\frac{1}{m}\sum x_i^{d/2})^2 = \tau^2$. In other words, if we select "typical" three coordinates $i, j, k$, then

$$x_i^{d/2} x_j^{d/2} \leqslant \tau x_k^{d/2} \tag{1.2}$$

Equation (1.2) will motivate us in defining an analytical proxy for the condition that the distribution $\{x\}$ over coefficients is $\tau$-sparse.[2]

Specifically, in the dictionary learning case, since we are interested in learning *all* column vectors, we want every coordinate $i$ to typical (for example, if the coefficient $x_i$ is always 0 or always 1, we will not be able to learn the corresponding

[1] While we do not use it in this paper, we note that in the dictionary learning problem it is always possible to learn a linear "whitening transformation" $B$ from the samples that would place the columns in isotropic position, at the cost of potentially changing the norms of the vectors. (There also exists a linear transformation that keeps the vectors normalized [Bar98, For01], but we do not know how to learn it from the samples.)

[2] By using an analytical proxy as opposed to requiring strict sparsity, we are only enlarging the set of distributions under consideration. However, we will make some additional conditions below, and in particular requiring low order non-square moments to vanish, that although seemingly mild compared to prior works, do restrict the family of distributions.

column vector). Moreover, a necessary condition for recovery is that every *pair* of coordinates is somewhat typical in the sense that the events that $x_i$ and $x_j$ are nonzero are not perfectly correlated. Indeed, suppose for simplicity that when $x_i$ is nonzero, then it is distributed like an independent standard Gaussian. Then if those two events were perfectly correlated, recovery would be impossible since the distribution over examples would be identical if we replaced $\{a_i, a_j\}$ with any pair of vectors $\{\Pi a^i, \Pi a^j\}$ where $\Pi$ is a rotation in the plane spanned by $\{a^i, a^j\}$.

Given these considerations, if we normalize our distribution so that $\mathbb{E}\, x_i^d = 1$ for all $i$, then it makes sense to assume:[3]

$$\mathbb{E}\, x_i^{d/2} x_j^{d/2} \leqslant \tau \ , \tag{1.3}$$

for all $i \neq j$ and for some $\tau \ll 1$.

We can assume without loss of generality that the marginal distribution $\{x_i\}$ is symmetric around zero (namely $\mathbb{P}[x_i = a] = \mathbb{P}[x_i = -a]$ for all $a$), since given two samples $y = Ax + e$ and $y' = Ax' + e'$ we can treat them as a single sample $y - y' = A(x - x') + e - e'$, and the distribution $x - x'$, which is only slightly less sparse (and slightly more noisy), has this property. In particular this means we can assume $\mathbb{E}\, x_i^{2k+1} = 0$ for every integer $k$ and $i \in [m]$. We will strengthen this condition to assume that

$$\mathbb{E}\, x^\alpha = 0 \tag{1.4}$$

for every non-square monomial $x^\alpha$ of degree at most $d$. (Here, $\alpha \in \{0, 1, \ldots\}^m$ is a *multiindex* and $x^\alpha$ denotes the monomial $\prod_i x_i^{\alpha_i}$. The degree of $x^\alpha$ is $|\alpha| := \sum_i \alpha_i$; we say that $x^\alpha$ is non-square if $x^\alpha$ is not the square of another monomial, i.e.,, if $\alpha$ has an odd coordinate.)

We say that a distribution $\{x\}$ is $(d, \tau)$-*nice* if it satisfies (1.3) and (1.4).[4] One example for a $(d, \tau)$-nice distribution is the *Bernoulli-Gaussian* distribution, where $x_i = y_i z_i$ with the $y_i$'s being independent 0/1 random variables satisfying $\mathbb{P}[y_i = 1] = \tau$ and the $z_i$'s being independent normally distributed random variables (normalized to satisfy $\mathbb{E}\, z_i^d = 1/\tau$). Indeed, in this case, since (using Cauchy-Schwarz) $\mathbb{E}\, z_i^{d/2} z_j^{d/2} \leqslant \sqrt{\mathbb{E}\, z_i^d\, \mathbb{E}\, z_j^d} = 1/\tau$,

$$\mathbb{E}\, x_i^{d/2} x_j^{d/2} = (\mathbb{E}\, y_i y_j)(\mathbb{E}\, z_i^{d/2} z_j^{d/2}) \leqslant \tau^2 (1/\tau) = \tau \ .$$

In fact, we can replace here the normal distribution with any distribution satisfying $\mathbb{E}\, z_i^d = 1$, and also allow some dependence between the variables (in particular encapsulating the models considered by [AGM13]). As our discussion above and this example demonstrates, the parameter $\tau$ serves as a proxy to the sparsity of $\{x\}$, where a $(d, \tau)$-nice distribution $\{x\}$ roughly corresponds to a distribution having at most $\tau n$ coordinates with significant mass. (For technical

[3] Our results generalize to the case where $\mathbb{E}\, x_i^d \in [c, C]$ for some constants $C > c > 0$.

[4] We will also assume that $\mathbb{E}\, x_i^{2d} \leqslant n^c$ for some constant $c$. This is a very mild assumption, and in some qualitative sense is necessary to avoid pathological cases such as a distribution that outputs the all zero vector with probability $1 - n^{-\omega(1)}$.

reasons, our formal definition of nice distributions, Definition 4.1, is somewhat different but is qualitatively equivalent to the above, see Remark 4.4.)

**Modeling noise.** Given a noisy dictionary learning example of the form $y = Ax + e$, one can also view it (assuming we are in the non-degenerate case of $A$ having full rank) as $y = A(x + e')$ for some $e'$. In fact, if $A$ is in "isotropic" or "whitened" form, then the norm of $e'$ is proportional to the norm of $e$. If $e$ has sufficiently small magnitude, and is comprised of i.i.d random variables (and even under some more general conditions), the distribution $\{x + e'\}$ will be nice as well. Therefore, we will not explicitly model the noise in the following, but rather treat it as part of the distribution $\{x\}$ which our definition allows to be only "approximately sparse".

## 1.2 Our results

Given samples of the form $\{y = Ax\}$ for a $(d, \tau)$-nice $\{x\}$, with $d$ a sufficiently large constant (corresponding to having $\tau n$ nonzero entries), we can approximately recover the dictionary $A$ in polynomial time as long as $\tau \leqslant n^{-\delta}$ for some $\delta > 0$, and in quasipolynomial time as long as $\tau$ is a sufficiently small constant. Prior polynomial-time algorithms required the distribution to range over vectors with less than $\sqrt{n}$ nonzero entries (and it was not known how to improve upon this even using quasipolynomial time).

We define the *correlation* of a pair of vectors $u$, $a$, to be $\mathrm{Cor}(u, a) = \langle u, a\rangle^2/(\|u\|\|a\|)^2$. We say that two sets $S, T$ of vectors are *$\varepsilon$-close* if for every $s \in S$ there is $t \in T$ such that $\mathrm{Cor}(s, t) \geqslant 1 - \varepsilon$, and for every $t \in T$ there is $s \in S$ such that $\mathrm{Cor}(s, t) \geqslant 1 - \varepsilon$.[5]

**Theorem 1.1** (Dictionary learning)**.** *For every $\varepsilon > 0, \sigma \geqslant 1, \delta > 0$ there exists $d$ and a polynomial-time algorithm $\mathcal{R}$ such that for every $\sigma$-dictionary $A = (a^1|\cdots|a^m)$ and $(d, \tau = n^{-\delta})$-nice $\{x\}$, given $n^{O(1)}$ samples from from $\{y = Ax\}$, $\mathcal{R}$ outputs with probability at least 0.9 a set that is $\varepsilon$-close to $\{a^1, \ldots, a^m\}$.*

The hidden constants in the $O(\cdot)$ notation may depend on $\varepsilon, \sigma, \delta$. The algorithm can recover the dictionary vectors even in the relatively dense case when $\tau$ is (a sufficiently small) constant, at the expense of a quasipolynomial (i.e., $n^{O(\log n)}$) running time. See Theorems 4.2 and 7.6 for a precise statement of the dependencies between the constants.

Our main tool is a new algorithm for the *noisy tensor decomposition problem*, which is of interest in its own right. This is the problem of recovering the set $\{a^1, \ldots, a^m\}$ of vectors given access to a noisy version of the polynomial $\sum_{i=1}^m \langle a^i, u\rangle^d = \|A^\top u\|_d^d$ in $\mathbb{R}[u]$, where $A = (a^1|\cdots|a^m)$ is an $n \times m$ matrix.[6] We

[5] This notion corresponds to the sets $\{s/\|s\| : s \in S\} \cup \{-s/\|s\| : s \in S\}$ and $\{t/\|t\| : t \in T\} \cup \{-t/\|t\| : t \in T\}$ being close in Hausdorff distance, which makes sense in our setting, since we can only hope to recover the dictionary columns up to permutation and scaling.

[6] For a vector $v \in \mathbb{R}^m$ and $p \geqslant 1$, we define $\|v\|_p = (\sum_i |v_i|^p)^{1/p}$.

give an algorithm that is worse than prior works in the sense that it requires a higher value of $d$, but can handle a much larger level of noise than these previous algorithms. The latter property turns out to be crucial for the dictionary learning application. Our result for noisy tensor decomposition is captured by the following theorem:

**Theorem 1.2** (Noisy tensor decomposition)**.** *For every $\varepsilon > 0, \sigma \geqslant 1$, there exists $d, \tau$ and a probabilistic $n^{O(\log n)}$-time algorithm $\mathcal{R}$ such that for every $\sigma$-dictionary $A = (a^1|\cdots|a^m)$, given a polynomial $P$ such that*

$$\|A^\top u\|_d^d - \tau\|u\|_2^d \preceq P \preceq \|A^\top u\|_d^d + \tau\|u\|_2^d \,, \tag{1.5}$$

*$\mathcal{R}$ outputs with probability at least 0.9 a set $S$ that is $\varepsilon$-close to $\{a^1, \ldots, a^m\}$.*

(We denote $P \preceq Q$ if $Q - P$ is a sum of squares of polynomials. Also, as in Theorem 1.1, there are certain conditions under which $\mathcal{R}$ runs in polynomial time; see Section 7.)

The condition (1.5) implies that the input $P$ to $\mathcal{R}$ is $\tau$-close to the tensor $\|A^\top u\|_d^d$, in the sense that $|P(u) - \|A^\top u\|_d^d| \leqslant \tau$ for every unit vector $u$. This allows for very significant noise, since for a typical vector $u$, we expect $\|A^\top u\|_d^d$ to be have magnitude roughly $mn^{-d/2}$ which would be *much* smaller than $\tau$ for every constant $\tau > 0$. Thus, on most of its inputs, $P$ can behave radically differently than $\|A^\top u\|_d^d$, and in particular have many local minima that do not correspond to local minima of the latter polynomial. For this reason, it seems unlikely that one can establish a result such as Theorem 1.2 using a local search algorithm.[7]

We give an overview of our algorithm and its analysis in Section 2. Sections 4, 6 and 5 contain the full formal proofs. In its current form, our algorithm is efficient only in the theoretical/asymptotic sense, but it is very simple to describe (modulo its calls to the SOS solver), see Figure 1. We believe that the Sum of Squares algorithm can be a very useful tool for attacking machine learning problems, yielding a first solution to the problem that can later be tailored and optimized.

### 1.3 Related work

Starting with the work of Olshausen and Field [OF96a, OF96b, OF97], there is a vast body of literature using various heuristics (most commonly alternating minimization) to learn dictionaries for sparse coding, and applying this tool to many applications. Here we focus on papers that gave algorithms with *proven* performance.

*Independent Component Analysis (ICA)* [Com94] is one method that can be used for the dictionary learning in the case the random variables $x_1, \ldots, x_n$ are statistically independent. For the case of $m = n$ this was shown in [Com94,

[7] The conditions (1.5) and $\max_{\|u\|_2=1} |P(u) - \|A^\top u\|_d^d| \leqslant \tau$ are not identical for $d > 2$. Nevertheless, the discussion above applies to both conditions, since (1.5) does allow for $P$ to have very different behavior than $\|A^\top u\|_d^d$.

FJK96, NR09], while the works [LCC07, GVX14] extend it for the overcomplete (i.e. $m > n$) case.

Another recent line of works analyzed different algorithms, which in some cases are more efficient or handle more general distributions than ICA. Spielman, Wang and Wright [SWW12] give an algorithm to exactly recover the dictionary in the $m = n$ case. Agarwal, Anandkumar, Jain, Netrapalli, and Tandon [AAJ+13] and Arora, Ge and Moitra [AGM13] obtain approximate recovery in the overcomplete (i.e. $m > n$) case, which can be boosted to exact recovery under some additional conditions on the sparsity and dictionary [AAN13, AGM13]. However, all these works require the distribution $x$ to be over *very* sparse vectors, specifically having less than $\sqrt{n}$ nonzero entries. As discussed in [SWW12, AGM13], $\sqrt{n}$ sparsity seemed like a natural barrier for this problem, and in fact, Spielman et al [SWW12] proved that every algorithm of similar nature to theirs will fail to recover the dictionary when when the coefficient vector can have $\Omega(\sqrt{n \log n})$ coordinates. The only work we know of that can handle vectors of support larger than $\sqrt{n}$ is the recent paper [ABGM14], but it achieves this at the expense of making fairly strong assumptions on the structure of the dictionary, in particular assuming some sparsity conditions on $A$ itself. In addition to the sparsity restriction, all these works had additional conditions on the distribution that are incomparable or stronger than ours, and the works [AAJ+13, AGM13, AAN13, ABGM14] make additional assumptions on the dictionary (namely incoherence) as well.

The tensor decomposition problem is also very widely studied with a long history (see e.g., [Tuc66, Har70, Kru77]). Some recent works providing algorithms and analysis include [AFH+12, AGM12, BCMV14, BCV14]. However, these works are in a rather different parameter regime than ours— assuming the tensor is given with very little noise (inverse polynomial in the spectral norm), but on the other hand requiring very low order moments (typically three or four, as opposed to the large constant or even logarithmic number we use).

