Low-Rank Matrix Completion with Adversarial Missing Entries

Tselil Schramm∗ Ben Weitz†

July 13, 2015

Abstract

We give an algorithm for the adversarial matrix completion problem, in which we wish to recover a low rank matrix when adversarially chosen entries are hidden. We show that, so long as the number of missing entries in any row or column is bounded by a function of the dimension, rank, and incoherence of the matrix, nuclear norm minimization recovers the target matrix exactly. The range for which this guarantee holds is surprisingly large—in an $n \times n$ matrix of constant rank, there may be as many as $\Omega(n)$ entries missing in every row and column. Conversely, if only constantly many entries are missing in any row or column, then we may recover matrices of rank $\Omega(n)$.

We also use adversarial matrix completion to give an algorithm for completing an order-$m$ symmetric low-rank tensor from its multilinear entries in time roughly proportional to the number of tensor entries. We apply our tensor completion algorithm to the problem of learning mixtures of product distributions over the hypercube, obtaining new algorithmic results. If the centers of the product distribution are linearly independent, then we recover distributions with as many as $\Omega(n)$ centers in polynomial time and sample complexity. In the general case, we recover distributions with as many as $\tilde{\Omega}(n)$ centers in quasi-polynomial time, answering an open problem of Feldman et al. (SIAM J. Comp.) for the special case of distributions with incoherent bias vectors.

∗UC Berkeley, tschramm@cs.berkeley.edu. Supported by an NSF Graduate Research Fellowship (NSF award no 1106400).
†UC Berkeley, bsweitz@cs.berkeley.edu. Supported by an NSF Graduate Research Fellowship (NSF award no DGE 1106400).
1 Introduction

Low-rank matrix completion is a computational primitive that has garnered a lot of attention in recent years due to its expressive nature and wide applicability. Matrix completion has found use from the straightforward applications of recommender systems (such as the famous Netflix Problem) and survey completion to the more nuanced settings of recovering 3D data from a set of 2D images [KBWT13, TK92]. A rich body of work has grown around algorithms for low-rank matrix completion in the recent decade, and two principal approaches have emerged.

The first approach employs semidefinite programming to minimize a convex relaxation of the rank, known as the nuclear norm. This approach is studied in [CR09, CT09, Rec09, RFP10], where the authors show that minimizing the nuclear norm recovers the hidden matrix exactly, even when the number of hidden entries is almost at the information-theoretic limit. The second approach is Alternating Minimization: in this iterative method, the hidden matrix is found by alternatingly optimizing one of the two factors in a low-rank approximation to the singular value decomposition. Since at each iterate one factor is fixed, each step is simply a least-squares minimization problem and can be solved efficiently. Alternating Minimization was first studied as a heuristic in [BK07, KBV09, HH09], and later progress on its theoretical guarantees can be found in [JNS13, GAGG13, Har14, HW14].

In all of the above works there is a common assumption that the subset of unknown entries is chosen uniformly at random. In many settings this is a reasonable assumption, but often it is likely that the missing entries are non-random or correlated: specific survey questions are more likely to be left blank, data loss or corruption occurs in a localized manner, and some movies few people are willing to admit they enjoyed. In this work, we present an algorithm for low-rank matrix completion in the “adversarial” regime, where the choice of missing entries is arbitrary (albeit somewhat restricted).

There is comparatively little work in this “adversarial” regime of matrix completion, where entries are hidden because of the specific constraints of the problem. In its full generality, finding any low-rank completion from adversarially revealed entries is NP-hard [Pee96, HMRW14]. Further, there are information-theoretic limits: if an entire row or column is missing, there is no hope of recovering the matrix exactly. Still, it is of interest to solve the case in which a specific number of entries must be revealed in each row and column. Also, the NP-hardness constructions have a very small fraction of entries revealed relative to the incoherence (see Definition 2.4 for the definition of incoherence), and so it is possible to efficiently perform completion in a more constrained setting.

Prior adversarial matrix completion algorithms focus exclusively on completing a matrix when only the diagonal entries are missing. In [SCPW12], the authors use SDP trace minimization to solve the related problem of, given a PSD matrix $X$, decomposing it into the sum of a diagonal matrix $D$ and a low-rank PSD matrix $L$. Their result can be applied to adversarial matrix completion in the case where the matrix is PSD and only the diagonal is missing. Finally, the authors of [JO13] show that Alternating Minimization succeeds when the missing entries are in a block-diagonal pattern.

The results we present in this paper use Nuclear Norm Minimization to perform matrix completion in a more general setting, where any entry of the matrix can be missing, but there are not too many hidden entries in any single row or column. To our knowledge, ours is the first work guaranteeing the completion of a matrix with adversarially hidden off-diagonal entries. The number of missing entries allowed in any row or column is an explicit function of the matrix dimension, rank, and incoherence—when the rank and incoherence are low, every row and column can contain up to $\Omega(n)$ missing entries. When restricted to the missing diagonal setting, our algorithm matches the
parameter range of [SCPW12] up to a factor of $\frac{1}{2}$ and greatly improves upon the ratio of rank to dimension for which [JO13] succeeds.

Similar to [JO13], our main motivation for studying adversarial matrix completion is its application to learning product mixtures over the Boolean hypercube. Suppose we are given access to a distribution over the hypercube $\{\pm 1\}^n$ generated in the following manner: there are $k$ product distributions over $\{\pm 1\}^n$ (the $k$ “centers” of the distribution), and each sample is generated by choosing one center with some probability and then generating a sample from this center. This distribution is called a product mixture over the hypercube, and the goal is to recover from samples the parameters of the individual product distributions. This problem has been studied extensively and approached with a variety of strategies (see e.g. [FM99, CR08, FOS08]).

A canonical approach to problems of this type is to empirically estimate the moments of the distribution, from which it may be possible to calculate the distribution parameters using linear-algebraic tools (see e.g. [FOS08, AGM12, HK13, AGH+14]). For product distributions over the hypercube, this technique does not directly apply because the square moments are always 1, and so they provide no information. This is where matrix completion comes to our aid. The origin of the matrix completion approach is in [FOS08], in which the authors employ a brute-force approach to complete the matrices in question, yielding a runtime exponential in the number of centers (corresponding to the matrix rank). More recently, [JO13] used alternating minimization to complete the (block)-diagonal of the second moments matrix to tackle this problem. These authors were able to obtain a significant improvement, obtaining a polynomial time algorithm for linearly independent mixtures over at most $O(n^{2/7})$ centers. In this paper, we improve upon this result, and can handle as many as $\Omega(n)$ centers in some parameter settings. Because our algorithm is able to complete matrices with missing off-diagonal entries, we are also able to obtain a quasipolynomial algorithm for the general case, in which the centers may be linearly dependent. This resolves an open problem of [FOS08], when restricted to distributions whose bias vectors satisfy a condition on their rank and incoherence (see Section 2.3.1 for a discussion of incoherence assumptions on product mixtures).

To learn the product mixtures, we give a matrix-completion-based algorithm for completing tensors of order $m$ from their multilinear moments in time $\tilde{O}(n^{m+1})$, which we believe may be of independent interest. There has been ample work in the area of noisy tensor decomposition (and completion), see e.g. [? , ? , ? , ?]. However, these works usually assume that the tensor is obscured by random noise, while in our setting the “noise” is the absence of all non-multilinear entries. An exception to this is the work of [?], where to obtain a quasi-polynomial algorithm it suffices to have the injective tensor norm of the noise be bounded via a Sum-of-Squares proof.\(^1\) To our knowledge, our algorithm is the only $n^{O(m)}$-time algorithm that solves the problem of completing a symmetric tensor when only multilinear entries are known.

1.1 Our Results

In this paper we show that Nuclear Norm Minimization applied to the matrix completion problem will exactly recover the hidden matrix even when the entries are revealed adversarially, provided that the fraction of missing entries per row and column is not too large compared to the incoherence of the matrix (see Definition 2.4 for the definition of incoherence). More formally, we prove the following theorem:

\(^1\)It may be possible that this condition is met for some symmetric tensors when only multilinear entries are known, but we do not know an SOS proof of this fact.
Theorem 1.1. Let $M$ be an $m \times n$ rank-$r$ matrix which is $(\mu_U, \mu_V)$-incoherent, and let $\overline{\Omega} \subset [m] \times [n]$ be the set of hidden indices. If there are at most $\kappa$ elements per column and $\rho$ elements per row of $\overline{\Omega}$, and if $2(\kappa \sqrt{\mu U} + \rho \sqrt{\mu V})r < 1$, then $M$ is the unique solution to the Nuclear Norm Minimization semidefinite program.

We note that there exist matrix completion instances with $\kappa/n = 1 - o(1)$, $\mu = 1$ and $r = 3$ for which finding any completion is NP-hard [HMRW14, Pee96] (via a reduction from three-coloring), so the constant on the right-hand side is necessarily at most six. We also note that the tradeoff between $\kappa/n$ and $\mu$ in Theorem 1.1 is necessary because for a matrix of fixed rank, one can add extra rows and columns of zeros in an attempt to reduce $\kappa/n$, but this process increases $\mu$ by an identical factor.

As an application of the above result, we show how to learn product mixtures with up to even $\Omega(n)$ centers in polynomial (or quasi-polynomial) time:

Theorem 1.2. Let $\mathcal{D}$ be a mixture over $k$ product distributions on $\{\pm 1\}^n$, with bias vectors $v_1, \ldots, v_k \in \mathbb{R}^n$ and mixing weights $w_1, \ldots, w_k > 0$. Let $\text{span}\{v_i\}$ have dimension $r$ and incoherence $\mu$. Suppose we are given as input the moments of $\mathcal{D}$.

1. If $v_1, \ldots, v_k$ are linearly independent, then as long as $4 \cdot \mu \cdot r < n$, there is a $\text{poly}(n,k)$ algorithm that recovers the parameters of $\mathcal{D}$.

2. Otherwise, if $|\langle v_i, v_j \rangle| < \|v_i\| \cdot \|v_j\| \cdot (1 - \eta)$ for every $i \neq j$ and $\eta > 0$, then as long as $4 \cdot \mu \cdot r \cdot \log k / \log \frac{1}{1-\eta} < n$, there is an $n^{O(\log k / \log \frac{1}{1-\eta})}$ time algorithm that recovers the parameters of $\mathcal{D}$.

