Overcomplete Order-3 Tensor Decomposition, Blind Deconvolution, and Gaussian Mixture Models

By

HAOLIN CHEN
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Luis Rademacher

Jesus De Loera

Miles Lopes

Committee in Charge

2022

To my mother and father.

Contents

Overcomplete Order-3 Tensor Decomposition, Blind Deconvolution, and Gaussian Mixture Models

Abstract

We propose a new algorithm for tensor decomposition, based on the simultaneous diagonalization algorithm, and apply our new algorithmic ideas to blind deconvolution and Gaussian mixture models. Our first contribution is a simple and efficient algorithm to decompose certain symmetric overcomplete order-3 tensors, that is, three dimensional arrays of the form $\boldsymbol{\mathcal{T}} = \sum_{i=1}^{n} \mathbf{a}_i \otimes \mathbf{a}_i \otimes \mathbf{a}_i$ where the $\mathbf{a}_i$s are not linearly independent. Our algorithm comes with a detailed robustness analysis. Our second contribution builds on top of our tensor decomposition algorithm to expand the family of Gaussian mixture models whose parameters can be estimated efficiently. These ideas are also presented in a more general framework of blind deconvolution that makes them applicable to mixture models of identical but very general distributions, including all centrally symmetric distributions with finite 6th moment.

Acknowledgments

I want to first thank my advisor, Professor Luis Rademacher for his guidance, wisdom, and support over the last three and half a years. He is such a great advisor that I would never be able to finish my Ph.D. without him. His patience in advising and diligence in working made me who I am. It is always very insightful and valuable to discuss research topics with him for his deep and versatile knowledge in different areas. I appreciate his generosity on the summer funding so that I can focus on our work. I will always remember the years we worked together and I can never repay the time and efforts he spent on me.

I want to also thank Professor Jesús De Loera and Professor Naoki Saito, for organizing various academic events and discussions, from which I learned a lot in different areas, e.g. optimization and optimal transport theories. I also want to thank Professors Jesús De Loera, Naoki Saito, Nina Amenta, and James Sharpnack for being in my qualifying exam committee and providing helpful discussions on the work presented in this dissertation. I want to thank Professors Jesús De Loera and Miles Lopes for being in my dissertation committee. This work is supported by National Science Foundation Grants CCF-1657939, CCF-1422830, CCF-2006994, and CCF-1934568.

My Ph.D. fellows, colleagues, and friends, I have my deepest appreciation for the inspiring or frustrating days and nights we spent together. I want to thank Zhenyang Zhang, for being my roommate for four years. We share similar values, interests, and personalities. It is always delightful talking to him and I can learn from him in every way. I am fortunate to know Qianhui Wan, Chang Shu, Austin Tran, and Dong Min Roh while in Davis. I am grateful that we share many moments in or outside research together. Many thanks to my lifelong friends, Shichen Tang, Shaoyu Huang, Haohan Wang, Wen Yan, Kangtao Wang, and Chenzhe Feng, for supporting me even on most frustrating days though we are far apart over the world. I wish the best for all of you.

Finally, I want to express my gratitude to my parents. It is your indispensable and invaluable love that brought me here. Thank you for your constant care and love, even if I am on the other side of the planet. Thank you for being good listeners for things you may not hear of or understand. And thank you for understanding and supporting all the choices and steps I take in my life so far.

CHAPTER 1

Introduction

Tensor structures arise naturally in machine learning and data sciences. They can be seen as multi-dimensional generalizations of vectors and matrices: An order-1 tensor is a vector and an order-2 tensor is a matrix, while an order-n tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$ is an n-mode array where d_i is the dimensionality of mode i.

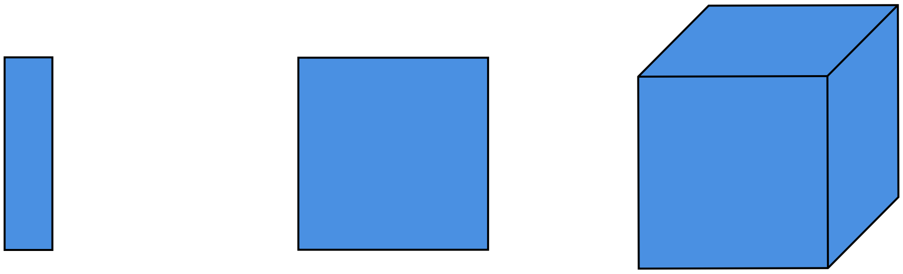


FIGURE 1.1. A vector, a matrix and an order-3 tensor.

There are two key concepts in machine learning: The data to be studied and the model to characterize the data. Both data and machine learning models can be represented using tensors. A greyscale image can be stored using a matrix, hence images with multiple color channels can be stored with order-3 tensors. Order-4 tensors can also be used to store time-dependent data: In medical imaging, 3-dimensional functional magnetic resonance (3D fMRI) images are represented using order-3 tensors and 3D fMRI scans over time can be stored with order-4 tensors. Beyond raw data, higher order moments of random vectors are naturally tensors. Therefore tensors are used to characterize many data models, e.g., exchangeable single topic models, Gaussian mixtures, or hidden Markov models, to name a few. When viewed as multi-linear functions, tensors are also used to represent deep learning models, e.g., multichannel convolutional kernels in convolutional neural networks. However, as the order of the tensor increases, it becomes less interpretable since each entry corresponds to more variables, and more importantly, it suffers from the curse of dimensionality since the number of entries grows exponentially with the order. In the matrix case, we have various decompositions to compress the number of entries and to give interpretable representations. For instance, there are the QR decomposition, the singular value decomposition, and the CUR decomposition [**MD09**]. Therefore it is natural to ask a similar question for tensors:

Can we find efficient and interpretable decompositions of tensors?

There have been many types of tensor decompositions proposed. The idea of tensor decomposition dated back to [**Hit27**] and was further developed as the CANDECOMP/PARAFAC (CP) decomposition in [**CC70**, **Har70**]. In parallel, the Tucker decomposition was proposed in [**Tuc66**]. Recently, the Tensor-Train (TT) decomposition [**Ose11**] was proposed. We will give a short introduction to them in Section 2.3. Tensor decomposition initially found its application in multi-linear models in chemometrics and psychometrics [**LR92**, **Bro97**, **AB03**, **Bro06**, **Hen94**, **KVM01**]. And in recent years the study and application of tensor decomposition have spread to many different areas: signal processing [**DLDM96**, **Com02**, **DC07**, **DLCC07**, **DLdB08**], numerical linear algebra [**LRA93**, **Kol01**, **DLDMV00**, **OST08**, **ZG01**, **WTSA15**, **Sai16**, **SWZ19**], optimization [**LT17**, **GM17**, **FG22**, **KKP21**, **RMC21**], data mining and recommendation system [**MMD06**, **XCH**$^{+}$**10**, **TMBY17**, **WMG19**, **BAH19**, **ZCW20**], neuroscience and medicine [**BS05**, **MHH**$^{+}$**06**, **DVD**$^{+}$**07**, **ABB**$^{+}$**07**, **HVB**$^{+}$**16**, **SBDS**$^{+}$**19**], computer vision [**VT02**, **VBPP06**, **WA03**, **WA04**, **SH05**, **HSK13**, **WPSZ18**], machine learning [**HK13**, **ABG**$^{+}$**14**, **GVX14**, **NROV14**, **AGH**$^{+}$**14**, **AGJ15**, **GHK15**] and deep learning [**NPOV15**, **GPNV16**, **LGR**$^{+}$**15**, **KPY**$^{+}$**16**, **YH17**, **TSN18**, **MZZ**$^{+}$**19**].

Among various applications of tensor decompositions, what in particular interests us is the learning of latent variable models. Latent variable models are statistical models consists of *observable variables* and *hidden variables*. As an introductory example, consider a simple mixture model: Let $X_1, \ldots, X_n$ be a family of random variables. Each observation of the mixture is sampled from X_i with probability w_i. Consider the choice of which random variable to sample as the hidden variable H, a discrete random variable such that the probability of $H = i$ equals w_i. The learning task is to recover the distribution of H (and sometimes the distribution of each X_i) when only the samples from the mixture are observable. We will review the latent variable models of our interests in Chapter 4. In recent years, tensor methods are developed to give consistent and efficient estimators of latent variable models [**HK13**, **GVX14**, **BCMV14**, **ABG**$^{+}$**14**, **AGH**$^{+}$**14**, **AGJ15**, **GHK15**]. They are based on simple intuitions from the method of moments [**Pea94**]: (1) compute some statistics, often sample moments from the data, and (2) estimate the model parameters that produce (almost) the same corresponding population statistics. Following a similar intuition, we have the following question:

Can we expand the family of efficiently-learnable latent variable models via tensor methods?

This dissertation is based on [**CR22**] and aims to answer the above two questions in part by expanding the family of efficiently decomposable tensors, and showing two direct applications, *blind deconvolution* and *Gaussian mixture model (GMM) parameter estimation*, of such decomposition in latent variable model learning.

Overview of Chapter 2

In Chapter 2, we introduce some of the most important concepts studied in this dissertation as well as notation used throughout the dissertation. We will provide introductions to tensors, cumulants, and mathematical tools used in our analysis.

Overview of Chapter 3

In Chapter 3, we study the overcomplete tensor decomposition problem. It is well-known that most tensor problems, including the decomposition, are NP-hard in the worst case [**Hås90**, **HL13**]. Hence we focus on a specific case that is previously unsolved: Given a symmetric order-3 tensor $\boldsymbol{\mathcal{T}} = \sum_{i\in[n]} \mathbf{a}_i^{\otimes 3}$ that is *overcomplete*, i.e., its components are not linearly independent, we measure the overcompleteness by writing $n = r + k$, where any subset of r $\mathbf{a}_i$s is linearly independent. When k is small, we give an intuitively simple algorithm (Algorithm 2) with running time polynomially in r and exponentially in k to decompose the tensor, even when the input tensor $\tilde{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}}_{in}$ of the algorithm contains some noise $\boldsymbol{\mathcal{E}}_{in}$. Our randomized algorithm extends the idea of *the simultaneous diagonalization algorithm* [**LRA93**]. Two backbones of our proof are:

- The robust analysis of the simultaneous diagonalization algorithm (Theorem 3.3.3) guarantees the correctness of the decomposition;
- The analysis of random contraction (Section 3.3.3) based on the Gaussian correlation inequality [**LM17**, **Roy14**] ensures that with non-negligible probability, a single loop of

Algorithm 2 will satisfy Theorem 3.3.3 and outputs the desired decomposition, hence ensures the polynomial running time.

Overview of Chapter 4

In Chapter 4, we study one application of tensor decomposition: learning a family of latent variable model of the form

$$\mathbf{x} = \mathbf{z} + \boldsymbol{\eta} \tag{1.0.1}$$

using samples from $\mathbf{x}$, where $\mathbf{z}$ and $\boldsymbol{\eta}$ are independent random vectors. The random vector $\mathbf{z}$ follows a discrete distribution that can be seen as the latent variable and $\boldsymbol{\eta}$ is the additive noise which has zero 1st and 3rd moments, and finite 6th moment. In particular, the family of distributions of this form contains the Gaussian mixtures with $\boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{\Sigma})$, therefore we also solved one GMM estimation problem. We show that the 3rd cumulant of such $\mathbf{x}$ has a nice tensor structure and thus our algorithm applies when a natural non-degenerate condition is satisfied, and then the distribution of $\mathbf{z}$ can be efficiently recovered. Furthermore, if $\boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{\Sigma})$, the distribution of $\boldsymbol{\eta}$ can be recovered as well.

CHAPTER 2

Notation and Preliminaries

In this chapter, we will cover notations, preliminaries and background knowledge used in the dissertation.

2.1. Notation

For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. Throughout the dissertation, lower case bold letters (e.g., $\mathbf{a}, \mathbf{b}, \mathbf{c}$) are reserved for vectors, upper case bold letters (e.g., $\mathbf{A}, \mathbf{B}, \mathbf{C}$) are reserved for matrices, and upper case bold calligraphic letters (e.g., $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}$) are reserved for tensors.

We use $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote the Gaussian distribution with center $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Other plain calligraphic letters (e.g., $\mathcal{A}, \mathcal{B}, \mathcal{C}$), are reserved for random events in the analysis of Section 3.3.3. We use $\mathbb{P}[\cdot]$ to denote the probability of a event, $\mathbb{E}[\cdot]$ to denote the expectation of a random variable, and $\mathrm{Var}(\cdot), \mathrm{Cov}(\cdot)$ to denote the variance of a random variable and covariance of a random vector, respectively.

For clarity of exposition we analyze our algorithms in a computational model where we assume arithmetic operations between real numbers take constant time. We use the notation $\mathrm{poly}(\cdot)$ to denote a fixed polynomial that is non-decreasing in every argument. We use $O(\cdot), \Omega(\cdot)$ and $\Theta(\cdot)$ to describe asymptotic behavior of functions: for $f, g : \mathbb{R} \to \mathbb{R}$, we say $f(x) = O(g(x))$ if there exists $x_0, M > 0$ such that $|f(x)| \leq Mg(x)$ for every $x \geq x_0$, $f(x) = \Omega(g(x))$ if there exists $x_0', m > 0$ such that $|f(x)| \geq mg(x)$ for every $x \geq x_0'$, and $f(x) = \Theta(g(x))$ if $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$.

For a vector $\mathbf{a} \in \mathbb{R}^d$, its i-th entry is denoted by $(\mathbf{a})_i$. For a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$, the (i, j)-th entry is denoted by $(\mathbf{A})_{ij}$. We denote by $\sigma_i(\mathbf{A})$ its i-th largest singular value, by $\mathbf{A}^\dagger$ its Moore-Penrose pseudoinverse, and by $\kappa(\mathbf{A}) = \sigma_1(\mathbf{A})/\sigma_{\min(d,n)}(\mathbf{A})$ its condition number. Let $\mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{dn}$ denote the vector obtained by stacking all columns of $\mathbf{A}$. Denote by $\mathrm{diag}(\mathbf{a})$ the diagonal matrix with diagonal entries from $\mathbf{a}$, where $\mathbf{a}$ is a (column) vector, or simply $\mathrm{diag}(a_i)$ when a_is are from a finite set $\{a_i : i \in [n]\}$. Let $\|\cdot\|_2$ denote the spectral norm of a matrix, or the Euclidean norm of a vector and $\|\cdot\|_{\mathrm{F}}$, the Frobenius norm of a matrix.

In $\mathbb{R}^d$, we denote by $\langle \mathbf{a}, \mathbf{b} \rangle$ the inner product of two vectors $\mathbf{a}, \mathbf{b}$, and by $\otimes$ the tensor product: For two vectors $\mathbf{a} \otimes \mathbf{b}$, $(\mathbf{a} \otimes \mathbf{b})_{ij} := (\mathbf{a})_i (\mathbf{b})_j$. Let $\hat{\mathbf{a}} = \mathbf{a}/\|\mathbf{a}\|_2$. For a set of vectors $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$, we denote their linear span by $\operatorname{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$. We use $[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ to denote the matrix containing $\mathbf{a}_i$s as columns. If $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$, we have $\hat{\mathbf{A}} = [\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \ldots, \hat{\mathbf{a}}_n]$ and $\tilde{\mathbf{A}}$ follows a similar definition, besides we denote by $\mathbf{A}_{:m} \in \mathbb{R}^{d \times m}$ the matrix $[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$ for some $m < n$ and by $\mathbf{A}_{m+1:} \in \mathbb{R}^{d \times (n-m)}$ the matrix $[\mathbf{a}_{m+1}, \ldots, \mathbf{a}_n]$. We say the matrix $\mathbf{A}$ is ρ-bounded if $\max_{i \in [n]} \|\mathbf{a}_i\|_2 \leq \rho$. We denote by $\mathbf{I}_n$ the identity matrix in $\mathbb{R}^{n \times n}$. Given a vector $\mathbf{a} \in \mathbb{R}^d$ or a diagonal matrix $\mathbf{D} \in \mathbb{R}^{d \times d}$, for $r \in \mathbb{R}$, notations $\mathbf{a}^r$ and $\mathbf{D}^r$ are used for entry-wise power.

2.2. Singular Value Decomposition

The theory of matrix decomposition is well-developed [**Ste98**, **Ste01**, **GVL13**, **Hog13**]. It is well known that each matrix $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$ has a singular value decomposition (SVD):

$$\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^\top = \sum_{i \in [\min\{d_1, d_2\}]} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top,$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{d_1}] \in \mathbb{R}^{d_1 \times d_1}$ and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{d_2}] \in \mathbb{R}^{d_2 \times d_2}$ are orthogonal matrices, $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_i)$, and $\sigma_1 \geq \cdots \geq \sigma_{\min\{d_1, d_2\}} \geq 0$.

SVD is such a power tool that the reader can see from Section 2.3 that it also works as the backbone of a well-known tensor decomposition algorithm: *the simultaneous diagonalization* algorithm. However in real world problems, input data matrix or tensor can always be perturbed or with noise. It is crucial to measure the difference between the SVD of a matrix $\mathbf{A}$ and that of the perturbed version, $\mathbf{A} + \mathbf{E}$. Hence we state Wedin's theorem, a "$\sin(\theta)$ theorem" for perturbed singular vectors as well as Weyl's inequality for SVD in this section. The following results are from [**Ste90**, **Ste91**].

Theorem 2.2.1 (Weyl's inequality). *Let $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{d_1 \times d_2}$ with $d_1 \geq d_2$. Denote the singular values in non-increasing order of $\mathbf{A}$ and $\mathbf{A} + \mathbf{E}$ by σ_i and $\tilde{\sigma}_i$, respectively. Then $|\sigma_i - \tilde{\sigma}_i| \leq \|\mathbf{E}\|_2$.*

Theorem 2.2.2 (Wedin). *With the notation from Theorem 2.2.1, let a singular value decomposition of* $\mathbf{A}$ *be:* $[\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]^\top \mathbf{A}[\mathbf{V}_1, \mathbf{V}_2] = \begin{bmatrix} \mathbf{\Sigma}_1 & 0 \\ 0 & \mathbf{\Sigma}_2 \\ 0 & 0 \end{bmatrix}$*, where the singular values can be in arbitrary order.*

Let the perturbed version be: $[\tilde{\mathbf{U}}_1, \tilde{\mathbf{U}}_2, \tilde{\mathbf{U}}_3]^\top (\mathbf{A}+\mathbf{E})[\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_2] = \begin{bmatrix} \tilde{\mathbf{\Sigma}}_1 & 0 \\ 0 & \tilde{\mathbf{\Sigma}}_2 \\ 0 & 0 \end{bmatrix}$*. Let* $\mathbf{\Phi}$ *be the matrix of canonical angles between the column spaces of* $\mathbf{U}_1$ *and* $\tilde{\mathbf{U}}_1$*, and* $\mathbf{\Theta}$ *be that of* $\mathbf{V}_1$ *and* $\tilde{\mathbf{V}}_1$*, respectively. Let* $\delta = \min\{\min_i (\tilde{\mathbf{\Sigma}}_1)_{ii}, \min_{i,j} |(\tilde{\mathbf{\Sigma}}_1)_{ii} - (\mathbf{\Sigma}_2)_{jj}|\}$*. Then* $\sqrt{\|\sin \mathbf{\Phi}\|_2^2 + \|\sin \mathbf{\Theta}\|_2^2} \le \sqrt{2}\|\mathbf{E}\|_2/\delta$*.*

2.3. Tensors

In this section, we aim to provide a short introduction to tensors. Tensors are the generalization of matrices and can be seen as multi-dimensional arrays. We use $\boldsymbol{\mathcal{T}}[i_1, \dots, i_n]$, or $(\boldsymbol{\mathcal{T}})_{i_1 \dots i_n}$ to denote the $(i_1, i_2, \dots, i_n)$-th entry of $\boldsymbol{\mathcal{T}}$ for the multi-index $(i_1, i_2, \dots, i_n) \in [d_1] \times [d_2] \times \cdots \times [d_n]$, and $\boldsymbol{\mathcal{T}}[i]$ to denote the i-th subtensor of shape $[d_2] \times \cdots \times [d_n]$ by fixing the first index. We say $\boldsymbol{\mathcal{T}}$ is symmetric if $d = d_1 = \cdots = d_n$ and every entry $(\boldsymbol{\mathcal{T}})_{i_1 \dots i_n}$ is invariant under permutation of indices. Since we will be mainly dealing with symmetric order-3 tensors in Chapters 3 and 4, in the rest of this section we will use order-3 tensors for simplicity, while most definitions can be generalized to higher order cases.

Definition 2.3.1 (Mode-wise product). *The mode-1 product of an order-3 tensor* $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ *and a matrix* $\mathbf{U} \in \mathbb{R}^{d_1' \times d_1}$ *is defined by*

$$\boldsymbol{\mathcal{T}}' = \boldsymbol{\mathcal{T}}(\mathbf{U}, \cdot, \cdot) \Longleftrightarrow \boldsymbol{\mathcal{T}}'[j, i_2, i_3] = \sum_{i_1 \in [d_1]} (\boldsymbol{\mathcal{T}})_{i_1 i_2 i_3} (\mathbf{U})_{j i_1}, \tag{2.3.1}$$

where $\boldsymbol{\mathcal{T}}'$ *becomes an order-3 tensor in* $\mathbb{R}^{d_1' \times d_2 \times d_3}$*. We have similar definitions of product on other modes: for* $\mathbf{U}^{(1)} \in \mathbb{R}^{d_1' \times d_1}, \mathbf{U}^{(2)} \in \mathbb{R}^{d_2' \times d_2}, \mathbf{U}^{(3)} \in \mathbb{R}^{d_3' \times d_3}$*, we have that:*

(2.3.2)

$$\boldsymbol{\mathcal{T}}' = \boldsymbol{\mathcal{T}}(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) \Longleftrightarrow \boldsymbol{\mathcal{T}}'[j_1, j_2, j_3] = \sum_{i_1 \in [d_1]} \sum_{i_2 \in [d_2]} \sum_{i_3 \in [d_3]} (\boldsymbol{\mathcal{T}})_{i_1 i_2 i_3} (\mathbf{U}^{(1)})_{j_1 i_1} (\mathbf{U}^{(2)})_{j_2 i_2} (\mathbf{U}^{(3)})_{j_3 i_3}.$$

Example 2.3.2. *Let* $\mathcal{T} \in \mathbb{R}^{2\times 2\times 2}$ *be a tensor with its matrix slices in the first index given by:*

$$\mathcal{T}[1] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \mathcal{T}[2] = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}.$$

And let $\mathbf{U} \in \mathbb{R}^{3\times 2}$ *be*

$$\mathbf{U} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Then the mode-1 product $\mathcal{T}(\mathbf{U}, \cdot, \cdot)$ *gives a tensor* $\mathcal{T}' \in \mathbb{R}^{3\times 2\times 2}$ *with slices*

$$\mathcal{T}'[1] = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}, \quad \mathcal{T}'[2] = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}, \quad \mathcal{T}'[3] = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}.$$

Furthermore, $\mathcal{T}(\mathbf{U}, \mathbf{U}, \mathbf{U})$ *gives a tensor* $\mathcal{T}'' \in \mathbb{R}^{3\times 3\times 3}$ *with slices in the first index:*

$$\mathcal{T}''[1] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathcal{T}''[2] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad \mathcal{T}''[3] = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 0 & 2 \\ -1 & 1 & 8 \end{bmatrix}.$$

Specifically, when $d_1' = 1$, i.e., for $\mathbf{u} \in \mathbb{R}^{d_1}$, $\mathcal{T}(\mathbf{u}, \cdot, \cdot)$ becomes a matrix in $\mathbb{R}^{d_2 \times d_3}$. In such cases we call this product contraction. In particular, for $\mathbf{u} \in \mathbb{R}^d$ and symmetric order-3 tensor $\mathcal{T}$ we shorthand the contraction $\mathcal{T}(\mathbf{u}, \cdot, \cdot)$ by $\mathbf{T_u}$. For instance, let $\mathbf{u} = [-1, 1]$, then for $\mathcal{T}$ in the example above, $\mathbf{T_u} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$.

Definition 2.3.3 (Rank-1 tensor). *A order-3 tensor* $\mathcal{T}$ *is rank-1 if it can be written as the tensor product of* 3 *vectors:*

$$\mathcal{T} = \mathbf{a}_1 \otimes \mathbf{a}_2 \otimes \mathbf{a}_3 \iff \mathcal{T}[i_1, i_2, i_3] = (\mathbf{a}_1)_{i_1} (\mathbf{a}_2)_{i_2} (\mathbf{a}_3)_{i_3}. \tag{2.3.3}$$

Specifically, a symmetric $\mathcal{T}$ *is rank-1 if* $\mathcal{T} = \mathbf{a} \otimes \mathbf{a} \otimes \mathbf{a} = \mathbf{a}^{\otimes 3}$.

Given an order-3 tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, the CP decomposition [**CC70**, **Har70**] tries to find a minimal set of vectors: $\{\mathbf{a}_i \in \mathbb{R}^{d_1}, \mathbf{b}_i \in \mathbb{R}^{d_2}, \mathbf{c}_i \in \mathbb{R}^{d_3} : i \in [n]\}$ such that $\mathcal{T}$ can be written as the sum of rank-1 tensors obtained from tensor products of these vectors: $\mathcal{T} = \sum_{i \in [n]} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$.

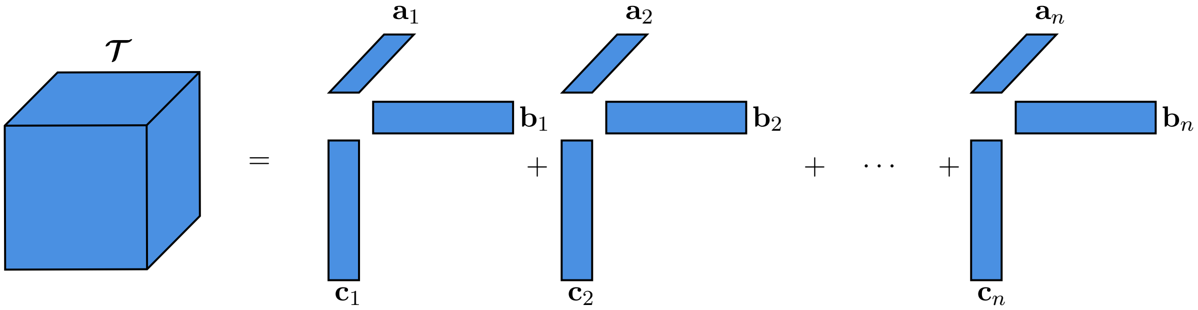


FIGURE 2.1. CP decomposition.

The Tucker decomposition [**Tuc66**] finds a core tensor $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and linear transformations $\mathbf{U}_1 \in \mathbb{R}^{d_1 \times n_1}, \mathbf{U}_2 \in \mathbb{R}^{d_2 \times n_2}, \mathbf{U}_3 \in \mathbb{R}^{d_3 \times n_3}$ such that applying the linear transformations on each mode of $\mathcal{G}$ yields $\mathcal{T}$: $\mathcal{T} = \mathcal{G}(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3)$. When $\mathcal{G}$ is diagonal, i.e., $\mathcal{G}$ is non-zero only when all indices coincide, it reduces to the CP decomposition. Hence the Tucker decomposition can be seen as a generalization of the CP decomposition.