As described in Sections 2 and 2.1 below, the main tool we use is the *Sum of Squares* (SOS) semidefinite programming hierarchy [Sho87, Nes00, Par00, Las01]. We invoke the SOS algorithm using the techniques and framework introduced by Barak, Kelner and Steurer [BKS14]. In addition to introducing this framework, [BKS14] showed how a somewhat similar technical barrier can be bypassed in a setting related to dictionary learning— the task of recovering a sparse vector that is planted in a random subspace of $\mathbb{R}^n$ given a basis for that subspace. Assuming the subspace has dimension at most $d$, [BKS14] showed that the vector can be recovered as long as it has less than $\min(\varepsilon n, n^2/d^2)$ nonzero coordinates for some constant $\varepsilon > 0$, thus improving (for $d \ll n^{2/3}$) on the prior work [DH13] that required the vector to be $o(n/\sqrt{d})$ sparse.

## Organization of this paper

In Section 2 we give a high level overview of our ideas. Sections 4–6 contain the full proof for solving the dictionary learning and tensor decomposition problems

in quasipolynomial time, where the sparsity parameter $\tau$ is a small constant. In Section 7 we show how this can be improved to polynomial time when $\tau \leqslant n^{-\delta}$ for some constant $\delta > 0$.

## 2 Overview of algorithm and its analysis

The dictionary learning problem can be easily reduced to the noisy tensor decomposition problem. Indeed, it is not too hard to show that for an appropriately chosen parameter $d$, given a sufficiently large number of examples $y_1, \ldots, y_N$ from the distribution $\{y = Ax\}$, the polynomial

$$P = \frac{1}{N} \sum_{i=1}^{N} \langle y_i, u \rangle^{2d} \tag{2.1}$$

will be roughly $\tau$ close (in the spectral norm) to the polynomial $\|A^\top u\|_d^d$, where $\tau$ is the "niceness"/"sparsity" parameter of the distribution $\{x\}$. Therefore, if we give $P$ as input to the tensor decomposition algorithm of Theorem 1.2, we will obtain a set that is close to the columns of $A$.[8]

The challenge is that because $\tau$ is a positive constant, no matter how many samples we take, the polynomial $P$ will always be bounded away from the tensor $\|A^\top u\|_d^d$. Hence we must use a tensor decomposition algorithm that can handle a very significant amount of noise. This is where the Sum-of-Squares algorithm comes in. This is a general tool for solving systems of polynomial equations [Sho87, Nes00, Par00, Las01]. Given the SOS algorithm, the description of our tensor decomposition algorithm is extremely simple (see Figure 1 below). We now describe the basic facts we use about the SOS algorithm, and sketch the analysis of our noisy tensor decomposition algorithm. See the survey [BS14] and the references therein for more detail on the SOS algorithm, and Sections 4, 5 and 6 for the full description of our algorithm and its analysis (including its variants that take polynomial time at the expense of requiring dictionary learning examples with sparser coefficients).

### 2.1 The SOS algorithm

The SOS algorithm is a method, based on semidefinite programming, for solving a system of polynomial equations. Alas, since this is a non-convex and NP-hard problem, the algorithm doesn't always succeed in producing a solution. However, it always returns some object, which in some sense can be interpreted as a "distribution" $\{u\}$ over solutions of the system of equations. It is not an actual distribution, and in particular we cannot sample from $\{u\}$ and get an individual

[8] The polynomial (2.1) and similar variants have been used before in works on dictionary learning. The crucial difference is that those works made strong assumptions, such as independence of the entries of $\{x\}$, that ensured this polynomial has a special structure that made it possible to efficiently optimize over it. In contrast, our work applies in a much more general setting.

solution, but we can compute low order moments of $\{u\}$. Specifically, we make the following definition:

**Definition 2.1** (Pseudo-expectations)**.** Let $\mathbb{R}[u]$ denote the ring of polynomials with real coefficients in variables $u = u_1 \dots u_n$. Let $\mathbb{R}[u]_k$ denote the set of polynomials in $\mathbb{R}[u]$ of degree at most $k$. A *degree-$k$ pseudoexpectation operator for* $\mathbb{R}[u]$ is a linear operator $\mathcal{L}$ that maps polynomials in $\mathbb{R}[u]_k$ into $\mathbb{R}$ and satisfies that $\mathcal{L}(1) = 1$ and $\mathcal{L}(P^2) \geqslant 0$ for every polynomial $P$ of degree at most $k/2$.

For every distribution $\mathcal{D}$ over $\mathbb{R}^n$ and $k \in \mathbb{N}$, the operator $\mathcal{L}$ defined as $\mathcal{L}(P) = \mathbb{E}_{\mathcal{D}} P$ for all $P \in \mathbb{R}[x]$ is degree $k$ pseudo-expectation operator. We will use notation that naturally extends the notation for actual expectations. We denote pseudoexpectation operators as $\tilde{\mathbb{E}}_{\mathcal{D}}$, where $\mathcal{D}$ acts as index to distinguish different operators. If $\tilde{\mathbb{E}}_{\mathcal{D}}$ is a degree-$k$ pseudoexpectation operator for $\mathbb{R}[u]$, we say that $\mathcal{D}$ is a *degree-$k$ pseudodistribution* for the indeterminates $u$. In order to emphasize or change indeterminates, we use the notation $\tilde{\mathbb{E}}_{v \sim \mathcal{D}} P(v)$. In case we have only one pseudodistribution $\mathcal{D}$ for indeterminates $u$, we denote it by $\{u\}$. In that case, we also often drop the subscript for the pseudoexpectation and write $\tilde{\mathbb{E}}\, P$ or $\tilde{\mathbb{E}}\, P(u)$ for $\tilde{\mathbb{E}}_{\{u\}} P$.

We say that a degree-$k$ pseudodistribution $\{u\}$ satisfies the constraint $\{P = 0\}$ if $\tilde{\mathbb{E}}\, P(u)Q(u) = 0$ for all $Q$ of degree at most $k - \deg P$. Note that this is a stronger condition than simply requiring $\tilde{\mathbb{E}}\, P(u) = 0$. We say that $\{u\}$ satisfies $\{P \geqslant 0\}$ if it satisfies the constraint $\{P - S = 0\}$ where $S$ is a sum-of-squares polynomial $S \in \mathbb{R}_k[u]$. It is not hard to see that if $\{u\}$ was an actual distribution, then these definitions imply that all points in the support of the distribution satisfy the constraints. We write $P \succeq 0$ to denote that $P$ is a sum of squares of polynomials, and similarly we write $P \succeq Q$ to denote $P - Q \succeq 0$.

A degree $k$ pseudo-distribution can be represented by the list of $n^{O(k)}$ values of the expectations of all monomials of degree up to $k$. It can also be written as an $n^{O(k)} \times n^{O(k)}$ matrix $M$ whose rows and columns correspond to monomials of degree up to $k/2$; the condition that $\tilde{\mathbb{E}}\, P(u)^2 \geqslant 0$ translates to the condition that this matrix is positive semidefinite. The latter observation can be used to prove the fundamental fact about pseudo-distributions, namely that we can efficiently optimize over them. This is captured in the following theorem:

**Theorem 2.2** (The SOS Algorithm [Sho87, Nes00, Par00, Las01])**.** *For every $\varepsilon > 0$, $k, n, m, M \in \mathbb{N}$ and $n$-variate polynomials $P_1, \dots, P_m$ in $\mathbb{R}_k[u]$, whose coefficients are in $\{0, \dots, M\}$, if there exists a degree $k$ pseudo-distribbution $\{u\}$ satisfying the constraint $\{P_i = 0\}$ for every $i \in [m]$, then we can find in $(n \operatorname{polylog}(M/\varepsilon))^{O(k)}$ time a pseudo-distribution $\{u'\}$ satisfying $\{P_i \leqslant \varepsilon\}$ and $\{P_i \geqslant -\varepsilon\}$ for every $i \in [m]$.*

Numerical accuracy will never play an important role in our results, and so we can just assume that we can always find in $n^{O(k)}$ time a degree-$k$ pseudo-distribution satisfying given polynomial constraints, if such a pseudo-distribution exists.

**Input:** Accuracy parameter $\varepsilon$. A degree $d$ polynomial $P$ such that

$$\|A^\top u\|_d^d - \tau\|u\|_d^d \leq P \leq \|A^\top u\|_d^d + \tau\|u\|_d^d$$

where $d$ is even.

**Operation:**

1. Use the SOS algorithm to find the degree-$k$ pseudo-distribution $\{u\}$ that maximizes $P(u)$ while satisfying $\|u\|^2 \equiv 1$.
2. Pick the polynomial $W$ to be a product of $O(\log n)$ random linear functions.
3. Output the top eigenvector of the matrix $M$ where $M_{i,j} = \tilde{\mathbb{E}}\, W(u)^2 u_i u_j$.

Figure 1: Basic Tensor Decomposition algorithm. The parameters $k, d, \tau$ are chosen as a function of the accuracy parameter $\varepsilon$ and the top eigenvalue $\sigma$ of $A^\top A$. The algorithm outputs a vector $u$ that is $\varepsilon$-close to a column of $A$ with inverse polynomial probability.

## 2.2 Noisy tensor decomposition

Our basic noisy tensor decomposition algorithm is described in Figure 1. This algorithm finds (a vector close to) a column of $A$ with inverse polynomial probability. Using similar ideas, one can extend it to an algorithm that outputs all vectors with high probability; we provide the details in Section 6. Following the approach of [BKS14], our analysis of this algorithm proceeds in two phases:

**(i)** We show that if the pseudo-distribution $\{u\}$ obtained in Step 1 is an *actual* distribution, then the vector output in Step 3 is close to one of the columns of $A$.

**(ii)** We then show that the arguments used in establishing **(i)** generalize to the case of pseudo-distributions as well.

**Part (i).** The first part is actually not so surprising. For starters, every unit vector $u$ that maximizes $P$ must be highly correlated with some column $a$ of $A$. Indeed, $\|A^\top a\|_d^d \geqslant 1$ for every column $a$ of $A$, and hence the maximum of $P(u)$ over a unit $u$ is at least $1 - \tau$. But if $\langle u, a\rangle^2 \leqslant 1 - \varepsilon$ for every column $a$ then $P(u)$ must be much smaller than 1. Indeed, in this case

$$\|A^\top u\|_d^d = \sum_i \langle a^i, u\rangle^d \leqslant \max_i \langle a^i, u\rangle^{d-2} \sum \langle a^i, u\rangle^2 \,. \tag{2.2}$$

Since $\sum \langle a^i, u\rangle^2 \leqslant \sqrt{\sigma}$, this implies that, as long as $d \gg \frac{\log \sigma}{\varepsilon}$, $\|A^\top u\|_d^d$ (and thus also $P(u)$) is much smaller than 1.

Therefore, if $\{u\}$ obtained in Step 1 is an actual distribution, then it would be essentially supported on the set $\mathcal{A} = \{\pm a^1, \ldots, \pm a^m\}$ of the columns of $A$ and their negations. Let us suppose that $\{u\}$ is simply the uniform distribution over $\mathcal{A}$. (It

can be shown that this essentially is the hardest case to tackle.) In this case the matrix $M$ considered in Step 3 can be written as

$$M = \tfrac{1}{m} \sum_{i=1}^{m} W(a^i)^2 (a^i)(a^i)^\top \ ,$$

where $W(\cdot)$ is the polynomial selected in Step 2. (This uses the fact that this polynomial is a product of linear functions and hence satisfies $W(-a)^2 = W(a)$ for all $a$.) If $W(\cdot)$ satisfies

$$|W(a^1)| \gg \sqrt{m}|W(a^i)| \tag{2.3}$$

for all $i \neq 1$ then $M$ is very close to (a constant times) the matrix $(a^1)(a^1)^\top$, and hence its top eigenvector is close to $a^1$ and we would be done. We want to show that the event (2.3) happens with probability at least inverse polynomial in $m$. Recall that $W$ is a product of $t = c \log n$ random linear functions for some constant $c$ (e.g., $c = 100$ will do). That is, $W(u) = \prod_{i=1}^{t} \langle v^i, u \rangle$, where $v^1, \ldots, v^t$ are standard random Gaussian vectors. Since $\mathbb{E}\langle v^j, a^i \rangle^2 = 1$ and these choices are independent, $\mathbb{E}\, W(a^i)^2 = 1$ for all $i$. However, with probability $\exp(-O(t)) = m^{-O(1)}$ it will hold that $|\langle v^j, a^1 \rangle| \geqslant 2$ for all $j = 1 \ldots t$. In this case $|W(a^1)| \geqslant 2^t$, while we can show that even conditioned on this event, with high probability we will have $|W(a^i)| < 1.9^t \ll |W(a^1)|/\sqrt{m}$ for all $i$, in which case (2.3) holds.[9]

**Part (ii).** The above argument establishes **(i)**, but this is all based on a rather bold piece of wishful thinking— that the object $\{u\}$ we obtained in Step 1 of the algorithm was actually a genuine distribution over unit vectors maximizing $P$. In actuality, we can only obtain the much weaker guarantee that $\{u\}$ is a degree $k$ *pseudo-distribution* for some $k = O(\log n)$. (An actual distribution corresponds to a degree-$\infty$ pseudo-distribution.) The technical novelty of our work lies in establishing **(ii)**. The key observation is that in all our arguments above, we never used any higher moments of $\{u\}$, and that all the inequalities we showed boil down to the simple fact that a square of a polynomial is never negative. (Such proofs are known as *Sum of Squares (SOS) proofs*.)

We will not give the full analysis here, but merely show a representative example of how one "lifts" arguments into the SOS setting. In (2.2) above we used the simple inequality that for every vector $v \in \mathbb{R}^m$

$$\|v\|_d^d \leqslant \|v\|_\infty^{d-2} \|v\|_2^2 \ , \tag{2.4}$$

applying it to the vector $v = A^\top u$ (where we denote $\|v\|_\infty = \max_i |v_i|$). The first (and most major) obstacle in giving a low degree "Sum of Squares" proof for (2.4) is that this is not a polynomial inequality. To turn it into one, we replace the $L_\infty$ norm with the $L_k$ norm for some large $k$ ($k = O(\log m)$ will do). If we replace $\|v\|_\infty$ with $\|v\|_k$ in (2.4), and raise it to the $k/(d-2)$-th power then we obtain the

[9] This argument assumes that no other column is 0.9 correlated with $a^1$. However our actual analysis does not use this assumption, since if two column vectors are closely correlated, we are fine with outputting any linear combination of them.

inequality

$$\left(\|v\|_d^d\right)^{k/(d-2)} \leqslant \|v\|_k^k \left(\|v\|_2^2\right)^{k/(d-2)} , \tag{2.5}$$

which is a valid inequality between polynomials in $v$ whenever $k$ is an integer multiple of $d-2$ (which we can ensure).