To our knowledge, Theorem 1.2 is the first quasi-polynomial algorithm that learns product mixtures whose centers are not linearly independent. [FOS08] give an exponential-time algorithm based on a restricted brute-force search, and [JO13] give a polynomial-time algorithm for learning product mixtures, but cannot recover linearly dependent bias vectors because of their use of the tensor power iteration algorithm of [AGH+14]. While we also use the tensor power iteration approach of [AGH+14], our tensor completion algorithm works for tensors of arbitrary order, allowing us to reduce the dependent case to the independent one by working in a high enough tensor power space.

To illustrate the additional power of our result, we note that a choice of random $v_1, \ldots, v_k$ in an $r$-dimensional subspace meet this condition extremely well, as we have $\eta = 1 - \tilde{O}(1/\sqrt{r})$ with high probability—for, say, $k = 2r$, the algorithm of [JO13] would fail in this case, since $v_1, \ldots, v_k$ are not linearly independent, but our algorithm succeeds in time $n^{O(1)}$.

In the main body of the paper we assume access to exact moments; in Appendix B we prove Theorem B.2, a version of Theorem 1.2 which accounts for sampling error.

1.2 Organization

The remainder of our paper is organized as follows. In Section 2, we give definitions and background, then outline our approach to solving the adversarial matrix completion problem and our approach to learning product mixtures over the hypercube. In Section 3, we prove Theorem 1.1. In Section 4, we give an algorithm for completing symmetric tensors given access only to their multilinear entries, using adversarial matrix completion as an algorithmic primitive. In Section 5, we apply our tensor completion result to learn mixtures of product distributions over the hypercube, assuming access to the precise second- and third-order moments of the distribution. Appendix A and Appendix B
contain discussions of matrix completion and learning product mixtures in the presence of sampling error, and Appendix C contains further details about the algorithmic primitives used in learning product mixtures.

1.3 Notation

In the matrix completion sections, we mimic the notation in [CR09] as this work closely follows the proof strategy therein. Calligraphic letters such as $\mathcal{A}$ denote operators on matrices, upper case letters such as $A$ denote matrices, and lower case letters such as $a$ denote vectors or scalars. We use $e_i$ to denote the $i$th standard basis vector.

For a tensor $T \in \mathbb{R}^{n \times n \times n}$, we use $T(a, b, c)$ to denote the entry of the tensor indexed by $a, b, c \in [n]$, and we use $T(i, \cdot, \cdot)$ to denote the $i$th slice of the tensor, or the subset of entries in which the first coordinate is fixed to $i \in [n]$. For an order-$m$ tensor $T \in \mathbb{R}^{n \times \cdots \times n}$, we use $T(X)$ to represent the entry indexed by the string $X \in [n]^m$, and we use $T(Y, \cdot, \cdot)$ to denote the slice of $T$ indexed by the string $Y \in [n]^{m-2}$. For a vector $v \in \mathbb{R}^n$, we use the shorthand $x \otimes k$ to denote the $k$-tensor $x \otimes x \cdot \cdot \cdot \otimes x \in \mathbb{R}^{n \times \cdots \times n}$.

For a matrix $A$, $\|A\|_p$ denotes the $p$-norm of $A$ as a vector, i.e. $\|A\|_p := (\sum_{ij} A_{ij}^p)^{1/p}$, and $\|A\|$ denotes the spectral norm of $A$, i.e. the top singular value of $A$. The vector $\sigma(A)$ denotes the vector whose entries are the singular values of $A$, and $\sigma_i(A)$ denotes the $i$th-largest singular value of $A$. For an operator $\mathcal{A}$, $\|\mathcal{A}\|_p$ denotes the induced $p$-norm of $\mathcal{A}$: $\|\mathcal{A}\|_p := \max_{M \text{ s.t. } \|M\|_p = 1} \|\mathcal{A}(M)\|_p$.

The Nuclear Norm of a matrix $A$ is defined $\|A\|_* := \|\sigma(A)\|_1$.

We use $\Omega \subseteq [m] \times [n]$ for the set of observed entries of the hidden matrix $M$, and $\mathcal{P}_\Omega$ denotes the projection onto those coordinates.

2 Preliminaries

In this section we present background necessary to prove our results, as well as provide an outline for the proofs of our main results. We begin by defining the matrix completion problem:

Problem 2.1 (Matrix Completion). Given a set $\Omega \subseteq [m] \times [n]$ of observed entries of a hidden rank-$r$ matrix $M$, the Matrix Completion Problem is to successfully recover the matrix $M$ given only $\mathcal{P}_\Omega(M)$.

2.1 Nuclear Norm Minimization

Nuclear Norm Minimization is an optimization technique that is by now standard for matrix completion, and is represented by the convex program

$$\begin{align*}
\text{minimize} & \quad \|X\|_* \\
\text{subject to} & \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M). \quad (2.1)
\end{align*}$$

This optimization problem can be seen as a convex relaxation of true rank-minimization, indeed the nuclear norm unit ball is exactly the convex hull of properly normalized rank-1 matrices, so nuclear norm minimization is the tightest convex relaxation of rank minimization. Unlike rank-minimization, the program (2.1) can be solved efficiently, and it can be seen (e.g. in [VB96]) that it is equivalent to

$$\begin{align*}
\text{minimize} & \quad \text{trace}(W_1 + W_2) \\
\text{subject to} & \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M) \\
& \quad \begin{bmatrix} W_1 & X \\ X^T & W_2 \end{bmatrix} \succeq 0
\end{align*}$$

4
which is a simple semidefinite program and can be solved in time \(O(n^3)\). In this work we show that under reasonable conditions, the solution \(X^*\) to (2.1) is unique and in fact equal to \(M\), even if \(\Omega\) is not random. Our analysis follows [CR09] closely and proceeds by constructing a dual certificate for \(M\).

### 2.1.1 Duality and the Subdifferential of the Nuclear Norm

Convex duality theory asserts that a matrix \(X\) is a solution to (2.1) if there exists a dual certificate \(Y\) such that \(\mathcal{P}_\Omega(Y) = Y\) and \(Y\) is a subgradient for the nuclear norm at \(X\), denoted \(Y \in \partial \|X\|_n\). If \(X = \sum_{i=1}^r \sigma_i u_i v_i^T\) is the singular value decomposition of \(X\), then \(\partial \|X\|_n\) can be described as follows (see [Wat92] or [CR09]): Let \(U = \text{span}\{u_1, \ldots, u_r\}\) and \(V = \text{span}\{v_1, \ldots, v_r\}\), and let

\[
T := \text{span}\{xv_i^T, uy_i^T : x \in \mathbb{R}^m, y \in \mathbb{R}^n\}. \tag{2.2}
\]

Furthermore, let \(P_T, P_V\), and \(P_T\) denote the projections onto the subspaces \(U, V\), and \(T\) respectively, and thus \(\mathcal{P}_T(X) = P_T(X) + XP_V - P_U(X)P_V\). Then \(Y \in \partial \|X\|_n\) if and only if \(\mathcal{P}_T(Y) = \sum_{i=1}^r u_i v_i^T\), and \(\|\mathcal{P}_T(Y)\| < 1\). Candes and Recht prove the following lemma in [CR09] using straightforward properties of the subgradient:

**Lemma 2.2.** Consider a matrix \(X_0 = \sum_{i=1}^r \sigma_i u_i v_i^T\) of rank \(r\) which is feasible for (2.1), and suppose that the following two conditions hold:

1. There exists a dual point \(Y\) such that \(\mathcal{P}_\Omega(Y) = Y\), \(\mathcal{P}_T(Y) = \sum_{i=1}^r u_i v_i^T\) and \(\|\mathcal{P}_T(Y)\| < 1\);

2. The map \(\mathcal{P}_\Omega\) is injective when restricted to elements in \(T\).

Then \(X_0\) is the unique minimizer to (2.1).

A clear consequence of this lemma is that to prove that \(M\) is the unique solution to (2.1), it suffices to construct a dual certificate for \(M\).

### 2.1.2 A Candidate Dual Certificate

In [CR09], the authors propose a suitable dual for \(M = \sum_{i=1}^r \sigma_i u_i v_i^T\) by considering a related optimization problem: Let \(Y\) be the solution to

\[
\text{minimize} \quad \|X\|_F \\
\text{subject to} \quad \mathcal{P}_T \mathcal{P}_\Omega(X) = \sum_{i=1}^r u_i v_i^T. \tag{2.3}
\]

Note that the solution must satisfy \(\mathcal{P}_\Omega(Y) = Y\) since otherwise \(\mathcal{P}_\Omega(Y)\) would be a solution with smaller Frobenius norm. The motivation for this optimization problem is twofold: Since

\[
\|Y\|_F^2 = \|\mathcal{P}_T(Y)\|_F^2 + \|\mathcal{P}_{T^\perp}(Y)\|_F^2 = \sum_{i=1}^r \|u_i v_i^T\|_F^2 + \|\mathcal{P}_{T^\perp}(Y)\|_F^2 = r + \|\mathcal{P}_{T^\perp}(Y)\|_F^2,
\]

one can view minimizing \(\|Y\|_F\) subject to the constraints as minimizing \(\|\mathcal{P}_{T^\perp}(Y)\|_F\), and one can hope that the Frobenius and spectral norms are not too different. Secondly, since (2.3) is a least-squares optimization, we can express the solution in a closed form. In fact, if \(\mathcal{P}_T \mathcal{P}_\Omega \mathcal{P}_T\) is invertible on \(T\), then

\[
Y = \mathcal{P}_\Omega \mathcal{P}_T (\mathcal{P}_T \mathcal{P}_\Omega \mathcal{P}_T)^{-1} \left( \sum_{i=1}^r u_i v_i^T \right). \tag{2.4}
\]
If $P_T P_{\Omega} P_T$ is invertible, then $P_{\Omega}$ is injective when restricted to elements in $T$, and so to use Lemma 2.2, it suffices to prove that $P_T P_{\Omega} P_T$ is injective and $\|P_T(Y)\| < 1$. In [CR09] the authors make use of randomness and matrix concentration bounds to prove these properties, so the main technical contribution of this paper is showing how to prove these properties when $\Omega$ is not random. This is shown in Section 3.

