Learning Low-Dimensional Models for Heterogeneous Data

by

David Hong

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2019

Doctoral Committee:

Professor Laura Balzano, Co-Chair
Professor Jeffrey A. Fessler, Co-Chair
Professor Rina Foygel Barber, The University of Chicago
Professor Anna Gilbert
Professor Raj Rao Nadakuditi

David Hong

dahong@umich.edu

ORCID iD: 0000-0003-4698-4175

To mom and dad.

ACKNOWLEDGEMENTS

As always, no list of dissertation acknowledgements can hope to be complete, and I have many people to thank for their wonderful influence over the past six years of graduate study. To start, I thank my advisors, Professor Laura Balzano and Professor Jeffrey Fessler, who introduced me to the exciting challenges of learning low-dimensional image models from heterogeneous data and taught me the fundamental ideas and techniques needed to tackle it; this dissertation is the direct result. Laura and Jeff, I am always amazed by your deep and broad technical insights, your sincere care for students and your great patience (with me!). Working with you has changed the way I think, write and teach, and I cannot say how thankful I am to have had the opportunity to learn from you. I hope to one day be as wonderful an advisor to students of my own.

I am also thankful for my excellent committee. Professor Rina Barber has many times provided deep statistical insight into my work, and her wonderful suggestions can be found throughout this dissertation. Professor Anna Gilbert has always challenged me to strive for work that is both deep in theory and significant in practice, and she continues to be a role model for me. Professor Raj Nadakuditi guided me when I first arrived in Ann Arbor, encouraged me during my early difficulties, and taught me many of the random matrix theory tools I used in Chapters III and IV. I am indebted to all three and am honored to have had them form my committee.

I also had the distinct pleasure of spending the summer of 2017 interning at Sandia National Laboratories in Livermore, California, where I worked on the Generalized CP tensor decomposition in Chapter V. I am so thankful for the wonderful mentorship of Dr. Tammy Kolda and Dr. Cliff Anderson-Bergman, who proposed this work and taught me nearly everything I now know about tensor decomposition and its numerous applications. I continue to learn from their many insights into the practice and theory of data analysis. At Sandia, I also had the pleasure of working alongside Dr. Jed Duersch and of sharing a cubicle with John Kallaugher, who both made my time there truly delightful.

I have also been fortunate to interact with many great postdoctoral scholars (all in fact hired by members of my committee!), who made Ann Arbor intellectually fertile grounds: Dr. Sai Ravishankar, Dr. Il Yong Chun, Dr. Greg Ongie and Dr. Lalit Jain. I am thankful for our many fun conversations that have shaped how I think about choosing and tackling research problems. Chapters III and IV of the dissertation also

benefitted from discussions with Professor Edgar Dobriban and Professor Romain Couillet, and I thank them for sharing their insights about spiked covariance models and random matrix theory tools for analyzing their asymptotic properties.

My many wonderful peers have also created a great learning environment. I thank Wooseok Ha (who in fact introduced me to Rina), Audra McMillan, and Yan Shuo Tan for our many stimulating discussions over the years; I am so glad we got to meet at the 2016 Park City Mathematics Institute. Likewise, I am thankful for Mohsen Heidari, Matthew Kvalheim, Aniket Deshmukh, John Lipor and Dejiao Zhang, who began the Ph.D. program with me and have been wonderful co-voyagers. A special thank you goes to John – J.J., you have often been a true encouragement and I have learned so much from you about life and research. I am also so thankful for all the students of the SPADA Lab and Lab of Jeff (LoJ) who make coming to campus a joy; I will miss our many fun chats. I thank Donghwan Kim, Madison McGaffin, Mai Le and Gopal Nataraj, who welcomed me and my many questions about optimization and imaging, as well as Curtis Jin, Raj Tejas Suryaprakash, Brian Moore, Himanshu Nayar and Arvind Prasadan, who welcomed my random matrix theory questions. I also had the pleasure of working with Robert Malinas while he was an undergraduate on the Sequences of Unions of Subspaces (SUoS) in Chapter VII; I am so excited that you are continuing in research and look forward to all that you will accomplish!

My graduate studies were supported by a Rackham Merit Fellowship, a National Science Foundation (NSF) Graduate Research Fellowship (DGE #1256260) and NSF Grant ECCS-1508943. I am thankful to these sources that afforded me great intellectual freedom and made this dissertation possible.

Hyunkoo and Yujin Chung, Chad Liu and Anna Cunningham, Hal Zhang, Ruiji Jiang and Max Li, thank you for your friendship over the years. Thanks also to the Campredon's – you are dear friends (and have fed me many times). Special thanks to David Allor, David Nesbitt and Chris Dinh, who were not only roommates but also true brothers in Christ, and to Pastor Shannon Nielsen and my church, who so faithfully point me to Christ and my ultimate hope in Him.

Finally and most importantly, I thank my family. My older sisters always took excellent care of their little brother, and I have followed in their footsteps all my life (since, you know, I want to be like you both). Pauline and Esther, thank you for being such loving sisters and incredible examples to me. Josh and Martin, it is wonderful to finally have older brothers(-in-law) too! Mirabelle, my super cute niece, thanks for all your awesome smiles! Mom and Dad, you introduced me to an exciting world full of wonder and adventure, taught me to have fun working hard, and lovingly picked me up all those times I fell down. Thank you. This dissertation is dedicated to you.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ALGORITHMS

ALGORITHM

ABSTRACT

Modern data analysis increasingly involves extracting insights, trends and patterns from large and messy data collected from myriad heterogeneous sources. The scale and heterogeneity present exciting new opportunities for discovery, but also create a need for new statistical techniques and theory tailored to these settings. Traditional intuitions often no longer apply, e.g., when the number of variables measured is comparable to the number of samples obtained. A deeper theoretical understanding is needed to develop principled methods and guidelines for statistical data analysis. This dissertation studies the low-dimensional modeling of high-dimensional data in three heterogeneous settings.

The first heterogeneity is in the *quality* of samples, and we consider the standard and ubiquitous low-dimensional modeling technique of Principal Component Analysis (PCA). We analyze how well PCA recovers underlying low-dimensional components from high-dimensional data when some samples are noisier than others (i.e., have heteroscedastic noise). Our analysis characterizes the penalty of heteroscedasticity for PCA, and we consider a weighted variant of PCA that explicitly accounts for heteroscedasticity by giving less weight to samples with more noise. We characterize the performance of weighted PCA for all choices of weights and derive optimal weights.

The second heterogeneity is in the *statistical properties* of data, and we generalize the (increasingly) standard method of Canonical Polyadic (CP) tensor decomposition to allow for general statistical assumptions. Traditional CP tensor decomposition is most natural for data with all entries having Gaussian noise of homogeneous variance. Instead, the Generalized CP (GCP) tensor decomposition we propose allows for other statistical assumptions, and we demonstrate its flexibility on various datasets arising in social networks, neuroscience studies and weather patterns. Fitting GCP with alternative statistical assumptions provides new ways to explore trends in the data and yields improved predictions, e.g., of social network and mouse neural data.

The third heterogeneity is in the *class* of samples, and we consider learning a mixture of low-dimensional subspaces. This model supposes that each sample comes from one of several (unknown) low-dimensional subspaces, that taken together form a union of subspaces (UoS). Samples from the same class come from the same subspace in the union. We consider an ensemble algorithm that clusters the samples,

and analyze the approach to provide recovery guarantees. Finally, we propose a *sequence* of unions of subspaces (SUoS) model that systematically captures samples with heterogeneous complexity, and we describe some early ideas for learning and using SUoS models in patch-based image denoising.

CHAPTER I

Introduction

Modern data analysis increasingly involves extracting insights and patterns from large and heterogeneous data. New technologies and increasing computational power are enabling us to collect and process more data from more sources than ever before, opening up exciting new opportunities for discovery. Recent examples of this trend abound in social networks, business, medicine, and astronomy, to name just a few. Obtaining meaningful insights from all this raw and messy data requires data analysis techniques that are both computationally efficient and statistically sound. Efficient methods are necessary for handling the scale of modern datasets, and soundness enables us to trust and reason about their outputs. This dissertation aims to address these challenges of modern data analysis in a few specific but fundamental settings; we discuss some ideas for future work and open problems at the end of the dissertation.

1.1 Low-dimensional models for high-dimensional data

Finding trends and patterns in data entails finding structure among the variables measured, i.e., ways in which the data are correlated rather than unrelated. Intuitively, there are often fewer actual "degrees of freedom" than variables measured, and we seek to find this lower-dimensional structure. For a simple and illustrative example, consider measuring the arm spans and heights of many people. Two variables, arm span and height, are measured but these two are actually tightly correlated, and much of the data is well-represented by a single variable capturing, roughly speaking, how large a person is. Such trends can be easy to visualize and spot when considering only a few measured variables, but we often measure numerous variables, i.e., the data are high-dimensional. Low-dimensional modeling seeks to find these trends in an automatic and principled way.

This dissertation focuses on low-dimensional linear models. The first two works

center specifically on Principal Component Analysis (PCA). The third centers on the Canonical Polyadic (CP) tensor decomposition; one can view this technique as a generalization of PCA to tensor-shaped data. The final two works center on union of subspace models; one can view these as generalizations of PCA to data arising from a mixture. Chapter II describes these three models in detail and provides some relevant mathematical background.

1.2 Data with heterogeneous quality

The first two contributions (Chapters III and IV) consider data with heterogenous quality. Specifically, samples have heterogeneous noise variances; some samples are noisier than others. A few natural questions arise for such data:

a) What is the performance of standard techniques that do not explicitly account for this heteroscedasticity? What is the "cost" of heteroscedasticity?

b) How should we adjust standard techniques to account for varying data quality?

Chapter III addresses (a) for PCA, a standard and widely used technique for dimensionality reduction. We provide expressions for the asymptotic performance of PCA, and our analysis quantifies the impact of heteroscedasticity. It reveals that PCA performance is always better for homoscedastic noise of the same average noise variance or of the same average inverse noise variance. As a result, both these average measures of noise give overly optimistic impressions of PCA performance for heteroscedastic noise.

Chapter IV addresses (b) by considering a weighted variant of PCA that gives less weight to noisier samples. The key question for weighted PCA becomes how best to assign weights; how much less weight should be given to samples that are twice as noisy? A natural approach is to assign weights reciprocal to noise variance, i.e., to give half the weight to samples with twice the noise variance. Another pragmatic approach is to exclude noisier samples, i.e., to give zero weight to samples with larger noise variance. Standard PCA gives equal weight to all samples. We provide expressions for the asymptotic performance of weighted PCA for any choice of weights. Our analysis reveals that none of the above weights are optimal, and we derive the optimal weights.

1.3 Data with heterogeneous statistical assumptions

The third contribution (Chapter V) generalizes CP tensor decomposition to allow for a greater variety of statistical assumptions than the Gaussian model with

homogeneous variances implicit in traditional CP. The generalized CP (GCP) tensor decomposition unifies various modifications of CP for differing statistical assumptions into a single algorithmic framework, and furthermore readily allows for heterogeneous statistical assumptions (and data types) throughout the data. Allowing for this generality is useful for modern data analysis since it enables the data analyst to

- easily incorporate domain knowledge about the statistical uncertainties in the data, i.e., to utilize existing statistical models for the data, and
- rapidly experiment with various notions of fit when available domain expertise does not clearly specify the appropriate choice.

As illustrated in Section 5.5, this flexibility provides a variety of new lenses through which to view multiway data and obtain new insights. Fitting GCP involves solving a new optimization problem that lacks the structure typically exploited in fitting CP, so the primary challenge addressed in Chapter V is how to carry out this optimization using practical techniques.

1.4 Data with heterogeneous linear structure

The final two contributions (Chapters VI and VII) consider unions of subspaces. Samples in this case are modeled as each lying close to one of several subspaces, or in other words, as lying in a mixture of low-dimensional linear models with class corresponding to subspace. This generalization of subspace models makes it appropriate for broader types of signals, such as those arising in computer vision, where no single subspace may be able to parsimoniously represent all the data. As an example, consider images of handwritten digits $0, 1, \ldots, 9$. Once appropriately aligned, it turns out that images of the same digit lie close to a low-dimensional subspace. However, the digits taken all together span a much higher-dimensional subspace; this data is more naturally modeled by a union of ten low-dimensional subspaces, one for each digit.

A major challenge is in developing algorithms that can efficiently and reliably fit union of subspace models. Chapter VI proposes one such algorithm, Ensemble K-subspaces, and provides partial guarantees for when it correctly identifies which samples came from the same subspace. Another interesting avenue is to draw connections to dictionary (or transform) sparsity models since they have a similar flavor to unions of subspaces. Chapter VII studies this connection, observed previously by many authors, and proposes a generalization for unions of subspaces that more firmly cements the connection by accounting for *heterogeneous* sparsity. We propose

a procedure for fitting the proposed sequence of unions of subspaces (SUoS) model, and demonstrate its potential benefits with an application to image denoising.

1.5 Organization of dissertation

Chapter II introduces the models considered in this dissertation in addition to some of the relevant mathematical tools. We also discuss some connections to medical imaging. While the topics in this dissertation apply broadly to modern data analysis, many of the questions asked were motivated by various challenges in imaging, and we take this opportunity to describe some of the connections. The subsequent chapters are based on the following papers.

Chapter III:
[94] David Hong, Laura Balzano, and Jeffrey A. Fessler. Towards a theoretical analysis of PCA for heteroscedastic data. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, September 2016. doi: 10.1109/allerton.2016.7852272.
[95] David Hong, Laura Balzano, and Jeffrey A. Fessler. Asymptotic performance of PCA for high-dimensional heteroscedastic data. *Journal of Multivariate Analysis*, 167:435–452, September 2018. doi: 10.1016/j.jmva.2018.06.002.

Chapter IV:
[96] David Hong, Jeffrey A. Fessler, and Laura Balzano. Optimally Weighted PCA for High-Dimensional Heteroscedastic Data, 2018. Submitted. arXiv: 1810.12862v2.

Chapter V:
[97] David Hong, Tamara G. Kolda, and Jed A. Duersch. Generalized Canonical Polyadic Tensor Decomposition. *SIAM Review*, 2019. To appear. arXiv: 1808.07452v2.

Chapter VI:
[98] David Hong*, John Lipor*, Yan Shuo Tan, and Laura Balzano. Subspace Clustering using Ensembles of K-Subspaces, 2018. Submitted. (*equal contribution). arXiv: 1709.04744v2.

Chapter VII:
[99] David Hong, Robert P. Malinas, Jeffrey A. Fessler, and Laura Balzano. Learning Dictionary-Based Unions of Subspaces for Image Denoising. In *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, September 2018. doi: 10.23919/eusipco.2018.8553117.

Chapter VIII concludes the dissertation with a discussion of some ideas for further work and open problems.

CHAPTER II

Background

This chapter introduces the low-dimensional models considered in this dissertation, as well as some relevant mathematical background and tools. We also briefly discuss connections to challenges in medical imaging that motivated some of the work on these topics for the author, though the work in this dissertation remains broadly applicable to modern data analysis and is otherwise presented in that generality. The sections after Section 2.1 are somewhat self-contained and can be read in any order.

2.1 Notations and Conventions

We define a few general notational conventions that we use throughout this dissertation. First, the fields of real and complex numbers are denoted in blackboard bold as $\mathbb{R}$ and $\mathbb{C}$, respectively. The real line restricted to nonnegative values is denoted as $\mathbb{R}_+$. Scalars and vector variables are typically lowercase and in normal weight, e.g., $\alpha \in \mathbb{R}_+$ or $u \in \mathbb{C}^d$. Matrices, on the other hand, are typically uppercase and bold, e.g., $\mathbf{U} \in \mathbb{R}^{d \times k}$. Tensors are typically denoted by uppercase bold Euler font, e.g., $\mathcal{X} \in \mathbb{C}^{m \times n \times p}$. Typically hats are used to decorate estimates or other quantities derived *from data*, e.g., $\hat{u} \in \mathbb{C}^d$ may be a principal component derived from data.

Superscript $\top$ and H denote non-conjugate and conjugate transpose, respectively, and vectors are treated as column matrices so that $v^{\mathsf{H}}u$ is an inner product and uv^{H} is an outer product. We also notate inner products as $\langle u, v \rangle = v^{\mathsf{H}}u$ and outer products as $u \circ v = uv^{\top}$. Linear span is notated by $\operatorname{span}(\cdot)$, trace by $\operatorname{tr}(\cdot)$ and determinant by $\det(\cdot)$. Uppercase calligraphy is typically used for sets, an primarily for subspaces, e.g., $\mathcal{S} = \operatorname{span}(e_1, e_2)$ is the subspace spanned by e_1 and e_2.

A couple miscellaneous notations are the Kronecker delta δ_{ij}, which is one if $i = j$ and zero otherwise, and the Dirac delta distribution δ_x centered at x. Typically it will be clear from the context which is meant. Another operation we will find convenient is the rectifier $(\cdot)_+ = \max(0, \cdot)$ that simply truncates at zero.

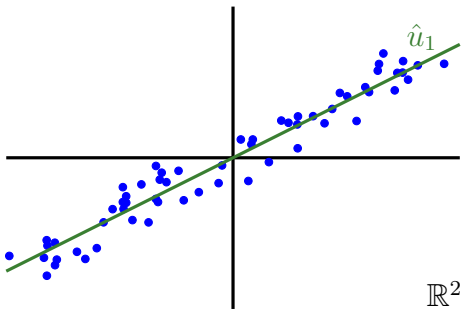


Figure 2.1: Illustration of PCA for sample vectors in $\mathbb{R}^2$, i.e., with two variables.

2.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a classical method for reducing the dimensionality of data by representing them in terms of a new set of variables, called principal components, where variation in the data is largely captured by the first few principal components. This section provides a brief introduction to this standard and ubiquitous data analysis technique, and places it in the context of low-dimensional subspace learning and matrix factorization. There are numerous motivations and derivations of PCA. See the textbook [114] for a comprehensive treatment.

The input to PCA is a sequence of sample vectors $y_1, \dots, y_n \in \mathbb{C}^d$ that are typically centered to have zero mean as a preprocessing step. Namely, the jth centered sample is formed as

$$\tilde{y}_j = y_j - \underbrace{\frac{1}{n}(y_1 + \cdots + y_n)}_{\text{empirical mean}}. \tag{2.1}$$

For simplicity, we will typically suppose the given sample vectors are already zero mean and work directly with the samples $y_1, \dots, y_n$. Each entry of a sample vector often corresponds to a measured variable, e.g., the temperature at a particular location. When the measured variables have different units, it is common to replace them with standardized (unitless) versions.

PCA seeks unit norm vectors called principal components[1] $\hat{u}_1, \dots, \hat{u}_k \in \mathbb{C}^d$ that maximally capture the variability in the data. Namely, the first principal component $\hat{u}_1$ maximizes the variance in the direction of $\hat{u}_1$, i.e.,

$$\hat{u}_1 \in \operatorname*{argmax}_{u:\|u\|_2=1} \frac{1}{n} \sum_{j=1}^{n} |\langle y_j, u \rangle|^2, \tag{2.2}$$

[1] In contrast to [114], "principal components" throughout this dissertation refers to the unit norm direction vectors and "scores" refers to the derived variables, i.e., the coefficients of the samples with respect to the components.

as illustrated in Fig. 2.1. Subsequent principal components solve the same optimization problem over perpendicular unit norm vectors:

$$\hat{u}_2 \in \operatorname*{argmax}_{\substack{u:\|u\|_2=1 \\ u^{\mathsf{H}}\hat{u}_1=0}} \frac{1}{n}\sum_{j=1}^{n} |\langle y_j, u\rangle|^2, \qquad \hat{u}_3 \in \operatorname*{argmax}_{\substack{u:\|u\|_2=1 \\ u^{\mathsf{H}}\hat{u}_1=0 \\ u^{\mathsf{H}}\hat{u}_2=0}} \frac{1}{n}\sum_{j=1}^{n} |\langle y_j, u\rangle|^2, \qquad \dots \tag{2.3}$$

The (empirical) variance in the direction of each principal component $\hat{u}_i$ is its corresponding amplitude, given by

$$\hat{\theta}_i^2 := \frac{1}{n}\sum_{j=1}^{n} |\langle y_j, \hat{u}_i\rangle|^2 \in \mathbb{R}_+. \tag{2.4}$$

Finally, each principal component $\hat{u}_i$ produces a new variable for each sample; these new variables can be used as a low-dimensional representation of the samples, e.g., for visualization. Collecting the values for all n samples produces the score vector

$$\hat{z}_i = \frac{1}{\hat{\theta}_i}(\langle y_1, \hat{u}_i\rangle, \dots, \langle y_n, \hat{u}_i\rangle)^{\mathsf{H}} \in \mathbb{C}^n \tag{2.5}$$

associated with the ith principal component $\hat{u}_i$. Note that dividing by $\hat{\theta}_i$ standardizes the scores to have unit (empirical) variance.

2.2.1 PCA in terms of the covariance matrix

Observe that the objective in (2.2) can be rewritten as

$$\frac{1}{n}\sum_{j=1}^{n} |\langle y_j, u\rangle|^2 = \frac{1}{n}\sum_{j=1}^{n} u^{\mathsf{H}} y_j y_j^{\mathsf{H}} u = u^{\mathsf{H}} \underbrace{\left(\frac{1}{n}\sum_{j=1}^{n} y_j y_j^{\mathsf{H}}\right)}_{\text{covariance matrix}} u = u^{\mathsf{H}}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}}u, \tag{2.6}$$

where $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}}$ is a unitary eigendecomposition of the (positive semi-definite) sample covariance matrix. $\mathbf{V} = (v_1, \dots, v_d)$ is a unitary matrix whose orthonormal columns $v_1, \dots, v_d \in \mathbb{C}^d$ are eigenvectors corresponding to the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ given in decreasing order by the diagonal entries of $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$. Covariance matrices are positive semi-definite with nonnegative eigenvalues, so (2.6) is maximized with value λ_1 when $\mathbf{V}^{\mathsf{H}}u = (1, 0, \dots, 0)^{\mathsf{H}}$, i.e., when $\hat{u}_1 = v_1$.

Likewise, (2.6) is maximized among unit norm vectors orthogonal to $\hat{u}_1$ by the second principal eigenvector $\hat{u}_2 = v_2$ with corresponding amplitude $\hat{\theta}_2 = \lambda_2$. Continuing along these lines yields that the principal components $\hat{u}_1, \dots, \hat{u}_k$ are the principal eigenvectors of the sample covariance matrix $(y_1 y_1^{\mathsf{H}} + \cdots + y_n y_n^{\mathsf{H}})/n$ and the amplitudes $\hat{\theta}_1^2, \dots, \hat{\theta}_k^2$ are the associated eigenvalues. Geometrically, (2.6) is a paraboloid whose contours are ellipses centered at the origin with principal axes $\lambda_1 v_1, \dots, \lambda_d v_d$, and these principal axes are exactly the principal components and amplitudes.

2.2.2 PCA as low-dimensional subspace learning

PCA can also be thought of as least-squares fitting of a low-dimensional subspace to data. Consider the problem of finding a k-dimensional subspace $\hat{\mathcal{S}}$ of $\mathbb{C}^d$ that minimizes the mean square residual

$$\hat{\mathcal{S}} \in \operatorname*{argmin}_{\mathcal{S}\in \mathrm{Gr}(k,\mathbb{C}^d)} \frac{1}{n}\sum_{j=1}^{n} \|y_j - \mathcal{P}_{\mathcal{S}} y_j\|_2^2, \tag{2.7}$$

where the Grassmannian $\mathrm{Gr}(k,\mathbb{C}^d)$ is the set of k-dimensional subspaces of $\mathbb{C}^d$, and $\mathcal{P}_{\mathcal{S}} : \mathbb{C}^d \to \mathcal{S}$ is the associated (orthogonal) projection operator. Projection onto a subspace satisfies Pythagorean theorem, i.e., $\|y - \mathcal{P}_{\mathcal{S}} y\|_2^2 + \|\mathcal{P}_{\mathcal{S}} y\|_2^2 = \|y\|_2^2$ for any $y \in \mathbb{C}^d$, so the optimization problem (2.7) is equivalently

$$\hat{\mathcal{S}} \in \operatorname*{argmax}_{\mathcal{S}\in \mathrm{Gr}(k,\mathbb{C}^d)} \frac{1}{n}\sum_{j=1}^{n} \|\mathcal{P}_{\mathcal{S}} y_j\|_2^2. \tag{2.8}$$

Moreover, one can equivalently optimize over orthonormal bases for k-dimensional subspaces given by orthonormal vectors $u_1, \ldots, u_k \in \mathbb{C}^d$, yielding

$$(\hat{u}_1, \ldots, \hat{u}_k) \in \operatorname*{argmax}_{u_1,\ldots,u_k\in\mathbb{C}^d} \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} |\langle y_j, u_i\rangle|^2 \qquad \text{s.t.} \quad \forall_{i,j}\ \langle u_i, u_j\rangle = \delta_{ij}, \tag{2.9}$$

since the projection of any $y \in \mathbb{C}^d$ onto the subspace $\mathcal{S} = \mathrm{span}(u_1, \ldots, u_k)$ and the resulting squared norm are given by

$$\mathcal{P}_{\mathcal{S}} y = \sum_{i=1}^{k} \langle y, u_i\rangle u_i, \qquad\qquad \|\mathcal{P}_{\mathcal{S}} y\|_2^2 = \sum_{i=1}^{k} |\langle y, u_i\rangle|^2. \tag{2.10}$$

Rewriting the objective in (2.9) as in (2.6) yields the objective

$$\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} |\langle y_j, u_i\rangle|^2 = \sum_{i=1}^{k}\left\{\frac{1}{n}\sum_{j=1}^{n} |\langle y_j, u_i\rangle|^2\right\} = \sum_{i=1}^{k} u_i^{\mathsf{H}} \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}} u_i, \tag{2.11}$$

where $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{H}}$ is again a unitary eigendecomposition of the (positive semi-definite) sample covariance matrix $(y_1 y_1^{\mathsf{H}} + \cdots + y_n y_n^{\mathsf{H}})/n$. As before, this objective is maximized by the first k principal eigenvectors $v_1, \ldots, v_k$ of the sample covariance matrix with maximum value $\lambda_1 + \cdots + \lambda_k$. Namely, the principal components $\hat{u}_1, \ldots, \hat{u}_k$ form an orthonormal basis for a least squares subspace fit, connecting PCA to low-dimensional subspace learning.

A subtle distinction is that subspace learning seeks a *subspace* $\hat{\mathcal{S}}$, while PCA seeks a set of principal component *vectors* $\hat{u}_1, \ldots, \hat{u}_k$. Hence, any orthonormal basis spanning the same subspace as the principal components also solves (2.9). Namely, for

any unitary $k \times k$ matrix $\mathbf{Q}$, the columns of $(\hat{u}_1, \ldots, \hat{u}_k)\mathbf{Q}$ form an orthonormal basis with equivalent span, and hence equivalent objective function value in (2.9). These vectors need not, however, be principal components since each may not maximally capture variance orthogonal to its preceding components; principal components form an orthonormal basis aligned with the principal axes of the sample covariance.

A natural setting for low-dimensional subspace learning is when we believe the samples are noisy measurements of signal vectors $x_1, \ldots, x_n \in \mathcal{S}^\star$ from some underlying subspace $\mathcal{S}^\star \subset \mathbb{C}^d$. The least-squares objective is especially natural when the noise is assumed to be isotropic Gaussian that is independent and identically distributed across samples, namely:

$$y_j = x_j + \varepsilon_j, \qquad j \in \{1, \ldots, n\}, \tag{2.12}$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$, and σ^2 is a homogeneous noise variance across all samples. Projecting onto the estimated subspace $\hat{\mathcal{S}}$ from (2.7) then yields estimates of the underlying signals $x_1, \ldots, x_n$. This projection can be written in a few equivalent ways that highlights the connection to principal components, amplitudes and scores:

$$\hat{x}_j := \mathcal{P}_{\hat{\mathcal{S}}} y_j = \sum_{i=1}^{k} \langle y_j, \hat{u}_i \rangle \hat{u}_i = \sum_{i=1}^{k} \hat{\theta}_i \hat{u}_i \big(\hat{z}_i^{(j)}\big)^*, \tag{2.13}$$

where the first equality arises from (2.10) and the second equality follows from (2.5). The quality of the estimates $\hat{x}_1, \ldots, \hat{x}_n$ depends on how well $\hat{u}_1, \ldots, \hat{u}_n$ capture $\mathcal{S}^\star$, so it is important to understand the performance of PCA. We study this question for heterogeneous noise in Chapters III and IV.

2.2.3 PCA as approximate low-rank matrix factorization

Yet another view into PCA is through the lens of matrix factorization; Chapters III and IV use this connection to apply tools from random matrix theory to analyze PCA. Consider the data matrix whose columns are the sample vectors

$$\mathbf{Y} := (y_1, \ldots, y_n) \in \mathbb{C}^{d \times n}. \tag{2.14}$$

In terms of the data matrix, the PCA objective in (2.2) and (2.3) becomes

$$\frac{1}{n} \sum_{j=1}^{n} |\langle y_j, u \rangle|^2 = \frac{1}{n} \|(\langle y_1, u \rangle, \ldots, \langle y_n, u \rangle)\|_2^2 = \frac{1}{n} \|u^{\mathsf{H}} \mathbf{Y}\|_2^2 \propto \|u^{\mathsf{H}} \mathbf{Y}\|_2^2. \tag{2.15}$$

so it follows that the first k principal components $\hat{u}_1, \ldots, \hat{u}_k$ are the first k left singular vectors of the data matrix $\mathbf{Y}$. Likewise, the ith amplitude $\hat{\theta}_i = \|\hat{u}_i^{\mathsf{H}} \mathbf{Y}\|_2 / \sqrt{n}$ is the ith largest singular value of $\mathbf{Y}$ divided by $\sqrt{n}$ and the ith score vector $\hat{z}_i = (\hat{u}_i^{\mathsf{H}} \mathbf{Y})^{\mathsf{H}} / \hat{\theta}_i$ is

the ith right singular vector of $\mathbf{Y}$ multiplied by $\sqrt{n}$. As a result, characterizing the properties of principal components, amplitudes and scores is equivalent to studying the singular value decomposition (SVD) of the data matrix $\mathbf{Y}$.

Reconstructing using the first k principal components yields

$$\begin{aligned}(2.16)\qquad \hat{\mathbf{X}} := (\hat{x}_1, \dots, \hat{x}_n) &= \left(\sum_{i=1}^{k} \hat{\theta}_i \hat{u}_i \big(\hat{z}_i^{(1)}\big)^*, \dots, \sum_{i=1}^{k} \hat{\theta}_i \hat{u}_i \big(\hat{z}_i^{(n)}\big)^* \right) \\ &= \sum_{i=1}^{k} \hat{\theta}_i \hat{u}_i \hat{z}_i^{\mathsf{H}} = \underbrace{(\hat{u}_1, \dots, \hat{u}_k)}_{\hat{\mathbf{U}} \in \mathbb{C}^{d \times k}} \underbrace{\operatorname{diag}(\hat{\theta}_1, \dots, \hat{\theta}_k)}_{\hat{\mathbf{\Theta}} \in \mathbb{R}_+^{k \times k}} \underbrace{(\hat{z}_1, \dots, \hat{z}_k)^{\mathsf{H}}}_{\hat{\mathbf{Z}} \in \mathbb{C}^{n \times k}} = \hat{\mathbf{U}} \hat{\mathbf{\Theta}} \hat{\mathbf{Z}}^{\mathsf{H}},\end{aligned}$$

and so the principal components, amplitudes and scores form a matrix factorization of $\hat{\mathbf{X}}$. Furthermore, (2.16) corresponds exactly to a rank-k truncated SVD of $\mathbf{Y}$. As a result, the principal components, amplitudes and scores form an approximate low-rank matrix factorization of $\mathbf{Y}$, namely [63]

$$(2.17)\qquad \sum_{i=1}^{k} \hat{\theta}_i \hat{u}_i \hat{z}_i^{\mathsf{H}} = \hat{\mathbf{U}} \hat{\mathbf{\Theta}} \hat{\mathbf{Z}}^{\mathsf{H}} \in \operatorname*{argmin}_{\mathbf{X} \in \mathbb{C}^{d \times n}} \|\mathbf{X} - \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \operatorname{rank} \mathbf{X} \le k.$$

Writing the sample covariance matrix in terms of $\mathbf{Y}$ yields the Gram matrix

$$(2.18)\qquad \frac{1}{n} \sum_{j=1}^{n} y_j y_j^{\mathsf{H}} = \frac{1}{n} \mathbf{Y} \mathbf{Y}^{\mathsf{H}}.$$

The eigenvalues and eigenvectors of $\mathbf{Y}\mathbf{Y}^{\mathsf{H}}$ are, respectively, the square singular values and left singular vectors of $\mathbf{Y}$, drawing a direct connection to the eigendecomposition in Section 2.2.1. Furthermore, projection onto a subspace $\mathcal{S} = \operatorname{span}(u_1, \dots, u_k)$ with orthonormal basis given by columns of a matrix $\mathbf{U} = (u_1, \dots, u_k) \in \mathbb{C}^{d \times k}$ is $\mathcal{P}_{\mathcal{S}} y = \mathbf{U}\mathbf{U}^{\mathsf{H}} y$ for any $y \in \mathbb{C}^d$. As a result, (2.7) can be written in terms of $\mathbf{U}$ and $\mathbf{Y}$ as

$$(2.19)\qquad \hat{\mathbf{U}} \in \operatorname*{argmin}_{\mathbf{U} \in \mathbb{C}^{d \times k}} \frac{1}{n} \|\mathbf{Y} - \mathbf{U}\mathbf{U}^{\mathsf{H}} \mathbf{Y}\|_F^2 \qquad \text{s.t.} \quad \mathbf{U}^{\mathsf{H}} \mathbf{U} = \mathbf{I}_k,$$

providing a simple relationship to subspace fitting as described in Section 2.2.2.

2.2.4 Asymptotic PCA performance and random matrix theory

A key question in understanding PCA performance is analyzing how well the principal components, amplitudes and scores recover underlying counterparts from noisy observations. This section introduces asymptotic approaches to this problem to provide some preparation for the work of Chapters III and IV and to describe

some of the connections to random matrix theory. See the recent survey [112] for an excellent overview of this topic.

Consider a data matrix (2.14) with noisy observations of underlying components

$$\mathbf{Y} := (y_1, \dots, y_n) = \sum_{i=1}^{k} \theta_i u_i z_i^{\mathsf{H}} + \sigma \underbrace{(\varepsilon_1, \dots, \varepsilon_n)}_{=\mathbf{E} \in \mathbb{C}^{d \times n}} = \sum_{i=1}^{k} \theta_i u_i z_i^{\mathsf{H}} + \sigma \mathbf{E}, \tag{2.20}$$

where $u_1, \dots, u_k \in \mathbb{C}^d$ are underlying (orthonormal) components, $\theta_1 > \cdots > \theta_k \in \mathbb{R}_+$ are underlying amplitudes,[2] and $z_1, \dots, z_k \in \mathbb{C}^n$ are underlying (orthonormal) score vectors. Typically, one models the noise matrix $\mathbf{E}$ as random, yielding a data matrix $\mathbf{Y}$ that is a rank-k latent (or signal) matrix plus a random noise matrix. The question becomes: how close are the principal components $\hat{u}_1, \dots, \hat{u}_k$, amplitudes $\hat{\theta}_1, \dots, \hat{\theta}_k$ and score vectors $\hat{z}_1, \dots, \hat{z}_k$ to their underlying counterparts?

Consider the first principal component $\hat{u}_1$. How close in angle is $\hat{u}_1$ to u_1, i.e., how close is the inner product $|\langle \hat{u}_1, u_1 \rangle|^2$ to one? Observe first that $|\langle \hat{u}_1, u_1 \rangle|^2$ is in fact a random variable because $\mathbf{Y}$ is a random matrix as a result of randomness in $\mathbf{E}$, making $\hat{u}_1$ a random vector. Furthermore, the columns of $\mathbf{Z}^{\mathsf{H}} = (z_1, \dots, z_k)^{\mathsf{H}} \in \mathbb{C}^{k \times n}$ are often modeled as being n i.i.d. random vectors to produce i.i.d. samples, providing another source of randomness. However, considering *well-chosen* asymptotic regimes can produce limiting behavior that is instead *deterministic* and easier to reason with, while remaining similar enough to provide useful insights for non-asymptotic settings.

One natural limit to consider is $n \to \infty$, i.e., numerous samples. Suppose both $\mathbf{Z}$ and $\mathbf{E}$ have i.i.d. normal entries (mean zero, variance one), yielding i.i.d. Gaussian sample vectors[3]

$$y_1, \dots, y_n \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \sum_{i=1}^{k} \theta_i^2 u_i u_i^{\mathsf{H}} + \sigma^2 \mathbf{I}_d\right). \tag{2.21}$$

When $n \to \infty$, the sample covariance matrix of i.i.d. samples consistently estimates the associated population covariance matrix:

$$\frac{1}{n} \sum_{j=1}^{n} y_j y_j^{\mathsf{H}} \xrightarrow{\text{a.s.}} \mathbb{E}(Y Y^{\mathsf{H}}) \quad \text{as} \quad n \to \infty, \tag{2.22}$$

where $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence, i.e., convergence with probability one. Moreover, the top k eigenvalues and eigenvectors of the sample covariance converge, yielding $\hat{\theta}_1^2 \xrightarrow{\text{a.s.}} \theta_1^2 + \sigma^2, \dots, \hat{\theta}_k^2 \xrightarrow{\text{a.s.}} \theta_k^2 + \sigma^2$ and $|\langle \hat{u}_1, u_1 \rangle|^2, \dots, |\langle \hat{u}_1, u_1 \rangle|^2 \xrightarrow{\text{a.s.}} 1$.

[2] Using distinct amplitudes simplifies discussion here; Chapters III and IV allow for equal amplitudes.

[3] The covariance is a scaled identity perturbed by the addition of k *spikes*. This data model is a type of Johnstone spiked covariance model [110, 111]; see also Section 3.8.1 for discussion of some subtle aspects of this connection.

However, this limit is not well-suited for modern high-dimensional data where the number of variables d is comparable to, or even larger than, the number of samples n, taking us quite far from the regime where $n \to \infty$ while d is fixed. To handle these new settings, we instead consider the limiting behavior as $n, d \to \infty$ with $n/d \to c > 0$. Since d is no longer fixed, note that the $d \times d$ covariance matrix grows in this limit; we will, however, assume that the number of components k and their amplitudes $\theta_1, \dots, \theta_k$ are fixed.

In the limit $n, d \to \infty$ with $n/d \to c$, the principal components no longer consistently estimate the underlying components, but, perhaps surprisingly, the angle between each principal component and its underlying counterpart still converges almost surely to a deterministic number. In particular, in this homoscedastic setting where the noise $\sigma\mathbf{E}$ has i.i.d. entries,

$$|\langle \hat{u}_i, u_i \rangle| \xrightarrow{\text{a.s.}} \left\{ \frac{c - (\sigma/\theta_i)^4}{c + (\sigma/\theta_i)^2} \right\}_+ \quad \text{as} \quad n, d \to \infty \text{ with } n/d \to c > 0, \tag{2.23}$$

for each $i \in \{1, \dots, k\}$. The recovery (2.23) depends on the samples per dimension c and the noise to signal ratio σ/θ_i. A large number of samples per dimension relative to the noise to signal ratio, i.e., $c \gg (\sigma/\theta_i)^4$, is needed for recovery close to one. When $c < (\sigma/\theta_i)^4$, the recovery is zero and the principal component is asymptotically orthogonal to the underlying component. Namely, there is a phase transition at $c = (\sigma/\theta_i)^4$ between positive recovery and no recovery; the asymptotic recovery does not smoothly approach zero, e.g., as the noise variance σ^2 increases.

The expression (2.23) for asymptotic recovery in the high-dimensional regime provides an elegant tool for understanding the behavior of principal components under the homoscedastic model (2.21). Chapter III analyzes the more general heteroscedastic setting where samples have potentially heterogeneous noise variances, and Chapter IV goes a step further by extending the analysis to weighted PCA, where samples are also given heterogeneous weight in PCA to account for the heterogeneous noise. The analyses are based on the perturbation technique of [22] that relates the singular values and vectors of low-rank plus random matrices like $\mathbf{Y}$ in (2.20) to (an integral with respect to) the singular values of the noise $\mathbf{E}$.

The analyses rely on a surprising fact from random matrix theory: the singular value distributions of these noise matrices (appropriately normalized) converge almost surely to deterministic distributions. Consider, for example, the above noise matrix $\mathbf{E} \in \mathbb{C}^{d \times n}$ with i.i.d. normal entries (mean zero, variance one), suppose $d \leq n$ for simplicity, and let $\tilde{\mathbf{E}} := \mathbf{E}/\sqrt{n} \in \mathbb{C}^{d \times n}$ be the normalized version. Any draw of the random matrix $\tilde{\mathbf{E}}$ has d singular values $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$ with corresponding (empirical) singular value distribution $\hat{\mu} := (\delta_{\lambda_1} + \cdots + \delta_{\lambda_d})/d$, where δ_λ is the Dirac delta distribution centered at λ. Since $\tilde{\mathbf{E}}$ is a random matrix, the associated singular

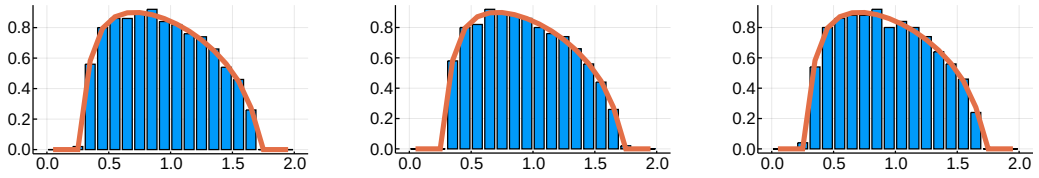


Figure 2.2: Histograms visualizing the empirical singular value distributions of three 500×1000 random matrices, each generated with i.i.d. (mean zero, variance $1/1000$) Gaussian entries. Overlaid is the Marcenko-Pastur distribution (2.24) in orange. The empirical singular value distribution is random, as indicated by slight differences among the three, but all are concentrating around the limiting Marcenko-Pastur distribution (2.24).

value distribution $\hat{\mu}$ is also random. However, this random singular value distribution converges almost surely [14, Chapter 3]: $\hat{\mu} \xrightarrow{\text{a.s.}} \mu_{\tilde{\mathbf{E}}}$ as $n, d \to \infty$ with $n/d \to c$ where the limiting singular value distribution $\mu_{\tilde{\mathbf{E}}}$ is deterministic and has density

$$d\mu_{\tilde{\mathbf{E}}}(x) = \frac{\sqrt{4c - (cx^2 - c - 1)^2}}{\pi x} \mathbb{1}_{(1-1/\sqrt{c}, 1+1/\sqrt{c})}(x) dx. \tag{2.24}$$

This distribution, shown in Fig. 2.2, is the Marcenko-Pastur distribution [138]. The existence of deterministic limits for these spectral properties of random matrices is a surprising fact, and has been the subject of a large body of work; see the textbooks [9, 14, 52] for excellent overviews of this field.