We now need to find a sum-of-squares proof for this inequality, namely that the right-hand side of (2.5) is equal to the left-hand side plus a sum of squares, that is, we are to show that for $s = k/(d-2)$,

$$\Big(\sum_i v_i^d\Big)^s \preceq \Big(\sum_i v_i^{(d-2)s}\Big)\Big(\sum_i v_i^2\Big)^s .$$

By expanding the $s$-th powers in this expression, we rewrite this polynomial inequality as

$$\sum_{|\alpha|=s} \binom{s}{\alpha} v^{d\alpha} \preceq \Big(\sum_i v_i^{(d-2)s}\Big) \sum_{|\alpha|=s} \binom{s}{\alpha} v^{2\alpha} = \sum_{|\alpha|=s} \binom{s}{\alpha} v^{2\alpha} \sum_i v_i^{(d-2)s}, \tag{2.6}$$

where the summations involving $\alpha$ are over degree-$s$ multiindices $\alpha \in \{0,\dots,s\}^n$, and $\binom{s}{\alpha}$ denotes the multinomial coefficient $\binom{n}{\alpha} = \frac{s!}{\alpha_1!\dots\alpha_m!}$. We will prove (2.6) term by term, i.e., we will show that $v^{d\alpha} \preceq v^{2\alpha} \sum_i v_i^{(d-2)s}$ for every multiindex $\alpha$. Since $v^{2\alpha} \succeq 0$, it is enough to show that $v^{(d-2)\alpha} \preceq \sum_i v_i^{(d-2)s}$. This is implied by the following general inequality, which we prove in Appendix A:

**Lemma 2.3.** *Let $w_1,\dots,w_n$ be polynomials. Suppose $w_1 \succeq 0,\dots,w_n \succeq 0$. Then, for every multiindex $\alpha$,*

$$w^\alpha \preceq \sum_i w_i^{|\alpha|} .$$

We note that $d$ is even, so $w_i = v_i^{d-2} \succeq 0$ is a square, as required by the lemma.

For the case that $|\alpha|$ is a power of 2, the inequality in the lemma follows by repeatedly applying the inequality $x \cdot y \preceq \frac{1}{2}x^2 + \frac{1}{2}y^2$, which in turn holds because the difference between the two sides equals $\frac{1}{2}(x-y)^2$. As a concrete example, we can derive $w_1^3 w_2 \preceq w_1^4 + w_2^4$ in this way,

$$w_1^3 w_2 = w_1^2 \cdot w_1 w_2 \preceq \tfrac{1}{2} w_1^4 + \tfrac{1}{2} w_1^2 \cdot w_2^2 \preceq \tfrac{1}{2} w_1^4 + \tfrac{1}{2}\Big(\tfrac{1}{2} w_1^4 + \tfrac{1}{2} w_2^4\Big) \preceq w_1^4 + w_2^4 .$$

(The first two steps use the inequality $x \cdot y \preceq \frac{1}{2}x^2 + \frac{1}{2}y^2$. The last step uses that both $w_1$ and $w_2$ are sum of squares.)

Once we have an SOS proof for (2.5) we can conclude that it holds for pseudo-distributions as well, and in particular that for every pseudo-distribution $\{u\}$ of degree at least $k + 2k/(d-2)$ satisfying $\{\|u\|_2^2 = 1\}$,

$$\tilde{\mathbb{E}}\left(\|A^\top u\|_d^d\right)^{k/(d-2)} \leqslant \tilde{\mathbb{E}}\|A^\top u\|_k^k \sigma^{k/(d-2)} .$$

We use similar ideas to port the rest of the proof to the SOS setting, concluding that whenever $\{u\}$ is a pseudo-distribution that satisfies $\{\|u\|_2^2 = 1\}$ and $\{P(u) \geqslant 1 - \tau\}$, then with inverse polynomial probability it will hold that

$$\tilde{\mathbb{E}}\, W(u)^2 \langle u, a\rangle^2 \geqslant (1-\varepsilon)\, \tilde{\mathbb{E}}\, W^2 \tag{2.7}$$

for some column $a$ of $A$ and $\varepsilon > 0$ that can be made arbitrarily close to 0. Once we have (2.7), it is not hard to show that the matrix $M = \tilde{\mathbb{E}}\, W(u)^2 uu^\top$ obtained in Step 3 of our algorithm is close to $aa^\top$. Hence, we can recover a vector close to $\pm a$ by computing the top eigenvector[10] of the matrix $M$.

## 3 Preliminaries

We recall some of the notation mentioned above. We use $P \preceq Q$ to denote that $Q - P$ is a sum of square polynomials. For a vector $v \in \mathbb{R}^d$ and $p \geqslant 1$, we denote $\|v\|_p = (\sum_{i=1}^d |v_i|^p)^{1/p}$ and $\|v\| = \|v\|_2$. For any $\sigma \geqslant 1$, a *$\sigma$-dictionary* is an $n \times m$ matrix $A = (a^1 | \cdots | a^m)$ such that $\|a^i\| = 1$ for all $i$ and the spectral norm of $A^\top A$ is at most $\sigma$ or ,equivalently, $\|Au\|_2^2 \leq \sigma \|u\|_2^2$. Two sets $S_0, S_1 \subseteq \mathbb{R}^n$ are *$\varepsilon$-close in symmetrized Hausdorff distance* if for all $b \in 0, 1$, $\min_{s \in S_b} \max_{t \in S_{1-b}} \mathrm{Cor}(s,t) \geqslant 1 - \varepsilon$, where $\mathrm{Cor}(s,t) = \langle s, t\rangle^2 / (\|s\|\|t\|)^2$; we often drop the qualifier "symmetrized Hausdorff distance" as we will not use another notion of distance between sets of vectors in this paper.

We use the notation of pseudo-expectations and pseudo-distributions from Section 2.1. We now state some basic useful facts about pseudo-distributions, see [BS14, BKS14, BBH+12] for a more comprehensive treatment.

One useful property of pseudo-distributions is that we can find actual distribution that match their first two moments.

**Lemma 3.1** (Matching first two moments)**.** *Let $\{u\}$ be a pseudo-distribution over $\mathbb{R}^n$ of degree at least 2. Then we can efficiently sample from a Gaussian distribution[11] $\{\xi\}$ over $\mathbb{R}^n$ such that for every polynomial $Q$ of degree at most 2,*

$$\mathbb{E}\, Q(\xi) = \tilde{\mathbb{E}}\, Q(u).$$

*Proof.* By shifting, it suffices to restrict attention to the case where $\mathbb{E}\, u_i = 0$ for all $i$. Consider the matrix $M$ such that $M = \tilde{\mathbb{E}}\, uu^\top$. The positivity condition implies that $M$ is a positive semidefinite matrix. Therefore, $M$ admits a Cholesky factorization $M = VV^\top$. Let $\{\zeta\}$ be the standard Gaussian distribution on $\mathbb{R}^n$ (mean 0 and variance 1 in each coordinate) and consider the Gaussian distribution $\{\xi = V\zeta\}$. We are to show that $\xi$ has the same degree-2 moments as the pseudo-distribution $\{u\}$. Indeed,

$$\mathbb{E}\, \xi\xi^\top = \mathbb{E}\, V\zeta\zeta^\top V^\top = VV^\top = M = \tilde{\mathbb{E}}\, uu^\top.$$

[10] In the final algorithm, instead of computing the top eigenvector of the matrix $M$, we will sample from a Gaussian distribution $\{\xi\}$ that satisfies $\mathbb{E}\, \xi\xi^\top = M$. If $M \approx aa^\top$, then such a Gaussian vector $\xi$ is close to $\pm a$ with high probability.

[11] A Gaussian distribution with covariance $\Sigma \in \mathbb{R}^{n \times n}$ and mean $\mu \in \mathbb{R}^n$ has density proportional to $x \mapsto \exp(-\langle x - \mu, \Sigma^{-1}(x - \mu)\rangle/2)$.

Here, we use that $\mathbb{E}\,\zeta\zeta^\top$ is the identity because $\zeta$ is a standard Gaussian vector. □

Another property we will use is that we can *reweigh* a pseudo-distribution by a positive polynomial $W$ to obtain a new pseudo-distribution that corresponds to the operation on actual distributions of reweighing the probability of an element $u$ proportional to $W(u)$.

**Lemma 3.2** (Reweighing)**.** *Let $\{u\}$ be a degree-$k$ pseudo-distribution. Then for every SOS polynomial $W$ of degree $d < k$ with $\tilde{\mathbb{E}}\, W > 0$, there exists a degree-$(k-d)$ pseudo-distribution $\{u'\}$ such that for every polynomial $P$ of degree at most $k-d$*

$$\tilde{\mathop{\mathbb{E}}_{\{u'\}}} P(u') = \frac{1}{\tilde{\mathbb{E}}_{\{u\}} W(u)} \tilde{\mathop{\mathbb{E}}_{\{u\}}} W(u)P(u)\,.$$

*Proof.* The functional $\tilde{\mathbb{E}}_{\{u'\}}$ is linear and satisfies $\tilde{\mathbb{E}}_{\{u'\}}\, 1 = 1$, and so we just need to verify the positivity property. For every polynomial $P$ of degree at most $(k - \deg W)/2$,

$$\tilde{\mathop{\mathbb{E}}_{\{u'\}}} P(u')^2 = (\tilde{\mathop{\mathbb{E}}_{\{u\}}} W(u)P(u)^2)/(\tilde{\mathop{\mathbb{E}}_{\{u\}}} W(u))$$

but since $W$ is a sum of squares, $WP^2$ is also a sum of squares and hence the denominator of the left-hand side is non-negative, while the numerator is by assumption positive. □

# 4 Dictionary Learning

We now state our formal theorem for dictionary learning. The following definition of nice distributions captures formally the conditions needed for recovery. (It is equivalent up to constants to the definition of Section 1.1, see Remark 4.4 below.)

**Definition 4.1** (Nice distribution)**.** Let $\tau \in (0,1)$ and $d \in \mathbb{N}$ with $d$ even. A distribution $\{x\}$ over $\mathbb{R}^m$ is $(d,\tau)$-nice if it satisfies the following properties:

1. $\mathbb{E}\, x_i^d = 1$ for all $i \in [m]$,
2. $\mathbb{E}\, x^\alpha \leqslant \tau$ for all degree-$d$ monomials $x^\alpha \notin \{x_1^d, \dots, x_m^d\}$, and
3. $\mathbb{E}\, x^\alpha = 0$ for all non-square degree-$d$ monomials $x^\alpha$.

Here, $x^\alpha$ denotes the monomial $x_1^{\alpha_1} \cdots x_m^{\alpha_m}$. Furthermore, we require that $x_i^d$ to have polynomial variance so that $\mathbb{E}\, x_i^{2d} = n^{O(1)}$. To avoid some technical issues, $(d,\tau)$-nice distributions are also assumed to be $(d',\tau)$-nice after rescaling for all even $d' \leqslant d$. Concretely, when we say that $\{x\}$ is a $(d,\tau)$-nice distribution, we also imply that for every positive even $d' < d$, there exists a rescaling factor $\lambda$ such that the distribution $\{\lambda \cdot x\}$ satisfies the three properties above (plus polynomial variance bound).

Let us briefly discuss the meaning of these conditions. The condition $\mathbb{E} x_i^{2d} = n^{O(1)}$ is a weak non-degeneracy condition, ruling out distributions where the main contribution to some low order moments comes from events that happen with super-polynomially small probability. Condition 1 stipulates that we are in the symmetric case, where all coefficients have more or less the same magnitude. (We can remove symmetry by either dropping this condition or allowing the dictionary vectors to have different norms; see Remark 6.2.) Condition 2 captures to a certain extent both the sparsity conditions and that that the random variables $x_i$ and $x_i$ for $i \neq j$ are not too correlated. Condition 3 stipulates that there is significant "cancellations" between the negative and positive coefficients. While it is satisfied by many natural distributions, it would be good to either show that it can be dropped, or that it is inherently necessary. The requirement of having expectation zero—perfect cancellation—can be somewhat relaxed to having a sufficiently small bound (inverse polynomial in $n$) on the magnitude of the non-square moments.

We can now state our result for dictionary learning in quasipolynomial time. The result for polynomial time is stated in Section 7.

**Theorem 4.2** (Dictionary learning, quasipolynomial time)**.** *There exists an algorithm that for every desired accuracy $\varepsilon > 0$ and overcompleteness $\sigma \geqslant 1$ solves the following problem for every $(d, \tau)$-nice distribution with $d \geqslant d(\varepsilon, \sigma) = O(\varepsilon^{-1} \log \sigma)$ and $\tau \leqslant \tau(\varepsilon, \sigma) = (\varepsilon^{-1} \log \sigma)^{O(\varepsilon^{-1} \log \sigma)}$ in time $n^{(1/\varepsilon)^{O(1)}(d+\log m)}$: Given $n^{O(d)}/\operatorname{poly}(\tau)$ samples from a distribution $\{y = Ax\}$ for a $\sigma$-overcomplete dictionary $A$ and $(d, \tau)$-nice distribution $\{x\}$, output a set of vectors that is $\varepsilon$-close to the set of columns of $A$ (in symmetrized Hausdorff distance).*

In the *tensor decomposition problem*, we are given a polynomial of the form $\|A^\top u\|_d^d \in \mathbb{R}[u]$ (or equivalently a tensor of the form $\sum_i a_i^{\otimes d}$) and our goal is to recover the vectors $a_1, \ldots, a_m$ (up to signs). It turns out the heart of the dictionary learning problem is solving a variant of the tensor decomposition problem, where we are not given the polynomial $\|A^\top u\|_d^d$ but a polynomial close to it in spectral norm. (The magnitude of this error is related to the niceness of the distribution, which means that we cannot assume it to be arbitrarily small.)