### 2.2 Learning Product Mixtures over the Hypercube

A distribution $D$ over $\{\pm 1\}^n$ is called a product distribution if every bit in a sample $x \sim D$ is independently chosen. Let $D_1, \ldots, D_k$ be a set of product distributions over $\{\pm 1\}^n$. Associate with each $D_i$ a vector $v_i \in [-1,1]^n$ whose $j$th entry encodes the bias of the $j$th coordinate, that is

$$\mathbb{P}_{x \sim D_i}[x(j) = 1] = \frac{1 + v_i(j)}{2}.$$ 

Define the distribution $D$ to be a convex combination of these product distributions, sampling $x \sim D = \{x \sim D_i \text{ with probability } w_i\}$, where $w_i > 0$ and $\sum_{i \in [k]} w_i = 1$. The distributions $D_1, \ldots, D_k$ are said to be the centers of $D$, the vectors $v_1, \ldots, v_k$ are said to be the bias vectors, and $w_1, \ldots, w_k$ are said to be the mixing weights of the distribution.

**Problem 2.3** (Learning a Product Mixture over the Hypercube). Given independent samples from a distribution $D$ which is a mixture over $k$ centers with bias vectors $v_1, \ldots, v_k \in [-1,1]^n$ and mixing weights $w_1, \ldots, w_k > 0$, recover $v_1, \ldots, v_k$ and $w_1, \ldots, w_k$.

This framework encodes many subproblems, including learning parities, a notorious problem in learning theory; the best current algorithm requires time $n^{O(k)}$, and the noisy version of this problem is a standard cryptographic primitive [MOS04, Fel07, Reg09, Val15]. We do not expect to be able to learn an arbitrary mixture over product distribution efficiently. We obtain a polynomial-time algorithm when the bias vectors are linearly independent, and a quasi-polynomial time algorithm in the general case, though we do require an incoherence assumption on the bias vectors (which parities do not meet), see Definition 2.4.

In [FOS08], the authors give an $n^{O(k)}$-time algorithm for the problem based on the following idea. With great accuracy in polynomial time we may compute the pairwise moments of $D$,

$$M = \mathbb{E}_{x \sim D}[xx^T] = D + \sum_{i \in [k]} w_i \cdot v_i v_i^T.$$ 

The matrix $D$ is a diagonal matrix which corrects for the fact that $M_{jj} = 1$ always. If we were able to learn $D$ and thus access $\sum_{i \in [k]} w_i v_i v_i^T$, the “augmented second moment matrix,” we may hope to use spectral information to learn $v_1, \ldots, v_k$.

The algorithm of [FOS08] performs a brute-force search to learn $D$, leading to a runtime exponential in the rank. By making additional assumptions on the input $D$ and computing third-order moments as well, we avoid this brute force search and give a polynomial-time algorithm for product distributions with linearly independent centers. To guess the matrix $D$, we use adversarial matrix completion. Then, if the bias vectors are linearly independent, an algorithm of [AGH+14] allows us to learn $D$ given access to both the augmented second- and third-order moments (note

---

\*2 There are actually several algorithms in this space; we use [AGH+14] specifically. We note that the works of [HK13, AGH+14, AGHK14] do not include product mixtures over the hypercube, and instead pertain to mixtures of gaussians and similar models for which the second moments are meaningful.
that the third moments of multiplicity > 1 are also lacking in information, and we will leverage adversarial matrix completion to access these as well). In the general case, when the bias vectors are not linearly independent, we instead exploit the fact that the tensor powers of the vectors are independent, and we work with the $\tilde{O}(\log k)$th moments of $D$, applying matrix completion to learn not only the diagonal but other entries of high multiplicity (the tilde hides a dependence on the separation between the bias vectors).

2.3 Matrix Completion and Incoherence

Given a set $\Omega \subseteq [m] \times [n]$ of observed entries of a hidden rank-$r$ matrix $M$, the Matrix Completion problem asks us to successfully recover the matrix $M$ given only $P_\Omega(M)$. Unfortunately, this problem is not always well-posed. For example, consider the input matrix $M = e_1 e_1^T + e_n e_n^T$. $M$ is rank-2, and has only 2 nonzero entries on the diagonal, and zeros elsewhere. Even if we observe almost the entire matrix (and even if the observed indices are random), it is likely that every entry we see will be zero, and so we cannot hope to recover $M$. Because of this, it is standard to ask for the input matrix to be incoherent:

Definition 2.4. Let $U \subseteq \mathbb{R}^n$ be a subspace of dimension $r$. We say that $U$ is incoherent with parameter $\mu$ if $\max_{i \in [n]} \| \text{proj}_U(e_i) \|^2 \leq \mu^2 r$. If $M$ is a matrix with left and right singular spaces $U$ and $V$, we say that $M$ is $(\mu_U, \mu_V)$-incoherent if $U$ (resp. $V$) is incoherent with parameter $\mu_U$ (resp $\mu_V$).

Incoherence means that the singular vectors are well-spread over their coordinates. Intuitively, this asks that every revealed entry actually gives information about the matrix. For a discussion on what kinds of matrices are incoherent, see e.g. [CR09].

2.3.1 Assumptions on Bias Vectors of Product Mixtures

To apply adversarial matrix completion to the problem of learning product mixtures, we must make the assumption that the subspace of $\mathbb{R}^n$ spanned by the bias vectors is incoherent. A similar assumption is made in [JO13], and indeed seems to be necessary whenever matrix completion is attempted. Though this assumption may at first seem peculiar, we will try to motivate it here.

The condition that the subspace spanned by $v_1, \ldots, v_k$ be incoherent can be recast as the condition that no convex combination of the vectors be close to an indicator vector. In a sense, this is a stringent requirement that the centers of the product mixture give the distribution’s unique representation of size at most $k$. Suppose that $v_1 = \lambda_1 u + c_1$ and $v_2 = \lambda_2 u + c_2$ for a vector $u$ and two very sparse vectors $c_1$ and $c_2$. Then the portion of the distribution corresponding to $v_1$ and $v_2$ can instead be represented by the center $(\lambda_1 + \lambda_2) u$, without affecting very many coordinates. This gives a smaller representation of an almost-identical product mixture, violating “strong uniqueness.”

2.4 Useful Identities

We will make use of the following well-known norm properties throughout the paper:

Lemma 2.5. The following norm properties hold:

(a) $\|A\|_1 = \max_{i,j} \| A(e_i e_j^T) \|_1$.
(b) $\|A\|_\infty = \max_{i,j} \| A^*(e_i e_j^T) \|_1$.
(c) For a self-adjoint operator $A$, $\|A\|_1 = \|A\|_\infty$.
(d) $\|A\|_2 \leq \|A\|_1$ and $\|A\|_2 \leq \|A\|_\infty$. This is a consequence of Gershgorin’s Circle Theorem.
we bound each sum separately. Note that are only elements of \( M \) be such that be such that there are only \( \kappa \) elements of \( \Omega \) per column and \( \rho \) per row. Let \( \lambda := (\kappa \frac{\mu_U}{m} + \rho \frac{\mu_V}{n} + \sqrt{\kappa \rho \frac{\mu_U \mu_V}{mn}}) \cdot r \). Then \( \|P_T P_T\|_1 \leq \lambda \).

Proof. First, recall \( \|A\|_1 = \max_{ij} |A(e_i e_j^T)|_1 \), and write

\[
\|P_T P_T(e_i e_j^T)\|_1 = \sum_{(k,l) \in \Omega} |e_k^T P_T(e_i e_j^T) e_l|
\]

\[
= \sum_{(k,l) \in \Omega} |\langle e_i, e_l \rangle e_k^T P_T e_i + \langle e_i, e_k \rangle e_j^T P_V e_l - (e_k^T P_T e_i)(e_j^T P_V e_l)|
\]

\[
\leq \sum_{k: (k,j) \in \Omega} |e_k^T P_U e_i| + \sum_{l: (i,l) \in \Omega} |e_j^T P_V e_l| + \sum_{(k,l) \in \Omega} |(e_k^T P_T e_i)(e_j^T P_V e_l)|.
\]

We bound each sum separately. Note that \( |e_k^T P_U e_i| \leq \|P_U e_i\| \cdot \|P_U e_i\| \), so by the bounds on the lengths of projections, we get a bound on the first and second terms in this expression:

\[
\|P_T P_T\|_1 \leq \kappa \frac{\mu_U r}{m} + \rho \frac{\mu_V r}{n} + \sum_{(k,l) \in \Omega} |(e_k^T P_T e_i)(e_j^T P_V e_l)|.
\]
To bound the last term, an application of Cauchy Schwarz yields

\[
\sum_{(k, \ell) \in \Omega} |(e_k^T P_U e_i)(e_k^T P_V e_\ell)| \leq \sqrt{\sum_{(k, \ell) \in \Omega} (e_k^T P_U e_i)^2} \sqrt{\sum_{(k, \ell) \in \Omega} (e_k^T P_V e_\ell)^2} \\
\leq \sqrt{\kappa \cdot \sum_{k \in [m]} (e_k^T P_U e_i)^2} \sqrt{\rho \cdot \sum_{\ell \in [n]} (e_\ell^T P_V e_j)^2} \\
= \sqrt{\kappa \cdot \|P_U e_i\|_2 \cdot \|P_V e_j\|_2} \\
\leq \sqrt{\kappa \cdot \frac{\mu_U \cdot \mu_V}{mn} \cdot r},
\]

which implies the bound in the lemma.

\[\square\]

**Corollary 3.3.** For \( \lambda := \left(\frac{\mu_U}{m} + \frac{\mu_V}{n} + \sqrt{\frac{\kappa \rho_{xy}}{mn}}\right) r \), \( \|P_{T_\Omega} P_T P_{T_\Omega}\|_\infty \leq \lambda \) and \( \|P_T P_{T_\Omega} P_T\|_2 \leq \lambda \).