2.3 Canonical Polyadic Tensor Decomposition

The canonical polyadic (CP) tensor decomposition is a generalization of PCA to tensor, or multiway, data. Whereas PCA produces a low-rank approximation of data shaped as a matrix, i.e., a two-indexed array, CP produces a low-rank approximation of data shaped as a tensor, i.e., a two- or more-indexed array. For example, Section 5.5.2 considers measurements of neural activity in mice over the course of a task that they repeat for multiple trials. This data is most naturally organized as a three-*way* tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with three *modes* corresponding to n_1 neurons, n_2 time steps and n_3 trials. One might reshape this tensor into an $n_1 n_2 \times n_3$ matrix to use PCA by flattening the neuron and time modes into a single mode; CP instead generalizes PCA to handle such data directly. For the purpose of the dissertation, this section provides a brief introduction to this increasingly standard data analysis tool; see the surveys [118, 183] for a more comprehensive introduction to this and other tensor decompositions, their properties and their many applications to data analysis.

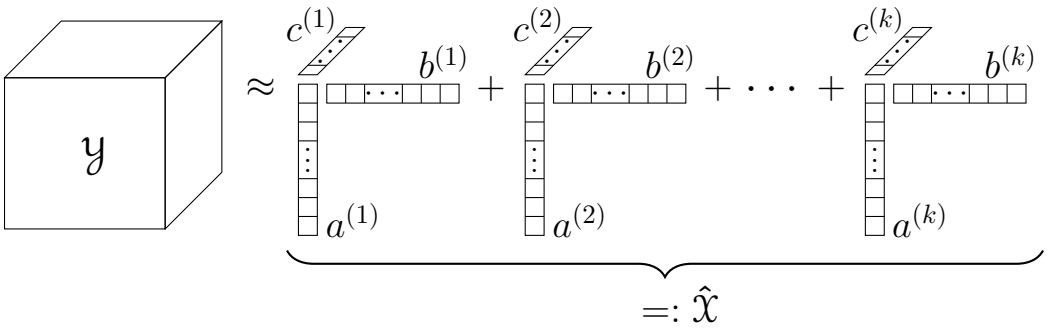


Figure 2.3: Illustration of a rank-k canonical polyadic (CP) structured 3-way tensor. The tensor is the sum of k components, each of which is the outer product of d vectors (here $d = 3$).

CP tensor decomposition seeks to approximate the data tensor with a low-rank tensor $\hat{\mathcal{X}}$, i.e., a sum of (only a few) rank-one outer products,

$$\mathcal{Y} \approx \hat{\mathcal{X}} = a^{(1)} \circ b^{(1)} \circ c^{(1)} + \cdots + a^{(k)} \circ b^{(k)} \circ c^{(k)} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \tag{2.25}$$

where $\circ$ denotes an outer product, and $a^{(1)}, \ldots, a^{(k)} \in \mathbb{R}^{n_1}$, $b^{(1)}, \ldots, b^{(k)} \in \mathbb{R}^{n_2}$ and $c^{(1)}, \ldots, c^{(k)} \in \mathbb{R}^{n_3}$ are k factors in each of the three modes, respectively. As illustrated in Fig. 2.3, the factors combine to approximate the data tensor $\mathcal{Y}$ in the same way that the principal components $\hat{u}_1, \ldots, \hat{u}_k \in \mathbb{C}^d$ and scores $\hat{z}_1, \ldots, \hat{z}_k \in \mathbb{C}^n$ in (2.16) combined to approximate the data matrix $\mathbf{Y}$. Namely, the (i, j, ℓ)th entry of the reconstructed tensor $\hat{\mathcal{X}}$ is

$$\hat{x}_{ij\ell} = \underbrace{a_i^{(1)} b_j^{(1)} c_\ell^{(1)}}_{(a^{(1)} \circ b^{(1)} \circ c^{(1)})_{ij\ell}} + \cdots + \underbrace{a_i^{(k)} b_j^{(k)} c_\ell^{(k)}}_{(a^{(k)} \circ b^{(k)} \circ c^{(k)})_{ij\ell}}, \tag{2.26}$$

where $i \in \{1, \ldots, n_1\}$, $j \in \{1, \ldots, n_2\}$ and $\ell \in \{1, \ldots, n_3\}$. Observe that while the data tensor $\mathcal{Y}$ has $n_1 n_2 n_3$ independent entries, the entries of the low-rank tensor $\hat{\mathcal{X}}$ are all determined by the $k(n_1 + n_2 + n_3)$ entries in the factors. It is this interdependence among data entries that a low-rank CP tensor decomposition captures.

2.3.1 Approximate low-rank CP tensor decomposition

We turn now to a precise description of CP tensor decomposition and its associated optimization problem. So far, we have described CP in the context of a three-way tensor, but the work in this dissertation applies to tensors with any number of modes and taking a moment to more clearly establish some notation will ease the discussion.

An $n_1 \times \cdots \times n_d$ tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ refers throughout this dissertation to a real array with d indices $i_1, \ldots, i_d$ that run from one to $n_1, \ldots, n_d$, respectively. The

number of ways d is the *order* of the tensor and each way is called a *mode*. Defining

$$n = \sqrt[d]{\prod_{k=1}^{d} n_k} \quad \text{and} \quad \bar{n} = \frac{1}{d}\sum_{k=1}^{d} n_k \tag{2.27}$$

to be the geometric and arithmetic means of the dimensions $n_1, \ldots, n_d$, the tensor has n^d entries with the sum of the dimensions given by $d\bar{n}$. The tensor is indexed by the *multiindex*

$$i := (i_1, \ldots, i_d) \in \mathcal{I} := \{\, 1, \ldots, n_1 \,\} \otimes \{\, 1, \ldots, n_2 \,\} \otimes \cdots \otimes \{\, 1, \ldots, n_d \,\}, \tag{2.28}$$

i.e., y_i is the entry of $\mathcal{Y}$ at $(i_1, \ldots, i_d)$.

A rank-k tensor $\mathcal{X}$ is one that can be expressed as a sum of k outer products[4]

$$\mathcal{X} = a_1^{(1)} \circ \cdots \circ a_d^{(1)} + \cdots + a_1^{(k)} \circ \cdots \circ a_d^{(k)}, \tag{2.29}$$

where $a_1^{(1)}, \ldots, a_1^{(k)} \in \mathbb{R}^{n_1}$, $a_2^{(1)}, \ldots, a_2^{(k)} \in \mathbb{R}^{n_2}$, and so on through $a_d^{(1)}, \ldots, a_d^{(k)} \in \mathbb{R}^{n_d}$, are k factors in each of the d modes. It is often convenient to refer to the factors for a mode together, so we define the factor matrices

$$\mathbf{A}_1 := (a_1^{(1)}, \ldots, a_1^{(k)}) \in \mathbb{R}^{n_1 \times k} \quad \cdots \quad \mathbf{A}_d := (a_d^{(1)}, \ldots, a_d^{(k)}) \in \mathbb{R}^{n_d \times k}, \tag{2.30}$$

with which we denote (2.29) compactly as $\mathcal{X} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$. Expressed in terms of both the factors and the factor matrices, the ith entry of $\mathcal{X}$ is

$$x_i = \sum_{j=1}^{k} a_1^{(j)}(i_1) \cdots a_d^{(j)}(i_d) = \sum_{j=1}^{k} \mathbf{A}_1(i_1, j) \cdots \mathbf{A}_d(i_d, j), \tag{2.31}$$

where we use parentheses here to denote indexing into a variable that already has a subscript. As before, $\mathcal{X}$ has n^d entries but is defined entirely by the $kd\bar{n} \ll n^d$ entries in the factors. The entries in $\mathcal{X}$ are not unrelated but are instead correlated through the CP structure. Put another way, $\mathcal{X}$ is parsimoniously represented by the factors.

Approximate CP tensor decomposition conventionally seeks a rank-k tensor $\mathcal{X}$ that is closest to the given data tensor $\mathcal{Y}$ by solving the optimization problem

$$\hat{\mathcal{X}} \in \operatorname*{argmin}_{\mathcal{X}:\operatorname{rank}\mathcal{X}\leq k} \|\mathcal{Y} - \mathcal{X}\|_F^2 := \sum_{i\in\mathcal{I}} (y_i - x_i)^2, \tag{2.32}$$

or in terms of factor matrices, $\hat{\mathcal{X}} = [\![\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2, \ldots, \hat{\mathbf{A}}_d]\!]$ where

$$(\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_d) \in \operatorname*{argmin}_{\substack{\mathbf{A}_1\in\mathbb{R}^{n_1\times k},\\ \ldots,\\ \mathbf{A}_d\in\mathbb{R}^{n_d\times k}}} \|\mathcal{Y} - [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]\|_F^2. \tag{2.33}$$

[4]Technically, such a tensor is rank *at most* k since it may be possible to represent it with fewer than k outer products. The rank is the minimum number of outer products needed.

In words, we seek a low-rank tensor that is a *least-squares* fit to the data tensor. As we describe in Section 5.3.1, this traditional choice of fit is most natural when data are believed to be Gaussian with homogeneous variances. Chapter V generalizes CP tensor decomposition to allow for different notions of fit, e.g., ones motivated by other statistical assumptions.

Observe that unlike PCA the factors in (2.33) are not constrained to be jointly orthogonal. Low-rank tensors with three or more modes typically have (essentially) unique factors, making the additional constraint less necessary; see [118, Section 3.2] for further discussion. Another significant difference from the matrix setting of PCA is that the best rank-k factors do not necessarily contain the best rank-$(k-1)$ factors, and in some cases a best rank-k approximation may not even exist. These aspects of tensors are beyond the scope of this dissertation, and are discussed with pointers to relevant works in [118, Section 3.3].

2.3.2 Computing CP by alternating least squares

Computing the (approximate) CP tensor decomposition of a data tensor $\mathcal{Y}$ means solving the optimization problem (2.33). The problem is not jointly convex since the entries of $\mathcal{X} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$ are products of the factor matrix entries, i.e., the optimization variables. However, the objective in (2.33) can be written as

$$\|\mathcal{Y} - [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]\|_F^2 = \|\mathbf{Y}_{(1)} - \mathbf{A}_1(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_2)^\top\|_F^2, \tag{2.34}$$

where $\mathbf{Y}_{(1)} \in \mathbb{R}^{n_1 \times (n^d/n_1)}$ is the *unfolding* or *matricization* of $\mathcal{Y}$ along the first mode, and $\odot$ denotes the Khatri-Rao product, i.e., the column-wise Kronecker product; see [118, Section 2.4] for discussion of matricization and [118, Section 2.6] for discussion of the Khatri-Rao product.

Observe that (2.34) is the usual least squares problem with respect to $\mathbf{A}_1$ and is minimized over $\mathbf{A}_1$ by $\mathbf{Y}_{(1)}\{(\mathbf{A}_d \odot \cdots \odot \mathbf{A}_2)^\top\}^\dagger$. The same is true for $\mathbf{A}_2, \ldots, \mathbf{A}_d$: (2.33) is a least squares problem with respect to each $\mathbf{A}_k$ individually and can be solved via a pseudo-inverse. This simple but powerful fact forms the basis for a standard approach to fitting CP: the alternating least squares algorithm. Roughly speaking, the algorithm initializes all the factor matrices then cycles through updating each one via least squares. Pseudo-inverses of Khatri-Rao products also turn out to have special structure that makes it possible to compute them efficiently, i.e., by pseudo-inverting only a $k \times k$ matrix [118, Equation (2.2)], making this approach a practical and fast means for the CP tensor decomposition of even large tensors. The alternation hits local minima depending on the initialization, but this issue is often effectively mitigated in practice by trying multiple initializations and selecting the run with the best fit.

Unfortunately, this structure does not surface for the general notions of fit we investigate in Chapter V. As a result, a major challenge becomes developing a more general algorithmic framework for fitting CP models; one such framework is the main contribution of Chapter V.

2.3.3 Aside: Forming tensors from data

We close this section on the CP tensor model with an aside on how data gets formed into a tensor, sometimes referred to as *quantization* of the tensor. This aspect of tensor decompositions is somewhat orthogonal to the work in this dissertation, but can have a significant practical impact and is worth discussing a bit.

As an example, consider measuring monthly rainfall in 36 regions over the course of 115 years. A natural approach is to form a 36×1380 two-way tensor, i.e., a matrix, with the first mode corresponding to region and the second mode corresponding to month. However, one could also form a $36 \times 12 \times 115$ three-way tensor with modes corresponding to region, month-in-year and year, as done in Section 5.5.3. One can even further split the year mode into decades and year-in-decade. How to quantize is one of the first questions faced by someone hoping to use a tensor decomposition. Selecting appropriate quantizations depends entirely on the application and a thorough discussion is beyond our scope; instead we highlight here a simple way quantization affects the resulting decomposition.

Consider first flattening two modes into one, e.g., flattening the month-in-year and year modes in the above rainfall example into a single month mode. Any rank-k tensor in the original quantization immediately yields a rank-at-most-k tensor in the new quantization. To see why, consider a four-way rank-k tensor $\mathcal{X}_4 = [\![\mathbf{A}_1, \ldots, \mathbf{A}_4]\!]$. Flattening its last two modes into one mode produces the three-way rank-k tensor $\mathcal{X}_3 = [\![\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \odot \mathbf{A}_4]\!]$. Intuitively, a low-rank tensor quantized into three modes has entries that are less constrained than those in its four-mode analogue. Conversely, splitting a single mode into two modes often increases the rank.

In practice, the impact of flattening two modes is that the resulting CP model may not capture patterns in the data as naturally as the more structured (higher-order) model. For example, the rainfall data described above has strong yearly patterns based on the season; a three-way tensor separates this pattern out from the year-to-year variations. Videos present another interesting quantization scenario. A natural choice is to form a three-way tensor with the first two modes corresponding to the rows and columns of each frame and the last mode corresponding to the frame. A low-rank approximation of this tensor effectively seeks low-rank structure across the spatial modes, i.e., within each frame, and some thought is needed to decide if such structure is expected. Note that a line connecting opposite corners

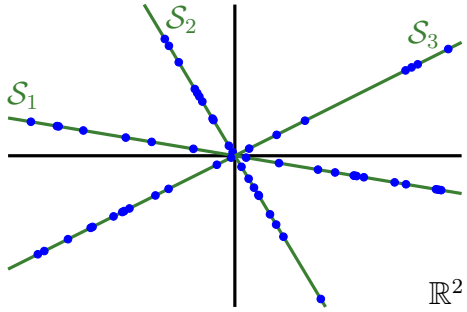


Figure 2.4: Illustration of a union of three one-dimensional subspaces for sample vectors in $\mathbb{R}^2$. No one-dimensional subspace can fit all the samples but the union of three subspaces can.

in a frame corresponds to a (full-rank) identity image matrix. Sometimes it can be more appropriate to flatten two modes into one. In the end, one often tries several quantizations; understanding the rough impact makes it easier to assess which choices may be reasonable given domain knowledge or even given patterns seen in the factors obtained for previously considered quantizations.

2.4 Unions of subspaces

Union of subspaces (UoS) models generalize PCA through the lens of subspace learning. Rather than modeling samples as all lying close to a single shared subspace, a UoS models samples as each lying close to one of several subspaces as shown in Fig. 2.4. This generality allows a UoS to model data that is too heterogeneous to be captured by a single low-dimensional subspace but that can be grouped into a few *classes* with each class well represented by a low-dimensional subspace. In Fig. 2.4, no one-dimensional subspace can fit all the samples but the combination of three subspaces does. This section provides a brief introduction to this model and a couple broad approaches to fitting it. See the textbook [207] for a comprehensive treatment of the model and its many modern applications ranging from face recognition [74] to handwritten digit recognition [84] and motion segmentation [198]. See especially [207, Chapter 5] for algebraic-geometric approaches that we skip here for brevity.

UoS learning seeks to recover a UoS given (possibly noisy) samples drawn from it. Consider a union of k subspaces $\mathcal{U} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_k \subset \mathbb{R}^D$, where $\mathcal{S}_1, \ldots, \mathcal{S}_k \subset \mathbb{R}^D$ are, respectively, $d_1, \ldots, d_k$-dimensional subspaces of the ambient space $\mathbb{R}^D$. Considering noiseless data for simplicity, the goal is to recover the UoS $\mathcal{U} \subset \mathbb{R}^d$ from given samples $y_1, \ldots, y_n \in \mathcal{U}$. A closely related task is to *cluster* the samples $y_1, \ldots, y_n$ by subspace, i.e., to put samples from $\mathcal{S}_1$ in a single cluster, samples from $\mathcal{S}_2$ in a single cluster, and so on. Subspaces to form a UoS can be obtained from a solution to

this *subspace clustering* problem by running PCA on each cluster of samples; correct clustering results in good UoS recovery. Likewise, clusters can be formed from a UoS by clustering samples to their nearest subspace,[5] and good UoS recovery will yield good clustering. Hence, for the purpose of this dissertation, we will generally treat these two goals as roughly equivalent. The remainder of this section discusses various approaches for subspace clustering in particular, and Chapter VI describes an ensemble approach to subspace clustering based on K-subspaces. Chapter VII proposes a generalization of UoS models to sequences of unions of subspaces (SUoS).

2.4.1 Self-expressive approaches

Several state of the art approaches to subspace clustering seek a parsimonious *self-expressive* representation of the samples, i.e., they express each sample in terms of the rest. Doing so leverages the key insight that samples tend to be more parsimoniously represented by other samples from the same subspace than by samples from different subspaces. A common approach is to solve an optimization problem of the form

$$\hat{\mathbf{Z}} \in \underset{\mathbf{Z}\in\mathbb{R}^{n\times n}}{\operatorname{argmin}} \; \frac{1}{n} \underbrace{\sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} z_{ji} y_j \right\|_2^2}_{=\|\mathbf{Y}-\mathbf{Y}\mathbf{Z}\|_F^2} + \lambda \|\mathbf{Z}\| \quad \text{s.t.} \quad \operatorname{diag}(\mathbf{Z}) = 0 \tag{2.35}$$

where $\mathbf{Y} := (y_1, \ldots, y_n) \in \mathbb{R}^{D\times n}$ is the data matrix, the constraint prevents samples from representing themselves, the first term in the objective encourages fidelity in the representation, and the second term typically regularizes the representation by using a parsimony-encouraging norm, such as

- the ℓ_1 norm: sparse subspace clustering (SSC) [67],
- the nuclear norm (with constraint omitted): low-rank representation [132], or
- some combination of these and other norms.

The representation $\hat{\mathbf{Z}}$ is then used to obtain a symmetric *affinity* matrix $|\hat{\mathbf{Z}}|+|\hat{\mathbf{Z}}|^\top$, so called because each entry gives an affinity between the corresponding pair of samples. Spectral clustering on this affinity matrix yields the output clusters.

Many recent approaches [135, 181, 187, 206] consider variations on (2.35) to improve robustness to noise and outliers. Self-expressive approaches typically come with theoretical results that guarantee no false connections (NFC), i.e., that points lying in different subspaces have zero affinity. See [210] for an excellent overview of the state of the art.

[5] If multiple subspaces are equally close to a sample, one might typically assign it to an arbitrary cluster. Carefully handling this case is beyond the scope of this dissertation.

2.4.2 Geometric approaches

Another group of approaches seek to leverage the geometry of a UoS more directly. An early example is the Spectral Local Best-Fit Flat (SLBF) algorithm [226]. The algorithm first identifies nearest neighbors in Euclidean distance, with the number of neighbors determined via an iterative local best-fit procedure [226, Algorithm 1], then forms pairwise affinities from the neighborhoods and applies spectral clustering. While the procedure is theoretically motivated [226, Section 2.1.1], no clustering guarantee accompanies the overall clustering approach. Greedy subspace clustering (GSC) [162] first greedily identifies *nearest subspace neighbors* by iteratively building a neighbor subspace around each sample [162, Algorithm 1], then greedily selects neighbor subspaces that approximately fit the most samples [162, Algorithm 2]. GSC is accompanied by theoretical clustering guarantees under both random and deterministic assumptions on the subspaces. Finally, thresholding-based subspace clustering (TSC) [91] chooses nearest neighbors by angle then applies spectral clustering. TSC comes with theoretical correct clustering guarantees under assumptions similar to those considered in the analysis of SSC.

2.4.3 K-subspaces

In contrast to the above methods, K-subspaces (KSS) [7, 30] seeks a least-squares fit of a union of k subspaces to the samples $y_1, \dots, y_n \in \mathbb{R}^D$. Given k subspace dimensions $d_1, \dots, d_k$, one seeks to solve the following optimization problem:

$$(\hat{\mathcal{S}}_1, \dots, \hat{\mathcal{S}}_k) \in \operatorname*{argmin}_{\substack{\mathcal{S}_1 \in \mathrm{Gr}(d_1, \mathbb{R}^D), \\ \dots, \\ \mathcal{S}_k \in \mathrm{Gr}(d_k, \mathbb{R}^D)}} \frac{1}{n} \sum_{j=1}^{n} \min_{\mathcal{S} \in \{\mathcal{S}_1, \dots, \mathcal{S}_k\}} \|y_j - \mathcal{P}_{\mathcal{S}} y_j\|_2^2, \tag{2.36}$$

where the resulting UoS is formed by taking the union $\hat{\mathcal{U}} := \hat{\mathcal{S}}_1 \cup \dots \cup \hat{\mathcal{S}}_k$. Note that $\min_{\mathcal{S} \in \{\mathcal{S}_1, \dots, \mathcal{S}_k\}} \|y_j - \mathcal{P}_{\mathcal{S}} y_j\|_2^2$ is the square residual to the UoS $\mathcal{U} := \mathcal{S}_1 \cup \dots \cup \mathcal{S}_k$.

Rewriting (2.36) by introducing cluster assignment variables yields the usual form

$$(\hat{c}_1, \dots, \hat{c}_n, \hat{\mathcal{S}}_1, \dots, \hat{\mathcal{S}}_k) \in \operatorname*{argmin}_{\substack{c_1, \dots, c_n \in \{1, \dots, k\}, \\ \mathcal{S}_1 \in \mathrm{Gr}(d_1, \mathbb{R}^D), \\ \dots, \\ \mathcal{S}_k \in \mathrm{Gr}(d_k, \mathbb{R}^D)}} \frac{1}{n} \sum_{j=1}^{n} \|y_j - \mathcal{P}_{\mathcal{S}_{c_j}} y_j\|_2^2. \tag{2.37}$$

A nice feature of the objective in (2.37) is that its minimum value depends only on the least squares residual to the best UoS fit; it is not a function of how well spread the subspaces are, providing some hope that minimizers to (2.37) may have good recovery even when subspaces are close together. Unfortunately, even approximating (2.37) turns out to be NP-hard [76].

Instead, KSS randomly initializes k candidate subspaces then alternates between the following two steps:

- cluster assignment: optimize (2.37) with respect to $\hat{c}_1, \ldots, \hat{c}_n$;
- subspace (PCA) update: optimize (2.37) with respect to $\mathcal{S}_1, \ldots, \mathcal{S}_k$.

Observe that cluster assignment amounts to assigning each sample to its nearest subspace, and the subspace update amounts to updating each subspace via subspace learning (PCA) on its assigned samples. As a result, KSS is computationally efficient with guaranteed convergence to a local minimum [30, 200]. However, the output is highly dependant on the initialization, so KSS is typically run many times after which the run with smallest cost (2.37) is chosen as the final output. Chapter VI instead combines the outputs from all the runs to form a more reliable ensemble estimate; this dissertation focuses on the current corresponding theoretical guarantees.

KSS has been extended and modified in a number of ways to improve its recovery performance. For example, [225] minimizes the sum of the residuals rather than the sum of the square residuals to improve outlier and noise robustness, [17] proposes a streaming version that can also handle missing data, and [89] proposes a streaming approach with a novel initialization based on ideas from [226]. Most recently, [76] replace PCA in the subspace update with coherence pursuit (CoP) [169].

2.5 Low-dimensional models and medical imaging

We close this background chapter with a brief discussion of some connections between low-dimensional models and medical imaging, specifically, image formation. Image formation is the problem of creating an image from data that do not correspond to direct measurements of the pixel values. Image here means a grid of d pixel (or voxel) values represented as a d-dimensional vector $x \in \mathbb{C}^d$. In many interesting and important settings, such as magnetic resonance imaging (MRI) and X-ray computed tomography (CT), the vector $y \in \mathbb{C}^m$ of m measurements from the imaging device can be reasonably modeled as linear measurements, i.e., $y \approx \mathbf{A}x$, where the *system matrix* $\mathbf{A} \in \mathbb{C}^{m \times d}$ models the imaging device. Image formation is the *inverse* problem of "inverting" this *forward* model to obtain x given y.

Typically, we expect images of interest x to not have arbitrary pixel values; this prior knowledge is what enables us to identify noise and artifacts in images. Mathematically modeling this prior knowledge holds great promise for improving the quality and safety of medical imaging systems. For example, lowering the X-ray dose in X-ray CT reduces patient exposure to radiation but results in noisier data. Image models can be used to discourage noisy images in image formation. In MRI,

forming an image from fewer measurements can help reduce scan times or can enable dynamic imaging with better temporal resolution. However, having fewer measurements than voxels, i.e., $m < d$, results in an under-determined linear system in image formation, with infinitely many images consistent with the measured data. Image models can help disambiguate among the consistent images, encouraging artifact-free images. Intuitively, good image models can help "fill the gaps" in measured data by incorporating the prior knowledge of what images should be expected.

Developing good image models and fast image formation algorithms that exploit them is both a well-established and active area of work, with significant attention in recent years on models that are *learned* from example images. A survey of this large (and rapidly growing) area is beyond the scope of this discussion. Instead, we describe a few connections to the low-dimensional models and the challenges considered in this dissertation.

2.5.1 Dictionary/transform sparsity and unions of subspaces

Dictionary sparse patch models for images suppose that patches of an image are well approximated by sparse combinations of *atomic* signals $d_1, \ldots, d_n \in \mathbb{C}^m$ that taken together form a *dictionary* $\mathbf{D} := (d_1, \ldots, d_n) \in \mathbb{C}^{m \times n}$. Namely, we model an m-pixel patch $x \in \mathbb{C}^m$ as

$$x \approx \mathbf{D}z = z_1 d_1 + \cdots + z_n d_n, \qquad \|z\|_0 \leq k, \tag{2.38}$$

where $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm that counts the number of nonzero entries of its argument. Since $\|z\|_0 \leq k$ the linear combination in (2.38) is a sparse combination of at most k of the atoms. A closely related model is the transform sparse patch model, where patches are modeled as being approximately sparse under some transform $\mathbf{T} \in \mathbb{C}^{n \times m}$. Namely

$$\mathbf{T}x \approx z, \qquad \|z\|_0 \leq k, \tag{2.39}$$

where $\mathbf{T}$ might, e.g., compute differences between neighboring pixels to encourage piecewise constant patches [176, 184]. Both models are well-motivated by the observation that images tend to be sparse in appropriate representations, and that this sparsity can be exploited to aid image formation. See [136] for some early applications in MRI.

The dictionary in (2.38) or transform in (2.39) is sometimes hand-crafted based on our intuitions about what structure we anticipate images to have. See, e.g., [136, Figure 3] and the associated discussion for examples in MRI where wavelet transform sparsity or DCT sparsity are natural choices. The opportunity to obtain dictionaries

and transforms tailored for particular types of data has also sparked significant recent work on learning dictionaries and transforms from data. Dictionary and transform learning is now a large and active area; see [8] and [172] for some of the early works in the context of image models.

In either setting, both dictionary and transform sparse models can also be viewed through the lens of unions of subspaces. Chapter VII discusses this connection in detail, and proposes a generalization to unions of subspaces that more firmly cements their relationship. The basic observation is that signals of a given sparsity level form a union of subspaces. For example, signals $x \in \mathbb{C}^m$ satisfying (2.38) with exact (not approximate) equality for sparsity level $k = 2$ are formed by a linear combination of at most two atoms. This set of signals is the union of two-dimensional subspaces

$$\mathcal{U}_{\mathbf{D}} := \{\mathbf{D}z : \|z\|_0 \leq 2\} = \bigcup_{\{i,j\} \in \Omega_2} \operatorname{span}(d_i, d_j), \tag{2.40}$$

where Ω_2 is the set of pairs of indices drawn from $\{1, \ldots, n\}$. Likewise, signals satisfying (2.39) with exact equality for sparsity level $k = 2$ form a union of two-dimensional nullspaces

$$\mathcal{U}_{\mathbf{T}} := \{x \in \mathbb{C}^m : \|\mathbf{T}x\|_0 \leq 2\} = \bigcup_{\mathcal{I} \in \Omega_{n-2}} \operatorname{null}(\mathbf{T}_{\mathcal{I}}), \tag{2.41}$$

where $\mathbf{T}_{\mathcal{I}} \in \mathbb{C}^{(n-2)\times m}$ is the matrix formed from the rows of $\mathbf{T}$ indexed by $\mathcal{I}$, and Ω_{n-2} is the set of $(n-2)$-sized index sets drawn from $\{1, \ldots, n\}$. Hence, dictionary and transform models can also be thought of as structured unions of subspaces.

2.5.2 Learning image models from heterogeneous images

Learning image models from previously acquired images poses challenges resulting from several sources of heterogeneity. A first source of heterogeneity is variability from person to person that can make it unclear how well a model learned on previously seen subjects will generalize to a new subject being imaged. For this reason, some methods do not use training images, choosing instead to fit the model jointly with image formation. A natural question is whether the two approaches can be combined: can previously acquired, potentially cleaner, images be used together with an image still being formed to learn a model that both takes advantage of historical scans while also being tailored to the new scan? Another source of heterogeneity in learning from historical scans is varying noise and artifact levels. Image quality depends on many factors from scanner configuration to even subject size, and these factors can change from image to image. As a consequence, samples provided to learning algorithms have varying quality, and an important question is how to account for this

heterogeneity in learning. Noisier, less informative, samples should roughly speaking be given less weight while fitting the model, but how much less weight is not always obvious. Chapter IV studies this question for learning subspace models with weighted PCA. In both cases, incorporating more data into model learning requires understanding and accounting for the various ways that they are heterogeneous. This is an exciting new challenge of modern data analysis and we discuss some ideas in Chapter VIII.

CHAPTER III

Asymptotic performance of PCA for high-dimensional heteroscedastic data

As described in Section 2.2, Principal Component Analysis (PCA) is a classical and ubiquitous method for reducing the dimensionality of data by projecting them onto components that captures most of their variation. Effective use of PCA in modern applications requires understanding its performance for data that are both high-dimensional and heteroscedastic. This chapter analyzes the statistical performance of PCA in this setting, i.e., for high-dimensional data drawn from a low-dimensional subspace and degraded by heteroscedastic noise. We provide simplified expressions for the asymptotic PCA recovery of the underlying subspace, subspace amplitudes and subspace coefficients; the expressions enable both easy and efficient calculation and reasoning about the performance of PCA. We exploit the structure of these expressions to show that, for a fixed average noise variance, the asymptotic recovery of PCA for heteroscedastic data is always worse than that for homoscedastic data (i.e., for noise variances that are equal across samples). Hence, while average noise variance can be a convenient measure for the overall quality of data, it gives overly optimistic estimates of the performance of PCA for heteroscedastic data.

This chapter specifically addresses the characterization of the classical and ubiquitous unweighted form of PCA that treats all samples equally and remains a natural choice in applications where estimates of the noise variances are unavailable or one hopes the noise is "close enough" to being homoscedastic. Our analysis uncovers several practical new insights for this setting; the findings both broaden our understanding of PCA and also precisely characterize the impact of heteroscedasticity. This work led to the following published conference and journal papers that this chapter presents:

[94] David Hong, Laura Balzano, and Jeffrey A. Fessler. Towards a theoretical analysis of PCA for heteroscedastic data. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, September 2016. doi: 10.1109/allerton.2016.7852272.
[95] David Hong, Laura Balzano, and Jeffrey A. Fessler. Asymptotic performance of PCA for high-dimensional heteroscedastic data. *Journal of Multivariate Analysis*, 167:435–452, September 2018. doi: 10.1016/j.jmva.2018.06.002.

3.1 Introduction

A natural setting for PCA is when data are noisy measurements of points drawn from a subspace. In this case, the first few principal components $\hat{u}_1, \ldots, \hat{u}_k$ form an estimated basis for the underlying subspace; if they recover the underlying subspace accurately then the low-dimensional scores $\hat{z}^{(1)}, \ldots, \hat{z}^{(k)}$ will largely capture the meaningful variation in the data. This chapter analyzes how well the first k principal components $\hat{u}_1, \ldots, \hat{u}_k$, PCA amplitudes $\hat{\theta}_1, \ldots, \hat{\theta}_k$ and score vectors $\hat{z}^{(1)}, \ldots, \hat{z}^{(k)}$ recover their underlying counterparts when the data are heteroscedastic, that is, when the noise in the data has non-uniform variance across samples.

3.1.1 High-dimensional, heteroscedastic data

Modern PCA applications span numerous and diverse areas, ranging from medical imaging [10, 164] to cancer data classification [179], genetics [124], and environmental sensing [160, 208], to name just a few. Increasingly, the number of variables measured is large, i.e., comparable to or even larger than the number of samples; the data are *high-dimensional*. Traditional asymptotic results that consider performance as only the number of samples grows do not apply well to such settings. New intuitions, theory and approaches are needed for the high-dimensional regime where the number of variables grows together with the number of samples [113].

When samples are obtained under varied conditions, they will likely have varied quality. In particular, some samples will have noise of larger variance than others, i.e., the data will have *heteroscedastic noise*. For example, Cochran and Horne [49] use a type of weighted PCA because their spectrophotometric data is obtained from averages taken over increasing windows of time; samples from longer windows have lower noise variance. Similarly, astronomical measurements of stars [194] have heterogeneous amounts of noise among samples due to differing atmospheric effects from one sample to the next. Finally, modern big data inference is increasingly done using

datasets built up from myriad sources, and hence one can expect heteroscedasticity will be the norm.

3.1.2 Contributions of this chapter

This chapter provides simplified expressions for the performance of PCA from heteroscedastic data in the limit as both the number of samples and dimension tend to infinity. The expressions quantify the asymptotic recovery of an underlying subspace, subspace amplitudes and coefficients by the principal components, PCA amplitudes and scores, respectively. The asymptotic recoveries are functions of the samples per ambient dimension, the underlying subspace amplitudes and the distribution of noise variances. Forming the expressions involves first connecting several results from random matrix theory [14, 22] to obtain initial expressions for asymptotic recovery that are difficult to evaluate and analyze, and then exploiting a nontrivial structure in the expressions to obtain much simpler algebraic descriptions. These descriptions enable both easy and efficient calculation and reasoning about the asymptotic performance of PCA. Identifying and exploiting the nontrivial structure here is the key technical innovation that helps us find and study the simplified expressions in this chapter.

The impact of heteroscedastic noise, in particular, is not immediately obvious given results of prior literature. How much do a few noisy samples degrade the performance of PCA? Is heteroscedasticity ever beneficial for PCA? Our simplified expressions enable such questions to be answered. In particular, we use these expressions to show that, for a fixed average noise variance, the asymptotic subspace recovery, amplitude recovery and coefficient recovery are all worse for heteroscedastic data than for homoscedastic data (i.e., for noise variances that are equal across samples), confirming a conjecture in [94]. Hence, while average noise variance is often a practically convenient measure for the overall quality of data, it gives an overly optimistic estimate of PCA performance. This analysis provides a deeper understanding of how PCA performs in the presence of heteroscedastic noise.

3.1.3 Relationship to previous works

Homoscedastic noise has been well-studied, and there are many nice results characterizing PCA in this setting. Benaych-Georges and Nadakuditi [22] give an expression for asymptotic subspace recovery, also found in [111, 147, 163], in the limit as both the number of samples and ambient dimension tend to infinity. As argued in [111], the expression in [22] reveals that asymptotic subspace recovery is perfect only when the number of samples per ambient dimension tends to infinity, so PCA is not (asymptotically) consistent for high-dimensional data. Various alter-

natives [28, 65, 111] can regain consistency by exploiting sparsity in the covariance matrix or in the principal components. As discussed in [22, 147], the expression in [22] also exhibits a phase transition: the number of samples per ambient dimension must be sufficiently high to obtain non-zero subspace recovery (i.e., for any subspace recovery to occur). This chapter generalizes the expression in [22] to heteroscedastic noise; homoscedastic noise is a special case and is discussed in Section 3.2.3. Once again, (asymptotic) consistency is obtained when the number of samples per ambient dimension tends to infinity, and there is a phase transition between zero recovery and non-zero recovery.

PCA is known to generally perform well in the presence of low to moderate homoscedastic noise and in the presence of missing data [43]. When the noise is standard normal, PCA gives the maximum likelihood (ML) estimate of the subspace [195]. In general, [195] proposes finding the ML estimate via expectation maximization. Conventional PCA is not an ML estimate of the subspace for heteroscedastic data, but it remains a natural choice in applications where we might expect noise to be heteroscedastic but hope it is "close enough" to being homoscedastic. Even for mostly homoscedastic data, however, PCA performs poorly when the heteroscedasticity is due to gross errors (i.e., outliers) [56, 104, 114], which has motivated the development and analysis of robust variants; see [38, 42, 54, 87, 88, 127, 168, 214, 220] and their corresponding bibliographies. This chapter provides expressions for asymptotic recovery that enable rigorous understanding of the impact of heteroscedasticity.

The generalized spiked covariance model, proposed and analyzed in [15] and [215], generalizes homoscedastic noise in an alternate way. It extends the Johnstone spiked covariance model [110, 111] (a particular homoscedastic setting) by using a population covariance that allows, among other things, non-uniform noise variances *within* each sample. Non-uniform noise variances within each sample may arise, for example, in applications where sample vectors are formed by concatenating the measurements of intrinsically different quantities. This chapter considers data with non-uniform noise variances *across* samples instead; we model noise variances *within* each sample as uniform. Data with non-uniform noise variances across samples arise, for example, in applications where samples come from heterogeneous sources, some of which are better quality (i.e., lower noise) than others. See Section 3.8.1 for a more detailed discussion of connections to spiked covariance models.

Our previous work [94] analyzed the subspace recovery of PCA for heteroscedastic noise but was limited to real-valued data coming from a random one-dimensional subspace where the number of samples exceeded the data dimension. This chapter extends that analysis to the more general setting of real- or complex-valued data coming from a deterministic low-dimensional subspace where the number of samples

no longer needs to exceed the data dimension. This chapter also extends the analysis of [94] to include the recovery of the underlying subspace amplitudes and coefficients. In both works, we use the main results of [22] to obtain initial expressions relating asymptotic recovery to the limiting noise singular value distribution.

The main results of [147] provide non-asymptotic results (i.e., probabilistic approximation results for finite samples in finite dimension) for homoscedastic noise limited to the special case of one-dimensional subspaces. Signal-dependent noise was recently considered in [203], where they analyze the performance of PCA and propose a new generalization of PCA that performs better in certain regimes. Another line of work [58] extends [22] to linearly reduced data. Chapter IV extends [22] to weighted data to analyze a weighted variant of PCA. Such analyses are beyond the scope of this chapter, but are interesting avenues for further study.

3.1.4 Organization of the chapter

Section 3.2 describes the model we consider and states the main results: simplified expressions for asymptotic PCA recovery and the fact that PCA performance is best (for a fixed average noise variance) when the noise is homoscedastic. Section 3.3 uses the main results to provide a qualitative analysis of how the model parameters (e.g., samples per ambient dimension and the distribution of noise variances) affect PCA performance under heteroscedastic noise. Section 3.4 compares the asymptotic recovery with non-asymptotic (i.e., finite) numerical simulations. The simulations demonstrate good agreement as the ambient dimension and number of samples grow large; when these values are small the asymptotic recovery and simulation differ but have the same general behavior. Sections 3.5 and 3.6 prove the main results. Finally, Section 3.7 discusses the findings and describes avenues for future work, and Section 3.8 provides some supplementary discussions to this chapter.

3.2 Main results

3.2.1 Model for heteroscedastic data

We model n heteroscedastic sample vectors $y_1, \dots, y_n \in \mathbb{C}^d$ from a k-dimensional subspace as

$$y_i = \mathbf{U}\mathbf{\Theta} z_i + \eta_i \varepsilon_i = \sum_{j=1}^{k} \theta_j u_j (z_i^{(j)})^* + \eta_i \varepsilon_i. \tag{3.1}$$

The following are deterministic:

- $\mathbf{U} = (u_1, \dots, u_k) \in \mathbb{C}^{d \times k}$ forms an orthonormal basis for the subspace,

- $\mathbf{\Theta} = \operatorname{diag}(\theta_1, \ldots, \theta_k) \in \mathbb{R}_+^{k \times k}$ is a diagonal matrix of amplitudes,
- $\eta_i \in \{\sigma_1, \ldots, \sigma_L\}$ are each one of L noise standard deviations $\sigma_1, \ldots, \sigma_L$,

and we define n_1 to be the number of samples with $\eta_i = \sigma_1$, n_2 to be the number of samples with $\eta_i = \sigma_2$ and so on, where $n_1 + \cdots + n_L = n$.

The following are random and independent:

- $z_i \in \mathbb{C}^k$ are iid sample coefficient vectors that have iid entries with mean $\mathbb{E}(z_{ij}) = 0$, variance $\mathbb{E}|z_{ij}|^2 = 1$, and a distribution satisfying the log-Sobolev inequality [9],
- $\varepsilon_i \in \mathbb{C}^d$ are unitarily invariant iid noise vectors that have iid entries with mean $\mathbb{E}(\varepsilon_{ij}) = 0$, variance $\mathbb{E}|\varepsilon_{ij}|^2 = 1$ and bounded fourth moment $\mathbb{E}|\varepsilon_{ij}|^4 < \infty$,

and we define the k (component) coefficient vectors $z^{(1)}, \ldots, z^{(k)} \in \mathbb{C}^n$ such that the ith entry of $z^{(j)}$ is $z_i^{(j)} = (z_{ij})^*$, the complex conjugate of the jth entry of z_i. Defining the coefficient vectors in this way is convenient for stating and proving the results that follow, as they more naturally correspond to right singular vectors of the data matrix formed by concatenating $y_1, \ldots, y_n$ as columns.