**Theorem 4.3** (Noisy tensor decomposition)**.** *There exists an algorithm that for every desired accuracy $\varepsilon > 0$ and overcompleteness $\sigma \geqslant 1$ solves the following problem for every degree $d \geqslant d(\varepsilon, \sigma) = O(1/\varepsilon) \cdot \log \sigma$ and noise parameter $\tau \leqslant \tau(\varepsilon) = \Omega(\varepsilon)$ in time $n^{(1/\varepsilon)^{O(1)}(d+\log m)}$: Given a degree-$d$ polynomial $P \in \mathbb{R}[u]$ that is $\tau$-close to $\|A^\top u\|_d^d$ in spectral norm for a $\sigma$-overcomplete dictionary $A$, i.e.,*

$$\|A^\top u\|_d^d + \tau \|u\|_2^d \geq P(u) \geq \|A^\top u\|_d^d - \tau \|u\|_2^d \,,$$

*output a set of vectors that is $\varepsilon$-close to the set of columns of $A$ (in symmetrized Hausdorff distance).*

*Remark* 4.4 (Different notions of niceness). In Section 1.1 we defined $(d, \tau)$-niceness in a different way. Instead of requiring $\mathbb{E} x^\alpha \leqslant \tau$ for every monomial $x^\alpha \notin$

$\{x_1^d, \ldots, x_m^d\}$, we only required this condition for some of these monomials, namely monomials of the form $x^\alpha = x_i^{d/2} x_j^{d/2}$. It turns out that these two definitions are equivalent up to a factor $d$ in the exponent of $\tau$. (This loss of a factor of $d$ in the exponent is OK, since in our applications $\tau$ will anyway be exponentially small in $d$.) To see the equivalence of the definitions, note that every degree-$d$ square monomial $x^\alpha \notin \{x_1^d, \ldots, x_m^d\}$ involves at least two distinct variables, say $x_i$ and $x_j$, and therefore $x^\alpha = \mathbb{E}\, x_i^2 x_j^2 x^{\alpha'}$, where $x^{\alpha'}$ is a monomial of degree $d-4$ (so that $\sum_k \alpha'_k = d-4$). By Hölder's Inequality, we can bound its expectation

$$\mathbb{E}\, x_i^2 x_j^2 x^{\alpha'} \leqslant \left(\mathbb{E}\, x_i^{d/2} x_j^{d/2}\right)^{4/d} \left(\mathbb{E}\, x^\beta\right)^{(d-4)/d} ,$$

for $\beta = \frac{d}{d-4}\alpha'$. Since $\sum \beta_k = d$, the Arithmetic-Mean Geometric-Mean Inequality together with our normalization $\mathbb{E}\, x_k^d = 1$ implies

$$\mathbb{E}\, x^\beta \leqslant \sum_k \tfrac{\beta_k}{d} \cdot \mathbb{E}\, x_k^d = 1 ,$$

thus proving that $\mathbb{E}\, x^\alpha \leqslant (\mathbb{E}\, x_i^{d/2} x_j^{d/2})^{4/d}$ for every degree-$d$ square monomial $x^\alpha \notin \{x_1^d, \ldots, x_m^d\}$.

### 4.1 Dictionary learning via noisy tensor decompostion

We will prove Theorem 4.3 (noisy tensor decomposition) in Section 5 and Section 6. At this point, let us see how it yields Theorem 4.2 (dictionary learning, quasipolynomial time). The following lemma gives the connection between tensor decomposition and dictionary learning.

**Lemma 4.5.** *Let $\{x\}$ be a $(d,\tau)$-nice distribution over $\mathbb{R}^m$ and $A$ a $\sigma$-overcomplete dictionary. Then,*[12]

$$\|A^\top u\|_d^d + \tau\sigma^d d^d \|u\|_2^d \geq \mathop{\mathbb{E}}_x \langle Ax, u\rangle^d \geq \|A^\top u\|_d^d .$$

*Proof.* Consider the polynomial $p(v) = \|v\|_d^d + \tau d^d \|v\|_2^d - \mathbb{E}_x \langle x, v\rangle^d$ in the monomial basis for the variables $v_1, \ldots, v_m$. All coefficients corresponding to non-squared monomials are zero (by the third property of nice istibutions). All other coefficients are nonnegative (by the first and second property of nice distributions). We conclude that $p$ is a sum of squares. The relation $\|A^\top u\|_d^d + \tau\sigma d^d \|u\|_2^d \geq \mathbb{E}_x \langle Ax, u\rangle^d$ follows by substituting $v = A^\top u$ and using the relation $\|A^\top u\|_2^d \leq \sigma^d \|u\|_2^d$.

For the lower bound, we see that the polynomial $q(v) = \mathbb{E}_x \langle x, v\rangle^d - \|v\|_d^d$ is a nonnegative combination of square monomials. Thus, $q(v) \geq 0$ and the desired bound follows by substituting $v = A^\top u$. □

[12] The factor $d^d$ can be somewhat reduced, e.g., to $d^{d/2}$. However, this improvement would be hidden by $O(\cdot)$ notation at a later point. For simplicity, we will work with the simple $d^d$ bound at this point.

*Proof of Theorem 4.2.* If we take a sufficiently large number of samples $y_1, \dots, y_N$ from the distribution $\{y = Ax\}$ (e.g., $N \geqslant n^{O(d)}/\tau^2$ will do), then with high probability every coefficient of the polynomial $P = \frac{1}{N}\sum\langle y_i, u\rangle^d \in \mathbb{R}[u]$ would be $\tau/n^d$-close to the corresponding coefficient of $\mathbb{E}\langle y, u\rangle^d$. Therefore, $\pm(P - \mathbb{E}\langle Ax, u\rangle^d) \preceq \tau \cdot \|u\|_2^d$. Together with Lemma 4.5 it follows that

$$\|A^\top u\|_d^d + 2\tau\sigma^d d^d \|u\|_2^d \geq P \geq \|A^\top u\|_d^d - 2\tau\sigma^d d^d \|u\|_2^d \,.$$

Therefore, we can apply the algorithm in Theorem 4.3 (noisy tensor decomposition) for noise parameter $\tau' = 2\tau k^d d^d$ to obtain a set $S$ of unit vectors that is $\varepsilon$-close to the set of columns of $A$ (in symmetrized Hausdorff distance). □

# 5 Sampling pseudo-distributions

In this section we will develop an efficient algorithm that behaves in certain ways like a hypothetical sampling procedure for low-degree pseudo-distributions. (Sampling procedures, even inefficient or approximate ones, cannot exist in general for low-degree pseudo-distributions [Gri01, Sch08].) This algorithm will be a key ingredient of our algorithm for Theorem 4.3 (noisy tensor decomposition, quasipolynomial time).

Here is the property of a sampling procedure that our algorithm mimics: Suppose we have a probability distribution $\{u\}$ over unit vectors in $\mathbb{R}^m$ that satisfies $\mathbb{E}\langle c, u\rangle^k \geqslant e^{-\varepsilon k}$ for some unit vector $c \in \mathbb{R}^m$, small $\varepsilon > 0$, and $k \gg 1/\varepsilon$ (so that $e^{-\varepsilon k}$ is very small). This condition implies that if we sample a vector $u$ from the distribution then with probability at least $e^{-\varepsilon k}/2$ the vector satisfies $\langle c, u\rangle^k \geqslant e^{-\varepsilon k}/2$, which means $\langle c, u\rangle^2 \geqslant e^{-2\varepsilon}/2^{-1/k} \geqslant 1 - O(\varepsilon)$. (Since $e^{-\varepsilon k}$ was very small to begin with, the additional factor 2 for the correlation and the probability is insubstantial.)

The algorithm in the following theorem achieves the above property of sampling procedures with the key advantage that it applies to any low-degree pseudo-distributions.

**Theorem 5.1** (Sampling pseudo-distributions)**.** *For every even $k \geqslant 0$, there exists a randomized algorithm with running time $n^{O(k)}$ and success probability $2^{-k/\operatorname{poly}(\varepsilon)}$ for the following problem: Given a degree-$k$ pseudo distribution $\{u\}$ over $\mathbb{R}^n$ that satisfies the polynomial constraint $\|u\|_2^2 = 1$ and the condition $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-\varepsilon k}$ for some unit vector $c \in \mathbb{R}^n$, output a unit vector $c' \in \mathbb{R}^n$ with $\langle c, c'\rangle \geqslant 1 - O(\varepsilon)$.*

The result follows from the following lemmas.

**Lemma 5.2.** *Let $c \in \mathbb{R}^n$ be a unit vector and let $\{u\}$ be a degree-$(k+2)$ pseudo-distribution over $\mathbb{R}^n$ that satisfies the polynomial constraint $\|u\|_2^2 = 1$. Suppose $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-\varepsilon k}$ for $\varepsilon > 0$. Then, there exists a degree-$k$ sum-of-squares polynomial $W$ such that*

$$\tilde{\mathbb{E}}\, W \cdot \langle c, u\rangle^2 \geqslant (1 - O(\varepsilon))\, \tilde{\mathbb{E}}\, W \,.$$

*Furthermore, there exists a randomized algorithm that runs in time $n^{O(k)}$ and computes such a polynomial $W$ with probability at least $2^{-O(k/\operatorname{poly}(\varepsilon))}$.*

*Proof.* Let us first analyze the random polynomial $w = \langle \xi, u \rangle^2$ for an $n$-dimensional standard Gaussian vector $\xi$. Let $\tau_M$ be such that a standard Gaussian variable $\xi_0$ conditioned on $\xi_0 \geqslant \tau_M$ has expectation $\mathbb{E}_{\xi_0 \geqslant \tau_M} \xi_0^2 = M$. This threshold satisfies $\tau_M \leqslant M$ and thus $\mathbb{P}\{\xi_0 \geqslant \tau_M\} \geqslant 2^{-O(M^2)}$. Conditioned on the event $\langle c, \xi \rangle \geqslant \tau_{M+1}$, the expectation of the random polynomial $w$ satisfies

$$\mathop{\mathbb{E}}_{\{\xi \,|\, \langle c, \xi \rangle \geqslant \tau_{M+1}\}} w = (M+1) \cdot \langle c, u \rangle^2 + \|u\|_2^2 - \langle c, u \rangle^2 = M \cdot \langle c, u \rangle^2 + \|u\|_2^2 .$$

(Here, we use that $\xi = \langle c, \xi \rangle c + \xi'$, where $\xi'$ is a standard Gaussian vector in the subspace orthogonal to $c$ so that $\mathbb{E}\langle \xi', u \rangle^2 = \|u\|_2^2 - \langle c, u \rangle^2$.)

Let $w^{(1)}, \ldots, w^{(k/2)}$ be independent samples from the distribution $\{w \mid \langle c, \xi \rangle \geqslant \tau_{M+1}\}$. Then, let $W = w^{(1)} \cdots w^{(k/2)} / M^{k/2}$. The expectation of this random polynomial satisfies

$$\mathbb{E}\, W = \left( \langle c, u \rangle^2 + \tfrac{1}{M} \cdot \|u\|_2^2 \right)^{k/2} .$$

Let $\overline{W} = (\langle c, u \rangle^2 + 1/M)^{k/2}$. Since the pseudo-distribution $\{u\}$ satisfies the constraint $\|u\|_2^2 = 1$, it also satisfies the constraint $\mathbb{E}\, W = \overline{W}$. We claim that,

$$\overline{W} \cdot \langle c, u \rangle^2 \geq \left(1 - \tfrac{2}{M}\right) \cdot \overline{W} - \left(1 - \tfrac{1}{M}\right)^{k/2} . \tag{5.1}$$

Consider the univariate polynomial

$$p(\alpha) = \alpha^2 \cdot (\alpha^2 + \tfrac{1}{M})^{k/2} + (1 - \tfrac{1}{M})^{k/2} - (1 - \tfrac{2}{M})(\alpha^2 + \tfrac{1}{M})^{k/2} .$$

This polynomial is nonnegative on $\mathbb{R}$, because for $\alpha^2 \geqslant 1 - 2/M$, the first term cancels the last term, and for $\alpha^2 < 1 - 2/M$, the second term cancels the last term. Since $p$ is univariate and nonnegative on $\mathbb{R}$, it follows that $p$ is a sum of squares. Hence, equation (5.1) follows by substituting $\alpha = \langle c, u \rangle$.

The following bound shows that there exists a polynomial $W$ that satisfies the conclusion of the lemma,

$$\begin{aligned} \mathop{\mathbb{E}}_{W} \tilde{\mathbb{E}}\, W \cdot \langle c, u \rangle^2 &\geqslant (1 - \tfrac{2}{M}) \mathop{\mathbb{E}}_{W} \tilde{\mathbb{E}}\, W - e^{-k/2M} \\ &\geqslant (1 - \tfrac{2}{M} - e^{-1/2\varepsilon M}) \mathop{\mathbb{E}}_{W} \tilde{\mathbb{E}}\, W \\ &\geqslant (1 - O(\varepsilon)) \mathop{\mathbb{E}}_{W} \tilde{\mathbb{E}}\, W . \end{aligned} \tag{5.2}$$

The first step uses (5.1) and the bound $(1 - 1/M) \leqslant e^{-1/M}$. The second step uses that $\mathbb{E}_W \tilde{\mathbb{E}}\, W = \tilde{\mathbb{E}}\, \overline{W} \geqslant \tilde{\mathbb{E}} \langle c, u \rangle^k \geqslant e^{-\varepsilon k}$ (premise of the lemma). For the third step, we choose $M = (1/\varepsilon) \cdot \log(1/\varepsilon)$ to trade-off the two error terms $2/M$ and $e^{-1/2\varepsilon M}$.

To show the second part of the lemma, we give a randomized algorithm that runs in time $n^{O(k)}$ and computes a polynomial $W_0$ with the desired properties with probability $2^{-O(k/\operatorname{poly}(\varepsilon))}$. The algorithm samples independent standard Gaussian vectors $\xi^{(1)}, \ldots, \xi^{(k/2)}$ and outputs the polynomial $W_0 = \frac{1}{M^{k/2}} \langle \xi^{(1)}, u \rangle^2 \cdots \langle \xi^{(k/2)}, u \rangle^2$. We are to show that $\tilde{\mathbb{E}}\, W_0 \langle c, u \rangle^2 \geqslant (1 - O(\varepsilon))\, \tilde{\mathbb{E}}\, W_0$ with probability $2^{-O(k/\operatorname{poly}(\varepsilon))}$ over the choice of $W_0$. The distribution $\{W\}$ has density $2^{-O(M^2)}$ in the distribution $\{W_0\}$,

in the sense that there exists an event $\mathcal{E}$ with $\mathbb{P}_{\{W_0\}} \mathcal{E} \geqslant 2^{-O(M^2)}$ and $\{W\} = \{W_0 \mid \mathcal{E}\}$. (The event is $\mathcal{E} = \{\langle \xi^{(1)}, c\rangle, \dots, \langle \xi^{(k/2)}, c\rangle \geqslant \tau_{M+1}\}$).