**Proof.** These properties follow from Lemma 2.5 and the bound in Lemma 3.2. Let \( A = P_{T_\Omega} P_T \), and note that since \( P_{T_\Omega} \) is a coordinate projection, \( \|P_{T_\Omega}\|_1 = 1 \), so

\[\|P_{T_\Omega} P_T P_{T_\Omega}\|_\infty = \|AA^*\|_\infty = \|AA^*\|_1 = \|A P_{T_\Omega}\|_1 \leq \|A\|_1 \leq \lambda.\]

Furthermore,

\[\|P_T P_{T_\Omega} P_T\|_2 = \|A^* A\|_2 = \|AA^*\|_2 \leq \|AA^*\|_1 \leq \lambda.\]

\[\square\]

**Corollary 3.4.** If \( \lambda < 1 \), then the operator \( P_{T_\Omega} P_T \) is invertible on \( T \).

**Proof.** Simply note that \( P_T P_{T_\Omega} P_T = P_T(I - P_{T_\Omega})P_T = P_T - P_T P_{T_\Omega} P_T \), so if \( X \in T \) and \( X \neq 0 \), then

\[\|P_T P_{T_\Omega} P_T(X)\|_F = \|X - P_T P_{T_\Omega} P_T(X)\|_F \geq (1 - \lambda)\|X\|_F > 0,\]

thus \( P_T P_{T_\Omega} P_T \) has no kernel in \( T \) and must be invertible when restricted to \( T \).

\[\square\]

**Corollary 3.4** is the first of two properties we have to prove to show the solution to (2.3) is a valid dual. It verifies that \( Y = P_{T_\Omega} P_T (P_T P_{T_\Omega} P_T)^{-1}(\sum_{i=1}^r u_i v_i^T) \) is well defined and the solution to (2.3). What remains to prove is that \( \|P_{T^\perp}(Y)\| < 1 \). We expand the inverse as a power series and bound the spectral norm of each term. Let \( \mathcal{H} = P_{T_\Omega} P_T P_{T_\Omega} \). Then since \( \|\mathcal{H}\| \leq \|\mathcal{H}\|_\infty \leq \lambda \), \( \mathcal{H} \) is a contraction so

\[
P_{T^\perp}(Y) = P_{T^\perp} P_{T_\Omega} P_T (E + (I - P_T P_{T_\Omega})P_T + (I - P_T P_{T_\Omega})^2 P_T + \ldots) \\
= P_{T^\perp} (I - P_{T_\Omega}) P_T (E + P_T P_{T_\Omega} P_T + (P_T P_{T_\Omega} P_T)^2 + \ldots) \\
= -P_{T^\perp} P_{T_\Omega} (E + \mathcal{H}(E) + \mathcal{H}^2(E) + \ldots)
\]

Since we have a bound on norms of \( \mathcal{H} \), we can use this to bound the spectral norm of \( P_{T^\perp}(Y) \):

**Theorem 3.5.** Let \( M \) be \((\mu_U, \mu_V)\)-incoherent, and let \( \Omega \subseteq [m] \times [n] \) be such that be such that there are only \( \kappa \) elements of \( \Omega \) per column and \( \rho \) per row. If \( (\kappa \frac{\mu_U}{m} + \rho \frac{\mu_V}{n}) \cdot r < \frac{1}{2} \), then \( \|P_{T^\perp}(Y)\| < 1 \).

**Proof.** Expanding \( P_{T^\perp}(Y) \) as above, and defining \( \mathcal{H} := P_{T_\Omega} P_T P_{T_\Omega} \), \( \|P_{T^\perp}(Y)\| \leq \sum_{k \geq 0} \|P_{T_\Omega} \mathcal{H}^k(E)\| \), where we used the fact that \( \|P_{T^\perp}(X)\| = \|(I - P_U)X(I - P_V)\| \leq \|X\| \) for every \( X \). First, we use a helpful lemma:
Lemma 3.6. Let $\Omega$ be as in Lemma 3.2. Then $\|P_{\Omega}(M)\| \leq \sqrt{\kappa \rho} \|M\|_{\infty}$ for all $M$.

Proof. We bound the spectral norm of $A = P_{\Omega}(M)$ by considering the absolute sums of every row:

$$\|A\|_2 = \|AA^T\| \leq \max_i \|e_i^T A A^T e_i\|_1 = \max_i \sum_j |e_i^T A A^T e_j|,$$

but for every $i$, there are only $\kappa$ nonzero entries of $e_i^T A$, and for each such entry, there are only $\rho$ indices $j$ such that $A^T e_j$ is also nonzero in the corresponding entry. Thus $\|A\|_2 \leq \kappa \rho \|A\|_2^\infty$. But since $P_{\Omega}$ is a coordinate projection, $\|P_{\Omega}(M)\|_\infty \leq \|M\|_\infty$. $\square$

To complete the proof of the theorem, note that $\|H^k(E)\|_\infty \leq \|H\|_\infty^k \|E\|_\infty \leq \lambda^k \|E\|_\infty$ (where we have applied the submultiplicativity of the norm and Corollary 3.3), thus $\|P_{T\perp}(Y)\| \leq \frac{\sqrt{\kappa \rho}}{1-\lambda} \|E\|_\infty$, and finally recall that by Lemma 3.1, $\|E\|_\infty \leq \sqrt{\frac{\mu U \mu V}{mn}} \cdot r$. From the AM-GM inequality and the condition of the theorem,

$$\lambda = \left(\kappa \frac{\mu U}{m} + \rho \frac{\mu V}{n} + \sqrt{\kappa \rho \frac{\mu U \mu V}{mn}}\right) \cdot r < 3/4,$$

and thus $\|P_{T\perp}(Y)\| < 1$. $\square$

Taking Lemma 2.2, Corollary 3.4, and Theorem 3.5 together proves Theorem 1.1.

4 Symmetric Tensor Completion from Multilinear Entries

In this section we use adversarial matrix completion as a primitive to give a completion algorithm for symmetric tensors when only a special kind of entry in the tensor is known. Specifically, we call a string $X \in [n]^m$ multilinear if every element of $X$ is distinct, and we will show how to complete a symmetric tensor $T \in \mathbb{R}^{n^m}$ when only given access to its multilinear entries, i.e. $T(X)$ is known if $X$ is multilinear. In the next section, we will apply our tensor completion algorithm to learn mixtures of product distributions over the boolean hypercube.

Our approach is a simple recursion: we complete the tensor slice-by-slice, using the entries we learn from completing one slice to provide us with enough known entries to complete the next. The following definition will be useful in precisely describing our recursive strategy:

Definition 4.1. Define the histogram of a string $X \in [n]^m$ to be the multiset containing the number of repetitions of each character making at least one appearance in $X$.

For example, the string $(1, 1, 2, 3)$ and the string $(4, 4, 5, 6)$ both have the histogram $(2, 1, 1)$. Note that the entries of the histogram of a string of length $m$ always sum to $m$, and that the length of the histogram is the number of distinct symbols in the string.

Having defined a histogram, we are now ready to describe our tensor completion algorithm.
Algorithm 4.2 (Symmetric Tensor Completion from Multilinear Moments). **Input:** The multilinear entries of the tensor $T = \sum_{i \in [k]} w_i \cdot v_i^{\otimes m} + E$, for vectors $v_1, \ldots, v_k \in \mathbb{R}^n$ and scalars $w_1, \ldots, w_k \geq 0$ and some error tensor $E$. **Goal:** Recover the symmetric tensor $T^* = \sum_{i \in [k]} w_i \cdot v_i^{\otimes 3m}$.

1. Initialize the tensor $\hat{T}$ with the known multilinear entries of $T$.
2. For each subset $Y \in [n]^{m-2}$ with no repetitions:
   - Let $\hat{T}(Y, \cdot, \cdot) \in \mathbb{R}^{n \times n}$ be the tensor slice indexed by $Y$.
   - Remove the rows and columns of $\hat{T}(Y, \cdot, \cdot)$ corresponding to indices present in $Y$. Complete the matrix using Nuclear Norm Minimization and add the learned entries to $\hat{T}$.
3. For $\ell = m - 2, \ldots, 1$:
   - For each $X \in [n]^m$ with a histogram of length $\ell$, if $\hat{T}(X)$ is empty:
     - If there is an element $x_i$ appearing at least 3 times, let $Y = X \setminus \{x_i, x_i\}$.
     - Else there are elements $x_i, x_j$ each appearing twice, let $Y = X \setminus \{x_i, x_j\}$.
     - Let $\hat{T}(Y, \cdot, \cdot) \in \mathbb{R}^{n \times n}$ be the tensor slice indexed by $Y$.
     - Complete the matrix $\hat{T}(Y, \cdot, \cdot)$ using Nuclear Norm Minimization and add the learned entries to $\hat{T}$.
4. Symmetrize $\hat{T}$ by taking each entry to be the average over entries indexed by the same subset.

**Output:** $\hat{T}$.

**Observation 4.3.** One might ask why we go through the effort of completing the tensor slice-by-slice, rather than simply flattening it to an $n^{m/2} \times n^{m/2}$ matrix and completing that. The reason is that when span $v_1, \ldots, v_k$ has incoherence $\mu$ and dimension $r$, span $v_1^{\otimes m/2}, \ldots, v_k^{\otimes m/2}$ may have incoherence as large as $\mu r^m/k$, which drastically reduces the range of parameters for which recovery is possible (for example, if $k = O(r)$ then we would need $r < n^{1/m}$). Working slice-by-slice keeps the incoherence of the input matrices small, allowing us to complete even up to rank $r = \Omega(n)$.

**Theorem 4.4.** Let $T$ be a symmetric tensor of order $m$, so that $T = \sum_{i \in [k]} w_i \cdot v_i^{\otimes m}$ for some vectors $v_1, \ldots, v_k \in \mathbb{R}^n$ and scalars $w_1, \ldots, w_k \neq 0$. Let span $\{v_i\}$ have incoherence $\mu$ and dimension $r$. Given perfect access to all multilinear entries of $T$ (i.e. $E = 0$), if $4 \cdot \mu \cdot r \cdot m/n < 1$, then Algorithm 4.2 returns the full tensor $T$ in time $\tilde{O}(n^{m+1})$.

In Appendix B, we give a version of Theorem 4.4 that accounts for error $E$ in the input.

**Proof.** We prove that Algorithm 4.2 successfully completes all the entries of $T$ by induction on the length of the histograms of the entries. By assumption, we are given as input every entry with a histogram of length $m$. For an entry $X$ with a histogram of length $m - 1$, exactly one of its elements has multiplicity two, call it $x_i$, and consider the set $Y = X \setminus \{x_i, x_i\}$. When step 2 reaches $Y$, the algorithm attempts to complete a matrix revealed from $T(Y, \cdot, \cdot) = \mathcal{P}_Y \left( \sum_{i \in [k]} w_i \cdot v_i(Y) \cdot v_i^T \right)$, where $v_i(Y) = \prod_{j \in Y} v_i(j)$, and $\mathcal{P}_Y$ is the projector to the matrix with the rows and columns corresponding to indices appearing in $Y$ removed. Exactly the diagonal of $T(Y, \cdot, \cdot)$ is missing since all other entries are multilinear moments, and the $(i, i)$th entry should be $T(X)$. Because the rank of this matrix is equal to dim(span($v_i$)) = $r$ and $4\mu r/n \leq 4\mu rm/n < 1$, by Theorem 1.1, Nuclear
Norm Minimization successfully recovers the diagonal, including $T(X)$. Thus by the end of step 2, $\hat{T}$ contains every entry with a histogram of length $\ell \geq m + 1$.