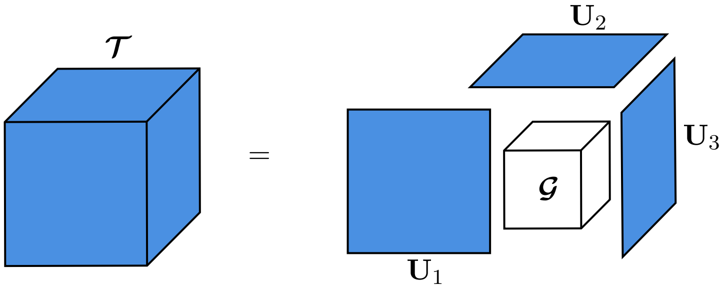


FIGURE 2.2. Tucker decomposition.

The recently developed Tensor-Train (TT) decomposition [**Ose11**] attempts to compress the tensor by parameter sharing between entries: it finds a set of vectors and matrices $\{\mathbf{a}_i \in \mathbb{R}^{n_1}, \mathbf{M}_j \in \mathbb{R}^{n_1 \times n_2}, \mathbf{b}_k \in \mathbb{R}^{n_2} : i \in [d_1], j \in [d_2], k \in [d_3]\}$ such that $\mathcal{T}[i,j,k] = \mathbf{a}_i^\top \mathbf{M}_j \mathbf{b}_k$.

In this dissertation, we are interested in the *symmetric decomposition*, which is a special case of the CP decomposition such that all rank-one components are symmetric. To characterize it, we give the definition of the *symmetric rank*:

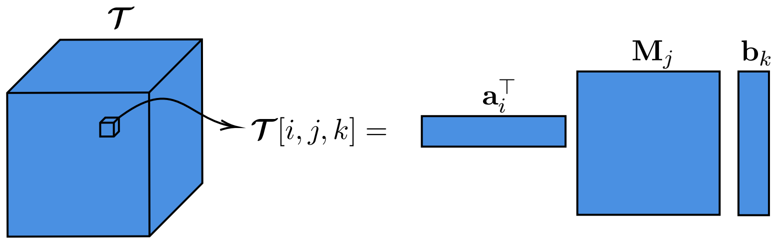


FIGURE 2.3. Tensor-Train decomposition.

Definition 2.3.4 (Symmetric rank). *Let $\mathcal{T} \in \mathbb{R}^{d\times d\times d}$ be symmetric. The symmetric rank of $\mathcal{T}$, or simply the rank of $\mathcal{T}$ is the minimal $n \in \mathbb{N}$ such that*

$$\mathcal{T} = \sum_{i\in[n]} \mathbf{a}_i^{\otimes 3} \tag{2.3.4}$$

for a set of vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

Remark 1. *Note that there are different notions of tensor rank [**KB09**]: E.g., CP rank, which is the minimal number of components in a CP decomposition, and Tucker rank or multi-linear rank, which is the tuple of column ranks of matrix flattenings of a tensor along different modes. The CP rank and symmetric rank may not coincide for a symmetric tensor [**Shi18**]. However, in this dissertation we study only symmetric tensors with the notion of the symmetric rank. In fact, Kruskal's theorem (Theorem 2.3.6) guarantees that these two ranks coincide when the theorem holds true for a symmetric tensor, therefore we do not distinguish them throughout the dissertation.*

For a rank n symmetric order-3 tensor $\mathcal{T} = \sum_{i\in[n]} \mathbf{a}_i^{\otimes 3}$, we say the tensor $\mathcal{T}$ is ρ-bounded if $\max_{i\in[n]} \|\mathbf{a}_i\|_2 \le \rho$.

The decomposition problem of symmetric tensor is stated as follows: Given a tensor $\mathcal{T} = \sum_{i\in[n]} \mathbf{a}_i^{\otimes 3}$ of rank n, recover the component vectors $\mathbf{a}_i \in \mathbb{R}^d$. [**Kru77**] gives the classical result on uniqueness of such decomposition, of which the version with robustness analysis (Theorem 3.3.1 and Corollary 3.3.2) will be the backbone of our theoretical analysis. We first provide the definition of Kruskal rank and state the case of symmetric tensors below.

Definition 2.3.5 (Kruskal rank). *Let* $\mathbf{A} \in \mathbb{R}^{d\times n}$*. The* Kruskal rank *of* $\mathbf{A}$*, denoted by* K-rank($\mathbf{A}$)*, is the maximum $k \in [n]$ such that any k columns of $\mathbf{A}$ are linearly independent.*

Theorem 2.3.6 ([Kru77]). *Let $\mathcal{T}$ be of rank n, and a length n decomposition of $\mathcal{T}$ is given by $\mathcal{T} = \mathbf{a}_1^{\otimes 3} + \cdots + \mathbf{a}_n^{\otimes 3}$. Suppose* $\text{K-rank}(\mathbf{A}) = R$ *for* $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ *and* $3R \geq 2n + 2$. *Then for any other decomposition* $\mathcal{T} = \sum_{i \in [n]} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i$ *with length n, there exists a permutation matrix* $\mathbf{\Pi}$, *and diagonal matrices* $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3$ *satisfying* $\mathbf{\Lambda}_1 \mathbf{\Lambda}_2 \mathbf{\Lambda}_3 = \mathbf{I}_n$, *such that* $[\mathbf{u}_1, \ldots, \mathbf{u}_n] = \mathbf{A}\mathbf{\Pi}\mathbf{\Lambda}_1, [\mathbf{v}_1, \ldots, \mathbf{v}_n] = \mathbf{A}\mathbf{\Pi}\mathbf{\Lambda}_2, [\mathbf{w}_1, \ldots, \mathbf{w}_n] = \mathbf{A}\mathbf{\Pi}\mathbf{\Lambda}_3$. *That is, the decomposition of $\mathcal{T}$ is unique up to permutation and scaling.*

To generalize the notion of Kruskal rank to quantify the well-conditionness, we give the definition of *robust Kruskal rank.*

Definition 2.3.7 (Robust Kruskal Rank, [BCV14, BCMV14]). *Let* $\mathbf{A} \in \mathbb{R}^{d \times n}$ *and* $\tau > 0$. *The* robust Kruskal rank (with threshold τ) *of* $\mathbf{A}$, *denoted* $\text{K-rank}_\tau(\mathbf{A})$, *is the maximum* $k \in [n]$ *such that for any subset* $S \subseteq [n]$ *of size k we have* $\sigma_k(\mathbf{A}_S) \geq 1/\tau$.

There are two widely-used algorithms for tensor decomposition: *the simultaneous diagonalization algorithm* and *tensor power iteration.* We briefly review them below.

The simultaneous diagonalization algorithm [LRA93]. The basic idea of the simultaneous diagonalization algorithm[1] to decompose a rank-d tensor $\mathcal{T} = \sum_{i \in [d]} \mathbf{a}_i^{\otimes 3}$ with linearly independent components $\mathbf{a}_1, \ldots, \mathbf{a}_d \in \mathbb{R}^d$ is the following: The contraction of $\mathcal{T}$ with $\mathbf{x} \in \mathbb{R}^d$ gives $\mathbf{T}_\mathbf{x} = \sum_{i \in [d]} \langle \mathbf{x}, \mathbf{a}_i \rangle \mathbf{a}_i \mathbf{a}_i^\top = \mathbf{A} \operatorname{diag}(\langle \mathbf{x}, \mathbf{a}_i \rangle) \mathbf{A}^\top$. For random unit vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, compute the (right) eigenvectors of

$$\mathbf{T}_\mathbf{x} \mathbf{T}_\mathbf{y}^{-1} = \mathbf{A} \operatorname{diag}\left(\frac{\langle \mathbf{x}, \mathbf{a}_i \rangle}{\langle \mathbf{y}, \mathbf{a}_i \rangle} \right) \mathbf{A}^{-1}.$$

With probability 1, the set of eigenvectors is equal to the set of directions of $\mathbf{a}_i$s (the eigenvectors recover the $\mathbf{a}_i$s up to sign and norm). We use a version (Algorithm 1) that allows for the number of $\mathbf{a}_i$s to be less than d and that includes an error analysis [**GVX13**, **GVX14**]. The running time of the simultaneous diagonalization algorithm is polynomial in the dimension d, the inverse of component-wise error $1/\varepsilon$, and natural conditioning parameters such as the minimum singular value of $\mathbf{T}_\mathbf{x} \mathbf{T}_\mathbf{y}^{-1}$ [**GVX13**, Section 5.3].

[1] The simultaneous diagonalization algorithm has been erroneously called Jennrich's algorithm, see [**Kol**] for a discussion.

Example 2.3.8. *Consider a tensor* $\boldsymbol{\mathcal{T}} = \mathbf{a}_1^{\otimes 3} + \mathbf{a}_2^{\otimes 3} \in \mathbb{R}^{2\times 2\times 2}$ *given by its matrix slices in the first index:*

$$\boldsymbol{\mathcal{T}}[1] = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad \boldsymbol{\mathcal{T}}[2] = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}.$$

Let $\mathbf{x} = [1, 0]$ *and* $\mathbf{y} = [0, 1]$. *Notice that* $\mathbf{T}_{\mathbf{x}} = \boldsymbol{\mathcal{T}}[1]$ *and* $\mathbf{T}_{\mathbf{y}} = \boldsymbol{\mathcal{T}}[2]$. *We have* $\mathbf{T}_{\mathbf{x}}\mathbf{T}_{\mathbf{y}}^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. *The right eigenvectors of* $\mathbf{T}_{\mathbf{x}}\mathbf{T}_{\mathbf{y}}^{-1}$ *are* $\mathbf{u}_1 = [1/\sqrt{2}, 1/\sqrt{2}]$ *and* $\mathbf{u}_2 = [1/\sqrt{2}, -1/\sqrt{2}]$*, and they are in the direction of* $\mathbf{a}_1, \mathbf{a}_2$. *Next, solving a regression problem:* $\min_{\lambda_1,\lambda_2} \|\boldsymbol{\mathcal{T}} - (\lambda_1 \mathbf{u}_1^{\otimes 3} + \lambda_2 \mathbf{u}_2^{\otimes 3})\|_{\mathrm{F}}$ *recovers* $\mathbf{a}_1 = [1, 1]$ *and* $\mathbf{a}_2 = [1, -1]$.

Tensor power iteration [DLDMV00, JGKA19]. Tensor power iteration is applicable when a rank-n tensor $\boldsymbol{\mathcal{T}} = \sum_{i\in[n]} \lambda_i \mathbf{a}_i^{\otimes 3}$ has orthonormal components $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$. The iteration updates:

$$\mathbf{v} \longleftarrow \frac{\boldsymbol{\mathcal{T}}(\mathbf{v}, \mathbf{v}, \cdot)}{\|\boldsymbol{\mathcal{T}}(\mathbf{v}, \mathbf{v}, \cdot)\|_2}$$

with random initialized $\mathbf{v}$ and $\mathbf{v}$ will converge to one of the $\mathbf{a}_i$s. The rest components can be recovered with the same technique after subtracting $\lambda \mathbf{v}^{\otimes 3}$ from $\boldsymbol{\mathcal{T}}$, where $\lambda = \boldsymbol{\mathcal{T}}(\mathbf{v}, \mathbf{v}, \mathbf{v})$. As an illustrative example, for $\boldsymbol{\mathcal{T}}$ in Example 2.3.8 and $\mathbf{v}$ initialized to be $[1, 2]$, $\mathbf{v}$ becomes close to $[1/\sqrt{2}, 1/\sqrt{2}]$ numerically after 5 iterations. The advantage of power iteration over the simultaneous diagonalization algorithm is that it converges much faster: $\mathbf{v}$ converges to one of the $\mathbf{a}_i$s within ε distance in $O(\log\log(1/\varepsilon))$ iterations, while the disadvantage is that for tensors with non-orthogonal components, a whitening transform is needed to orthogonalize the input [**JGKA19**].

2.4. Cumulants

In this section we provide to the reader a short introduction of cumulants, which will be the main statistical object we study in Chapter 4. We give their definition, properties of interests, and the technical details about the unbiased estimators of cumulants, called k-statistics.

Cumulants. Cumulants are a family of statistical parameters similar to moments. We first give the definition of cumulants of a random variable.

Definition 2.4.1 (Cumulants). *Let X be a random variable. Cumulants of X, denoted by $K_m(X)$ for $m \in \mathbb{N}$, are defined using the coefficients of the Maclaurin series of cumulant generating function:*

$$K(t) = \log \mathbb{E}[\exp(tX)] = \sum_{m=1}^{\infty} \frac{K_m(X)}{m!} X^m, \tag{2.4.1}$$

where $K_m(X) = \left.\frac{\mathrm{d}^m K(t)}{\mathrm{d}t^m}\right|_{t=0}$.

The m-th cumulant $K_m(X)$ is a polynomial in $\mathbb{E}[X], \mathbb{E}[X^2], \ldots, \mathbb{E}[X^m]$. In particular, the first cumulant coincides with the first moment, and the second and third cumulants coincide with the corresponding central moments: $K_1(X) = \mathbb{E}[X]$, $K_2(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathrm{Cov}(X)$, and $K_3(X) = \mathbb{E}[(X - \mathbb{E}[X])^3]$.

Similarly, the cumulants of a random vector $\mathbf{x}$, which are defined using the coefficients of the Maclaurin series of $K(\mathbf{t}) = \log \mathbb{E}[\exp(\mathbf{t}^\top \mathbf{x})]$, are a sequence of tensors related to the moment tensors of $\mathbf{x}$. Similar to the univariate case, we have: $K_1(\mathbf{x}) = \mathbb{E}[\mathbf{x}], K_2(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top] = \mathrm{Cov}(\mathbf{x}), K_3(\mathbf{x}) = \mathbb{E}\big[(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\otimes 3}\big]$. Two properties of cumulants motivated us to study them:

Proposition 2.4.2. *Let X, Y be two independent random variables. Then $K_m(X+Y) = K_m(X) + K_m(Y)$ for all m.*

Proof. It follows from the independence and the definition of cumulant generating function, we have:

$$\begin{aligned} K_{X+Y}(t) &= \log \mathbb{E}[\exp(t(X+Y))] = \log(\mathbb{E}[\exp(tX)]\,\mathbb{E}[\exp(tY)]) \\ &= \log(\mathbb{E}[\exp(tX)]) + \log(\mathbb{E}[\exp(tY)]) = K_X(t) + K_Y(t), \end{aligned}$$

which implies that for all m, $K_m(X+Y) = K_m(X) + K_m(Y)$. □

Proposition 2.4.3. *Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian random vector. Then $K_1(\mathbf{x}) = \boldsymbol{\mu}, K_2(\mathbf{x}) = \boldsymbol{\Sigma}$, and $K_m(\mathbf{x}) = 0$ for $m \geq 3$.*

Proof. It follows from the fact that the cumulant generating function of $\mathbf{x}$ is

$$K(\mathbf{t}) = \boldsymbol{\mu}^\top \mathbf{t} + \frac{1}{2}\mathbf{t}^\top \boldsymbol{\Sigma} \mathbf{t}.$$

□

Besides, all symmetric distributions will have $K_3 = 0$ since K_3 coincides with the 3rd central moment which equals to zero.

Sample cumulants. We now show the reader how to estimate sample cumulants using unbiased estimators called k-statistics. k-statistics are the unbiased estimator for cumulants with the minimum variance, and are long studied in the statistics community. We use such estimators to compute the sample 3rd cumulant in Algorithm 3.

We first provide the formula for the 3rd k-statistic given in [**McC18**, Section 4.2] here:

Proposition 2.4.4 ([McC18]). *Given iid. samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of random vector $\mathbf{x}$, the k-statistic for the 3rd cumulant of $\mathbf{x}$ is: $k_3(r,s,t) = \frac{1}{N}\sum_{i,j,k\in[N]} \phi^{(ijk)}(\mathbf{x}_i)_r(\mathbf{x}_j)_s(\mathbf{x}_k)_t$, where r, s, t are the position indices in the tensor, and $\phi^{(ijk)}$ is the coefficient given by: it is invariant under permutation of indices, and for distinct $i, j, k \in [N]$:*

$$\phi^{(iii)} = \frac{1}{N}, \quad \phi^{(iij)} = -\frac{1}{N-1}, \quad \phi^{(ijk)} = \frac{2}{(N-1)(N-2)}. \tag{2.4.2}$$

Let $\tilde{\boldsymbol{\mathcal{T}}}[r,s,t] = k_3(r,s,t)$ be the 3rd k-statistic tensor of a random vector $\mathbf{x}$. To obtain the entry-wise concentration bound for $\tilde{\boldsymbol{\mathcal{T}}}$, we begin by bounding the variance of each entry in $\tilde{\boldsymbol{\mathcal{T}}}$:

Lemma 2.4.5. *Let $\mathbf{x} = [X_1, \ldots, X_d]$ follow a distribution as in* (1.0.1). *The 3rd sample cumulant $\tilde{\boldsymbol{\mathcal{T}}}$ of $\mathbf{x}$ satisfies:* $\mathrm{Var}\big(\tilde{\boldsymbol{\mathcal{T}}}[r,s,t]\big) = O(\max_{t\in[d]} \mathbb{E}[X_t^6]/N)$.

Proof. This is essentially the version for the 3rd cumulant of [**BRV13**, Lemma 4]. The proof uses the same argument.

We start with defining some multi-indices to simply the computation: let $I = (r,s,t) \in [d]^3$ and $\alpha = (i,j,k) \in [N]^3$, then write $\phi^{(\alpha)}\mathbf{x}_{(\alpha)}^{(I)} = \phi^{(ijk)}(\mathbf{x}_i)_r(\mathbf{x}_j)_s(\mathbf{x}_k)_t$. Also, we define the intersection of 2 multi-indices α, β as: $\alpha \cap \beta = \{i : i \in \alpha \text{ and } i \in \beta\}$. Now we continue to bound the variance of $\tilde{\boldsymbol{\mathcal{T}}}$:

$$
\begin{aligned}
\operatorname{Var}(\tilde{\boldsymbol{\mathcal{T}}}[r,s,t]) &= \mathbb{E}\big[\big(\frac{1}{N}\sum_{\alpha\in[N]^3}\phi^{(\alpha)}\mathbf{x}^{(I)}_{(\alpha)}\big)^2\big] - \mathbb{E}\big[\frac{1}{N}\sum_{\alpha\in[N]^3}\phi^{(\alpha)}\mathbf{x}^{(I)}_{(\alpha)}\big]^2 \\
&= \frac{1}{N^2}\sum_{\alpha,\beta\in[N]^3}\phi^{(\alpha)}\phi^{(\beta)}\,\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}\mathbf{x}^{(I)}_{(\beta)}] - \frac{1}{N^2}\sum_{\alpha,\beta\in[N]^3}\phi^{(\alpha)}\phi^{(\beta)}\,\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}]\,\mathbb{E}[\mathbf{x}^{(I)}_{(\beta)}] \\
&= \frac{1}{N^2}\sum_{\alpha\cap\beta\neq\emptyset}\phi^{(\alpha)}\phi^{(\beta)}\,\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}\mathbf{x}^{(I)}_{(\beta)}] - \frac{1}{N^2}\sum_{\alpha\cap\beta\neq\emptyset}\phi^{(\alpha)}\phi^{(\beta)}\,\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}]\,\mathbb{E}[\mathbf{x}^{(I)}_{(\beta)}] \\
&\leq \frac{1}{N^2}\sum_{\alpha\cap\beta\neq\emptyset}\phi^{(\alpha)}\phi^{(\beta)}\,\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}\mathbf{x}^{(I)}_{(\beta)}].
\end{aligned}
\tag{2.4.3}
$$

Now let us consider how many terms there are indexed by β that does not intersect with α. Let $\operatorname{dist}(\beta)$ be the number of distinct indices in β. Then there are $\binom{N}{\operatorname{dist}(\beta)}$ ways to generate β, out of which there are $\binom{N-3}{\operatorname{dist}(\beta)}$ that will absolutely have no intersection with α as α can have at most 3 different indices. That is, when fixing the number of distinct indices in β, at most a

$$
\left[\binom{N}{\operatorname{dist}(\beta)} - \binom{N-3}{\operatorname{dist}(\beta)}\right] \Big/ \binom{N}{\operatorname{dist}(\beta)}
$$

fraction of index sets in β will intersect with α. To proceed, we need some other estimations on the coefficients and the expectation. By (2.4.2) we have:

$$
|\phi^{(\beta)}| = O(1/N^{\operatorname{dist}(\beta)-1}), \qquad \sum_{\alpha\in[N]^3}|\phi^{(\alpha)}| = O(N).
$$

For $\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}\mathbf{x}^{(I)}_{(\beta)}]$, by Cauchy-Schwarz inequality we have:

$$
\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}\mathbf{x}^{(I)}_{(\beta)}] \leq \max\{\mathbb{E}[(\mathbf{x}^{(I)}_{(\alpha)})^2], \mathbb{E}[(\mathbf{x}^{(I)}_{(\beta)})^2]\}.
$$

Applying Cauchy-Schwarz inequality and Hölder's inequality gives $\mathbb{E}[\mathbf{x}^{(I)}_{(\alpha)}\mathbf{x}^{(I)}_{(\beta)}] \leq \max_{t\in[d]}\mathbb{E}[X_t^6]$.

Then (2.4.3) can be further bounded by

$$\begin{aligned}
&\mathrm{Var}(\tilde{\boldsymbol{\mathcal{T}}}[r,s,t]) \leq \frac{1}{N^2}\sum_{\alpha\cap\beta\neq\emptyset}|\phi^{(\alpha)}\phi^{(\beta)}\,\mathbb{E}[\mathbf{x}_{(\alpha)}^{(I)}\mathbf{x}_{(\beta)}^{(I)}]| \\
&\leq \frac{\max_{t\in[d]}\mathbb{E}[X_t^6]}{N^2}\sum_{\alpha\in[N]^3}|\phi^{(\alpha)}|\sum_{c=1}^{3}\sum_{\mathrm{dist}(\beta)=c,\alpha\cap\beta\neq\emptyset}|\phi^{(\beta)}| \\
&= O\big(\frac{\max_{t\in[d]}\mathbb{E}[X_t^6]}{N^2}\sum_{\alpha\in[N]^3}|\phi^{(\alpha)}|\sum_{c=1}^{3}\frac{\binom{N}{c}-\binom{N-3}{c}}{\binom{N}{c}}N^{1-c}\sum_{\mathrm{dist}(\beta)=c}1\big) \\
&= O\big(\frac{\max_{t\in[d]}\mathbb{E}[X_t^6]}{N^2}\sum_{\alpha\in[N]^3}|\phi^{(\alpha)}|\sum_{c=1}^{3}N^{-1}N^{1-c}N^{c}\big) \\
&= O\big(\frac{\max_{t\in[d]}\mathbb{E}[X_t^6]}{N}\big).
\end{aligned}$$

where the second last equality comes from that $1-\binom{N-3}{c}/\binom{N}{c} = O(N^{-1})$ and $\sum_{\mathrm{dist}(\beta)=c}1 = O(N^c)$. □

Lemma 2.4.6. *Given $\epsilon,\delta\in(0,1)$, the entry-wise error between $\tilde{\boldsymbol{\mathcal{T}}}$ and $\boldsymbol{\mathcal{T}} = K_3(\mathbf{x})$ is at most ϵ with probability at least $1-\delta$ when using $N \geq \Omega\left(\epsilon^{-2}\delta^{-1}\max_{t\in[d]}\mathbb{E}[X_t^6]\right)$ samples.*

Proof. Using Chebyshev's inequality with the variance of $\tilde{\boldsymbol{\mathcal{T}}}$ given in Lemma 2.4.5 yields the result immediately. □

Lemma 2.4.7 (Estimation of the 3rd cumulant). *Let $\boldsymbol{\mathcal{T}},\tilde{\boldsymbol{\mathcal{T}}}$ be the 3rd cumulant of $\mathbf{x} = [X_1,\ldots,X_d]$ and its unbiased estimate (k-statistic) using Proposition 2.4.4, respectively. Given any $\varepsilon,\delta\in(0,1)$, and $N = \Omega(d^9\varepsilon^{-2}\delta^{-1}\max_{i\in[d]}\mathbb{E}[X_i^6]\})$, with probability $1-\delta$ we have $\|\boldsymbol{\mathcal{T}}-\tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} \leq \varepsilon$.*

Proof. Apply Lemma 2.4.6 with accuracy ε/d^3, failure probability δ/d^3 and taking the union bound over d^3 entries, to see that $N = \Omega(d^9\varepsilon^{-2}\delta^{-1}\max_{i\in[d]}\mathbb{E}[X_i^6]\})$ samples are sufficient. □

2.5. Concentration and Anti-concentration Inequalities

The essential tools to analyze our algorithm in a high dimensional setting are the concentration and anti-concentration inequalities. We provide the necessary results used in the dissertation here.

These lemmas work as the fundamental building blocks in the analysis of our randomized algorithm in Section 3.3.3.

Lemma 2.5.1 ([DG03, HK13]). *Suppose $\delta \in (0,1)$, $\mathbf{M} \in \mathbb{R}^{d\times d}$, Q is a finite subset of $\mathbb{R}^d$ and $\mathbf{x}$ is a uniformly random vector in $\mathbb{S}^{d-1}$. Then $\mathbb{P}\left[\min_{\mathbf{q}\in Q}|\langle \mathbf{x}, \mathbf{Mq}\rangle| \geq \frac{\delta \min_{\mathbf{q}\in Q}\|\mathbf{Mq}\|_2}{\sqrt{ed}|Q|}\right] \geq 1-\delta$.*

For the next lemma we need the Gaussian correlation inequality:

Theorem 2.5.2 (Gaussian correlation inequality [LM17, Roy14]). *For any convex centrally symmetric sets K, L in $\mathbb{R}^d$ and any centered Gaussian measure μ on $\mathbb{R}^d$, we have $\mu(K\cap L) \geq \mu(K)\mu(L)$.*

Lemma 2.5.3 ([Kha67, Ši67]). *Let $\mathbf{x} \in \mathbb{R}^d$ be a standard Gaussian random vector, $\mathbf{a}_1, \ldots, \mathbf{a}_k \in \mathbb{S}^{d-1}$, and $t \in [0,1]$. Then $\mathbb{P}\big[(\forall i)|\langle \mathbf{x}, \mathbf{a}_i\rangle| \leq t\big] \geq (t/4)^k$.*

Proof. The claim follows immediately from Theorem 2.5.2 and the fact that the one-dimensional standard Gaussian density in $[-1,1]$ is at least $(2\pi e)^{-1/2} \geq 1/8$. □

CHAPTER 3

Overcomplete Tensor Decomposition

In this chapter we study the decomposition of symmetric order-3 tensors:

Given a tensor $\boldsymbol{\mathcal{T}} = \sum_{i\in[n]} \mathbf{a}_i^{\otimes 3}$ of rank n, recover the component vectors $\mathbf{a}_i \in \mathbb{R}^d$.

The problem is *undercomplete* if the $\mathbf{a}_i$s are linearly independent, otherwise it is *overcomplete*. In contrast to the undercomplete regime, where efficient algorithms are developed, e.g., *simultaneous diagonalization* [**LRA93**] and *tensor power iteration* [**DLDMV00**, **JGKA19**], the overcomplete regime, especially the order-3 case, is much more challenging and less understood [**JGKA19**, Chapter 7] in two ways:

- It is not obvious to apply algorithms like the simultaneous diagonalization algorithm and tensor power iteration directly given the linear dependency of $\mathbf{a}_i$s.
- There are fewer techniques available for the order-3 case than there are for higher order [**JGKA19**, Section 7.3].

These challenges motivate us to study the following question:

Can we efficiently recover $\mathbf{a}_i$s when $\boldsymbol{\mathcal{T}}$ is overcomplete?