We will first bound the second moment $\mathbb{E}_W(\tilde{\mathbb{E}}\, W)^2$. The main step is the following bound on the expectation of the random polynomial $w(u)w(u') \in \mathbb{R}[u, u']_4$,

$$\begin{aligned}\mathop{\mathbb{E}}_{\{\xi \mid \langle c,\xi\rangle \geqslant \tau_{M+1}\}} w(u)w(u') &= \mathop{\mathbb{E}}_{\{\xi \mid \langle c,\xi\rangle \geqslant \tau_{M+1}\}} \Big(\langle c,\xi\rangle\langle c,u\rangle + \langle \xi', u\rangle\Big)^2 \Big(\langle c,\xi\rangle\langle c,u'\rangle + \langle \xi', u'\rangle\Big)^2 \\ &\leq 2^{100M^2}\Big(\langle c,u\rangle^2 + \tfrac{1}{M}\|u\|\Big)^2\Big(\langle c,u'\rangle^2 + \tfrac{1}{M}\|u'\|\Big)^2 \qquad (5.3)\end{aligned}$$

In the second step, $\xi'$ is a standard Gaussian vector in the subspace orthogonal to $c$. The third step uses the crude upper bound $\mathbb{E}_{\xi \mid \langle c,\xi\rangle \geqslant \tau_{M+1}} \langle c, \xi\rangle^4 \leqslant 2^{10M^2}$ for $M \geqslant 1$.

The inequality (5.3) implies the second moment bound $\mathbb{E}_W(\tilde{\mathbb{E}}\, W)^2 \leqslant 2^{100kM^2}(\mathbb{E}_W \tilde{\mathbb{E}}\, W)^2$. By Lemma 5.3 and (5.2), it follows that

$$\mathop{\mathbb{P}}_{W}\Big\{\tilde{\mathbb{E}}\, W \cdot (1 - \langle c,u\rangle^2) \leqslant O(\varepsilon)\, \tilde{\mathbb{E}}\, W\Big\} \geqslant \varepsilon^2 \cdot 2^{-100kM^2} = 2^{-O(kM^2)}\,.$$

Since $\{W\}$ has density $2^{-O(M^2)}$ in $\{W_0\}$, it also follows that

$$\mathop{\mathbb{P}}_{W_0}\Big\{\tilde{\mathbb{E}}\, W_0 \cdot \langle c,u\rangle^2 \geqslant (1 - O(\varepsilon))\, \tilde{\mathbb{E}}\, W_0\Big\} \geqslant 2^{-O(kM^2)}\,. \qquad \square$$

**Lemma 5.3.** *Let $\{A, B\}$ be a distribution that satisfies $0 \leqslant A \leqslant B$. Suppose $\mathbb{E}\, A \leqslant \varepsilon\, \mathbb{E}\, B$ and $\mathbb{E}\, B^2 \leqslant t(\mathbb{E}\, B)^2$. Then, $\mathbb{P}\{A \leqslant e^\delta \varepsilon B\} \geqslant \delta^2/9t$ for all $0 \leqslant \delta \leqslant 1$.*

*Proof.* Let $\mathbb{1}_{\text{good}}$ be the 0/1 indicator of the event $\{A \leqslant e^\delta \varepsilon B\}$ and let $p_{\text{good}} = \mathbb{E}\, \mathbb{1}_{\text{good}}$. Let $\mathbb{1}_{\text{bad}} = 1 - \mathbb{1}_{\text{good}}$ be the 0/1 indicator of the complement. The expecation of $\mathbb{1}_{\text{good}} B$ satisfies the lower bound $\tilde{\mathbb{E}}\, \mathbb{1}_{\text{good}} B \geqslant (1 - e^{-\delta})\, \tilde{\mathbb{E}}\, B$ because $\varepsilon\, \mathbb{E}\, B \geqslant \mathbb{E}\, A \geqslant e^\delta \varepsilon\, \mathbb{E}\, \mathbb{1}_{\text{bad}} B$ and thus $\tilde{\mathbb{E}}\, \mathbb{1}_{\text{bad}} B \geqslant e^{-\delta}\, \tilde{\mathbb{E}}\, B$. At the same time, we can upper bound the expectation of $\mathbb{1}_{\text{good}} B$ in terms of $p_{\text{good}}$ using Cauchy–Schwarz and the second moment bound $\mathbb{E}\, B^2 \leqslant t(\mathbb{E}\, B)^2$,

$$\mathbb{E}\, \mathbb{1}_{\text{good}} B \leqslant (\mathbb{E}\, \mathbb{1}_{\text{good}}^2 \cdot \mathbb{E}\, B^2)^{1/2} \leqslant (p_{\text{good}} \cdot t)^{1/2}\, \mathbb{E}\, B.$$

It follows that $p_{\text{good}} \geqslant (1 - e^{-\delta})^2/t \geqslant \delta^2/9t$. $\square$

**Lemma 5.4.** *Let $c \in \mathbb{R}^n$ be a unit vector and let $\{u\}$ be a degree-2 pseudo-distribution over $\mathbb{R}^n$ that satisfies the constraint $\|u\|_2^2 = 1$. Suppose $\tilde{\mathbb{E}}\langle c,u\rangle^2 \geqslant 1 - \varepsilon$ for $\varepsilon > 0$. Then, there exists a distribution $\{v\}$ over unit vectors in $\mathbb{R}^n$ such that $\mathbb{P}\{\langle c,v\rangle^2 \geqslant 1 - 2\varepsilon\} = \Omega(1)$. Moreover, there exists a randomized polynomial-time algorithm to sample from such a distribution $\{v\}$.*

*Proof.* Let $\{\xi\}$ be the Gaussian distribution with the same first two moments as $\{u\}$ (so that $\mathbb{E}\, Q(v) = \tilde{\mathbb{E}}\, Q(u)$ for every degree-2 polynomial $Q$). (See Lemma 3.1.) We choose $v = \xi/\|\xi\|_2$. Since the first two moments of $\{\xi\}$ and $\{u\}$ match, we have $\mathbb{E}(\|\xi\|_2^2 - \langle c,\xi\rangle^2) \leqslant \varepsilon\, \mathbb{E}\|\xi\|_2^2$. Since $\{\xi\}$ is a Gaussian distribution, it satisfies $\mathbb{E}\|\xi\|_2^4 \leqslant O(\mathbb{E}\|\xi\|_2^2)^2$. By Lemma 5.3, it follows that the event $\{\langle c,\xi\rangle^2 \geqslant (1-2\varepsilon)\|\xi\|_2^2\}$ has constant probability. This event is equivalent to the event $\{\langle c,v\rangle^2 \geqslant 1-2\varepsilon\}$. $\square$

## 6 Noisy tensor decomposition

In this section we will prove Theorem 4.3 (noisy tensor decomposition, quasi-polynomial time).

**Theorem** (Restatement of Theorem 4.3)**.** *There exists an algorithm that for every desired accuracy $\varepsilon > 0$ and overcompleteness $\sigma \geqslant 1$ solves the following problem for every degree $d \geqslant d(\varepsilon, \sigma) = O(1/\varepsilon) \cdot \log \sigma$ and noise parameter $\tau \leqslant \tau(\varepsilon) = \Omega(\varepsilon)$ in time $n^{(1/\varepsilon)^{O(1)}(d+\log m)}$: Given a degree-$d$ polynomial $P \in \mathbb{R}[u]$ that is $\tau$-close to $\|A^\top u\|_d^d$ in spectral norm for a $\sigma$-overcomplete dictionary $A$, i.e.,*

$$\|A^\top u\|_d^d + \tau\|u\|_2^d \geq P(u) \geq \|A^\top u\|_d^d - \tau\|u\|_2^d \,,$$

*output a set of vectors that is $\varepsilon$-close to the set of columns of $A$ (in symmetrized Hausdorff distance).*

The proof combines the following lemma with Theorem 5.1 (sampling pseudo-distributions). The lemma formalizes the following fact in terms of low-degree pseudo-distributions: the polynomial $\|A^\top u\|_d^d \in \mathbb{R}[u]$ assumes large values over the sphere only at points close to one of the columns of $A$. Note that the conclusion of the lemma allows us to reconstruct a column of $A$ in time $n^{O(k)}$ using Theorem 5.1 (sampling pseudo-distributions).

**Lemma 6.1.** *Let $A$ be a $\sigma$-overcomplete dictionary and let $\{u\}$ be a degree-$3k$ pseudo-distribution over $\mathbb{R}^n$ that satisfies the polynomial constraints $\{\|A^\top u\|_d^d \geqslant e^{-\delta d}, \|u\|_2^2 = 1\}$. Then, there exists a column $c$ of $A$ such that $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-\varepsilon k}$ for $\varepsilon = O(\delta + \frac{\log \sigma}{d} + \frac{\log m}{k})$.*

*Proof.* First, we claim that the pseudo-distribution $\{u\}$ also satisfies the constraint $\{\|A^\top u\|_k^k \geqslant e^{-\delta' k}\}$ where $\delta' = \frac{d}{d-2}\delta + \frac{\log \sigma}{d-2}$. The proof of this claim follows by a sum-of-squares version of the following form of Hölder's inequality,

$$\|v\|_d \leqslant \|v\|_k^{1-2/d} \cdot \|v\|_2^{2/d} \,.$$

(This inequality holds for all norms $\|\cdot\|_k$ with $k \geqslant 1$, including $\|\cdot\|_\infty$.) In particular, if $k$ is an integer multiple of $d-2$, the following relation of degree $k + 2k/(d-2)$ holds among polynomials in $\mathbb{R}[v]$,

$$(\|v\|_d^d)^{k/(d-2)} \leq (\|v\|_2^2)^{k/(d-2)} \cdot \|v\|_k^k \,.$$

See the overview section for a proof of this fact. By substituting $v = A^\top u$ and using the facts that $\|A^\top u\|_2^2 \leq \sigma\|u\|^2$ and that $\{u\}$ satisfies the constraint $\{\|u\|^2 = 1\}$, we get that $\{u\}$ satisfies $\{\|A^\top u\|_k^k \geqslant (\|A^\top u\|_d^d)^{k/(d-2)}/\sigma^{k/(d-2)}\}$, which implies the claim because $\{\|A^\top u\|_d^d \geqslant e^{-\delta d}\}$.

By an averaging argument, there exists some column $c$ of $A$ that satisfies $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant \tilde{\mathbb{E}}\|A^\top u\|_k^k/m \geqslant e^{-\delta' k}/m = e^{-\varepsilon k}$ for $\varepsilon = \delta' + \frac{\log m}{k}$ as desired. □

**Proof of Theorem 4.3 from Lemma 6.1 and Theorem 5.1.** Our tensor decomposition algorithms constructs a set of unit vectors $S \subseteq \mathbb{R}^m$ in an iterative way. We will determine the choice of the parameters $k \geqslant 1$ and $\gamma > 0$ later.

- Start with $S = \emptyset$.
- While there exists a degree-$k$ pseudo-distribution $\{u\}$ that satisfies the constraints $\{P(u) \geqslant 1 - \tau, \|u\|_2^2 = 1\}$ and $\{\langle s, u\rangle^2 \leqslant 1 - \gamma\}$ for every $s \in S$:
  - Use the algorithm in Theorem 5.1 (sampling pseudo-distributions) to obtain in time $n^{k/\operatorname{poly}(\varepsilon)}$ a unit vector $c' \in \mathbb{R}^m$ that satisfies $P(c') \geqslant e^{-\varepsilon d} - \tau$ for $\varepsilon = O(\frac{\tau}{d} + \frac{\log\sigma}{d}\frac{\log m}{k})$ (by Lemma 6.1) and $\langle c', s\rangle^2 \leqslant 1 - \gamma/10$ for every vector $s \in S$.
  - Add the vector $c'$ to the set $S$.

Let us first explain why we can find a vector $c'$ that satisfies the above conditions if there exists such a pseudo-distribution $\{u\}$. Recall that the input polynomial $P$ satisfies $\pm(P - \|A^\top u\|_d^d) \preceq \tau\|u\|_2^d$. Therefore, the above pseudo-distributions satisfy $\{\|A^\top u\|_d^d \geqslant 1 - 2\tau = e^{-\delta d}\}$ for $\delta = O(\tau/d)$. Hence, Lemma 6.1 implies that a column $c$ of $A$ satisfies $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-\varepsilon' k}$ for $\varepsilon' = O((\frac{\tau}{d} + \frac{\log\sigma}{d}\frac{\log m}{k})$. Thus, the algorithm of Theorem 5.1 will output a unit vector $c'$ with $\langle c, c'\rangle^k \geqslant e^{-O(\varepsilon')k} = e^{-\varepsilon k}$ with probability at least $2^{-k/\operatorname{poly}(\varepsilon)}$. Note that the condition $\langle c, c'\rangle^k \geqslant e^{-\varepsilon k}$ implies that $P(c') \geqslant e^{-\varepsilon d} - \tau$. By repeating the algorithm $2^{k/\operatorname{poly}(\varepsilon)}$ times we can ensure that with high probability one of the vectors found in this way satisfies the desired condition. We claim that the condition $\langle c, c'\rangle^2 \geqslant 1 - O(\varepsilon)$ implies that $\langle c', s\rangle \leqslant 1 - \gamma/10$ for all $s \in S$ (assuming a suitable choice of $\gamma$). Let $\gamma' = \frac{1}{2}\|(c')^{\otimes 2} - s^{\otimes 2}\|^2$. We are to show $\gamma' \geqslant \gamma/10$. By the triangle inequality, $\|c^{\otimes 2} - s^{\otimes 2}\|^2 \leqslant O(\varepsilon) + 4\gamma'$. Together with an SOS version of the triangle inequalitity, $\|s^{\otimes 2} - u^{\otimes 2}\|^2 \preceq 8\gamma' + O(\varepsilon) + 2\|c^{\otimes 2} - u^{\otimes 2}\|^2$. Since $\{u\}$ satisfies $\{\langle s, u\rangle^2 \leqslant 1 - \gamma\}$ it follows that $\{2\gamma \leqslant 8\gamma' + O(\varepsilon) + 2\|c^{\otimes 2} - u^{\otimes 2}\|^2\}$, which implies the constraint $\{\langle c, u\rangle^2 \leqslant 1 - \gamma/2 + 2\gamma' + O(\varepsilon)\}$ (using the constraint $\{\|u\|^2 = 1\}$). However, since $c$ satisfies $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-\varepsilon k}$, we have $\gamma/2 - 2\gamma' - O(\varepsilon) \leqslant O(\varepsilon)$, which means that $\gamma' \geqslant \gamma/4 - O(\varepsilon) \geqslant \gamma/10$ as desired. (Here, we are assuming that $\gamma$ was chosen so that $\gamma/\varepsilon$ is a large enough constant.)