For the inductive step, we prove that each time step 3 completes an iteration, $\hat{T}$ contains every entry with a histogram of length at least $\ell$. Let $X$ be an entry with a histogram of length $\ell$. When step 3 reaches $X$ in the $\ell$th iteration, if $\hat{T}$ does not already contain $T(X)$, the algorithm attempts to complete a matrix with entries revealed from $T(Y, \cdot) = \sum_{i\in[k]} w_i \cdot v_i(Y) \cdot v_i^T$, where $Y$ is a substring of $X$ with a histogram of the same length. Since $Y$ has a histogram of length $\ell$, every entry of $T(Y, \cdot)$ corresponds to an entry with a histogram of length at least $\ell + 1$, except for the $\ell \times \ell$ principal submatrix whose rows and columns correspond to elements in $Y$. Thus by the inductive hypothesis, $\hat{T}(Y)$ is only missing the aforementioned submatrix, and since $4\mu r \ell / n \leq 4\mu rm / n < 1$, by Theorem 1.1, nuclear norm minimization successfully recovers this submatrix, including $T(X)$. Once all of the entries of $\hat{T}$ are filled in, the algorithm terminates.

Finally, we note that the runtime is $\tilde{O}(n^{m+1})$, because the Nuclear Norm Minimization runs in time $\tilde{O}(n^3)$, and we perform at most $n^{m-2}$ matrix completions because there are $n^{m-2}$ strings of length $m - 2$ over the alphabet $[n]$, and we perform at most one matrix completion for each such string. \qed

5 Learning Product Mixtures over the Hypercube

In this section, we apply our symmetric tensor completion algorithm (Algorithm 4.2) to learning mixtures of product distributions over the hypercube, proving Theorem 1.2. Throughout this section we will assume exact access to moments of our input distribution, deferring finite-sample error analysis to Appendix B. We begin by introducing convenient notation.

Let $D$ be a mixture over $k$ centers with bias vectors $v_1, \ldots, v_k \in [-1, 1]^n$ and mixing weights $w_1, \ldots, w_k > 0$. Define $\mathcal{M}_m^D \in \mathbb{R}^{n^m}$ to be the tensor of order-$m$ moments of the distribution $D$, so that $\mathcal{M}_m^D = E_{x \sim D} [x^{\otimes m}]$. Define $T_m^D \in \mathbb{R}^{n^m}$ to be the symmetric tensor given by the weighted bias vectors of the distribution, so that $T_m^D = \sum_{i \in [k]} w_i \cdot v_i^{\otimes m}$.

Note that $T_m^D$ and $\mathcal{M}_m^D$ are equal on their multilinear entries, and not necessarily equal elsewhere. For example, when $m$ is even, entries of $\mathcal{M}_m^D$ indexed by a single repeating character (the “diagonal”) are always equal to 1. Also observe that if one can sample from distribution $D$, then estimating $\mathcal{M}_m^D$ is easy.

Suppose that the bias vectors of $D$ are linearly independent. Then by Theorem 5.1 (due to [AGH+14], with similar statements appearing in [AHK12, HIK13, AGHK14]), there is a spectral algorithm which learns $D$ given $T_2^D$ and $T_3^D$ (we give an account of the algorithm in Appendix C).

Theorem 5.1 (Consequence of Theorem 4.3 and Lemma 5.1 in [AGH+14]). Let $D$ be a mixture over $k$ centers with bias vectors $v_1, \ldots, v_k \in [-1, 1]^n$ and mixing weights $w_1, \ldots, w_k > 0$. Suppose we are given access to $T_2^D = \sum_{i \in [k]} w_i \cdot v_i v_i^T$ and $T_3^D = \sum_{i \in [k]} w_i \cdot v_i^{\otimes 3}$. Then there is an algorithm which recovers the bias vectors and mixing weights of $D$ within $\varepsilon$ in time $O(n^3 + k^4 \cdot (\log \log \frac{1}{\varepsilon} / w_{\min}))$.

Because $T_2^D$ and $T_3^D$ are equal to $\mathcal{M}_2^D$ and $\mathcal{M}_3^D$ on their multilinear entries, the tensor completion algorithm of the previous section allows us to find $T_2^D$ and $T_3^D$ from $\mathcal{M}_2^D$ and $\mathcal{M}_3^D$ (this is only

\footnote{We remark again that the result in [AGH+14] is quite general, and applies to a large class of probability distributions of this character. However the work deals exclusively with distributions for which $M_2 = T_2$ and $M_3 = T_3$, and assumes access to $T_2$ and $T_3$ through moment estimation.}
possible because $\mathcal{T}_2^D$ and $\mathcal{T}_3^D$ are low-rank, whereas $\mathcal{M}_2^D$ and $\mathcal{M}_3^D$ are high-rank). We then learn $\mathcal{D}$ by applying Theorem 5.1.

A complication is that Theorem 5.1 only allows us to recover the parameters of $\mathcal{D}$ if the bias vectors are linearly independent. However, if the vectors $v_1, \ldots, v_k$ are not linearly independent, we can reduce to the independent case by working instead with $v_1^\otimes m, \ldots, v_k^\otimes m$ for sufficiently large $m$. The tensor power we require depends on the separation between the bias vectors:

**Definition 5.2.** We call a set of vectors $v_1, \ldots, v_k$ $\eta$-separated if for every $i, j \in [k]$ such that $i \neq j$,

$$|\langle v_i, v_j \rangle| \leq \|v_i\| \cdot \|v_j\| \cdot (1 - \eta).$$

**Lemma 5.3.** Suppose that $v_1, \ldots, v_k \in \mathbb{R}^n$ are vectors which are $\eta$-separated, for $\eta > 0$. Let $m \geq \lceil \log \frac{\eta}{1 - \eta} \rceil$. Then $v_1^\otimes m, \ldots, v_k^\otimes m$ are linearly independent.

**Proof.** For vectors $u, w \in \mathbb{R}^n$ and for an integer $t \geq 0$, we have that $\langle u^\otimes t, w^\otimes t \rangle = \langle u, w \rangle^t$. If $v_1, \ldots, v_k$ are $\eta$-separated, then for all $i \neq j$,

$$\left| \left\langle \frac{v_i^\otimes m}{\|v_i\|^m}, \frac{v_j^\otimes m}{\|v_j\|^m} \right\rangle \right| \leq |(1 - \eta)^m| \leq \frac{1}{k}.$$ 

Now considering the Gram matrix of the vectors $(\frac{m}{\|v_i\|^m})^\otimes m$, we have a $k \times k$ matrix with diagonal entries of value 1 and off-diagonal entries with maximum absolute value $\frac{1}{k}$. This matrix is strictly diagonally dominant, and thus full rank, so the vectors must be linearly independent. \[\Box\]

Thus, in the linearly dependent case, we may choose an appropriate power $m$, and instead apply the tensor completion algorithm to $\mathcal{M}_2^D$ and $\mathcal{M}_3^D$ to recover $\mathcal{T}_2^D$ and $\mathcal{T}_3^D$. We will then apply Theorem 5.1 to the vectors $v_1^\otimes m, \ldots, v_k^\otimes m$ in the same fashion.

Here we give the algorithm assuming perfect access to the moments of $\mathcal{D}$ and defer discussion of the finite-sample case to Appendix B.

**Algorithm 5.4 (Learning Mixtures of Product Distributions).** **Input:** Moments of the distribution $\mathcal{D}$. **Goal:** Recover $v_1, \ldots, v_k$ and $w_1, \ldots, w_k$.

Let $m$ be the smallest odd integer such that $v_1^\otimes m, \ldots, v_k^\otimes m$ are linearly independent. Let $\hat{M} = \mathcal{M}_{2m} + \hat{E}_2$ and $\hat{T} = \mathcal{M}_{3m} + \hat{E}_3$ be approximations to the moment tensors of order $2m$ and $3m$.

1. Set the non-multilinear entries of $\hat{M}$ and $\hat{T}$ to “missing,” and run Algorithm 4.2 on $\hat{M}$ and $\hat{T}$ to recover $M' = \sum_i w_i \cdot v_i^\otimes 2m + E'_2$ and $T' = \sum_i w_i \cdot v_i^\otimes 3m + E'_3$.
2. Flatten $M'$ to the $n^m \times n^m$ matrix $M = \sum_i w_i \cdot v_i^\otimes (v_i^\otimes m)^\top + E_2$ and similarly flatten $T'$ to the $n^m \times n^m$ matrix $T = \sum_i w_i \cdot (v_i^\otimes m)^\otimes 3 + E_3$.
3. Run the “whitening” algorithm from Theorem 5.1 (see Appendix C) on $(M, T)$ to recover $w_1, \ldots, w_k$ and $v_1^\otimes m, \ldots, v_k^\otimes m$.
4. Recover $v_1, \ldots, v_k$ entry-by-entry, by taking the $m$th root of the corresponding entry in $v_1^\otimes m, \ldots, v_k^\otimes m$.

**Output:** $w_1, \ldots, w_k$ and $v_1, \ldots, v_k$.

Now Theorem 1.2 is a direct result of the correctness of Algorithm 5.4:

**Proof of Theorem 1.2.** The proof follows immediately by combining Theorem 5.1 and Theorem 4.4, and noting that the parameter $m$ is bounded by $m \leq 2 + \log \frac{1}{1 - \eta} \ k$. \[\Box\]
Acknowledgements

We would like to thank Prasad Raghavendra, Satish Rao, and Ben Recht for helpful discussions, and Moritz Hardt and Samuel B. Hopkins for helpful questions.