Organization of the chapter. In Section 3.1, we discuss some of techniques and challenges in overcomplete tensor decomposition, and present our results on the high level. In Section 3.2, we present our algorithmic ideas, then formally propose our algorithm (Algorithm 2), its analysis (Theorem 3.2.1), as well as high level ideas to prove the correctness of Algorithm 2. And we implement these proof ideas in Section 3.3. In Section 3.4, we provide numerical simulation results of Algorithm 2 on synthetic data.

3.1. Introduction

Among basic tensor decomposition techniques for the undercomplete case we have *tensor power iteration* and *the simultaneous diagonalization algorithm* (See Section 2.3 for a short introduction).

Tensor power iteration is more robust than the simultaneous diagonalization algorithm, while the simultaneous diagonalization algorithm can be applied more generally in the undercomplete case: Tensor power iteration is mainly an algorithm for orthogonal tensors (orthogonal $\mathbf{a}_i$s) and the general case with additional information, usually a whitening transform that orthogonalizes independent components, e.g., the second moment matrix in [**HK13**], while the simultaneous diagonalization algorithm can decompose the general case without additional information.

Our contributions below are based on the simultaneous diagonalization algorithm because of this additional power. The robustness of the simultaneous diagonalization algorithm is studied in several papers; our analysis builds on top of [**BCMV14**, **BCV14**] and [**GVX14**].

For the overcomplete regime, many techniques have been already developed. We have algorithms such as FOOBI [**LCC07**], an algorithm to decompose 4th cumulant tensor based on the simultaneous diagonalization, and the work of [**AGJ14**, **AGJ15**], which are based on tensor power iteration with incoherent components. Anandkumar et al. [**AGJ17**] analyze the dynamic and convergence of tensor power iteration for overcomplete order-3 tensors, and Ge and Ma [**GM17**] provide similar analysis for random order-4 tensors. In [**BCV14**], the authors analyze the robustness of the simultaneous diagonalization, and proposed an algorithm that has running time exponential in the rank n. The robustness result is further extended to the smoothed analysis case [**BCMV14**], where the input tensor has its component vectors perturbed with a small Gaussian noise, and an algorithm with maximal rank $n \leq d^{\lfloor l-1 \rfloor/2}/2$ for order-l tensors is proposed. In the order-3 case, Recently, a line of research utilizes the sum-of-squares technique to perform tensor decomposition in the overcomplete regime: Ge and Ma [**GM15**] give a polynomial algorithm for overcomplete order-3 random tensors, [**MSS16**] improves the previous results and gives a sum-of-squares algorithm for order-4 tensors. The order-4 case is further improved in [**HSS19**].

Many techniques for the overcomplete case only make sense for orders 4 and higher or have weaker guarantees in the order-3 case. For example, some techniques use the fact that a $d \times d \times d \times d$ tensor can be seen as an $d^2 \times d^2$ matrix (and similarly for orders higher than 4), while no equally useful operation is available for order-3 tensors. Algorithms for order-3 tensors based on tensor power iteration or sum-of-squares usually require certain assumptions on the input tensor, e.g., incoherent components or randomness. Nevertheless, there are several results about decomposition in the order-3 case that are relevant to our work.

Among works closest to ours, [**DDL14**, **DDL17**] propose an algorithm that is efficient in the mildly overcomplete case for overcomplete order-3 tensor decomposition under natural non-degeneracy conditions. Though our results have similar assumptions and computational cost compared to [**DDL14**,**DDL17**], our algorithm is comparatively a very simple randomized algorithm and we provide a rigorous robustness analysis.

Our contributions. We propose an algorithm based on the simultaneous diagonalization algorithm for overcomplete tensor decomposition. Our informal claim is as follows:

Claim 3.1.1 (Informal statement of Theorem 3.2.1). *Given a symmetric order-3 tensor $\boldsymbol{\mathcal{T}} = \sum_{i \in [d+k]} \mathbf{a}_i^{\otimes 3} \in \mathbb{R}^{d \times d \times d}$ and when any d-subset of the $\mathbf{a}_i$s is linearly independent, there is a randomized algorithm that recovers $\mathbf{a}_i$s within ε error and with expected running time polynomial in d^k, $1/\varepsilon^k$ and natural conditioning parameters.*

Note that our goal is to show that the running time has polynomial dependence in that sense and the error has inverse polynomial dependence but we do not optimize the degrees of the polynomials. Even though the algorithm is exponential in k, we show in Chapter 4 the case $k = 1$ already makes possible a new result on Gaussian mixture learning.

Our proposed algorithm (Algorithm 2) and its analysis (Theorem 3.2.1) are stronger than Claim 3.1.1 in two important ways: It is robust in the sense that it approximates the $\mathbf{a}_i$s even when the input is a tensor that is ε'-close to $\boldsymbol{\mathcal{T}}$. Also, it turns out that parameter k above, the number of $\mathbf{a}_i$s beyond the dimension d, is not the best notion of overcompleteness. In our result the tensor is of the form $\boldsymbol{\mathcal{T}} = \sum_{i=1}^{r+k} \mathbf{a}_i^{\otimes 3}$, where r is the robust Kruskal rank of $\mathbf{a}_i$s (informally the maximum r such that any r-subset is well-conditioned, Definition 2.3.7), so that k is the number of components above the robust Kruskal rank. Thus, our analysis also applies when the Kruskal rank is less than d.

3.2. Overcomplete Order-3 Tensor Decomposition

We consider the problem of decomposing (recovering $\mathbf{a}_i$s) a symmetric order-3 tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d\times d\times d}$ of rank n:

$$\boldsymbol{\mathcal{T}} = \sum_{i\in[n]} \mathbf{a}_i^{\otimes 3}. \tag{3.2.1}$$

When the $\mathbf{a}_i$s are linearly independent, the simultaneous diagonalization algorithm efficiently recovers them given T. But it has no guarantees if the components are linearly dependent. Our main idea for the linearly dependent case is *divide and conquer*: it is still possible that a large subset $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\}$ of components is linearly independent, so if we cancel out the other components, $\{\mathbf{a}_{r+1}, \ldots, \mathbf{a}_n\}$, the residual tensor can be efficiently decomposed via the simultaneous diagonalization algorithm. To cancel the other components, we search for a vector $\mathbf{x}$ orthogonal to them so that $\mathbf{T}_\mathbf{x} = \sum_{i\in[n]} \langle \mathbf{x}, \mathbf{a}_i\rangle \mathbf{a}_i \mathbf{a}_i^\top$ only involves the linearly independent components. A random or grid search for an approximately orthogonal $\mathbf{x}$ is efficient if the number of components to cancel out is small.

For clarity, we now describe an idealized version of our algorithm as if we had two vectors $\mathbf{x}, \mathbf{y}$ that are exactly orthogonal to the other components, while the actual algorithm uses a random search to find them. We also want $\mathbf{x}$, $\mathbf{y}$ to be *generic* with this orthogonality property, so that they can also play the roles of $\mathbf{x}$, $\mathbf{y}$ in the simultaneous diagonalization algorithm (see Section 2.3). Specifically, the genericity here is that the eigenvalues of $\mathbf{T}_\mathbf{x}\mathbf{T}_\mathbf{y}^{-1}$ are distinct. In that case, the eigendecomposition of $\mathbf{T}_\mathbf{x}\mathbf{T}_\mathbf{y}^{-1}$ recovers the directions of $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\}$. Then, a linear system of equations provides the lengths of $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\}$. Once $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\}$ is recovered, the components of $\boldsymbol{\mathcal{T}}$ associated to them can be removed from $\boldsymbol{\mathcal{T}}$ via deflation and the simultaneous diagonalization algorithm can be applied a second time to the residual tensor to recover $\{\mathbf{a}_{r+1}, \ldots, \mathbf{a}_n\}$.

Our formal statements are Algorithm 2 and Theorem 3.2.1. Our algorithm uses the simultaneous diagonalization algorithm (Algorithm 1, as presented in [**GVX14**]) as a subroutine.

Algorithm 1 DIAGONALIZE [**GVX14**]

```
Inputs: $\mathbf{M}_{\boldsymbol{\mu}}, \mathbf{M}_{\boldsymbol{\lambda}} \in \mathbb{R}^{d\times d}$, number of vectors $r$.
compute the SVD of $\mathbf{M}_{\boldsymbol{\mu}} = \mathbf{V}\mathbf{D}\mathbf{U}^\top$. Let $\mathbf{W}$ be matrix whose columns are the left singular vectors (columns of $\mathbf{V}$) corresponding to the top $r$ singular values;
compute $\mathbf{M} = (\mathbf{W}^\top \mathbf{M}_{\boldsymbol{\mu}} \mathbf{W})(\mathbf{W}^\top \mathbf{M}_{\boldsymbol{\lambda}} \mathbf{W})^{-1}$;
compute the eigendecomposition: $\mathbf{M} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^{-1}$;
Outputs: columns of $\mathbf{W}\mathbf{P}$.
```

Algorithm 2 Approximate tensor decomposition

```
Inputs: tensor $\tilde{\boldsymbol{\mathcal{T}}} \in \mathbb{R}^{d\times d\times d}$, error tolerance $\varepsilon$, tensor rank $n$, overcompleteness $k$, upper bound $M$ on $\|\mathbf{a}_i\|_2$ for $i \in [n]$. Let $r = n - k$ (Kruskal rank).
repeat
    pick $\mathbf{x}, \mathbf{y}$ i.i.d. uniformly at random in $\mathbb{S}^{d-1}$;
    invoke Algorithm 1 with $\tilde{\mathbf{T}}_{\mathbf{x}}, \tilde{\mathbf{T}}_{\mathbf{y}}$ and $r$. Denote the outputs by $\tilde{\mathbf{a}}_i$ for $i \in [r]$;
    solve the least squares problem: $\min_{\xi_1,\ldots,\xi_r} \|\tilde{\mathbf{A}}_{:r} \operatorname{diag}(\xi_i \langle \mathbf{x}, \tilde{\mathbf{a}}_i\rangle) \tilde{\mathbf{A}}_{:r}^\top - \tilde{\mathbf{T}}_x\|_2$.
    set $\boldsymbol{\mathcal{R}} = \tilde{\boldsymbol{\mathcal{T}}} - \sum_{i\in[r]} \xi_i \tilde{\mathbf{a}}_i^{\otimes 3}$;
    pick $\mathbf{x}', \mathbf{y}'$ i.i.d. uniformly at random in $\mathbb{S}^{d-1}$;
    invoke Algorithm 1 with $\mathbf{R}_{\mathbf{x}'}, \mathbf{R}_{\mathbf{y}'}$ and $k$. Denote the outputs by $\tilde{\mathbf{a}}_{r+i}$ for $i \in [k]$;
    solve the least squares problem: $\min_{\xi_{r+1},\ldots,\xi_{r+k}} \|\tilde{\mathbf{A}}_{r+1:} \operatorname{diag}(\xi_{r+i} \langle \mathbf{x}', \tilde{\mathbf{a}}_{r+i}\rangle) \tilde{\mathbf{A}}_{r+1:}^\top - \mathbf{R}_{x'}\|_2$.
    reconstruct the tensor $\boldsymbol{\mathcal{T}}' = \sum_{i\in[r+k]} \xi_i \tilde{\mathbf{a}}_i^{\otimes 3}$;
until $\|\boldsymbol{\mathcal{T}}' - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} \le \varepsilon$, $\max_{i\in[r+k]} |\xi_i|^{1/3} \le 2M$
Outputs: $\mathbf{a}_i' := \xi_i^{1/3} \tilde{\mathbf{a}}_i$, for $i \in [r+k]$.
```

Theorem 3.2.1 (Correctness of Algorithm 2). *Let* $\boldsymbol{\mathcal{T}} = \sum_{i\in[r+k]} \mathbf{a}_i^{\otimes 3}$, $1 \le k \le (r-2)/2$, *and* $\mathbf{a}_i \in \mathbb{R}^d$. *Let* $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_{r+k}]$ *and* $\text{K-rank}_\tau(\mathbf{A}) \ge r$. *Let* $\tau > 0$, $M \ge \max_{i\in[r+k]} \|\mathbf{a}_i\|_2$, $0 < m \le \min_{i\in[r+k]} \|\mathbf{a}_i\|_2$ *and* $0 < \varepsilon_{out} \le \min\{1, m^3\}$. *There exist polynomials* $\operatorname{poly}_{3.2.1}(d, \tau, M)$, $\operatorname{poly}'_{3.2.1}(d, \tau, M, m^{-1})$, *such that if* $\varepsilon_{in} \le \varepsilon_{out} / \operatorname{poly}'_{3.2.1}$ *and* $\tilde{\boldsymbol{\mathcal{T}}}$ *is a tensor such that* $\|\boldsymbol{\mathcal{T}} - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} \le \varepsilon_{in}$, *then Algorithm 2 on input* $\tilde{\boldsymbol{\mathcal{T}}}$ *and* $\varepsilon = \varepsilon_{out} / \operatorname{poly}_{3.2.1}$, *outputs vectors* $\mathbf{a}_1', \ldots, \mathbf{a}_{r+k}'$ *such that for some permutation* π *of* $[r+k]$, *we have* $\|\mathbf{a}_{\pi(i)} - \mathbf{a}_i'\|_2 \le \varepsilon_{out}$, $\forall i \in [r+k]$. *The expected running time is at most* $\operatorname{poly}(d^k, \varepsilon_{out}^{-k}, \tau^k, M^k, m^{-k})$.

Proof idea. The proof has three parts. First we show that if $\boldsymbol{\mathcal{T}}'$ (with which the algorithm finishes) is close to $\tilde{\boldsymbol{\mathcal{T}}}$ and has bounded components, then the components of $\boldsymbol{\mathcal{T}}'$, $\{\mathbf{a}_i' := \xi_i^{1/3} \tilde{\mathbf{a}}_i : i \in [r+k]\}$, are close to those of $\boldsymbol{\mathcal{T}}$. In the second part we show that, assuming good $\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}'$ have been found, how the error propagates in our algorithm and the algorithm indeed finishes with a tensor

$\boldsymbol{\mathcal{T}}'$ that is close to $\tilde{\boldsymbol{\mathcal{T}}}$ (and therefore, close to $\boldsymbol{\mathcal{T}}$ via triangle inequality). In the third part we show the probabilistic bounds that guarantee efficient search of good $\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}'$.

The first part follows from [**BCV14**, Theorem 2.6] (the version we need is Theorem 3.3.1 here).

We now informally state what *good* $\mathbf{x}, \mathbf{y}$ means. Note that in the idealized case, $\mathbf{x}, \mathbf{y}$ are chosen to be orthogonal to $k = n - r$ vectors and to be generic, meaning that $\mathbf{T_x}\mathbf{T_y}^{-1}$ has distinct eigenvalues. In Algorithm 2 we rely on random search to find *good* $\mathbf{x}, \mathbf{y}$, namely $\mathbf{x}, \mathbf{y}$ that satisfy:

(1) nearly orthogonal to last k terms: $|\langle \mathbf{x}, \hat{\mathbf{a}}_{r+i}\rangle|, |\langle \mathbf{y}, \hat{\mathbf{a}}_{r+i}\rangle|$ are small for $i \in [k]$;

(2) non-orthogonality on first r terms: $|\langle \mathbf{x}, \hat{\mathbf{a}}_{i}\rangle|, |\langle \mathbf{y}, \hat{\mathbf{a}}_{i}\rangle|$ are lower bounded for $i \in [r]$;

(3) the eigenvalues of $\mathbf{T_x}\mathbf{T_y}^{-1}$, $\langle \mathbf{x}, \hat{\mathbf{a}}_{i}\rangle / \langle \mathbf{y}, \hat{\mathbf{a}}_{i}\rangle$, are well-separated.

Properties 1 and 2 guarantee that we have r components with noise after contraction, and property 3 guarantees that the simultaneous diagonalization algorithm can be applied to contracted matrices. We will revisit and quantify these properties in Section 3.3.3. There will be also similar properties for $\mathbf{x}', \mathbf{y}'$.

For the second part, we will assume that we have found good vectors $\mathbf{x}, \mathbf{y}$. Theorem 3.3.3 (from [**GVX14**]) and Lemma 3.3.4 guarantee that we can simultaneously diagonalize matrices $\tilde{\mathbf{T}}_\mathbf{x}$ and $\tilde{\mathbf{T}}_\mathbf{y}$ using the simultaneous diagonalization algorithm (Algorithm 1), and the outputs are close to the directions of $\mathbf{a}_i$s. Lemma 3.3.5 shows that we can recover approximately the lengths of $\mathbf{a}_i$s by solving a least squares problem once we have the directions. At this point we completed the recovery of r components. Lemma 3.3.6 shows that when the deflation error is small, the residual tensor $\boldsymbol{\mathcal{R}}$ can be decomposed in the same way and the last k directions are recovered. At the end of the second part, Lemma 3.3.7 shows that the lengths of the last k components are approximately recovered.

The third part is shown in Lemmas 3.3.10 and 3.3.11. □

Remark 2. *The constraint* $1 \leq k \leq (r-2)/2$ *on the rank is because of Kruskal's theorem (Theorem 2.3.6): we need* $2(r+k)+2 \leq 3r$ *to guarantee identifiability.*

Remark 3. *Theorem 3.2.1 has an immediate extension to order-*$3p$ *symmetric tensors for integer* $p > 1$ *by "batching" each set of* p *modes together and reshaping into a* $d^p \times d^p \times d^p$ *tensor. However,* *for higher order tensors, additional tools are available. Hence we restrict ourselves to the (in this sense) harder case of order-*3 *tensors.*

3.3. Proof of Theorem 3.2.1

3.3.1. Uniqueness of Decomposition. We show that if Algorithm 2 satisfies its termination condition, then its outputs are close to the components of $\boldsymbol{\mathcal{T}}$. We deduce this directly from the following known result on the stability of tensor decompositions.

Theorem 3.3.1 ([BCV14, Theorem 5]). *Suppose a rank R tensor $\boldsymbol{\mathcal{T}} = \sum_{i\in[R]} \mathbf{a}_i^{\otimes 3} \in \mathbb{R}^{d\times d\times d}$ is ρ-bounded. Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R]$ with $3\,\text{K-rank}_\tau(\mathbf{A}) \geq 2R+2$. Then for every $\varepsilon' \in (0,1)$, there exists $\varepsilon = \varepsilon' / \operatorname{poly}_{3.3.1}(R, \tau, \rho, \rho', d)$ for a fixed polynomial $\operatorname{poly}_{3.3.1}$ so that for any other ρ'-bounded decomposition $\boldsymbol{\mathcal{T}}' = \sum_{i\in[R]} (\mathbf{a}_i')^{\otimes 3}$ with $\|\boldsymbol{\mathcal{T}}' - \boldsymbol{\mathcal{T}}\|_{\mathrm{F}} \leq \varepsilon$, there exists a permutation matrix Π and diagonal matrix $\mathbf{\Lambda}$ such that $\|\mathbf{\Lambda}^3 - \mathbf{I}\|_{\mathrm{F}} \leq \varepsilon'$ and $\|\mathbf{A}' - \mathbf{A\Pi\Lambda}\|_{\mathrm{F}} \leq \varepsilon'$.*

The original statement in [**BCV14**] explicitly assumes that $\boldsymbol{\mathcal{T}}$ (the sum of R rank-1 tensors) has rank R, but this assumption is redundant: a tensor $\boldsymbol{\mathcal{T}} = \sum_{i\in[R]} \mathbf{a}_i^{\otimes 3} \in \mathbb{R}^{d\times d\times d}$ with $3\,\text{K-rank}(\mathbf{A}) \geq 2R+2$ cannot have another decomposition with less than R terms because of Theorem 2.3.6. Also, the original statement in [**BCV14**] is for the non-symmetric case and we only state here the version we need, specialized to the symmetric case. This restatement is not completely obvious because a symmetric tensor with minimal length symmetric decomposition of length R (i.e., with symmetric rank equal to R) could have a non-symmetric decomposition of shorter length in general. But under Kruskal's condition, $3\,\text{K-rank}_\tau(\mathbf{A}) \geq 2R+2$ (implied by the robust Kruskal condition in Theorem 3.3.1), Kruskal's uniqueness theorem (Theorem 2.3.6) implies that the symmetric and the non-symmetric decompositions (and ranks) of such a $\boldsymbol{\mathcal{T}}$ coincide.

Note that in Theorem 3.3.1 a scaling matrix $\mathbf{\Lambda}$ is introduced. We will use the following corollary instead to have a handier result without the scaling matrix:

Corollary 3.3.2. *In the setting of Theorem 3.3.1, there exists a polynomial $\operatorname{poly}_{3.3.2}(R, \tau, \rho, \rho', d)$ such that if $\varepsilon' \in (0,1)$ and $\varepsilon = \varepsilon' / \operatorname{poly}_{3.3.2}(R, \tau, \rho, \rho', d)$, then for any other ρ'-bounded decomposition $\boldsymbol{\mathcal{T}}' = \sum_{i\in[R]} (\mathbf{a}_i')^{\otimes 3}$ with $\|\boldsymbol{\mathcal{T}}' - \boldsymbol{\mathcal{T}}\|_{\mathrm{F}} \leq \varepsilon$, there exists a permutation π of $[R]$ such that $\forall i \in [R]$, $\|\mathbf{a}_{\pi(i)} - \mathbf{a}_i'\|_2 \leq \varepsilon'$.*

PROOF. We assume that the permutation is the identity. Let $c = (1 + 4\rho/3)$ and $\text{poly}_{3.3.2} = c\,\text{poly}_{3.3.1}$. By Theorem 3.3.1, we have that for each $i \in [R]$:

$$\|\mathbf{a}'_i - \lambda_i \mathbf{a}_i\|_2 \leq c^{-1}\varepsilon', \text{ and } \quad |\lambda_i^3 - 1| \leq c^{-1}\varepsilon'.$$

Since $|x - 1| \leq 4|x^3 - 1|/3$ for all $x \in \mathbb{R}$, the second inequality implies that: $|\lambda_i - 1| \leq 4|\lambda_i^3 - 1|/3 \leq 4c^{-1}\varepsilon'/3$. Therefore

$$\|\mathbf{a}'_i - \mathbf{a}_i\|_2 \leq \|\mathbf{a}'_i - \lambda_i \mathbf{a}_i\|_2 + |\lambda_i - 1|\|\mathbf{a}_i\|_2 \leq (1 + 4\rho/3)c^{-1}\varepsilon' = \varepsilon'.$$

□

3.3.2. Robust Decomposition. In this subsection, we will derive the forward error propagation of Algorithm 2, i.e., how the output error depends on the input error in each step of Algorithm 2. We will assume throughout this subsection that we already have two unit vectors $\mathbf{x}, \mathbf{y}$ that are nearly orthogonal to $\hat{\mathbf{a}}_{r+1}, \ldots, \hat{\mathbf{a}}_{r+k}$, that is, $|\langle \mathbf{x}, \hat{\mathbf{a}}_{r+i}\rangle|, |\langle \mathbf{y}, \hat{\mathbf{a}}_{r+i}\rangle| \leq \theta$ for $i \in [k]$, where θ will be chosen later, and $\text{K-rank}_\tau(\mathbf{A}) \geq r$. Let $\boldsymbol{\mathcal{E}}_{in} = \boldsymbol{\mathcal{T}} - \tilde{\boldsymbol{\mathcal{T}}}$ be the input error tensor. Also recall that $\|\mathbf{a}_i\| \in [m, M]$. We summarize the roadmap of this subsection below in Fig. 3.1.

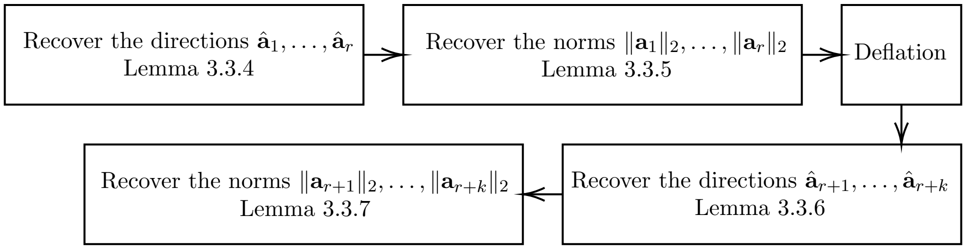

Recover the directions $\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_r$
Lemma 3.3.4
Recover the norms $\|\mathbf{a}_1\|_2, \ldots, \|\mathbf{a}_r\|_2$
Lemma 3.3.5
Deflation
Recover the directions $\hat{\mathbf{a}}_{r+1}, \ldots, \hat{\mathbf{a}}_{r+k}$
Lemma 3.3.6
Recover the norms $\|\mathbf{a}_{r+1}\|_2, \ldots, \|\mathbf{a}_{r+k}\|_2$
Lemma 3.3.7

FIGURE 3.1. Roadmap of Section 3.3.2.

Part 1: robust diagonalization. We first cite the robust analysis of Algorithm 1.

Theorem 3.3.3 ([GVX14, Theorem 5.2]). *Let* $\mathbf{T}_{\boldsymbol{\mu}} = \sum_{i \in [r]} \mu_i \mathbf{a}_i \mathbf{a}_i^\top = \mathbf{A}\,\text{diag}(\boldsymbol{\mu})\mathbf{A}^\top$, $\mathbf{T}_{\boldsymbol{\lambda}} = \sum_{i \in [r]} \lambda_i \mathbf{a}_i \mathbf{a}_i^\top = \mathbf{A}\,\text{diag}(\boldsymbol{\lambda})\mathbf{A}^\top$, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_r]$, $\mathbf{a}_i \in \mathbb{R}^d$, $\|\mathbf{a}_i\| = 1$, $\lambda_i, \mu_i \in \mathbb{R}$ *for* $i \in [r]$. *Suppose*

(1) $\sigma_r(\mathbf{A}) > 0$,

(2) $\mu_i, \lambda_i \neq 0$ *for all* i.

(3) $(\forall i)\ 0 < k_l \leq |\mu_i|, |\lambda_i| \leq k_u$, *and*

(4) $(\forall i \neq j)$ $|\mu_i/\lambda_i - \mu_j/\lambda_j| \geq \alpha > 0$.

Let $0 < \varepsilon_{3.3.3} < 1$ *and* $\tilde{\mathbf{T}}_{\boldsymbol{\mu}}, \tilde{\mathbf{T}}_{\boldsymbol{\lambda}}$ *be matrices such that*

$$\|\mathbf{T}_{\boldsymbol{\mu}} - \tilde{\mathbf{T}}_{\boldsymbol{\mu}}\|_{\mathrm{F}}, \|\mathbf{T}_{\boldsymbol{\lambda}} - \tilde{\mathbf{T}}_{\boldsymbol{\lambda}}\|_{\mathrm{F}} \leq \frac{\varepsilon_{3.3.3} k_l^2 \sigma_r(\mathbf{A})^3 \min\{\alpha, 1\}}{2^{11}\kappa(\mathbf{A}) k_u r^2}.$$

Then Algorithm 1 on input $\tilde{\mathbf{T}}_{\boldsymbol{\mu}}, \tilde{\mathbf{T}}_{\boldsymbol{\lambda}}$ *outputs unit vectors* $\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_r$ *such that for some permutation* π *of* $[r]$ *and signs* $s_1, \ldots, s_r \in \{\pm 1\}$*, and for all* $i \in [r]$ *we have* $\|\mathbf{a}_{\pi(i)} - s_i \tilde{\mathbf{a}}_i\| \leq \varepsilon_{3.3.3}$*. It runs in time* $\mathrm{poly}(d, 1/\alpha, 1/k_l, 1/\sigma_r(\mathbf{A}), 1/\varepsilon_{3.3.3})$.