The model extends the Johnstone spiked covariance model [110, 111] by incorporating heteroscedasticity (see Section 3.8.1 for a detailed discussion). We also allow complex-valued data, as it is of interest in important signal processing applications such as medical imaging; for example, data obtained in magnetic resonance imaging are complex-valued.

Remark 3.1. By unitarily invariant, we mean that left multiplication of ε_i by any unitary matrix does not change the joint distribution of its entries. As in our previous work [94], this assumption can be dropped if instead the subspace $\mathbf{U}$ is randomly drawn according to either the "orthonormalized model" or "iid model" of [22]. Under these models, the subspace $\mathbf{U}$ is randomly chosen in an isotropic manner.

Remark 3.2. The above conditions are satisfied, e.g., when the entries z_{ij} and ε_{ij} are circularly symmetric complex normal $\mathcal{CN}(0, 1)$ random variables. Rademacher random variables (i.e., ± 1 with equal probability) are another choice for coefficient entries z_{ij}; see Section 2.3.2 of [9] for discussion of the log-Sobolev inequality. Only circularly symmetric complex normal distributions satisfy all conditions for noise entries ε_{ij},[1] but as noted in Remark 3.1, unitary invariance can be dropped if we assume the subspace is randomly drawn as in [22].

[1] Gaussianity follows from orthogonal invariance via the Herschel-Maxwell theorem [33, Theorem 0.0.1] for real-valued random vectors. Its extension to complex-valued random vectors can be shown by observing that unitary invariance implies orthogonal invariance of its real part and circular symmetry of each entry in the complex plane.

Remark 3.3. The assumption that noise entries ε_{ij} are identically distributed with bounded fourth moment can be relaxed when they are real-valued as long as an aggregate of their tails still decays sufficiently quickly, i.e., as long as they satisfy Condition 1.3 from [157]. In this setting, the results of [157] replace those of [14] in the proof.

3.2.2 Simplified expressions for asymptotic recovery

The following theorem describes how well the PCA estimates $\hat{u}_1, \dots, \hat{u}_k$, $\hat{\theta}_1, \dots, \hat{\theta}_k$ and $\hat{z}^{(1)}, \dots, \hat{z}^{(k)}$ recover the underlying subspace basis $u_1, \dots, u_k$, subspace amplitudes $\theta_1, \dots, \theta_k$ and coefficient vectors $z^{(1)}, \dots, z^{(k)}$, as a function of the sample-to-dimension ratio $n/d \to c > 0$, the subspace amplitudes $\theta_1, \dots, \theta_k$, the noise variances $\sigma_1^2, \dots, \sigma_L^2$ and corresponding proportions $n_\ell/n \to p_\ell$ for each $\ell \in \{1, \dots, L\}$. One may generally expect performance to improve with increasing sample-to-dimension ratio and subspace amplitudes; Theorem 3.4 provides the precise dependence on these parameters as well as on the noise variances and their proportions.

Theorem 3.4 (Recovery of individual components)**.** *Suppose that the sample-to-dimension ratio $n/d \to c > 0$ and the noise variance proportions $n_\ell/n \to p_\ell$ for $\ell \in \{1, \dots, L\}$ as $n, d \to \infty$. Then the ith PCA amplitude $\hat{\theta}_i$ is such that*

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \frac{1}{c} \max(\alpha, \beta_i) \left\{ 1 + c \sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{\max(\alpha, \beta_i) - \sigma_\ell^2} \right\}, \tag{3.2}$$

where α and β_i are, respectively, the largest real roots of

$$A(x) = 1 - c \sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^4}{(x - \sigma_\ell^2)^2}, \quad B_i(x) = 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{x - \sigma_\ell^2}. \tag{3.3}$$

Furthermore, if $A(\beta_i) > 0$, then the ith principal component $\hat{u}_i$ is such that

$$|\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\} \rangle|^2 \xrightarrow{\text{a.s.}} \frac{A(\beta_i)}{\beta_i B_i'(\beta_i)}, \quad |\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j \neq \theta_i\} \rangle|^2 \xrightarrow{\text{a.s.}} 0, \tag{3.4}$$

the normalized score vector $\hat{z}^{(i)}/\sqrt{n}$ is such that

$$\begin{aligned} \left| \left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j = \theta_i\} \right\rangle \right|^2 &\xrightarrow{\text{a.s.}} \frac{A(\beta_i)}{c\{\beta_i + (1-c)\theta_i^2\} B_i'(\beta_i)}, \\ \left| \left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j \neq \theta_i\} \right\rangle \right|^2 &\xrightarrow{\text{a.s.}} 0, \end{aligned} \tag{3.5}$$

and

$$\sum_{j : \theta_j = \theta_i} \langle \hat{u}_i, u_j \rangle \left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \frac{z^{(j)}}{\|z^{(j)}\|} \right\rangle^* \xrightarrow{\text{a.s.}} \frac{A(\beta_i)}{\sqrt{c\beta_i\{\beta_i + (1-c)\theta_i^2\}} B_i'(\beta_i)}. \tag{3.6}$$

Section 3.5 presents the proof of Theorem 3.4. The expressions can be easily and efficiently computed. The hardest part is finding the largest roots of the univariate rational functions $A(x)$ and $B_i(x)$, but off-the-shelf solvers can do this efficiently. See [94] for an example of similar calculations.

The projection $|\langle \hat{u}_i, \text{span}\{u_j : \theta_j = \theta_i\}\rangle|^2$ in Theorem 3.4 is the square cosine principal angle between the ith principal component $\hat{u}_i$ and the span of the basis elements with subspace amplitudes equal to θ_i. When the subspace amplitudes are distinct, $|\langle \hat{u}_i, \text{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 = |\langle \hat{u}_i, u_i\rangle|^2$ is the square cosine angle between $\hat{u}_i$ and u_i. This value is related by a constant to the squared error between the two (unit norm) vectors and is one among several natural performance metrics for subspace estimation. Similar observations hold for $|\langle \hat{z}^{(i)}/\sqrt{n}, \text{span}\{z^{(j)} : \theta_j = \theta_i\}\rangle|^2$. Note that $\hat{z}^{(i)}/\sqrt{n}$ has unit norm.

The expressions (3.4), (3.5) and (3.6) apply only if $A(\beta_i) > 0$. The following conjecture predicts a phase transition at $A(\beta_i) = 0$ so that asymptotic recovery is zero for $A(\beta_i) \leq 0$.

Conjecture 3.5 (Phase transition)**.** *Suppose (as in Theorem 3.4) that the sample-to-dimension ratio $n/d \to c > 0$ and the noise variance proportions $n_\ell/n \to p_\ell$ for $\ell \in \{1, \ldots, L\}$ as $n, d \to \infty$. If $A(\beta_i) \leq 0$, then the ith principal component $\hat{u}_i$ and the normalized score vector $\hat{z}^{(i)}/\sqrt{n}$ are such that*

$$|\langle \hat{u}_i, \text{span}\{u_1, \ldots, u_k\}\rangle|^2 \xrightarrow{\text{a.s.}} 0, \quad \left|\left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \text{span}\{z^{(1)}, \ldots, z^{(k)}\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} 0.$$

This conjecture is true for a data model having Gaussian coefficients and homoscedastic Gaussian noise as shown in [163]. It is also true for a one-dimensional subspace (i.e., $k = 1$) as we showed in [94]. Proving it in general would involve showing that the singular values of the matrix whose columns are the noise vectors exhibit repulsion behavior; see Remark 2.13 of [22].

3.2.3 Homoscedastic noise as a special case

For homoscedastic noise with variance σ^2, $A(x) = 1 - c\sigma^4/(x - \sigma^2)^2$ and $B_i(x) = 1 - c\theta_i^2/(x - \sigma^2)$. The largest real roots of these functions are, respectively, $\alpha = (1 + \sqrt{c})\sigma^2$ and $\beta_i = \sigma^2 + c\theta_i^2$. Thus the asymptotic PCA amplitude (3.2) becomes

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \begin{cases} \theta_i^2\{1 + \sigma^2/(c\theta_i^2)\}(1 + \sigma^2/\theta_i^2) & \text{if } c\theta_i^4 > \sigma^4, \\ \sigma^2(1 + 1/\sqrt{c})^2 & \text{otherwise.} \end{cases} \tag{3.7}$$

Further, if $c\theta_i^4 > \sigma^4$, then the non-zero portions of asymptotic subspace recovery (3.4) and coefficient recovery (3.5) simplify to

$$|\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} \frac{c - \sigma^4/\theta_i^4}{c + \sigma^2/\theta_i^2}, \tag{3.8}$$

$$\left|\left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j = \theta_i\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} \frac{c - \sigma^4/\theta_i^4}{c(1 + \sigma^2/\theta_i^2)},$$

These limits agree with the homoscedastic results in [22, 29, 111, 147, 163]. As noted in Section 3.2.2, Conjecture 3.5 is known to be true when the coefficients are Gaussian and the noise is both homoscedastic and Gaussian, in which case (3.8) becomes

$$|\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} \max\left(0, \frac{c - \sigma^4/\theta_i^4}{c + \sigma^2/\theta_i^2}\right),$$

$$\left|\left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j = \theta_i\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} \max\left\{0, \frac{c - \sigma^4/\theta_i^4}{c(1 + \sigma^2/\theta_i^2)}\right\}.$$

See Section 2 of [111] and Section 2.3 of [163] for a discussion of this result.

3.2.4 Bias of the PCA amplitudes

The simplified expression in (3.2) enables us to immediately make two observations about the recovery of the subspace amplitudes $\theta_1, \ldots, \theta_k$ by the PCA amplitudes $\hat{\theta}_1, \ldots, \hat{\theta}_k$.

Remark 3.6 (Positive bias in PCA amplitudes). The largest real root β_i of $B_i(x)$ is greater than $\max_\ell(\sigma_\ell^2)$. Thus $1/(\beta_i - \sigma_\ell^2) > 1/\beta_i$ for $\ell \in \{1, \ldots, L\}$ and so evaluating (3.3) at β_i yields

$$0 = B_i(\beta_i) = 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{\beta_i - \sigma_\ell^2} < 1 - c\theta_i^2 \frac{1}{\beta_i}.$$

As a result, $\beta_i > c\theta_i^2$, so the asymptotic PCA amplitude (3.2) exceeds the subspace amplitude, i.e., $\hat{\theta}_i$ is positively biased and is thus an inconsistent estimate of θ_i. This is a general phenomenon for noisy data and motivates asymptotically optimal shrinkage in [145].

Remark 3.7 (Alternate formula for amplitude bias). If $A(\beta_i) \geq 0$, then $\beta_i \geq \alpha$ because $A(x)$ and $B_i(x)$ are both increasing functions for $x > \max_\ell(\sigma_\ell^2)$. Thus, the

asymptotic amplitude bias is

$$\begin{aligned}\frac{\hat{\theta}_i^2}{\theta_i^2} &\xrightarrow{\text{a.s.}} \frac{\beta_i}{c\theta_i^2}\left(1 + c\sum_{\ell=1}^{L}\frac{p_\ell\sigma_\ell^2}{\beta_i - \sigma_\ell^2}\right) = \frac{\beta_i}{c\theta_i^2}\left\{1 + c\sum_{\ell=1}^{L} p_\ell\left(-1 + \frac{\beta_i}{\beta_i - \sigma_\ell^2}\right)\right\} \\ &= \frac{\beta_i}{c\theta_i^2}\left(1 + \beta_i c\sum_{\ell=1}^{L}\frac{p_\ell}{\beta_i - \sigma_\ell^2} - c\right) = \frac{\beta_i}{c\theta_i^2}\left[1 + \frac{\beta_i}{\theta_i^2}\{1 - B_i(\beta_i)\} - c\right] \\ &= \frac{\beta_i}{c\theta_i^2}\left(1 + \frac{\beta_i}{\theta_i^2} - c\right) = 1 + \left(\frac{\beta_i}{c\theta_i^2} - 1\right)\left(\frac{\beta_i}{\theta_i^2} + 1\right), \qquad (3.9)\end{aligned}$$

where we have applied (3.2), divided the summand with respect to σ_ℓ^2, used the facts that $p_1 + \cdots + p_L = 1$ and $B_i(\beta_i) = 0$, and finally factored. The expression (3.9) shows that the positive bias is an increasing function of β_i when $A(\beta_i) \geq 0$.

3.2.5 Overall subspace and signal recovery

Overall subspace recovery is more useful than individual component recovery when subspace amplitudes are equal and so individual basis elements are not identifiable. It is also more relevant when we are most interested in recovering or denoising low-dimensional signals in a subspace. Overall recovery of the low-dimensional signal, quantified here by mean square error, is useful for understanding how well PCA "denoises" the data taken as a whole.

Corollary 3.8 (Overall recovery)**.** *Suppose (as in Theorem 3.4) that the sample-to-dimension ratio $n/d \to c > 0$ and the noise variance proportions $n_\ell/n \to p_\ell$ for $\ell \in \{1, \ldots, L\}$ as $n, d \to \infty$. If $A(\beta_1), \ldots, A(\beta_k) > 0$, then the subspace estimate $\hat{\mathbf{U}} = (\hat{u}_1, \ldots, \hat{u}_k) \in \mathbb{C}^{d\times k}$ from PCA is such that*

$$\frac{1}{k}\left\|\hat{\mathbf{U}}^{\mathsf{H}}\mathbf{U}\right\|_F^2 \xrightarrow{\text{a.s.}} \frac{1}{k}\sum_{i=1}^{k}\frac{A(\beta_i)}{\beta_i B_i'(\beta_i)}, \qquad (3.10)$$

and the mean square error is

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{U}\mathbf{\Theta}z_i - \hat{\mathbf{U}}\hat{\mathbf{\Theta}}\hat{z}_i\right\|_2^2 \xrightarrow{\text{a.s.}} \sum_{i=1}^{k} 2\left\{\theta_i^2 - \frac{A(\beta_i)}{cB'(\beta_i)}\right\} + \left(\frac{\beta_i}{c\theta_i^2} - 1\right)(\beta_i + \theta_i^2), \qquad (3.11)$$

where $A(x)$, $B_i(x)$ and β_i are as in Theorem 3.4, and $\hat{z}_i$ is the vector of score entries for the ith sample.

Proof of Corollary 3.8. The subspace recovery can be decomposed as

$$\frac{1}{k}\left\|\hat{\mathbf{U}}^{\mathsf{H}}\mathbf{U}\right\|_F^2 = \frac{1}{k}\sum_{i=1}^{k}\left\|\hat{u}_i^{\mathsf{H}}\mathbf{U}_{j:\theta_j=\theta_i}\right\|_2^2 + \left\|\hat{u}_i^{\mathsf{H}}\mathbf{U}_{j:\theta_j\neq\theta_i}\right\|_2^2,$$

where the columns of $\mathbf{U}_{j:\theta_j=\theta_i}$ are the basis elements u_j with subspace amplitude θ_j equal to θ_i, and the remaining basis elements are the columns of $\mathbf{U}_{j:\theta_j\neq\theta_i}$. Asymptotic overall subspace recovery (3.10) follows by noting that these terms are exactly the square cosine principal angles in (3.4) of Theorem 3.4.

The mean square error can also be decomposed as

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{U}\boldsymbol{\Theta}z_i-\hat{\mathbf{U}}\hat{\boldsymbol{\Theta}}\hat{z}_i\right\|_2^2 &= \left\|\mathbf{U}\boldsymbol{\Theta}\left(\frac{1}{\sqrt{n}}\mathbf{Z}\right)^{\mathsf{H}}-\hat{\mathbf{U}}\hat{\boldsymbol{\Theta}}\left(\frac{1}{\sqrt{n}}\hat{\mathbf{Z}}\right)^{\mathsf{H}}\right\|_F^2 \\ &= \sum_{i=1}^{k}\theta_i^2\left[\left\|\frac{z^{(i)}}{\sqrt{n}}\right\|_2^2+\frac{\hat{\theta}_i^2}{\theta_i^2}-2\Re\left\{\frac{\hat{\theta}_i}{\theta_i}\sum_{j=1}^{k}\frac{\theta_j}{\theta_i}\langle\hat{u}_i,u_j\rangle\left\langle\frac{\hat{z}^{(i)}}{\sqrt{n}},\frac{z^{(j)}}{\sqrt{n}}\right\rangle^*\right\}\right], \qquad (3.12)\end{aligned}$$

where $\mathbf{Z}=(z^{(1)},\ldots,z^{(k)})\in\mathbb{C}^{n\times k}$, $\hat{\mathbf{Z}}=(\hat{z}^{(1)},\ldots,\hat{z}^{(k)})\in\mathbb{C}^{n\times k}$ and $\Re$ denotes the real part of its argument. The first term of (3.12) has almost sure limit 1 by the law of large numbers. The almost sure limit of the second term is obtained from (3.9). We can disregard the summands in the inner sum for which $\theta_j\neq\theta_i$; by (3.4) and (3.5) these terms have an almost sure limit of zero (the inner products both vanish). The rest of the inner sum

$$\sum_{j:\theta_j=\theta_i}\frac{\theta_j}{\theta_i}\langle\hat{u}_i,u_j\rangle\left\langle\frac{\hat{z}^{(i)}}{\sqrt{n}},\frac{z^{(j)}}{\sqrt{n}}\right\rangle^*=\sum_{j:\theta_j=\theta_i}(1)\langle\hat{u}_i,u_j\rangle\left\langle\frac{\hat{z}^{(i)}}{\sqrt{n}},\frac{z^{(j)}}{\sqrt{n}}\right\rangle^*$$

has the same almost sure limit as in (3.6) because $\|z^{(i)}/\sqrt{n}\|^2\to 1$ as $n\to\infty$. Combining these almost sure limits and simplifying yields (3.11). □

3.2.6 Importance of homoscedasticity

How important is homoscedasticity for PCA? Does having some low noise data outweigh the cost of introducing heteroscedasticity? Consider the following three settings:

1.- All samples have noise variance 1 (i.e., data are homoscedastic).

2.- 99% of samples have noise variance 1.01 but 1% have noise variance 0.01.

3.- 99% of samples have noise variance 0.01 but 1% have noise variance 99.01.

In all three settings, the average noise variance is 1. We might expect PCA to perform well in Setting 1 because it has the smallest maximum noise variance. However, Setting 2 may seem favorable because we obtain samples with very small noise, and suffer only a slight increase in noise for the rest. Setting 3 may seem favorable because most of the samples have very small noise. However, we might also expect

PCA to perform poorly because 1% of samples have very large noise and will likely produce gross errors (i.e., outliers). Between all three, it is not initially obvious what setting PCA will perform best in. The following theorem shows that PCA performs best when the noise is homoscedastic, as in Setting 1.

Theorem 3.9. *Homoscedastic noise yields the best asymptotic PCA amplitude* (3.2)*, subspace recovery* (3.4) *and coefficient recovery* (3.5) *in Theorem 3.4 for a given average noise variance* $\bar{\sigma}^2 = p_1\sigma_1^2 + \cdots + p_L\sigma_L^2$ *over all distributions of noise variances for which* $A(\beta_i) > 0$*. Namely, homoscedastic noise minimizes* (3.2) *(and hence the positive bias) and it maximizes* (3.4) *and* (3.5)*.*

Concretely, suppose we had $c = 10$ samples per dimension and a subspace amplitude of $\theta_i = 1$. Then the asymptotic subspace recoveries (3.4) given in Theorem 3.4 evaluate to 0.818 in Setting 1, 0.817 in Setting 2 and 0 in Setting 3; asymptotic recovery is best in Setting 1 as predicted by Theorem 3.9. Recovery is entirely lost in Setting 3, consistent with the observation that PCA is not robust to gross errors. In Setting 2, only using the 1% of samples with noise variance 0.01 (resulting in 0.1 samples per dimension) yields an asymptotic subspace recovery of 0.908 and so we may hope that recovery with all data could be better. Theorem 3.9 rigorously shows that PCA does not fully exploit these high quality samples and instead performs worse in Setting 2 than in Setting 1, if only slightly.

Section 3.6 presents the proof of Theorem 3.9. It is notable that Theorem 3.9 holds for all proportions p, sample-to-dimension ratios c and subspace amplitudes θ_i; there are no settings where PCA benefits from heteroscedastic noise over homoscedastic noise with the same average variance. The following corollary is equivalent and provides an alternate way of viewing the result.

Corollary 3.10 (Bounds on asymptotic recovery)**.** *If* $A(\beta_i) \geq 0$ *then the asymptotic PCA amplitude* (3.2) *is bounded as*

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \theta_i^2 + \theta_i^2\left(\frac{\beta_i}{c\theta_i^2} - 1\right)\left(\frac{\beta_i}{\theta_i^2} + 1\right) \geq \theta_i^2\left(1 + \frac{\bar{\sigma}^2}{c\theta_i^2}\right)\left(1 + \frac{\bar{\sigma}^2}{\theta_i^2}\right), \tag{3.13}$$

the asymptotic subspace recovery (3.4) *is bounded as*

$$|\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} \frac{A(\beta_i)}{\beta_i B_i'(\beta_i)} \leq \frac{c - \bar{\sigma}^4/\theta_i^4}{c + \bar{\sigma}^2/\theta_i^2}, \tag{3.14}$$

and the asymptotic coefficient recovery (3.5) *is bounded as*

$$\left|\left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j = \theta_i\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} \frac{A(\beta_i)}{c\{\beta_i + (1-c)\theta_i^2\}B_i'(\beta_i)} \leq \frac{c - \bar{\sigma}^4/\theta_i^4}{c(1 + \bar{\sigma}^2/\theta_i^2)}, \tag{3.15}$$

where $\bar{\sigma}^2 = p_1\sigma_1^2 + \cdots + p_L\sigma_L^2$ *is the average noise variance and the bounds are met with equality if and only if* $\sigma_1^2 = \cdots = \sigma_L^2$*.*

Proof of Corollary 3.10. The bounds (3.13), (3.14) and (3.15) follow immediately from Theorem 3.9 and the expressions for homoscedastic noise (3.7) and (3.8) in Section 3.2.3. □

Corollary 3.10 highlights that while average noise variance may be a practically convenient measure for the overall quality of data, it can lead to an overly optimistic estimate of the performance of PCA for heteroscedastic data. The expressions (3.2), (3.4) and (3.5) in Theorem 3.4 are more accurate.

Remark 3.11 (Average inverse noise variance). Average inverse noise variance $\mathcal{I} = p_1 \times 1/\sigma_1^2 + \cdots + p_L \times 1/\sigma_L^2$ is another natural measure for the overall quality of data. In particular, it is the (scaled) Fisher information for heteroscedastic Gaussian measurements of a fixed scalar. Theorem 3.9 implies that homoscedastic noise also produces the best asymptotic PCA performance for a given average inverse noise variance; note that homoscedastic noise minimizes the average noise variance in this case. Thus, average inverse noise variance can also lead to an overly optimistic estimate of the performance of PCA for heteroscedastic data.

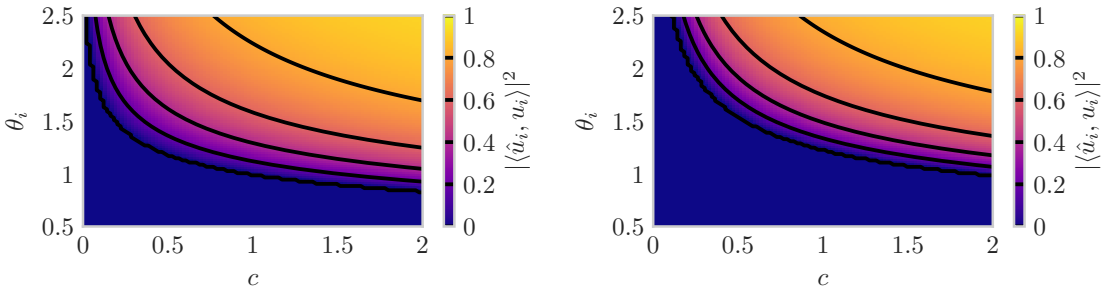
3.3 Impact of parameters

The simplified expressions in Theorem 3.4 for the asymptotic performance of PCA provide insight into the impact of the model parameters: sample-to-dimension ratio c, subspace amplitudes $\theta_1, \ldots, \theta_k$, proportions $p_1, \ldots, p_L$ and noise variances $\sigma_1^2, \ldots, \sigma_L^2$. For brevity, we focus on the asymptotic subspace recovery (3.4) of the ith component; similar phenomena occur for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5) as we show in Section 3.8.3.

3.3.1 Impact of sample-to-dimension ratio c and subspace amplitude θ_i

Suppose first that there is only one noise variance fixed at $\sigma_1^2 = 1$, i.e., $L = 1$, while we vary the sample-to-dimension ratio c and subspace amplitude θ_i. This is the homoscedastic setting described in Section 3.2.3. Figure 3.1a illustrates the expected behavior: decreasing the subspace amplitude θ_i degrades asymptotic subspace recovery (3.4) but the lost performance could be regained by increasing the number of samples. Figure 3.1a also illustrates a phase transition: a sufficient number of samples with a sufficiently large subspace amplitude is necessary to have an asymptotic recovery greater than zero. Note that in all such figures, we label the axis $|\langle \hat{u}_i, u_i \rangle|^2$ to indicate the asymptotic recovery on the right hand side of (3.4).

Now suppose that there are two noise variances $\sigma_1^2 = 0.8$ and $\sigma_2^2 = 1.8$ occurring in proportions $p_1 = 80\%$ and $p_2 = 20\%$. The average noise variance is still 1, and



(a) Homoscedastic noise with $\sigma_1^2 = 1$.

(b) Heteroscedastic noise with $p_1 = 80\%$ of samples at $\sigma_1^2 = 0.8$ and $p_2 = 20\%$ of samples at $\sigma_2^2 = 1.8$.

Figure 3.1: Asymptotic subspace recovery (3.4) of the ith component as a function of sample-to-dimension ratio c and subspace amplitude θ_i with average noise variance equal to one. Contours are overlaid in black and the region where $A(\beta_i) \leq 0$ is shown as zero (the prediction of Conjecture 3.5). The phase transition in (b) is further right than in (a); more samples are needed to recover the same strength signal.

Figure 3.1b illustrates similar overall features to the homoscedastic case. Decreasing subspace amplitude θ_i once again degrades asymptotic subspace recovery (3.4) and the lost performance could be regained by increasing the number of samples. However, the phase transition is further up and to the right compared to the homoscedastic case. This is consistent with Theorem 3.9; PCA performs worse on heteroscedastic data than it does on homoscedastic data of the same average noise variance, and thus more samples or a larger subspace amplitude are needed to recover the subspace basis element.

3.3.2 Impact of proportions $p_1, \ldots, p_L$

Suppose that there are two noise variances $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 3.25$ occurring in proportions $p_1 = 1 - p_2$ and p_2, where the sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. Figure 3.2 shows the asymptotic subspace recovery (3.4) as a function of the proportion p_2. Since σ_2^2 is significantly larger, it is natural to think of p_2 as a fraction of contaminated samples. As expected, performance generally degrades as p_2 increases and low noise samples with noise variance σ_1^2 are traded for high noise samples with noise variance σ_2^2. The performance is best when $p_2 = 0$ and all the samples have the smaller noise variance σ_1^2 (i.e., there is no contamination).

It is interesting that the asymptotic subspace recovery in Figure 3.2 has a steeper slope initially for p_2 close to zero and then a shallower slope for p_2 close to one. Thus

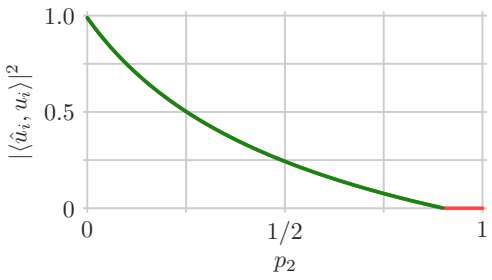


Figure 3.2: Asymptotic subspace recovery (3.4) of the ith component as a function of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. The region where $A(\beta_i) \leq 0$ is the red horizontal segment with value zero (the prediction of Conjecture 3.5).

the benefit of reducing the contamination fraction varies across the range.

3.3.3 Impact of noise variances $\sigma_1^2, \ldots, \sigma_L^2$

Suppose that there are two noise variances σ_1^2 and σ_2^2 occurring in proportions $p_1 = 70\%$ and $p_2 = 30\%$, where the sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. Figure 3.3 shows the asymptotic subspace recovery (3.4) as a function of the noise variances σ_1^2 and σ_2^2. As expected, performance typically degrades with increasing noise variances. However, there is a curious regime around $\sigma_1^2 = 0$ and $\sigma_2^2 = 4$ where increasing σ_1^2 slightly from zero improves asymptotic performance; the contour lines point slightly up and to the right. We have also observed this phenomenon in finite-dimensional simulations (see Fig. 8.1), so this effect is not simply an asymptotic artifact. This surprising phenomenon is an interesting avenue for future exploration.

The contours in Figure 3.3 are generally horizontal for small σ_1^2 and vertical for small σ_2^2. This indicates that when the gap between the two largest noise variances is "sufficiently" wide, the asymptotic subspace recovery (3.4) is roughly determined by the largest noise variance. While initially unexpected, this property can be intuitively understood by recalling that β_i is the largest value of x satisfying

$$\frac{1}{c\theta_i^2} = \sum_{\ell=1}^{L} \frac{p_\ell}{x - \sigma_\ell^2}. \tag{3.16}$$

When the gap between the two largest noise variances is wide, the largest noise variance is significantly larger than the rest and it dominates the sum in (3.16) for

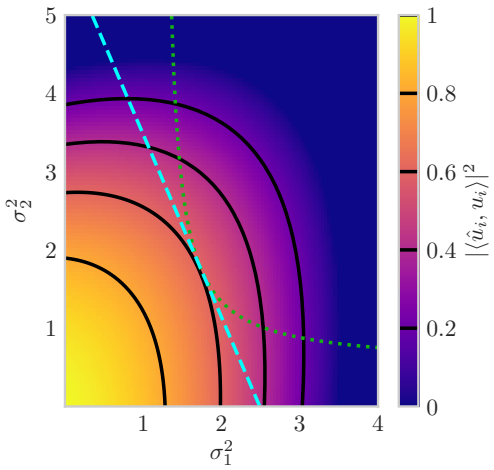


Figure 3.3: Asymptotic subspace recovery (3.4) of the ith component as a function of noise variances σ_1^2 and σ_2^2 occurring in proportions $p_1 = 70\%$ and $p_2 = 30\%$, where the sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. Contours are overlaid in black and the region where $A(\beta_i) \leq 0$ is shown as zero (the prediction of Conjecture 3.5). Along the dashed cyan line, the average noise variance is $\bar{\sigma}^2 \approx 1.74$ and the best performance occurs when $\sigma_1^2 = \sigma_2^2 = \bar{\sigma}^2$. Along the dotted green curve, the average inverse noise variance is $\mathcal{I} \approx 0.575$ and the best performance again occurs when $\sigma_1^2 = \sigma_2^2$.

$x > \max_\ell(\sigma_\ell^2)$, i.e., where β_i occurs. Thus β_i, and similarly, $A(\beta_i)$ and $B_i'(\beta_i)$ are roughly determined by the largest noise variance.

The precise relative impact of each noise variance σ_ℓ^2 depends on its corresponding proportion p_ℓ, as shown by the asymmetry of Figure 3.3 around the line $\sigma_1^2 = \sigma_2^2$. Nevertheless, very large noise variances can drown out the impact of small noise variances, regardless of their relative proportions. This behavior provides a rough explanation for the sensitivity of PCA to even a few gross errors (i.e., outliers); even in small proportions, sufficiently large errors dominate the performance of PCA.

Along the dashed cyan line in Figure 3.3, the average noise variance is $\bar{\sigma}^2 \approx 1.74$ and the best performance occurs when $\sigma_1^2 = \sigma_2^2 = \bar{\sigma}^2$, as predicted by Theorem 3.9. Along the dotted green curve, the average inverse noise variance is $\mathcal{I} \approx 0.575$ and the best performance again occurs when $\sigma_1^2 = \sigma_2^2$, as predicted in Remark 3.11. Note, in particular, that the dashed line and dotted curve are both tangent to the contour at exactly $\sigma_1^2 = \sigma_2^2$. The observation that larger noise variances have "more impact" provides a rough explanation for this phenomenon; homoscedasticity minimizes the largest noise variance for both the line and the curve. In some sense, as discussed in Section 3.2.6, the degradation from samples with larger noise is greater than the

benefit of having samples with correspondingly smaller noise.

3.3.4 Impact of adding data

Consider adding data with noise variance σ_2^2 and sample-to-dimension ratio c_2 to an existing dataset that has noise variance $\sigma_1^2 = 1$, sample-to-dimension ratio $c_1 = 10$ and subspace amplitude $\theta_i = 1$ for the ith component. The combined dataset has a sample-to-dimension ratio of $c = c_1 + c_2$ and is potentially heteroscedastic with noise variances σ_1^2 and σ_2^2 appearing in proportions $p_1 = c_1/c$ and $p_2 = c_2/c$. Figure 3.4 shows the asymptotic subspace recovery (3.4) of the ith component for this combined dataset as a function of the sample-to-dimension ratio c_2 of the added data for a variety of noise variances σ_2^2. The dashed orange curve, showing the recovery when $\sigma_2^2 = 1 = \sigma_1^2$, illustrates the benefit we would expect for homoscedastic data: increasing the samples per dimension improves recovery. The dotted red curve shows the recovery when $\sigma_2^2 = 4 > \sigma_1^2$. For a small number of added samples, the harm of introducing noisier data outweighs the benefit of having more samples. For sufficiently many samples, however, the tradeoff reverses and recovery for the combined dataset exceeds that for the original dataset; the break-even point can be calculated using expression (3.4). Finally, the green curve shows the performance when $\sigma_2^2 = 1.4 > \sigma_1^2$. As before, the added samples are noisier than the original samples and so we might expect performance to initially decline again. In this case, however, the performance improves for any number of added samples. In all three cases, the added samples dominate in the limit $c_2 \to \infty$ and PCA approaches perfect subspace recovery as one may expect. However, perfect recovery in the limit does not typically happen for PCA amplitudes (3.2) and coefficient recovery (3.5); see Section 3.8.3.4 for more details.

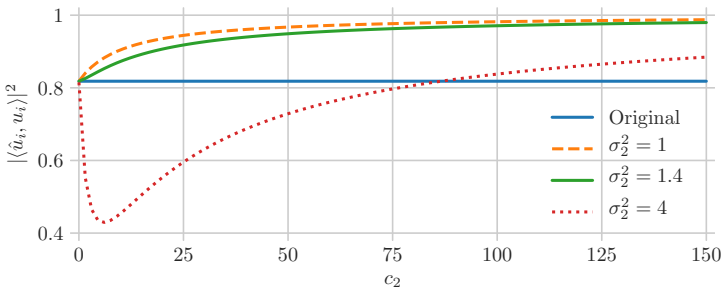
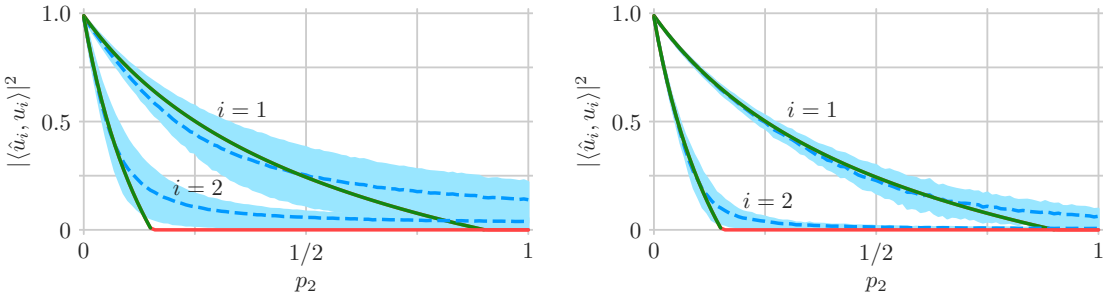


Figure 3.4: Asymptotic subspace recovery (3.4) of the ith component for samples added with noise variance σ_2^2 and samples-per-dimension c_2 to an existing dataset with noise variance $\sigma_1^2 = 1$, sample-to-dimension ratio $c_1 = 10$ and subspace amplitude $\theta_i = 1$.

Note that it is equivalent to think about removing noisy samples from a dataset by thinking of the combined dataset as the original full dataset. The green curve in Figure 3.4 then suggests that slightly noisier samples should not be removed; it would be best if the full data was homoscedastic but removing slightly noisier data (and reducing the dataset size) does more harm than good. The dotted red curve in Figure 3.4 suggests that much noisier samples should be removed unless they are numerous enough to outweigh the cost of adding them. Once again, expression (3.4) can be used to calculate the break-even point.

3.4 Numerical simulation

This section simulates data generated by the model described in Section 3.2.1 to illustrate the main result, Theorem 3.4, and to demonstrate that the asymptotic results provided are meaningful for practical settings with finitely many samples in a finite-dimensional space. As in Section 3.3, we show results only for the asymptotic subspace recovery (3.4) for brevity; the same phenomena occur for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5) as we show in Section 3.8.4. Consider data from a two-dimensional subspace with subspace amplitudes $\theta_1 = 1$ and



(a) 10^3 samples in 10^2 dimensions.

(b) 10^4 samples in 10^3 dimensions.

Figure 3.5: Simulated subspace recovery (3.4) as a function of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitudes are $\theta_1 = 1$ and $\theta_2 = 0.8$. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic recovery (3.4) of Theorem 3.4 (green curve). The region where $A(\beta_i) \leq 0$ is the red horizontal segment with value zero (the prediction of Conjecture 3.5). Increasing data size from (a) to (b) results in smaller interquartile intervals, indicating concentration to the mean, which is itself converging to the asymptotic recovery.

$\theta_2 = 0.8$, two noise variances $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 3.25$, and a sample-to-dimension ratio of $c = 10$. We sweep the proportion of high noise samples p_2 from zero to one, setting $p_1 = 1 - p_2$ as in Section 3.3.2. The first simulation considers $n = 10^3$ samples in a $d = 10^2$ dimensional ambient space (10^4 trials). The second increases these to $n = 10^4$ samples in a $d = 10^3$ dimensional ambient space (10^3 trials). Both simulations generate data from the standard normal distribution, i.e., $z_{ij}, \varepsilon_{ij} \sim \mathcal{N}(0, 1)$. Note that sweeping over p_2 covers homoscedastic settings at the extremes ($p_2 = 0, 1$) and evenly split heteroscedastic data in the middle ($p_2 = 1/2$).

Figure 3.5 plots the recovery of subspace components $|\langle \hat{u}_i, u_i \rangle|^2$ for both simulations with the mean (dashed blue curve) and interquartile interval (light blue ribbon) shown with the asymptotic recovery (3.4) of Theorem 3.4 (green curve). The region where $A(\beta_i) \leq 0$ is the red horizontal segment with value zero (the prediction of Conjecture 3.5). Figure 3.5a illustrates general agreement between the mean and the asymptotic recovery, especially far away from the non-differentiable points where the recovery becomes zero and Conjecture 3.5 predicts a phase transition. This is a general phenomenon we observed: near the phase transition the smooth simulation mean deviates from the non-smooth asymptotic recovery. Intuitively, an asymptotic recovery of zero corresponds to PCA components that are like isotropically random vectors and so have vanishing square inner product with the true components as the dimension grows. In finite dimension, however, there is a chance of alignment that results in a positive square inner product.

Figure 3.5b shows what happens when the number of samples and ambient dimension are increased to $n = 10^4$ and $d = 10^3$. The interquartile intervals are roughly half the size of those in Figure 3.5a, indicating concentration of the recovery of each component (a random quantity) around its mean. Furthermore, there is better agreement between the mean and the asymptotic recovery, with the maximum deviation between simulation and asymptotic prediction still occurring nearby the phase transition. In particular for $p_2 < 0.75$ the largest deviation for $|\langle \hat{u}_1, u_1 \rangle|^2$ is around 0.03. For $p_2 \notin (0.1, 0.35)$, the largest deviation for $|\langle \hat{u}_2, u_2 \rangle|^2$ is around 0.02. To summarize, the numerical simulations indicate that the subspace recovery concentrates to its mean and that the mean approaches the asymptotic recovery. Furthermore, good agreement with Conjecture 3.5 provides further evidence that there is indeed a phase transition below which the subspace is not recovered. These findings are similar to those in [94] for a one-dimensional subspace with two noise variances.

3.5 Proof of Theorem 3.4

The proof has six main parts. Section 3.5.1 connects several results from random matrix theory to obtain an initial expression for asymptotic recovery. This expression is difficult to evaluate and analyze because it involves an integral transform of the (nontrivial) limiting singular value distribution for a random (noise) matrix as well as the corresponding limiting largest singular value. However, we have discovered a nontrivial structure in this expression that enables us to derive a much simpler form in Sections 3.5.2-3.5.6.

3.5.1 Obtain an initial expression

Rewriting the model in (3.1) in matrix form yields

$$\mathbf{Y} = (y_1, \ldots, y_n) = \mathbf{U}\boldsymbol{\Theta}\mathbf{Z}^{\mathsf{H}} + \mathbf{E}\mathbf{H} \in \mathbb{C}^{d\times n}, \tag{3.17}$$

where

- $\mathbf{Z} = (z^{(1)}, \ldots, z^{(k)}) \in \mathbb{C}^{n\times k}$ is the coefficient matrix,
- $\mathbf{E} = (\varepsilon_1, \ldots, \varepsilon_n) \in \mathbb{C}^{d\times n}$ is the (unscaled) noise matrix,
- $\mathbf{H} = \operatorname{diag}(\eta_1, \ldots, \eta_n) \in \mathbb{R}_+^{n\times n}$ is a diagonal matrix of noise standard deviations.

The first k principal components $\hat{u}_1, \ldots, \hat{u}_k$, PCA amplitudes $\hat{\theta}_1, \ldots, \hat{\theta}_k$ and (normalized) scores $\hat{z}^{(1)}/\sqrt{n}, \ldots, \hat{z}^{(k)}/\sqrt{n}$ defined in Section 3.1 are exactly the first k left singular vectors, singular values and right singular vectors, respectively, of the scaled data matrix $\mathbf{Y}/\sqrt{n}$.

To match the model of [22], we introduce the random unitary matrix

$$\mathbf{R} = \begin{bmatrix} \breve{\mathbf{U}} & \breve{\mathbf{U}}^{\perp} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{U}^{\perp} \end{bmatrix}^{\mathsf{H}} = \breve{\mathbf{U}}\mathbf{U}^{\mathsf{H}} + \breve{\mathbf{U}}^{\perp}(\mathbf{U}^{\perp})^{\mathsf{H}},$$

where the random matrix $\breve{\mathbf{U}} \in \mathbb{C}^{d\times k}$ is the Gram-Schmidt orthonormalization of a $d \times k$ random matrix that has iid (mean zero, variance one) circularly symmetric complex normal $\mathcal{CN}(0,1)$ entries. We use the superscript $\perp$ to denote a matrix of orthonormal basis elements for the orthogonal complement; the columns of $\mathbf{U}^{\perp}$ form an orthonormal basis for the orthogonal complement of the column span of $\mathbf{U}$.