References

[AGH+14] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky, Tensor decompositions for learning latent variable models, Journal of Machine Learning Research 15 (2014), no. 1, 2773–2832. 2, 3, 6, 12, 19, 20, 21

[AGHK14] Animashree Anandkumar, Rong Ge, Daniel Hsu, and Sham M. Kakade, A tensor approach to learning mixed membership community models, Journal of Machine Learning Research 15 (2014), no. 1, 2239–2312. 6, 12, 20

[AGJ14a] Anima Anandkumar, Rong Ge, and Majid Janzamin, Analyzing tensor power method dynamics: Applications to learning overcomplete latent variable models, CoRR abs/1411.1488 (2014). 21

[AGJ14b] Animashree Anandkumar, Rong Ge, and Majid Janzamin, Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates, CoRR abs/1402.5180 (2014). 21

[AGM12] Sanjeev Arora, Rong Ge, and Ankur Moitra, Learning topic models - going beyond SVD, 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, 2012, pp. 1–10. 2

[AHK12] Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade, A method of moments for mixture models and hidden markov models, COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland, 2012, pp. 33.1–33.34. 12, 20

[BK07] Robert M. Bell and Yehuda Koren, Scalable collaborative filtering with jointly derived neighborhood interpolation weights, Proceedings of the 2007 Seventh IEEE International Conference on Data Mining (Washington, DC, USA), ICDM ’07, IEEE Computer Society, 2007, pp. 43–52. 1

[CP09] Emmanuel J. Candès and Yaniv Plan, Matrix completion with noise, CoRR abs/0903.3131 (2009). 16, 17

[CR08] Kamalika Chaudhuri and Satish Rao, Learning mixtures of product distributions using correlations and independence, 21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008, 2008, pp. 9–20. 2

[CR09] Emmanuel J. Candès and Benjamin Recht, Exact matrix completion via convex optimization, Foundations of Computational Mathematics 9 (2009), no. 6, 717–772. 1, 4, 5, 6, 7, 8

[CT09] Emmanuel J. Candès and Terence Tao, The power of convex relaxation: Near-optimal matrix completion, CoRR abs/0903.1476 (2009). 1
A Nuclear Norm Minimization and Tensor Completion with Noise

Here we will present versions of Theorem 1.1 and Theorem 4.4 which account for noise in the input to the algorithm.

A.1 Noisy Nuclear Norm Minimization

In many practical applications, one is not capable of exactly observing the entries of the hidden low-rank matrix $M$. Instead, one observes a matrix $X = M + Z$, where $Z$ is assumed to be small in some sense, usually Frobenius norm. The most natural generalization of the optimization (2.1) is

$$\begin{align*}
\text{minimize} & \quad \|X\|_* \\
\text{subject to} & \quad \|P_\Omega(X) - P_\Omega(M)\|_F \leq \delta.
\end{align*}$$

(A.1)

Indeed, in [CP09], the authors show that (A.1) is a good relaxation when entries are revealed randomly, and it turns out that in the adversarial case, essentially the same proof goes through. We include their proof here for completeness.
Theorem A.1. Let $M = UΣV^T$ and $Ω = [m] \times [n]$, and define $E = UV^T$ and the subspace $T$ as earlier. If $\|P_T - P_T P_Ω P_T\|_F \leq λ < 1$ and there exists a $Y$ (the dual certificate) such that $P_Ω(Y) = Y$, $P_T(Y) = E$, and $P_{T^⊥}(Y) = c < 1$, then the solution to (A.1), denoted $\hat{M}$, satisfies

$$\|\hat{M} - M\|_F \leq 2δ + \frac{2δ\sqrt{\min(n,m)}}{1 - c} \sqrt{1 + \frac{1}{1 - λ}}.$$ 

Before proving Theorem A.1 we need the following lemma:

Lemma A.2 (Lemma 4 in [CP09]). Suppose there exists a $Y$ as in the hypotheses of Theorem A.1 and consider any $H$ satisfying $P_Ω(H) = 0$. Then $\|M + H\|_* \geq \|M\|_* - \|1 - P_{T^⊥}(Y)\|\|P_{T^⊥}(H)\|_*$. 

Proof. By the definition of subgradient, for any $Z \in \partial\|M\|_*$, $\|M + H\|_* \geq \|M\|_* + \langle Z, H \rangle - \|Z, P_{T^⊥}(H)\|$. Writing $Y = E + P_{T^⊥}(Y)$ and $Z = E + P_{T^⊥}(Z)$, we have

$$\|M + H\|_* \geq \|M\|_* + \langle Y, H \rangle + \langle P_{T^⊥}(Z - Y), H \rangle = \|M\|_* + \langle Z - Y, P_{T^⊥}(H) \rangle.$$ 

Since the spectral and nuclear norms are dual to each other, there exists a $Z$ such that $\|Z\| \leq 1$ and $\langle Z, P_{T^⊥}(H) \rangle = \|P_{T^⊥}(H)\|_*$, and $\|Y, P_{T^⊥}(H)\| \leq \|P_{T^⊥}(Y)\| \cdot \|P_{T^⊥}(H)\|_*$. Plugging these into the above equation proves the lemma.

Now we can prove the main theorem:

Proof of Theorem A.1. Denote $H = \hat{M} - M$, and recall that our goal is to bound $\|H\|_F$. Since $M$ is feasible for (A.1), it is clear that $\|\hat{M}\|_* \leq \|M\|_*$, and $\|P_Ω(H)\| = \|P_Ω(\hat{M}) - P_Ω(M)\|_F \leq 2δ$. It remains to bound $\|P_{T^⊥} P_Ω(H)\|_*$. Triangle inequality gives

$$\|M + H\|_* \geq \|M\|_* + \|P_{T^⊥} P_Ω(H)\|_* - \|P_Ω(H)\|_*,$$

and combining this with Lemma 4, we get

$$\|M\|_* \geq \|M + H\|_* \geq \|M\|_* + (1 - c)\|P_{T^⊥} P_Ω(H)\|_* - \|P_Ω(H)\|_*.$$ 

Rearranging, we get

$$\|P_{T^⊥} P_Ω(H)\|_* \leq \frac{1}{1 - c} \|P_Ω(H)\|_* \leq \frac{\sqrt{n}}{1 - c} \|P_Ω(H)\|_F \leq \frac{2δ\sqrt{n}}{1 - c},$$

and recall that the nuclear norm dominates the frobenius norm, so the same bound holds for $\|P_{T^⊥} P_Ω(H)\|_F$. To bound $\|P_{T^⊥} P_Ω(H)\|_F$, note

$$\|P_Ω P_T P_Ω(H)\|^2_2 = \langle P_Ω P_T P_Ω(H), P_Ω P_T P_Ω(H) \rangle = \langle P_T P_Ω P_T P_Ω(H), P_T P_Ω P_Ω(H) \rangle \geq (1 - λ)\|P_T P_Ω P_Ω(H)\|^2_2.$$ 

Furthermore, $\|P_Ω P_T P_Ω(H)\|_F = \|P_T P_Ω P_Ω(H)\|_F \leq \|P_T P_Ω P_Ω(H)\|_F$, and combining this with the previous inequality, we get $\|P_T P_Ω P_Ω(H)\|^2_2 \leq \frac{1}{1 - λ} \|P_T P_Ω P_Ω(H)\|^2_2$. Finally, we get

$$\|P_Ω(H)\|^2_2 \leq \left(1 + \frac{1}{1 - λ}\right) \|P_T P_Ω P_Ω(H)\|^2_2,$$

and the theorem follows by taking this and the bound on $\|P_T P_Ω P_Ω(H)\|_F$. 

17
A.2 Noisy Symmetric Tensor Completion

We also give an analysis for the performance of our tensor completion algorithm, Algorithm 4.2, in the presence of noise in the input moments. This will enable us to use the algorithm on empirically estimated moments.

Theorem A.3. Let $T^*$ be a symmetric tensor of order $m$, so that $T^* = \sum_{i \in [k]} w_i \cdot v_i^{\otimes m}$ for some vectors $v_1, \ldots, v_k \in \mathbb{R}^n$ and scalars $w_1, \ldots, w_k \neq 0$. Let $\text{span}\{v_i\}$ have incoherence $\mu$ and dimension $r$. Suppose we are given access to $T = T^* + E$, where $E$ is a noise tensor with $|E(Y)| \leq \varepsilon$ for every $Y \in [n]^m$. Then if
\[
4 \cdot k \cdot \mu \cdot m \leq n,
\]
Algorithm 4.2 recovers a symmetric tensor $\hat{T}$ such that
\[
\|\hat{T}(X, \cdot, \cdot) - T^*(X, \cdot, \cdot)\|_F \leq 4 \cdot \varepsilon \cdot (5n^{3/2})^{m-1},
\]
for any slice $T(X, \cdot, \cdot)$ indexed by a string $X \in [n]^{m-2}$, in time $\tilde{O}(n^{m+1})$. In particular, the total Frobenius norm error $\|\hat{T} - T^*\|_F$ is bounded by $4 \cdot \varepsilon \cdot (5n^{3/2})^{3m-2}$.

Proof. We proceed by induction on the histogram length of the entries: we will prove that an entry with a histogram of length $\ell$ has error at most $\varepsilon(5n^{3/2})^{m-\ell}$.

In the base case of $\ell = m$, we have that by assumption, every entry of $E$ is bounded by $\varepsilon$.

Now, for the inductive step, consider an entry $X$ with a histogram of length $\ell \leq m - 1$. In filling in the entry $T(X)$, we only use information from entries with shorter histograms, which by the inductive hypothesis each have error at most $\alpha = \varepsilon(5n^{3/2})^{m-\ell-1}$. Summing over the squared errors of the individual entries, the squared Frobenius norm error of the known entries in the slice in which $T(X)$ was completed, pre-completion is at most $n^2 \alpha^2$. Due to the assumptions on $k, \mu, m, n$, by Theorem A.1, Nuclear Norm Minimization amplifies the Frobenius norm error of $\beta$ to at most a Frobenius norm error of $5\beta \cdot n^{1/2}$. Thus, we have that the Frobenius norm of the slice $T(X)$ was completed in, post-completion, is at most $5n^{3/2} \alpha$, and therefore that the error in the entry $T(X)$ is as most $\varepsilon \cdot (5n^{3/2})^{m-\ell}$, as desired.

This concludes the induction. Finally, as our error bound is per entry, it is not increased by the symmetrization in step 4. Any slice has at most one entry with a histogram of length one, $2n - 2$ entries with a histogram of length two, and $n^2 - (2n - 1)$ entries with a histogram of length three. Thus the total error in a slice is at most $4 \cdot \varepsilon \cdot (5n^{3/2})^{m-1}$, and there are $n^{m-2}$ slices.