Next we claim that every vector in $s \in S$ is close to one of the columns of $A$. Indeed, every such vector satisfies $\|A^\top s\|_d^d \geqslant e^{-\varepsilon d} - 2\tau$, which by Lemma 6.1 implies that $\langle s, c\rangle^2 \geqslant 1 - O(\varepsilon + \tau/d + (\log\sigma)/d) = 1 - O(\varepsilon)$ for a column $c$ of $A$.

Next we claim that if the algorithm terminates then for every column $c$ of $A$ there exists a vector $s \in S$ with $\langle c, s\rangle^2 \geqslant 1 - \gamma$. Indeed, if there exists a column that violates this condition, then it would satisfy all constraints for the pseudo-distribution, which means that the algorithm does not terminate at this point.

To finish the proof of the theorem it remains to bound the number of iterations of the algorithm. We claim that the number of iterations is bounded by the number $m$ of columns of $A$ because in each iteration the vectors in $S$ will cover at least one more of the columns of $A$. As observed before, every vector $s \in S$ is close to a column $c_s$ of $A$ in the sense that $\|s^{\otimes 2} - c_s^{\otimes 2}\|^2 = O(\varepsilon)$. However, since $c'$

satisfies $\langle c', s\rangle^2 \leqslant 1 - \gamma/10$, we have by triangle inequality $\gamma/5 \leqslant \|(c')^{\otimes 2} - s^{\otimes 2}\|^2 \leqslant 2\|(c')^{\otimes 2} - c_s^{\otimes 2}\|^2 + 2\|s^{\otimes 2} - c_s^{\otimes 2}\|^2$, which means that $\|(c')^{\otimes 2} - c_s^{\otimes 2}\|^2 \geqslant \gamma/10 - O(\varepsilon)$. Therefore, the vector $c'$ is not close to any of the vectors $c_s$ for $s \in S$, which means that it has to be close to another column of $A$. (Here, we are again assuming that $\gamma$ was chosen so that $\gamma/\varepsilon$ is a large enough constant.) □

*Remark* 6.2 (Handling columns with varying norms). Many of our techniques also apply to dictionaries with columns of different $\ell_2$ norms. In particular, using the same algorithm, we can reconstruct in this case a single vector close to one of the columns. More generally, we can reconstruct a set of vectors that is close to the set of columns with maximum norm.

By adapting the algorithm somewhat we can also achieve recovery guarantees for columns with significantly smaller norm than the maximum norm. Concretely, we can modify the algorithm so that we ask for pseudo-distributions satisfying $P(u) \geqslant \rho$, where $\rho$ is a parameter that we gradually decrease so we can get all the vectors. However, we need to also change the right-hand side of the constraint $\langle u, s\rangle^2 \leqslant 1 - \gamma$ to a value that decreases with $\rho$. Otherwise, the algorithm might not terminate, as there can be exponentially vectors that are somewhat far from a column vector $c$, and all of them will have fairly large value for $P(\cdot)$. Such a modified algorithm can still obtain all the column vectors (up to a small error) if we assume that the they are sufficiently *incoherent*. That is, $\langle a, a'\rangle \leqslant \mu$ for every distinct columns $a$, $a'$ of $A$ with $\mu$ depending on the norm ratios. Similar (and in fact often stronger) assumptions were made in prior works on dictionary learning. (However, we need these assumptions only when the vectors have different norms.)

# 7 Polynomial-time algorithms

In this section we show how we can improve our tensor decomposition algorithm when we have access to examples of very sparse linear combinations of the dictionary columns, culminating in Theorem 7.6 that gives a polynomial-time algorithm for the dictionary problem for the case the distribution is $(d, \tau)$-nice for $\tau = n^{-\Omega(1)}$.

## 7.1 Sampling pseudo-distributions

The following theorem refines Theorem 5.1 (sampling pseudo-distributions) reconstructing a vector $c'$ that is close to a target vector $c$. We make an additional assumption about having access to samples from a distribution $\{W\}$ over sum-of-squares polynomials. This distribution comes with a noise parameter $\tau$ that controls how well the distribution correlated with the target vector $c$. If this noise parameter is sufficiently small, samples from distribution allow the algorithm to work under a more refined but milder condition on the pseudo-distribution $\{u\}$. For our dictionary learning algorithm, we can satisfy this condition when the

noise parameter $\tau$ of the distribution $\{W\}$ satisfies $\tau \ll m^{1/k}$. (The noise parameter $\tau$ roughly coincides with the niceness parameter of the distribution $\{x\}$.)

**Theorem 7.1** (refined sampling from pseudo-distributions)**.** *For every $k \geqslant 1$, there exists a $n^{O(k)}$-time algorithm with the following guarantees: Suppose the input of the algorithm is a pseudo-distribution $\{u\}$ over $\mathbb{R}^n$ and a sum-of-squares polynomial $W \in \mathbb{R}[u]$ satisfying the following properties for some unit vector $c \in \mathbb{R}^n$:*

- *The sum-of-squares polynomial $W$ is chosen from a distribution $\{W\}$ with with mean $\bar{W} = \mathbb{E}_W W$ and second moment $\mathbb{E}_W W(u)W(u') \leq M \cdot \bar{W}(u) \cdot \bar{W}(u')$ satisfying*
$$\langle c, u\rangle^{2(1+k)} \leq \bar{W} \leq (\langle c, u\rangle^2 + \tau\|u\|_2^2)^{1+k}\,. \tag{7.1}$$
- *The pseudo-distribution $\{u\}$ has degree $2(1+2k)$ and satisfies the polynomial constraint $\|u\|_2^2 = 1$ and the conditions*
$$\tilde{\mathbb{E}}\langle c, u\rangle^{2(1+2k)} \geqslant e^{-\varepsilon k}\,\tilde{\mathbb{E}}\langle c, u\rangle^2 \text{ and } \tilde{\mathbb{E}}\langle c, u\rangle^2 \geqslant \tau^k. \tag{7.2}$$

*Then, the output of the algorithm is a unit vector $c' \in \mathbb{R}^n$ such that with probability at least $\tau^2/M2^{O(k)/\operatorname{poly}(\varepsilon)}$,*
$$\langle c, c'\rangle^2 \geqslant e^{-O(\varepsilon+3(1/k+1)\tau)}\,.$$

The following lemma is the main new ingredient of the proof of this theorem.

**Lemma 7.2.** *Let $\{u\}$ be a degree-$2(1+2k)$ pseudodistribution that satisfies the constraint $\|u\|_2^2 = 1$. Let $\{W\}$ be a distribution over sum-of-squares polynomials. Suppose $\{u\}$ and $\{W\}$ satisfy the conditions in Theorem 7.1. Then, $\tilde{\mathbb{E}}_u W \cdot \langle c, u\rangle^{2k} \geqslant e^{-\varepsilon' k}\,\tilde{\mathbb{E}}_u W$ with probability $\tau^2/M2^{O(k)}$ over the choice of $W$ for $\varepsilon' = \varepsilon + 3(1/k+1)\tau$.*

Note that the conclusion of the lemma implies that we can recover a vector $c'$ with $\langle c', c\rangle^2 \geqslant 1 - O(\varepsilon')$ using Theorem 5.1 in time $n^{k/poly(\varepsilon')}$ with probability $2^{O(k)/\operatorname{poly}(\varepsilon')}$. Therefore, Theorem 7.1 follows by combining Lemma 7.2 with Theorem 5.1.

*Proof of Lemma 7.2.* We will show that the polynomials $\bar{W}\langle c, u\rangle^k$ and $\bar{W}$ have similar pseudo-expectations by comparing them to the polynomials $\langle c, u\rangle^2$. We will show that $\tilde{\mathbb{E}}\,\bar{W}\langle c, u\rangle^k$ For brevity, choose polynomials $\alpha = \langle c, u\rangle^2 \in \mathbb{R}[u]$ and $\beta = \|u\|^2 \in \mathbb{R}[u]$ so that $0 \leq \alpha \leq \beta$. Then,

$$\begin{aligned}\alpha^{1+k} \leq \bar{W} \leq (\alpha + \tau\beta)^{1+k} &= \alpha\sum_{i=0}^{k}\binom{1+k}{i}\alpha^{k-i}(\tau\beta)^i + (\tau\beta)^{k+1} \\ &\leq \alpha\beta^k\sum_{i=0}^{k}(1+k)^i\tau^i + \tau^{k+1}\beta^{k+1} \leq \Big(1 + 2(t+k)\tau\Big)\alpha\beta^k + \tau^{k+1}\beta^{k+1}\,. \end{aligned}\tag{7.3}$$

Here, the last step uses the assumption $(1+k)\tau \leqslant 1/2$ to bound the series $\sum_{i=1}^{k}(1+k)^i\tau^i \leqslant 2(t+k)\tau$. It follows that

$$\tilde{\mathbb{E}}\,\bar{W}\langle c,u\rangle^k \geqslant \tilde{\mathbb{E}}\,\alpha^{1+2k} \geqslant e^{-\varepsilon k}\,\tilde{\mathbb{E}}\,\alpha\,.$$

(Here, we used (7.2)) At the same time,

$$\tilde{\mathbb{E}}\,\bar{W} \leqslant \Big(1+2(1+k)\tau\Big)\,\tilde{\mathbb{E}}\,\alpha + \tau^{k+1} \leqslant \Big(1+2(1+k+1)\tau\Big)\,\tilde{\mathbb{E}}\,\alpha \leqslant e^{2(2+k)\tau}\,\tilde{\mathbb{E}}\,\alpha\,.$$

Here, the second step uses the assumption $\tau^{k+1} \leqslant \tau\,\tilde{\mathbb{E}}\langle c,u\rangle^2$. Together, the two bounds imply $\tilde{\mathbb{E}}\,\bar{W}\langle c,u\rangle^{2k} \geqslant e^{-\varepsilon k-2(2+k)\tau}\,\tilde{\mathbb{E}}\,\bar{W}$.

In order to lower bound the probability of the event $\{\tilde{\mathbb{E}}\,W\langle c,u\rangle^{2k} \geqslant e^{-\varepsilon' k}\,\tilde{\mathbb{E}}\,W\}$, we will upper bound the second moment $\mathbb{E}(\tilde{\mathbb{E}}\,W)^2$ and apply Lemma 5.3. By the premise $\mathbb{E}\,W(u)W(u') \preceq M\cdot\bar{W}(u)\bar{W}(u')$, we get

$$\mathbb{E}\Big(\tilde{\mathbb{E}}\,W\Big)^2 = \tilde{\mathbb{E}}_{\{u\}\{u'\}}\,\mathbb{E}\,W(u)W(u') \preceq \tilde{\mathbb{E}}_{\{u\}\{u'\}}\,M\cdot\bar{W}(u)\bar{W}(u') = M\cdot\Big(\tilde{\mathbb{E}}\,\bar{W}\Big)^2\,.$$

By Lemma 5.3, the probabilility of the event $\{\tilde{\mathbb{E}}\,W\langle c,u\rangle^{2k} \geqslant (e^{-\varepsilon k-2(2+k)\tau}-\delta)\,\tilde{\mathbb{E}}\,W\}$ is at least $\Omega(\delta^2/M)$. We choose $\delta = \tau 2^{-O(k)}$ to lower bound the probability of the event $\{\tilde{\mathbb{E}}\,W\langle c,u\rangle^{2k} \geqslant e^{-\varepsilon' k}\,\tilde{\mathbb{E}}\,W\}$ for $\varepsilon' = \varepsilon + 3(1/k+1)\tau$ by $\Omega(\tau^2/M2^{O(k)})$. □

## 7.2 Tensor decomposition

The following lemma shows that a pseudo-distribution $\{u\}$ that satisfies the constraints $\{\|A^\top u\|^{2(t+k)}_{2(t+k)} \approx 1, \|u\|_2^2 = 1\}$ also satisfies the condition of Theorem 7.1 for one of the columns of the dictionary $A$.

**Lemma 7.3.** *Let $A \in \mathbb{R}^{n\times m}$ be a $\sigma$-overcomplete dictionary and let $\{u\}$ be a degree-$2(k+t)$ pseudo-distribution over $\mathbb{R}^n$ that satisfies $\|u\|_2^2 = 1$. Suppose $\{u\}$ also satisfies the polynomial constraint $\|A^\top u\|^{2(1+k)}_{2(1+k)} \geqslant e^{-2(k-1)\varepsilon}\sigma$ for $k \geqslant 1$. Then, there exists a column $c$ of $A$ such that $\tilde{\mathbb{E}}\langle c,u\rangle^{2k} \geqslant e^{-2\varepsilon k}\,\tilde{\mathbb{E}}\langle c,u\rangle^2$ and $\tilde{\mathbb{E}}\langle c,u\rangle^2 \geqslant \varepsilon e^{-2k\varepsilon}/m$.*

*Remark.* For the lower bound on $\tilde{\mathbb{E}}\langle c,u\rangle^2$, we typically only need that it is polynomial. The algorithm in Theorem 7.1 allows us to recover a vector close to $c$ in time $n^t$ assuming that $\tau^k \ll 1/m$.