Left multiplying (3.17) by the scaled rotation $\mathbf{R}/\sqrt{n}$ yields that $\mathbf{R}\hat{u}_1, \ldots, \mathbf{R}\hat{u}_k$, $\hat{\theta}_1, \ldots, \hat{\theta}_k$ and $\hat{z}^{(1)}/\sqrt{n}, \ldots, \hat{z}^{(k)}/\sqrt{n}$ are the first k left singular vectors, singular values and right singular vectors, respectively, of the scaled and rotated data matrix

$$\tilde{\mathbf{Y}} = \frac{1}{\sqrt{n}}\mathbf{R}\mathbf{Y}.$$

The matrix $\tilde{\mathbf{Y}}$ matches the low rank (i.e., rank k) perturbation of a random matrix model considered in [22] because

$$\tilde{\mathbf{Y}} = \mathbf{P} + \mathbf{X},$$

where

$$\mathbf{P} = \frac{1}{\sqrt{n}}\,\mathbf{R}\left(\mathbf{U}\mathbf{\Theta}\mathbf{Z}^{\mathsf{H}}\right) = \frac{1}{\sqrt{n}}\,\breve{\mathbf{U}}\mathbf{\Theta}\mathbf{Z}^{\mathsf{H}} = \sum_{i=1}^{k}\theta_i\breve{u}_i\left(\frac{1}{\sqrt{n}}\,z^{(i)}\right)^{\mathsf{H}},$$
$$\mathbf{X} = \frac{1}{\sqrt{n}}\,\mathbf{R}\left(\mathbf{E}\mathbf{H}\right) = \left(\frac{1}{\sqrt{n}}\,\mathbf{R}\mathbf{E}\right)\mathbf{H}.$$

Here $\mathbf{P}$ is generated according to the "orthonormalized model" in [22] for the vectors $\breve{u}_i$ and the "iid model" for the vectors $z^{(i)}$ and $\mathbf{P}$ satisfies Assumption 2.4 of [22]; the latter considers $\breve{u}_i$ and $z^{(i)}$ to be generated according to the same model, but its proof extends to this case. Furthermore $\mathbf{RE}$ has iid entries with zero mean, unit variance and bounded fourth moment (by the assumption that ε_i are unitarily invariant), and $\mathbf{H}$ is a non-random diagonal positive definite matrix with bounded spectral norm and limiting eigenvalue distribution $p_1\delta_{\sigma_1^2} + \cdots + p_L\delta_{\sigma_L^2}$, where $\delta_{\sigma_\ell^2}$ is the Dirac delta distribution centered at σ_ℓ^2. Under these conditions, Theorem 4.3 and Corollary 6.6 of [14] state that $\mathbf{X}$ has a non-random compactly supported limiting singular value distribution $\mu_{\mathbf{X}}$ and the largest singular value of $\mathbf{X}$ converges almost surely to the supremum of the support of $\mu_{\mathbf{X}}$. Thus Assumptions 2.1 and 2.3 of [22] are also satisfied.

Furthermore, $\hat{u}_i^{\mathsf{H}}u_j = \hat{u}_i^{\mathsf{H}}\mathbf{R}^{\mathsf{H}}\mathbf{R}u_j = (\mathbf{R}\hat{u}_i)^{\mathsf{H}}\breve{u}_j$ for all $i, j \in \{1, \ldots, k\}$ so

$$|\langle \mathbf{R}\hat{u}_i, \operatorname{span}\{\breve{u}_j : \theta_j = \theta_i\}\rangle|^2 = |\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\}\rangle|^2,$$
$$|\langle \mathbf{R}\hat{u}_i, \operatorname{span}\{\breve{u}_j : \theta_j \neq \theta_i\}\rangle|^2 = |\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j \neq \theta_i\}\rangle|^2,$$

and hence Theorem 2.10 from [22] implies that, for each $i \in \{1, \ldots, k\}$,

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \begin{cases} \rho_i^2 & \text{if } \theta_i^2 > \bar{\theta}^2, \\ b^2 & \text{otherwise,} \end{cases} \tag{3.18}$$

and that if $\theta_i^2 > \bar{\theta}^2$, then

$$|\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} \frac{-2\varphi(\rho_i)}{\theta_i^2 D'(\rho_i)}, \tag{3.19}$$
$$\left|\left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j = \theta_i\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} \frac{-2\{c^{-1}\varphi(\rho_i) + (1 - c^{-1})/\rho_i\}}{\theta_i^2 D'(\rho_i)},$$

and

$$|\langle \hat{u}_i, \text{span}\{u_j : \theta_j \neq \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} 0, \tag{3.20}$$
$$\left|\left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \text{span}\{z^{(j)} : \theta_j \neq \theta_i\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} 0,$$

where $\rho_i = D^{-1}(1/\theta_i^2)$, $\bar{\theta}^2 = 1/D\left(b^+\right)$, $D(z) = \varphi(z)\{c^{-1}\varphi\left(z\right) + (1-c^{-1})/z\}$ for $z > b$, $\varphi\left(z\right) = \int z/(z^2 - t^2)\, d\mu_{\mathbf{X}}\left(t\right)$, b is the supremum of the support of $\mu_{\mathbf{X}}$ and $\mu_{\mathbf{X}}$ is the limiting singular value distribution of $\mathbf{X}$ (compactly supported by Assumption 2.1 of [22]). We use the notation $f\left(b^+\right) = \lim_{z \to b^+} f\left(z\right)$ as a convenient shorthand for the limit from above of a function $f\left(z\right)$.

Theorem 2.10 from [22] is presented therein for $d \leq n$ (i.e., $c \geq 1$) to simplify their proofs. However, it also holds without modification for $d > n$ if the limiting singular value distribution $\mu_{\mathbf{X}}$ is always taken to be the limit of the empirical distribution of the d largest singular values ($d - n$ of which will be zero). Thus we proceed without the condition that $c > 1$.

Furthermore, even though it is not explicitly stated as a main result in [22], the proof of Theorem 2.10 in [22] implies that

$$\sum_{j:\theta_j = \theta_i} \langle \hat{u}_i, u_j \rangle \left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \frac{z^{(j)}}{\|z^{(j)}\|} \right\rangle^* \xrightarrow{\text{a.s.}} \sqrt{\frac{-2\varphi(\rho_i)}{\theta_i^2 D'(\rho_i)} \frac{-2\{c^{-1}\varphi(\rho_i) + (1-c^{-1})/\rho_i\}}{\theta_i^2 D'(\rho_i)}}, \tag{3.21}$$

as was also noted in [145] for the special case of distinct subspace amplitudes.

Evaluating the expressions (3.18), (3.19) and (3.21) would consist of evaluating the intermediates listed above from last to first. These steps are challenging because they involve an integral transform of the limiting singular value distribution $\mu_{\mathbf{X}}$ for the random (noise) matrix $\mathbf{X}$ as well as the corresponding limiting largest singular value b, both of which depend nontrivially on the model parameters. Our analysis uncovers a nontrivial structure that we exploit to derive simpler expressions.

Before proceeding, observe that the almost sure limit in (3.21) is just the geometric mean of the two almost sure limits in (3.19). Hence, we proceed to derive simplified expressions for (3.18) and (3.19); (3.6) follows as the geometric mean of the simplified expressions obtained for the almost sure limits in (3.19).

3.5.2 Perform a change of variables

We introduce the function defined, for $z > b$, by

$$\psi\left(z\right) = \frac{cz}{\varphi\left(z\right)} = \left\{\frac{1}{c}\int \frac{1}{z^2 - t^2} d\mu_{\mathbf{X}}\left(t\right)\right\}^{-1}, \tag{3.22}$$

because it turns out to have several nice properties that simplify all of the following analysis. Rewriting (3.19) using $\psi(z)$ instead of $\varphi(z)$ and factoring appropriately yields that if $\theta_i^2 > \bar{\theta}^2$ then

$$|\langle \hat{u}_i, \operatorname{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} \frac{1}{\psi(\rho_i)} \frac{-2c}{\theta_i^2 D'(\rho_i)/\rho_i}, \tag{3.23}$$

$$\left| \left\langle \frac{\hat{z}^{(i)}}{\sqrt{n}}, \operatorname{span}\{z^{(j)} : \theta_j = \theta_i\} \right\rangle \right|^2 \xrightarrow{\text{a.s.}} \frac{1}{c\{\psi(\rho_i) + (1-c)\theta_i^2\}} \frac{-2c}{\theta_i^2 D'(\rho_i)/\rho_i},$$

where now

$$D(z) = \frac{cz^2}{\psi^2(z)} + \frac{c-1}{\psi(z)} \tag{3.24}$$

for $z > b$ and we have used the fact that

$$\frac{1}{c}\left\{\frac{1}{\psi(\rho_i)} + \frac{1-c^{-1}}{\rho_i^2}\right\} = \frac{1}{c}\left\{\psi(\rho_i) + \frac{1-c}{D(\rho_i)}\right\}^{-1} = \frac{1}{c\{\psi(\rho_i) + (1-c)\theta_i^2\}}.$$

3.5.3 Find useful properties of $\psi(z)$

Establishing some properties of $\psi(z)$ aids simplification significantly.

Property 1. We show that $\psi(z)$ satisfies a certain rational equation for all $z > b$ and derive its inverse function $\psi^{-1}(x)$. Observe that the square singular values of the noise matrix $\mathbf{X}$ are exactly the eigenvalues of $c\mathbf{X}\mathbf{X}^{\mathsf{H}}$, divided by c. Thus we first consider the limiting eigenvalue distribution $\mu_{c\mathbf{X}\mathbf{X}^{\mathsf{H}}}$ of $c\mathbf{X}\mathbf{X}^{\mathsf{H}}$ and then relate its Stieltjes transform $m(\zeta)$ to $\psi(z)$.

Theorem 4.3 in [14] establishes that the random matrix $c\mathbf{X}\mathbf{X}^{\mathsf{H}} = (1/d)\mathbf{E}\mathbf{H}^2\mathbf{E}^{\mathsf{H}}$ has a limiting eigenvalue distribution $\mu_{c\mathbf{X}\mathbf{X}^{\mathsf{H}}}$ whose Stieltjes transform is given, for $\zeta \in \mathbb{C}^+$, by

$$m(\zeta) = \int \frac{1}{t-\zeta} d\mu_{c\mathbf{X}\mathbf{X}^{\mathsf{H}}}(t), \tag{3.25}$$

and satisfies the condition

$$\forall_{\zeta \in \mathbb{C}^+} \quad m(\zeta) = -\left\{\zeta - c\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{1 + \sigma_\ell^2 m(\zeta)}\right\}^{-1}, \tag{3.26}$$

where $\mathbb{C}^+$ is the set of all complex numbers with positive imaginary part.

Since the d square singular values of $\mathbf{X}$ are exactly the d eigenvalues of $c\mathbf{X}\mathbf{X}^{\mathsf{H}}$ divided by c, we have for all $z > b$

$$\psi(z) = \left\{\frac{1}{c}\int \frac{1}{z^2 - t^2} d\mu_{\mathbf{X}}(t)\right\}^{-1} = -\left\{\int \frac{1}{t - z^2 c} d\mu_{c\mathbf{X}\mathbf{X}^{\mathsf{H}}}(t)\right\}^{-1}. \tag{3.27}$$

For all z and $\xi > 0$, $z^2 c + i\xi \in \mathbb{C}^+$ and so combining (3.25)–(3.27) yields that for all $z > b$

$$\psi(z) = -\left\{\lim_{\xi \to 0^+} m(z^2 c + i\xi)\right\}^{-1} = z^2 c - c\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{1 - \sigma_\ell^2/\psi(z)}.$$

Rearranging yields

$$0 = \frac{cz^2}{\psi^2(z)} - \frac{1}{\psi(z)} - \frac{c}{\psi(z)} \sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{\psi(z) - \sigma_\ell^2}, \tag{3.28}$$

for all $z > b$, where the last term is

$$-\frac{c}{\psi(z)} \sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{\psi(z) - \sigma_\ell^2} = \frac{c}{\psi(z)} - c\sum_{\ell=1}^{L} \frac{p_\ell}{\psi(z) - \sigma_\ell^2},$$

because $p_1 + \cdots + p_L = 1$. Substituting back into (3.28) yields $0 = Q\{\psi(z), z\}$ for all $z > b$, where

$$Q(s, z) = \frac{cz^2}{s^2} + \frac{c-1}{s} - c\sum_{\ell=1}^{L} \frac{p_\ell}{s - \sigma_\ell^2}. \tag{3.29}$$

Thus $\psi(z)$ is an algebraic function (the associated polynomial can be formed by clearing the denominator of Q). Solving (3.29) for $z > b$ yields the inverse

$$\psi^{-1}(x) = \sqrt{\frac{1-c}{c}x + x^2 \sum_{\ell=1}^{L} \frac{p_\ell}{x - \sigma_\ell^2}} = \sqrt{\frac{x}{c}\left(1 + c\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{x - \sigma_\ell^2}\right)}. \tag{3.30}$$

Property 2. We show that $\max_\ell(\sigma_\ell^2) < \psi(z) < cz^2$ for $z > b$. For $z > b$, one can show from (3.22) that $\psi(y)$ increases continuously and monotonically from $\psi(z)$ to infinity as y increases from z to infinity, and hence $\psi^{-1}(x)$ must increase continuously and monotonically from z to infinity as x increases from $\psi(z)$ to infinity. However, $\psi^{-1}(x)$ is discontinuous at $x = \max_\ell(\sigma_\ell^2)$ because $\psi^{-1}(x) \to \infty$ as $x \to \max_\ell(\sigma_\ell^2)$ from the right, and so it follows that $\psi(z) > \max_\ell(\sigma_\ell^2)$. Thus $1/\{\psi(z) - \sigma_\ell^2\} > 0$ for all $\ell \in \{1, \ldots, L\}$ and so

$$cz^2 = c[\psi^{-1}\{\psi(z)\}]^2 = \psi(z)\left\{1 + c\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{\psi(z) - \sigma_\ell^2}\right\} > \psi(z).$$

Property 3. We show that $0 < \psi(b^+) < \infty$ and $\psi'(b^+) = \infty$. Property 2 in the limit $z = b^+$ implies that

$$0 < \max_\ell(\sigma_\ell^2) \le \psi(b^+) \le cb^2 < \infty.$$

Taking the total derivative of $0 = Q\{\psi(z), z\}$ with respect to z and solving for $\psi'(z)$ yields

$$\psi'(z) = -\frac{\partial Q}{\partial z}\{\psi(z), z\} \Big/ \frac{\partial Q}{\partial s}\{\psi(z), z\}. \tag{3.31}$$

As observed in [146], the non-pole boundary points of compactly supported distributions like $\mu_{c\mathbf{X}\mathbf{X}^{\mathsf{H}}}$ occur where the polynomial defining their Stieltjes transform has multiple roots. Thus $\psi(b^+)$ is a multiple root of $Q(\cdot, b)$ and so

$$\frac{\partial Q}{\partial s}\{\psi(b^+), b\} = 0, \quad \frac{\partial Q}{\partial z}\{\psi(b^+), b\} = \frac{2cb}{\psi^2(b^+)} > 0.$$

Thus $\psi'(b^+) = \infty$, where the sign is positive because $\psi(z)$ is an increasing function on $z > b$.

Summarizing, we have shown that

a) $0 = Q\{\psi(z), z\}$ for all $z > b$ where Q is defined in (3.29), and the inverse function $\psi^{-1}(x)$ is given in (3.30),

b) $\max_\ell(\sigma_\ell^2) < \psi(z) < cz^2$,

c) $0 < \psi(b^+) < \infty$ and $\psi'(b^+) = \infty$.

We now use these properties to aid simplification.

3.5.4 Express $D(z)$ and $D'(z)/z$ in terms of only $\psi(z)$

We can rewrite (3.24) as

$$D(z) = Q\{\psi(z), z\} + c\sum_{\ell=1}^{L} \frac{p_\ell}{\psi(z) - \sigma_\ell^2} = c\sum_{\ell=1}^{L} \frac{p_\ell}{\psi(z) - \sigma_\ell^2}. \tag{3.32}$$

because $0 = Q\{\psi(z), z\}$ by Property 1 of Section 3.5.3. Differentiating (3.32) with respect to z yields

$$D'(z) = -c\psi'(z)\sum_{\ell=1}^{L} \frac{p_\ell}{\{\psi(z) - \sigma_\ell^2\}^2},$$

and so we need to find $\psi'(z)$ in terms of $\psi(z)$. Substituting the expressions for the partial derivatives $\partial Q\{\psi(z), z\}/\partial z$ and $\partial Q\{\psi(z), z\}/\partial s$ into (3.31) and simplifying we obtain $\psi'(z) = 2cz/\gamma(z)$, where the denominator is

$$\gamma(z) = c - 1 + \frac{2cz^2}{\psi(z)} - c\sum_{\ell=1}^{L} \frac{p_\ell \psi^2(z)}{\{\psi(z) - \sigma_\ell^2\}^2}.$$

Note that

$$\frac{2cz^2}{\psi(z)} = -2(c-1) + c\sum_{\ell=1}^{L}\frac{2p_\ell\psi(z)}{\psi(z)-\sigma_\ell^2},$$

because $0 = Q\{\psi(z), z\}$ for $z > b$. Substituting into $\gamma(z)$ and forming a common denominator, then dividing with respect to $\psi(z)$ yields

$$\gamma(z) = 1 - c + c\sum_{\ell=1}^{L} p_\ell \frac{\psi^2(z) - 2\psi(z)\sigma_\ell^2}{\{\psi(z)-\sigma_\ell^2\}^2} = 1 - c\sum_{\ell=1}^{L}\frac{p_\ell\sigma_\ell^4}{\{\psi(z)-\sigma_\ell^2\}^2} = A\{\psi(z)\},$$

where $A(x)$ was defined in (3.3). Thus

$$\psi'(z) = \frac{2cz}{A\{\psi(z)\}}, \tag{3.33}$$

and

$$\frac{D'(z)}{z} = -\frac{2c^2}{A\{\psi(z)\}}\sum_{\ell=1}^{L}\frac{p_\ell}{\{\psi(z)-\sigma_\ell^2\}^2} = -\frac{2c}{\theta_i^2}\frac{B_i'\{\psi(z)\}}{A\{\psi(z)\}}, \tag{3.34}$$

where $B_i'(x)$ is the derivative of $B_i(x)$ defined in (3.3).

3.5.5 Express the asymptotic recoveries in terms of only $\psi(b^+)$ and $\psi(\rho_i)$

Evaluating (3.32) in the limit $z = b^+$ and recalling that $D(b^+) = 1/\bar{\theta}^2$ yields

$$\theta_i^2 > \bar{\theta}^2 \quad \Leftrightarrow \quad 0 > 1 - \frac{\theta_i^2}{\bar{\theta}^2} = 1 - c\theta_i^2\sum_{\ell=1}^{L}\frac{p_\ell}{\psi(b^+)-\sigma_\ell^2} = B_i\{\psi(b^+)\}, \tag{3.35}$$

where $B_i(x)$ was defined in (3.3). Evaluating the inverse function (3.30) both for $\psi(\rho_i)$ and in the limit $\psi(b^+)$ then substituting into (3.18) yields

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \begin{cases} \dfrac{\psi(\rho_i)}{c}\left\{1 + c\displaystyle\sum_{\ell=1}^{L}\frac{p_\ell\sigma_\ell^2}{\psi(\rho_i)-\sigma_\ell^2}\right\} & \text{if } B_i\{\psi(b^+)\} < 0, \\ \dfrac{\psi(b^+)}{c}\left\{1 + c\displaystyle\sum_{\ell=1}^{L}\frac{p_\ell\sigma_\ell^2}{\psi(b^+)-\sigma_\ell^2}\right\} & \text{otherwise.} \end{cases} \tag{3.36}$$

Evaluating (3.34) for $z = \rho_i$ and substituting into (3.23) yields

$$|\langle\hat{u}_i, \text{span}\{u_j : \theta_j = \theta_i\}\rangle|^2 \xrightarrow{\text{a.s.}} \frac{1}{\psi(\rho_i)}\frac{A\{\psi(\rho_i)\}}{B_i'\{\psi(\rho_i)\}}, \tag{3.37}$$

$$\left|\left\langle\frac{\hat{z}^{(i)}}{\sqrt{n}}, \text{span}\{z^{(j)} : \theta_j = \theta_i\}\right\rangle\right|^2 \xrightarrow{\text{a.s.}} \frac{1}{c\{\psi(\rho_i) + (1-c)\theta_i^2\}}\frac{A\{\psi(\rho_i)\}}{B_i'\{\psi(\rho_i)\}},$$

if $B_i\{\psi(b^+)\} < 0$.

3.5.6 Obtain algebraic descriptions

This subsection obtains algebraic descriptions of (3.35), (3.36) and (3.37) by showing that $\psi(b^+)$ is the largest real root of $A(x)$ and that $\psi(\rho_i)$ is the largest real root of $B_i(x)$ when $\theta_i^2 > \bar{\theta}^2$. Evaluating (3.33) in the limit $z = b^+$ yields

$$A\{\psi\left(b^+\right)\} = \frac{2cb}{\psi'\left(b^+\right)} = 0, \tag{3.38}$$

because $\psi'\left(b^+\right) = \infty$ by Property 3 of Section 3.5.3. If $\theta_i^2 > \bar{\theta}^2$ then $\rho_i = D^{-1}(1/\theta_i^2)$ and so

$$0 = 1 - \theta_i^2 D\left(\rho_i\right) = 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{\psi\left(\rho_i\right) - \sigma_\ell^2} = B_i\{\psi\left(\rho_i\right)\}. \tag{3.39}$$

(3.38) shows that $\psi\left(b^+\right)$ is a real root of $A(x)$, and (3.39) shows that $\psi\left(\rho_i\right)$ is a real root of $B_i(x)$.

Recall that $\psi(b^+), \psi(\rho_i) \geq \max_\ell(\sigma_\ell^2)$ by Property 2 of Section 3.5.3, and note that both $A(x)$ and $B_i(x)$ monotonically increase for $x > \max_\ell(\sigma_\ell^2)$. Thus each has exactly one real root larger than $\max_\ell(\sigma_\ell^2)$, i.e., its largest real root, and so $\psi(b^+) = \alpha$ and $\psi(\rho_i) = \beta_i$ when $\theta_i^2 > \bar{\theta}^2$, where α and β_i are the largest real roots of $A(x)$ and $B_i(x)$, respectively.

A subtle point is that $A(x)$ and $B_i(x)$ always have largest real roots α and β even though $\psi(\rho_i)$ is defined only when $\theta_i^2 > \bar{\theta}^2$. Furthermore, α and β are always larger than $\max_\ell(\sigma_\ell^2)$ and both $A(x)$ and $B_i(x)$ are monotonically increasing in this regime and so we have the equivalence

$$B_i\left(\alpha\right) < 0 \quad \Leftrightarrow \quad \alpha < \beta_i \quad \Leftrightarrow \quad 0 < A\left(\beta_i\right). \tag{3.40}$$

Writing (3.35), (3.36) and (3.37) in terms of α and β_i, then applying the equivalence (3.40) and combining with (3.20) yields the main results (3.2), (3.4) and (3.5).

3.6 Proof of Theorem 3.9

If $A(\beta_i) \geq 0$ then (3.4) and (3.5) increase with $A(\beta_i)$ and decrease with β_i and $B'(\beta_i)$. Similarly, (3.2) increases with β_i, as illustrated by (3.9). As a result, Theorem 3.9 follows immediately from the following bounds, all of which are met with equality if and only if $\sigma_1^2 = \cdots = \sigma_L^2$:

$$\beta_i \geq c\theta_i^2 + \bar{\sigma}^2, \quad B_i'(\beta_i) \geq \frac{1}{c\theta_i^2}, \quad A(\beta_i) \leq 1 - \frac{1}{c}\left(\frac{\bar{\sigma}}{\theta_i}\right)^4. \tag{3.41}$$

The bounds (3.41) are shown by exploiting convexity to appropriately bound the rational functions $B_i(x)$, $B_i'(x)$ and $A(x)$. We bound β_i by noting that

$$0 = B_i(\beta_i) = 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{\beta_i - \sigma_\ell^2} \leq 1 - \frac{c\theta_i^2}{\beta_i - \bar{\sigma}^2},$$

because $\sigma_\ell^2 < \beta_i$ and $f(v) = 1/(\beta_i - v)$ is a strictly convex function over $v < \beta_i$. Thus $\beta_i \geq c\theta_i^2 + \bar{\sigma}^2$. We bound $B_i'(\beta_i)$ by noting that

$$B_i'(\beta_i) = c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{(\beta_i - \sigma_\ell^2)^2} \geq c\theta_i^2 \left(\sum_{\ell=1}^{L} \frac{p_\ell}{\beta_i - \sigma_\ell^2} \right)^2 = c\theta_i^2 \left(\frac{1}{c\theta_i^2} \right)^2 = \frac{1}{c\theta_i^2},$$

because the quadratic function z^2 is strictly convex. Similarly,

$$A(\beta_i) = 1 - c \sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^4}{(\beta_i - \sigma_\ell^2)^2} \leq 1 - c \left(\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{\beta_i - \sigma_\ell^2} \right)^2 \leq 1 - \frac{1}{c} \left(\frac{\bar{\sigma}}{\theta_i} \right)^4,$$

because the quadratic function z^2 is strictly convex and

$$\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^2}{\beta_i - \sigma_\ell^2} = \beta_i \sum_{\ell=1}^{L} \frac{p_\ell}{\beta_i - \sigma_\ell^2} - 1 = \frac{\beta_i}{c\theta_i^2} - 1 \geq \frac{c\theta_i^2 + \bar{\sigma}^2}{c\theta_i^2} - 1 = \frac{\bar{\sigma}^2}{c\theta_i^2}.$$

All of the above bounds are met with equality if and only if $\sigma_1^2 = \cdots = \sigma_L^2$ because the convexity in all cases is strict. As a result, homoscedastic noise minimizes (3.2), and it maximizes (3.4) and (3.5). See Section 3.8.2 for some interesting additional properties in this context.

3.7 Discussion

This chapter provided simplified expressions (Theorem 3.4) for the asymptotic recovery of a low-dimensional subspace, the corresponding subspace amplitudes and the corresponding coefficients by the principal components, PCA amplitudes and scores, respectively, obtained from applying PCA to noisy high-dimensional heteroscedastic data. The simplified expressions provide generalizations of previous results for the special case of homoscedastic data. They were derived by first connecting several recent results from random matrix theory [14, 22] to obtain initial expressions for asymptotic recovery that are difficult to evaluate and analyze, then identifying and exploiting a nontrivial structure in the expressions to find the much simpler algebraic descriptions of Theorem 3.4.

These descriptions enable both easy and efficient calculation as well as reasoning about the asymptotic performance of PCA. In particular, we use the simplified

expressions to show that, for a fixed average noise variance, asymptotic subspace recovery, amplitude recovery and coefficient recovery are all worse when the noise is heteroscedastic as opposed to homoscedastic (Theorem 3.9). Hence, while average noise variance is often a practically convenient measure for the overall quality of data, it gives an overly optimistic estimate of PCA performance. Our expressions (3.2), (3.4) and (3.5) in Theorem 3.4 are more accurate.

We also investigated examples to gain insight into how the asymptotic performance of PCA depends on the model parameters: sample-to-dimension ratio c, subspace amplitudes $\theta_1, \ldots, \theta_k$, proportions $p_1, \ldots, p_L$ and noise variances $\sigma_1^2, \ldots, \sigma_L^2$. We found that performance depends in expected ways on

a) sample-to-dimension ratio: performance improves with more samples;

b) subspace amplitudes: performance improves with larger amplitudes;

c) proportions: performance improves when more samples have low noise.

We also learned that when the gap between the two largest noise variances is "sufficiently wide", the performance is dominated by the largest noise variance. This result provides insight into why PCA performs poorly in the presence of gross errors and why heteroscedasticity degrades performance in the sense of Theorem 3.9. Nevertheless, adding "slightly" noisier samples to an existing dataset can still improve PCA performance; even adding significantly noisier samples can be beneficial if they are sufficiently numerous.

Finally, we presented numerical simulations that demonstrated concentration of subspace recovery to the asymptotic prediction (3.4) with good agreement for practical problem sizes. The same agreement occurs for the PCA amplitudes and coefficient recovery. The simulations also showed good agreement with the conjectured phase transition (Conjecture 3.5).

There are many exciting avenues for extensions and further work. Chapter IV extends the analysis here to a weighted variant of PCA that gives noisier samples less weight. Another natural direction is to consider general noise variance distributions ν, where the empirical noise distribution $(\delta_{\eta_1^2} + \cdots + \delta_{\eta_n^2})/n \xrightarrow{\text{a.s.}} \nu$ as $n \to \infty$. We conjecture that if $\eta_1, \ldots, \eta_n$ are bounded for all n and $\int d\nu(\tau)/(x-\tau) \to \infty$ as $x \to \tau_{\max}^+$, then the almost sure limits in this chapter hold but with

$$A(x) = 1 - c \int \frac{\tau^2 d\nu(\tau)}{(x-\tau)^2}, \quad B_i(x) = 1 - c\theta_i^2 \int \frac{d\nu(\tau)}{x-\tau},$$

where $\tau_{\max}$ is the supremum of the support of ν. The proofs of Theorem 3.4 and Theorem 3.9 both generalize straightforwardly for the most part; the main trickiness comes in carefully arguing that limits pass through integrals in Section 3.5.3.

Proving that there is indeed a phase transition in the asymptotic subspace recovery and coefficient recovery, as conjectured in Conjecture 3.5, is another area of future work. That proof may be of greater interest in the context of a weighted PCA method. Another area of future work is explaining the puzzling phenomenon described in Section 3.3.3, where, in some regimes, performance improves by increasing the noise variance. More detailed analysis of the general impacts of the model parameters could also be interesting. A final direction of future work is deriving finite sample results for heteroscedastic noise as was done for homoscedastic noise in [147].

3.8 Supplementary material

This section provides supplementary discussion of some details from Chapter III. Section 3.8.1 relates the model (3.1) in this chapter to spiked covariance models [15, 110]. Section 3.8.2 discusses interesting properties of the simplified expressions. Section 3.8.3 shows the impact of the parameters on the asymptotic PCA amplitudes and coefficient recovery. Section 3.8.4 contains numerical simulation results for PCA amplitudes and coefficient recovery, and Section 3.8.5 simulates complex-valued and Gaussian mixture data.

3.8.1 Relationship to spiked covariance models

The model (3.1) considered in this chapter is similar in spirit to the generalized spiked covariance model of [15]. To discuss the relationship more easily, we will refer to the model (3.1) as the "inter-sample heteroscedastic model". Both this and the generalized spiked covariance model generalize the Johnstone spiked covariance model proposed in [110]. In the Johnstone spiked covariance model [15], sample vectors $y_1, \dots, y_n \in \mathbb{C}^d$ are generated as

$$y_i = \operatorname{diag}(\alpha_1^2, \dots, \alpha_k^2, \underbrace{1, \dots, 1}_{d-k \text{ copies}})^{1/2} x_i, \tag{3.42}$$

where $x_i \in \mathbb{C}^d$ are independent identically distributed (iid) vectors with iid entries that have mean $\mathbb{E}(x_{ij}) = 0$ and variance $\mathbb{E}|x_{ij}|^2 = 1$.

For normally distributed subspace coefficients and noise vectors, the inter-sample heteroscedastic model (3.1) is equivalent (up to rotation) to generating sample vectors $y_1, \dots, y_n \in \mathbb{C}^d$ as

$$y_i = \operatorname{diag}(\theta_1^2 + \eta_i^2, \dots, \theta_k^2 + \eta_i^2, \underbrace{\eta_i^2, \dots, \eta_i^2}_{d-k \text{ copies}})^{1/2} x_i, \tag{3.43}$$

where $x_i \in \mathbb{C}^d$ are iid with iid normally distributed entries. (3.43) generalizes the Johnstone spiked covariance model because the covariance matrix can vary across samples. Heterogeneity here is *across* samples; all entries $(y_i)_1, \dots, (y_i)_d$ within each sample y_i have equal noise variance η_i^2.

The generalized spiked covariance model generalizes the Johnstone spiked covariance model differently. In the generalized spiked covariance model [15], sample vectors $y_1, \dots, y_n \in \mathbb{C}^d$ are generated as

$$y_i = \begin{bmatrix} \mathbf{\Lambda} & \\ & \mathbf{V}_{d-k} \end{bmatrix}^{1/2} x_i, \tag{3.44}$$

where $x_i \in \mathbb{C}^d$ are iid with iid entries as in (3.42), $\mathbf{\Lambda} \in \mathbb{C}^{k \times k}$ is a deterministic Hermitian matrix with eigenvalues $\alpha_1^2, \dots, \alpha_k^2$, $\mathbf{V}_{d-k} \in \mathbb{R}^{(d-k)\times(d-k)}$ has limiting eigenvalue distribution ν, and these all satisfy a few technical conditions [15]. All samples share a common covariance matrix, but the model allows, among other things, for heterogenous variance within the samples. To illustrate this flexibility, note that we could set

$$\mathbf{\Lambda} = \operatorname{diag}(\theta_1^2 + \eta_1^2, \dots, \theta_k^2 + \eta_k^2), \qquad \mathbf{V}_{d-k} = \operatorname{diag}(\eta_{k+1}^2, \dots, \eta_d^2). \tag{3.45}$$

In this case, there is heteroscedasticity among the entries of each sample vector. Heterogeneity here is *within* each sample, not across them; recall that all samples have the same covariance matrix.

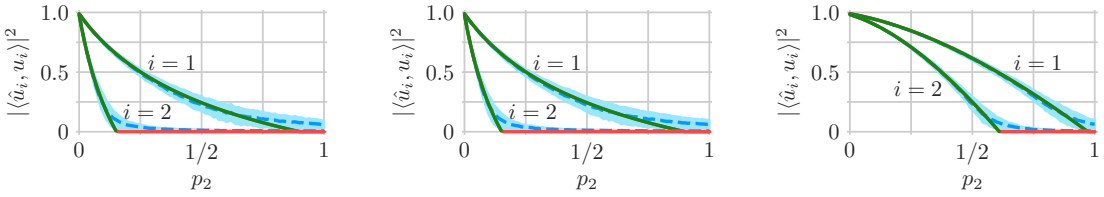
Therefore, for data with *intra*-sample heteroscedasticity, one should use the results of [15] and [215] for the generalized spiked covariance model. For data with *inter*-sample heteroscedasticity, one should use the new results presented in Theorem 3.4. A couple variants of the inter-sample heteroscedastic model are also natural to consider in the context of spiked covariance models; the next two subsections discuss these.

3.8.1.1 Random noise variances

The noise variances $\eta_1^2, \dots, \eta_n^2$ in the inter-sample heteroscedastic model (3.1) are deterministic. A natural variation could be to instead make them iid random variables defined as

$$\eta_i^2 = \begin{cases} \sigma_1^2 & \text{with probability } p_1, \\ \vdots & \\ \sigma_L^2 & \text{with probability } p_L, \end{cases} \tag{3.46}$$

where $p_1 + \cdots + p_L = 1$. To ease discussion, this section will use the words "deterministic" and "random" before "inter-sample heteroscedastic model" to differentiate



(a) Deterministic inter-sample heteroscedastic model.

(b) Random inter-sample heteroscedastic model.

(c) Johnstone spiked covariance model with covariance (3.47).

Figure 3.6: Simulated subspace recovery as a function of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. Subspace amplitudes are $\theta_1 = 1$ and $\theta_2 = 0.8$, and there are 10^4 samples in 10^3 dimensions. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic recovery (3.4) of Theorem 3.4 (green curve). The region where $A(\beta_i) \leq 0$ is the red horizontal segment with value zero (the prediction of Conjecture 3.5). Deterministic noise variances $\eta_1^2, \ldots, \eta_n^2$ are used for simulations in (a), random ones are used for those in (b), and (c) has data generated according to the Johnstone spiked covariance model with covariance matrix set as (3.47).

between the model (3.1) that has deterministic noise variances and its variant that instead has iid random noise variances (3.46). In the random inter-sample heteroscedastic model, scaled noise vectors $\eta_1 \varepsilon_1, \ldots, \eta_n \varepsilon_n$ are iid vectors drawn from a mixture. As a result, sample vectors $y_1, \ldots, y_n$ are also iid vectors with covariance matrix (up to rotation)

$$\mathbb{E}(y_i y_i^{\mathsf{H}}) = \operatorname{diag}(\theta_1^2 + \bar{\sigma}^2, \ldots, \theta_k^2 + \bar{\sigma}^2, \underbrace{\bar{\sigma}^2, \ldots, \bar{\sigma}^2}_{d-k \text{ copies}}), \tag{3.47}$$

where $\bar{\sigma}^2 = p_1 \sigma_1^2 + \cdots + p_L \sigma_L^2$ is the average variance.

(3.47) is a spiked covariance matrix and the samples $y_1, \ldots, y_n$ are iid vectors, and so it could be tempting to think that the data can be equivalently generated from the Johnstone spiked covariance model with covariance matrix (3.47). However this is not true. The PCA performance of the random inter-sample heteroscedastic model is similar to that of the deterministic version and is different from that of the Johnstone spiked covariance model with covariance matrix (3.47). Figure 3.6 illustrates the distinction in numerical simulations. In all simulations, we drew 10^4 samples from a 10^3 dimensional ambient space, where the subspace amplitudes were $\theta_1 = 1$ and $\theta_2 = 0.8$. Two noise variances $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 3.25$ have proportions $p_1 = 1 - p_2$ and p_2. In Figure 3.6a, data are generated according to the deterministic

inter-sample heteroscedastic model. In Figure 3.6b, data are generated according to the random inter-sample heteroscedastic model. In Figure 3.6c, data are generated according to the Johnstone spiked covariance model with covariance matrix (3.47).

Figures 3.6a and 3.6b demonstrate that data generated according to the inter-sample heteroscedastic model have similar behavior whether the noise variances $\eta_1^2, \dots, \eta_n^2$ are set deterministically or randomly as (3.46). The similarity is expected because the random noise variances in the limit will equal $\sigma_1^2, \dots, \sigma_L^2$ in proportions approaching $p_1, \dots, p_L$ by the law of large numbers. Thus data generated with random noise variances should have similar asymptotic PCA performance as data generated with deterministic noise variances.

Figures 3.6b and 3.6c demonstrate that data generated according to the random inter-sample heteroscedastic model behave quite differently from data generated according to the Johnstone spiked covariance model, even though both have iid sample vectors with covariance matrix (3.47). To understand why, recall that in the random inter-sample heteroscedastic model, the noise standard deviation η_i is shared among the entries of the scaled noise vector $\eta_i \varepsilon_i$. This induces statistical dependence among the entries of the sample vector y_i that is not eliminated by whitening with $\mathbb{E}(y_i y_i^{\mathsf{H}})^{-1/2}$. Whitening a sample vector y_i generated according to the Johnstone spiked covariance model, on the other hand, produces the vector x_i that has iid entries by definition. Thus, the random inter-sample heteroscedastic model is not equivalent to the Johnstone spiked covariance model. One should use Theorem 3.4 to analyze asymptotic PCA performance in this setting rather than existing results for the Johnstone spiked covariance model [22, 29, 111, 147, 163].

3.8.1.2 Row samples

In matrix form, the inter-sample heteroscedastic model can be written as

$$\mathbf{Y} = (y_1, \dots, y_n) = \mathbf{U}\mathbf{\Theta}\mathbf{Z}^{\mathsf{H}} + \mathbf{E}\mathbf{H} \in \mathbb{C}^{d \times n},$$

where

$\mathbf{Z} = (z^{(1)}, \dots, z^{(k)}) \in \mathbb{C}^{n \times k}$ is the coefficient matrix,

$\mathbf{E} = (\varepsilon_1, \dots, \varepsilon_n) \in \mathbb{C}^{d \times n}$ is the (unscaled) noise matrix,

$\mathbf{H} = \operatorname{diag}(\eta_1, \dots, \eta_n) \in \mathbb{R}_+^{n \times n}$ is a diagonal matrix of noise standard deviations.

Samples in this chapter are the columns $y_1, \ldots, y_n$ of the data matrix $\mathbf{Y}$, but one could alternatively form samples from the rows

$$y^{(i)} = \begin{bmatrix} (y_1)_i \\ \vdots \\ (y_n)_i \end{bmatrix} = \mathbf{Z}^* \mathbf{\Theta} u^{(i)} + \mathbf{H} \varepsilon^{(i)}, \tag{3.48}$$

where $u^{(i)} = ((u_1)_i, \ldots, (u_n)_i)$ and $\varepsilon^{(i)} = ((\varepsilon_1)_i, \ldots, (\varepsilon_n)_i)$ are the ith rows of $\mathbf{U}$ and $\mathbf{E}$, respectively. Row samples (3.48) are exactly the columns of the transposed data matrix $\mathbf{Y}^\top$ and so row samples have the same PCA amplitudes as column samples; principal components and score vectors swap.

In (3.48), noise heteroscedasticity is within each row sample $y^{(i)}$ rather than across row samples $y^{(1)}, \ldots, y^{(d)}$, and so one might think that the row samples could be equivalently generated from the generalized spiked covariance model (3.44) with a covariance similar to (3.45). However, the row samples are neither independent nor identically distributed; $\mathbf{U}$ induces dependence across rows as well as variety in their distributions. As a result, the row samples do not match the generalized spiked covariance model.

One could make $\mathbf{U}$ random according to the "i.i.d. model" of [22]. As noted in Remark 3.1, Theorem 3.4 still holds and the asymptotic PCA performance is unchanged. For such $\mathbf{U}$, the row samples $y^{(1)}, \ldots, y^{(d)}$ are now identically distributed but they are still not independent; dependence arises because $\mathbf{Z}$ is shared. To remove the dependence, one could make $\mathbf{Z}$ deterministic and also design it so that the row samples are iid with covariance matrix matching that of (3.44), but doing so no longer matches the inter-sample heteroscedastic model. It corresponds instead to having deterministic coefficients associated with a random subspace. Thus to analyze asymptotic PCA performance for row samples one should still use Theorem 3.4 rather than existing results for the generalized spiked covariance model [15, 215].