Now we apply Theorem 3.3.3 to our case: let $\mathbf{E}_{\mathbf{x}} = \mathbf{T}_{\mathbf{x}} - \tilde{\mathbf{T}}_{\mathbf{x}}$ and $\mathbf{E}_{\mathbf{y}} = \mathbf{T}_{\mathbf{y}} - \tilde{\mathbf{T}}_{\mathbf{y}}$. Write

$$\tilde{\mathbf{T}}_{\mathbf{x}} = \hat{\mathbf{A}}_{:r}\mathbf{D}_{\mathbf{x}}\hat{\mathbf{A}}_{:r}^{\top} + \hat{\mathbf{A}}_{r+1:}\mathbf{D}'_{\mathbf{x}}\hat{\mathbf{A}}_{r+1:}^{\top} + (\mathbf{E}_{in})_{\mathbf{x}},$$

where $\hat{\mathbf{A}}_{:r}$ contains $\hat{a}_i$s as columns, $\mathbf{D}_{\mathbf{x}} = \mathrm{diag}(\|\mathbf{a}_i\|^3\langle \mathbf{x}, \hat{\mathbf{a}}_i\rangle)$ for $i \in [r]$ and $\hat{\mathbf{A}}_{r+1:}$ contains $\hat{\mathbf{a}}_{r+i}$s, $\mathbf{D}'_{\mathbf{x}} = \mathrm{diag}(\|\mathbf{a}_{r+i}\|^3\langle \mathbf{x}, \hat{\mathbf{a}}_{r+i}\rangle)$ for $i \in [k]$. Then we have

$$\|\mathbf{E}_{\mathbf{x}}\|_{\mathrm{F}} = \|\hat{\mathbf{A}}_{r+1:}\mathbf{D}'_{\mathbf{x}}\hat{\mathbf{A}}_{r+1:}^{\top} + (\mathbf{E}_{in})_{\mathbf{x}}\|_{\mathrm{F}} \leq kM^3\theta + \varepsilon_{in}, \tag{3.3.1}$$

and similarly for $\mathbf{E}_{\mathbf{y}}$. The following lemma guarantees the correctness of step 3 in Algorithm 2.

Lemma 3.3.4 (Direction estimation). *Let* $\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_r$ *be the outputs of step 3 in Algorithm 2. If*

(1) $\forall i \in [r]$*:* $0 < k_l/m^3 \leq |\langle \mathbf{x}, \hat{\mathbf{a}}_i\rangle|, |\langle \mathbf{y}, \hat{\mathbf{a}}_i\rangle| \leq 1$*;*

(2) $\forall i, j \in [r], i \neq j$*:* $\left|\langle \mathbf{x}, \hat{\mathbf{a}}_r\rangle/\langle \mathbf{y}, \hat{\mathbf{a}}_r\rangle - \langle \mathbf{x}, \hat{\mathbf{a}}_r\rangle/\langle \mathbf{y}, \hat{\mathbf{a}}_r\rangle\right| \geq \alpha > 0$,

then there are signs $s_1, \ldots, s_r \in \{\pm 1\}$ *and a permutation* π *of* $[r]$ *such that* $\forall i \in [r]$*:*

$$\|\hat{\mathbf{a}}_{\pi(i)} - s_i \tilde{\mathbf{a}}_i\| \leq \varepsilon_{3.3.4} := \frac{2^{11}\tau^4 M^7 r^{5/2}(kM^3\theta + \varepsilon_{in})}{k_l^2 \min\{\alpha, 1\}}.$$

This step runs in time $\mathrm{poly}(d, \alpha^{-1}, k_l^{-1}, \tau, M, \varepsilon_{3.3.4}^{-1})$.

Proof. Condition 1 in Theorem 3.3.3 holds since $\text{K-rank}_\tau(\mathbf{A}) \geq r$:

$$\sigma_r(\hat{\mathbf{A}}_{:r}) \geq \sigma_r(\mathbf{A}_{:r})/M \geq 1/(\tau M).$$

Conditions 2 and 3 in Theorem 3.3.3 hold because of our assumptions. Combining (3.3.1) and $\text{K-rank}_\tau(A) \geq r$ which implies

$$\sigma_r(\hat{\mathbf{A}}_{:r})^3 \kappa(\hat{\mathbf{A}}_{:r})^{-1} = \sigma_r(\hat{\mathbf{A}}_{:r})^4 \sigma_1(\hat{\mathbf{A}}_{:r})^{-1} \geq (\sqrt{r}\tau^4 M^4)^{-1},$$

the assumptions of Theorem 3.3.3 are satisfied with parameter $k_u = M^3$. The claim follows. □

Since $\mathbf{x}, \mathbf{y}$ are actually chosen at random, we provide the probability for assumptions of Lemma 3.3.4 to hold in Section 3.3.3.

Part 2: norm estimation. The next step is to recover $\|\mathbf{a}_i\|_2$. This can be done by solving the least squares problem in step 4. To see this, one can verify that when $\tilde{\mathbf{a}}_i = \hat{\mathbf{a}}_i$ and $\tilde{\mathbf{T}}_{\mathbf{x}} = \mathbf{T}_{\mathbf{x}}$ (no error in earlier steps), $\xi_i = \|\mathbf{a}_i\|_2^3$ is a zero error solution to step 4. The following lemma guarantees that we can approximate the norm via step 4:

Lemma 3.3.5 (Norm estimation). *Let $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_r$ be the columns of $(\tilde{\mathbf{A}}_{:r}^\dagger)^\top$. If Lemma 3.3.4 holds with $\varepsilon_{3.3.4} \leq \min\{k_l/(2m^3), (2\sqrt{r}\tau M)^{-1}\}$, then $\xi_i = \tilde{\boldsymbol{\mathcal{T}}}(\mathbf{x}, \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i)/\langle \mathbf{x}, \tilde{\mathbf{a}}_i\rangle$ for $i \in [r]$ is the unique solution to step 4 in Algorithm 2 and for the permutation π, signs s_i in Lemma 3.3.4 and all $i \in [r]$ we have*

$$\left| \|\mathbf{a}_{\pi(i)}\|_2^3 - s_i \xi_i \right| \leq \varepsilon_{3.3.5} := 2k_l^{-1} m^3 M^2 \left[3M\varepsilon_{3.3.4} + rM\varepsilon_{3.3.4}^2 + 4\tau^2(kM^3\theta + \varepsilon_{in}) \right].$$

Proof. For simplicity we assume the permutation is the identity. We start by showing $\sigma_r(\tilde{\mathbf{A}}_{:r}) > 0$, which implies $\tilde{\mathbf{A}}_{:r}^\dagger \tilde{\mathbf{A}}_{:r} = \mathbf{I}_r$ and thus $\tilde{\mathbf{b}}_i$ is orthogonal to $\tilde{\mathbf{a}}_j$ for $i, j \in [r], i \neq j$. By Lemma 3.3.4, the distance between corresponding columns of $\tilde{\mathbf{A}}_{:r}\operatorname{diag}(s_i)$ and $\hat{\mathbf{A}}_{:r}$ is at most $\varepsilon_{3.3.4}$, therefore by Theorem 2.2.1 we have

$$\left|\sigma_r\big(\tilde{\mathbf{A}}_{:r}\operatorname{diag}(s_i)\big) - \sigma_r(\hat{\mathbf{A}}_{:r})\right| \leq \|\tilde{\mathbf{A}}_{:r}\operatorname{diag}(s_i) - \hat{\mathbf{A}}_{:r}\|_2 \leq \sqrt{r}\varepsilon_{3.3.4},$$

which implies

$$\sigma_r(\tilde{\mathbf{A}}_{:r}) = \sigma_r\big(\tilde{\mathbf{A}}_{:r}\operatorname{diag}(s_i)\big) \geq \sigma_r(\hat{\mathbf{A}}_{:r}) - \sqrt{r}\varepsilon_{3.3.4} \geq (\tau M)^{-1} - \sqrt{r}\varepsilon_{3.3.4} \geq 1/(2\tau M). \tag{3.3.2}$$

Next, we show that ξ_i is the unique solution to step 4. We restate the least squares problem in a matrix-vector product form:

$$\min_{\xi_i}\|\tilde{\mathbf{A}}^{\odot 2}[\langle \mathbf{x},\tilde{\mathbf{a}}_1\rangle\xi_1,\ldots,\langle \mathbf{x},\tilde{\mathbf{a}}_r\rangle\xi_r]^\top - \mathrm{vec}(\tilde{\mathbf{T}}_{\mathbf{x}})\|_2,$$

where $\tilde{\mathbf{A}}^{\odot 2} = \left[\mathrm{vec}(\tilde{\mathbf{a}}_1\tilde{\mathbf{a}}_1^\top),\ldots,\mathrm{vec}(\tilde{\mathbf{a}}_r\tilde{\mathbf{a}}_r^\top)\right] \in \mathbb{R}^{d^2\times r}$. It follows that $\sigma_r(\tilde{\mathbf{A}}^{\odot 2}) = \sigma_r(\tilde{\mathbf{A}}_{:r})^2 > 0$ and thus the solution is unique. Let $\tilde{\mathbf{B}}^{\odot 2} = \left[\mathrm{vec}(\tilde{\mathbf{b}}_1\tilde{\mathbf{b}}_1^\top),\ldots,\mathrm{vec}(\tilde{\mathbf{b}}_r\tilde{\mathbf{b}}_r^\top)\right]^\top$ and notice that $\tilde{\mathbf{B}}^{\odot 2}\tilde{\mathbf{A}}^{\odot 2} = \mathbf{I}_r$. The solution to the least squares problem is then given by $\xi_i = \tilde{\boldsymbol{\mathcal{T}}}(\mathbf{x},\tilde{\mathbf{b}}_i,\tilde{\mathbf{b}}_i)/\langle \mathbf{x},\tilde{\mathbf{a}}_i\rangle$ since

$$\left[\langle \mathbf{x},\tilde{\mathbf{a}}_1\rangle\xi_1,\ldots,\langle \mathbf{x},\tilde{\mathbf{a}}_r\rangle\xi_r\right]^\top = \tilde{\mathbf{B}}^{\odot 2}\,\mathrm{vec}(\tilde{\mathbf{T}}_{\mathbf{x}}) = \left[\tilde{\mathbf{b}}_1^\top\tilde{\mathbf{T}}_{\mathbf{x}}\tilde{\mathbf{b}}_1,\ldots,\tilde{\mathbf{b}}_r^\top\tilde{\mathbf{T}}_{\mathbf{x}}\tilde{\mathbf{b}}_r\right]^\top.$$

Finally we show that $s_i\xi_i$ is close to $\|\mathbf{a}_i\|_2^3$. The deviation of $s_i\xi_i$ from $\|\mathbf{a}_i\|_2^3$ is bounded by:

$$\begin{aligned}\left|\|\mathbf{a}_i\|_2^3 - s_i\xi_i\right| &= \left|\|\mathbf{a}_i\|_2^3 - \frac{1}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle}\Big(\sum_{j\in[r]}\langle \mathbf{x},\mathbf{a}_j\rangle\langle\tilde{\mathbf{b}}_i,\mathbf{a}_j\rangle^2 + \tilde{\mathbf{b}}_j^\top\mathbf{E}_{\mathbf{x}}\tilde{\mathbf{b}}_j\Big)\right| \\ &\le \underbrace{\left|\frac{\langle \mathbf{x},\hat{\mathbf{a}}_i\rangle\langle\tilde{\mathbf{b}}_i,\hat{\mathbf{a}}_i\rangle^2}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle} - 1\right|}_{\substack{\text{small when } s_i\tilde{\mathbf{a}}_i \text{ close to } \hat{\mathbf{a}}_i \\ \text{(note that } \langle\tilde{\mathbf{b}}_i,\tilde{\mathbf{a}}_i\rangle=1)}}\|\mathbf{a}_i\|_2^3 + \sum_{j\in[r],j\neq i}\left(\|\mathbf{a}_j\|_2^3\underbrace{\left|\frac{\langle \mathbf{x},\hat{\mathbf{a}}_j\rangle\langle\tilde{\mathbf{b}}_i,\hat{\mathbf{a}}_j\rangle^2}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle}\right|}_{\substack{\text{small when } s_j\tilde{\mathbf{a}}_j \text{ close to } \hat{\mathbf{a}}_j \\ \text{(note that } \langle\tilde{\mathbf{b}}_i,\tilde{\mathbf{a}}_j\rangle=0)}} + \underbrace{\left|\frac{\tilde{\mathbf{b}}_i^\top\mathbf{E}_{\mathbf{x}}\tilde{\mathbf{b}}_i}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle}\right|}_{\text{error from } \mathbf{E}_{\mathbf{x}}}\right).\end{aligned}\tag{3.3.3}$$

We analyze the deviation of each term in (3.3.3). By standard arguments using triangle and Cauchy-Schwarz inequalities, we have for all $i, j \in [r]$:

$$\begin{aligned}&|\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle| \ge |\langle \mathbf{x},\hat{\mathbf{a}}_i\rangle| - \varepsilon_{3.3.4} \ge k_l/m^3 - \varepsilon_{3.3.4} \ge k_l/(2m^3), \\ &|\langle \mathbf{x}, s_j\tilde{\mathbf{a}}_j\rangle - \langle \mathbf{x},\hat{\mathbf{a}}_j\rangle| \le \varepsilon_{3.3.4}, \quad |\langle\tilde{\mathbf{b}}_i, s_j\tilde{\mathbf{a}}_j\rangle - \langle\tilde{\mathbf{b}}_i,\hat{\mathbf{a}}_j\rangle| \le \varepsilon_{3.3.4},\end{aligned}\tag{3.3.4}$$

where the first line comes from the assumptions of the lemma, and the last line follows from Lemma 3.3.4. Notice that $\tilde{\mathbf{b}}_i$ is orthogonal to $\tilde{\mathbf{a}}_j$ for $j \neq i$, and $\langle\tilde{\mathbf{b}}_i,\tilde{\mathbf{a}}_i\rangle = 1$. (3.3.4) implies that:

$$\left|\frac{\langle \mathbf{x},\hat{\mathbf{a}}_i\rangle\langle\tilde{\mathbf{b}}_i,\hat{\mathbf{a}}_i\rangle^2}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle} - 1\right| \le 6k_l^{-1}m^3\varepsilon_{3.3.4}, \quad \left|\frac{\langle \mathbf{x},\hat{\mathbf{a}}_j\rangle\langle\tilde{\mathbf{b}}_i,\hat{\mathbf{a}}_j\rangle^2}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle}\right| \le 2k_l^{-1}m^3\varepsilon_{3.3.4}^2.\tag{3.3.5}$$

The last term in (3.3.3) is bounded by:

$$\left|\frac{\tilde{\mathbf{b}}_i^\top\mathbf{E}_{\mathbf{x}}\tilde{\mathbf{b}}_i}{\langle \mathbf{x}, s_i\tilde{\mathbf{a}}_i\rangle}\right| \le 2k_l^{-1}m^3\|\mathbf{E}_{\mathbf{x}}\|_2\|\tilde{\mathbf{b}}_i\|_2^2 \le 2k_l^{-1}m^3\|\mathbf{E}_{\mathbf{x}}\|_{\mathrm{F}}\sigma_r(\tilde{\mathbf{A}}_{:r})^{-2} \le 8k_l^{-1}m^3\tau^2M^2\|\mathbf{E}_{\mathbf{x}}\|_{\mathrm{F}},\tag{3.3.6}$$

where the second inequality follows from the definition of $\tilde{\mathbf{b}}_i$, and the last inequality applies (3.3.2). Combining (3.3.1), (3.3.3), (3.3.5) and (3.3.6) gives the desired result. □

Part 3: deflation. After we deflate $\boldsymbol{\mathcal{T}}$ with the previously recovered r components, the induced error with respect to the exact deflation $\sum_{i\in[k]}\mathbf{a}_{r+i}^{\otimes 3}$ is given by $\boldsymbol{\mathcal{E}}' = \boldsymbol{\mathcal{E}}_{in} + \sum_{i\in[r]}(\mathbf{a}_i^{\otimes 3} - \xi_i\tilde{\mathbf{a}}_i^{\otimes 3})$. Now we show that the remaining tensor can be decomposed with the same strategy via step 7 in Algorithm 2.

Lemma 3.3.6 (Direction estimation). *Let $\tilde{\mathbf{a}}_{r+1},\dots,\tilde{\mathbf{a}}_{r+k}$ be the outputs of step 7 in Algorithm 2. If*

(1) $\forall i\in[k]$*:* $0 < k_l'/m^3 \le \left|\langle \mathbf{x}', \hat{\mathbf{a}}_{r+i}\rangle\right|, \left|\langle \mathbf{y}', \hat{\mathbf{a}}_{r+i}\rangle\right| \le 1$*;*

(2) $\forall i,j\in[k], i\neq j$*:* $\left|\langle \mathbf{x}', \hat{\mathbf{a}}_{r+i}\rangle/\langle \mathbf{y}', \hat{\mathbf{a}}_{r+i}\rangle - \langle \mathbf{x}', \hat{\mathbf{a}}_{r+j}\rangle/\langle \mathbf{y}', \hat{\mathbf{a}}_{r+j}\rangle\right| \ge \alpha' > 0$*,*

then there are signs $s_{r+1},\dots,s_{r+k}\in\{\pm 1\}$ and a permutation π' of $[k]$ such that $\forall i\in[k]$:

$$\|\hat{\mathbf{a}}_{r+\pi'(i)} - s_{r+i}\tilde{\mathbf{a}}_{r+i}\|_2 \le \varepsilon_{3.3.6} := \frac{2^{11}\tau^4 M^7 k^{5/2}\|\boldsymbol{\mathcal{E}}'\|_{\mathrm{F}}}{(k_l')^2\min\{\alpha',1\}}.$$

This step runs in time $\mathrm{poly}(d, k_l'^{-1}, \alpha'^{-1}, \tau, M, \varepsilon_{3.3.6}^{-1})$.

Proof. The proof is similar to the proof of Lemma 3.3.4.

We only need to show that Theorem 3.3.3 can be applied here. Take

$$\tilde{\mathbf{T}}_{\boldsymbol{\mu}} = \mathbf{R}_{\mathbf{x}'} \quad \mathbf{T}_{\boldsymbol{\mu}} = \hat{\mathbf{A}}_{r+1:}\,\mathrm{diag}(\|\mathbf{a}_{r+i}\|_2^3\langle x', \hat{\mathbf{a}}_{r+i}\rangle)\hat{\mathbf{A}}_{r+1:}^\top$$
$$\tilde{\mathbf{T}}_{\boldsymbol{\lambda}} = \mathbf{R}_{\mathbf{y}'} \quad \mathbf{T}_{\boldsymbol{\lambda}} = \hat{\mathbf{A}}_{r+1:}\,\mathrm{diag}(\|\mathbf{a}_{r+i}\|_2^3\langle \mathbf{y}', \hat{\mathbf{a}}_{r+i}\rangle)\hat{\mathbf{A}}_{r+1:}^\top.$$

Condition 1 in Theorem 3.3.3 holds because K-rank$_\tau(\mathbf{A}) \ge r$. Condition 2 and 3 in Theorem 3.3.3 follow with parameters k_l', M^3, α'. Combining $\|\tilde{\mathbf{T}}_{\boldsymbol{\mu}} - \mathbf{T}_{\boldsymbol{\mu}}\|_{\mathrm{F}}, \|\tilde{\mathbf{T}}_{\boldsymbol{\lambda}} - \mathbf{T}_{\boldsymbol{\lambda}}\|_{\mathrm{F}} \le \|\boldsymbol{\mathcal{E}}'\|_{\mathrm{F}}$ with the robust Kruskal rank condition K-rank$_\tau(\mathbf{A}) \ge r$ which guarantees

$$\sigma_k(\hat{\mathbf{A}}_{r+1:})^3\kappa(\hat{\mathbf{A}}_{r+1:})^{-1} = \sigma_k(\hat{\mathbf{A}}_{r+1:})^4\sigma_1(\hat{\mathbf{A}}_{r+1:})^{-1}(\sqrt{k}\tau^4M^4)^{-1},$$

Theorem 3.3.3 hold and the claim follows. □

With $\tilde{\mathbf{a}}_{r+1},\dots,\tilde{\mathbf{a}}_{r+k}$, we can further approximate the norm of $\mathbf{a}_{r+1},\dots,\mathbf{a}_{r+k}$, in the same way we did for the first r components, via step 8. The following lemma guarantees it works:

Lemma 3.3.7 (Norm estimation). *Let $\tilde{\mathbf{b}}_{r+1}, \dots, \tilde{\mathbf{b}}_{r+k}$ be the columns of $(\tilde{\mathbf{A}}^{\dagger}_{r+1:})^{\top}$. If Lemma 3.3.6 holds with $\varepsilon_{3.3.6} \leq \min\{k'_l/(2m^3), (2\sqrt{k}\tau M)^{-1}\}$, then $\xi_{r+i} = \boldsymbol{\mathcal{R}}(\mathbf{x}', \tilde{\mathbf{b}}_{r+i}, \tilde{\mathbf{b}}_{r+i})/\langle \mathbf{x}', \tilde{\mathbf{a}}_{r+i}\rangle$, for $i \in [k]$ is the unique solution to step 8 in Algorithm 2 and for the permutation π', signs s_{r+i} in Lemma 3.3.6, and all $i \in [k]$ we have*

$$\big|\|\mathbf{a}_{r+\pi'(i)}\|^3 - s_{r+i}\xi_{r+i}\big| \leq \varepsilon_{3.3.7} := 2k_l'^{-1} m^3 M^2 \big[3M\varepsilon_{3.3.6} + kM\varepsilon^2_{3.3.6} + 4\tau^2 \|\boldsymbol{\mathcal{E}}'\|_{\mathrm{F}}\big].$$

Proof. The proof is similar to the proof of Lemma 3.3.5.

We start by bounding $\sigma_k(\tilde{\mathbf{A}}_{r+1:})$ from below. By Lemma 3.3.6, the distance between corresponding columns of $\tilde{\mathbf{A}}_{:r}\operatorname{diag}(s_i)$ and $\hat{\mathbf{A}}_{:r}$ is at most $\varepsilon_{3.3.6}$. Similarly, by Theorem 2.2.1:

$$\sigma_k(\tilde{\mathbf{A}}_{r+1:}) = \sigma_k(\operatorname{diag}(s_{r+i})\tilde{\mathbf{A}}_{r+1:}) \geq \sigma_k(\hat{\mathbf{A}}_{r+1:}) - \sqrt{k}\varepsilon_{3.3.6} \geq \frac{1}{\tau M} - \sqrt{k}\varepsilon_{3.3.6} \geq \frac{1}{2\tau M}. \tag{3.3.7}$$

Thus by reformulating step 8, we can show that $\xi_{r+i} = \boldsymbol{\mathcal{R}}(\mathbf{x}', \tilde{\mathbf{b}}_{r+i}, \tilde{\mathbf{b}}_{r+i})/\langle \mathbf{x}', \tilde{\mathbf{a}}_{r+i}\rangle$ is the unique solution to the least squares problem and $\big|\|\mathbf{a}_{r+i}\|^3 - s_{r+i}\xi_{r+i}\big|$ is bounded by:

$$\begin{aligned}\big|\|\mathbf{a}_{r+i}\|^3 - s_{r+i}\xi_{r+i}\big| \leq \|\mathbf{a}_{r+i}\|^3 &\left|\frac{\langle \mathbf{x}', \hat{\mathbf{a}}_{r+i}\rangle\langle \tilde{\mathbf{b}}_{r+i}, \hat{\mathbf{a}}_{r+i}\rangle^2}{\langle \mathbf{x}', s_{r+i}\tilde{\mathbf{a}}_{r+i}\rangle} - 1\right| \\ &+ \sum_{j\in[k], j\neq i}\Big(\|\mathbf{a}_{r+j}\|^3\left|\frac{\langle \mathbf{x}', \hat{\mathbf{a}}_{r+j}\rangle\langle \tilde{\mathbf{b}}_{r+i}, \hat{\mathbf{a}}_{r+j}\rangle^2}{\langle \mathbf{x}', s_{r+i}\tilde{\mathbf{a}}_{r+i}\rangle}\right| + \left|\frac{\tilde{\mathbf{b}}^{\top}_{r+i}\mathbf{E}'_{\mathbf{x}}\tilde{\mathbf{b}}_{r+i}}{\langle \mathbf{x}', s_{r+i}\tilde{\mathbf{a}}_{r+i}\rangle}\right|\Big).\end{aligned} \tag{3.3.8}$$

Similar to (3.3.4), we have the following bounds for the terms in (3.3.8):

(3.3.9)

$$|\langle \mathbf{x}', s_{r+i}\tilde{\mathbf{a}}_{r+i}\rangle| \geq k'_l/m^3 - \varepsilon_{3.3.6} \geq k'_l/(2m^3) \quad \text{for } i \in [k],$$

$$|\langle \mathbf{x}', \hat{\mathbf{a}}_{r+j}\rangle - \langle \mathbf{x}', s_{r+i}\tilde{\mathbf{a}}_{r+j}\rangle| \leq \varepsilon_{3.3.6}, \quad |\langle \tilde{\mathbf{b}}_{r+i}, \hat{\mathbf{a}}_{r+j}\rangle - \langle \tilde{\mathbf{b}}_{r+i}, s_{r+i}\tilde{\mathbf{a}}_{r+j}\rangle| \leq \varepsilon_{3.3.6} \quad \text{for } i \in [k],$$

where the first inequality is from assumptions of the lemma, the second and the last are from the conclusion of Lemma 3.3.6. Now we can bound (3.3.8) with (3.3.7) and (3.3.9):

$$\big|\|\mathbf{a}_{r+i}\|^3 - s_{r+i}\xi_{r+i}\big| \leq 2k_l'^{-1}m^3M^3(3\varepsilon_{3.3.6} + (k-1)\varepsilon^2_{3.3.6}) + 8k_l'^{-1}m^3\tau^2M^2\|\boldsymbol{\mathcal{E}}'\|_{\mathrm{F}}.$$

□

3.3.3. Probability Bounds. We give here bounds on the probability of finding good vectors for contraction so that Algorithm 2 will finally terminate in polynomial time. Throughout this subsection, let $\mathbf{x}, \mathbf{y}$ be two independent random vectors distributed uniformly on $\mathbb{S}^{d-1}$, and let $\mathbf{a}_1, \mathbf{a}_2 \ldots, \mathbf{a}_{r+k}$ be such that $\|\mathbf{a}_i\| \in [m, M]$ and $\text{K-rank}_\tau([\mathbf{a}_1, \ldots, \mathbf{a}_{r+k}]) \geq r$, which implies that their directions satisfy $\text{K-rank}_{\tau M}([\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{r+k}]) \geq r$.

We first list the events for *good* $\mathbf{x}, \mathbf{y}$ to hold to apply Lemma 3.3.4:

(1) nearly orthogonal to last k terms: $\mathcal{A}_{1,\mathbf{y}} = \{\forall i \in [k], |\langle \mathbf{y}, \hat{\mathbf{a}}_{r+i}\rangle| \leq \theta\}$;
(2) non-orthogonality on first r terms: $\mathcal{A}_{2,\mathbf{y}} = \{\forall i \in [r], |\langle \mathbf{y}, \hat{\mathbf{a}}_i\rangle| \geq k_l/m^3\}$;
(3) the eigenvalue gap: $\mathcal{A}_3 = \{\forall i \neq j, i, j \in [r], |\langle \mathbf{x}, \hat{\mathbf{a}}_i\rangle / \langle \mathbf{y}, \hat{\mathbf{a}}_i\rangle - \langle \mathbf{x}, \hat{\mathbf{a}}_j\rangle / \langle \mathbf{y}, \hat{\mathbf{a}}_j\rangle| \geq \alpha > 0\}$.