B Empirical Moment Estimation for Learning Product Mixtures

In Section 5, we detailed our algorithm for learning mixtures of product distributions while assuming access to exact moments of the distribution $D$. Here, we will give an analysis which accounts for the errors introduced by empirical moment estimation. We note that we made no effort to optimize the sample complexity, and that a tighter analysis of the error propagation may well be possible.
**Algorithm B.1** (Learning product mixture over separated centers). **Input:** $N$ independent samples $x_1, \ldots, x_N$ from $\mathcal{D}$, where $\mathcal{D}$ has bias vectors with separation $\eta > 0$. **Goal:** Recover the bias vectors and mixing weights of $\mathcal{D}$.

Let $m$ be the smallest odd integer for which $v_1^{\otimes m}, \ldots, v_k^{\otimes m}$ become linearly independent.

1. Empirically estimate $M_{2m}$ and $M_{3m}$ by calculation $M := \frac{1}{N} \sum_{i \in [N]} (x_i^{\otimes m})(x_i^{\otimes m})^\top$ and $T := \frac{1}{N} \sum_{i \in [N]} (x_i^{\otimes m}) \otimes 3$.

2. Run Algorithm 5.4 on $M$ and $T$.

**Output:** The approximate mixing weights $\hat{w}_1, \ldots, \hat{w}_k$, and the approximate vectors $\hat{v}_1, \ldots, \hat{v}_k$.

**Theorem B.2** (Theorem 1.2 with empirical moment estimation). Let $\mathcal{D}$ be a product mixture over $k$ centers with bias vectors $v_1, \ldots, v_k \in [-1, 1]^n$ and mixing weights $w_1, \ldots, w_k > 0$. Let $m$ be the smallest odd integer for which $v_1^{\otimes m}, \ldots, v_k^{\otimes m}$ are linearly independent (if $v_1, \ldots, v_k$ are $\eta$-separated for $\eta > 0$, then $m \leq \log \frac{1}{\eta} k$). Define $M_{2m} = \sum_{i \in [k]} v_i^{\otimes m}(v_i^{\otimes m})^\top$. Suppose

$$4 \cdot m \cdot r \cdot \mu \leq n,$$

where $\mu$ and $r$ are the incoherence and dimension of the space span$\{v_i\}$ respectively. Furthermore, let $\beta \leq \min \left( O(1/k \sqrt{w_{\max}}), \frac{1}{10} \right)$ be suitably small, and let the parameter $N$ in Algorithm B.1 satisfy $N \geq \frac{2}{\varepsilon^2} (4 \log n + \log \frac{1}{\beta})$ for $\varepsilon$ satisfying

$$\varepsilon \leq \frac{\beta \cdot \sigma_k(M_{2m})}{4 \cdot (5n^{3/2})^{3m-2}} \min \left( \frac{1}{6 \sqrt{w_{\max}}}, \frac{\sigma_k(M)^{1/2}}{(5n^{3/2})^{3m/2}} \right).$$

Finally, pick any $\eta \in (0, 1)$. Then with probability at least $1 - \delta - \eta$, Algorithm B.1 returns vectors $\hat{v}_1, \ldots, \hat{v}_k$ and mixing weights $\hat{w}_1, \ldots, \hat{w}_k$ such that

$$\|\hat{v}_i - v_i\| \leq \sqrt{n} \left( 10 \cdot \beta + 60 \cdot \beta \cdot \|M_{2m}\|^{1/2} + \frac{\beta \cdot \sigma_k(M_{2m})}{6 \sqrt{w_{\max}}} \right)^{1/m}, \quad \text{and} \quad |\hat{w}_i - w_i| \leq 40 \beta,$$

and runs in time $n^{O(m)} \cdot O(N \cdot \text{poly}(k) \log(1/\eta) \cdot (\log k + \log \log(w_{\max})))$. In particular, a choice of $N \geq n^{\tilde{O}(m)}$ gives sub-constant error, where the tilde hides the dependence on $w_{\min}$ and $\sigma_k(M_{2m})$.

Before proving Theorem B.2, we will state state the guarantees of the whitening algorithm of [AGH+14] on noisy inputs, which is used as a black box in Algorithm 5.4. We have somewhat modified the statement in [AGH+14] for convenience; for a brief account of their algorithm, as well as an account of our modifications to the results as stated in [AGH+14], we refer the reader to Appendix C.

**Theorem B.3** (Corollary of Theorem 4.3 in [AGH+14]). Let $v_1, \ldots, v_k \in [-1, 1]^n$ be vectors and let $w_1, \ldots, w_k > 0$ be weights. Define $M := \sum_{i \in [k]} w_i \cdot v_i v_i^\top$ and $T := \sum_{i \in [k]} w_i \cdot v_i^{\otimes 3}$, and suppose we are given $\hat{M} = M + E_M$ and $\hat{T} = T + E_T$, where $E_M \in \mathbb{R}^{n \times n}$ and $E_T \in \mathbb{R}^{n \times n \times n}$ are symmetric error terms such that

$$2\beta := \frac{6\|E_M\| \sqrt{w_{\max}}}{\sigma_k(M)} + \frac{\sqrt{E_T}\|E_T\|}{\sigma_k(M)^{3/2}} \leq O \left( \frac{1}{\sqrt{w_{\max} \cdot k}} \right).$$
Then there is an algorithm that recovers vectors $\hat{v}_1, \ldots, \hat{v}_k$ and weights $\hat{w}_1, \ldots, \hat{w}_k$ such that for all $i \in [n]$,

$$\|v_i - \hat{v}_i\| \leq \|E_M\|^{1/2} + 60\|M\|^{1/2} + 10\beta, \quad \text{and} \quad |w_i - \hat{w}_i| \leq 40\beta,$$

with probability $1 - \eta$ in time $O(L \cdot k^3 \cdot (\log k + \log \log(\frac{1}{\max w}))$, where $L$ is $\text{poly}(k) \log(1/\eta)$.

Having stated the guarantees of the whitening algorithm, we are ready to prove Theorem B.2.

**Proof of Theorem B.2.** We account for the noise amplification in each step.

**Step 1:** In this step, we empirically estimate the multilinear moments of the distribution. We will apply concentration inequalities on each entry individually. By a Hoeffding bound, each entry concentrates within $\varepsilon$ of its expectation with probability $1 - \exp(-\frac{1}{2}N \cdot \varepsilon^2)$. Taking a union bound over the $\binom{n}{2m} + \binom{n}{3m}$ moments we must estimate, we conclude that with probability at least $1 - \exp(-\frac{1}{2}N \cdot \varepsilon^2 + 4m \log n)$, all moments concentrate to within $\varepsilon$ of their expectation. Setting $N = \frac{2}{\varepsilon^2}(4m \log n + \log \frac{1}{\delta})$, we have that with probability $1 - \delta$, every entry concentrates to within $\varepsilon$ of its expectation.

Now, we run Algorithm 5.4 on the estimated moments.

**Step 1 of Algorithm 5.4:** Applying Theorem A.3, we see that the error satisfies $\|E'_2\|_F \leq 4 \cdot \varepsilon \cdot (5n^{3/2})^{3m-2}$ and $\|E'_3\|_F \leq 4 \cdot \varepsilon \cdot (5n^{3/2})^{2m-2}$.

**Step 2 of Algorithm 5.4:** No error is introduced in this step.

**Step 3 of Algorithm 5.4:** Here, we apply Theorem B.3 out of the box, where our vectors are now the $v_i^{\otimes m}$. The desired result now follows immediately for the estimated mixing weights, and for the estimated tensored vectors we have $\|u_i - v_i^{\otimes m}\| \leq 10 \cdot \beta + 60 \cdot \beta \|M\|^{1/2} + \|E'_2\|$, for $\beta$ as defined in Theorem B.2. Note that $\|E'_2\| \leq \|E'_2\|_F \leq \beta \cdot \sigma_k(M_{2m})/6\sqrt{w_{\max}}$, so let $\gamma = 10 \cdot \beta + 60 \cdot \beta \|M\|^{1/2} + \frac{\beta \cdot \sigma_k(M_{2m})}{6\sqrt{w_{\max}}}$.\[\]

**Step 4 of Algorithm 5.4:** Let $u_i^*$ be the restriction of $u_i$ to the single-index entries, and let $v_i^*$ be the same restriction for $v_i^{\otimes m}$. The bound on the error of the $u_i$ applies to restrictions, so we have $\|u_i^* - v_i^*\| \leq \gamma$. So the error in each entry is bounded by $\gamma$. By the concavity of the $m$th root, we thus have that $\|v_i - \hat{v}_i\| \leq \sqrt{\gamma} \cdot \gamma^{1/m}$.

To see that choosing $N \geq n^{\tilde{O}(m)}$ gives sub-constant error, calculations suffice; we only add that $\|M_{2m}\| \leq rn^m$, where we have applied a bound on the Frobenius norm of $\|M_{2m}\|$. The tilde hides the dependence on $w_{\min}$ and $\sigma_k(M_{2m})$. This concludes the proof.\[\]

**C Recovering Distributions from Second- and Third-Order Tensors**

In this appendix, we give an account of the algorithm of [AGH+14] which, given access to estimates of $M^D_{\otimes 2}$ and $T^D_{\otimes 2}$, can recover the parameters of $\mathcal{D}$. We note that the technique is very similar to those of [AHK12, HK13, AGHK14], but we use the particular algorithm of [AGH+14]. In previous sections, we have given a statement that follows from their results; here we will detail the connection.

In [AGH+14], the authors show that for a family of distributions with parameters $v_1, \ldots, v_k \in \mathbb{R}^n$ and $w_1, \ldots, w_k > 0$, if the $v_1, \ldots, v_k$ are linearly independent and one has approximate access to
\[ M_V := \sum_{i \in [k]} w_i v_i v_i^T \text{ and } T_V := \sum_{i \in [k]} w_i \cdot v_i^\otimes 3, \] 
then the parameters can be recovered. For this, they use two algorithmic primitives: singular value decompositions and tensor power iteration.

Tensor power iteration is a generalization of the power iteration technique for finding matrix eigenvectors to the tensor setting (see e.g. \cite{AGH14}). The generalization is not complete, and the convergence criteria for the method are quite delicate and not completely understood, although there has been much progress in this area of late (\cite{AGJ14b, AGJ14a, GHJY15}). However, it is well-known that when the input tensor \( T \in \mathbb{R}^{n \times n \times n} \) is decomposable into \( k < n \) symmetric orthogonal rank-1 tensors, i.e. \( T = \sum_{i \in [k]} v_i^\otimes 3 \) where \( k < n \) and \( \langle v_i, v_j \rangle = 0 \) for \( i \neq j \), then it is possible to recover \( v_1, \ldots, v_k \) using tensor power iteration.