3.8.2 Additional properties

This section highlights a few additional properties of β_i, $B_i'(\beta_i)$ and $A(\beta_i)$ that lend deeper insight into how they vary with the noise variances $\sigma_1^2, \ldots, \sigma_L^2$.

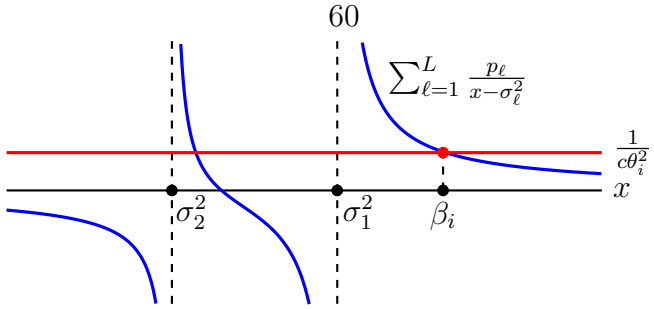


Figure 3.7: Location of the largest real root β_i of $B_i(x)$ for two noise variances $\sigma_1^2 = 2$ and $\sigma_2^2 = 0.75$, occurring in proportions $p_1 = 70\%$ and $p_2 = 30\%$, where the sample-to-dimension ratio is $c = 1$ and the subspace amplitude is $\theta_i = 1$.

3.8.2.1 Expressing $A(\beta_i)$ in terms of β_i and $B_i'(\beta_i)$

We can rewrite $A(\beta_i)$ in terms of β_i and $B_i'(\beta_i)$ as follows:

$$\begin{aligned}
A\left(\beta_i\right) &= 1 - c\sum_{\ell=1}^{L} \frac{p_\ell \sigma_\ell^4}{\left(\beta_i - \sigma_\ell^2\right)^2} = 1 - c\sum_{\ell=1}^{L} p_\ell \left\{1 - \frac{-2\beta_i\sigma_\ell^2 + \beta_i^2}{\left(\beta_i - \sigma_\ell^2\right)^2}\right\} \\
&= 1 - c\sum_{\ell=1}^{L} p_\ell \left\{1 - \frac{-2\beta_i\sigma_\ell^2 + 2\beta_i^2 - \beta_i^2}{\left(\beta_i - \sigma_\ell^2\right)^2}\right\} \\
&= 1 - c\sum_{\ell=1}^{L} p_\ell \left\{1 + \beta_i^2 \frac{1}{\left(\beta_i - \sigma_\ell^2\right)^2} - 2\beta_i \frac{1}{\beta_i - \sigma_\ell^2}\right\} \\
&= 1 - c\sum_{\ell=1}^{L} p_\ell - c\beta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{\left(\beta_i - \sigma_\ell^2\right)^2} + 2c\beta_i \sum_{\ell=1}^{L} \frac{p_\ell}{\beta_i - \sigma_\ell^2} \\
&= 1 - c - c\beta_i^2 \left\{\frac{1}{c\theta_i^2} B_i'\left(\beta_i\right)\right\} + 2c\beta_i \left\{\frac{1 - B_i\left(\beta_i\right)}{c\theta_i^2}\right\} \\
&= 1 - c - \frac{\beta_i}{\theta_i^2}\{\beta_i B_i'\left(\beta_i\right) - 2\}, \qquad (3.49)
\end{aligned}$$

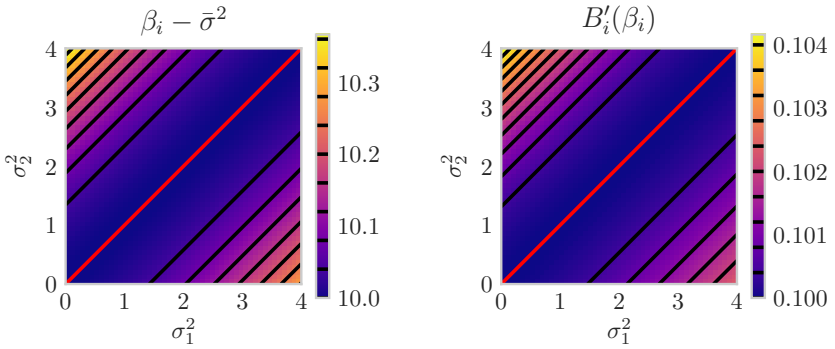
since $B_i(\beta_i) = 0$. Thus we focus on properties of β_i and $B_i'(\beta_i)$ for the remainder of Section 3.8.2; (3.49) relates them back to $A(\beta_i)$.

3.8.2.2 Graphical illustration of β_i

Note that β_i is the largest solution of

$$\frac{1}{c\theta_i^2} = \sum_{\ell=1}^{L} \frac{p_\ell}{x - \sigma_\ell^2}, \qquad (3.50)$$

because β_i is the largest real root of $B_i(x)$. Figure 3.7 illustrates (3.50) for two noise variances $\sigma_1^2 = 2$ and $\sigma_2^2 = 0.75$, occurring in proportions $p_1 = 70\%$ and $p_2 = 30\%$, where the sample-to-dimension ratio is $c = 1$ and the subspace amplitude



(a) $\beta_i - \bar{\sigma}^2$ over σ_1^2 and σ_2^2. (b) $B_i'(\beta_i)$ over σ_1^2 and σ_2^2.

Figure 3.8: Illustration of $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ as a function of two noise variances σ_1^2 and σ_2^2. The level curves are along lines parallel to $\sigma_1^2 = \sigma_2^2$ for all values of sample-to-dimension ratio c, proportions p_1 and p_2, and subspace amplitude θ_i

is $\theta_i = 1$. The plot is a graphical representation of β_i and gives a way to visualize the relationship between β_i and the model parameters. Observe, for example, that β_i is larger than all the noise variances and that increasing θ_i or c amounts to moving the horizontal red line down and tracking the location of the intersection.

3.8.2.3 Level curves

Figure 3.8 shows $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ as functions (implicitly) of $L = 2$ noise variances σ_1^2 and σ_2^2, where

$$\bar{\sigma}^2 = p_1\sigma_1^2 + \cdots + p_L\sigma_L^2$$

is the average noise variance. Figure 3.8 illustrates that lines parallel to the diagonal $\sigma_1^2 = \sigma_2^2$ are level curves for both $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$. This is a general phenomenon: lines parallel to the diagonal $\sigma_1^2 = \cdots = \sigma_L^2$ are level curves of both $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ for all sample-to-dimension ratios c, proportions $p_1, \ldots, p_L$ and subspace amplitudes θ_i.

To show this fact, note that $\beta_i - \bar{\sigma}^2$ is the largest real solution to

$$0 = B_i(x + \bar{\sigma}^2) = 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{x - (\sigma_\ell^2 - \bar{\sigma}^2)}, \tag{3.51}$$

because $0 = B_i(\beta_i)$. Changing the noise variances to $\sigma_1^2 + \Delta, \ldots, \sigma_L^2 + \Delta$ for some Δ also changes the average noise variance to $\bar{\sigma}^2 + \Delta$ and so $\sigma_\ell^2 - \bar{\sigma}^2$ remains unchanged. As a result, the solutions to (3.51) remain unchanged.

Similarly, note that

$$B_i'(\beta_i) = c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{(\beta_i - \sigma_\ell^2)^2} = c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{\{(\beta_i - \bar{\sigma}^2) - (\sigma_\ell^2 - \bar{\sigma}^2)\}^2} \tag{3.52}$$

remains unchanged when changing the noise variances to $\sigma_1^2 + \Delta, \ldots, \sigma_L^2 + \Delta$.

Thus we conclude from (3.51) and (3.52) that lines parallel to $\sigma_1^2 = \cdots = \sigma_L^2$ are level curves for both $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$. The line $\sigma_1^2 = \cdots = \sigma_L^2$ in particular minimizes the value of both, as was established in the proof of Theorem 3.9.

3.8.2.4 Hessians along the line $\sigma_1^2 = \cdots = \sigma_L^2$

We consider $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ as functions (implicitly) of the noise variances $\sigma_1^2, \ldots, \sigma_L^2$. To denote derivatives more clearly, we denote the ith noise variance as $v_i = \sigma_i^2$.

Written in this notation, we have

$$0 = 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{\beta_i - v_\ell}, \tag{3.53}$$

$$B_i'(\beta_i) = c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell}{(\beta_i - v_\ell)^2}. \tag{3.54}$$

Taking the total derivative of (3.53) with respect to v_s and v_t and solving for $\partial^2\beta_i/(\partial v_t \partial v_s)$ yields an initially complicated expression, but evaluating it on the line $v_1 = \cdots = v_L$ vastly simplifies it, yielding:

$$\frac{\partial^2(\beta_i - \bar{\sigma}^2)}{\partial v_t \partial v_s} = \frac{2}{c\theta_i^2}(p_s\delta_{s,t} - p_s p_t). \tag{3.55}$$

where $\delta_{s,t} = 1$ if $s = t$ and 0 otherwise. Notably, $\bar{\sigma}^2 = p_1 v_1 + \cdots + p_L v_L$ has zero Hessian everywhere.

Likewise, taking the total derivative of (3.54) with respect to v_s and v_t yields an initially complicated expression that is again vastly simplified by evaluating it on the line $v_1 = \cdots = v_L$, yielding:

$$\frac{\partial^2 B_i'(\beta_i)}{\partial v_t \partial v_s} = \frac{2}{(c\theta_i^2)^4}(p_s\delta_{s,t} - p_s p_t). \tag{3.56}$$

(3.55) and (3.56) show that the Hessian matrices for $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ are both scaled versions of the matrix

$$\mathbf{H} = \underbrace{\begin{bmatrix} p_1 & & \\ & \ddots & \\ & & p_L \end{bmatrix}}_{\mathrm{diag}(p)} - \underbrace{\begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \begin{bmatrix} p_1 & \cdots & p_L \end{bmatrix}}_{pp^\top} \tag{3.57}$$

on the line $v_1 = \cdots = v_L$. The (scaled) Hessian matrix (3.57) is a rank one perturbation by $-pp^\top$ of $\mathrm{diag}(p)$, and so its eigenvalues downward interlace with those of $\mathrm{diag}(p)$ (see Theorem 8.1.8 of [77]). Namely, $\mathbf{H}$ has eigenvalues $\lambda_1, \ldots, \lambda_L$ satisfying

$$\lambda_1 \leq p_{(1)} \leq \lambda_2 \leq \cdots \leq \lambda_L \leq p_{(L)},$$

where $p_{(1)}, \ldots, p_{(L)}$ are the proportions in increasing order. The vector $\mathbf{1}$ of all ones, i.e., the vector in the direction of $v_1 = \cdots = v_L$, is an eigenvector of $\mathbf{H}$ with eigenvalue zero; note that $\mathbf{H1} = \mathrm{diag}(p)\mathbf{1} - pp^\top\mathbf{1} = p - p = 0$. This eigenvalue is less than $p_{(1)} > 0$ and so $\lambda_1 = 0$ and $\lambda_2, \ldots, \lambda_L \geq p_{(1)} > 0$. Hence the Hessians of $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ are both zero in the direction of the line $v_1 = \cdots = v_L$ and positive definite in other directions. This property provides deeper insight into the fact that $\beta_i - \bar{\sigma}^2$ and $B_i'(\beta_i)$ are minimized on the line $\sigma_1^2 = \cdots = \sigma_L^2$, as was established in the proof of Theorem 3.9.

3.8.3 Impact of parameters: amplitude and coefficient recovery

Section 3.3 discusses how the asymptotic subspace recovery (3.4) of Theorem 3.4 depends on the model parameters: sample-to-dimension ratio c, subspace amplitudes $\theta_1, \ldots, \theta_k$, proportions $p_1, \ldots, p_L$ and noise variances $\sigma_1^2, \ldots, \sigma_L^2$. This section shows that the same phenomena occur for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5). For the asymptotic PCA amplitudes, we consider the ratio $\hat{\theta}_i^2/\theta_i^2$. As discussed in Remark 3.6, the asymptotic PCA amplitude $\hat{\theta}_i$ is positively biased relative to the subspace amplitude θ_i, and so the almost sure limit of $\hat{\theta}_i^2/\theta_i^2$ is greater than one, with larger values indicating more bias.

3.8.3.1 Impact of sample-to-dimension ratio c and subspace amplitude θ_i

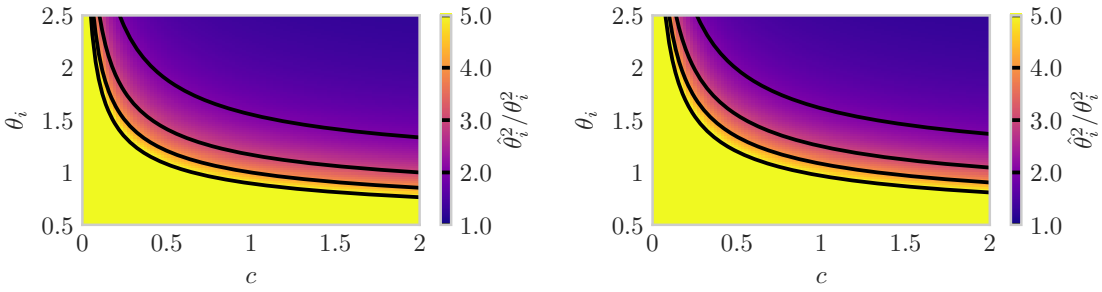
As in Section 3.3.1, we vary the sample-to-dimension ratio c and subspace amplitude θ_i in two scenarios:

a) there is only one noise variance fixed at $\sigma_1^2 = 1$

b) there are two noise variances $\sigma_1^2 = 0.8$ and $\sigma_2^2 = 1.8$ occurring in proportions $p_1 = 80\%$ and $p_2 = 20\%$.

Both scenarios have average noise variance 1. Figures 3.9 and 3.10 show analogous plots to Figure 3.1 but for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5), respectively.

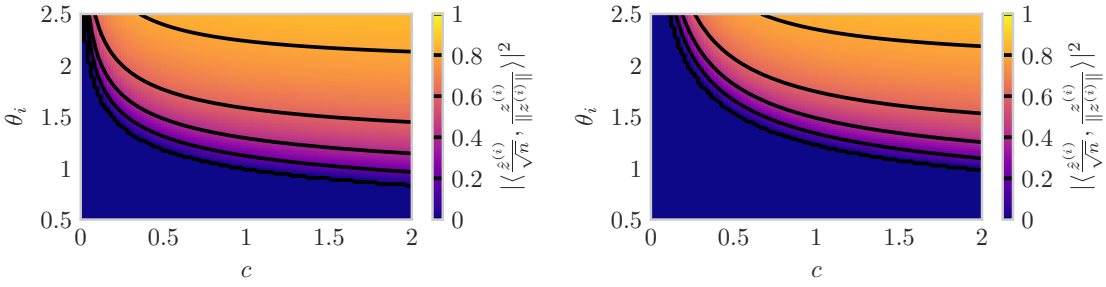
As was the case for Figure 3.1 in Section 3.3.1, decreasing the subspace amplitude θ_i degrades both the asymptotic amplitude performance (i.e., increases bias) shown in Figure 3.9 and the asymptotic coefficient recovery shown in Figure 3.10, but the lost



(a) Homoscedastic noise with $\sigma_1^2 = 1$.

(b) Heteroscedastic noise with $p_1 = 80\%$ of samples at $\sigma_1^2 = 0.8$ and $p_2 = 20\%$ of samples at $\sigma_2^2 = 1.8$.

Figure 3.9: Asymptotic amplitude bias (3.2) of the ith PCA amplitude as a function of sample-to-dimension ratio c and subspace amplitude θ_i with average noise variance equal to one. Contours are overlaid in black. The contours in (b) are slightly further up and to the right than in (a); more samples are needed to reduce the positive bias.

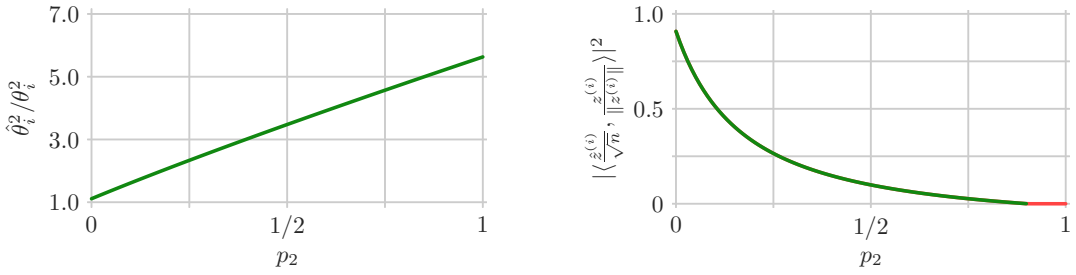


(a) Homoscedastic noise with $\sigma_1^2 = 1$.

(b) Heteroscedastic noise with $p_1 = 80\%$ of samples at $\sigma_1^2 = 0.8$ and $p_2 = 20\%$ of samples at $\sigma_2^2 = 1.8$.

Figure 3.10: Asymptotic coefficient recovery (3.5) of the ith score vector as a function of sample-to-dimension ratio c and subspace amplitude θ_i with average noise variance equal to one. Contours are overlaid in black and the region where $A(\beta_i) \leq 0$ is shown as zero (the prediction of Conjecture 3.5). The phase transition in (b) is further right than in (a); more samples are needed to recover the same strength signal.

performance could be regained by increasing the number of samples. Furthermore, both the asymptotic amplitude performance shown in Figure 3.9 and the asymptotic coefficient recovery shown in Figure 3.10 decline when the noise is heteroscedastic. Though the difference is subtle for the asymptotic amplitude bias, the contours move up and to the right in both cases. This degradation is consistent with Theorem 3.9; PCA performs worse on heteroscedastic data than it does on homoscedastic data of



(a) Asymptotic amplitude bias (3.2).

(b) Asymptotic coefficient recovery (3.5).

Figure 3.11: Asymptotic amplitude bias (3.2) and coefficient recovery (3.5) of the ith PCA amplitude and score vector as functions of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. The region where $A(\beta_i) \leq 0$ is the red horizontal segment in (b) with value zero (the prediction of Conjecture 3.5).

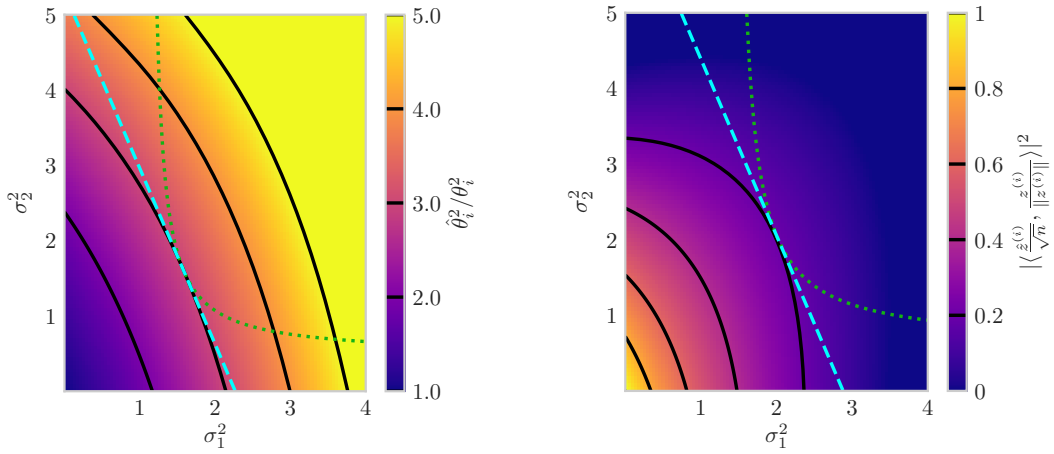
the same average noise variance and more samples or a larger subspace amplitude are needed to compensate.

3.8.3.2 Impact of proportions $p_1, \ldots, p_L$

As in Section 3.3.2, we consider two noise variances $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 3.25$ occurring in proportions $p_1 = 1 - p_2$ and p_2, where the sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. Figure 3.11 shows analogous plots to Figure 3.2 but for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5). As was the case for Figure 3.2 in Section 3.3.2, performance generally degrades in Figure 3.11 as p_2 increases and low noise samples with noise variance σ_1^2 are traded for high noise samples with noise variance σ_2^2. The performance is best when $p_2 = 0$ and all the samples have the smaller noise variance σ_1^2, i.e., there is no contamination.

3.8.3.3 Impact of noise variances $\sigma_1^2, \ldots, \sigma_L^2$

As in Section 3.3.3, we consider two noise variances σ_1^2 and σ_2^2 occurring in proportions $p_1 = 70\%$ and $p_2 = 30\%$, where the sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. Figure 3.12 shows analogous plots to Figure 3.3 but for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5). As was the case for Figure 3.3 in Section 3.3.3, performance typically degrades with increasing noise variances. The contours in Figure 3.12b are also generally horizontal for small σ_1^2 and vertical for small σ_2^2. They indicate that when the gap between the two largest noise variances is "sufficiently" wide, the asymptotic coefficient recov-



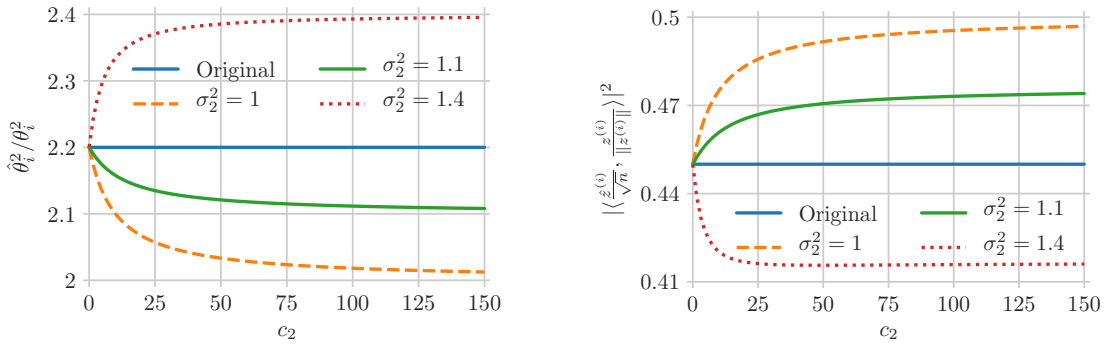
(a) Asymptotic amplitude bias (3.2). (b) Asymptotic coefficient recovery (3.5).

Figure 3.12: Asymptotic amplitude bias (3.2) and coefficient recovery (3.5) of the ith PCA amplitude and score vector as functions of noise variances σ_1^2 and σ_2^2 occurring in proportions $p_1 = 70\%$ and $p_2 = 30\%$, where the sample-to-dimension ratio is $c = 10$ and the subspace amplitude is $\theta_i = 1$. Contours are overlaid in black and the region where $A(\beta_i) \leq 0$ is shown as zero in (b), matching the prediction of Conjecture 3.5. Along each dashed cyan line, the average noise variance is fixed and the best performance occurs when $\sigma_1^2 = \sigma_2^2 = \bar{\sigma}^2$. Along each dotted green curve, the average inverse noise variance is fixed and the best performance again occurs when $\sigma_1^2 = \sigma_2^2$.

ery is roughly determined by the largest noise variance. This property mirrors the asymptotic subspace recovery and occurs for similar reasons, discussed in detail in Section 3.3.3. Along each dashed cyan line in Figure 3.12, the average noise variance is fixed and the best performance for both the PCA amplitudes and coefficient recovery again occurs when $\sigma_1^2 = \sigma_2^2 = \bar{\sigma}^2$, as was predicted by Theorem 3.9. Along each dotted green curve in Figure 3.12, the average inverse noise variance is fixed and the best performance for both the PCA amplitudes and coefficient recovery again occurs when $\sigma_1^2 = \sigma_2^2$, as was predicted in Remark 3.11.

3.8.3.4 Impact of adding data

As in Section 3.3.4, we consider adding data with noise variance σ_2^2 and sample-to-dimension ratio c_2 to an existing dataset that has noise variance $\sigma_1^2 = 1$, sample-to-dimension ratio $c_1 = 10$ and subspace amplitude $\theta_i = 1$ for the ith component. The combined dataset has a sample-to-dimension ratio of $c = c_1 + c_2$ and is potentially heteroscedastic with noise variances σ_1^2 and σ_2^2 appearing in proportions $p_1 = c_1/c$



(a) Asymptotic amplitude bias (3.2).

(b) Asymptotic coefficient recovery (3.5).

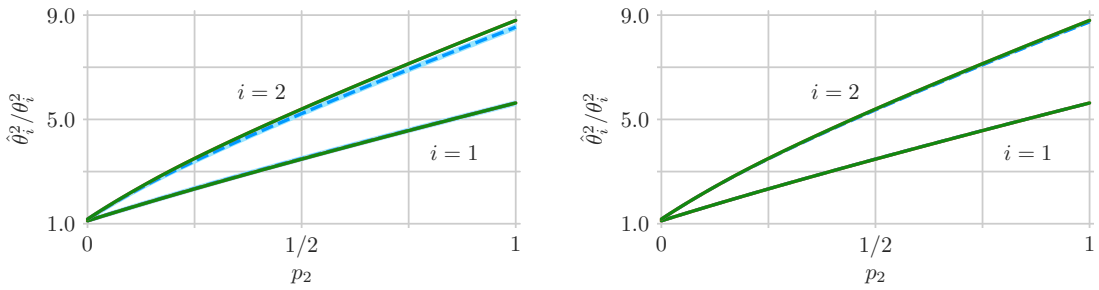
Figure 3.13: Asymptotic amplitude bias (3.2) and coefficient recovery (3.5) of the ith PCA amplitude and score vector for samples added with noise variance σ_2^2 and samples-per-dimension c_2 to an existing dataset with noise variance $\sigma_1^2 = 1$, sample-to-dimension ratio $c_1 = 10$ and subspace amplitude $\theta_i = 1$.

and $p_2 = c_2/c$.

Figure 3.13 shows analogous plots to Figure 3.4 in Section 3.3.4 but for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5). As was the case for Figure 3.4, the dashed orange curves show the recovery when $\sigma_2^2 = 1 = \sigma_1^2$ and illustrate the benefit we would expect for homoscedastic data: increasing the samples per dimension improves recovery. The green curves show the performance when $\sigma_2^2 = 1.1 > \sigma_1^2$; as before, these samples are "slightly" noisier and performance improves for any number added. Finally, the dotted red curves show the performance when $\sigma_2^2 = 1.4 > \sigma_1^2$. As before, performance degrades when adding a small number of these noisier samples. However, unlike subspace recovery, performance degrades when adding any amount of these samples. In the limit $c_2 \to \infty$, the asymptotic amplitude bias is $1 + \sigma_2^2/\theta_i^2$ and the asymptotic coefficient recovery is $1/(1 + \sigma_2^2/\theta_i^2)$; neither has perfect recovery in the limit when added samples are noisy.

3.8.4 Numerical simulation: amplitude and coefficient recovery

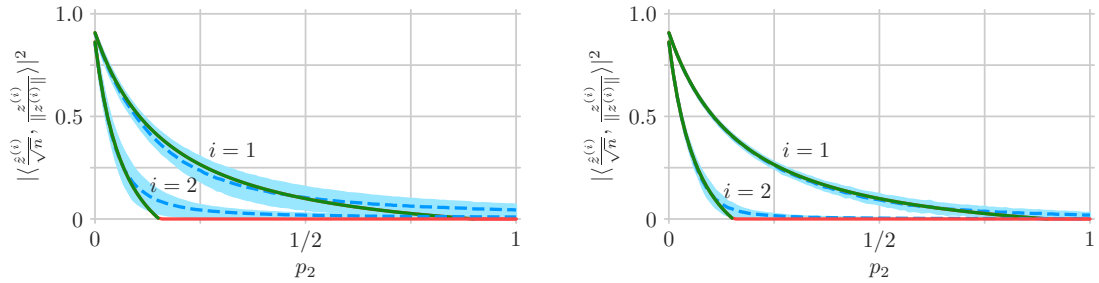
Section 3.4 shows that the asymptotic subspace recovery (3.4) of Theorem 3.4 is meaningful for practical settings with finitely many samples in a finite-dimensional space. This section shows the same for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5). For the asymptotic PCA amplitudes, we again consider the ratio $\hat{\theta}_i^2/\theta_i^2$. As discussed in Remark 3.6, the asymptotic PCA amplitude $\hat{\theta}_i$ is positively biased relative to the subspace amplitude θ_i, and so the almost sure limit of $\hat{\theta}_i^2/\theta_i^2$ is greater than one, with larger values indicating more bias.



(a) 10^3 samples in 10^2 dimensions.

(b) 10^4 samples in 10^3 dimensions.

Figure 3.14: Simulated amplitude bias (3.2) as a function of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitudes are $\theta_1 = 1$ and $\theta_2 = 0.8$. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic bias (3.2) of Theorem 3.4 (green curve). Increasing data size from (a) to (b) results in even smaller interquartile intervals, indicating concentration to the mean, which is converging to the asymptotic bias.



(a) 10^3 samples in 10^2 dimensions.

(b) 10^4 samples in 10^3 dimensions.

Figure 3.15: Simulated coefficient recovery (3.5) as a function of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitudes are $\theta_1 = 1$ and $\theta_2 = 0.8$. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic recovery (3.5) of Theorem 3.4 (green curve). The region where $A(\beta_i) \leq 0$ is the red horizontal segment with value zero (the prediction of Conjecture 3.5). Increasing data size from (a) to (b) results in smaller interquartile intervals, indicating concentration to the mean, which is converging to the asymptotic recovery.

As in Section 3.4, this section simulates data according to the model described in Section 3.2.1 for a two-dimensional subspace with subspace amplitudes $\theta_1 = 1$ and $\theta_2 = 0.8$, two noise variances $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 3.25$, and a sample-to-dimension ratio of $c = 10$. We sweep the proportion of high noise points p_2 from zero to one, setting $p_1 = 1 - p_2$ as in Section 3.4. The first simulation considers $n = 10^3$ samples in a $d = 10^2$ dimensional ambient space (10^4 trials). The second increases these to $n = 10^4$ samples in a $d = 10^3$ dimensional ambient space (10^3 trials). All simulations generate data from the standard normal distribution, i.e., $z_{ij}, \varepsilon_{ij} \sim \mathcal{N}(0, 1)$. Figures 3.14 and 3.15 show analogous plots to Figure 3.5 but for the asymptotic PCA amplitudes (3.2) and coefficient recovery (3.5), respectively.

As was the case for Figure 3.5 in Section 3.4, both Figures 3.14 and 3.15 illustrate the following general observations:

a) the simulation mean and almost sure limit generally agree in the smaller simulation of 10^3 samples in a 10^2 dimensional ambient space

b) the smooth simulation mean deviates from the non-smooth almost sure limit near the phase transition

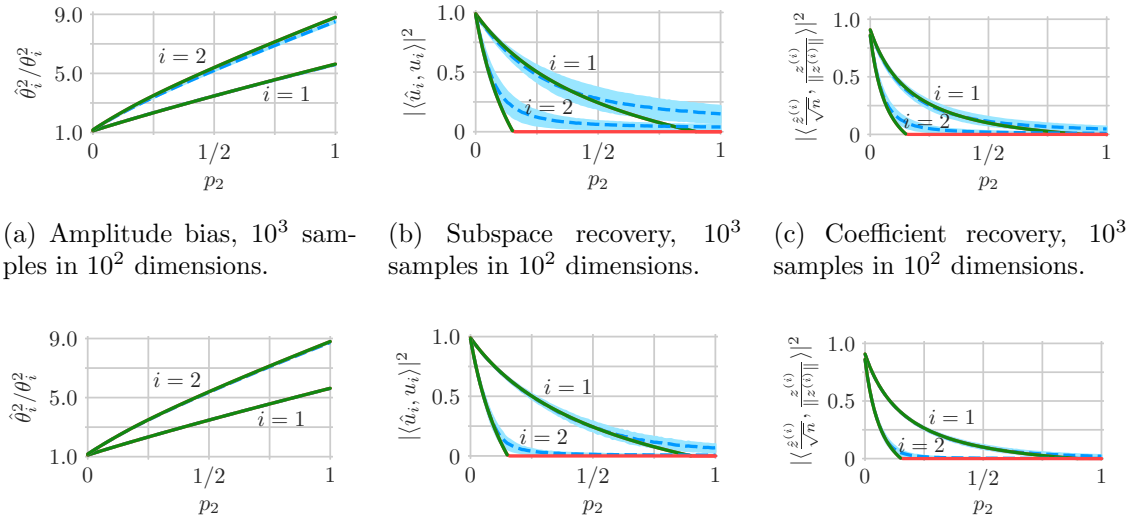
c) the simulation mean and almost sure limit agree better for the larger simulation of 10^4 samples in a 10^3 dimensional ambient space

d) the interquartile intervals for the larger simulations are roughly half the size of those in the smaller simulations, indicating concentration to the means.

In fact, the amplitude bias in Figure 3.14 and the coefficient recovery in Figure 3.15 both have significantly better agreement with their almost sure limits than the subspace recovery in Figure 3.5 has with its almost sure limit. The amplitude bias in Figure 3.14, in particular, is tightly concentrated around its almost sure limit (3.2). Furthermore, Figure 3.15 demonstrates good agreement with Conjecture 3.5, providing evidence that there is indeed a phase transition below which the coefficients are also not recovered.

3.8.5 Additional numerical simulations

Section 3.4 and Section 3.8.4 provide numerical simulation results for real-valued data generated using normal distributions. This section illustrates the generality of the model in Section 3.2.1 by showing analogous simulation results for circularly symmetric complex normal data in Figure 3.16 and for a mixture of Gaussians in Figure 3.17. As before, we show the results of two simulations for each setting. The first simulation considers $n = 10^3$ samples in a $d = 10^2$ dimensional ambient space



(a) Amplitude bias, 10^3 samples in 10^2 dimensions.

(b) Subspace recovery, 10^3 samples in 10^2 dimensions.

(c) Coefficient recovery, 10^3 samples in 10^2 dimensions.

(d) Amplitude bias, 10^4 samples in 10^3 dimensions.

(e) Subspace recovery, 10^4 samples in 10^3 dimensions.

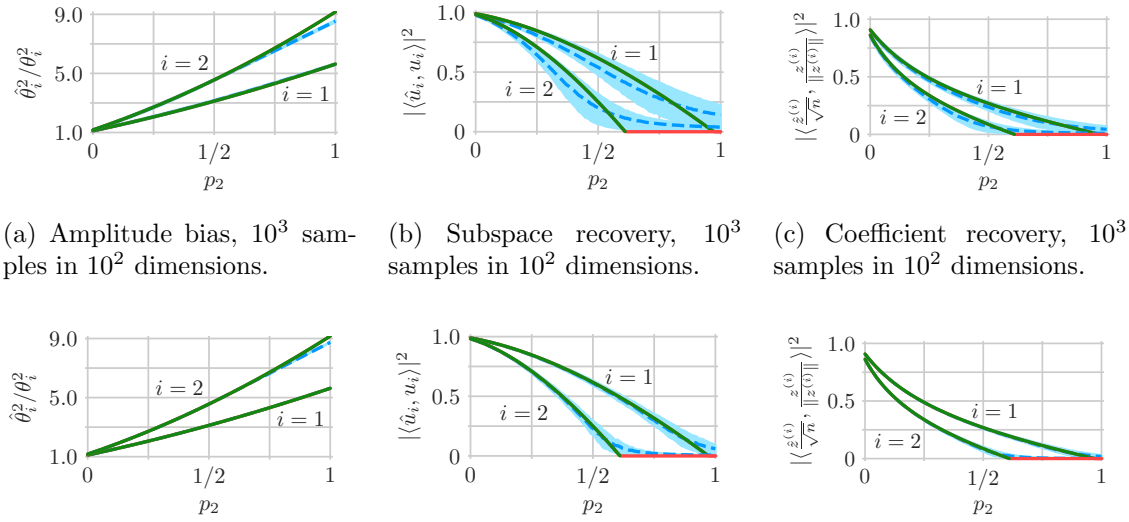
(f) Coefficient recovery, 10^4 samples in 10^3 dimensions.

Figure 3.16: Simulated complex-normal PCA performance as a function of the contamination fraction p_2, the proportion of samples with noise variance $\sigma_2^2 = 3.25$, where the other noise variance $\sigma_1^2 = 0.1$ occurs in proportion $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitudes are $\theta_1 = 1$ and $\theta_2 = 0.8$. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the almost sure limits of Theorem 3.4 (green curve). The region where $A(\beta_i) \leq 0$ is shown as red horizontal segments with value zero (the prediction of Conjecture 3.5).

(10^4 trials). The second increases these to $n = 10^4$ samples in a $d = 10^3$ dimensional ambient space (10^3 trials).

Figure 3.16 mirrors Sections 3.4 and 3.8.4 and simulates data according to the model described in Section 3.2.1 for a two-dimensional subspace with subspace amplitudes $\theta_1 = 1$ and $\theta_2 = 0.8$, two noise variances $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 3.25$, and a sample-to-dimension ratio of $c = 10$. We again sweep the proportion of high noise points p_2 from zero to one, setting $p_1 = 1 - p_2$. The only difference is that Figure 3.16 generates data from the standard *complex* normal distribution, i.e., $z_{ij}, \varepsilon_{ij} \sim \mathcal{CN}(0, 1)$.

Figure 3.17 instead simulates a *homoscedastic* setting of the model described in Section 3.2.1 over a range of noise distributions, all *mixtures* of Gaussians. As before, we consider a two-dimensional subspace with subspace amplitudes $\theta_1 = 1$ and $\theta_2 = 0.8$, and a sample-to-dimension ratio of $c = 10$. Figure 3.17 generates coefficients $z_{ij} \sim \mathcal{N}(0, 1)$ from the standard normal distribution and generates noise



(a) Amplitude bias, 10^3 samples in 10^2 dimensions.

(b) Subspace recovery, 10^3 samples in 10^2 dimensions.

(c) Coefficient recovery, 10^3 samples in 10^2 dimensions.

(d) Amplitude bias, 10^4 samples in 10^3 dimensions.

(e) Subspace recovery, 10^4 samples in 10^3 dimensions.

(f) Coefficient recovery, 10^4 samples in 10^3 dimensions.

Figure 3.17: Simulated mixture model PCA performance as a function of the mixture probability p_2, the probability that a scaled noise entry $\eta_i \varepsilon_{ij}$ is Gaussian with variance $\lambda_2^2 = 3.25$, where it is Gaussian with variance $\lambda_1^2 = 0.1$ otherwise, i.e., with probability $p_1 = 1 - p_2$. The sample-to-dimension ratio is $c = 10$ and the subspace amplitudes are $\theta_1 = 1$ and $\theta_2 = 0.8$. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the almost sure limits of Theorem 3.4 (green curve). The region where $A(\beta_i) \leq 0$ is shown as red horizontal segments with value zero (the prediction of Conjecture 3.5).

entries ε_{ij} from the Gaussian mixture model

$$\varepsilon_{ij} \sim \begin{cases} \mathcal{N}(0, \lambda_1^2/\sigma^2) & \text{with probability } p_1, \\ \mathcal{N}(0, \lambda_2^2/\sigma^2) & \text{with probability } p_2, \end{cases}$$

where $\lambda_1^2 = 0.1$ and $\lambda_2^2 = 3.25$, and the *single* noise variance is set to

$$\sigma^2 = p_1 \lambda_1^2 + p_2 \lambda_2^2. \tag{3.58}$$

Each scaled noise entry $\eta_i \varepsilon_{ij} = \sigma \varepsilon_{ij}$ is a mixture of two Gaussian distributions with variances λ_1^2 and λ_2^2. We sweep the mixture probability p_2 from zero to one, setting $p_1 = 1 - p_2$. Thus, Figure 3.17 illustrates performance over a range of noise distributions. The noise variance (3.58) in Figure 3.17 matches the average noise variance in Figure 3.16 as we sweep p_2. However, Figures 3.17 and 3.16 differ because Figure 3.17 simulates a *homoscedastic* setting while Figure 3.16 simulates a *heteroscedastic* setting. Figure 3.17 also differs from Figure 3.6b that simulates data

from the random inter-sample heteroscedastic model of Section 3.8.1.1. While both simulate (scaled) noise from a mixture model, scaled noise entries $\eta_i \varepsilon_{ij}$ in Figure 3.17 are all iid. Scaled noise entries $\eta_i \varepsilon_{ij}$ in the random inter-sample heteroscedastic model are independent only across samples; they are *not* independent within each sample. Figure 3.17 is instead more like Figure 3.6c that simulates data from the Johnstone spiked covariance model. See Section 3.8.1.1 for a comparison of these models.

As was the case for (real-valued) standard normal data in Sections 3.4 and 3.8.4, Figures 3.16 and 3.17 illustrate the following general observations:

a) the simulation means and almost sure limits generally agree in the smaller simulations of 10^3 samples in a 10^2 dimensional ambient space

b) the smooth simulation means deviate from the non-smooth almost sure limits near the phase transitions

c) the simulation means and almost sure limits agree better for the larger simulations of 10^4 samples in a 10^3 dimensional ambient space

d) the interquartile intervals for the larger simulations are roughly half the size of those in the smaller simulations, indicating concentration to the means.

The agreement between simulations and almost sure limits demonstrated in both Figures 3.16 and 3.17 highlights the generality of the model considered in this chapter: it allows for both complex-valued data and non-Gaussian distributions. In both cases, the asymptotic results of Theorem 3.4 remain meaningful for practical settings with finitely many samples in a finite-dimensional space.

CHAPTER IV

Optimally weighted PCA for high-dimensional heteroscedastic data

As the analysis of Chapter III quantified, PCA does not robustly recover underlying components in the presence of heteroscedastic noise. Specifically, PCA suffers from treating all data samples as if they are equally informative. This chapter generalizes the analysis of Chapter III to characterize a weighted variant of PCA that can account for heteroscedasticity by giving samples with larger noise variance less influence. The analysis provides expressions for the asymptotic recovery of underlying low-dimensional components for any choice of weights. Surprisingly, it turns out that whitening the noise by using inverse noise variance weights is suboptimal. We derive optimal weights, characterize the performance of weighted PCA, and consider the problem of optimally collecting samples under budget constraints. The work in this chapter led to the following submitted journal paper that this chapter presents:

> [96] David Hong, Jeffrey A. Fessler, and Laura Balzano. Optimally Weighted PCA for High-Dimensional Heteroscedastic Data, 2018. Submitted. arXiv: 1810.12862v2.