We have similar events $\mathcal{A}_{1,\mathbf{x}}, \mathcal{A}_{2,\mathbf{x}}$. Note that in this subsection k_l, θ and α are considered as fixed parameters.

The structure of this subsection is stated as follows: we will first demonstrate our proof idea for controlling the probability of $\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}$. After presenting our idea, we will first analyze the probability of $\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}$, then the probability of $\mathcal{A}_{1,\mathbf{x}} \cap \mathcal{A}_{2,\mathbf{x}} \cap \mathcal{A}_3$ when conditioned on the other events of y. Finally we will collect these sub-events and give the probability that all of them will hold.

It seems that direct union bound-type arguments are insufficient and some non-trivial conditioning is necessary: First, $\mathcal{A}_{1,\mathbf{x}}$ happens with small probability as meaningful values of θ have to be much smaller than k_l/m^3 and α. Naively applying the union bound on some events and analyzing the rest does not give enough wiggle room for a positive probability. Besides, $\mathcal{A}_3$ is the most complicated in the sense that it controls the difference of two ratios. As we will see later, after conditioning on y, the ideas of analyzing $\mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{y}}$ can be reused for the rest events, which makes the analysis easier to follow. We now state the idea of our argument to bound the probability of $\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}$:

Bands argument. We analyze the events geometrically and replace random unit vectors by random Gaussian vectors together with concentration of their norm. Let z be a random Gaussian vector let a and b be two unit vectors. An event of the form $\{|\langle \mathbf{z}, \mathbf{a}\rangle| \leq t_1\}$ corresponds to a band, while an event like $\{|\langle \mathbf{z}, \mathbf{b}\rangle| \geq t_2\}$ corresponds to the complement of a band. We call them bands of type I and type II, denoted by $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. To better illustrate this, we give a demonstration of bands as the shaded areas in Fig. 3.2.

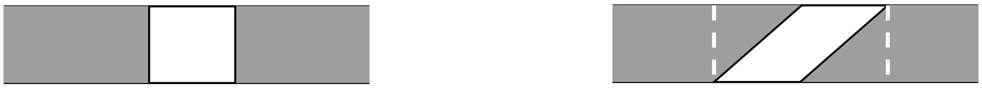
$\mathcal{B}_1$
$\mathcal{B}_2$

(a) Band of type I.

(b) Band of type II.

FIGURE 3.2. Example of bands.

The intersection of bands of type I can be lower-bounded with a direct use of the Gaussian correlation inequality, Lemma 2.5.3, while the intersection of bands of different types needs special care. Consider $\mathcal{B}_1 \cap \mathcal{B}_2$: when $\langle \mathbf{a}, \mathbf{b} \rangle = 0$, the intersection becomes $\mathcal{B}_1$ with a rectangular region excluded. In this case, the two bands will be orthogonal, and the two events are independent. In the general case, the excluded region is a parallelogram depending on $\langle \mathbf{a}, \mathbf{b} \rangle$. See Fig. 3.3 for illustration. In the extreme case, two bands are parallel and hence the probability will be zero when $t_1 \leq t_2$. But when $\langle \mathbf{a}, \mathbf{b} \rangle$ is not too close to one, we can, when bounding the probability, replace the parallelogram by a slightly larger rectangular region without decreasing the final probability too much, which is shown by the white dashed lines in Section 3.3.3. This is essentially done by projecting $\mathbf{b}$ onto span $\{\mathbf{a}\}$ and span $\{\mathbf{a}\}^{\perp}$.



(a) Orthogonal intersection.

(b) Non-orthogonal intersection.

FIGURE 3.3. Intersection of bands.

We see that events $\mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{y}}$ are the intersection of bands and their probability is the probability measure of their intersection. Specifically, we have: $\mathcal{A}_{1,\mathbf{y}} = \cap_{i=i}^{k} \mathcal{B}_{1,i}$, $\mathcal{A}_{2,\mathbf{y}} = \cap_{j=1}^{r} \mathcal{B}_{2,j}$, where $\mathcal{B}_{1,i} := \{|\langle \mathbf{y}, \hat{\mathbf{a}}_{r+i} \rangle| \leq \theta\}$ and $\mathcal{B}_{2,j} := \{|\langle \mathbf{y}, \hat{\mathbf{a}}_j \rangle| \geq k_l/m^3\}$. For the rest of this subsection, let $S = \text{span}\{\hat{\mathbf{a}}_{r+1}, \ldots, \hat{\mathbf{a}}_{r+k}\}^{\perp}$, $S^{\perp} = \text{span}\{\hat{\mathbf{a}}_{r+1}, \ldots, \hat{\mathbf{a}}_{r+k}\}$, and proj_S be the orthogonal projection onto S and $\text{proj}_{S^{\perp}} = \mathbf{I}_k - \text{proj}_S$. Now we can bound the probability of $\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}$:

Lemma 3.3.8. *If $k_l > 0$ and $0 < \theta \leq 2/\sqrt{d}$, then*

$$\mathbb{P}[\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}] \geq p_1 := (\theta\sqrt{d}/8)^k \big(1/4 - r\sqrt{d/2\pi}\tau M(4k_l/m^3 + \sqrt{k}\tau M\theta)\big).$$

Proof. Write $\mathbf{y} = \mathbf{z}/\|\mathbf{z}\|_2$, where $\mathbf{z}$ is a standard Gaussian random vector. Consider the following events corresponding to $\mathbf{z}$, for $R_1 < R_2$ to be chosen later: $\mathcal{B}'_{1,i} := \{|\langle \mathbf{z}, \hat{\mathbf{a}}_{r+i}\rangle| \le R_1\theta\}$ and $\mathcal{B}'_{2,j} := \{|\langle \mathbf{z}, \hat{\mathbf{a}}_j\rangle| \ge R_2 k_l/m^3\}$. We have

$$\begin{aligned}
&\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}} = (\cap_i \mathcal{B}_{1,i}) \cap (\cap_j \mathcal{B}_{2,j}) = (\cap_i \mathcal{B}_{1,i}) \setminus (\cup_j \mathcal{B}^c_{2,j}) \\
&\supseteq \qquad \underbrace{(\cap_i \mathcal{B}'_{1,i} \setminus \{\|\mathbf{z}\|_2 \le R_1\})}_{\mathbf{z}\text{ nearly orthogonal to }\hat{\mathbf{a}}_{r+i}\text{ while }\|\mathbf{z}\|_2\text{ not too small}} \qquad \setminus \qquad \underbrace{\cup_j \big((\mathcal{B}'_{2,j})^c \cup \{\|\mathbf{z}\|_2 \ge R_2\}\big)}_{\mathbf{z}\text{ nearly orthogonal to }\hat{\mathbf{a}}_j\text{ for some }j\text{ or }\|\mathbf{z}\|_2\text{ too large}} \qquad .
\end{aligned}$$

Set $\mathcal{A} = \cap_{i\in[k]} \mathcal{B}'_{1,i}$. Since $\mathcal{A} \setminus \{\|\mathbf{z}\|_2 \le R_1\} = \mathcal{A} \setminus (\{\|\mathbf{z}\|_2 \le R_1\} \cap \mathcal{A}) \supseteq \mathcal{A} \setminus (\{\|\mathrm{proj}_S\, \mathbf{z}\|_2 \le R_1\} \cap \mathcal{A})$:

$$\begin{aligned}
\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}} &\supseteq \Big(\mathcal{A} \setminus (\{\|\mathrm{proj}_S\, \mathbf{z}\|_2 \le R_1\} \cap \mathcal{A})\Big) \setminus \bigcup_j \Big((\mathcal{B}'_{2,j})^c \cup \{\|\mathbf{z}\|_2 \ge R_2\}\Big) \\
&= \mathcal{A} \setminus \Big(\cup_{j\in[r]}((\mathcal{B}'_{2,j})^c \cap \mathcal{A}) \bigcup (\{\|\mathrm{proj}_S\, \mathbf{z}\|_2 \le R_1\} \cap \mathcal{A}) \bigcup (\{\|\mathbf{z}\|_2 \ge R_2\} \cap \mathcal{A})\Big),
\end{aligned}$$

which implies

$$\begin{aligned}
&\mathbb{P}[\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}] \\
&\qquad \ge \mathbb{P}[\mathcal{A}] - \sum_{j\in[r]} \mathbb{P}[(\mathcal{B}'_{2,j})^c, \mathcal{A}] - \mathbb{P}[\{\|\mathrm{proj}_S\, \mathbf{z}\|_2 \le R_1\}, \mathcal{A}] - \mathbb{P}[\{\|\mathbf{z}\|_2 \ge R_2\}, \mathcal{A}].
\end{aligned} \tag{3.3.10}$$

We now bound the terms in (3.3.10). First,

$$\mathbb{P}[(\mathcal{B}'_{2,j})^c, \mathcal{A}] = \mathbb{P}\big[|\langle \mathbf{z}, \hat{\mathbf{a}}_j\rangle| \le R_2 k_l/m^3 \;\big|\; \mathcal{A}\big]\, \mathbb{P}[\mathcal{A}].$$

Notice that when conditioning on the event $|\langle \mathbf{z}, \hat{\mathbf{a}}_{r+i}\rangle| \le R_1\theta$ for $i \in [k]$ we have:

$$|\langle \mathbf{z}, \mathrm{proj}_{S^\perp}\, \hat{\mathbf{a}}_j\rangle| = |\mathbf{z}^\top \hat{\mathbf{A}}_{r+1:} \hat{\mathbf{A}}^\dagger_{r+1:} \hat{\mathbf{a}}_j| \le R_1\sqrt{k}\theta \|\hat{\mathbf{A}}^\dagger_{r+1:} \hat{\mathbf{a}}_j\|_2 \le R_1\sqrt{k}\tau M\theta, \tag{3.3.11}$$

where the first equality comes from the definition of the projection, the second inequality follows from the conditioning, and the last comes from the robust Kruskal rank condition. Furthermore, we notice that $\mathrm{proj}_S\, \hat{\mathbf{a}}_j$ is orthogonal to $\hat{\mathbf{a}}_{r+1}, \ldots, \hat{\mathbf{a}}_{r+k}$ and the conditioning can therefore be dropped after applying (3.3.11):

$$\begin{aligned}\mathbb{P}\big[|\langle \mathbf{z}, \hat{\mathbf{a}}_j\rangle| \leq R_2 k_l/m^3 \;\big|\; \mathcal{A}\big] &\leq \mathbb{P}\big[|\langle \mathbf{z}, \operatorname{proj}_S \hat{\mathbf{a}}_j\rangle| \leq R_2 k_l/m^3 + |\langle \mathbf{z}, \operatorname{proj}_{S^\perp} \hat{\mathbf{a}}_1\rangle| \;\big|\; \mathcal{A}\big] \\ &\leq \mathbb{P}\big[|\langle \mathbf{z}, \operatorname{proj}_S \hat{\mathbf{a}}_j\rangle| \leq R_2 k_l/m^3 + R_1\sqrt{k}\tau M\theta\big] \\ &\leq 2(\sqrt{2\pi}\|\operatorname{proj}_S \hat{\mathbf{a}}_1\|_2)^{-1}(R_2 k_l/m^3 + R_1\sqrt{k}\tau M\theta) \\ &\leq \sqrt{2/\pi}\tau M(R_2 k_l/m^3 + R_1\sqrt{k}\tau M\theta),\end{aligned}$$

where the last two steps follow from bounding the density of a Gaussian distribution from above and the fact that $\{\hat{\mathbf{a}}_j, \hat{\mathbf{a}}_{r+1}, \dots, \hat{\mathbf{a}}_{r+k}\}$ also satisfies the robust Kruskal rank condition so that $\|\operatorname{proj}_S \hat{\mathbf{a}}_j\|_2 \geq (\tau M)^{-1}$.

We use the following bounds for the last two terms in (3.3.10):

$$\mathbb{P}[\|\operatorname{proj}_S \mathbf{z}\|_2 \leq R_1, \mathcal{A}] = \mathbb{P}[\|\operatorname{proj}_S \mathbf{z}\|_2 \leq R_1]\,\mathbb{P}[\mathcal{A}] \leq \mathbb{P}[\mathcal{A}]/2 \quad (\text{set } R_1 = \sqrt{d}/2),$$

$$\begin{aligned}\mathbb{P}[\|\mathbf{z}\|_2 \geq R_2, \mathcal{A}] &= \mathbb{P}[\|\mathbf{z}\|_2 \geq R_2 \mid \mathcal{A}]\,\mathbb{P}[\mathcal{A}] = (1 - \mathbb{P}[\|\mathbf{z}\|_2 \leq R_2 \mid \mathcal{A}])\,\mathbb{P}[\mathcal{A}] \\ &\leq (1 - \mathbb{P}[\|\mathbf{z}\|_2 \leq R_2])\,\mathbb{P}[\mathcal{A}] \quad (\text{Gaussian correlation ineq., Theorem 2.5.2}) \\ &\leq \mathbb{P}[\mathcal{A}]/4 \qquad (\text{Markov's inequality, set } R_2 = 2\sqrt{d}).\end{aligned}$$

Combining the previous estimates we get

$$\mathbb{P}[\mathcal{A}_{1,\mathbf{y}} \cap \mathcal{A}_{2,\mathbf{y}}] \geq \mathbb{P}[\mathcal{A}]\big(1 - 1/2 - r\sqrt{2/\pi}\tau M(R_2 k_l/m^3 + R_1\sqrt{k}\tau M\theta) - 1/4\big).$$

By Lemma 2.5.3, $\mathbb{P}[\mathcal{A}] \geq (R_1\theta/4)^k$. The claim follows. □

At this point, we are ready to analyze the probability of $\mathcal{A}_3$.

Lemma 3.3.9. *In the setting of Lemma 3.3.8, let* $p_2 = p_1 - (\theta\sqrt{d}/8)^k r^2 \tau M(\sqrt{dk}\theta\tau M k_l^{-1} m^3 + \alpha)$. *Then* $\mathbb{P}[\mathcal{A}_3 \cap \mathcal{A}_{1,\mathbf{x}} \cap \mathcal{A}_{2,\mathbf{x}} \mid \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{y}}] \geq p_2$.

Proof. We start with our idea to bound the probability of the “eigenvalue gap”:

$$\left|\frac{\langle \mathbf{x}, \hat{\mathbf{a}}_s\rangle}{\langle \mathbf{y}, \hat{\mathbf{a}}_s\rangle} - \frac{\langle \mathbf{x}, \hat{\mathbf{a}}_t\rangle}{\langle \mathbf{y}, \hat{\mathbf{a}}_t\rangle}\right| \geq \alpha \quad \text{for } s, t \in [r], s \neq t.$$

Since we condition on $|\langle \mathbf{y}, \hat{\mathbf{a}}_i\rangle|$ not being too small for all $i \in [r]$, when further conditioned on $\mathbf{y}$, we have:

$$\mathbb{P}\left[\Big|\frac{\langle \mathbf{x}, \hat{\mathbf{a}}_s\rangle}{\langle \mathbf{y}, \hat{\mathbf{a}}_s\rangle} - \frac{\langle \mathbf{x}, \hat{\mathbf{a}}_t\rangle}{\langle \mathbf{y}, \hat{\mathbf{a}}_t\rangle}\Big| \geq \alpha \;\Big|\; \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{y}}\right] = \mathbb{E}\left[\mathbb{P}\left[\underbrace{\Big|\frac{\langle \mathbf{x}, \hat{\mathbf{a}}_s\rangle}{\langle \mathbf{y}, \hat{\mathbf{a}}_s\rangle} - \frac{\langle \mathbf{x}, \hat{\mathbf{a}}_t\rangle}{\langle \mathbf{y}, \hat{\mathbf{a}}_t\rangle}\Big|}_{\text{denominators are fixed}} \geq \alpha \;\Big|\; \mathbf{y}\right] \;\Big|\; \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{y}}\right].$$

Therefore it is enough to show a uniform lower bound for $\mathbb{P}[|\langle \mathbf{x}, C_s\hat{\mathbf{a}}_s - C_t\hat{\mathbf{a}}_t\rangle| \geq \alpha]$, where $|C_s|, |C_t|$ are in $[1, k_l^{-1}m^3]$. Notice that the set $\{|\langle \mathbf{x}, C_s\hat{\mathbf{a}}_s - C_t\hat{\mathbf{a}}_t\rangle| \geq \alpha\}$ is a type II band, which we denoted by $\mathcal{B}_{3,st}$. Therefore the target event is the intersection of k type I bands $\mathcal{B}_{1,i}$, r type II bands $\mathcal{B}_{2,j}$ and $\binom{r}{2}$ type II bands $\mathcal{B}_{3,st}$. More precisely,

$$\mathbb{P}[\mathcal{A}_3, \mathcal{A}_{1,\mathbf{x}}, \mathcal{A}_{2,\mathbf{x}} \mid \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{y}}] \geq \inf_{|C_s|,|C_t| \in [1, k_l^{-1}m^3]} \mathbb{P}[\cap_{i\in[k]}\mathcal{B}_{1,i}, \cap_{j\in[r]}\mathcal{B}_{2,j}, \cap_{s,t\in[r],s\neq t}\mathcal{B}_{3,st}].$$

We reuse ideas from the proof of Lemma 3.3.8. Write $\mathbf{x} = \mathbf{u}/\|\mathbf{u}\|_2$ with $\mathbf{u}$ being standard Gaussian. Consider the following events for $\mathbf{u}$: $\mathcal{B}'_{1,i} := \{|\langle \mathbf{u}, \hat{\mathbf{a}}_{r+i}\rangle| \leq \sqrt{d}\theta/2\}$, $\mathcal{B}'_{2,j} := \{|\langle \mathbf{u}, \hat{\mathbf{a}}_j\rangle| \geq 2\sqrt{d}k_l/m^3\}$, and $\mathcal{B}'_{3,st} := \{|\langle \mathbf{u}, C_s\hat{\mathbf{a}}_s - C_t\hat{\mathbf{a}}_t\rangle| \geq 2\sqrt{d}\alpha\}$. Set $\mathcal{A} = \cap_{i\in[k]}\mathcal{B}'_{1,i}$. With the concentration of $\|\mathbf{u}\|_2$ in $[\sqrt{d}/2, 2\sqrt{d}]$, the target probability becomes:

$$\begin{aligned}
\mathbb{P}[\cap_{i\in[k]}\mathcal{B}_{1,i}, \cap_{j\in[r]}\,\mathcal{B}_{2,j}, \cap_{s,t\in[r],s\neq t}\mathcal{B}_{3,st}] &\geq \mathbb{P}\Big[\mathcal{A} \setminus \Big((\{\|\mathrm{proj}_S\,\mathbf{u}\|_2 \leq \sqrt{d}/2\} \cap \mathcal{A}) \\
&\quad \bigcup \cup_{j\in[r]}((\mathcal{B}'_{2,j})^c \cap \mathcal{A}) \bigcup (\{\|\mathbf{u}\|_2 \geq 2\sqrt{d}\} \cap \mathcal{A}) \bigcup \cup_{s\neq t\in[r]}((\mathcal{B}'_{3,st})^c \cap \mathcal{A})\Big)\Big] \\
&\geq p_1 - \sum_{s,t\in[r],s\neq t} \mathbb{P}[\mathcal{A}, (\mathcal{B}'_{3,st})^c].
\end{aligned} \tag{3.3.12}$$

Now we consider the summand, which is the intersection of $k+1$ type I bands. Take $s = 1, t = 2$ (the rest is similar) and write $\mathbf{v} = C_1\hat{\mathbf{a}}_1 - C_2\hat{\mathbf{a}}_2 = \mathrm{proj}_S\,\mathbf{v} + \mathrm{proj}_{S^\perp}\,\mathbf{v}$. Then:

$$\begin{aligned}
\mathbb{P}[\mathcal{A}, (\mathcal{B}'_{3,12})^c] &= \mathbb{P}\big[|\langle \mathbf{u}, \mathbf{v}\rangle| \leq 2\sqrt{d}\alpha \;\big|\; \mathcal{A}\big]\,\mathbb{P}[\mathcal{A}] \\
&\leq \mathbb{P}\big[|\langle \mathbf{u}, \mathrm{proj}_S\,\mathbf{v}\rangle| \leq 2\sqrt{d}\alpha + |\langle \mathbf{u}, \mathrm{proj}_{S^\perp}\,\mathbf{v}\rangle| \;\big|\; \mathcal{A}\big]\,\mathbb{P}[\mathcal{A}].
\end{aligned} \tag{3.3.13}$$

When conditioning on $\mathcal{A}$, $\langle \mathbf{u}, \mathrm{proj}_{S^\perp}\,\mathbf{v}\rangle$ is bounded by:

$$\begin{aligned}
|\langle \mathbf{u}, \mathrm{proj}_{S^\perp}\,\mathbf{v}\rangle| &= |\mathbf{u}^\top \hat{\mathbf{A}}_{r+1:}\hat{\mathbf{A}}_{r+1:}^\dagger(C_1\hat{\mathbf{a}}_1 - C_2\hat{\mathbf{a}}_2)| \leq \sqrt{dk}\theta\|\hat{\mathbf{A}}_{r+1:}^\dagger(C_1\hat{\mathbf{a}}_1 - C_2\hat{\mathbf{a}}_2)\|_2/2 \\
&\leq \sqrt{dk}\theta\tau M k_l^{-1}m^3.
\end{aligned} \tag{3.3.14}$$

With (3.3.14), we can drop the conditioning in (3.3.13):

$$\begin{aligned}\mathbb{P}[\mathcal{A}\cap(\mathcal{B}'_{3,12})^c] &\le \mathbb{P}\big[|\langle \mathbf{u}, \operatorname{proj}_S \mathbf{v}\rangle| \le \alpha + \sqrt{dk}\theta\tau M k_l^{-1} m^3\big]\,\mathbb{P}[\mathcal{A}] \\ &\le 2(\alpha + \sqrt{dk}\theta\tau M k_l^{-1} m^3)/(\sqrt{2\pi}\|\operatorname{proj}_S \mathbf{v}\|_2)\,\mathbb{P}[\mathcal{A}] \qquad (3.3.15) \\ &\le 2\tau M(\alpha + \sqrt{dk}\theta\tau M k_l^{-1} m^3)\,\mathbb{P}[\mathcal{A}].\end{aligned}$$

The last inequality holds because the set $\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \hat{\mathbf{a}}_{r+1}, \dots, \hat{\mathbf{a}}_{r+k}\}$ satisfies the robust Kruskal rank condition, and thus

$$\|\operatorname{proj}_S \mathbf{v}\|_2 = \|C_1\hat{\mathbf{a}}_1 - C_2\hat{\mathbf{a}}_2 - \hat{\mathbf{A}}_{r+1:}\hat{\mathbf{A}}^{\dagger}_{r+1:}\mathbf{v}\|_2 \ge (\tau M)^{-1}\sqrt{C_1^2 + C_2^2 + \|\hat{\mathbf{A}}^{\dagger}_{r+1:}\mathbf{v}\|_2^2} \ge \sqrt{2}(\tau M)^{-1}.$$

The combination of Lemma 2.5.3 and (3.3.12) and (3.3.15) gives the desired probability. □

Finally, we are in a place to give the probability that all the events are true for $\mathbf{x}, \mathbf{y}$:

Lemma 3.3.10. *In the setting of Lemma 3.3.8,* $\mathbb{P}[\mathcal{A}_{1,\mathbf{x}}, \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{x}}, \mathcal{A}_{2,\mathbf{y}}, \mathcal{A}_3] \ge p_1 p_2$. *In particular, the choices* $k_l = \sqrt{2\pi}\tau^{-1}M^{-1}m^3 r^{-1} d^{-1/2}/64$, $\alpha = \tau^{-1}M^{-1}r^{-2}/16$ *and* $\theta(r\sqrt{dk}\tau^2 M^2 + 64r^3\tau^3M^3 d\sqrt{k}/\sqrt{2\pi}) \le 1/16$ *imply that* $\mathbb{P}[\mathcal{A}_{1,\mathbf{x}}, \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{x}}, \mathcal{A}_{2,\mathbf{y}}, \mathcal{A}_3] \ge \big(\theta\sqrt{d}/8\big)^{2k}/256$.

Proof. The first part follows by combining Lemmas 3.3.8 and 3.3.9. For the second part, since $p_2 \le p_1$, the claim follows by using our choices in $\mathbb{P}[\mathcal{A}_{1,\mathbf{x}}, \mathcal{A}_{1,\mathbf{y}}, \mathcal{A}_{2,\mathbf{x}}, \mathcal{A}_{2,\mathbf{y}}, \mathcal{A}_3] \ge p_2^2$. □

At this point, we finished the analysis of the randomness in the first partial tensor decomposition, to recover the first r components. In the next lemma we give the probability that random vectors $\mathbf{x}', \mathbf{y}'$ satisfy the assumptions of Lemma 3.3.6. The events we are analyzing are:

$$\begin{aligned}\mathcal{A}'_{2,\mathbf{x}} &= \{\forall i \in [k], |\langle \mathbf{x}', \hat{\mathbf{a}}_{r+i}\rangle| \ge k'_l/m^3\}, \\ \mathcal{A}'_{2,\mathbf{y}} &= \{\forall i \in [k], |\langle \mathbf{y}', \hat{\mathbf{a}}_{r+i}\rangle| \ge k'_l/m^3\}, \text{ and} \\ \mathcal{A}'_3 &= \{\forall i \ne j, i, j \in [k], |\langle \mathbf{x}', \hat{\mathbf{a}}_{r+i}\rangle/\langle \mathbf{y}', \hat{\mathbf{a}}_{r+i}\rangle - \langle \mathbf{x}', \hat{\mathbf{a}}_{r+j}\rangle/\langle \mathbf{y}', \hat{\mathbf{a}}_{r+j}\rangle| \ge \alpha' > 0\}.\end{aligned}$$

Lemma 3.3.11. *Let* $\mathbf{x}', \mathbf{y}'$ *be i.i.d. uniformly random in* $\mathbb{S}^{d-1}$*. For* $\hat{\mathbf{a}}_{r+1}, \dots, \hat{\mathbf{a}}_{r+k}$*, and* $k'_l, \alpha' > 0$*, we have*

$$\mathbb{P}[\mathcal{A}'_{2,\mathbf{x}}, \mathcal{A}'_{2,\mathbf{y}}, \mathcal{A}'_3] \ge (1 - k^2\sqrt{ed}\tau M\alpha' - \sqrt{ed}kk'_l/m^3)(1 - \sqrt{ed}kk'_l/m^3).$$

In particular, the choices $k'_l = m^3k^{-1}d^{-1/2}/(4\sqrt{e})$*,* $\alpha' = \tau^{-1}M^{-1}k^{-2}d^{-1/2}/(4\sqrt{e})$ *imply that* $\mathbb{P}[\mathcal{A}'_{2,\mathbf{x}}, \mathcal{A}'_{2,\mathbf{y}}, \mathcal{A}'_3] \geq 3/8$.