The authors of \cite{AGH14} prove that this process is robust to some noising of \( T \):

**Theorem C.1** (Theorem 5.1 in \cite{AGH14}). Let \( \tilde{T} = T + E \in \mathbb{R}^{k \times k \times k} \) be a symmetric tensor, where \( T \) has the decomposition \( T = \sum_{i \in [k]} \lambda_i \cdot u_i \otimes u_i \otimes u_i \) for orthonormal vectors \( u_1, \ldots, u_k \) and \( \lambda_1, \ldots, \lambda_k > 0 \), and \( E \) is a tensor such that \( \|E\|_F \leq \beta \). Then there exist universal constants \( C_1, C_2, C_3 > 0 \) such that the following holds. Choose \( \eta \in (0, 1) \), and suppose

\[
\beta \leq C_1 \cdot \frac{\lambda_{\min}}{k}
\]

and also

\[
\sqrt{\frac{\ln(L/\log(2/k/\eta))}{\ln k}} \cdot \left(1 - \frac{\ln(\ln(L/\log(2/k/\eta)))}{4 \ln(\ln(\ln(L/\log(2/k/\eta))))} + C_3 \right) \geq 1.02 \left(1 + \sqrt{\frac{\ln 4}{\ln k}}\right).
\]

Then there is a tensor power iteration based algorithm that recovers vectors \( \hat{u}_1, \ldots, \hat{u}_k \) and coefficients \( \hat{\lambda}_1, \ldots, \hat{\lambda}_k \) with probability at least \( 1 - \eta \) such that for all \( i \in [n] \),

\[
\|\hat{u}_i - u_i\| \leq \beta \frac{8}{\lambda_i}, \quad \text{and} \quad |\hat{\lambda}_i - \lambda_i| \leq 5\beta,
\]

in \( O(L \cdot k^3 \cdot (\log k + \log \log(\lambda_{\min})) \) time. The conditions are met when \( L = \text{poly}(k \log(1/\eta)) \).

The idea is then to take the matrix \( M_V \), and apply a whitening map \( W = (M_V^1)^{1/2} \) to orthonormalize the vectors. Because \( v_1, \ldots, v_k \) are assumed to be linearly independent, and because \( WWM = \sum_{i \in [k]} w_i (Wv_i) (Wv_i)^T = \text{Id}_k \), it follows that the \( \sqrt{w_i} \cdot Wv_i \) are orthogonal vectors. Now, applying the map \( W \in \mathbb{R}^{k \times n} \) to every slice of \( T \) in every direction, we obtain a new tensor \( T_W = \sum_{i \in [k]} w_i (Wv_i)^\otimes 3 \), by computing each entry:

\[
T(W, W, W)_{a, b, c} := T_W(a, b, c) = \sum_{1 \leq a', b', c' \leq n} W^T(a', a) \cdot W^T(b', b) \cdot W^T(c', c) \cdot T(a', b', c').
\]

From here on out we will use \( T(A, A, A) \) to denote this operation on tensors. The tensor \( T_W \) thus has an orthogonal decomposition. Letting \( u_i = \sqrt{w_i} Wv_i \), we have that \( T = \sum_{i \in [k]} \frac{1}{\sqrt{w_i}} \cdot v_i^\otimes 3 \).

Applying tensor power iteration allows the recovery of the \( u_i = \sqrt{w_i} Wv_i \) and the weights \( \frac{1}{\sqrt{w_i}} \), from which the \( v_i \) are recoverable.

The theorem Theorem B.3 is actually the consequence of Theorem C.1 and the following proposition, which controls the error propagation in the whitening step.

**Proposition C.2** (Consequence of Lemma 12 of \cite{HK13}). Let \( M_2 = \sum_{i \in [k]} \lambda_i \cdot u_i u_i^T \) be a rank-\( k \) PSD matrix, and let \( \hat{M} \) be a symmetric matrix whose top \( k \) eigenvalues are positive. Let \( T = \sum_{i \in [k]} \lambda_i \cdot u_i^\otimes 3 \), and let \( \hat{T} = T + E_T \) where \( E_T \) is a symmetric tensor with \( \|E_T\|_F \leq \gamma \).
Suppose \( \| M_2 - \hat{M} \|_F \leq \varepsilon \sigma_k(M_2) \), where \( \sigma_k(M) \) is the \( k \)th eigenvalue of \( M_2 \). Let \( U \) be the square root of the pseudoinverse of \( M_2 \), and let \( \hat{U} \) be the square root of the pseudoinverse of the projection of \( \hat{M} \) to its top \( k \) eigenvectors. Then

\[
\| T(U, U, U) - \hat{T}(\hat{U}, \hat{U}, \hat{U}) \| \leq \frac{6}{\sqrt{\lambda_{\min}}} \varepsilon + \gamma \cdot \| \hat{U} \|_F^2 \| \hat{U} \|_F
\]

Proof. We use the following fact, which is given as Lemma 12 in [HK13].

\[
\| T(U, U, U) - \hat{T}(\hat{U}, \hat{U}, \hat{U}) \| \leq \frac{6 \varepsilon}{\sqrt{\lambda_{\min}}} + \| E_T(\hat{U}, \hat{U}, \hat{U}) \|_2.
\]

The proof of this fact is straightforward, but requires a great deal of bookkeeping; we refer the reader to [HK13].

It remains to bound \( \| E_T(\hat{U}, \hat{U}, \hat{U}) \|_2 \). Some straightforward calculations yield the desired bound,

\[
\| E(\hat{U}, \hat{U}, \hat{U}) \|_2 \leq \sum_i \| (\hat{U} e_i) \otimes \hat{U}^T E_i \| \leq \sum_i \| \hat{U} e_i \|_2 \| \hat{U}^T E_i \| \leq \| \hat{U} \|_2 \cdot \sum_i \| \hat{U} e_i \|_2 \| E_i \|_F \leq \| \hat{U} \|_2 \cdot \sum_i \| E_i \|_F \leq \| \hat{U} \|_F \cdot \| E \|_F,
\]

where we have applied the triangle inequality, the behavior of the spectral norm under tensoring, the submultiplicativity of the norm, and Cauchy-Schwarz.

We now prove Theorem B.3.

Proof of Theorem B.3. Let \( \hat{U} \) be the square root of the projection of \( \hat{M} \) to its top \( k \) eigenvectors. Note that \( \| \hat{U} \| \leq \sigma_k(M_2)^{-1/2} \), \( \| U \|_F \leq \sqrt{k} \sigma_k(M_2)^{-1/2} \), and thus by Proposition C.2, the error \( E \) in Theorem C.1 satisfies

\[
2\beta := \| E \|_F \leq \frac{6 \| E_M \|_F}{\sigma_k(M_2) \sqrt{\lambda_{\min}}} + \frac{\| E_T \|_F \sqrt{k}}{\sigma_k(M_2)^{3/2}}.
\]

Suppose \( 1/40 \geq 2\beta \geq \| E \|_F \). Applying Proposition C.2, we obtain vectors \( u_1, \ldots, u_k \) and scaling factors \( \lambda_1, \ldots, \lambda_k \) such that \( \| u_i - \sqrt{w_i} \cdot M^{-1/2} v_i \| \leq 16 \beta \cdot \sqrt{w_i} \) and \( \left| \frac{1}{\sqrt{w_i}} - \lambda_i \right| \leq 5 \beta \cdot \lambda_i \). The \( w_i \) are now recovered by taking the inverse square of the \( \lambda_i \), so we have that when \( 10\beta < \frac{1}{4} \leq \frac{1}{4} \lambda_i \),

\[
| \hat{w}_i - w_i | = \left| \frac{1}{\lambda_i^2} - w_i \right| \leq \frac{1}{\lambda_i^2} \left| \frac{1}{(\lambda_i \pm 10\beta)^2} \right| \leq 5 \beta \cdot \frac{2 \lambda_i - 10\beta}{\lambda_i^2 (\lambda_i - 10\beta)^2} \leq 40 \beta,
\]

where to obtain the second inequality we have taken a Taylor expansion, and in the final inequality we have used the fact that \( 10\beta < \frac{1}{4} \leq \frac{1}{4} \lambda_i \).

We now recover \( v_i \) by taking \( \hat{v}_i = \lambda_i \cdot \hat{U} u_i \), so we have

\[
\| \hat{v}_i - v_i \| \leq \| \lambda_i \cdot \hat{U} \sqrt{w_i} \cdot M^{-1/2} v_i - v_i \| + \| \lambda_i \cdot \hat{U} (u_i - \sqrt{w_i} M^{-1/2} v_i) \|
\leq (\lambda_i \cdot \sqrt{w_i}) \| (\hat{U} \cdot M^{-1/2} - I) v_i \| + \| (\hat{U} - \lambda_i \sqrt{w_i}) \| v_i \| + \| \hat{U} \| \cdot 16 \beta \lambda_i \sqrt{w_i}
\leq (1 + 10\beta) \| \hat{U} \cdot M^{-1/2} - I \| + 10\beta + \| \hat{U} \| \cdot 16 \beta (1 + 10\beta)
\]
\[ \leq (1 + 10\beta)\|\hat{U} \cdot M^{-1/2} - I\| + 10\beta + \|\hat{U}\| \cdot 16\beta(1 + 10\beta). \]

It now suffices to bound \(\|\hat{U} M^{-1/2} - I\|\), for which it in turn suffices to bound \(\|M^{-1/2}\hat{U} \hat{U} M^{-1/2} - I\|\), since the eigenvalues of \(A A^T\) are the square eigenvalues of \(A\). Consider \(\|(M^{-1/2}\Pi_k(M + E_M)\Pi_k)M^{-1/2} - I\|\), where \(\Pi_k\) is the projector to the top \(k\) eigenvectors of \(M\). Because both matrices are PSD, finally this reduces to bounding \(\|M - \Pi_k(M + E_M)\Pi_k\|\). Since \(M\) is rank \(k\), we have that \(\|M - \Pi_k(M + E_M)\Pi_k\| = \sigma_{k+1}(E_M) \leq \|E_M\|\).

Thus, taking loose bounds, we have

\[ \|v_i - \hat{v}_i\| \leq \|E_M\|^{1/2} + 60\beta \cdot \|M_2\|^{1/2} + 10\beta, \]

as desired. \(\square\)