4.1 Introduction

We consider a sample-weighted PCA [114, Section 14.2.1] to account for heteroscedastic noise in the data; giving noisier samples smaller weights reduces their influence. Sample-weighted PCA (WPCA) replaces the sample covariance matrix with a weighted sample covariance matrix $(\omega_1 y_1 y_1^{\mathsf{H}} + \cdots + \omega_n y_n y_n^{\mathsf{H}})/n$ where $y_1, \ldots, y_n \in \mathbb{C}^d$ are zero-mean sample vectors, $\omega_1, \ldots, \omega_n \geq 0$ are the weights, and the superscript H denotes Hermitian transpose. As in PCA, the principal components[1] $\hat{u}_1, \ldots, \hat{u}_k \in \mathbb{C}^d$

[1] As in Section 2.2, "principal components" here refers to eigenvectors of the (weighted) sample covariance matrix and "scores" refers to the derived variables, i.e., the coefficients of the samples

and amplitudes $\hat{\theta}_1^2, \dots, \hat{\theta}_k^2$ are the first k eigenvectors and eigenvalues, respectively, of the weighted sample covariance matrix. The scores $\hat{z}_1, \dots, \hat{z}_k \in \mathbb{C}^n$ are given by $\hat{z}_i = (1/\hat{\theta}_i)\{\hat{u}_i^{\mathsf{H}}(y_1, \dots, y_n)\}^{\mathsf{H}}$ for each $i \in \{1, \dots, k\}$. Taken together, the principal components, amplitudes and scores solve the weighted approximation problem

$$\min_{\substack{\tilde{u}_1,\dots,\tilde{u}_k \in \mathbb{C}^d \\ \tilde{\theta}_1,\dots,\tilde{\theta}_k \geq 0 \\ \tilde{z}_1,\dots,\tilde{z}_k \in \mathbb{C}^n}} \sum_{j=1}^{n} \omega_j^2 \left\| y_j - \sum_{i=1}^{k} \tilde{u}_i \tilde{\theta}_i \big(\tilde{z}_i^{(j)}\big)^* \right\|_2^2 \tag{4.1}$$
$$\text{such that } \tilde{u}_s^{\mathsf{H}} \tilde{u}_t = \delta_{st}, \ \tilde{z}_s^{\mathsf{H}} \mathbf{W}^2 \tilde{z}_t = n\delta_{st},$$

where $\mathbf{W} := \operatorname{diag}(\omega_1, \dots, \omega_n)$ is a diagonal matrix of weights, and $\delta_{st} = 1$ if $s = t$ and 0 otherwise. Namely, they form a truncated generalized singular value decomposition [79, Appendix A] of the data matrix $\mathbf{Y} := (y_1, \dots, y_n) \in \mathbb{C}^{d \times n}$ formed with samples as columns. Note that the scores $\hat{z}_1, \dots, \hat{z}_k$ are orthonormal with respect to the weighted Euclidean metric $\mathbf{W}^2$, and are not necessarily so with respect to the Euclidean metric. Reconstructed samples $\hat{x}_1, \dots, \hat{x}_n \in \mathbb{C}^d$ are formed for each $j \in \{1, \dots, n\}$ as

$$\hat{x}_j := \sum_{i=1}^{k} \hat{u}_i \hat{\theta}_i \big(\hat{z}_i^{(j)}\big)^*, \tag{4.2}$$

and are projections of the samples $y_1, \dots, y_n$ onto the principal component subspace, i.e., the span of $\hat{u}_1, \dots, \hat{u}_k$.

To use WPCA, one must first select weights. Some natural choices to consider for heteroscedastic data are:

- uniform weights $\omega_j^2 = 1$: standard (unweighted) PCA may be a natural choice when data are "nearly" homoscedastic, but its performance generally degrades with increasing heteroscedasticity as shown, e.g., in Theorem 3.9.
- binary weights $\omega_j^2 = 0$ for noisier samples and $\omega_j^2 = 1$ for the rest: excluding samples that are much noisier is both practical and natural, but how much noisier they need to be is not obvious. Our analysis quantifies when doing so is nearly optimal.
- inverse noise variance weights $\omega_j^2 = 1/\eta_j^2$ where η_j^2 is the jth sample noise variance: weighting by inverse noise variance whitens the noise, making it homoscedastic, and can be interpreted as a maximum likelihood weighting [218], but given that conventional PCA is not robust to gross outliers, e.g., from very noisy samples, it is uncertain whether inverse noise variance downweights such samples aggressively enough.

with respect to the components.

It has been unclear which, if any, of these three options should be chosen, but among them inverse noise variance weights generally appear most natural, especially when all noise variances are moderate. Surprisingly, our analysis shows that none of these options optimally recover underlying components when the data have heteroscedastic noise. In some cases, they are near optimal, and our analysis uncovers these regimes as well.

4.1.1 Contributions of this chapter

This chapter analyzes WPCA and characterizes, for any choice of weights, the asymptotic recovery of underlying components, amplitudes and scores from data samples with heteroscedastic noise (Theorem 4.3). The main technical challenge lies in characterizing the almost sure limit of a weighted resolvent (Lemma 4.9) to extend [22, Theorems 2.9-2.10] to account for the weights, and we use a convenient expansion to divide and tackle the problem (Section 4.11.2). We provide simplified expressions as we did in Chapter III that allow us to obtain insights into the performance of WPCA as well as optimize the weights for various types of recovery, and we derive a surprisingly simple closed-form expression (Theorem 4.10) for weights that optimally recover an underlying component of amplitude θ_i: $\omega_j^2 = 1/\{\eta_j^2(\theta_i^2 + \eta_j^2)\}$. Deriving optimal weights involves identifying and exploiting nontrivial structure in the simplified expressions to characterize the critical points of the asymptotic component recovery with respect to square inverse weights. The simplified expressions also allow us to find optimal strategies for collecting samples under budget constraints (Theorem 4.11). Finally, we investigate some cases where suboptimal weights may be practical and sufficient and study how weighting changes the ways that data properties, e.g., noise variances and number of samples, affect PCA performance.

4.1.2 Relationship to previous works

Jolliffe [114, Section 14.2.1] describes a more general WPCA; one may, for example, also weight the coordinates of each sample. Within-sample weighting is discussed in [55, Sections 5.4–5.5] to account for variables with differing noise variances; the weights are inverse noise variance and the authors note that it corresponds to maximum likelihood estimation for the factor analysis model [55, Equation (20)]. Weighting both across and within samples is proposed in [49, Equation (28)] for analyzing spectrophotometric data from scanning wavelength kinetics experiments. The weights in [49] are also inverse noise variance. Similar weighting is used in [194, Equation (1)] for analyzing photometric light curve data from astronomical studies, and in [108] for analyzing metabolomics data. Weighting data by their inverse noise

variance has been a recurring theme, but the resulting performance has not been studied in the high-dimensional regime. This chapter analyzes the high-dimensional asymptotic performance of general across-sample weighting in WPCA for noise with heteroscedasticity across samples. Generalizing the analysis of this chapter to heteroscedasticity that is both across and within samples with correspondingly general weighting is an interesting area of future work.

Weighted variants of PCA have also been applied to account for other heterogeneities in the data. Jolliffe [114, Section 14.2.1] surveys and discusses several such settings, and Yue and Tomoyasu [219, Sections 3–5] use weights to account for, among other aspects, the relative importance of variables. Weighted variants of PCA are also closely tied to the problem of computing weighted low-rank approximations of matrices; see, e.g., [189] and [202, Section 4.2], where weights are used to account for unobserved data or to denote relative importance. Understanding how to handle such heterogeneities is an exciting area for future work and will become increasingly important for big data inference from "messy" data.

Choosing uniform weights specializes WPCA to (unweighted) PCA, so the analysis here generalizes that of Chapter III. There we analyzed the asymptotic recovery of PCA and characterized the impact of heteroscedastic noise, showing, in particular, that PCA performance is always best (for fixed average noise variance) when the noise is homoscedastic. See Section 3.1.3 for a discussion of the many connections to previous analyses for homoscedastic noise, and Section 3.8.1 for a detailed discussion of connections to spiked covariance models.

Recent work [221] considers noise that is heteroscedastic within each sample, producing a non-uniform bias along the diagonal of the covariance matrix that skews its eigenvectors. To address this issue, they propose an algorithm called HeteroPCA that iteratively replaces the diagonal entries with those of the current estimate's low-rank approximation, and they show that it has minimax optimal rate for recovering the principal subspace. Dobriban, Leeb and Singer [59] also study a data model with noise heteroscedasticity within samples, but with the goal of optimally shrinking singular values to recover low-rank signals from linearly transformed data. In contrast to both these works, we seek to optimally weight samples in PCA to address noise with across-sample heteroscedasticity. Understanding if and how these various questions and techniques relate is an interesting area for future investigation.

4.1.3 Organization of the chapter

Section 4.2 describes the model we consider for underlying components in heteroscedastic noise, and Section 4.3 states the main analysis result (Theorem 4.3): expressions for asymptotic WPCA recovery. Section 4.4 outlines its proof. Sec-

tion 4.5 derives optimal weights for component recovery (Theorem 4.10), and Section 4.6 discusses the suboptimality, or in some cases, the near optimality, of other choices. Section 4.7 illustrates the ways weighting affects how recovery depends on the data parameters. Section 4.8 derives optimal sampling strategies under budget constraints (Theorem 4.11). Section 4.9 illustrates in simulation how the asymptotic predictions compare to the empirical performance of WPCA for various problem sizes. Section 4.11 contains detailed proofs and additional simulations, and code for reproducing the figures in this chapter can be found online at: https://gitlab.com/dahong/optimally-weighted-pca-heteroscedastic-data

4.2 Model for heteroscedastic data

As in Section 3.2.1, we model n sample vectors $y_1, \dots, y_n \in \mathbb{C}^d$ as

$$y_j = \underbrace{\sum_{i=1}^{k} u_i \theta_i \big(z_i^{(j)}\big)^*}_{x_j \in \mathbb{C}^d} + \eta_j \varepsilon_j = x_j + \eta_j \varepsilon_j. \tag{4.3}$$

The following are deterministic:

- $u_1, \dots, u_k \in \mathbb{C}^d$ are orthonormal components,
- $\theta_1, \dots, \theta_k > 0$ are amplitudes,
- $\eta_j \in \{\sigma_1, \dots, \sigma_L\}$ are each one of L noise standard deviations $\sigma_1, \dots, \sigma_L$,

and we define n_1 to be the number of samples with $\eta_j = \sigma_1$, n_2 to be the number of samples with $\eta_j = \sigma_2$, and so on, where $n_1 + \cdots + n_L = n$.

The following are random:

- $z_1, \dots, z_k \in \mathbb{C}^n$ are iid score vectors whose entries are iid with mean $\mathbb{E}(z_i^{(j)}) = 0$, variance $\mathbb{E}|z_i^{(j)}|^2 = 1$, and a distribution satisfying a log-Sobolev inequality [9, Section 2.3.2],
- $\varepsilon_j \in \mathbb{C}^d$ are unitarily invariant iid noise vectors that have iid entries with mean $\mathbb{E}(\varepsilon_j^{(s)}) = 0$, variance $\mathbb{E}|\varepsilon_j^{(s)}|^2 = 1$ and bounded fourth moment $\mathbb{E}|\varepsilon_j^{(s)}|^4 < \infty$.

In words, (4.3) models data samples as containing k underlying components with additive mean zero heteroscedastic noise. Without loss of generality, we further assume that the weights correspond to the noise variances, that is, samples with noise variance $\eta_j^2 = \sigma_1^2$ are weighted as $\omega_j^2 = w_1^2$, and so on.

Remark 4.1 (Unitary invariance). Unitarily invariant noise means that left multiplication of each noise vector ε_j by any unitary matrix does not affect the joint distribution of its entries. As in Section 3.2.1, this assumption can be removed if the set of components $u_1, \ldots, u_k$ is isotropically drawn at random as in [22, Section 2.1].

Remark 4.2 (Example distributions). The conditions above are all satisfied when the entries $z_i^{(j)}$ and $\varepsilon_j^{(s)}$ are, for example, circularly symmetric complex normal $\mathcal{CN}(0,1)$. Rademacher random variables (i.e., ± 1 with equal probability) are another choice for scores $z_i^{(j)}$. Only circularly symmetric complex normal distributions satisfy all the noise conditions,[2] but as noted in Remark 4.1, unitary invariance can be removed if the components are random.

4.3 Asymptotic performance of weighted PCA

The following theorem quantifies how well the weighted PCA estimates $\hat{u}_1, \ldots, \hat{u}_k$, $\hat{\theta}_1, \ldots, \hat{\theta}_k$, and $\hat{z}_1, \ldots, \hat{z}_k$ recover the underlying components $u_1, \ldots, u_k$, amplitudes $\theta_1, \ldots, \theta_k$, and scores $z_1, \ldots, z_k$, from (4.3) as a function of:

- limiting sample-to-dimension ratio $n/d \to c > 0$,
- underlying amplitudes $\theta_1, \ldots, \theta_k$,
- noise variances $\sigma_1^2, \ldots, \sigma_L^2$,
- weights $w_1^2, \ldots, w_L^2$, and
- limiting proportions $n_1/n \to p_1, \ldots, n_L/n \to p_L$.

The expressions enable us to later study the behavior of weighted PCA and to optimize the weights.

Theorem 4.3 (Asymptotic recovery of amplitudes, components, and scores)**.** *Suppose the sample-to-dimension ratio $n/d \to c > 0$ and the noise variance proportions $n_\ell/n \to p_\ell$ for $\ell = 1, \ldots, L$ as $n, d \to \infty$. Then the ith WPCA amplitude $\hat{\theta}_i$ converges as*

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \frac{1}{c} \max(\alpha, \beta_i)\, C(\max(\alpha, \beta_i)) =: r_i^{(\theta)}, \tag{4.4}$$

[2] Gaussianity follows from orthogonal invariance via the Herschel-Maxwell theorem [33, Theorem 0.0.1] for real-valued random vectors. Its extension to complex-valued random vectors can be shown by observing that unitary invariance implies orthogonal invariance of its real part and circular symmetry of each entry in the complex plane.

where α and β_i are, respectively, the largest real roots of

$$A(x) := 1 - c\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^4 \sigma_\ell^4}{(x - w_\ell^2\sigma_\ell^2)^2}, \qquad B_i(x) := 1 - c\theta_i^2 \sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{x - w_\ell^2\sigma_\ell^2}, \tag{4.5}$$

and where

$$C(x) := 1 + c\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2 \sigma_\ell^2}{x - w_\ell^2\sigma_\ell^2}. \tag{4.6}$$

Furthermore, if $A(\beta_i) > 0$ then the ith component $\hat{u}_i$ has asymptotic recovery

$$\sum_{j:\theta_j=\theta_i} |\langle \hat{u}_i, u_j\rangle|^2 \xrightarrow{\text{a.s.}} \frac{1}{\beta_i}\frac{A(\beta_i)}{B_i'(\beta_i)} =: r_i^{(u)}, \qquad \sum_{j:\theta_j\neq\theta_i} |\langle \hat{u}_i, u_j\rangle|^2 \xrightarrow{\text{a.s.}} 0, \tag{4.7}$$

the normalized ith score $\hat{z}_i/\sqrt{n}$ has asymptotic weighted recovery

$$\sum_{j:\theta_j=\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} \frac{1}{c\theta_i^2 C(\beta_i)}\frac{A(\beta_i)}{B_i'(\beta_i)} =: r_i^{(z)}, \tag{4.8}$$

$$\sum_{j:\theta_j\neq\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} 0,$$

and

$$\sum_{j:\theta_j=\theta_i} \langle \hat{u}_i, u_j\rangle \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2}^{*} \xrightarrow{\text{a.s.}} \sqrt{r_i^{(u)} r_i^{(z)}} = \frac{1}{\sqrt{c\theta_i^2\beta_i C(\beta_i)}}\frac{A(\beta_i)}{B_i'(\beta_i)}. \tag{4.9}$$

Section 4.4 outlines the proof of Theorem 4.3 with the details deferred to Section 4.11.1. An overall roadmap is as follows: a) analyze almost sure limits of two key normalized traces, b) extend [22, Theorems 2.9-2.10] using these limits to account for weighting, then c) simplify the resulting expressions. Among other challenges, the fact that weights are associated with specific samples complicates the analysis.

Remark 4.4 (Location of the largest real roots). Finding the largest real roots of the univariate rational functions $A(x)$ and $B_i(x)$ is the most challenging aspect of computing the expressions in Theorem 4.3, but they can be found efficiently, e.g., with bisection, by observing that they are the only roots larger than the largest pole $\max_\ell(w_\ell^2\sigma_\ell^2)$ as shown in Fig. 4.1.

Remark 4.5 (Scaling properties for the weights). Scaling all the weights $w_1^2, \ldots, w_L^2$ does not affect the relative influence given to samples, and as a result, doing so only scales the WPCA amplitudes and scores. Theorem 4.3 reflects this scaling property of WPCA. Scaling all the weights by a constant λ, scales β_i by λ. As a result, $A(\beta_i)$ and $C(\beta_i)$ are unchanged, and $B_i'(\beta_i)$ is scaled by $1/\lambda$. Thus, as expected, the asymptotic component recovery (4.7) is unchanged, and the asymptotic amplitude (4.4) and asymptotic weighted score recovery (4.8) are both scaled by λ.

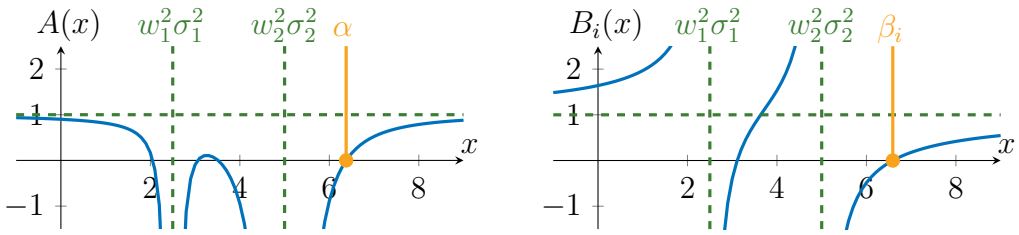


Figure 4.1: Location of the largest real roots α and β_i of A and B_i, respectively, for $c = 0.1$ samples per dimension, underlying amplitude $\theta_i^2 = 16$, $p_1 = 25\%$ of samples having noise variance $\sigma_1^2 = 1$ and weight $w_1^2 = 2.5$, and $p_2 = 75\%$ of samples having noise variance $\sigma_2^2 = 5$ and weight $w_2^2 = 1$.

4.3.1 Special cases: uniform, binary, and inverse noise variance weights

Uniform weights $w_\ell^2 = 1$ correspond to unweighted PCA, and binary weights $w_\ell^2 \in \{0, 1\}$ correspond to unweighted PCA carried out on only samples with nonzero weight. As a result, the analysis of unweighted PCA in Section 3.2 applies to uniform and binary weights. Theorem 4.3 specializes exactly to Theorem 3.4 for these weights. As shown in Section 3.2.6, the performance with these weights degrades (for both fixed average noise variance and for fixed average inverse noise variance) when the noise is heteroscedastic among the samples used. Binary weights can be chosen to use only samples with the same noise variance but doing so would preclude using all the data. Further discussion of the resulting tradeoff is in Section 3.3.4 and Section 4.7.4.

Inverse noise variance weights $w_\ell^2 = 1/\sigma_\ell^2$ do not correspond to an unweighted PCA and were not analyzed in Chapter III. The following corollary uses Theorem 4.3 to provide new simple expressions for these weights.

Corollary 4.6 (Asymptotic recoveries for inverse noise variance weights)**.** *Suppose the sample-to-dimension ratio $n/d \to c > 0$ and the noise variance proportions $n_\ell/n \to p_\ell$ for $\ell = 1, \dots, L$ as $n, d \to \infty$, and let the weights be set as $w_\ell^2 = \bar{\sigma}^2/\sigma_\ell^2$ where $\bar{\sigma}^{-2} := p_1/\sigma_1^2 + \cdots + p_L/\sigma_L^2$ is the average inverse noise variance. Then the ith WPCA amplitude $\hat{\theta}_i$ converges as*

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} r_i^{(\theta)} = \begin{cases} \theta_i^2\{1 + \bar{\sigma}^2/(c\theta_i^2)\}(1 + \bar{\sigma}^2/\theta_i^2) & \text{if } c\theta_i^4 > \bar{\sigma}^4, \\ \bar{\sigma}^2(1 + 1/\sqrt{c})^2 & \text{otherwise.} \end{cases} \tag{4.10}$$

Furthermore, if $c\theta_i^4 > \bar{\sigma}^4$ then the ith component $\hat{u}_i$ has asymptotic recovery

$$\sum_{j:\theta_j=\theta_i} |\langle \hat{u}_i, u_j \rangle|^2 \xrightarrow{\text{a.s.}} r_i^{(u)} = \frac{c - \bar{\sigma}^4/\theta_i^4}{c + \bar{\sigma}^2/\theta_i^2}, \qquad \sum_{j:\theta_j\neq\theta_i} |\langle \hat{u}_i, u_j \rangle|^2 \xrightarrow{\text{a.s.}} 0, \tag{4.11}$$

and the normalized ith score $\hat{z}_i/\sqrt{n}$ has asymptotic weighted recovery

$$\sum_{j:\theta_j=\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} r_i^{(z)} = \frac{c - \bar{\sigma}^4/\theta_i^4}{c(1 + \bar{\sigma}^2/\theta_i^2)}, \tag{4.12}$$
$$\sum_{j:\theta_j\neq\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} 0.$$

Proof of Corollary 4.6. When $w_\ell^2 = \bar{\sigma}^2/\sigma_\ell^2$, (4.5) and (4.6) simplify to

$$A(x) = 1 - \frac{c\bar{\sigma}^4}{(x-\bar{\sigma}^2)^2}, \qquad B_i(x) = 1 - \frac{c\theta_i^2}{x-\bar{\sigma}^2}, \qquad C(x) = 1 + \frac{c\bar{\sigma}^2}{x-\bar{\sigma}^2},$$

yielding $\alpha = \bar{\sigma}^2(1+\sqrt{c})$ and $\beta_i = \bar{\sigma}^2 + c\theta_i^2$. Substituting into (4.4), (4.7) and (4.8) in Theorem 4.3 yields (4.10)–(4.12). □

Observe that $\bar{\sigma}^2$ captures the overall noise level, and performance with inverse noise variance weights is the same as that for homoscedastic noise of variance $\bar{\sigma}^2$. In contrast to uniform and binary weights, the performance of inverse noise variance weights for fixed average inverse noise variance does not degrade with heteroscedasticity because the weights always whiten the noise to be homoscedastic. In fact, performance for fixed average noise variance improves with heteroscedasticity, with perfect recovery occurring when one noise variance is taken to zero with the rest set to have the desired average. As we show in Section 4.5, however, these weights generally result in suboptimal asymptotic component recovery (4.7).

4.3.2 Aggregate performance of weighted PCA

The following corollary applies Theorem 4.3 to analyze aggregate recovery of the components, scores and samples.

Corollary 4.7 (Aggregate recovery)**.** *Suppose the conditions of Theorem 4.3 hold, and additionally $A(\beta_1), \dots, A(\beta_k) > 0$. Then the WPCA component subspace basis $\hat{\mathbf{U}} := (\hat{u}_1, \dots, \hat{u}_k) \in \mathbb{C}^{d\times k}$ recovers the underlying subspace basis $\mathbf{U} := (u_1, \dots, u_k) \in \mathbb{C}^{d\times k}$ asymptotically as*

$$\|\hat{\mathbf{U}}^{\mathsf{H}}\mathbf{U}\|_F^2 \xrightarrow{\text{a.s.}} \sum_{i=1}^k r_i^{(u)} = \sum_{i=1}^k \frac{1}{\beta_i}\frac{A(\beta_i)}{B_i'(\beta_i)}, \tag{4.13}$$

the aggregate WPCA scores $\hat{\mathbf{Z}} := (\hat{z}_1, \dots, \hat{z}_k) \in \mathbb{C}^{n\times k}$ recover their underlying counterparts $\mathbf{Z} := (z_1, \dots, z_k) \in \mathbb{C}^{n\times k}$ asymptotically as

$$\frac{1}{n^2}\|\hat{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}^2\mathbf{Z}\|_F^2 \xrightarrow{\text{a.s.}} \sum_{i=1}^k r_i^{(z)} = \sum_{i=1}^k \frac{1}{c\theta_i^2 C(\beta_i)}\frac{A(\beta_i)}{B_i'(\beta_i)}, \tag{4.14}$$

and the reconstructed samples $\hat{x}_1, \dots, \hat{x}_n$ have asymptotic (weighted) mean square error with respect to the underlying samples $x_1, \dots, x_n$ given by

$$\frac{1}{n}\sum_{j=1}^{n} \omega_j^2 \|\hat{x}_j - x_j\|_2^2 \xrightarrow{\text{a.s.}} \frac{1}{c}\sum_{i=1}^{k} \left\{ c\bar{w}^2\theta_i^2 + \beta_i C(\beta_i) - 2\frac{A(\beta_i)}{B_i'(\beta_i)} \right\}, \tag{4.15}$$

where $\bar{w}^2 := p_1 w_1^2 + \cdots + p_L w_L^2$.

Proof of Corollary 4.7. The subspace and aggregate score recoveries decompose as

$$\|\hat{\mathbf{U}}^{\mathsf{H}}\mathbf{U}\|_F^2 = \sum_{i=1}^{k} \left(\sum_{j:\theta_j=\theta_i} |\langle \hat{u}_i, u_j \rangle|^2 + \sum_{j:\theta_j\neq\theta_i} |\langle \hat{u}_i, u_j \rangle|^2 \right), \tag{4.16}$$

$$\frac{1}{n^2}\|\hat{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}^2\mathbf{Z}\|_F^2 = \sum_{i=1}^{k} \left(\sum_{j:\theta_j=\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 + \sum_{j:\theta_j\neq\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \right). \tag{4.17}$$

Substituting (4.7)–(4.8) into (4.16)–(4.17) yields (4.13)–(4.14).

The (weighted) mean square error decomposes as

$$\begin{aligned} \frac{1}{n}\sum_{j=1}^{n} \omega_j^2 \|\hat{x}_j - x_j\|_2^2 &= \left\| \hat{\mathbf{U}}\hat{\mathbf{\Theta}}\Big(\frac{1}{\sqrt{n}}\hat{\mathbf{Z}}\Big)^{\mathsf{H}}\mathbf{W} - \mathbf{U}\mathbf{\Theta}\Big(\frac{1}{\sqrt{n}}\mathbf{Z}\Big)^{\mathsf{H}}\mathbf{W} \right\|_F^2 \\ &= \sum_{i=1}^{k} \hat{\theta}_i^2 + \frac{\theta_i^2}{n}\|\mathbf{W}z_i\|_2^2 - 2\Re\left(\hat{\theta}_i \sum_{j=1}^{k} \theta_j \langle \hat{u}_i, u_j \rangle \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2}^{*} \right) \end{aligned} \tag{4.18}$$

where $\hat{\mathbf{\Theta}} := \operatorname{diag}(\hat{\theta}_1, \dots, \hat{\theta}_k) \in \mathbb{R}^{k\times k}$ and $\mathbf{\Theta} := \operatorname{diag}(\theta_1, \dots, \theta_k) \in \mathbb{R}^{k\times k}$ are diagonal matrices of amplitudes, and $\Re$ denotes the real part of its argument. The first term of (4.18) has almost sure limit given by (4.4), and the second term has almost sure limit $\theta_i^2(p_1 w_1^2 + \cdots + p_L w_L^2)$ by the law of large numbers. The inner sum of the third term simplifies since summands with $\theta_j \neq \theta_i$ are zero by (4.7)–(4.8); the remaining sum has almost sure limit given by (4.9). Substituting the almost sure limits and simplifying yields (4.15). □

4.3.3 Conjectured phase transition

The expressions for asymptotic component recovery (4.7) and asymptotic score recovery (4.8) in Theorem 4.3 and the resulting recoveries in Corollary 4.7 apply only when $A(\beta_i) > 0$. The following conjecture predicts a phase transition when $A(\beta_i) = 0$ resulting in zero asymptotic recovery when $A(\beta_i) \leq 0$.

Conjecture 4.8 (Phase transition)**.** *Suppose the sample-to-dimension ratio $n/d \to c > 0$ and the noise variance proportions $n_\ell/n \to p_\ell$ for $\ell = 1, \dots, L$ as $n, d \to \infty$. If $A(\beta_i) \le 0$ then*

$$\sum_{j=1}^{k} |\langle \hat{u}_i, u_j \rangle|^2 \xrightarrow{\text{a.s.}} 0, \qquad \sum_{j=1}^{k} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} 0. \tag{4.19}$$

Namely, (4.7) and (4.8) extend to $A(\beta_i) \le 0$ by truncating $r_i^{(u)}$ and $r_i^{(z)}$ at zero.

4.4 Proof sketch for Theorem 4.3

This section provides a rough outline, deferring the details to Section 4.11.1. Observe first that in matrix form, the model (4.3) for the data matrix $\mathbf{Y} := (y_1, \dots, y_n) \in \mathbb{C}^{d \times n}$ is

$$\mathbf{Y} = \underbrace{(u_1, \dots, u_k)}_{\mathbf{U} \in \mathbb{C}^{d \times k}} \underbrace{\operatorname{diag}(\theta_1, \dots, \theta_k)}_{\mathbf{\Theta} \in \mathbb{R}^{k \times k}} \underbrace{(z_1, \dots, z_k)^{\mathsf{H}}}_{\mathbf{Z} \in \mathbb{C}^{n \times k}} + \underbrace{(\varepsilon_1, \dots, \varepsilon_n)}_{\mathbf{E} \in \mathbb{C}^{d \times n}} \underbrace{\operatorname{diag}(\eta_1, \dots, \eta_n)}_{\mathbf{H} \in \mathbb{R}^{n \times n}}. \tag{4.20}$$

The weighted PCA components $\hat{u}_1, \dots, \hat{u}_k$, amplitudes $\hat{\theta}_1, \dots, \hat{\theta}_k$, and normalized weighted scores $\mathbf{W}\hat{z}_1/\sqrt{n}, \dots, \mathbf{W}\hat{z}_k/\sqrt{n}$ are, respectively, principal left singular vectors, singular values, and right singular vectors of the normalized and weighted data matrix

$$\tilde{\mathbf{Y}} := \frac{1}{\sqrt{n}} \mathbf{Y} \underbrace{\operatorname{diag}(\omega_1^2, \dots, \omega_n^2)}_{\mathbf{W} \in \mathbb{R}^{n \times n}} = \mathbf{U}\mathbf{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W} + \tilde{\mathbf{E}}, \tag{4.21}$$

where $\tilde{\mathbf{Z}} := \mathbf{Z}/\sqrt{n}$ are normalized underlying scores and $\tilde{\mathbf{E}} := \mathbf{E}\mathbf{H}\mathbf{W}/\sqrt{n}$ are normalized and weighted noise. Namely, $\tilde{\mathbf{Y}}$ is a low-rank perturbation of a random matrix. We extend [22, Theorems 2.9-2.10] to account for weights, then exploit structure in the expressions similar to the proof in Section 3.5.

As shown in [14, Chapters 4, 6] and discussed in Section 3.5.1, the singular value distribution of $\tilde{\mathbf{E}}$ converges almost surely weakly to a nonrandom compactly supported measure $\mu_{\tilde{\mathbf{E}}}$, and the largest singular value of $\tilde{\mathbf{E}}$ converges almost surely to the supremum b of the support of $\mu_{\tilde{\mathbf{E}}}$. Hence, as reviewed in Section 4.11.1.1,

$$\frac{1}{d} \operatorname{tr} \zeta(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1} \xrightarrow{\text{a.s.}} \varphi_1(\zeta) := \int \frac{\zeta}{\zeta^2 - t^2} d\mu_{\tilde{\mathbf{E}}}(t), \tag{4.22}$$

where the convergence is uniform on $\{\zeta \in \mathbb{C} : \Re(\zeta) > b + \tau\}$ for any $\tau > 0$, and φ_1 has the following properties:

$$\forall_{\zeta > b}\ \varphi_1(\zeta) > 0, \qquad \varphi_1(\zeta) \to 0 \text{ as } |\zeta| \to \infty, \qquad \varphi_1(\zeta) \in \mathbb{R} \Leftrightarrow \zeta \in \mathbb{R}. \tag{4.23}$$

Furthermore, for any $\zeta \in \mathbb{C}$ with $\Re(\zeta) > b$,

$$\frac{\partial}{\partial \zeta} \frac{1}{d} \operatorname{tr} \zeta (\zeta^2 \mathbf{I} - \tilde{\mathbf{E}} \tilde{\mathbf{E}}^{\mathsf{H}})^{-1} \xrightarrow{\text{a.s.}} \varphi_1'(\zeta). \tag{4.24}$$

The main technical challenge in extending [22, Theorems 2.9-2.10] to account for the weights lies in proving analogous weighted results stated in the following lemma.

Lemma 4.9. *Under the model assumptions in Section 4.2,*

$$\frac{1}{n} \operatorname{tr} \zeta \mathbf{W} (\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}} \tilde{\mathbf{E}})^{-1} \mathbf{W} \xrightarrow{\text{a.s.}} \varphi_2(\zeta) := \sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{\zeta - w_\ell^2 \sigma_\ell^2 \varphi_1(\zeta)/c}, \tag{4.25}$$

where the convergence is uniform on $\{\zeta \in \mathbb{C} : \Re(\zeta) > b + \tau\}$ for any $\tau > 0$, and φ_2 has the following properties:

$$\forall_{\zeta > b}\ \varphi_2(\zeta) > 0, \qquad \varphi_2(\zeta) \to 0 \text{ as } |\zeta| \to \infty, \qquad \varphi_2(\zeta) \in \mathbb{R} \Leftrightarrow \zeta \in \mathbb{R}. \tag{4.26}$$

Furthermore, for any $\zeta \in \mathbb{C}$ with $\Re(\zeta) > b$,

$$\frac{\partial}{\partial \zeta} \frac{1}{n} \operatorname{tr} \zeta \mathbf{W} (\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}} \tilde{\mathbf{E}})^{-1} \mathbf{W} \xrightarrow{\text{a.s.}} \varphi_2'(\zeta). \tag{4.27}$$

Lemma 4.9 is proved in Section 4.11.2 and enables us to extend [22, Theorems 2.9-2.10] in Sections 4.11.1.2 and 4.11.1.3 to conclude that for each $i \in \{1, \dots, k\}$,

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \begin{cases} \rho_i^2 & \text{if } \theta_i^2 > \bar{\theta}^2, \\ b^2 & \text{otherwise,} \end{cases} =: r_i^{(\theta)} \tag{4.28}$$

and when $\theta_i^2 > \bar{\theta}^2$,

$$\sum_{j:\theta_j = \theta_i} |\langle \hat{u}_i, u_j \rangle|^2 \xrightarrow{\text{a.s.}} \frac{-2\varphi_1(\rho_i)}{\theta_i^2 D'(\rho_i)} =: r_i^{(u)}, \tag{4.29}$$

$$\sum_{j:\theta_j = \theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} \frac{-2\varphi_2(\rho_i)}{\theta_i^2 D'(\rho_i)} =: r_i^{(z)}, \tag{4.30}$$

$$\sum_{j:\theta_j \neq \theta_i} |\langle \hat{u}_i, u_j \rangle|^2, \sum_{j:\theta_j \neq \theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \right|^2 \xrightarrow{\text{a.s.}} 0, \tag{4.31}$$

and

$$\sum_{j:\theta_j = \theta_i} \langle \hat{u}_i, u_j \rangle \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{W}^2}^* \xrightarrow{\text{a.s.}} \sqrt{r_i^{(u)} r_i^{(z)}}, \tag{4.32}$$

where $D(\zeta) := \varphi_1(\zeta)\varphi_2(\zeta)$, $\rho_i := D^{-1}(1/\theta_i^2)$ and $\bar{\theta}^2 := 1/\lim_{\zeta \to b^+} D(\zeta)$.

The final step (Section 4.11.1.4) is to find algebraic descriptions of $r_i^{(u)}$ and $r_i^{(z)}$. We change variables to $\psi(\zeta) := c\zeta/\varphi_1(\zeta)$ and, analogous to Section 3.5.3, observe that ψ has the following properties:

a) $0 = Q(\psi(\zeta), \zeta)$ for all $\zeta > b$ where

$$Q(s,\zeta) := \frac{c\zeta^2}{s^2} + \frac{c-1}{s} - c\sum_{\ell=1}^{L} \frac{p_\ell}{s - w_\ell^2\sigma_\ell^2}, \tag{4.33}$$

with the inverse function given by

$$\psi^{-1}(\gamma) = \sqrt{\frac{\gamma}{c}\left(1 + c\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2 \sigma_\ell^2}{\gamma - w_\ell^2\sigma_\ell^2}\right)}, \tag{4.34}$$

b) $\max_\ell(\sigma_\ell^2 w_\ell^2) < \psi(\zeta) < c\zeta^2$,

c) $0 < \lim_{\zeta\to b^+} \psi(\zeta) < \infty$ and $\lim_{\zeta\to b^+} \psi'(\zeta) = \infty$.

Combining these properties with the observation that

$$D(\zeta) = \varphi_1(\zeta)\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{z - w_\ell^2\sigma_\ell^2\varphi_1(\zeta)/c} = c\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{\psi(\zeta) - w_\ell^2\sigma_\ell^2}, \tag{4.35}$$

then simplifying analogously to Sections 3.5.4 to 3.5.6, yields (4.4)–(4.9) and concludes the proof.

4.5 Optimally weighted PCA

The following theorem optimizes the expressions in Theorem 4.3 to find weights that maximize component recovery. The absolute scale and units of the weights are arbitrary here since the components depend on only the *relative* weights given to samples, as discussed in Remark 4.5.

Theorem 4.10 (Optimal component recovery)**.** *The weights*

$$w_\ell^2 = \frac{1}{\sigma_\ell^2}\frac{1}{\theta_i^2 + \sigma_\ell^2}, \tag{4.36}$$

maximize the asymptotic recovery $r_i^{(u)}$ of the ith underlying component u_i by the WPCA component $\hat{u}_i$ with the corresponding optimal value of $r_i^{(u)}$ given by the largest real root of

$$R_i^{(u)}(x) := 1 - c\theta_i^2\sum_{\ell=1}^{L} \frac{p_\ell}{\sigma_\ell^2}\frac{1-x}{\sigma_\ell^2/\theta_i^2 + x}. \tag{4.37}$$

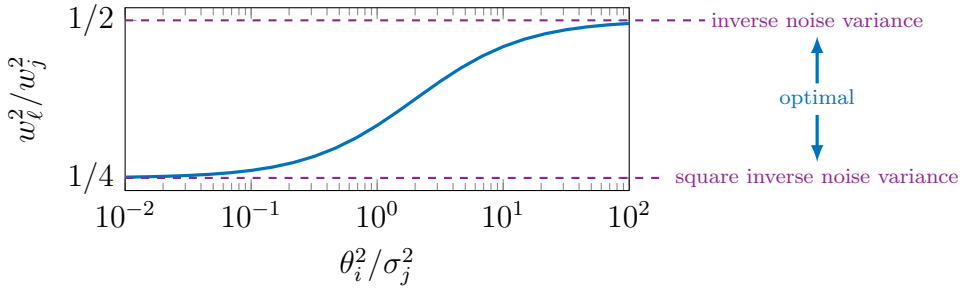


Figure 4.2: Relative weight w_ℓ^2/w_j^2 given by optimal weights (4.36) to samples with twice the noise variance $\sigma_\ell^2 = 2\sigma_j^2$ as a function of the underlying amplitude θ_i^2. As the underlying amplitude increases, optimal weighting interpolates between square inverse noise variance weights ($w_\ell^2/w_j^2 = 1/4$) and inverse noise variance weights ($w_\ell^2/w_j^2 = 1/2$).

When $\theta_i^2 \gg \sigma_1^2, \dots, \sigma_L^2$, i.e., when the noise is relatively small, $1/(\theta_i^2 + \sigma_\ell^2)$ becomes uniform over ℓ and the optimal weights (4.36) reduce to inverse noise variance weights, providing further justification for these commonly used weights. However, when $\theta_i^2 \ll \sigma_1^2, \dots, \sigma_L^2$ and the noise is relatively large, $1/(\theta_i^2 + \sigma_\ell^2)$ becomes $1/\sigma_\ell^2$ and the optimal weights reduce to *square* inverse noise variance weights. Inverse noise variance weights do not downweight noisier samples aggressively enough when the signal-to-noise ratio is small. Rather than give samples with twice the noise variance half the weight as with inverse noise variance weights, it is better to give them a quarter the weight in this regime. In general, optimal weights strike a balance between inverse noise variance weights and square inverse noise variance weights, as

$$\frac{1/\sigma_\ell^4}{1/\sigma_j^4} < \frac{\sigma_j^4}{\sigma_\ell^4}\frac{\theta_i^2/\sigma_j^2 + 1}{\theta_i^2/\sigma_\ell^2 + 1} = \frac{w_\ell^2}{w_j^2} = \frac{\sigma_j^2}{\sigma_\ell^2}\frac{\theta_i^2 + \sigma_j^2}{\theta_i^2 + \sigma_\ell^2} < \frac{1/\sigma_\ell^2}{1/\sigma_j^2},$$

for any two noise variances $\sigma_\ell^2 > \sigma_j^2$. Samples with twice the noise variance are given between a half and a quarter of the weight, with the particular balance dictated by the underlying amplitude θ_i^2, as shown in Fig. 4.2. In practice, one may estimate the underlying amplitudes θ_i^2 by de-biasing PCA estimates $\hat{\theta}_i^2$ using expressions like (4.4).

Interestingly, the optimal weights (4.36) do not depend on the sample-to-dimension ratio c or proportions $p_1, \dots, p_L$, though these properties greatly impact how informative each group of samples is on the whole, as shown in Section 4.6. Consequently, there is no benefit to using different weights for samples with the same noise variance. Furthermore, note that the second term $1/(\theta_i^2 + \sigma_\ell^2)$ normalizes samples by their variance in the direction of u_i.

The remainder of this section proves Theorem 4.10. Though the result (4.36) is simple to state, deriving it is nontrivial in part due to the fact that any scaling of

the weights produces the same components. The proof exploits this structure to find optimal weights and their corresponding recovery.

Proof of Theorem 4.10. The objective is to maximize $r_i^{(u)}$ with respect to the weights $w_1^2, \dots, w_L^2$ under the implicit constraint that the weights are nonnegative. Partition the feasible region into $2^L - 1$ sets each defined by which weights are zero. Namely, consider partitions of the form

$$\mathcal{P}_{\mathcal{L}} := \{(w_1^2, \dots, w_L^2) : \forall \ell \in \mathcal{L}\ w_\ell^2 = 0, \forall \ell \notin \mathcal{L}\ w_\ell^2 > 0\}, \tag{4.38}$$

where $\mathcal{L} \subset \{1, \dots, L\}$ is a proper, but potentially empty, subset. Note that the origin, where all the weights are zero, is not within the domain of $r_i^{(u)}$. Since $r_i^{(u)}$ is invariant to scaling of the weights, as discussed in Remark 4.5, a maximizer exists within at least one of the partitions. Moreover, since $r_i^{(u)}$ is a differentiable function of the weights, $r_i^{(u)}$ is maximized at a critical point of a partition $\mathcal{P}_{\mathcal{L}}$. It remains to identify and compare the critical points of all the partitions.