Proof. The first part reuses ideas from the proofs of Lemmas 3.3.8 and 3.3.9. We first separate the intersection of events:

$$\mathbb{P}[\mathcal{A}'_{2,\mathbf{x}} \cap \mathcal{A}'_{2,\mathbf{y}} \cap \mathcal{A}'_3] = \mathbb{P}[\mathcal{A}'_{2,\mathbf{x}} \cap \mathcal{A}'_3 \mid \mathcal{A}'_{2,\mathbf{y}}]P[\mathcal{A}'_{2,\mathbf{y}}] \geq (\mathbb{P}[\mathcal{A}'_3 \mid \mathcal{A}'_{2,\mathbf{y}}] - \mathbb{P}[(\mathcal{A}'_{2,\mathbf{x}})^c])\,\mathbb{P}[\mathcal{A}'_{2,\mathbf{y}}].$$

By Lemma 2.5.1, $\mathbb{P}[\mathcal{A}'_{2,\mathbf{x}}]$ and $\mathbb{P}[\mathcal{A}'_{2,\mathbf{y}}]$ are at least $1 - \sqrt{e}dkk'_l/m^3$. Also

$$\mathbb{P}[(\mathcal{A}'_3)^c \mid \mathcal{A}'_{2,\mathbf{y}}] = \mathbb{E}\left[\mathbb{P}\left[\min_{i \neq j, i,j \in [k]} \left| \frac{\langle \mathbf{x}', \hat{\mathbf{a}}_{r+i} \rangle}{\langle \mathbf{y}', \hat{\mathbf{a}}_{r+i} \rangle} - \frac{\langle \mathbf{x}', \hat{\mathbf{a}}_{r+j} \rangle}{\langle \mathbf{y}', \hat{\mathbf{a}}_{r+j} \rangle} \right| \leq \alpha' \;\middle|\; \mathbf{y}' \right] \;\middle|\; \mathcal{A}'_{2,\mathbf{y}} \right].$$

Consider a uniform upper bound for $\mathbb{P}[\min_{i \neq j, i,j \in [k]} |\langle \mathbf{x}', C'_i\hat{\mathbf{a}}_{r+i} - C'_j\hat{\mathbf{a}}_{r+j} \rangle| \leq \alpha']$, where $|C'_i|, |C'_j|$ are lower bounded by 1. Therefore, again by Lemma 2.5.1, we have

$$\mathbb{P}[(\mathcal{A}'_3)^c | \, \mathcal{A}'_{2,\mathbf{y}}] \leq k(k-1)\sqrt{ed}\tau M\alpha'/(2\sqrt{2}) \leq k^2\sqrt{ed}\tau M\alpha'.$$

Combining everything gives the desired result. The second part follows directly from our choices of k'_l and α'. □

3.3.4. Proof of Theorem 3.2.1. In this subsection we prove Theorem 3.2.1.

Proof of Theorem 3.2.1. Without loss of generality, assume π is the identity, and assume for a moment that ε_{in}, θ are small enough so that: (1) the assumptions of Lemmas 3.3.5 and 3.3.7 are satisfied; and (2) $\varepsilon_{3.3.4}$ and $\varepsilon_{3.3.6}$ are smaller than 1 so that we can replace $\varepsilon^2_{3.3.4}$ and $\varepsilon^2_{3.3.6}$ by $\varepsilon_{3.3.4}$ and $\varepsilon_{3.3.6}$ in the expression of $\varepsilon_{3.3.5}$ and $\varepsilon_{3.3.7}$. We trace the error propagation backwards and show how we can reach ε accuracy for the algorithm to terminate while achieving non-negligible success probability per iteration. The reconstruction error is bounded with Lemmas 3.3.4 to 3.3.7:

$$\begin{aligned} \|\boldsymbol{\mathcal{T}}' - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} &\leq \|\tilde{\boldsymbol{\mathcal{T}}} - \boldsymbol{\mathcal{T}}\|_{\mathrm{F}} + \sum_{i \in [r+k]} \|\mathbf{a}_i^{\otimes 3} - \xi_i \tilde{\mathbf{a}}_i^{\otimes 3}\|_{\mathrm{F}} \\ &\leq \varepsilon_{in} + \sum_{i \in [r+k]} \big|\|\mathbf{a}_i\|_2^3 - s_i\xi_i\big| \|\tilde{\mathbf{a}}_i^{\otimes 3}\|_{\mathrm{F}} + \|\hat{\mathbf{a}}_i^{\otimes 3} - s_i^3\tilde{\mathbf{a}}_i^{\otimes 3}\|_{\mathrm{F}}\|\mathbf{a}_i\|_2^3 \qquad (3.3.16) \\ &\leq \varepsilon_{in} + 3rM^3\varepsilon_{3.3.4} + r\varepsilon_{3.3.5} + 3kM^3\varepsilon_{3.3.6} + k\varepsilon_{3.3.7}. \end{aligned}$$

Collecting the results from Lemmas 3.3.4 to 3.3.7, we have:

$$\begin{aligned} \varepsilon_{3.3.4} &= O\big(\tau^4 M^{10} k r^{5/2} k_l^{-2} \alpha^{-1}(\varepsilon_{in} + \theta)\big) & \varepsilon_{3.3.6} &= O(\tau^4 M^7 k^{5/2} r k_l'^{-2} \alpha'^{-1} \varepsilon_{3.3.5}) \\ \varepsilon_{3.3.5} &= O(M^3 m^3 r k_l^{-1} \varepsilon_{3.3.4}) & \varepsilon_{3.3.7} &= O(M^3 m^3 k k_l'^{-1} \varepsilon_{3.3.6}), \end{aligned}$$

which can be written in terms of ε_{in} and θ:

(3.3.17)

$$\begin{aligned} \varepsilon_{3.3.4} &= O\big(\tau^7 M^{13} m^{-6} k r^{13/2} d(\varepsilon_{in} + \theta)\big) & \varepsilon_{3.3.6} &= O\big(\tau^{13} M^{25} m^{-12} k^{11/2} r^{19/2} d^3(\varepsilon_{in} + \theta)\big) \\ \varepsilon_{3.3.5} &= O\big(\tau^8 M^{17} m^{-6} k r^{17/2} d^{3/2}(\varepsilon_{in} + \theta)\big) & \varepsilon_{3.3.7} &= O\big(\tau^{13} M^{28} m^{-12} k^{13/2} r^{19/2} d^{7/2}(\varepsilon_{in} + \theta)\big). \end{aligned}$$

Equation (3.3.17) implies the reconstruction error is bounded by

$$\|\boldsymbol{\mathcal{T}}' - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} = O\big(\tau^{13} M^{28} m^{-12} k^{15/2} r^{19/2} d^{7/2}(\varepsilon_{in} + \theta)\big).$$

This gives a polynomial $q(d, r, k, \tau, M, m^{-1}) = \Theta(\tau^{13} M^{28} m^{-12} k^{15/2} r^{19/2} d^{7/2})$, increasing in every argument, such that if we request that $\varepsilon_{in} \le \varepsilon / q(d, r, k, \tau, M, m^{-1})$ and we set $\theta = \varepsilon / q(d, r, k, \tau, M, m^{-1})$, then $\|\boldsymbol{\mathcal{T}}' - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} \le \varepsilon$ (the first termination condition). With this choice: (1) the assumptions of Lemma 3.3.10 are satisfied; (2) for each iteration, with positive probability the events in Lemmas 3.3.10 and 3.3.11 happen; and (3) we can take $\varepsilon_{3.3.4} = \Theta(\tau^{-6} M^{-15} m^6 r^{-3} d^{-5/2} \varepsilon)$, $\varepsilon_{3.3.6} = \Theta(M^{-3} k^{-4} d^{-1/2} \varepsilon)$ and they satisfy the assumptions of Lemmas 3.3.5 and 3.3.7, respectively.

Now we argue that the second termination condition, $\max_{i \in [r+k]} |\xi_i|^{1/3} \le 2M$, holds when the events in Lemmas 3.3.10 and 3.3.11 happen. Notice that at this point $|\xi_i|$ is close to $\|\mathbf{a}_i\|_2^3$. Without loss of generality, assume $\max_{i \in [r+k]} |\xi_i|^{1/3} = |\xi_1|^{1/3}$. Since $\forall x, y > 0$, $|y^{1/3} - x^{1/3}| \le y^{-2/3}|y - x|$, we have $\big|\|\mathbf{a}_i\|_2 - |\xi_i|^{1/3}\big| \le \|\mathbf{a}_i\|_2^{-2} \big|\|\mathbf{a}_i\|_2^3 - |\xi_i|\big|$, for all $i \in [r + k]$, which implies

$$|\xi_1|^{1/3} \le \|\mathbf{a}_1\|_2 + \big|\|\mathbf{a}_1\|_2 - |\xi_1|^{1/3}\big| \le \|\mathbf{a}_1\|_2 + \|\mathbf{a}_1\|_2^{-2} \varepsilon \le M + m \le 2M,$$

where the second inequality comes from $\varepsilon_{3.3.5} \le \varepsilon$ and the third inequality comes from $\varepsilon \le \varepsilon_{out} \le m^3$. Therefore, the algorithm terminates with a $2M$-bounded decomposition with reconstruction error at most ε.

Set

$$\operatorname{poly}_{3.2.1}(d,\tau,M) = 2\operatorname{poly}_{3.3.2}(2d,\tau,M,2M,d) \geq 2\operatorname{poly}_{3.3.2}(r+k,\tau,M,2M,d),$$

$$\operatorname{poly}'_{3.2.1}(d,\tau,M,m^{-1}) = q(d,d,d,\tau,M,m^{-1})\operatorname{poly}_{3.2.1} \geq q(d,r,k,\tau,M,m^{-1})\operatorname{poly}_{3.2.1}.$$

When the algorithm terminates, we have

$$\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}}'\|_{\mathrm{F}} \leq \varepsilon + \varepsilon_{in} \leq \varepsilon + \frac{\varepsilon}{q} \leq \frac{\varepsilon_{out}}{\operatorname{poly}_{3.2.1}} + \frac{\varepsilon_{out}}{q\operatorname{poly}_{3.2.1}} \leq \frac{\varepsilon_{out}}{\operatorname{poly}_{3.3.2}(r+k,\tau,M,2M,d)}. \tag{3.3.18}$$

Thus, we can apply Corollary 3.3.2 and obtain component-wise ε_{out} accuracy.

For the running time, in each iteration, steps 3 and 7 run in time $\operatorname{poly}(d,\varepsilon^{-1},\tau,M,m^{-1})$. Least squares steps 4 and 8 and the rest take $\operatorname{poly}(d)$ time. By Lemmas 3.3.10 and 3.3.11, the success probability per iteration is at least $3\big(\theta\sqrt{d}/8\big)^{2k}/2^{11}$, which implies that the expected number of iterations is at most $2^{11}(\theta\sqrt{d}/8)^{-2k}/3$ and the expected running time is at most $\operatorname{poly}(d^k,\varepsilon^{-k},\tau^k,M^k,m^{-k})$. Since $\varepsilon = \varepsilon_{out}/\operatorname{poly}_{3.2.1}$, the expected running time is also at most $\operatorname{poly}(d^k,1/\varepsilon_{out}^k,\tau^k,M^k,m^{-k})$. □

3.4. Numerical Simulations

In this section, we report some numerical simulation results of Algorithm 2 to verify its correctness and study its feasibility. We implement and run Algorithm 2 on synthetic data generated as follows: For the given dimension d and the overcomplete parameter r, we first randomly sample a matrix $\mathbf{A} = [\mathbf{a}_1,\ldots,\mathbf{a}_{d+r}] \in \mathbb{R}^{d\times(d+r)}$ with i.i.d. entries from $\mathcal{N}(0,1)$, then a random symmetric tensor $\boldsymbol{\mathcal{T}}_{in}$ is constructed using the columns of $\mathbf{A}$: $\boldsymbol{\mathcal{T}}_{in} = \sum_{i\in[d+r]}\mathbf{a}_i^{\otimes 3}$. Note that the actual implementation has a different termination condition from Algorithm 2: the algorithm terminates with a success if the relative error between the output tensor $\boldsymbol{\mathcal{T}}'$ and input $\boldsymbol{\mathcal{T}}$, $\|\boldsymbol{\mathcal{T}}'-\boldsymbol{\mathcal{T}}\|_{\mathrm{F}}/\|\boldsymbol{\mathcal{T}}\|_{\mathrm{F}}$ is smaller than a threshold ε, while it terminates with a failure if the number of re-initializations reaches a threshold N.

Algorithm 2 is applied to random tensors of different $d \in \{10,20,30,40,50,60\}$ and $r \in \{1,2,3,4\}$. For each (d,r) pair, 50 random tensors are generated and the results are the average on them. In experiments, we set $\varepsilon = 0.1$ and $N = 10000$. Fig. 3.4 shows the rate of success of Algorithm 2 on each (d,r) pair. We see that, even in the mildly overcomplete settings, increasing d or r significantly reduces the rate of success.

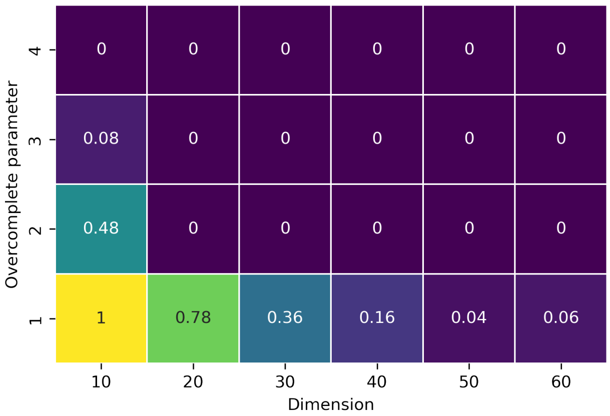


Figure 3.4. Rate of success on random Gaussian inputs. Horizontal axis indicates the dimension d, and the vertical axis indicates the overcomplete parameter r. The number in each cell indicates the rate of success of Algorithm 2 over 50 random tensor in $\mathbb{R}^d$ with $d + r$ Gaussian components.

We summarize the running time and relative error results for the cases with a non-zero rate of success in our experiments in Table 3.1. Though we see successful cases when d, r are small, which verifies the correctness of Algorithm 2 and Theorem 3.2.1, as d, r increases, we see an increase in the running time and relative error. The rapidly-decreasing rate of success makes Algorithm 2 impractical, which can also be partially recognized in our analysis: Equation (3.3.17) suggests that polynomials $\text{poly}_{3.2.1}, \text{poly}'_{3.2.1}$ are actually of very high degree in each of its inputs.

Parameters		Results			
		Running time (s)		Relative error	
d	r	Successful runs	All runs	Successful runs	All runs
60	1	183.6719	259.3723	0.0528	0.2450
50	1	127.9894	128.3883	0.0903	0.2252
40	1	57.3451	62.2241	0.0668	0.1667
30	1	23.6430	26.8917	0.0631	0.1201
20	1	10.3054	11.0066	0.0594	0.0725
10	1	1.1336	1.1336	0.0326	0.0326
10	2	6.4174	6.8246	0.0724	0.1084
10	3	7.2966	7.3902	0.0799	0.1697

Table 3.1. Running time and relative error for the cases with a non-zero rate of success in Fig. 3.4. The results are averaged on successfully decomposed tensors in the column “Successful runs” and on all tensors in the column “All runs”.

CHAPTER 4

Blind Deconvolution and Gaussian Mixture Models

In this chapter we study two problems in data sciences that motivate and are direct applications of Algorithm 2: *blind deconvolution* and *GMM parameter estimation.*

A *deconvolution* problem can be formulated as follows: We have a d-dimensional random vector

$$\mathbf{x} = \mathbf{z} + \boldsymbol{\eta} \tag{4.0.1}$$

where $\mathbf{z}$ and $\boldsymbol{\eta}$ are independent random vectors. Given samples from $\mathbf{x}$, the goal is to determine the distribution of $\mathbf{z}$. It is called *deconvolution* because the distribution of $\mathbf{x}$ is the convolution of the distributions of $\mathbf{z}$ and $\boldsymbol{\eta}$. In principle, $\mathbf{z}$ can be seen as the signal we are interested in while $\boldsymbol{\eta}$ is some random noise. *Blind deconvolution* tries to answer the following question:

Can we recover the distribution of $\mathbf{z}$ *when the distribution of* $\boldsymbol{\eta}$ *is unknown?*

The following *mixture model parameter estimation problem* can be recast as a blind deconvolution problem: Let $\mathbf{x}$ be a d-dimensional random vector distributed as the following mixture model: First sample i from $[d]$, each value with probability w_i ($w_i > 0$, $\sum_i w_i = 1$), then let $\mathbf{x} = \boldsymbol{\mu}_i + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is a given d-dimensional random vector and $\boldsymbol{\mu}_i \in \mathbb{R}^d$. The estimation problem is to estimate $\boldsymbol{\mu}_i$s and w_is from samples of $\mathbf{x}$. It is a deconvolution problem $\mathbf{x} = \mathbf{z} + \boldsymbol{\eta}$ when $\mathbf{z}$ follows the discrete distribution equal to $\boldsymbol{\mu}_i$ with probability w_i and is blind when the distribution of $\boldsymbol{\eta}$ is unknown.

The *GMM parameter estimation problem* can be described as follows: Let $\mathbf{x} \in \mathbb{R}^d$ be a random vector with density function $\mathbf{x} \mapsto \sum_{i\in[k]} w_i f_i(\mathbf{x})$ where $w_i > 0$, $\sum_{i\in[k]} w_i = 1$ and f_i is the Gaussian density function with mean $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d\times d}$. GMM parameter estimation is the following algorithmic question:

Given i.i.d. samples from $\mathbf{x}$*, suppose* k *is known, can we estimate* w_i*s,* $\boldsymbol{\mu}_i$*s and* $\boldsymbol{\Sigma}_i$*s?*

Specifically, the GMM parameter estimation problem is a deconvolution problem when the covariance matrices of the components are the same, namely $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$. In such cases, the problem is recast as

$\mathbf{x} = \mathbf{z} + \boldsymbol{\eta}$ where $\mathbf{z}$ follows a discrete distribution taking value $\boldsymbol{\mu}_i$ with probability w_i, $i = 1, \ldots, k$ and $\boldsymbol{\eta}$ is Gaussian with mean 0 and covariance $\boldsymbol{\Sigma}$. It is blind if $\boldsymbol{\Sigma}$ is unknown.

Inspired by our tensor decomposition algorithm (Algorithm 2), we are interested in a specific family of $\boldsymbol{\eta}$ which contains Gaussian distribution $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$:

Can we recover the distribution of $\mathbf{z}$ *when* $\boldsymbol{\eta}$ *has zero 1st, 3rd moments and finite 6th moment?*

A practitioner may use the expectation-maximization(EM) algorithm [**DLR77**], [**HTF09**, Section 8.5] for parameter estimation. However due to the non-convex nature of the objective function, the EM algorithm is not favored by the theorists, while the tensor structure in the higher order moments of GMMs implied guaranteed and efficient parameter estimation via tensor decomposition [**HK13**, **GVX14**, **BCMV14**, **ABG$^+$14**, **AGH$^+$14**, **AGJ15**, **GHK15**].

The intuition of applying tensor decomposition is based on the method of moments: The distribution of $\mathbf{z}$ is identifiable from its 3rd moment tensor

$$\mathbb{E}[\mathbf{z}^{\otimes 3}] = \sum_{i=1}^{k} w_i \boldsymbol{\mu}_i^{\otimes 3}$$

if the decomposition of $\mathbb{E}[\mathbf{z}^{\otimes 3}]$ coincides with the right hand side. In such cases, decomposing $\mathbb{E}[\mathbf{z}^{\otimes 3}]$ leads to the recovery of $w_i^{1/3}\boldsymbol{\mu}_i$s, which will further be decoupled and the distribution of $\mathbf{z}$ can be recovered. To obtain $\mathbb{E}[\mathbf{z}^{\otimes 3}]$, since $\mathbb{E}[\boldsymbol{\eta}^{\otimes 3}] = 0$, one can exploit moment structures such as:

$$\mathbb{E}[\mathbf{x}^{\otimes 3}] = \mathbb{E}[\mathbf{z}^{\otimes 3}] + (\mathbb{E}[\mathbf{z} \otimes \boldsymbol{\eta} \otimes \boldsymbol{\eta}] + \mathbb{E}[\boldsymbol{\eta} \otimes \mathbf{z} \otimes \boldsymbol{\eta}] + \mathbb{E}[\boldsymbol{\eta} \otimes \boldsymbol{\eta} \otimes \mathbf{z}]). \tag{4.0.2}$$

Notice that the second term in (4.0.2) can be seen as the sum of $\mathbb{E}[\mathbf{x}] \otimes \boldsymbol{\Sigma}$ and its two different "transposed" tensors, which suggests that when $\boldsymbol{\Sigma}$ is known, (4.0.2) can be used to compute $\mathbb{E}[\mathbf{z}^{\otimes 3}]$. When $\boldsymbol{\Sigma}$ is not given, some special cases are solved. For example, in [**HK13**] the authors showed that when $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma \mathbf{I}_d)$, one has that

$$\mathbb{E}[\mathbf{z}^{\otimes 3}] = \mathbb{E}[\mathbf{x}^{\otimes 3}] - \sigma^2 \sum_{i \in [d]} (\mathbb{E}[\mathbf{x}] \otimes \mathbf{e}_i \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbb{E}[\mathbf{x}] \otimes \mathbf{e}_i + \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbb{E}[\mathbf{x}]),$$

where $\mathbf{e}_i$ is the i-th canonical basis vector in $\mathbb{R}^d$, and that σ^2 is the smallest eigenvalue of $\mathrm{Cov}(\mathbf{x})$. However, the challenges of our problem come with the *unknown* and possibly *non-spherical* covariance of $\boldsymbol{\eta}$: it is not obvious anymore to apply moment structures like (4.0.2) and our contribution is a formulation that involves an overcomplete tensor decomposition and uses our overcomplete tensor decomposition algorithm in an essential way.

Organization of the chapter. In Section 4.1, we will revisit the existing works on blind deconvolution and GMMs, and present our high level results. In Sections 4.2 and 4.3, we will present our algorithms on blind deconvolution of discrete distribution and the parameter estimation of GMMs and prove their correctness.

4.1. Introduction

Blind deconvolution-type problems have a long history in signal processing and specifically in image processing as a deblurring technique (see, e.g., [**LWDF11**]). The idea of using higher order moments in blind identification problems is standard too in signal processing, specifically in Independent Component Analysis (see e.g., [**Car91, CJ10**]). Our model (4.0.1) is somewhat different but very natural and inspired by mixture models.

With respect to GMMs, we are interested in parameter estimation in high dimension with no separation assumption (i.e., the means $\boldsymbol{\mu}_i$ can be arbitrarily close). Among the most relevant results in this context we have the following polynomial time algorithms: [**HK13**], for linearly independent means and spherical components (each $\boldsymbol{\Sigma}_i$ is a multiple of the identity); [**ABG$^+$14**], for $O(d^c)$ components with identical and known covariance $\boldsymbol{\Sigma}$; [**BCMV14**], for $O(d^c)$ components with each $\boldsymbol{\Sigma}_i$ being diagonal in the smoothed analysis sense; [**GVX13**, Section 7], [**GVX14**], for linearly independent means and spherical components in the presence of Gaussian noise; and [**GHK15**], for a general GMM with $O(\sqrt{d})$ components in the sense of smoothed analysis. Our algorithm expands the family of GMMs for which efficient parameter estimation is possible. It does not require prior knowledge of the covariance matrix unlike [**ABG$^+$14**] and can handle more components (d components) than [**GHK15**] at the price of assuming that all covariance matrices are identical. With respect to recent results on clustering-based algorithms [**DHKK20, JV19**], we consider these works incomparable to ours since clustering-based algorithms typically require some separation assumptions in the parameters.

Our contributions. We provide an efficient algorithm for the following blind deconvolution problem:

Claim 4.1.1 (Informal statement of Theorem 4.2.2). *Let* $\mathbf{x} = \mathbf{z} + \boldsymbol{\eta}$ *be a random vector as in* (4.0.1)*, where* $\mathbf{z}$ *is a* d*-dimensional discrete distribution supported on* d *points and* $\boldsymbol{\eta}$ *has zero mean, zero 3rd moment and finite 6th moment. Suppose* $\mathbf{z}$ *satisfies a natural non-degeneracy condition (Assumption 4.2.1). Then there is an randomized algorithm that, with probability* $1 - \delta$ *over the randomness in the samples, recovers* $\mathbf{z}$ *within* ε *error. The expected running time and sample complexity are polynomial in* $d, \varepsilon^{-1}, \delta^{-1}$ *and natural condition parameters.*

Equivalently, it can solve the mixture model parameter estimation problem above under the same conditions (Algorithm 3 and Theorem 4.2.2). We show an efficient algorithm for the following GMM parameter estimation problem:

Claim 4.1.2 (Informal statement of Theorem 4.3.1). *Given samples from a* d*-dimensional mixture of* d *identical and not necessarily spherical Gaussians with unknown parameters* w_i*,* $\boldsymbol{\mu}_i$*,* $\boldsymbol{\Sigma}$ *satisfying a natural non-degeneracy condition (Assumption 4.2.1), there is an randomized algorithm that with probability* $1 - \delta$ *over the randomness in the samples, estimates all parameters within* ε *error. The expected running time and sample complexity are polynomial in* $d, \varepsilon^{-1}, \delta^{-1}$ *and natural conditioning parameters.*

4.2. Blind Deconvolution of Discrete Distribution

In this section we provide an application of Algorithm 2: To perform blind deconvolution of an additive mixture model of the form

$$\mathbf{x} = \mathbf{z} + \boldsymbol{\eta} \tag{4.2.1}$$

in $\mathbb{R}^d$, where $\mathbf{z}$ follows a discrete distribution that takes value $\boldsymbol{\mu}_i$ with probability w_i for $i \in [d]$, and $\boldsymbol{\eta}$ is an unknown random variable independent of $\mathbf{z}$ with zero mean, zero 3rd moment and finite 6th moment. Our goal is to recover the parameters of $\mathbf{z}$ when given samples from $\mathbf{x}$. By estimating the overall mean and translating the samples we can, without loss of generality, assume that $\mathbb{E}[\mathbf{x}] = \sum_{i\in[d]} w_i \boldsymbol{\mu}_i = 0$ for the rest of this section.

We will see that, under a natural non-degeneracy condition, Assumption 4.2.1, the parameters of $\mathbf{z}$ are identifiable from the 3rd cumulant of $\mathbf{x}$ as the first and third moments of $\boldsymbol{\eta}$ are zero. Let $K_m(\mathbf{x})$ be the m-th cumulant of $\mathbf{x}$. By properties of cumulants (see Section 2.4):

$$K_3(\mathbf{x}) = K_3(\mathbf{z}) + K_3(\boldsymbol{\eta}) = \sum_{i\in[d]} w_i \boldsymbol{\mu}_i^{\otimes 3}. \tag{4.2.2}$$

If the symmetric decomposition of $\boldsymbol{\mathcal{T}} = K_3(\mathbf{x})$ coincides with (4.2.2), then the function $w_i^{1/3}\boldsymbol{\mu}_i$ of the centers $\boldsymbol{\mu}_i$ and the mixing weights w_i is identifiable. However the component vectors satisfy $\sum_i w_i \boldsymbol{\mu}_i = 0$ (they are *always* linearly dependent) and therefore applying the simultaneous diagonalization algorithm naively has no guarantee.[1] We show that, under the following non-degeneracy condition, our overcomplete tensor decomposition algorithm (Algorithm 2) works successfully.

Assumption 4.2.1. K-rank$_\tau([\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_d]) = d - 1$.