*Proof.* We will prove the contrapositive. Let $a_1,\dots,a_m \in \mathbb{R}^n$ be the columns of $A$ and let $\varepsilon' = \varepsilon e^{-2k\varepsilon}$. Suppose every column $c$ satisfies either $\tilde{\mathbb{E}}\langle c,u\rangle^{2(1+k)} < e^{-2\varepsilon k}\,\tilde{\mathbb{E}}\langle c,u\rangle^2$ or $\tilde{\mathbb{E}}\langle c,u\rangle^2 < \varepsilon'/m$. We are to show that the pseudo-distribution $\{u\}$ cannot satisfy the constraint $\|A^\top u\|^{2(1+k)}_{2(1+k)} \geqslant e^{2(k-1)\varepsilon}\sigma$. Indeed, these conditions allow us to upper bound

$$\tilde{\mathbb{E}}\|A^\top u\|^{2(1+k)}_{2(1+k)} = \sum_{i=1}^{m}\tilde{\mathbb{E}}\langle a_i,u\rangle^{2(1+k)} \leqslant e^{-2\varepsilon k}\,\tilde{\mathbb{E}}\|A^\top u\|_2^2 + \varepsilon' \leqslant (1+\varepsilon)e^{-2k\varepsilon}\sigma.$$

It follows that the pseudo-distribution $\{u\}$ cannot satisfy the constraint $\|A^\top u\|^{2(1+k)}_{2(1+k)} \geqslant e^{2(k-1)\varepsilon}\sigma$.

□

### 7.3 Dictionary learning

The following lemma shows that up to polynomial reweighing the distribution $\{y = Ax\}$ gives us access to a distribution $\{W\}$ that satisfies the condition of Theorem 7.1.

**Lemma 7.4.** *Let $A \in \mathbb{R}^{n\times m}$ be a $\sigma$-overcomplete dictionary and and let $\{x\}$ be a $(k,\tau)$-nice distribution over $\mathbb{R}^m$ with $k \geqslant 4$. For $i \in [m]$, let $\mathcal{D}_i$ be the distribution obtained from reweighing the distribution $\{w = c\langle Ax, u\rangle^2\}$ by $x_i^2$, where $c = \mathbb{E}\, x_i^2 / \mathbb{E}\, x_i^4$. Then, $\langle a^{(i)}, u\rangle^2 \leq \mathbb{E}_{\mathcal{D}_i} w \leq \langle a^{(i)}, u\rangle^2 + \tau\sigma\|u\|_2^2$.*

*Proof.* The expectation of $w$ after reweighing by $x_i^2$ satisfies

$$\mathop{\mathbb{E}}_{\mathcal{D}_i} w = \frac{1}{\mathbb{E}_{\{x\}} x_i^2} \mathop{\mathbb{E}}_{\{x\}} x_i^2 \cdot c\langle Ax, u\rangle^2 = \sum_j \frac{1}{\mathbb{E}_{\{x\}} x_i^4} \mathop{\mathbb{E}}_{\{x\}} x_i^2 x_j^2 \langle a^{(i)}, u\rangle^2 \langle a^{(j)}, u\rangle^2$$

The last step uses that all non-square moments of $\{x\}$ vanish. The desired bounds follow because the coefficient of $\langle a^{(i)}, u\rangle^2$ is 1 and for all indices $j \neq i$, the coefficients of $\langle a^{(j)}, u\rangle^2$ are all between 0 and $\tau$. For the final bounds, we also use $\|A^\top u\|_2^2 \leq \sigma\|u\|^2$. □

**Theorem 7.5** (Dictionary learning, polynomial time, single dictionary vector)**.** *There exists an algorithm that solves the following problem for every desired accuracy $\varepsilon > 0$, overcompletess $\sigma \geqslant 2$, in time $n^{O(k)}$ with success probability $n^{-\tilde{O}(k)/\operatorname{poly}(\varepsilon)}$ for noise $\tau \leqslant O(\varepsilon)$, where $k = (1/\varepsilon)\log\sigma + \frac{\log m}{\log(1/\tau)}$: Given $k$ samples from a distribution of the form $\{y = Ax\}$ and a degree-$k$ pseudo-distribution $\{u\}$ that satisfies $\{\|Au\|_k^k \geqslant e^{-\varepsilon k}, \|u\|_2^2 = 1\}$, where $A$ is a $\sigma$-overcomplete dicionary and $\{x\}$ is a $(4,\tau)$-nice distribution, output a unit vector $c'$ such that there exists a column $c$ of $A$ with $\langle c, c'\rangle^2 \geqslant 1 - O(\varepsilon)$ and $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-O(\varepsilon)k}\,\tilde{\mathbb{E}}\langle c, u\rangle^2$.*

*Proof.* We run the algorithm in Theorem 7.1 on the pseudo-distribution $\{u\}$ and the following distribution $\{W\}$ over squared polynomials: Choose $k' = k/2 - 1$ independent samples $y_1, \dots, y_{k'}$ from the distribution $\{y = Ax\}$ and form the degree-$(k-2)$ polynomial $W = \langle y_1, u\rangle^2 \cdots \langle y_{k'}, u\rangle^2$. This distribution $\{W\}$ does not satisfy the condition in Theorem 7.1 but it turns out to be sufficiently close to a distribution that satisfies the condition. Let us first verify that the pseudo-distribution $\{u\}$ satisfies the condition of Theorem 7.1 for a vector $c$ as in the theorem above. Indeed, by Lemma 7.3, there exists a column $c$ of $A$ such that $\tilde{\mathbb{E}}\langle c, u\rangle^k \geqslant e^{-O(\varepsilon)k}\,\tilde{\mathbb{E}}\langle c, u\rangle^2$ and $\tilde{\mathbb{E}}\langle c, u\rangle^2 \geqslant O(\varepsilon)e^{-O(\varepsilon k)}/m \geqslant \tau^k$. (Since $k \geqslant (1/\varepsilon)\log\sigma$, the pseudo-distribution $\{u\}$ satisfies the constraint $\{\|A^\top u\|_k^k \geqslant e^{-O(\varepsilon)k}\sigma\}$ as required by Lemma 7.3.) It follows that if we run the algorithm in Theorem 7.1 for a distribution over polynomials that satisfies condition (7.1) for this column $c$ of the dictionary $A$, then the algorithm outputs a vector $c'$ with the above properties with significant probability.

We will use Lemma 7.4 to reason about the distribution $\{W\}$. Without loss of generality, we assume that $c$ is the first column of the dictionary $A$. Let

$\bar{x} = (x_1, \dots, x_{k'})$ be $k'$ independent samples from $\{x\}$. (The distribution $\{W\}$ is the same as $\{\langle Ax_1, u\rangle^2 \cdots \langle Ax_{k'}, u\rangle^2\}$.) We claim that the distribution $\{W\}$ satisfies (7.1) after reweighing by the function $r(\bar{x})^2 = x_{1,1}^2 \cdots x_{k',1}^2$ (the product of the square of the first coordinates of $x_1, \dots, x_{k'}$). The distribution after reweighing is, up to scaling of the polynomials, equal to the distribution $\mathcal{D} = \{W = w_1 \cdots w_{k'}\}$, where $w_1, \dots, w_{k'}$ are independent samples from the distribution $\mathcal{D}_1$ in Lemma 7.4. By Lemma 7.4, this reweighted distribution satisfies the condition (7.1), that is,

$$\langle c, u\rangle^{2k'} \preceq \mathop{\mathbb{E}}_{\mathcal{D}} W \preceq (\langle c, u\rangle^2 + \tau\sigma\|u\|_2^2)^{k'} .$$

Since we assume $\{x\}$ to be $(4, \tau)$-nice, the variance of $\mathcal{D}$ is bounded by $n^{O(k)}$.

Let $\mathcal{A}$ be the algorithm in Theorem 7.1. Since $\mathcal{D}$ satisfies the conditions of Theorem 7.1, if we run $\mathcal{A}$ on the pseudo-distribution $\{u\}$ and the distribution $\mathcal{D}$ over polynomials, it will succeed with probability $n^{-O(k)/\operatorname{poly}(\varepsilon)}$. We claim that the success probability on the distribution $\{W\}$ (before reweighing) is comparable. Let $p(W)$ be the probability that the algorithm succeeds for a particular input polynomial $W$. Under the distribution $\mathcal{D}$, algorithm $\mathcal{A}$ has success probability $\mathbb{E}_{\mathcal{D}} p(W) \geqslant n^{-O(k)/\operatorname{poly}(\varepsilon)}$. We relate this success probability to the success probability under $\{W\}$ as follows,

$$n^{-O(k)/\operatorname{poly}(\varepsilon)} \leqslant \mathop{\mathbb{E}}_{\mathcal{D}} p(W) = \frac{1}{\mathbb{E}_{\{\bar{x}\}} r(x)^2} \mathop{\mathbb{E}}_{\{\bar{x}\}} r(\bar{x})^2 p(W) \leqslant \frac{1}{\mathbb{E}_{\{\bar{x}\}} r(x)^2} \left( \mathop{\mathbb{E}}_{\{\bar{x}\}} r(\bar{x})^4 p(W) \cdot \mathop{\mathbb{E}}_{\{W\}} p(W) \right)^{1/2} ,$$

where the last step uses Cauchy–Schwarz. The niceness property of $\{x\}$ implies that $\mathbb{E}_{\bar{x}} r(\bar{x})^4 p(W) \leqslant \mathbb{E}_{\bar{x}} r(\bar{x})^4 = (\mathbb{E}_{\{x\}} x_1^4)^{k'} = n^{O(k)} \cdot (\mathbb{E}_{\{x\}} x_1^2)^{2k'} = n^{O(k)} (\mathbb{E}_{\{\bar{x}\}} r(\bar{x})^2)^2$. Therefore, the success probability of $\mathcal{A}$ under the distribution $\{W\}$ (before reweighing) satisfies $\mathbb{E}_{\{W\}} p(W) \geqslant n^{-O(k)/\operatorname{poly}(\varepsilon)}$.

□

The following theorem gives a polynomial time algorithm for dicionary learning under $(d, \tau)$-nice distributions for all $\tau = n^{\Omega(1)}$.

**Theorem 7.6** (Dictionary learning, polynomial time)**.** *There exists an algorithm that for every desired accuracy $\varepsilon > 0$ and overcompleteness $\sigma \geqslant 1$ solves the following problem for every $(d, \tau)$-nice distribution with $d \geqslant d(\varepsilon, \sigma) = O(d^{-1} \log \sigma)$ and $\tau \leqslant \tau(\varepsilon, \sigma) = (\varepsilon^{-1} \log \sigma)^{O(\varepsilon^{-1} \log \sigma)}$ in time $n^{(1/\varepsilon)^{O(1)} k}$ for $k = d + O(\frac{\log m}{\log(1/\tau)})$: Given $n^{O(d)}/\operatorname{poly}(\tau)$ samples from a distribution $\{y = Ax\}$ for a $\sigma$-overcomplete dictionary $A$ and $(d, \tau)$-nice distribution $\{x\}$, output a set of vectors that is $\varepsilon$-close to the set of columns of $A$ (in symmetrized Hausdorff distance).*

*Proof.* We will show how to use Theorem 7.5 to recover a single vector that is close to one of the columns of $A$. By repeating this step in the same way as in the proof of Theorem 4.3 (noisy tensor decomposition) we can recover a set of vectors that is close to the set of columns of $A$.

To recover a single vector, we estimate from the samples of $\{y = Ax\}$ a polynomial $P$ that is close to $\|A^\top u\|_d^d$ in the same way as in the proof of Theorem 4.2.

(The distance of $P$ from $\|A^\top u\|_d^d$ in spectral norm will be $O(\tau d^d) = O(\varepsilon)$.) Next, we compute a degree-$k$ pseudo-distribution $\{u\}$ that satisfies the constraints $\{P \geqslant 1 - \varepsilon, \|u\|_2^2 = 1\}$.[13] The same argument as in the proof of Lemma 6.1 shows that $\{u\}$ also satisfies the constraint $\{\|Au\|_k^k \geqslant e^{O(\varepsilon)k}\}$, which means that $\{u\}$ satisfies the premise of Theorem 7.5. Therefore, the algorithm in Theorem 7.5 recovers a vector close to one of the columns of $A$.

□

## 8 Conclusions and Open Problems

The *Sum of Squares* method has found many uses across a variety of disciplines, and in this work we demonstrate its potential for solving unsupervised learning problems in regimes that have so far eluded other algorithms. It is an interesting direction to identify other problems that can be solved using this algorithm.

The generality of the SOS method comes at a steep cost of efficiency. It is a fascinating open problem, and one we are quite optimistic about, to use the ideas from the SOS-based algorithm to design practically efficient algorithms.

## References


[AAJ+13] Alekh Agarwal, Animashree Anandkumar, Prateek Jain, Praneeth Netrapalli, and Rashish Tandon, *Learning sparsely used overcomplete dictionaries via alternating minimization*, CoRR **abs/1310.7991** (2013). 3, 8

[AAN13] Alekh Agarwal, Animashree Anandkumar, and Praneeth Netrapalli, *Exact recovery of sparsely used overcomplete dictionaries*, CoRR **abs/1309.1952** (2013). 3, 8

[ABGM14] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma, *More algorithms for provable dictionary learning*, CoRR **abs/1401.0579** (2014). 3, 8

[AFH+12] Anima Anandkumar, Dean P. Foster, Daniel Hsu, Sham Kakade, and Yi-Kai Liu, *A spectral algorithm for latent dirichlet allocation*, NIPS, 2012, pp. 926–934. 8

[AGH+13] Sanjeev Arora, Rong Ge, Yonatan Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu, *A practical algorithm for topic modeling with provable guarantees*, ICML (2), 2013, pp. 280–288. 4


[13] To recover all vectors, we would also add constraints $\{\langle s, u\rangle^2 \leqslant 1 - \gamma\}$ for all vectors $s$ that have already been recovered (see proof of Theorem 4.3).