First consider $\mathcal{P}_\emptyset$, i.e., the set of positive weights $w_1^2, \dots, w_L^2 > 0$, and let $\tilde{w}_j := 1/w_j^2$. This reparameterization ends up simplifying the manipulations. Differentiating key terms from Theorem 4.3, specifically (4.7) and (4.5), with respect to $\tilde{w}_j$ yields

$$\frac{\partial r_i^{(u)}}{\partial \tilde{w}_j} = r_i^{(u)} \left\{ -\frac{1}{\beta_i}\frac{\partial \beta_i}{\partial \tilde{w}_j} + \frac{1}{A(\beta_i)}\frac{\partial A(\beta_i)}{\partial \tilde{w}_j} - \frac{1}{B_i'(\beta_i)}\frac{\partial B_i'(\beta_i)}{\partial \tilde{w}_j} \right\}, \tag{4.39}$$

$$\frac{\partial A(\beta_i)}{\partial \tilde{w}_j} = A'(\beta_i)\frac{\partial \beta_i}{\partial \tilde{w}_j} + 2c\frac{p_j \sigma_j^4}{(\beta_i \tilde{w}_j - \sigma_j^2)^3}\beta_i, \tag{4.40}$$

$$\frac{\partial B_i'(\beta_i)}{\partial \tilde{w}_j} = B_i''(\beta_i)\frac{\partial \beta_i}{\partial \tilde{w}_j} - 2c\theta_i^2\frac{p_j}{(\beta_i \tilde{w}_j - \sigma_j^2)^3}\beta_i \tilde{w}_j + c\theta_i^2\frac{p_j}{(\beta_i \tilde{w}_j - \sigma_j^2)^2}, \tag{4.41}$$

$$0 = \frac{\partial B_i(\beta_i)}{\partial \tilde{w}_j} = B_i'(\beta_i)\frac{\partial \beta_i}{\partial \tilde{w}_j} + c\theta_i^2\frac{p_j}{(\beta_i \tilde{w}_j - \sigma_j^2)^2}\beta_i, \tag{4.42}$$

where one must carefully account for the fact that A and B_i are implicit functions of $\tilde{w}_j$, so β_i is as well. Rewriting (4.40) and (4.41) in terms of $\partial \beta_i / \partial \tilde{w}_j$ using (4.42) yields

$$\frac{\partial A(\beta_i)}{\partial \tilde{w}_j} = \left\{ A'(\beta_i) - \frac{2B_i'(\beta_i)}{\theta_i^2}\frac{\sigma_j^4}{\beta_i \tilde{w}_j - \sigma_j^2} \right\}\frac{\partial \beta_i}{\partial \tilde{w}_j}, \tag{4.43}$$

$$\frac{\partial B_i'(\beta_i)}{\partial \tilde{w}_j} = \left\{ B_i''(\beta_i) + 2B_i'(\beta_i)\frac{\tilde{w}_j}{\beta_i \tilde{w}_j - \sigma_j^2} \right\}\frac{\partial \beta_i}{\partial \tilde{w}_j} - \frac{1}{\beta_i}B_i'(\beta_i)\frac{\partial \beta_i}{\partial \tilde{w}_j}. \tag{4.44}$$

Substituting (4.43) and (4.44) into (4.39) then rearranging yields

$$\frac{\partial r_i^{(u)}}{\partial \tilde{w}_j} = \frac{2}{\theta_i^2 \beta_i}\frac{\partial \beta_i}{\partial \tilde{w}_j}\left\{ \theta_i^2 \Delta_i - \frac{\theta_i^2 \beta_i r_i^{(u)} \tilde{w}_j + \sigma_j^4}{\beta_i \tilde{w}_j - \sigma_j^2} \right\}, \tag{4.45}$$

where the following term is independent of j:

$$\Delta_i := \frac{1}{2}\frac{A(\beta_i)}{B_i'(\beta_i)}\left\{\frac{A'(\beta_i)}{A(\beta_i)} - \frac{B_i''(\beta_i)}{B_i'(\beta_i)}\right\}. \tag{4.46}$$

Since $\beta_i > \max_\ell(w_\ell^2\sigma_\ell^2) > 0$ it follows that $\partial\beta_i/\partial\tilde{w}_j \neq 0$, so (4.45) is zero exactly when

$$\theta_i^2\Delta_i = \frac{\theta_i^2\beta_i r_i^{(u)}\tilde{w}_j + \sigma_j^4}{\beta_i\tilde{w}_j - \sigma_j^2}. \tag{4.47}$$

Rearranging (4.7) and substituting (4.47) yields

$$\begin{aligned} 0 = A(\beta_i) - r_i^{(u)}\beta_i B_i'(\beta_i) &= 1 - c\sum_{\ell=1}^{L}\frac{p_\ell(\sigma_\ell^4 + \theta_i^2\beta_i r_i^{(u)}\tilde{w}_\ell)}{(\beta_i\tilde{w}_\ell - \sigma_\ell^2)^2} \\ &= 1 - \Delta_i c\theta_i^2\sum_{\ell=1}^{L}\frac{p_\ell}{\beta_i\tilde{w}_\ell - \sigma_\ell^2} = 1 - \Delta_i(1 - B_i(\beta_i)) = 1 - \Delta_i, \end{aligned} \tag{4.48}$$

so $\Delta_i = 1$. Substituting into (4.47) and solving for $\tilde{w}_j$ yields

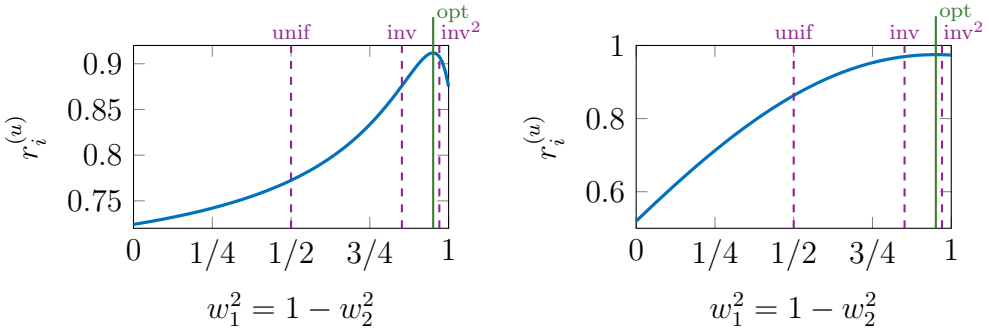
$$w_j^2 = \frac{1}{\tilde{w}_j} = \frac{(1 - r_i^{(u)})\theta_i^2\beta_i}{\sigma_j^2(\theta_i^2 + \sigma_j^2)} = \frac{\kappa_i}{\sigma_j^2(\theta_i^2 + \sigma_j^2)}, \tag{4.49}$$

where the constant $\kappa_i := (1 - r_i^{(u)})\theta_i^2\beta_i$ is: a) independent of j, b) parameterizes the ray of critical points in $\mathcal{P}_\emptyset$, and c) can be chosen freely, e.g., as unity yielding (4.36). Solving (4.49) for $\beta_i\tilde{w}_j$, substituting into (4.5), and rearranging yields that the corresponding $r_i^{(u)}$ is a root of $R_i^{(u)}$ in (4.37). Since $R_i^{(u)}(x)$ increases from negative infinity to one as x increases from $-\min_\ell(\sigma_\ell^2)/\theta_i^2$ to one, it has exactly one real root in that domain. In particular, this root is the largest real root since $R_i^{(u)}(x) \geq 1$ for $x \geq 1$. Furthermore, $r_i^{(u)}$ increases continuously to one as c increases to infinity, so $r_i^{(u)}$ is the largest real root.

Likewise, the critical points of other partitions $\mathcal{P}_\mathcal{L}$ are given by setting the positive weights proportional to (4.36) with the corresponding $r_i^{(u)}$ given by the largest real root of

$$R_{i,\mathcal{L}}^{(u)}(x) := 1 - c\theta_i^2\sum_{\ell\notin\mathcal{L}}\frac{p_\ell}{\sigma_\ell^2}\frac{1 - x}{\sigma_\ell^2/\theta_i^2 + x}. \tag{4.50}$$

For $\mathcal{L}_1 \subset \mathcal{L}_2$ a proper subset, the largest real root of $R_{i,\mathcal{L}_1}^{(u)}$ is greater than that of $R_{i,\mathcal{L}_2}^{(u)}$ since $R_{i,\mathcal{L}_1}^{(u)}(x) < R_{i,\mathcal{L}_2}^{(u)}(x)$ for any $x \in (-\min_\ell(\sigma_\ell^2)/\theta_i^2, 1)$. As a result, $r_i^{(u)}$ is maximized in $\mathcal{P}_\emptyset$. □



(a) $p_1 = 10\%$, $p_2 = 90\%$.

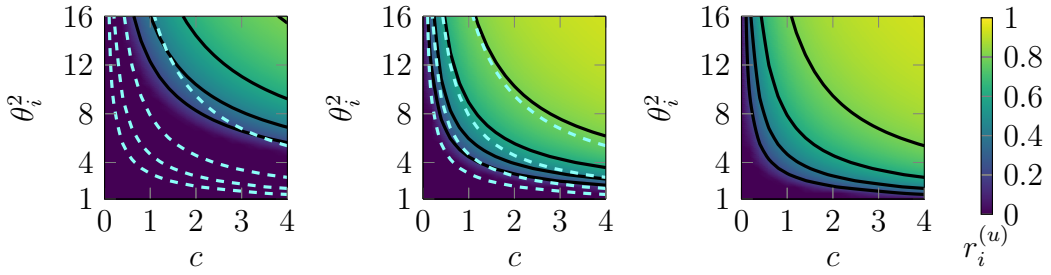
(b) $p_1 = p_2 = 50\%$.

Figure 4.3: Asymptotic component recovery (4.7) for $c = 150$ samples per dimension, underlying amplitude $\theta_i^2 = 1$, and noise variances $\sigma_1^2 = 1$ and $\sigma_2^2 = 5.75$, as the weight $w_1^2 = 1 - w_2^2$ for the cleaner samples sweeps from zero to one. At the extremes only noiser samples are used ($w_1^2 = 0$) or only cleaner samples are used ($w_1^2 = 1$). Vertical lines indicate which weights correspond to unweighted PCA (unif), inverse noise variance weights (inv), square inverse noise variance weights (inv^2), and optimal weights (opt) from (4.36). Theorem 4.3 quantifies the benefit of combining in (a), and the near optimality of using only cleaner data in (b).

4.6 Suboptimal weighting

Theorem 4.3 provides a way to not only find optimal weights, but to also quantify how suboptimal other weights are. Suppose there are $c = 150$ samples per dimension, the underlying amplitude is $\theta_i^2 = 1$ and $p_1 = 10\%$ of samples have noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 90\%$ having noise variance $\sigma_2^2 = 5.75$. Figure 4.3a shows the asymptotic component recovery (4.7) as the weight w_1^2 given to the cleaner samples increases, with the weight for the noisier samples set as $w_2^2 = 1 - w_1^2$; this sweep covers all possible weights since the components depend on only the relative weights as discussed in Remark 4.5. In this case, excluding either set of samples is significantly suboptimal. Using the noisier data alone ($w_1^2 = 0$) achieves $r_i^{(u)} \approx 0.72$, using the cleaner data alone ($w_1^2 = 1$) achieves $r_i^{(u)} \approx 0.88$, and optimal weighting achieves $r_i^{(u)} \approx 0.91$. Inverse noise variance weights achieve $r_i^{(u)} \approx 0.88$ and are similar to using only the cleaner data. The optimal weights here are closer to square inverse noise variance.

Now suppose the proportions are $p_1 = p_2 = 50\%$ with all other parameters the same. Figure 4.3b shows the asymptotic component recovery (4.7). In this case, using only the cleaner data, using inverse noise variance weights, or using square inverse noise variance weights are all nearly optimal; these choices and the optimal weighting all have recovery $r_i^{(u)} \approx 0.97$. Observe that all the indicated weights are



(a) Unweighted PCA. (b) Inverse noise variance. (c) Optimal weights.

Figure 4.4: Asymptotic component recovery (4.7) as a function of the number of samples per dimension c and the underlying amplitude θ_i^2, where $p_1 = 20\%$ of samples have noise variance $\sigma_1^2 = 1$, and the remaining $p_2 = 80\%$ have noise variance $\sigma_2^2 = 10$. Contours are shown in black, and the contours for optimal weights (c) are overlaid as light blue dashed lines in (a) and (b). Inverse noise variance and optimal weights significantly improve PCA performance, with optimal weights providing greater improvement for small amplitudes.

the same as those in (a) since none depend on proportions. However, the recovery depends on weights in a dramatically different way. The cleaner data is sufficiently abundant in this setting to achieve great recovery, and the noisy data add little.

Using suboptimal weights is sometimes convenient. For example, (square) inverse noise variance weights can be applied without estimating θ_i^2. Dropping noisier samples can reduce computational or administrative burden. For some applications, these suboptimal weights may perform sufficiently well; Theorem 4.3 enables quantitative reasoning about the trade-offs.

4.7 Impact of model parameters

Theorem 4.3 also provides new insight into the ways weighting changes the performance of PCA with respect to the model parameters: sample-to-dimension ratio c, amplitudes $\theta_1^2, \ldots, \theta_k^2$, proportions $p_1, \ldots, p_L$ and noise variances $\sigma_1^2, \ldots, \sigma_L^2$. This section compares the impact on: a) unweighted PCA, b) inverse noise variance weighted PCA, and c) optimally weighted PCA. We illustrate the phenomena with two noise variances for simplicity; the same insights apply more broadly. See also Section 3.3 for related discussion regarding unweighted PCA.

4.7.1 Impact of sample-to-dimension ratio c and amplitude θ_i^2

Suppose that $p_1 = 20\%$ of samples have noise variance $\sigma_1^2 = 1$, and the remaining $p_2 = 80\%$ have noise variance $\sigma_2^2 = 10$. Figure 4.4 shows the asymptotic compo-

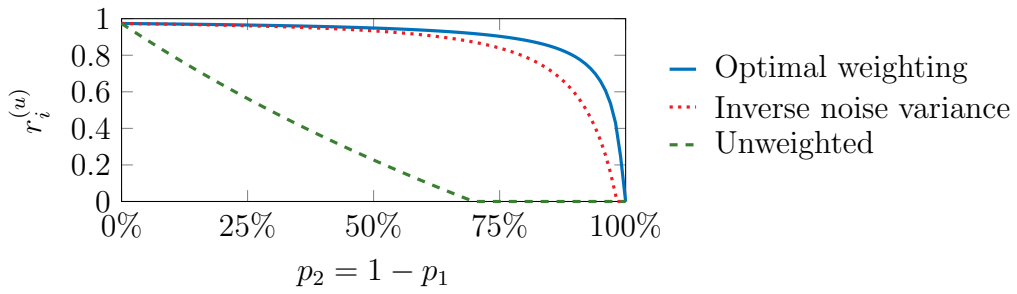


Figure 4.5: Asymptotic component recovery (4.7) as a function of the proportion p_2 of samples corrupted by noise with a large variance $\sigma_2^2 = 10$ while the remaining $p_1 = 1 - p_2$ samples have noise variance $\sigma_1^2 = 1$. There are $c = 75$ samples per dimension and the underlying amplitude is $\theta_i^2 = 1$. Inverse noise variance weighted PCA is more robust to such contaminations than unweighted PCA, and optimally weighted PCA is even more robust.

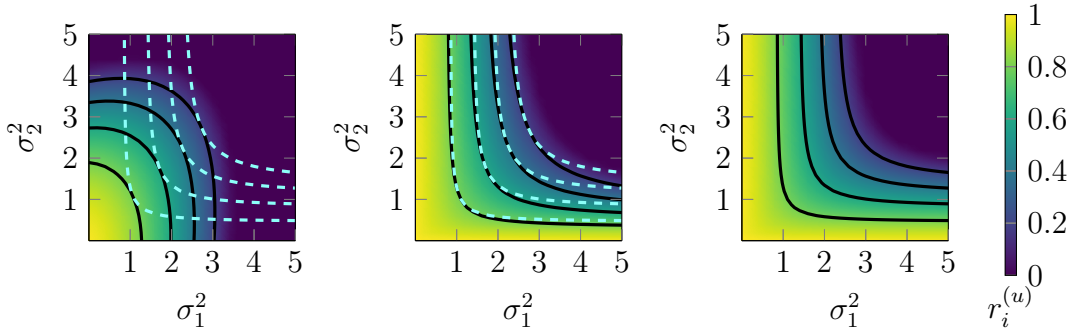
nent recovery (4.7) as the samples per dimension c and the underlying amplitude θ_i^2 vary. Decreasing the amplitude degrades recovery, and the lost performance can be regained by increasing the number of samples per dimension. Both inverse noise variance and optimal weights significantly outperform unweighted PCA, with optimal weights providing more improvement for small underlying amplitudes. Each contour for inverse noise variance weights is defined by (4.11) in Corollary 4.6, and each contour for optimal weights is defined by (4.37) in Theorem 4.10.

4.7.2 Impact of proportions $p_1, \ldots, p_L$

Suppose there are $c = 75$ samples per dimension, the underlying amplitude is $\theta_i^2 = 1$, and contaminated samples with noise variance $\sigma_2^2 = 10$ occur in proportion p_2 while the remaining $p_1 = 1 - p_2$ proportion of samples have noise variance $\sigma_1^2 = 1$. Figure 4.5 shows the asymptotic component recovery (4.7) as the contamination proportion p_2 increases. Unweighted PCA is not robust to such contamination, but inverse noise variance weights achieve good recovery for even significant amounts of contamination. Optimal weights are even more robust at extreme levels of contamination, since they more aggressively downweight noisier samples.

4.7.3 Impact of noise variances $\sigma_1^2, \ldots, \sigma_L^2$

Suppose $p_1 = 70\%$ of samples have noise variance σ_1^2, $p_2 = 30\%$ have noise variance σ_2^2, and there are $c = 10$ samples per dimension with underlying amplitude $\theta_i^2 = 1$. Figure 4.6 shows the asymptotic component recovery (4.7) as σ_1^2 and σ_2^2 vary. In general, performance degrades as noise variances increase. As discussed in Section 3.3.3, a large noise variance can dominate unweighted PCA performance



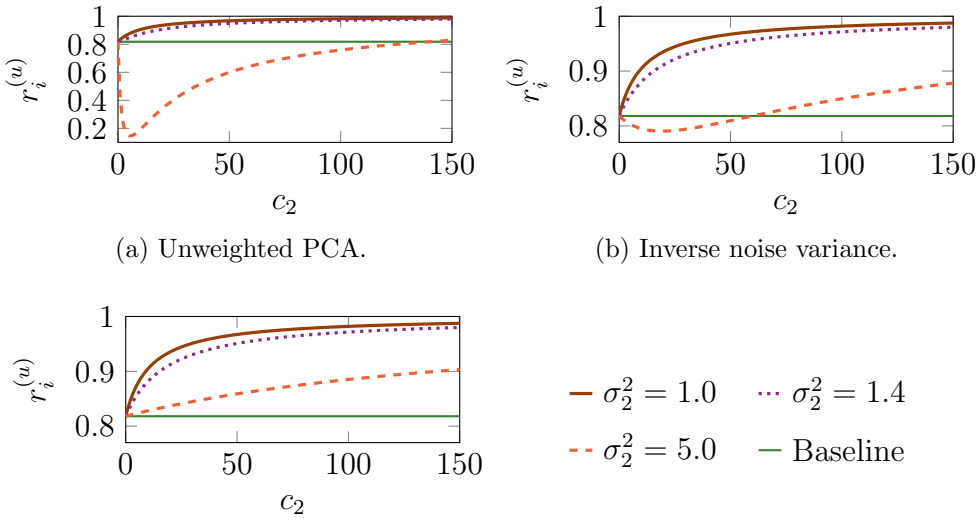
(a) Unweighted PCA. (b) Inverse noise variance. (c) Optimal weights.

Figure 4.6: Asymptotic component recovery (4.7) as a function of noise variances σ_1^2 and σ_2^2 appearing in proportions $p_1 = 70\%$ and $p_2 = 30\%$. There are $c = 10$ samples per dimension and the underlying amplitude is $\theta_i^2 = 1$. Contours are shown in black, and the contours for optimal weights (c) are overlaid as light blue dashed lines in (a) and (b). While unweighted PCA is most sensitive to the largest noise variance, inverse noise variance and optimal weights are most sensitive to the smallest noise variance, with optimal weights providing more improvement for large heteroscedasticity.

even when it occurs in a small proportion of samples; unweighted PCA is not robust to gross errors, i.e., outliers. In Fig. 4.6a, the contours show that decreasing σ_1^2 does not significantly improve performance when σ_2^2 is large.

In contrast, weighted PCA performance depends more on the smallest noise variance for both inverse noise variance weights and optimal weights since both types of weights give cleaner samples more influence. In Figs. 4.6b and 4.6c, the contours show that increasing σ_1^2 does not significantly degrade performance when σ_2^2 is small and vice versa for small σ_1^2. In particular, each contour in Fig. 4.6b is defined by having equal average inverse noise variance $\bar{\sigma}^{-2} := p_1/\sigma_1^2 + \cdots + p_L/\sigma_L^2$; see Corollary 4.6. Similarly, each contour in Fig. 4.6c is defined by (4.37) in Theorem 4.10. In both cases, as a noise variance grows to infinity, its impact diminishes and the other noise variances determine the resulting performance. For optimal weights, this limiting performance matches that of excluding the noisiest data. Inverse noise variance weights, however, achieve worse performance in this limit as shown by the overlaid contours; excluding the noisiest data is better. Since inverse noise variance weights always scale samples to have unit variance noise, the noisiest samples remain in the weighted PCA even though their signal to noise ratio diminishes to zero as their noise variance grows to infinity. Optimal weights are more aggressive and do remove the noisiest samples in this limit.

A surprising finding of Section 3.3.3 was that adding noise sometimes improves the performance of unweighted PCA. The same is not true for inverse noise variance



(a) Unweighted PCA.

(b) Inverse noise variance.

(c) Optimal weights.

Figure 4.7: Asymptotic component recovery (4.7) as c_2 samples per dimension with noise variance σ_2^2 are added to $c_1 = 10$ samples per dimension having noise variance $\sigma_1^2 = 1$. The underlying amplitude is $\theta_i^2 = 1$. Including noisier samples can degrade the performance of unweighted PCA or inverse noise variance weights, but optimally weighted PCA always improves when given more data.

or optimal weights. Adding any noise increases $\bar{\sigma}^2$, degrading the performance for inverse noise variance weights. Likewise, adding noise increases the function $R_i^{(u)}$ in (4.37), decreasing its largest root and degrading the performance for optimal weights.

4.7.4 Impact of including noisier data

Consider adding c_2 samples per dimension with noise variance σ_2^2 to a dataset containing $c_1 = 10$ samples per dimension with noise variance $\sigma_1^2 = 1$, all with underlying amplitude $\theta_i^2 = 1$. The combined dataset has $c = c_1 + c_2$ samples per dimension with noise variances σ_1^2 and σ_2^2 occurring with proportions $p_1 = c_1/c$ and $p_2 = c_2/c$. Figure 4.7 shows the asymptotic component recovery (4.7) for various noise variances σ_2^2 as a function of the amount of samples c_2. When $c_2 = 0$ only the original data are used; horizontal green lines indicate this baseline recovery.

As discussed in Section 3.3.4, the additional samples improve unweighted PCA performance when σ_2^2 is small enough or when c_2 is large enough to overcome the additional noise. Including a small number of much noisier samples degrades performance since unweighted PCA is not robust to them. Inverse noise variance weighted PCA is more robust and outperforms unweighted PCA, but including very noisy

samples again degrades performance unless c_2 is large enough. Inverse noise variance weights do not downweight the noisier samples enough, and sometimes excluding noisier data is better.

With optimally weighted PCA, on the other hand, using more data always improves performance. Since the weights are optimal, they are necessarily at least as good as binary weights that exclude the noisier data. The optimal combination of original and noisier data is no worse than either one alone, so including more samples only helps. See Remark 4.14 for related discussion. This benefit is most dramatically seen when the data being included are much noisier, and it would be interesting to characterize the regimes where optimal weighting most significantly impacts this aspect of weighted PCA performance.

4.8 Optimal sampling under budget constraints

This section uses Theorem 4.10 to consider optimizing a sampling strategy to maximize the recovery of optimally weighted PCA. Specifically, consider acquiring samples of varying quality, cost and availability under a budget. Given that the samples will be optimally weighted, what combination of inexpensive noisy samples and expensive clean samples maximizes asymptotic component recovery? What if we already have previously collected data? The following theorem uses (4.37) to answer these questions. Note that previously collected data are simply samples with limited availability and zero acquisition cost.

Theorem 4.11 (Optimal sampling for a budget)**.** *Consider L sources of d-dimensional samples with associated noise variances $\sigma_1^2, \dots, \sigma_L^2$ and corresponding costs $\tau_1, \dots, \tau_L$. Let $n_1, \dots, n_L \geq 0$ be the numbers of samples collected. Suppose the total cost is constrained by the available budget T as*

$$n_1\tau_1 + \cdots + n_L\tau_L \leq T, \tag{4.51}$$

and $n_1, \dots, n_L$ are constrained by limited availability of samples as

$$n_\ell \leq q_\ell, \quad \ell \in \{1, \dots, L\}, \tag{4.52}$$

where q_ℓ is the quantity available for source ℓ. Then the sample-to-dimension ratios $c_1, \dots, c_L \geq 0$, defined for each $\ell \in \{1, \dots, L\}$ as $c_\ell := n_\ell/d$, are constrained to the polyhedron in the nonnegative orthant defined by

$$c_1\tau_1 + \cdots + c_L\tau_L \leq T/d, \qquad c_\ell \leq q_\ell/d, \quad \ell \in \{1, \dots, L\}, \tag{4.53}$$

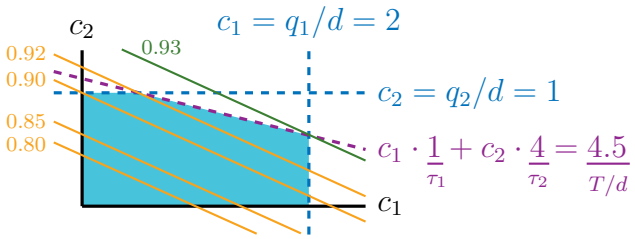


Figure 4.8: Optimal sampling under a budget occurs at extreme points of the polyhedron in the nonnegative orthant defined by the budget and availability constraints (4.53) shown in purple and blue, respectively. The total budget per dimension is $T/d = 4.5$, and samples cost $\tau_1 = 1$ and $\tau_2 = 4$ with associated availabilities per dimension $q_1/d = 2$ and $q_2/d = 1$, i.e., samples from the first source are cheaper and more abundant. Contours of $r_i^{(u)}$ for optimal weights are overlaid for noise variances $\sigma_1^2 = 2$ and $\sigma_2^2 = 1$ and an underlying amplitude $\theta_i^2 = 10$. The best contour (green) intersects the feasible polyhedron at $c_1 = 2, c_2 = 5/8$, where all available cheaper, noisier samples are collected with the remaining budget used for the higher quality samples.

and the asymptotic component recovery (4.37) *with optimal weights is maximized with respect to* $c_1, \ldots, c_L$ *at an extreme point of the polyhedron* (4.53)*. Furthermore, all maximizers occur at points where increasing any one of* $c_1, \ldots, c_L$ *would violate* (4.53)*, i.e., at points where the budget and availability are fully utilized.*

Remark 4.12 (Additional budget constraints). Theorem 4.11 considers a single budget constraint (4.51) for simplicity, but the same result holds with multiple linear constraints. For example, one constraint may pertain to the time needed for acquiring samples and another could pertain to the money needed.

Remark 4.13 (Unlimited availability). Theorem 4.11 assumes that all sources have a limited availability of samples q_ℓ for simplicity, but the same result holds as long as all sources have either nonzero cost, limited availability or both. If a source has both no cost and unlimited availability, asymptotic component recovery is maximized by acquiring increasingly many of its samples.

Remark 4.14 (Samples with no cost). An immediate consequence of Theorem 4.11 is that any samples with no cost, e.g., previously collected data, should always be included when using optimal weights. Doing so is, perhaps surprisingly, not always best when using unweighted or inverse noise variance weighted PCA. As demonstrated in Section 4.7.4, including noisier samples can degrade performance for suboptimal weights.

To illustrate Theorem 4.11, suppose that samples with noise variance $\sigma_1^2 = 2$ cost $\tau_1 = 1$ and have availability per dimension $q_1/d = 2$, and samples with noise variance

$\sigma_2^2 = 1$ cost $\tau_2 = 4$ and have availability per dimension $q_2/d = 1$, where the overall budget per dimension is $T/d = 4.5$. Namely, the first source of samples is twice as noisy but also a quarter the cost and twice as abundant. What combination of sampling rates c_1 and c_2 maximizes recovery by optimally weighted PCA of an underlying component with associated amplitude $\theta_i^2 = 10$? As predicted by Theorem 4.11, the maximum in Fig. 4.8 occurs at an extreme point of the polyhedron in the nonnegative orthant defined by (4.53). Furthermore, it occurs at an extreme point where increasing either c_1 or c_2 would violate the constraints, i.e., at $c_1 = 2, c_2 = 5/8$. The other candidate extreme point is $c_1 = 1/2, c_2 = 1$, but $r_i^{(u)}$ is smaller there. In words, the optimal choice is to collect all available cheaper, noisier samples then spend the remaining budget on the more costly, higher quality samples.

The proof of Theorem 4.11 relies on the following lemma that generalizes the optimality of extreme points in linear programs (see, e.g., [24, Theorem 2.7]) to nonlinear objective functions for which each level set is a flat. A flat here refers to the solution set of an (underdetermined) linear system of equations, polyhedron means a finite intersection of half-spaces, and an extreme point of a set is a point that is not a convex combination of any other points in the set; see [24, Chapter 2] for further discussion and properties. We prove Lemma 4.15 in Section 4.11.3.

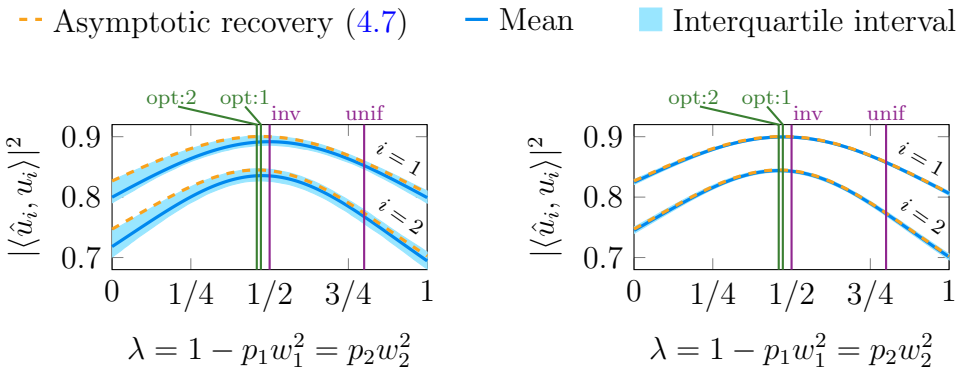
Lemma 4.15 (Optimality of extreme points)**.** *Let $P \subset \mathbb{R}^n$ be a polyhedron with at least one extreme point, and let $f : P \to \mathbb{R}$ be a continuous function such that each level set is a flat. If there exists a point in P that maximizes f, then there exists an extreme point of P that maximizes f.*

Proof of Theorem 4.11. Observe first that $c = c_1 + \cdots + c_L$ and $p_\ell = c_\ell/c$ for each $\ell \in \{1, \ldots, L\}$, so rewriting (4.37) yields that $r_i^{(u)}$ is the largest real value that satisfies

$$0 = R_i^{(u)}(r_i^{(u)}) = 1 - \sum_{\ell=1}^{L} c_\ell \frac{\theta_i^2}{\sigma_\ell^2} \frac{1 - r_i^{(u)}}{\sigma_\ell^2/\theta_i^2 + r_i^{(u)}}, \tag{4.54}$$

when the weights are set optimally. Thus, $r_i^{(u)}$ is a continuous function of $c_1, \ldots, c_L$ over the domain $c_1, \ldots, c_L \geq 0$ with level sets that are affine hyperplanes. The constraint set P defined by $c_1, \ldots, c_L \geq 0$ and (4.53) is a bounded polyhedron, so contains an extreme point as well as a maximizer of $r_i^{(u)}$. Thus, Lemma 4.15 implies that an extreme point of P maximizes $r_i^{(u)}$.

The final statement of the theorem follows by observing that the right hand side of (4.54) decreases when any one of $c_1, \ldots, c_L$ increases, increasing the resulting $r_i^{(u)}$. Namely, $r_i^{(u)}$ with optimal weighting improves when any of $c_1, \ldots, c_L$ increases. As a result, any point where $c_1, \ldots, c_L$ could be increased without violating (4.53) cannot be a maximizer. □



(a) 10^3 samples in 10^3 dimensions.

(b) 10^4 samples in 10^4 dimensions.

Figure 4.9: Simulated component recoveries $|\langle \hat{u}_i, u_i \rangle|^2$ for data generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise variance $\sigma_2^2 = 4$. Weights are set as $w_1^2 = (1-\lambda)/p_1$ and $w_2^2 = \lambda/p_2$. Simulation mean (blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic prediction (4.7) of Theorem 4.3 (orange dashed curve). Vertical lines indicate uniform weights (unif) for unweighted PCA, inverse noise variance weights (inv) and optimal weights (opt). Increasing the data size from (a) to (b) shrinks the interquartile intervals, indicating concentration to the mean, which is itself converging to the asymptotic recovery.

4.9 Numerical simulation

This section uses numerical simulations to demonstrate that the asymptotic results of Theorem 4.3 provide meaningful predictions for finitely many samples in finitely many dimensions. Data are generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise variance $\sigma_2^2 = 4$. Underlying scores and unscaled noise entries are both generated from the standard normal distribution, i.e., $z_{ij}, \varepsilon_{ij} \sim \mathcal{N}(0, 1)$, and the weights are set to $w_1^2 = (1 - \lambda)/p_1$ and $w_2^2 = \lambda/p_2$ where λ is swept from zero to one. Setting the weights in this way keeps the average weighting fixed at $p_1 w_1^2 + p_2 w_2^2 = 1$ and places using only samples with noise variance σ_1^2 at $\lambda = 0$ and using only samples with noise variance σ_2^2 at $\lambda = 1$. Unweighted PCA corresponds to uniform weights and occurs when $\lambda = p_2$, and inverse noise variance weights occurs when $\lambda = (p_2/\sigma_2^2)/(p_1/\sigma_1^2 + p_2/\sigma_2^2)$.

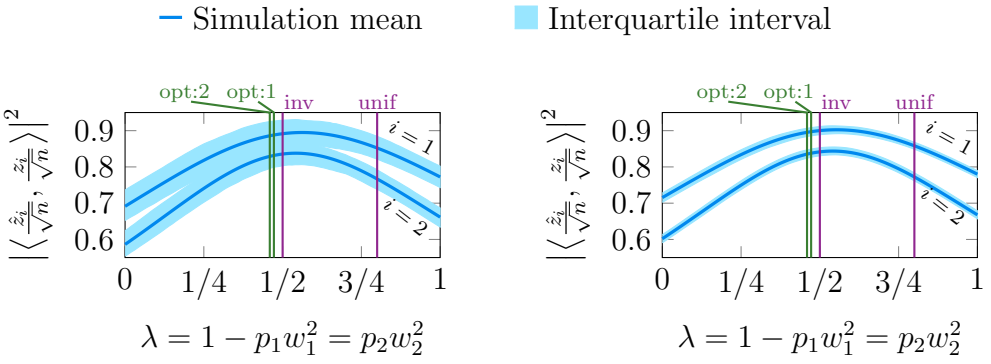
We carry out two simulations: the first has $n = 10^3$ samples in $d = 10^3$ dimensions, and the second increases these to $n = 10^4$ samples in $d = 10^4$ dimensions.

Both are repeated for 500 trials. Figure 4.9 plots the component recoveries $|\langle \hat{u}_i, u_i \rangle|^2$ for both simulations with the mean (blue curve) and interquartile interval (light blue ribbon) shown with the asymptotic component recovery (4.7) of Theorem 4.3 (orange dashed curve). Vertical lines denote uniform weights for unweighted PCA, inverse noise variance weights and optimal weights (4.36). Figure 4.9a illustrates general agreement in behavior between the non-asymptotic recovery and its asymptotic prediction. Though the asymptotic recovery is larger than the interquartile recovery, both have the same qualitative trend. In our experience, this phenomenon occurs in general. Figure 4.9b shows what happens when the number of samples and dimensions are increased. The interquartile intervals shrink dramatically, indicating concentration of each component recovery (a random quantity) around its mean. Furthermore, each mean component recovery closely matches the asymptotic recovery, indicating convergence to the limit. Convergence also appears to be faster for larger λ, i.e., where more weight is given to the larger set of samples. Characterizing non-asymptotic component recoveries is an important and challenging area of future work; the agreement here gives confidence that the asymptotic predictions provide meaningful insights for finite dimensions. In this setting, for example, it was significantly suboptimal to use unweighted PCA or to use only some of the samples, and using inverse noise variance weights was close to optimal. Section 4.11.4 shows analogous plots for the amplitudes $\hat{\theta}_i^2$, weighted score recoveries $|\langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle_{\mathbf{W}^2}|^2$ and products $\langle \hat{u}_i, u_i \rangle \langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle_{\mathbf{W}^2}^*$.

Figure 4.10 plots the unweighted score recoveries $|\langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle|^2$. Though Theorem 4.3 does not provide their asymptotic counterparts, one might expect that they have similar behavior to the component recoveries. Better component recoveries should generally lead to better score recoveries. Comparing with Fig. 4.9, the peak occurs for slightly larger λ indicating better performance when slightly more weight is given to the larger set of samples, but has an otherwise similar shape and trend, as well as statistical concentration. Hence, the asymptotic component recovery (4.7) of Theorem 4.3 also provides some insight into how well the underlying scores are recovered. Note that normalizing the weights to fix the average $p_1 w_1^2 + p_2 w_2^2$ is critical for these comparisons since, e.g., doubling the weights effectively halves the resulting scores and hence halves the resulting unweighted recoveries $|\langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle|^2$.

4.10 Discussion

This chapter analyzes weighted PCA in the high-dimensional asymptotic regime where both the number of samples n and ambient dimension d grow. We provide expressions for the asymptotic recovery of underlying amplitudes θ_i^2, components u_i



(a) 10^3 samples in 10^3 dimensions.

(b) 10^4 samples in 10^4 dimensions.

Figure 4.10: Simulated unweighted score recoveries $|\langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n}\rangle|^2$ for data generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise variance $\sigma_2^2 = 4$. Weights are set as $w_1^2 = (1-\lambda)/p_1$ and $w_2^2 = \lambda/p_2$. Simulation mean (blue curve) and interquartile interval (light blue ribbon) are shown with vertical lines indicating uniform weights (unif) that correspond to unweighted PCA, inverse noise variance weights (inv), and weights that optimize component recovery (opt). The peak score recovery shown here occurs at a slightly larger λ than the peak component recovery in Fig. 4.9, but they have otherwise similar behavior.

and scores z_i by the WPCA amplitudes $\hat{\theta}_i^2$, components $\hat{u}_i$ and scores $\hat{z}_i$. These expressions provide new insight into how weighting affects the performance of PCA, and also led to weights that optimize the recovery of an underlying component. We also use the analysis to investigate how to optimize sampling strategies under budget constraints.

An interesting avenue of future work is further study of the benefits of optimal weighting, e.g., to characterize when optimal weights provide significant benefit over inverse noise variance or square inverse noise variance weights. A benefit of such weights over the optimal choice is that they are independent of the underlying amplitude θ_i^2. Understanding the range of performance between inverse noise variance and square inverse noise variance weights might reveal simple choices for weights that are near-optimal for all components. Section 4.7.4 also demonstrated that including noisier data can degrade inverse noise variance weighted PCA, and it would be great to check if the same is true for square inverse noise variance weighted PCA. Some quick tests suggest that square inverse noise variance weights may in fact always improve given more data; the analysis of this chapter provides tools to answer this question more thoroughly.

Another interesting direction is to estimate the underlying amplitudes from an initial PCA of the data, e.g., using (4.4) in Theorem 4.3. Likewise, estimating noise variances could aid many important applications. The normalized squared norm $\|y_i\|_2^2/d$ of any single sample should concentrate around its noise variance in high dimensions since the signal component has asymptotically zero relative energy, so this is a reasonable candidate for estimating noise variances. Grouping samples into clusters of similar noise variances could also be used to improve the estimates, though this clustering can become challenging if the number of groups L grows with the number of samples n. Incorporating spectrum estimation methods such as [120, 141] is another promising approach, and one can further exploit knowledge of which samples share a noise variance by considering the spectrums of subsets of data. The noise spectrum might be isolated by dropping the first few singular values or by permuting the data as done in parallel analysis [57]; alternating between estimating components with weighted PCA and estimating noise variances can help mitigate interference from large principal components. Investigating these various approaches is ongoing and future work. This chapter's analysis can already quantify how much the performance of weighted PCA degrades when weights deviate from optimal, so it may help characterize the impact of errors in estimating the underlying amplitudes and noise variances.

Alternative approaches to finding the optimal weights could also be interesting. This chapter analyzes the asymptotic recovery first then optimizes that *deterministic* quantity. Another approach could be to try to optimize the random non-asymptotic recovery, perhaps by some kind of leave-one-out analysis, resulting in *random* weights that we then attempt to show converge almost surely to deterministic weights.

Finally, extending the analysis here to more general forms of weighted PCA is an important and nontrivial direction. In particular, one might consider weighting that is across variables in addition to across samples, e.g., to handle heterogeneous amounts of noise among the variables. Such analysis could also provide insight into the common preprocessing step of standardizing the variables to have unit variance. One might also consider a variant of (4.1) with a general weighted orthonormality in place of $\mathbf{W}^2$. Developing and studying alternative ways to account for heteroscedasticity in PCA is another avenue for future work. For example, one might consider a probabilistic PCA [195] approach that accounts for heteroscedasticity; the nonuniform noise variances complicate the resulting optimization, making algorithm development for this approach nontrivial and interesting. Generally speaking, considering broader types of heterogeneity is an important area of future work. Increasingly, data from multiple sources are combined to find latent phenomenon so investigating how to fully utilize the available data is important both for furthering our understanding

and for developing practical guidelines.