Remark 4. *Note that at this point we are working with a centered mixture ($\sum_{i\in[d]} w_i \boldsymbol{\mu}_i = 0$) and thus the assumption is on the centered mixture. Note also that if $\mathbf{x} = \mathbf{z} + \boldsymbol{\eta}$ is a not necessarily centered mixture, the assumption is satisfied automatically by the centered version of $\mathbf{x}$ when $\mathbf{z}$ has affinely independent support.*

Under Assumption 4.2.1, we can decompose (4.2.2) with Algorithm 2. To distinguish the direction and the scaling factor of component vectors, we reformulate the problem: let $\mathbf{a}_i = \hat{\boldsymbol{\mu}}_i$, and $\rho_i = \|\boldsymbol{\mu}_i\|_2$, our goal becomes to decompose $\boldsymbol{\mathcal{T}} = \sum_{i\in[d]} w_i \rho_i^3 \mathbf{a}_i^{\otimes 3}$ subject to $\sum_{i\in[d]} w_i = 1$ and $\sum_{i\in[d]} w_i \rho_i \mathbf{a}_i = 0$.

We now state our algorithm (Algorithm 2) for blind deconvolution of discrete distribution.

[1] Note that even when the overall mean is non-zero and the means are linearly independent, $\boldsymbol{\mathcal{T}}$ still has linearly dependent components as it is the *central* 3rd moment. If one does not use $\boldsymbol{\mathcal{T}}$, then one loses (4.2.2).

Algorithm 3 Blind deconvolution of discrete distribution

```
   Inputs: i.i.d. samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ from mixture $\mathbf{x}$, error tolerance $\varepsilon'$, upper bound $\rho_{max}$ on $\|\boldsymbol{\mu}_i\|_2$ for $i \in [d]$, lower bound $w_{min}$ on $w_i$ for $i \in [d]$, robust Kruskal rank threshold $\tau$.
1: compute the sample 3rd cumulant $\tilde{\boldsymbol{\mathcal{T}}}$ using Proposition 2.4.4;
2: invoke Algorithm 2 with error tolerance $\varepsilon_{4.2.2} = \varepsilon' / \operatorname{poly}_{4.2.2}$, tensor rank $d$ and overcompleteness 1 to decompose $\tilde{\boldsymbol{\mathcal{T}}}$, thus obtain $\mathbf{a}'_i$, the estimated version of $w_i^{1/3}\boldsymbol{\mu}_i$, for $i \in [d]$;
3: set $\tilde{\mathbf{v}}$ to the right singular vector associated with the minimum singular value of $\mathbf{A}' = [\mathbf{a}'_1, \ldots, \mathbf{a}'_d]$;
4: set $\tilde{\mathbf{w}} := [\tilde{w}_1, \ldots, \tilde{w}_d] = \tilde{\mathbf{v}}^{3/2} / (\sum_{i \in [d]} \tilde{\mathbf{v}}_i^{3/2})$, $\tilde{\boldsymbol{\mu}}_i = \tilde{w}_i^{-1/3}\mathbf{a}'_i$ for $i \in [d]$;
   Outputs: estimated mixing weights $\tilde{w}_1, \ldots, \tilde{w}_d$, and estimated means $\tilde{\boldsymbol{\mu}}_1, \ldots, \tilde{\boldsymbol{\mu}}_d$.
```

Theorem 4.2.2 (Correctness of Algorithm 3). *Let* $\mathbf{x} = [X_1, \ldots, X_d] = \mathbf{z} + \boldsymbol{\eta}$ *be a random vector as in* (4.2.1) *satisfying Assumption 4.2.1. Assume* $0 < w_{min} \le \min_{i \in [d]} w_i$*,* $\rho_{max} \ge \max_{i \in [d]} \rho_i$*,* $0 < \rho_{min} \le \min_{i \in [d]} \rho_i$*,* $0 < \varepsilon' \le \min\{1, w_{min}\rho_{min}^3\}$ *and* $\delta \in (0,1)$*. There exists a polynomial* $\operatorname{poly}_{4.2.2}(d, \tau, \rho_{max}, w_{min}^{-1})$ *such that if* $\varepsilon_{4.2.2} = \varepsilon' / \operatorname{poly}_{4.2.2}$*, then given* N *i.i.d. samples of* $\mathbf{x}$*, with probability* $1 - \delta$ *over the randomness in the samples, Algorithm 3 outputs* $\tilde{\boldsymbol{\mu}}_1, \ldots, \tilde{\boldsymbol{\mu}}_d$ *and* $\tilde{w}_1, \ldots, \tilde{w}_d$ *such that for some permutation* π *of* $[d]$ *and for all* $i \in [d]$ *we have* $\|\boldsymbol{\mu}_{\pi(i)} - \tilde{\boldsymbol{\mu}}_i\|_2 \le \varepsilon'$ *and* $|w_{\pi(i)} - \tilde{w}_i| \le \varepsilon'$*. The expected running time over the randomness of Algorithm 2 is at most* $\operatorname{poly}(d, \varepsilon'^{-1}, \delta^{-1}, \tau, \rho_{max}, \rho_{min}^{-1}, w_{min}^{-1}, \max_i \mathbb{E}[X_i^6])$ *and will use* $N = \Omega\big(\varepsilon'^{-2}\delta^{-1}d^{11} \max_{i \in [d]} \mathbb{E}[X_i^6]\big(\operatorname{poly}'_{3.2.1}(d, \tau, \rho_{max}, w_{min}^{-1/3}\rho_{min}^{-1})\big)^2\big)$ *samples.*

The proof of Theorem 4.2.2 has three parts. The first part is about the tensor decomposition. Note that, assuming $\|\tilde{\boldsymbol{\mathcal{T}}} - \boldsymbol{\mathcal{T}}\|_{\mathrm{F}}$ is small enough, Theorem 3.2.1 guarantees that we can recover $\tilde{\mathbf{a}}_i$ approximately in the direction of $\mathbf{a}_{\pi(i)}$ and ξ_i close to $w_{\pi(i)}\rho_{\pi(i)}^3$ for some permutation π. However we are not finished yet as our goal is to recover both the centers and the mixing weights. Therefore in the second part we need to decouple w_i and ρ_i from $w_i\rho_i^3$, which corresponds to steps 3 and 4 in Algorithm 3. The correctness of these steps are shown in Theorem 4.2.3. At last, we show that the 3rd cumulant can be estimated to within ε accuracy, i.e., $\|\tilde{\boldsymbol{\mathcal{T}}} - \boldsymbol{\mathcal{T}}\|_{\mathrm{F}} \le \varepsilon$, with polynomially many samples for any $\varepsilon > 0$. This follows from a standard argument using k-statistics, shown in Section 2.4 and Lemma 2.4.7.

Decoupling. Before we move on to the proof of Theorem 4.2.2, we show the reader how the mixing weights w_i and the norms ρ_i are decoupled after we decompose the tensor $\tilde{\boldsymbol{\mathcal{T}}}$. As $\mathbb{E}[\mathbf{x}] = 0$, the true parameters satisfy $\sum_{i\in[d]} w_i\rho_i\mathbf{a}_i = 0$, which can be reformulated as a linear system

$$\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}\mathbf{w}^{2/3} = 0, \tag{4.2.3}$$

where $\mathbf{w} = [w_1, \ldots, w_d]$ and $\mathbf{A}$ contains $\mathbf{a}_i$s as columns. To decouple these parameters in the noiseless setting, one only needs to solve this system under the constraint that $\mathbf{w}$ is a probability vector. As $\operatorname{rank}(\mathbf{A}) = d-1$, $\mathbf{w}^{2/3}$ lies in the direction of the right singular vector associated with the only zero singular value. Then $\mathbf{w}$ will be uniquely determined since $\sum_{i\in[d]} w_i = 1$. It is natural then to recover the weights using our approximations to terms in the linear system, namely in the direction of the right singular vector associated to the minimum singular value of $\tilde{\mathbf{A}}\operatorname{diag}(\xi_i)^{1/3}$, where $\tilde{\mathbf{A}} = [\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_d]$. The following theorem guarantees this will work:

Theorem 4.2.3 (Decoupling). *Let $0 < w_{min} \le \min_{i\in[d]} w_i$, and $\rho_{max} \ge \max_{i\in[d]}\|\boldsymbol{\mu}_i\|_2$. Suppose the outputs of step 2 in Algorithm 3, namely $\xi_1, \ldots, \xi_d$ and $\tilde{\mathbf{A}} = [\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_d]$, satisfy Theorem 3.2.1 with $\varepsilon_{out} < w_{min}^{4/3}/(24d\tau)$ and permutation π. One can choose positive right singular vectors $\mathbf{v}, \tilde{\mathbf{v}}$ associated with the minimum singular value of $\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}, \tilde{\mathbf{A}}\operatorname{diag}(\xi_i)^{1/3}$, respectively. Define $\tilde{\mathbf{w}} = \tilde{\mathbf{v}}^{3/2}/\sum_{i\in[d]}\tilde{\mathbf{v}}_i^{3/2}$ and $\tilde{\rho}_i = (\xi_i/\tilde{w}_i)^{1/3}$. Then $|w_{\pi(i)} - \tilde{w}_i| \le 12w_{min}^{-1/3}d\tau\varepsilon_{out}$ and $|\rho_{\pi(i)} - \tilde{\rho}_i| \le 48w_{min}^{-4/3}\rho_{max}d\tau\varepsilon_{out}$.*

Proof. Let $\mathbf{v} = [v_1, \ldots, v_d], \tilde{\mathbf{v}} = [\tilde{v}_1, \ldots, \tilde{v}_d], \tilde{\mathbf{w}} = [\tilde{w}_1, \ldots, \tilde{w}_d]$. We start by showing that $\mathbf{v}, \tilde{\mathbf{v}}$ and $\tilde{\mathbf{w}}$ are well-defined. Since $\mathbf{w}^{2/3}$ is a solution to (4.2.3) and $\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}$ is of rank $d-1$, we pick $\mathbf{v} = \mathbf{w}^{2/3}/\|\mathbf{w}^{2/3}\|_2$. To show that $\tilde{\mathbf{v}}$ is well-defined, first we bound the singular values and vectors of $\tilde{\mathbf{A}}\operatorname{diag}(\xi_i)^{1/3}$. Let $\tilde{\sigma}_i = \sigma_i(\tilde{\mathbf{A}}\operatorname{diag}(\xi_i)^{1/3})$. By Theorem 2.2.1,

$$\tilde{\sigma}_d \le \|\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3} - \tilde{\mathbf{A}}\operatorname{diag}(\xi_i)^{1/3}\|_2 \le \sqrt{d}\varepsilon_{out} < w_{min}^{4/3}/(24\sqrt{d}\tau).$$

To obtain the deviation in the singular vectors, we first show that $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{d-1}$ are bounded away from zero. Let $\boldsymbol{\Sigma}_1 = \operatorname{diag}\big(\sigma_1(\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}), \ldots, \sigma_{d-1}(\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3})\big)$, $\tilde{\boldsymbol{\Sigma}}_1 = \operatorname{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{d-1})$ and $\Delta = w_{min}^{1/3}/(2\tau)$. Suppose $\hat{\sigma}_{d-1}$ is the least singular value of the matrix obtained by deleting the first column of $\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}$, then it follows that $\sigma_{d-1}(\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}) \ge \hat{\sigma}_{d-1} \ge w_{min}^{1/3}/\tau$, where the first inequality follows from the interlacing property of singular values of a matrix and its submatrix obtained by deleting any column, and the second inequality comes from Assumption 4.2.1.

The minimum diagonal term in $\tilde{\mathbf{\Sigma}}_1$ satisfies:

$$\min_i \, (\tilde{\mathbf{\Sigma}}_1)_{ii} \geq \sigma_{d-1}(\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}) - \sqrt{d}\varepsilon_{out} \geq \frac{w_{min}^{1/3}}{\tau} - \frac{w_{min}^{4/3}}{24\sqrt{d}\tau} \geq \frac{w_{min}^{1/3}}{2\tau} = \Delta.$$

Therefore by Theorem 2.2.2 with $\mathbf{\Sigma}_2 = 0$, we have for the singular vectors[2]:

$$\|\mathbf{v} - \tilde{\mathbf{v}}\|_2 \leq \sqrt{2d}\varepsilon_{out}/\Delta = 2\sqrt{2d}w_{min}^{-1/3}\tau\varepsilon_{out}.$$

We get $\tilde{v}_i \geq v_i - 2\sqrt{2d}w_{min}^{-1/3}\tau\varepsilon_{out} \geq w_{min}^{2/3} - 2\sqrt{2d}w_{min}^{-1/3}\tau\varepsilon_{out} > 0$, where the second inequality follows from $\sum_{i\in[d]} v_i^{3/2} \geq \sum_{i\in[d]} v_i^2 = 1$. Hence $\tilde{\mathbf{v}}$ also has positive entries and $\tilde{\mathbf{w}}$ is well-defined.

We now derive the bounds on the mixing weights and norms. Without loss of generality π is the identity. The mixing weight error is bounded by:

$$\begin{aligned} \|\tilde{\mathbf{w}} - \mathbf{w}\|_2 &= \left\| \frac{\tilde{\mathbf{v}}^{3/2}}{\sum_{i\in[d]} \tilde{v}_i^{3/2}} - \frac{\mathbf{v}^{3/2}}{\sum_{i\in[d]} v_i^{3/2}} \right\|_2 \\ &\leq \frac{\|\tilde{\mathbf{v}}^{3/2} - \mathbf{v}^{3/2}\|_2}{\sum_{i\in[d]} v_i^{3/2}} + \frac{\|\tilde{\mathbf{v}}^{3/2}\|_2}{(\sum_{i\in[d]} v_i^{3/2})(\sum_{i\in[d]} \tilde{v}_i^{3/2})} \left| \sum_{i\in[d]} (v_i^{3/2} - \tilde{v}_i^{3/2}) \right|. \end{aligned} \tag{4.2.4}$$

We bound each term in (4.2.4) below, since $\tilde{v}, v$ both have entries in $(0, 1]$: $\sum_{i\in[d]} v_i^{3/2} \geq \|\mathbf{v}\|_2^2 = 1$, $\sum_{i\in[d]} \tilde{v}_i^{3/2} \geq \|\tilde{\mathbf{v}}\|_2^2 = 1$, and $\|\tilde{\mathbf{v}}^{3/2}\|_2 = (\sum_{i\in[d]} \tilde{v}_i^3)^{1/2} \leq \|\tilde{\mathbf{v}}\|_2 = 1$. Moreover:

$$\|\tilde{\mathbf{v}}^{3/2} - \mathbf{v}^{3/2}\|_2 = \Big(\sum_{i\in[d]} \big(\tilde{v}_i^{3/2} - v_i^{3/2}\big)^2\Big)^{1/2} \leq \frac{3}{2}\Big(\sum_{i\in[d]} (\tilde{v}_i - v_i)^2\Big)^{1/2} = \frac{3}{2}\|\tilde{\mathbf{v}} - \mathbf{v}\|_2,$$

$$\Big|\sum_{i\in[d]} (v_i^{3/2} - \tilde{v}_i^{3/2})\Big| \leq \sum_{i\in[d]} |v_i^{3/2} - \tilde{v}_i^{3/2}| \leq \frac{3}{2}\sum_{i\in[d]} |v_i - \tilde{v}_i| \leq \frac{3\sqrt{d}}{2}\|\tilde{\mathbf{v}} - \mathbf{v}\|_2,$$

where the above two inequalities follow from $|x^{3/2} - y^{3/2}| \leq 3|x - y|/2$ for $x, y \in [0, 1]$. We obtain the following bound on the error in mixing weights:

$$\|\tilde{\mathbf{w}} - \mathbf{w}\|_2 \leq \frac{3(1 + \sqrt{d})}{2}\|\tilde{\mathbf{v}} - \mathbf{v}\|_2 \leq 3\sqrt{2}w_{min}^{-1/3}(d + \sqrt{d})\tau\varepsilon_{out} \leq 12w_{min}^{-1/3}d\tau\varepsilon_{out}. \tag{4.2.5}$$

[2] Note that even though Theorem 2.2.2 gives the angle between the subspaces spanned by the first $d - 1$ right singular vectors of $\mathbf{A}\operatorname{diag}(w_i\rho_i^3)^{1/3}$ and their perturbed counterparts, the same bound applies to the orthogonal complement, spanned by $\mathbf{v}$.

Notice that our assumption on ε_{out} guarantees that $\tilde{w}_i \geq w_{min}/2$, therefore the error in the norm is bounded by:

$$\begin{aligned}
|\tilde{\rho}_i - \rho_i| &= |(\xi_i/\tilde{w}_i)^{1/3} - \rho_i| \leq \tilde{w}_i^{-1/3}\big(|\xi_i^{1/3} - w_i^{1/3}\rho_i| + \rho_i|w_i^{1/3} - \tilde{w}_i^{1/3}|\big) \\
&\leq \tilde{w}_i^{-1/3}\big(\varepsilon_{out} + \rho_{max}|w_i^{1/3} - \tilde{w}_i^{1/3}|\big) \leq (2w_{min}^{-1})^{1/3}(\varepsilon_{out} + \rho_{max}w_{min}^{-2/3}|w_i - \tilde{w}_i|) \\
&\leq 48w_{min}^{-4/3}\rho_{max}d\tau\varepsilon_{out}
\end{aligned}$$

where the second inequality comes from Theorem 3.2.1, the third inequality comes from the fact $|x^{1/3} - y^{1/3}|/|x-y| \leq y^{-2/3}$ for any $x, y > 0$, and the last follows from (4.2.5). □

We are now ready to prove Theorem 4.2.2.

Proof of Theorem 4.2.2. Set the arguments of $\mathrm{poly}'_{3.2.1}, \mathrm{poly}_{3.2.1}$, to $(d, \tau, \rho_{max}, w_{min}^{-1/3}\rho_{min}^{-1})$ and (d, τ, ρ_{max}), respectively. Assume for a moment that N is large enough so that $\tilde{\boldsymbol{\mathcal{T}}}$ in step 1 satisfies $\|\boldsymbol{\mathcal{T}} - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} \leq \varepsilon_{out}/(\mathrm{poly}'_{3.2.1})$ and we can apply Theorem 3.2.1. We start by verifying that we can apply Theorem 4.2.3. Set $\mathrm{poly}_{4.2.2} = 49w_{min}^{-4/3}\max\{\rho_{max}, 1\}d\tau\,\mathrm{poly}_{3.2.1}$. By Theorem 3.2.1, using our assumption $\varepsilon' \leq 1$, our choice of $\varepsilon_{4.2.2}$ guarantees that the output error of step 2 in Algorithm 3 is

$$\varepsilon_{out} = \varepsilon_{4.2.2}\,\mathrm{poly}_{3.2.1} = \varepsilon'/(49w_{min}^{-4/3}\max\{\rho_{max}, 1\}d\tau) < w_{min}^{4/3}/(24d\tau).$$

We now bound our estimation error for $\|\boldsymbol{\mu}_i\|_2$ and w_i with Theorem 4.2.3. Assuming the permutation is the identity we have for $i \in [d]$:

$$\begin{aligned}
\|\boldsymbol{\mu}_i - \tilde{\boldsymbol{\mu}}_i\|_2 &\leq |\rho_i - \tilde{\rho}_i|\|\tilde{\mathbf{a}}_i\|_2 + \rho_i\|\mathbf{a}_i - \tilde{\mathbf{a}}_i\|_2 \leq (48w_{min}^{-4/3}\rho_{max}d\tau + \rho_{max})\varepsilon_{out} \leq \varepsilon', \\
|w_i - \tilde{w}_i| &\leq 12w_{min}^{-1/3}d\tau\varepsilon_{out} \leq \varepsilon'.
\end{aligned}$$

Next, we derive the sample complexity. We need:

$$\|\boldsymbol{\mathcal{T}} - \tilde{\boldsymbol{\mathcal{T}}}\|_{\mathrm{F}} \leq \varepsilon_{in} \leq \varepsilon_{out}/(\mathrm{poly}'_{3.2.1}) = \varepsilon'\,\mathrm{poly}_{3.2.1}/(\mathrm{poly}'_{3.2.1}\,\mathrm{poly}_{4.2.2}).$$

By Lemma 2.4.7, $N = \Omega(\varepsilon'^{-2}\delta^{-1}d^{11}\max_{i\in[d]}\mathbb{E}[X_i^6](\mathrm{poly}'_{3.2.1})^2)$ many samples are sufficient for ε_{in} to meet the assumption. Since N is polynomial in δ^{-1} and $\max_{i\in[d]}\mathbb{E}[X_i^6]$, the expected running time will also be polynomial in them. □

4.3. Parameter estimation of Gaussian Mixture Models

In this section we consider a specific family of mixture models, namely GMM with identical but unknown covariance matrices. The model is as in (4.2.1), where $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. Our goal is to approximate all parameters of the mixture: $\boldsymbol{\Sigma}$, w_is and $\boldsymbol{\mu}_i$s. Again, suppose Assumption 4.2.1 holds and the mean of the mixture is zero (by translating the samples as in Section 4.2). Algorithm 3 guarantees that we can recover the mixing weights w_is and centers $\boldsymbol{\mu}_i$s of $\mathbf{z}$. To recover $\boldsymbol{\Sigma}$, notice that since the mean is zero, $\operatorname{Cov}(\mathbf{x}) = \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \sum_{i \in [d]} w_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top + \boldsymbol{\Sigma}$. The covariance matrix can be approximated then by taking the difference between the sample second moment of $\mathbf{x}$ and the second moment of the reconstructed discrete distribution. We make this precise in Algorithm 4 and Theorem 4.3.1.

Algorithm 4 Parameter estimation for GMM

```
Inputs: i.i.d. samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ from mixture $\mathbf{x}$, error tolerance $\varepsilon''$, upper bound $\rho_{max}$ on $\|\boldsymbol{\mu}_i\|_2$ for $i \in [d]$, lower bound $w_{min}$ on $w_i$ for $i \in [d]$, robust Kruskal rank threshold $\tau$.
invoke Algorithm 3 with samples from $\mathbf{x}$ and parameters $\varepsilon' = \varepsilon'' / \operatorname{poly}_{4.3.1}$, $\rho_{max}, w_{min}, \tau$ to get $\tilde{w}_i$ and $\tilde{\boldsymbol{\mu}}_i$ for $i \in [d]$;
set $\tilde{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{j \in [N]} \mathbf{x}_j \mathbf{x}_j^\top - \sum_{i \in [d]} \tilde{w}_i \tilde{\boldsymbol{\mu}}_i \tilde{\boldsymbol{\mu}}_i^\top$;
Outputs: estimated covariance matrix $\tilde{\boldsymbol{\Sigma}}$, mixing weights and means $\tilde{w}_i, \tilde{\boldsymbol{\mu}}_i : i \in [d]$.
```

Theorem 4.3.1 (Correctness of Algorithm 4). *Let $\mathbf{x}$ be a GMM with identical but unknown covariance matrices satisfying Assumption 4.2.1. Assume $0 < w_{min} \leq \min_{i \in [d]} w_i$, $\rho_{max} \geq \max_{i \in [d]} \rho_i$, $0 < \rho_{min} \leq \min_{i \in [d]} \rho_i$, $0 < \varepsilon'' \leq \min\{1, w_{min} \rho_{min}^3\}$ and $\delta \in (0,1)$. There exist a polynomial $\operatorname{poly}_{4.3.1}(d, \rho_{max})$ such that if $\varepsilon' = \varepsilon'' / \operatorname{poly}_{4.3.1}$, then given N i.i.d. samples of $\mathbf{x}$ and with probability $1 - \delta$ over the randomness in the samples Algorithm 4 outputs $\tilde{\boldsymbol{\mu}}_1, \dots, \tilde{\boldsymbol{\mu}}_d$, $\tilde{w}_1, \dots, \tilde{w}_d$ and $\tilde{\boldsymbol{\Sigma}}$ such that for some permutation π of $[d]$ and $\forall i \in [d]$: $\|\tilde{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\|_{\mathrm{F}} \leq \varepsilon''$, $|w_{\pi(i)} - \tilde{w}_i| \leq \varepsilon'$ and $\|\boldsymbol{\mu}_{\pi(i)} - \tilde{\boldsymbol{\mu}}_i\|_2 \leq \varepsilon'$. The expected running time over the randomness of Algorithm 2 is at most $\operatorname{poly}(d, \varepsilon''^{-1}, \delta^{-1}, \tau, \rho_{max}, \rho_{min}^{-1}, w_{min}^{-1}, \max_{i \in [d]} \boldsymbol{\Sigma}_{ii}^3)$ and will use $N = \Omega\big(\varepsilon''^{-2} \delta^{-1} d^{13} \max_{i \in [d]} \boldsymbol{\Sigma}_{ii}^3 \left(\operatorname{poly}'_{3.2.1}(d, \tau, \rho_{max}, w_{min}^{-1/3} \rho_{min}^{-1})\right)^2\big)$ samples.*

Proof. Let $\operatorname{poly}_{4.3.1}(d, \rho_{max}) = 1 + d\rho_{max}^2 + 2d(2\rho_{max} + 1)$. By Theorem 4.2.2, with probability $1 - \delta$, Algorithm 3 will output the estimated mixing weights $\tilde{w}_i$ and means $\tilde{\boldsymbol{\mu}}_i$ within ε' additive accuracy. The sample complexity and running time follows therein, where we have $\max_{i\in[d]} \mathbb{E}[X_i^6] = \max_{i\in[d]} 15\boldsymbol{\Sigma}_{ii}^3$ for GMM.

Next, we bound the error in the covariance matrix. Note that when the number of samples guarantees that $\boldsymbol{\mathcal{T}}$ is estimated to ε_{in} accuracy with probability $1-\delta$, it can also guarantee $\mathrm{Cov}(\mathbf{x})$ is estimated to ε_{in} accuracy with probability $1-\delta$ since the latter takes $\Omega(d^6\varepsilon_{in}^{-2}\delta^{-1}\max_{i\in[d]}\boldsymbol{\Sigma}_{ii}^2)$ many samples by a similar argument to Lemmas 2.4.5 and 2.4.6. So

$$\begin{aligned}\|\tilde{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}\|_{\mathrm{F}} &= \left\|\frac{1}{N}\sum_{j\in[N]}\mathbf{x}_j\mathbf{x}_j^\top - \sum_{i\in[d]}\tilde{w}_i\tilde{\boldsymbol{\mu}}_i\tilde{\boldsymbol{\mu}}_i^\top - \boldsymbol{\Sigma}\right\|_{\mathrm{F}} \\ &\le \left\|\frac{1}{N}\sum_{j\in[N]}\mathbf{x}_j\mathbf{x}_j^\top - \mathrm{Cov}(\mathbf{x})\right\|_{\mathrm{F}} + \sum_{i\in[d]}|w_i-\tilde{w}_i|\|\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top\|_{\mathrm{F}} + \sum_{i\in[d]}\tilde{w}_i\|\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top - \tilde{\boldsymbol{\mu}}_i\tilde{\boldsymbol{\mu}}_i^\top\|_{\mathrm{F}} \\ &\le \varepsilon_{in} + d\rho_{max}^2\varepsilon' + \sum_{i\in[d]}(w_i+\varepsilon')(2\|\boldsymbol{\mu}_i\|_2+\varepsilon')\varepsilon' \le \mathrm{poly}_{4.3.1}\,\varepsilon' \le \varepsilon'',\end{aligned}$$

where the second to last inequality follows from bounding ε_{in} by ε' and w_i, ε' by 1. □

CHAPTER 5

Conclusions and Open Questions

5.1. Order-3 Tensor Decomposition

In Chapter 3, we showed that an order-3 tensor can still be robustly decomposed in polynomial time, even if its components are linearly dependent. By Kruskal's theorem, our algorithm works up to $n \leq 3d/2 - 1$ components with every d components being linearly independent. Our work does not assume randomness or incoherence of the components and is a complement to the existing literature. However, many questions on tensor decomposition still remain to be answered. We summarize the decomposability of tensors below.