[AGM12] Sanjeev Arora, Rong Ge, and Ankur Moitra, *Learning topic models - going beyond svd*, FOCS, IEEE Computer Society, 2012, pp. 1–10. 8

[AGM13] ———, *New algorithms for learning incoherent and overcomplete dictionaries*, arXiv preprint 1308.6723 (2013), http://arxiv.org/abs/1308.6273. 3, 5, 8

[Bar98] Franck Barthe, *On a reverse form of the brascamp-lieb inequality*, Inventiones mathematicae **134** (1998), no. 2, 335–361. 4

[BBH+12] Boaz Barak, Fernando G. S. L. Brandão, Aram Wettroth Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou, *Hypercontractivity, sum-of-squares proofs, and their applications*, STOC, 2012, pp. 307–326. 14

[BCMV14] Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan, *Smoothed analysis of tensor decompositions*, STOC, 2014. 8

[BCV14] Aditya Bhaskara, Moses Charikar, and Aravindan Vijayaraghavan, *Uniqueness of tensor decompositions with applications to polynomial identifiability*, COLT (Maria-Florina Balcan and Csaba Szepesvári, eds.), JMLR Proceedings, vol. 35, JMLR.org, 2014, pp. 742–778. 8

[BKS14] B. Barak, J.A. Kelner, and D. Steurer, *Rounding sum of squares relaxations*, STOC, 2014. 8, 11, 14

[BS14] Boaz Barak and David Steurer, *Sum-of-squares proofs and the quest toward optimal algorithms*, Proceedings of International Congress of Mathematicians (ICM), 2014, To appear. 9, 14

[Com94] Pierre Comon, *Independent component analysis, a new concept?*, Signal processing **36** (1994), no. 3, 287–314. 7, 8

[CRT06] Emmanuel J Candes, Justin K Romberg, and Terence Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on pure and applied mathematics **59** (2006), no. 8, 1207–1223. 3

[DH13] L. Demanet and P. Hand, *Recovering the Sparsest Element in a Subspace*, October 2013, Arxiv preprint 1310.1654. 8

[Don06] David L Donoho, *Compressed sensing*, Information Theory, IEEE Transactions on **52** (2006), no. 4, 1289–1306. 3

[DPS02] Andrew C Doherty, Pablo A Parrilo, and Federico M Spedalieri, *Distinguishing separable and entangled states*, Physical Review Letters **88** (2002), no. 18, 187904. 3

[EA06] Michael Elad and Michal Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, Image Processing, IEEE Transactions on **15** (2006), no. 12, 3736–3745. 3

[EP07] Andreas Argyriou Theodoros Evgeniou and Massimiliano Pontil, *Multi-task feature learning*, Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, vol. 19, MIT Press, 2007, pp. 41–48. 3

[FH14] Péter E Frenkel and Péter Horváth, *Minkowski's inequality and sums of squares*, Central European Journal of Mathematics **12** (2014), no. 3, 510–516. 33

[FJK96] Alan Frieze, Mark Jerrum, and Ravi Kannan, *Learning linear transformations*, 37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996), IEEE Comput. Soc. Press, Los Alamitos, CA, 1996, pp. 359–368. MR 1450634 8

[For01] Jürgen Forster, *A linear lower bound on the unbounded error probabilistic communication complexity*, IEEE Conference on Computational Complexity, IEEE Computer Society, 2001, pp. 100–106. 4

[GP04] Karin Gatermann and Pablo A Parrilo, *Symmetry groups, semidefinite programs, and sums of squares*, Journal of Pure and Applied Algebra **192** (2004), no. 1, 95–128. 4

[Gri01] Dima Grigoriev, *Linear lower bound on degrees of positivstellensatz calculus proofs for the parity*, Theor. Comput. Sci. **259** (2001), no. 1-2, 613–622. 18

[GVX14] Navin Goyal, Santosh Vempala, and Ying Xiao, *Fourier pca*, STOC, 2014, Also available as arXiv report 1306.5825. 8

[Har70] Richard A Harshman, *Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis*. 8

[Har07] John Harrison, *Verifying nonlinear real formulas via sums of squares*, Theorem Proving in Higher Order Logics, Springer, 2007, pp. 102–118. 3

[HG05] Didier Henrion and Andrea Garulli, *Positive polynomials in control*, vol. 312, Springer, 2005. 3

[Hur91] A. Hurwitz, *Ueber den vergleich des arithmetischen und des geometrischen mittels.*, Journal für die reine und angewandte Mathematik **108** (1891), 266–268, Available online at `http://eudml.org/doc/148823`. 33

[Kru77] Joseph B Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear algebra and its applications **18** (1977), no. 2, 95–138. 8

[Las01] Jean B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization **11** (2001), no. 3, 796–817. 3, 8, 9, 10

[LCC07] Lieven De Lathauwer, Joséphine Castaing, and Jean-François Cardoso, *Fourth-order cumulant-based blind identification of underdetermined mixtures*, IEEE Transactions on Signal Processing **55** (2007), no. 6-2, 2965–2973. 8

[LRS+10] Jason Lee, Ben Recht, Nathan Srebro, Joel Tropp, and Ruslan Salakhutdinov, *Practical large-scale optimization for max-norm regularization*, Advances in Neural Information Processing Systems, 2010, pp. 1297–1305. 4

[MLB+08] Julien Mairal, Marius Leordeanu, Francis Bach, Martial Hebert, and Jean Ponce, *Discriminative sparse image models for class-specific edge detection and image interpretation*, Computer Vision–ECCV 2008, Springer, 2008, pp. 43–56. 3

[MRBL07] Y Marc'Aurelio Ranzato, Lan Boureau, and Yann LeCun, *Sparse feature learning for deep belief networks*, Advances in neural information processing systems **20** (2007), 1185–1192. 3

[Nes00] Y. Nesterov, *Squared functional systems and optimization problems*, High performance optimization **13** (2000), 405–440. 3, 8, 9, 10

[NR09] Phong Q. Nguyen and Oded Regev, *Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures*, J. Cryptology **22** (2009), no. 2, 139–160, Preliminary version in EUROCRYPT 2006. 8

[OF96a] Bruno A Olshausen and David J Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, Nature **381** (1996), no. 6583, 607–609. 3, 7

[OF96b] ———, *Natural image statistics and efficient coding**, Network: computation in neural systems **7** (1996), no. 2, 333–339. 3, 7

[OF97] Bruno A. Olshausen and David J. Field, *Sparse coding with an overcomplete basis set: A strategy employed by v1?*, Vision Research **37** (1997), no. 23, 3311 – 3325. 3, 7

[Par00] Pablo A Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D. thesis, California Institute of Technology, 2000. 3, 8, 9, 10

[Par06] ———, *Polynomial games and sum of squares optimization*, Decision and Control, 2006 45th IEEE Conference on, IEEE, 2006, pp. 2855–2860. 3

[Rez87] Bruce Reznick, *A quantitative version of Hurwitz' theorem on the arithmetic-geometric inequality*, J. reine angew. Math **377** (1987), no. 108-112. 33

[Rez89] _____, *Forms derived from the arithmetic-geometric inequality*, Mathematische Annalen **283** (1989), no. 3, 431–464. 33

[Sch08] Grant Schoenebeck, *Linear level Lasserre lower bounds for certain k-CSPs*, FOCS, 2008, pp. 593–602. 18

[She09] Jonah Sherman, *Breaking the multicommodity flow barrier for o(vlog n)-approximations to sparsest cut*, FOCS, 2009, pp. 363–372. 4

[Sho87] NZ Shor, *An approach to obtaining global extremums in polynomial mathematical programming problems*, Cybernetics and Systems Analysis **23** (1987), no. 5, 695–700. 3, 8, 9, 10

[SWW12] Daniel A. Spielman, Huan Wang, and John Wright, *Exact recovery of sparsely-used dictionaries*, Journal of Machine Learning Research - Proceedings Track **23** (2012), 37.1–37.18. 3, 8

[Tuc66] Ledyard R Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika **31** (1966), no. 3, 279–311. 8

[YWHM08] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma, *Image super-resolution as sparse representation of raw image patches*, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8. 3

# A Proof of Lemma 2.3

Lemma 2.3 is a consequence of the following sum-of-squares version of the AM-GM inequality.[14]

**Lemma A.1.** *Let $w_1, \dots, w_n$ be polynomials. Suppose $w_1, \dots, w_n \geq 0$. Then,*

$$\frac{w_1^n + \cdots + w_n^n}{n} \geq w_1 w_2 \cdots w_n \,.$$

To see that this lemma implies Lemma 2.3, write for a multi-index $\alpha$ with $|\alpha| = s$ the polynomial $w^\alpha$ as a product $w^\alpha = \prod_{j=1}^s w_{i_j}$, where $w_i$ is repeated $\alpha_i$ times. (E.g., we would write $w_1^2 w_2 w_3^2$ as $w_1 w_1 w_2 w_3 w_3$ and we would have $(i_1, \dots, i_5) = (1, 1, 2, 3, 3)$.) Then applying Lemma A.1 to the polynomials $w_{i_1}, \dots, w_{i_s}$ gives the inequality asserted in Lemma 2.3,

$$w^\alpha = w_{i_1} \cdots w_{i_s} \leq \frac{w_{i_1}^s + \cdots + w_{i_s}^s}{s} = \sum_i \frac{\alpha_i}{|\alpha|} w_i \leq \sum_i w_i^s,$$

where the second inequality uses that $0 \leqslant \alpha_i/|\alpha| \leqslant 1$ and the premise $w_i \geq 0$.

**Proof of Lemma A.1.** To prove Lemma A.1, we will give a sequence of polynomials $R_0, \dots, R_{n-1}$ such that $R_0 = (z_1^n + \dots z_n^n)/n$, $R_{n-1} = z_1 \dots z_n$, and $R_0 \geq \dots \geq R_{n-1}$. To this end, let

$$R_k = \frac{1}{n!} \sum_{\sigma \in S_n} w_{\sigma_1}^{n-k} \prod_{j=2}^{k+1} w_{\sigma_j},$$

where $S_n$ denotes the symmetric group on $n$ elements. So, for instance,

$$R_0 = \frac{1}{n!} \sum_{\sigma \in S_n} w_{\sigma_1}^n = \frac{1}{n} \left( w_1^n + \cdots + w_n^n \right),$$

$$R_1 = \frac{1}{n!} \sum_{\sigma \in S_n} w_{\sigma_1}^{n-1} w_{\sigma_2} = \frac{1}{n(n-1)} \left( w_1^{n-1} w_2 + w_1^{n-1} w_3 + w_1^{n-1} w_4 + \cdots + w_n^{n-1} w_{n-1} \right),$$

$$R_2 = \frac{1}{n!} \sum_{\sigma \in S_n} w_{\sigma_1}^{n-2} w_{\sigma_2} w_{\sigma_3} = \frac{1}{n \binom{n-1}{2}} \left( w_1^{n-2} w_2 w_3 + w_1^{n-2} w_2 w_4 + \cdots + w_n^{n-2} w_{n-2} w_{n-1} \right), \text{ and}$$

$$R_{n-1} = \frac{1}{n!} \sum_{\sigma \in S_n} w_{\sigma_1} w_{\sigma_2} \cdots w_{\sigma_n} = w_1 w_2 \cdots w_n.$$

The following claim will then complete the proof:

*Claim* A.2. For any $k \in \{1, \dots, n-1\}$, $R_{k-1} - R_k$ is a sum of squares.

[14] The first sum-of-squares proof of the AM-GM inequality dates back to Hurwitz in 1891 [Hur91]. For related results and sums-of-squares proofs of more general sets of inequalities, see [Rez87, Rez89, FH14].

*Proof.* For a given permutation $\sigma \in S_n$, the corresponding monomials in $R_k$ and $R_{k-1}$ will share many of the same variables, differing only in the exponents of $w_{\sigma_1}$ and $w_{\sigma_{k+1}}$. We will thus try to arrange the terms of $R_{k-1} - R_k$ so that we can pull out the common variables, which will let us reduce our inequality to one involving only two variables.

$$\begin{aligned}
R_{k-1} - R_k &= \frac{1}{n!} \sum_{\sigma \in S_n} \left( \left( w_{\sigma_1}^{n-k+1} \prod_{j=2}^{k} w_{\sigma_j} \right) - \left( w_{\sigma_1}^{n-k} \prod_{j=2}^{k+1} w_{\sigma_j} \right) \right) \\
&= \frac{1}{n!} \sum_{\sigma \in S_n} w_{\sigma_1}^{n-k} \left( w_{\sigma_1} - w_{\sigma_{k+1}} \right) \left( \prod_{j=2}^{k} w_{\sigma_j} \right) \\
&= \frac{1}{n!} \sum_{\substack{a,b \in [n] \\ a \neq b}} \sum_{\substack{\sigma \in S_n \\ \sigma_1 = a \\ \sigma_{k+1} = b}} w_{\sigma_1}^{n-k} \left( w_{\sigma_1} - w_{\sigma_{k+1}} \right) \left( \prod_{j=2}^{k} w_{\sigma_j} \right) \\
&= \frac{1}{n!} \sum_{\substack{a,b \in [n] \\ a \neq b}} w_a^{n-k} \left( w_a - w_b \right) \sum_{\substack{\sigma \in S_n \\ \sigma_1 = a \\ \sigma_{k+1} = b}} \prod_{j=2}^{k} w_{\sigma_j} \\
&= \frac{1}{n!} \sum_{\substack{a,b \in [n] \\ a < b}} \left( w_a^{n-k} - w_b^{n-k} \right) \left( w_a - w_b \right) \cdot \left\{ \sum_{\substack{\sigma \in S_n \\ \sigma_1 = a \\ \sigma_{k+1} = b}} \prod_{j=2}^{k} w_{\sigma_j} \right\}.
\end{aligned}$$

Since the $w_i$ are sums of squares, the expression inside the braces is as well. It is therefore enough to show that $\left( w_a^{n-k} - w_b^{n-k} \right) \left( w_a - w_b \right)$ is a sum of squares. This follows from the fact that

$$w_a^{n-k} - w_b^{n-k} = \left( w_a - w_b \right) \left( w_a^{n-k-1} + w_a^{n-k-2} w_b + \cdots + w_b^{n-k-2} w_a + w_b^{n-k-1} \right),$$

and thus

$$\left( w_a^{n-k} - w_b^{n-k} \right) \left( w_a - w_b \right) = \left( w_a - w_b \right)^2 \left( w_a^{n-k-1} + w_a^{n-k-2} w_b + \cdots + w_b^{n-k-2} w_a + w_b^{n-k-1} \right).$$

□