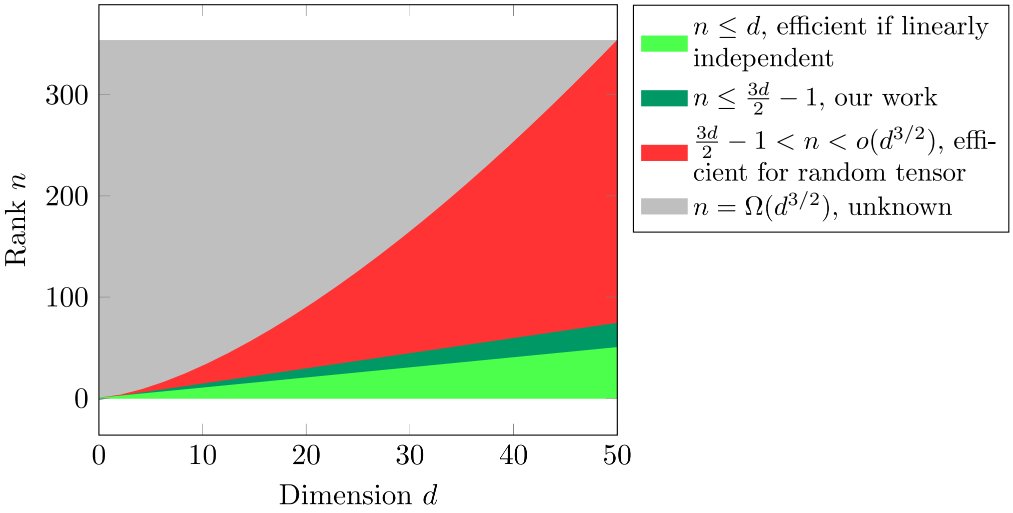


FIGURE 5.1. Decomposability of order-3 tensors.

In the mildly overcomplete regime, ($n \leq Cd$ for some fixed C), our work and [**DDL14**, **DDL17**] guarantee the decomposability for generic tensors, but the running time, though being polynomial in d, depends exponentially on the notion of *overcompleteness*, while [**AGJ14**] gives a polynomial time algorithm under the incoherence or randomness assumption, and their results generalize to the highly overcomplete regime: $n = o(d^{3/2})$ components at most. It is not clear if we can take the advantage of all these algorithms:

Can we decompose generic tensors in polynomial time with $n = o(d^{3/2})$ components?

Moreover, it is unknown, to our knowledge, if there exists any efficient algorithm, even for random tensors when $n = \Omega(d^{3/2})$.

From a practitioner's perspective, our results are more of theoretical values and far from being practical. Current tensor power iteration-based algorithms in the overcomplete regime [**AGJ14**, **AGJ17**] depend on deterministic assumptions such as incoherent components or stochastic assumptions such as components being drawn uniformly on the sphere. It is an interesting question to develop algorithms for overcomplete tensor decomposition with both less theoretical assumptions and stronger practical performance.

5.2. Blind Deconvolution and GMM Estimation

Our algorithm guarantees that if the natural non-degeneracy condition (Assumption 4.2.1) is met, a Gaussian mixture with unknown but identical covariance matrices can be learned. However, it is still an open question that whether algorithms based on tensor decomposition can estimate arbitrary Gaussian mixtures. Our algorithm requires all covariance matrices being identical, while [**GHK15**] learns only $O(\sqrt{d})$ arbitrary Gaussians in $\mathbb{R}^d$. Therefore it is interesting to investigate:

Can we learn a mixture of $O(d^c)$ arbitrary Gaussians $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ in polynomial time for $c \geq 1$?

APPENDIX A

Codes for simulation

In this Appendix, we collect the python codes used for numerical simulations in Section 3.4.

```python
import numpy as np
from tqdm import tqdm
def simul_diag(M1, M2, r: int):
    """Simulatenous diagonalization algorithm."""
    U, _, __ = np.linalg.svd(M1)
    W = U[:, 0 : r]
    M1_whitened = W.T @ M1 @ W
    M2_whitened = W.T @ M2 @ W
    M = M1_whitened @ np.linalg.inv(M2_whitened)
    e, P = np.linalg.eig(M)
    return W @ P.real
def solve_matrix_sys(A_r, Tx, x):
    """
    Solve a matrix valued linear system A_r * Xi * diag(A_r.T * x) * A_r.T = Tx"""
    B = np.linalg.pinv(A_r).T
    Xi = np.diagonal(B.T @ Tx @ B)
    return Xi / np.tensordot(A_r, x, axes= [0,0])
def decompose(T, x, y, r: int):
    """
    Given an input tensor, run the simultaneous diagonalization algorithm and length recovery"""
    T_x = np.tensordot(T, x, axes=[[0],[0]])
    T_y = np.tensordot(T, y, axes=[[0],[0]])
    A = simul_diag(T_x, T_y, r = r)
    Xi = solve_matrix_sys(A, T_x, x)
    return A, Xi
class overcomplete_tensor_decomposition():
    """Overcomplete tensor decomposition"""
    def __init__(self, dimension, init_x = None, init_y = None, rng = None):
        if init_x is not None:
            self.x_magic = init_x
        elif rng is not None:
            self.x_magic = rng.normal(size = (dimension))
        else:
            self.x_magic = np.random.normal(size = (dimension))
        if init_y is not None:
            self.y_magic = init_y
        elif rng is not None:
            self.y_magic = rng.normal(size = (dimension))
        else:
            self.y_magic = np.random.normal(size = (dimension))
        if rng is not None:
            self.x_2nd = rng.normal(size = (dimension))
            self.y_2nd = rng.normal(size = (dimension))
        else:
            self.x_2nd = np.random.normal(size = (dimension))
            self.y_2nd = np.random.normal(size = (dimension))
    def __call__(self, T, tensor_rank, overcomplete_param):
        # Decompose the first r compoents
        A_r, Xi_r = decompose(T, self.x_magic, self.y_magic, r = tensor_rank - overcomplete_param)
        # Deflation
        T_first = construct_sym_tensor(components= A_r, weights= Xi_r)
        R = T - T_first
        # 2nd decomposition
        A_k, Xi_k = decompose(R, self.x_2nd, self.y_2nd, r = overcomplete_param)
        # Reconstruction
```

```python
        T_second = construct_sym_tensor(components= A_k, weights= Xi_k)
        return T_first + T_second, (A_r, Xi_r, A_k, Xi_k)
def construct_sym_tensor(components, weights = None):
    """Construct a 3rd order symmetric tensor with columns of components and muliplicative constants from weights"""
    _, rank = np.shape(components)
    if weights is None:
        weights = np.ones((1,rank))
    return np.einsum(’il,jl,kl->ijk’, components * weights, components, components)

dimensions = np.arange(10,70,10) # input dimension & Kruskal rank
overcomplete_params = np.arange(1,5) # different overcompleteness from 1 to 4
rng = np.random.default_rng(seed=202206241)
MAX_TRIAL, EXP_PER_DIM = 10000, 50
exp_errors = {d: {r: np.zeros(EXP_PER_DIM) for r in overcomplete_params} for d in dimensions}
exp_time = {d: {r: np.zeros(EXP_PER_DIM) for r in overcomplete_params} for d in dimensions}
for d in tqdm(dimensions):
    for r in tqdm(overcomplete_params):
        for i in tqdm(range(EXP_PER_DIM)):
            # params per experiment
            min_err = err = float(’inf’)
            trials = 0
            # initialize inputs
            components = rng.normal(size = (d, d+r)) # generate d+r Gaussian vectors in R^d
            tensor_in = construct_sym_tensor(components = components)
            norm_tensor_in = np.linalg.norm(tensor_in)
            # decomposition
            start_time = time.time()
            while err > 1e-1 and trials <= MAX_TRIAL:
                # try decomposing the input tensor
                decomposer = overcomplete_tensor_decomposition(dimension = d,rng = rng)
                outputs = decomposer(T = tensor_in, tensor_rank = d+r, overcomplete_param = r)
                tensor_out = outputs[0]
                # check termination
                err = np.linalg.norm(tensor_in - tensor_out)/norm_tensor_in # measure relative error
                min_err = min(min_err,err)
                trials += 1
            end_time = time.time()
            exp_errors[d][r][i] = min_err
            exp_time[d][r][i] = end_time - start_time
```

Bibliography

[AB03] C. M. Andersen and R. Bro, *Practical aspects of PARAFAC modeling of fluorescence excitation-emission data*, Journal of Chemometrics **17** (2003), no. 4, 200–215, https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.790.

[ABB$^+$07] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, *Seizure recognition on epilepsy feature tensor*, 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 4273–4276.

[ABG$^+$14] J. Anderson, M. Belkin, N. Goyal, L. Rademacher, and J. Voss, *The more, the merrier: the blessing of dimensionality for learning large Gaussian mixtures*, Conference on Learning Theory, 2014, pp. 1135–1164.

[AGH$^+$14] A. Anandkumar, R. Ge, D. J. Hsu, S. M. Kakade, and M. Telgarsky, *Tensor decompositions for learning latent variable models*, J. Mach. Learn. Res. **15** (2014), no. 1, 2773–2832.

[AGJ14] A. Anandkumar, R. Ge, and M. Janzamin, *Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates*, CoRR **abs/1402.5180** (2014), 1402.5180.

[AGJ15] ———, *Learning overcomplete latent variable models through tensor methods*, Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015 (P. Grünwald, E. Hazan, and S. Kale, eds.), JMLR Workshop and Conference Proceedings, vol. 40, JMLR.org, 2015, pp. 36–112.

[AGJ17] ———, *Analyzing tensor power method dynamics in overcomplete regime*, J. Mach. Learn. Res. **18** (2017), 22:1–22:40.

[BAH19] I. Balazevic, C. Allen, and T. M. Hospedales, *Tucker: Tensor factorization for knowledge graph completion*, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), Association for Computational Linguistics, 2019, pp. 5184–5193.

[BCMV14] A. Bhaskara, M. Charikar, A. Moitra, and A. Vijayaraghavan, *Smoothed analysis of tensor decompositions*, Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '14, Association for Computing Machinery, 2014, p. 594–603.

[BCV14] A. Bhaskara, M. Charikar, and A. Vijayaraghavan, *Uniqueness of tensor decompositions with applications to polynomial identifiability*, Conference on Learning Theory, 2014, pp. 742–778.

[Bro97] R. Bro, *PARAFAC. tutorial and applications*, Chemometrics and Intelligent Laboratory Systems **38** (1997), no. 2, 149–171.

[Bro06] ———, *Review on multiway analysis in chemistry—2000–2005*, Critical Reviews in Analytical Chemistry **36** (2006), no. 3-4, 279–293, https://doi.org/10.1080/10408340600969965.

[BRV13] M. Belkin, L. Rademacher, and J. R. Voss, *Blind signal separation in the presence of Gaussian noise*, COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA (S. Shalev-Shwartz and I. Steinwart, eds.), JMLR Workshop and Conference Proceedings, vol. 30, JMLR.org, 2013, pp. 270–287.

[BS05] C. Beckmann and S. Smith, *Tensorial extensions of independent component analysis for multisubject FMRI analysis*, NeuroImage **25** (2005), no. 1, 294–311.

[Car91] J. Cardoso, *Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors*, 1991 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '91, Toronto, Ontario, Canada, May 14-17, 1991, IEEE Computer Society, 1991, pp. 3109–3112.

[CC70] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition*, Psychometrika **35** (1970), 283–319.

[CJ10] P. Comon and C. Jutten, *Handbook of blind source separation: Independent component analysis and applications*, 1st ed., Academic Press, 2010.

[Com02] P. Comon, *Tensor Decompositions, State of the Art and Applications*, Mathematics in Signal Processing V (J. G. McWhirter and I. K. Proudler, eds.), Clarendon Press, Oxford, 2002, pp. 1–24, ISBN = 0-19-850734-8 ; 9780198507345.

[CR22] H. Chen and L. Rademacher, *Overcomplete order-3 tensor decomposition, blind deconvolution, and Gaussian mixture models*, SIAM Journal on Mathematics of Data Science **4** (2022), no. 1, 336–361, https://doi.org/10.1137/21M1399415.

[DC07] L. De Lathauwer and J. Castaing, *Tensor-based techniques for the blind separation of DS-CDMA signals*, Signal Processing **87** (2007), no. 2, 322–336, Tensor Signal Processing.

[DDL14] I. Domanov and L. De Lathauwer, *Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition*, SIAM J. Matrix Anal. Appl. **35** (2014), no. 2, 636–660.

[DDL17] ———, *Canonical polyadic decomposition of third-order tensors: relaxed uniqueness conditions and algebraic algorithm*, Linear Algebra Appl. **513** (2017), 342–375.

[DG03] S. Dasgupta and A. Gupta, *An elementary proof of a theorem of Johnson and Lindenstrauss*, Random Structures & Algorithms **22** (2003), no. 1, 60–65.

[DHKK20] I. Diakonikolas, S. B. Hopkins, D. Kane, and S. Karmalkar, *Robustly learning any clusterable mixture of Gaussians*, CoRR **abs/2005.06417** (2020), 2005.06417.

[DLCC07] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, *Fourth-order cumulant-based blind identification of underdetermined mixtures*, IEEE Transactions on Signal Processing **55** (2007), no. 6, 2965–2973.

[DLdB08] L. De Lathauwer and A. de Baynast, *Blind deconvolution of DS-CDMA signals by means of decomposition in rank-$(1, L, L)$ terms*, IEEE Transactions on Signal Processing **56** (2008), no. 4, 1562–1571.

[DLDM96] L. De Lathauwer and B. De Moor, *From matrix to tensor: multilinear algebra and signal processing*, vol. 67, McWhirter, JG, CLARENDON PRESS, 1996, pp. 1–11 (eng).

[DLDMV00] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM Journal on Matrix Analysis and Applications **21** (2000), no. 4, 1324–1342, https://doi.org/10.1137/S0895479898346995.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc. Ser. B **39** (1977), no. 1, 1–38, With discussion.

[DVD$^+$07] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen, *Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone*, NeuroImage **37** (2007), no. 3, 844–854.

[FG22] A. Frandsen and R. Ge, *Optimization landscape of Tucker decomposition*, Math. Program. **193** (2022), no. 2, 687–712.

[GHK15] R. Ge, Q. Huang, and S. M. Kakade, *Learning mixtures of Gaussians in high dimensions*, Proceedings of the forty-seventh annual ACM symposium on Theory of computing, ACM, 2015, pp. 761–770.

[GM15] R. Ge and T. Ma, *Decomposing overcomplete 3rd order tensors using sum-of-squares algorithms*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA (N. Garg, K. Jansen, A. Rao, and J. D. P. Rolim, eds.), LIPIcs, vol. 40, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 829–849.

[GM17] ———, *On the optimization landscape of tensor decompositions*, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA (I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds.), 2017, pp. 3653–3663.

[GPNV16] T. Garipov, D. Podoprikhin, A. Novikov, and D. P. Vetrov, *Ultimate tensorization: compressing convolutional and FC layers alike*, CoRR **abs/1611.03214** (2016), 1611.03214.

[GVL13] G. H. Golub and C. F. Van Loan, *Matrix computations (4th ed.)*, Johns Hopkins University Press, 2013.

[GVX13] N. Goyal, S. S. Vempala, and Y. Xiao, *Fourier PCA*, CoRR **abs/1306.5825** (2013), 1306.5825.

[GVX14] ———, *Fourier PCA and robust tensor decomposition*, Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014 (D. B. Shmoys, ed.), ACM, 2014, pp. 584–593.

[Har70] R. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis*, UCLA Working Papers in Phonetics **16** (1970), 1–84.

[Hås90] J. Håstad, *Tensor rank is NP-complete*, J. Algorithms **11** (1990), no. 4, 644–654.

[Hen94] R. Henrion, *N-way principal component analysis theory, algorithms and applications*, Chemometrics and Intelligent Laboratory Systems **25** (1994), no. 1, 1–23.

[Hit27] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Journal of Mathematics and Physics **6** (1927), no. 1-4, 164–189, https://onlinelibrary.wiley.com/doi/pdf/10.1002/sapm192761164.

[HK13] D. Hsu and S. M. Kakade, *Learning mixtures of spherical Gaussians: moment methods and spectral decompositions*, Proceedings of the 4th conference on Innovations in Theoretical Computer Science, ACM, 2013, pp. 11–20.

[HL13] C. J. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, Journal of the ACM (JACM) **60** (2013), no. 6, 45.

[Hog13] L. Hogben (ed.), *Handbook of linear algebra (2nd ed.)*, CRC press, 2013.

[HSK13] S. Hawe, M. Seibert, and M. Kleinsteuber, *Separable dictionary learning*, 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013, IEEE Computer Society, 2013, pp. 438–445.

[HSS19] S. B. Hopkins, T. Schramm, and J. Shi, *A robust spectral algorithm for overcomplete tensor decomposition*, Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA (A. Beygelzimer and D. Hsu, eds.), Proceedings of Machine Learning Research, vol. 99, PMLR, 2019, pp. 1683–1722.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, Springer series in statistics, Springer, 2009.

[HVB$^+$16] V. Hore, A. Viñuela, A. Buil, J. Knight, M. I. McCarthy, K. Small, and J. Marchini, *Tensor decomposition for multiple-tissue gene expression experiments*, Nature Genetics **48** (2016), no. 9, 1094–1100.

[JGKA19] M. Janzamin, R. Ge, J. Kossaifi, and A. Anandkumar, *Spectral learning on matrices and tensors*, Foundations and Trends® in Machine Learning **12** (2019), no. 5-6, 393–536.

[JV19] H. Jia and S. Vempala, *Robustly clustering a mixture of Gaussians*, arXiv preprint arXiv:1911.11838 (2019).

[KB09] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review **51** (2009), no. 3, 455–500, https://doi.org/10.1137/07070111X.

[Kha67] C. G. Khatri, *On certain inequalities for normal distributions and their applications to simultaneous confidence bounds*, The Annals of Mathematical Statistics **38** (1967), no. 6, 1853–1867.

[KKP21] J. Kileel, T. Klock, and J. M. Pereira, *Landscape analysis of an improved power method for tensor decomposition*, Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual (M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds.), 2021, pp. 6253–6265.

[Kol] T. G. Kolda, *Will the real Jennrich's Algorithm please stand up?*, `https://www.mathsci.ai/post/jennrich/`, Accessed: 2021–12–21.

[Kol01] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM Journal on Matrix Analysis and Applications **23** (2001), no. 1, 243–255, https://doi.org/10.1137/S0895479800368354.

[KPY$^+$16] Y. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, *Compression of deep convolutional neural networks for fast and low power mobile applications*, 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (Y. Bengio and Y. LeCun, eds.), 2016.

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

[KVM01] H. Kiers and I. Van Mechelen, *Three-way component analysis: Principles and illustrative application*, Psychological Methods **6** (2001), no. 1, 84–110 (English).

[LCC07] L. D. Lathauwer, J. Castaing, and J. Cardoso, *Fourth-order cumulant-based blind identification of underdetermined mixtures*, IEEE Trans. Signal Process. **55** (2007), no. 6-2, 2965–2973.

[LGR$^+$15] V. Lebedev, Y. Ganin, M. Rakhuba, I. V. Oseledets, and V. S. Lempitsky, *Speeding-up convolutional neural networks using fine-tuned CP-decomposition*, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (Y. Bengio and Y. LeCun, eds.), 2015.

[LM17] R. Latała and D. Matlak, *Royen's proof of the Gaussian correlation inequality*, Geometric aspects of functional analysis, Springer, 2017, pp. 265–275.

[LR92] S. Leurgans and R. T. Ross, *Multilinear models: Applications in spectroscopy*, Statistical Science **7** (1992), no. 3, 289–310.

[LRA93] S. E. Leurgans, R. T. Ross, and R. B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl. **14** (1993), no. 4, 1064–1083.

[LT17] Q. Li and G. Tang, *Convex and nonconvex geometries of symmetric tensor factorization*, 51st Asilomar Conference on Signals, Systems, and Computers, ACSSC 2017, Pacific Grove, CA, USA, October 29 - November 1, 2017 (M. B. Matthews, ed.), IEEE, 2017, pp. 305–309.

[LWDF11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, *Understanding blind deconvolution algorithms*, IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011), no. 12, 2354–2367.

[McC18] P. McCullagh, *Tensor methods in statistics*, Courier Dover Publications, 2018.

[MD09] M. W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proceedings of the National Academy of Sciences **106** (2009), no. 3, 697–702, https://www.pnas.org/doi/pdf/10.1073/pnas.0803205106.

[MHH$^+$06] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, *Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG*, NeuroImage **29** (2006), no. 3, 938–947.

[MMD06] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006 (T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, eds.), ACM, 2006, pp. 327–336.

[MSS16] T. Ma, J. Shi, and D. Steurer, *Polynomial-time tensor decompositions with sum-of-squares*, IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA (I. Dinur, ed.), IEEE Computer Society, 2016, pp. 438–446.

[MZZ$^+$19] X. Ma, P. Zhang, S. Zhang, N. Duan, Y. Hou, M. Zhou, and D. Song, *A tensorized transformer for language modeling*, Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds.), 2019, pp. 2229–2239.

[NPOV15] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, *Tensorizing neural networks*, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), 2015, pp. 442–450.

[NROV14] A. Novikov, A. Rodomanov, A. Osokin, and D. P. Vetrov, *Putting MRFs on a tensor train*, Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, JMLR Workshop and Conference Proceedings, vol. 32, JMLR.org, 2014, pp. 811–819.

[Ose11] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput. **33** (2011), no. 5, 2295–2317.

[OST08] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov, *Tucker dimensionality reduction of three-dimensional arrays in linear time*, SIAM Journal on Matrix Analysis and Applications **30** (2008), no. 3, 939–956, https://doi.org/10.1137/060655894.

[Pea94] K. Pearson, *Contributions to the mathematical theory of evolution*, Philosophical Transactions of the Royal Society of London. A **185** (1894), 71–110.

[RMC21] N. Razin, A. Maman, and N. Cohen, *Implicit regularization in tensor factorization*, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (M. Meila and T. Zhang, eds.), Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 8913–8924.

[Roy14] T. Royen, *A simple proof of the Gaussian correlation conjecture extended to multivariate gamma distributions*, Far East Journal of Theoretical Statistics **48** (2014), 139–145.

[Sai16] A. K. Saibaba, *HOID: Higher order interpolatory decomposition for tensors based on Tucker representation*, SIAM Journal on Matrix Analysis and Applications **37** (2016), no. 3, 1223–1249, https://doi.org/10.1137/15M1048628.

[SBDS$^+$19] A. Seigal, M. Beguerisse-Díaz, B. Schoeberl, M. Niepel, and H. A. Harrington, *Tensor clustering with algebraic constraints gives interpretable groups of crosstalk mechanisms in breast cancer*, Journal of The Royal Society Interface **16** (2019), no. 151, 20180661, https://royalsocietypublishing.org/doi/pdf/10.1098/rsif.2018.0661.

[SH05] A. Shashua and T. Hazan, *Non-negative tensor factorization with applications to statistics and computer vision*, Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005 (L. D. Raedt and S. Wrobel, eds.), ACM International Conference Proceeding Series, vol. 119, ACM, 2005, pp. 792–799.

[Shi18] Y. Shitov, *A counterexample to Comon's conjecture*, SIAM J. Appl. Algebra Geom. **2** (2018), no. 3, 428–443.

[Ste90] G. W. Stewart, *Perturbation theory for the singular value decomposition*, Tech. report, University of Maryland Institute for Advanced Computer Studies, 1990.

[Ste91] ———, *Perturbation theory for the singular value decomposition*, SVD and Signal Processing, II: Algorithms, Analysis and Applications, R. J. Vacarro, ed., Elsevier, 1991, pp. 99–109.

[Ste98] ———, *Matrix algorithms: Volume 1: Basic decompositions*, Society for Industrial and Applied Mathematics, 1998.

[Ste01] ———, *Matrix algorithms: Volume 2: Eigensystems*, Society for Industrial and Applied Mathematics, 2001.

[SWZ19] Z. Song, D. P. Woodruff, and P. Zhong, *Relative error tensor low rank approximation*, Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (USA), SODA '19, Society for Industrial and Applied Mathematics, 2019, p. 2772–2789.

[TMBY17] V. Tresp, Y. Ma, S. Baier, and Y. Yang, *Embedding learning for declarative memories*, The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I (E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, eds.), Lecture Notes in Computer Science, vol. 10249, 2017, pp. 202–216.

[TSN18] A. Tjandra, S. Sakti, and S. Nakamura, *Tensor decomposition for compressing recurrent neural network*, 2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018, IEEE, 2018, pp. 1–8.

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

[VBPP06] D. Vlasic, M. Brand, H. Pfister, and J. Popovic, *Face transfer with multilinear models*, International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2006, Boston, Massachusetts, USA, July 30 - August 3, 2006, Courses (J. W. Finnegan and D. Shreiner, eds.), ACM, 2006, p. 24.

[VT02] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear analysis of image ensembles: Tensorfaces*, Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I (A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, eds.), Lecture Notes in Computer Science, vol. 2350, Springer, 2002, pp. 447–460.

[WA03] H. Wang and Ahuja, *Facial expression decomposition*, Proceedings Ninth IEEE International Conference on Computer Vision, 2003, pp. 958–965 vol.2.

[WA04] H. Wang and N. Ahuja, *Compact representation of multidimensional data using tensor rank-one decomposition*, Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 1, 2004, pp. 44–47 Vol.1.

[WMG19] R. Warlop, J. Mary, and M. Gartrell, *Tensorized determinantal point processes for recommendation*, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019 (A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, eds.), ACM, 2019, pp. 1605–1615.

[WPSZ18] M. Wang, Y. Panagakis, P. Snape, and S. P. Zafeiriou, *Disentangling the modes of variation in unlabelled data*, IEEE Trans. Pattern Anal. Mach. Intell. **40** (2018), no. 11, 2682–2695.

[WTSA15] Y. Wang, H.-Y. Tung, A. Smola, and A. Anandkumar, *Fast and guaranteed tensor decomposition via sketching*, Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (Cambridge, MA, USA), NIPS'15, MIT Press, 2015, p. 991–999.

[XCH$^+$10] L. Xiong, X. Chen, T. Huang, J. G. Schneider, and J. G. Carbonell, *Temporal collaborative filtering with bayesian probabilistic tensor factorization*, Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA, SIAM, 2010, pp. 211–222.

[YH17] Y. Yang and T. M. Hospedales, *Deep multi-task representation learning: A tensor factorisation approach*, 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[ZCW20] Z. Zhang, J. Cai, and J. Wang, *Duality-induced regularizer for tensor factorization based knowledge graph completion*, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[ZG01] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM Journal on Matrix Analysis and Applications **23** (2001), no. 2, 534–550, https://doi.org/10.1137/S0895479899352045.

[Ši67] Z. Šidák, *Rectangular confidence regions for the means of multivariate normal distributions*, Journal of the American Statistical Association **62** (1967), no. 318, 626–633, https://doi.org/10.1080/01621459.1967.10482935.