4.11 Supplementary material

4.11.1 Proof of Theorem 4.3

The model (4.3) for the data matrix $\mathbf{Y} := (y_1, \dots, y_n) \in \mathbb{C}^{d\times n}$ is the low-rank perturbation of a random matrix

$$\begin{aligned}(4.55)\quad \mathbf{Y} &= \underbrace{(u_1,\dots,u_k)}_{\mathbf{U}\in\mathbb{C}^{d\times k}}\underbrace{\operatorname{diag}(\theta_1,\dots,\theta_k)}_{\mathbf{\Theta}\in\mathbb{R}^{k\times k}}\underbrace{(z_1,\dots,z_k)^{\mathsf{H}}}_{\mathbf{Z}\in\mathbb{C}^{n\times k}} + \underbrace{(\varepsilon_1,\dots,\varepsilon_n)}_{\mathbf{E}\in\mathbb{C}^{d\times n}}\underbrace{\operatorname{diag}(\eta_1,\dots,\eta_n)}_{\mathbf{H}\in\mathbb{R}^{n\times n}} \\ &= \mathbf{U}\mathbf{\Theta}\mathbf{Z}^{\mathsf{H}} + \mathbf{E}\mathbf{H},\end{aligned}$$

The weighted PCA components $\hat{u}_1, \dots, \hat{u}_k$, amplitudes $\hat{\theta}_1, \dots, \hat{\theta}_k$, and normalized weighted scores $\mathbf{W}\hat{z}_1/\sqrt{n}, \dots, \mathbf{W}\hat{z}_k/\sqrt{n}$ are, respectively, principal left singular vectors, singular values, and right singular vectors of the normalized and weighted data matrix

$$(4.56)\qquad \tilde{\mathbf{Y}} := \frac{1}{\sqrt{n}}\mathbf{Y}\underbrace{\operatorname{diag}(\omega_1^2,\dots,\omega_n^2)}_{\mathbf{W}\in\mathbb{R}^{n\times n}} = \mathbf{U}\mathbf{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W} + \tilde{\mathbf{E}},$$

where $\tilde{\mathbf{Z}} := \mathbf{Z}/\sqrt{n}$ are normalized underlying scores and $\tilde{\mathbf{E}} := \mathbf{E}\mathbf{H}\mathbf{W}/\sqrt{n}$ are normalized and weighted noise.

Without loss of generality, suppose that the components $\mathbf{U} := (u_1, \dots, u_k)$ are randomly generated according to the "orthonormalized model" of [22, Section 2.1]; since the noise vectors are unitarily invariant, this assumption is equivalent to considering a random rotation of data from a deterministic $\mathbf{U}$ as done in Section 3.5.1. The normalized scores $\tilde{\mathbf{Z}} = (z_1/\sqrt{n}, \dots, z_k/\sqrt{n})$ are generated according to the "iid model" of [22, Section 2.1], and $\mathbf{E}$ has iid entries with zero mean, unit variance and bounded fourth moment. Finally, $\mathbf{HW}$ is a non-random diagonal nonnegative definite matrix with bounded spectral norm and limiting eigenvalue distribution $p_1\delta_{w_1^2\sigma_1^2} + \cdots + p_L\delta_{w_L^2\sigma_L^2}$, where δ_x denotes the Dirac delta distribution at x.

A roadmap for the proof is as follows:

1. State some preliminary results on $\tilde{\mathbf{E}}$ that, taken with Lemma 4.9, provide a foundation for the remainder of the analysis.

2. Extend [22, Theorem 2.9] to find asymptotic weighted PCA amplitudes.

3. Extend [22, Theorem 2.10] to find asymptotic component recovery and asymptotic weighted score recovery.

4. Similar to Sections 3.5.2 to 3.5.6, find algebraic descriptions for the expressions derived in Section 4.11.1.2 and Section 4.11.1.3. The original expressions are challenging to evaluate and analyze.

Unless otherwise specified, limits are as $n, d \to \infty$. Lemma 4.9 was crucial to carrying out the above extensions, and its proof (Section 4.11.2) is one of our main technical contributions.

4.11.1.1 Preliminary results on $\tilde{\mathbf{E}}$

The normalized and weighted noise matrix $\tilde{\mathbf{E}}$ fits within the random matrix model studied in [14, Chapters 4, 6]. In particular, from [14, Theorem 4.3] and [14, Corollary 6.6] we conclude that the singular value distribution of $\tilde{\mathbf{E}}$ converges weakly almost surely to a nonrandom compactly supported measure $\mu_{\tilde{\mathbf{E}}}$, and the largest singular value of $\tilde{\mathbf{E}}$ converges almost surely to the supremum b of the support of $\mu_{\tilde{\mathbf{E}}}$.

It follows then that

$$\frac{1}{d}\operatorname{tr}\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1} \xrightarrow{\text{a.s.}} \varphi_1(\zeta) := \int \frac{\zeta}{\zeta^2-t^2} d\mu_{\tilde{\mathbf{E}}}(t), \tag{4.57}$$

where the convergence is uniform on $\{\zeta \in \mathbb{C} : \Re(\zeta) > b+\tau\}$ for any $\tau > 0$. Furthermore, for any $\zeta \in \mathbb{C}$ with $\Re(\zeta) > b$,

$$\frac{\partial}{\partial\zeta}\frac{1}{d}\operatorname{tr}\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1} \xrightarrow{\text{a.s.}} \varphi_1'(\zeta). \tag{4.58}$$

We conclude the preliminaries by verifying some properties of φ_1.

a) For any $\zeta > b$, the integrand in (4.57) is positive and bounded away from zero since the support of $\mu_{\tilde{\mathbf{E}}}$ lies between zero and b.
Thus, $\forall_{\zeta>b}\ \varphi_1(\zeta) > 0$.

b) As $|\zeta| \to \infty$, the integrand in (4.57) goes to zero uniformly in t.
Thus, $\varphi_1(\zeta) \to 0$ as $|\zeta| \to \infty$.

c) The imaginary part of $\varphi_1(\zeta)$ is

$$\Im\{\varphi_1(\zeta)\} = \int \Im\Big(\frac{\zeta}{\zeta^2-t^2}\Big) d\mu_{\tilde{\mathbf{E}}}(t) = -\Im(\zeta) \underbrace{\int \frac{|\zeta|^2+t^2}{|\zeta^2-t^2|^2} d\mu_{\tilde{\mathbf{E}}}(t)}_{>0}.$$

Thus, $\varphi_1(\zeta) \in \mathbb{R} \Leftrightarrow \zeta \in \mathbb{R}$.

Lemma 4.9 establishes the analogous results for the weighted trace.

4.11.1.2 Largest singular values

This section extends [22, Theorem 2.9] to find the limiting largest singular values of the weighted matrix $\tilde{\mathbf{Y}}$ in (4.56). As in [22, Section 4], $\liminf \hat{\theta}_i \geq b$ almost surely for each $i \in \{1, \ldots, k\}$ so we focus on singular values larger than $b + \tau$ where $\tau > 0$ is arbitrary. The following lemma generalizes [22, Lemma 4.1] to account for the weights.

Lemma 4.16. *Let $\zeta > 0$ be arbitrary but not a singular value of $\tilde{\mathbf{E}}$. Then ζ is a singular value of $\tilde{\mathbf{Y}} = \mathbf{U}\boldsymbol{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W} + \tilde{\mathbf{E}}$ if and only if the following matrix is singular:*

$$\mathbf{M}(\zeta) := \begin{bmatrix} \mathbf{U}^{\mathsf{H}}\zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}\mathbf{U} & \mathbf{U}^{\mathsf{H}}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}\tilde{\mathbf{E}}\mathbf{W}\tilde{\mathbf{Z}} \\ \tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}\tilde{\mathbf{E}}^{\mathsf{H}}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}\mathbf{U} & \tilde{\mathbf{Z}}^{\mathsf{H}}\zeta\mathbf{W}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W}\tilde{\mathbf{Z}} \end{bmatrix} - \begin{bmatrix} & \boldsymbol{\Theta}^{-1} \\ \boldsymbol{\Theta}^{-1} & \end{bmatrix}. \tag{4.59}$$

Lemma 4.16 is proved in the same way as [22, Lemma 4.1] but with the weights incorporated; for convenience, we state it here with some additional detail.

Proof of Lemma 4.16. By [100, Theorem 7.3.3], ζ is a singular value of $\tilde{\mathbf{Y}}$ if and only if it is a root of the characteristic polynomial

$$0 = \det\left\{\zeta\mathbf{I} - \begin{pmatrix} & \tilde{\mathbf{Y}} \\ \tilde{\mathbf{Y}}^{\mathsf{H}} & \end{pmatrix}\right\} \tag{4.60}$$

$$= \det\left\{\zeta\mathbf{I} - \begin{pmatrix} & \tilde{\mathbf{E}} \\ \tilde{\mathbf{E}}^{\mathsf{H}} & \end{pmatrix} - \begin{pmatrix} \mathbf{U} & \\ & \mathbf{W}\tilde{\mathbf{Z}} \end{pmatrix}\begin{pmatrix} & \boldsymbol{\Theta} \\ \boldsymbol{\Theta} & \end{pmatrix}\begin{pmatrix} \mathbf{U} & \\ & \mathbf{W}\tilde{\mathbf{Z}} \end{pmatrix}^{\mathsf{H}}\right\} \tag{4.61}$$

$$= \det\left\{\zeta\mathbf{I} - \begin{pmatrix} & \tilde{\mathbf{E}} \\ \tilde{\mathbf{E}}^{\mathsf{H}} & \end{pmatrix}\right\}\det\begin{pmatrix} & \boldsymbol{\Theta} \\ \boldsymbol{\Theta} & \end{pmatrix}\det\{-\mathbf{M}(\zeta)\}, \tag{4.62}$$

where (4.61) is a convenient form of the matrix, and (4.62) follows from the determinant identity

$$\det(\mathbf{A} - \mathbf{BDC}) = \det(\mathbf{A})\det(\mathbf{D})\det(\mathbf{D}^{-1} - \mathbf{CA}^{-1}\mathbf{B}), \tag{4.63}$$

for invertible matrices $\mathbf{A}$ and $\mathbf{D}$ and the block matrix inverse [100, Equation (0.7.3.1)]

$$\left\{\zeta\mathbf{I} - \begin{pmatrix} & \tilde{\mathbf{E}} \\ \tilde{\mathbf{E}}^{\mathsf{H}} & \end{pmatrix}\right\}^{-1} = \begin{Bmatrix} \zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1} & (\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}\tilde{\mathbf{E}} \\ \tilde{\mathbf{E}}^{\mathsf{H}}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1} & \zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1} \end{Bmatrix}. \tag{4.64}$$

Note that (4.64) is invertible because ζ is not a singular value of $\tilde{\mathbf{E}}$. As a further consequence, (4.62) is zero exactly when $\mathbf{M}(\zeta)$ is singular. □

Applying Ascoli's theorem, [22, Proposition A.2], (4.57) and Lemma 4.9 yields

$$\mathbf{M}(\zeta) \xrightarrow{\text{a.s.}} \tilde{\mathbf{M}}(\zeta) := \begin{pmatrix} \varphi_1(\zeta)\mathbf{I}_k & \\ & \varphi_2(\zeta)\mathbf{I}_k \end{pmatrix} - \begin{pmatrix} & \mathbf{\Theta}^{-1} \\ \mathbf{\Theta}^{-1} & \end{pmatrix}, \tag{4.65}$$

where the convergence is uniform on $\{\zeta \in \mathbb{C} : \Re(\zeta) > b + \tau\}$. Finally, applying [22, Lemma A.1] in the same way as [22, Section 4] yields

$$\hat{\theta}_i^2 \xrightarrow{\text{a.s.}} \begin{cases} \rho_i^2 & \text{if } \theta_i^2 > \bar{\theta}^2, \\ b^2 & \text{otherwise,} \end{cases} =: r_i^{(\theta)} \tag{4.66}$$

where $D(\zeta) := \varphi_1(\zeta)\varphi_2(\zeta)$ for $\zeta > b$, $\rho_i := D^{-1}(1/\theta_i^2)$, $\bar{\theta}^2 := 1/D(b^+)$, and $f(b^+) := \lim_{\zeta \to b^+} f(\zeta)$ denotes a limit from above.

4.11.1.3 Recovery of singular vectors

This section extends [22, Theorem 2.10] to find the limiting recovery of singular vectors. Suppose $\theta_i > \bar{\theta}$. Then $\hat{\theta}_i \xrightarrow{\text{a.s.}} \rho_i > b$ and so, almost surely, $\hat{\theta}_i > \|\tilde{\mathbf{E}}\|$ eventually. Namely, $\hat{\theta}_i$ is almost surely eventually not a singular value of $\tilde{\mathbf{E}}$. The following lemma generalizes [22, Lemma 5.1] to account for the weights.

Lemma 4.17. *Suppose $\hat{\theta}_i$ is not a singular value of $\tilde{\mathbf{E}}$. Then*

$$\mathbf{M}(\hat{\theta}_i) \begin{pmatrix} \mathbf{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}^2\hat{z}_i/\sqrt{n} \\ \mathbf{\Theta}\mathbf{U}^{\mathsf{H}}\hat{u}_i \end{pmatrix} = 0, \tag{4.67}$$

and

$$1 = \chi_1 + \chi_2 + 2\Re(\chi_3), \tag{4.68}$$

where $\mathbf{\Gamma} := (\hat{\theta}_i^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}$ and

$$\begin{aligned} \chi_1 &:= \sum_{j_1,j_2=1}^{k} \theta_{j_1}\theta_{j_2} \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_{j_1}}{\sqrt{n}} \right\rangle_{\mathbf{W}^2} \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_{j_2}}{\sqrt{n}} \right\rangle_{\mathbf{W}^2}^{*} u_{j_2}^{\mathsf{H}}\hat{\theta}_i^2\mathbf{\Gamma}^2 u_{j_1}, \\ \chi_2 &:= \sum_{j_1,j_2=1}^{k} \theta_{j_1}\theta_{j_2}\langle \hat{u}_i, u_{j_1}\rangle\langle \hat{u}_i, u_{j_2}\rangle^{*} \tilde{z}_{j_2}^{\mathsf{H}}\mathbf{W}\tilde{\mathbf{E}}^{\mathsf{H}}\mathbf{\Gamma}^2\tilde{\mathbf{E}}\mathbf{W}\tilde{z}_{j_1}, \\ \chi_3 &:= \sum_{j_1,j_2=1}^{k} \theta_{j_1}\theta_{j_2}\langle \hat{u}_i, u_{j_1}\rangle \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_{j_2}}{\sqrt{n}} \right\rangle_{\mathbf{W}^2}^{*} u_{j_2}^{\mathsf{H}}\hat{\theta}_i\mathbf{\Gamma}^2\tilde{\mathbf{E}}\mathbf{W}\tilde{z}_{j_1}. \end{aligned} \tag{4.69}$$

Lemma 4.17 is proved in the same way as [22, Lemma 5.1] but with the weights incorporated.

Proof of Lemma 4.17. Let $\tilde{\mathbf{X}} := \mathbf{U}\boldsymbol{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}$ be the weighted and normalized underlying data. Substituting (4.59) into (4.67) and factoring yields

$$\mathbf{M}(\hat{\theta}_i)\begin{bmatrix}\boldsymbol{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}^2\hat{z}_i/\sqrt{n}\\ \boldsymbol{\Theta}\mathbf{U}^{\mathsf{H}}\hat{u}_i\end{bmatrix} \tag{4.70}$$

$$= \begin{bmatrix}\mathbf{U}^{\mathsf{H}}\{(\hat{\theta}_i^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}(\hat{\theta}_i\tilde{\mathbf{X}}\mathbf{W}\hat{z}_i/\sqrt{n}+\tilde{\mathbf{E}}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i)-\hat{u}_i\}\\ \tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}\{(\hat{\theta}_i^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}(\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{X}}\mathbf{W}\hat{z}_i/\sqrt{n}+\hat{\theta}_i\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i)-\mathbf{W}\hat{z}_i/\sqrt{n}\}\end{bmatrix} \tag{4.71}$$

$$= \begin{bmatrix}\mathbf{U}^{\mathsf{H}}(\hat{u}_i-\hat{u}_i)\\ \tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}(\mathbf{W}\hat{z}_i/\sqrt{n}-\mathbf{W}\hat{z}_i/\sqrt{n})\end{bmatrix} = 0 \tag{4.72}$$

where (4.71) uses the matrix identity $\tilde{\mathbf{E}}^{\mathsf{H}}(\hat{\theta}_i^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1} = (\hat{\theta}_i^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\tilde{\mathbf{E}}^{\mathsf{H}}$, and (4.72) follows by substituting $\tilde{\mathbf{X}} = \tilde{\mathbf{Y}} - \tilde{\mathbf{E}}$ and using the singular vector identities

$$\tilde{\mathbf{Y}}\mathbf{W}\hat{z}_i/\sqrt{n} = \hat{\theta}_i\hat{u}_i, \qquad \tilde{\mathbf{Y}}^{\mathsf{H}}\hat{u}_i = \hat{\theta}_i\mathbf{W}\hat{z}_i/\sqrt{n}. \tag{4.73}$$

To obtain (4.68), reuse the identity $\hat{u}_i = (\hat{\theta}_i^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}(\hat{\theta}_i\tilde{\mathbf{X}}\mathbf{W}\hat{z}_i/\sqrt{n}+\tilde{\mathbf{E}}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i)$ used to obtain (4.72) and expand as

$$\begin{aligned}1 &= \hat{u}_i^{\mathsf{H}}\hat{u}_i\\ &= \left(\hat{\theta}_i\tilde{\mathbf{X}}\mathbf{W}\frac{\hat{z}_i}{\sqrt{n}}+\tilde{\mathbf{E}}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i\right)^{\mathsf{H}}(\hat{\theta}_i^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-2}\left(\hat{\theta}_i\tilde{\mathbf{X}}\mathbf{W}\frac{\hat{z}_i}{\sqrt{n}}+\tilde{\mathbf{E}}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i\right)\\ &= \chi_1+\chi_2+2\Re(\chi_3),\end{aligned} \tag{4.74}$$

where the outer terms are

$$\chi_1 := \frac{\hat{z}_i^{\mathsf{H}}}{\sqrt{n}}\mathbf{W}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{\theta}_i^2\boldsymbol{\Gamma}^2\tilde{\mathbf{X}}\mathbf{W}\frac{\hat{z}_i}{\sqrt{n}}, \qquad \chi_2 := \hat{u}_i^{\mathsf{H}}\tilde{\mathbf{X}}\tilde{\mathbf{E}}^{\mathsf{H}}\boldsymbol{\Gamma}^2\tilde{\mathbf{E}}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i, \tag{4.75}$$

and the cross term is

$$\chi_3 := \frac{\hat{z}_i^{\mathsf{H}}}{\sqrt{n}}\mathbf{W}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{\theta}_i\boldsymbol{\Gamma}^2\tilde{\mathbf{E}}\tilde{\mathbf{X}}^{\mathsf{H}}\hat{u}_i. \tag{4.76}$$

Expanding $\tilde{\mathbf{X}} = \mathbf{U}\boldsymbol{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W} = \theta_1u_1(z_1/\sqrt{n})^{\mathsf{H}}\mathbf{W}+\cdots+\theta_ku_k(z_k/\sqrt{n})^{\mathsf{H}}\mathbf{W}$ in the terms (4.75)–(4.76) and simplifying yields (4.69). □

Applying the convergence $\mathbf{M}(\hat{\theta}_i) \xrightarrow{\text{a.s.}} \tilde{\mathbf{M}}(\rho_i)$ to (4.67) in Lemma 4.17 yields

$$\begin{pmatrix}\xi\\ \delta\end{pmatrix} := \mathrm{proj}_{\{\ker\tilde{\mathbf{M}}(\rho_i)\}^\perp}\begin{pmatrix}\boldsymbol{\Theta}\tilde{\mathbf{Z}}^{\mathsf{H}}\mathbf{W}^2\hat{z}_i/\sqrt{n}\\ \boldsymbol{\Theta}\mathbf{U}^{\mathsf{H}}\hat{u}_i\end{pmatrix} \xrightarrow{\text{a.s.}} 0. \tag{4.77}$$

Observe next that, similar to [22, Section 5],

$$\ker\tilde{\mathbf{M}}(\rho_i) = \left\{\begin{pmatrix}s\\ t\end{pmatrix}\in\mathbb{C}^{2k} : \begin{array}{ll} t_j = s_j = 0 & \text{if } \theta_j\neq\theta_i\\ t_j = \theta_i\varphi_1(\rho_i)s_j & \text{if } \theta_j=\theta_i\end{array}\right\}, \tag{4.78}$$

so the projection entries are

$$\begin{pmatrix}\xi_j\\ \delta_j\end{pmatrix} = \theta_j \begin{pmatrix}\langle \hat{z}_i/\sqrt{n}, z_j/\sqrt{n}\rangle_{\mathbf{W}^2}\\ \langle \hat{u}_i, u_j\rangle\end{pmatrix}, \tag{4.79}$$

for j such that $\theta_j \neq \theta_i$, and

$$\begin{pmatrix}\xi_j\\ \delta_j\end{pmatrix} = \left\{\theta_i\varphi_1(\rho_i)\left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2} - \langle \hat{u}_i, u_j\rangle\right\}\frac{\theta_i}{\theta_i^2\varphi_1^2(\rho_i)+1}\begin{pmatrix}\theta_i\varphi_1(\rho_i)\\ -1\end{pmatrix}, \tag{4.80}$$

for j such that $\theta_j = \theta_i$. Applying the convergence (4.77) to (4.79) yields

$$\sum_{j:\theta_j\neq\theta_i} |\langle \hat{u}_i, u_j\rangle|^2 + \left|\left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2}\right|^2 \xrightarrow{\text{a.s.}} 0, \tag{4.81}$$

and applying the convergence (4.77) to (4.80) yields

$$\sum_{j:\theta_j=\theta_i} \left|\sqrt{\frac{\varphi_1(\rho_i)}{\varphi_2(\rho_i)}}\left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2} - \langle \hat{u}_i, u_j\rangle\right|^2 \xrightarrow{\text{a.s.}} 0, \tag{4.82}$$

recalling that $D(\rho_i) = \varphi_1(\rho_i)\varphi_2(\rho_i) = 1/\theta_i^2$.

Turning now to (4.68) in Lemma 4.17, note that applying [22, Proposition A.2] yields the convergence $\chi_3 \xrightarrow{\text{a.s.}} 0$ as well as the almost sure convergence to zero of the summands in (4.69) for χ_1 and χ_2 for which $j_1 \neq j_2$. By (4.81), the summands for which $\theta_{j_1}, \theta_{j_2} \neq \theta_i$ also converge almost surely to zero. Furthermore, by (4.58) and Lemma 4.9

$$\begin{aligned}\frac{1}{d}\operatorname{tr}\hat{\theta}_i^2\mathbf{\Gamma}^2 &= \frac{1}{d}\operatorname{tr}\zeta^2(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-2}\Big|_{\zeta=\hat{\theta}_i}\\ &= \left(\frac{1}{2\zeta} - \frac{1}{2}\frac{\partial}{\partial\zeta}\right)\left\{\frac{1}{d}\operatorname{tr}\zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}\right\}\Big|_{\zeta=\hat{\theta}_i}\\ &\xrightarrow{\text{a.s.}} \frac{\varphi_1(\rho_i)}{2\rho_i} - \frac{\varphi_1'(\rho_i)}{2},\end{aligned} \tag{4.83}$$

$$\begin{aligned}\frac{1}{n}\operatorname{tr}\mathbf{W}\tilde{\mathbf{E}}^{\mathsf{H}}\mathbf{\Gamma}^2\tilde{\mathbf{E}}\mathbf{W} &= \frac{1}{n}\operatorname{tr}\mathbf{W}\tilde{\mathbf{E}}^{\mathsf{H}}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-2}\tilde{\mathbf{E}}\mathbf{W}\Big|_{\zeta=\hat{\theta}_i}\\ &= \left(-\frac{1}{2\zeta} - \frac{1}{2}\frac{\partial}{\partial\zeta}\right)\left\{\frac{1}{n}\operatorname{tr}\zeta\mathbf{W}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W}\right\}\Big|_{\zeta=\hat{\theta}_i}\\ &\xrightarrow{\text{a.s.}} -\frac{\varphi_2(\rho_i)}{2\rho_i} - \frac{\varphi_2'(\rho_i)}{2},\end{aligned} \tag{4.84}$$

so applying [22, Proposition A.2] once more we have

$$\chi_1 = \theta_i^2\Big\{\frac{\varphi_1(\rho_i)}{2\rho_i} - \frac{\varphi_1'(\rho_i)}{2}\Big\}\sum_{j:\theta_j=\theta_i}\left|\left\langle\frac{\hat{z}_i}{\sqrt{n}},\frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2}\right|^2 + o(1), \tag{4.85}$$

$$\chi_2 = \theta_i^2\Big\{-\frac{\varphi_2(\rho_i)}{2\rho_i} - \frac{\varphi_2'(\rho_i)}{2}\Big\}\sum_{j:\theta_j=\theta_i}|\langle\hat{u}_i,u_j\rangle|^2 + o(1),$$

where $o(1)$ denotes a sequence that almost surely converges to zero. Combining (4.68), (4.82) and (4.85) yields

$$1 = -\frac{\theta_i^2 D'(\rho_i)}{2\varphi_1(\rho_i)}\sum_{j:\theta_j=\theta_i}|\langle\hat{u}_i,u_j\rangle|^2 + o(1), \tag{4.86}$$

$$1 = -\frac{\theta_i^2 D'(\rho_i)}{2\varphi_2(\rho_i)}\sum_{j:\theta_j=\theta_i}\left|\left\langle\frac{\hat{z}_i}{\sqrt{n}},\frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2}\right|^2 + o(1),$$

where we use the fact that $D'(\zeta) = \varphi_1'(\zeta)\varphi_2(\zeta) + \varphi_1(\zeta)\varphi_2'(\zeta)$. Solving (4.86) for the recoveries and recalling (4.81) yields

$$\sum_{j:\theta_j=\theta_i}|\langle\hat{u}_i,u_j\rangle|^2 \xrightarrow{\text{a.s.}} \frac{-2\varphi_1(\rho_i)}{\theta_i^2 D'(\rho_i)} =: r_i^{(u)}, \tag{4.87}$$

$$\sum_{j:\theta_j=\theta_i}\left|\left\langle\frac{\hat{z}_i}{\sqrt{n}},\frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2}\right|^2 \xrightarrow{\text{a.s.}} \frac{-2\varphi_2(\rho_i)}{\theta_i^2 D'(\rho_i)} =: r_i^{(z)}, \tag{4.88}$$

$$\sum_{j:\theta_j\neq\theta_i}|\langle\hat{u}_i,u_j\rangle|^2, \sum_{j:\theta_j\neq\theta_i}\left|\left\langle\frac{\hat{z}_i}{\sqrt{n}},\frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2}\right|^2 \xrightarrow{\text{a.s.}} 0. \tag{4.89}$$

Furthermore, combining (4.82) and (4.86) yields

$$\sum_{j:\theta_j=\theta_i}\langle\hat{u}_i,u_j\rangle\left\langle\frac{\hat{z}_i}{\sqrt{n}},\frac{z_j}{\sqrt{n}}\right\rangle_{\mathbf{W}^2}^* \xrightarrow{\text{a.s.}} \frac{-2\varphi_1(\rho_i)}{\theta_i^2 D'(\rho_i)}\sqrt{\frac{\varphi_2(\rho_i)}{\varphi_1(\rho_i)}} = \sqrt{r_i^{(u)}r_i^{(z)}}. \tag{4.90}$$

4.11.1.4 Algebraic description

This section concludes the proof by finding algebraic descriptions of the almost sure limits (4.66), (4.87)–(4.88) and (4.90). As in Section 3.5.2, we change variables to

$$\psi(\zeta) := \frac{c\zeta}{\varphi_1(\zeta)} = \left\{\frac{1}{c}\int\frac{d\mu_{\tilde{\mathbf{E}}}(t)}{\zeta^2 - t^2}\right\}^{-1}, \tag{4.91}$$

and observe that analogously to Section 3.5.3 ψ has the properties:

a) $0 = Q(\psi(\zeta), \zeta)$ for all $\zeta > b$ where

$$Q(s,\zeta) := \frac{c\zeta^2}{s^2} + \frac{c-1}{s} - c\sum_{\ell=1}^{L} \frac{p_\ell}{s - w_\ell^2\sigma_\ell^2}, \tag{4.92}$$

and the inverse function is given by

$$\psi^{-1}(x) = \sqrt{\frac{x}{c}\left(1 + c\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2\sigma_\ell^2}{x - w_\ell^2\sigma_\ell^2}\right)} = \sqrt{\frac{xC(x)}{c}}, \tag{4.93}$$

where C is defined in (4.6);

b) $\max_\ell(w_\ell^2\sigma_\ell^2) < \psi(\zeta) < c\zeta^2$;

c) $0 < \psi(b^+) < \infty$ and $\psi'(b^+) = \infty$.

Expressing D in terms of ψ yields

$$D(\zeta) = \varphi_1(\zeta)\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{\zeta - w_\ell^2\sigma_\ell^2\varphi_1(\zeta)/c} = c\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{\psi(\zeta) - w_\ell^2\sigma_\ell^2} = \frac{1 - B_i(\psi(\zeta))}{\theta_i^2}, \tag{4.94}$$

and

$$\frac{D'(\zeta)}{\zeta} = -\frac{c\psi'(\zeta)}{\zeta}\sum_{\ell=1}^{L} \frac{p_\ell w_\ell^2}{\{\psi(\zeta) - w_\ell^2\sigma_\ell^2\}^2} = -\frac{2c}{\theta_i^2}\frac{B_i'(\psi(\zeta))}{A(\psi(\zeta))}, \tag{4.95}$$

where A and B_i are defined in (4.5) and the second equality in (4.95) follows analogously to Section 3.5.4 by deriving the identity

$$\psi'(\zeta) = \frac{2c\zeta}{A(\psi(\zeta))}, \tag{4.96}$$

from Property (a) then simplifying.

Rearranging (4.96) then applying Property (c) yields

$$A(\psi(b^+)) = \frac{2cb}{\psi'(b^+)} = 0, \tag{4.97}$$

so $\psi(b^+)$ is a root of A. If $\theta_i^2 > \bar{\theta}^2$, then $\rho_i = D^{-1}(1/\theta_i^2)$ and rearranging (4.94) yields

$$B_i(\psi(\rho_i)) = 1 - \theta_i^2 D(\rho_i) = 0, \tag{4.98}$$

so $\psi(\rho_i)$ is a root of B_i. Recall that $\psi(b^+), \psi(\rho_i) \geq \max_\ell(w_\ell^2\sigma_\ell^2)$ by Property (b), and observe that both $A(x)$ and $B_i(x)$ monotonically increase for $x > \max_\ell(w_\ell^2\sigma_\ell^2)$ from negative infinity to one. Thus, each has exactly one real root larger than $\max_\ell(w_\ell^2\sigma_\ell^2)$,

i.e., its largest real root, and so $\psi(b^+) = \alpha$ and $\psi(\rho_i) = \beta_i$ when $\theta_i^2 > \bar{\theta}^2$, where α and β_i are the largest real roots of A and B_i, respectively.

Even though $\psi(\rho_i)$ is defined only when $\theta_i^2 > \bar{\theta}^2$, the largest real roots α and β are always defined and always larger than $\max_\ell(w_\ell^2\sigma_\ell^2)$. Thus

$$\theta_i^2 > \bar{\theta}^2 = \frac{1}{D(b^+)} = \frac{\theta_i^2}{1 - B_i(\psi(b^+))} \;\Leftrightarrow\; B_i(\alpha) < 0 \;\Leftrightarrow\; \alpha < \beta_i \;\Leftrightarrow\; A(\beta_i) > 0 \tag{4.99}$$

where the final equivalence holds because $A(x)$ and $B_i(x)$ are both strictly increasing functions for $x > \max_\ell(w_\ell^2\sigma_\ell^2)$ and $A(\alpha) = B_i(\beta_i) = 0$.

Using the inverse function (4.93) in Property (a) and (4.99), write (4.66) as

$$r_i^{(\theta)} = \begin{cases} \{\psi^{-1}(\psi(\rho_i))\}^2 & \text{if } \theta_i^2 > \bar{\theta}^2, \\ \{\psi^{-1}(\psi(b))\}^2 & \text{otherwise,} \end{cases} = \begin{cases} \beta_i C(\beta_i)/c & \text{if } \alpha < \beta_i, \\ \alpha C(\alpha)/c & \text{otherwise.} \end{cases} \tag{4.100}$$

Using max to be succinct yields (4.4). Likewise, rewrite (4.87) and (4.88) using ψ and (4.95), obtaining

$$r_i^{(u)} = \frac{-2\varphi_1(\rho_i)}{\theta_i^2 D'(\rho_i)} = \frac{1}{\psi(\rho_i)} \frac{A(\psi(\rho_i))}{B_i'(\psi(\rho_i))} = \frac{1}{\beta_i} \frac{A(\beta_i)}{B_i'(\beta_i)}, \tag{4.101}$$

$$\begin{aligned} r_i^{(z)} &= \frac{-2\varphi_2(\rho_i)}{\theta_i^2 D'(\rho_i)} = \frac{\varphi_2(\rho_i)}{c\rho_i} \frac{A(\psi(\rho_i))}{B_i'(\psi(\rho_i))} = \frac{\varphi_1(\rho_i)\varphi_2(\rho_i)}{c\rho_i\varphi_1(\rho_i)} \frac{A(\psi(\rho_i))}{B_i'(\psi(\rho_i))} \\ &= \frac{\psi(\rho_i)}{c^2\theta_i^2\rho_i^2} \frac{A(\psi(\rho_i))}{B_i'(\psi(\rho_i))} = \frac{1}{c\theta_i^2 C(\beta_i)} \frac{A(\psi(\rho_i))}{B_i'(\psi(\rho_i))}, \end{aligned} \tag{4.102}$$

and combine with (4.89) to obtain (4.7)–(4.8). Taking the geometric mean likewise yields (4.9) as an algebraic description of the almost sure limit (4.90). □

4.11.2 Proof of Lemma 4.9

Unless otherwise specified, limits are as $n, d \to \infty$. Consider the expansion

$$\frac{1}{n} \operatorname{tr} \zeta \mathbf{W}(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W} = \sum_{\ell=1}^{L} \frac{n_\ell}{n} w_\ell^2 \left\{ \frac{1}{n_\ell} \operatorname{tr} \boldsymbol{\Delta}_\ell(\zeta) \right\} \tag{4.103}$$

where $\boldsymbol{\Delta}_\ell(\zeta) \in \mathbb{C}^{n_\ell \times n_\ell}$ is the ℓth diagonal block of $\zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}$. The proof proceeds as follows:

1. Prove that for any fixed $\zeta = r + \imath s \in \mathbb{C}$ with $r, s \neq 0$,

$$\frac{1}{n_\ell} \operatorname{tr} \boldsymbol{\Delta}_\ell(\zeta) \xrightarrow{\text{a.s.}} \mathbb{E} \frac{1}{n_\ell} \operatorname{tr} \boldsymbol{\Delta}_\ell(\zeta). \tag{4.104}$$

2. Prove that for any fixed $\zeta = r + \imath s \in \mathbb{C}$ with $r, s \neq 0$,

$$\mathbb{E}\frac{1}{n_\ell} \operatorname{tr} \mathbf{\Delta}_\ell(\zeta) \to \frac{1}{\zeta - w_\ell^2 \sigma_\ell^2 \varphi_1(\zeta)/c}. \tag{4.105}$$

3. Combine (4.104) and (4.105) to obtain pointwise almost sure convergence then extend to the almost sure uniform convergence (4.25) and the convergence of the derivative (4.27) in Lemma 4.9.

4. Prove that φ_2 has the properties (4.26) in Lemma 4.9.

(4.11.2.1)–(4.11.2.3) follows the approach of the analogous proofs in [14, Section 2.3.2]. In (4.11.2.1) and (4.11.2.2), we let $\ell = 1$ to simplify notation; the results hold for all ℓ in the same way.

4.11.2.1 Pointwise almost sure convergence to the mean

Let $\zeta = r + \imath s \in \mathbb{C}$ with $r, s \neq 0$, and consider the expansion [14, Section 2.3.2]

$$\frac{1}{n_1} \operatorname{tr} \mathbf{\Delta}_1(\zeta) - \mathbb{E}\frac{1}{n_1} \operatorname{tr} \mathbf{\Delta}_1(\zeta) = \sum_{i=1}^{n} \underbrace{(\mathbb{E}_{i-1} - \mathbb{E}_i)\Big\{\frac{1}{n_1} \operatorname{tr} \mathbf{\Delta}_1(\zeta)\Big\}}_{=:\gamma_i}, \tag{4.106}$$

where $\mathbb{E}_i$ denotes expectation over the first i columns of $\tilde{\mathbf{E}}$. Note that

$$\begin{aligned} \operatorname{tr} \mathbf{\Delta}_1(\zeta) &= \zeta \operatorname{tr} \mathbf{\Omega}(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{\Omega}^{\mathsf{H}} \\ &= \zeta\Big[\delta_i\{(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{ii} + \operatorname{tr} \mathbf{\Omega}_{-i}\{(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{-ii}\mathbf{\Omega}_{-i}^{\mathsf{H}}\Big] \end{aligned} \tag{4.107}$$

for each $i \in \{1, \ldots, n\}$ where

- $\mathbf{\Omega} := [\mathbf{I}_{n_1 \times n_1}\ \mathbf{0}_{n_1 \times (n-n_1)}] \in \{0,1\}^{n_1 \times n}$ is used to extract the first $n_1 \times n_1$ diagonal block of $(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}$,
- δ_i is one when $i \in [1, n_1]$ and zero otherwise,
- $\{(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{ii}$ is the ith diagonal entry of $(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}$,
- $\mathbf{\Omega}_{-i} \in \{0,1\}^{n_1 \times (n-1)}$ is $\mathbf{\Omega}$ with the ith column removed, and
- $\{(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{-ii}$ is $(\zeta^2 \mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}$ with both the ith column and the ith row removed.

Taking block matrix inverses [100, Equation (0.7.3.1)] yields

$$\{(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{ii}=\frac{1}{\zeta^2-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\varepsilon}_i-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i}, \tag{4.108}$$

and with the Sherman-Morrison-Woodbury formula [100, Equation (0.7.4.1)]

$$\begin{aligned}\{(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{-ii}&=(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\\&\quad+\frac{(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}}{\zeta^2-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\varepsilon}_i-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i},\end{aligned} \tag{4.109}$$

where $\tilde{\varepsilon}_i$ is the ith column of $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{E}}_{-i}$ is $\tilde{\mathbf{E}}$ with the ith column removed. As a result,

$$\operatorname{tr}\boldsymbol{\Delta}_1(\zeta)=\zeta\Big\{\operatorname{tr}\boldsymbol{\Omega}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\boldsymbol{\Omega}_{-i}^{\mathsf{H}}+\tilde{\gamma}_i\Big\}, \tag{4.110}$$

where

$$\tilde{\gamma}_i:=\frac{\delta_i+\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\boldsymbol{\Omega}_{-i}^{\mathsf{H}}\boldsymbol{\Omega}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i}{\zeta^2-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\varepsilon}_i-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i}. \tag{4.111}$$

Since $\operatorname{tr}\boldsymbol{\Omega}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\boldsymbol{\Omega}_{-i}^{\mathsf{H}}$ does not depend on $\tilde{\varepsilon}_i$,

$$(\mathbb{E}_{i-1}-\mathbb{E}_i)\big\{\operatorname{tr}\boldsymbol{\Omega}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\boldsymbol{\Omega}_{-i}^{\mathsf{H}}\big\}=0,$$

and so

$$\gamma_i=(\mathbb{E}_{i-1}-\mathbb{E}_i)\Big\{\frac{1}{n_1}\operatorname{tr}\boldsymbol{\Delta}_1(\zeta)\Big\}=\frac{\zeta}{n_1}(\mathbb{E}_{i-1}-\mathbb{E}_i)(\tilde{\gamma}_i). \tag{4.112}$$

We now bound the magnitude of $\tilde{\gamma}_i$ by observing first that

$$\begin{aligned}&|\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\boldsymbol{\Omega}_{-i}^{\mathsf{H}}\boldsymbol{\Omega}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i|\\&\quad\le\|\{(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{\mathsf{H}}\}^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2\|(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2\\&\quad=\|(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2^2,\end{aligned} \tag{4.113}$$

where the inequality follows by Cauchy-Schwarz and $\|\boldsymbol{\Omega}_{-i}^{\mathsf{H}}\boldsymbol{\Omega}_{-i}\|=1$, and the equality holds because $\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}$ is a normal matrix even though it is *not* Hermitian. On the other hand

$$\begin{aligned}&\big|\zeta^2-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\varepsilon}_i-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\big|\\&\quad\ge\big|\Im\{\zeta^2-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\varepsilon}_i-\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\}\big|\\&\quad=\big|\Im(\zeta^2)+\Im(\zeta^2)\|(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2^2\big|\\&\quad=|\Im(\zeta^2)|\Big\{1+\|(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2^2\Big\},\end{aligned} \tag{4.114}$$

where the first equality follows by applying [14, Equation (A.1.11)] to the term $(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}$ to obtain

$$\Im\{\tilde{\varepsilon}_i^{\mathsf{H}}\tilde{\mathbf{E}}_{-i}(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\} = -\Im(\zeta^2)\|(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2^2.$$

Applying (4.113) and (4.114) to (4.111), and observing that $|\delta_i| \leq 1$, yields

$$|\tilde{\gamma}_i| \leq \frac{1 + \|(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2^2}{|\Im(\zeta^2)|\Big\{1 + \|(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\mathbf{E}}_{-i})^{-1}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}}\tilde{\varepsilon}_i\|_2^2\Big\}} = \frac{1}{|\Im(\zeta^2)|} = \frac{1}{2|rs|}. \tag{4.115}$$

As a result $\gamma_1, \ldots, \gamma_n$ are bounded and form a complex martingale difference sequence, and applying the extended Burkholder inequality [14, Lemma 2.12] for the fourth moment yields

$$\begin{aligned}\mathbb{E}\left|\frac{1}{n_1}\operatorname{tr}\boldsymbol{\Delta}_1(\zeta) - \mathbb{E}\frac{1}{n_1}\operatorname{tr}\boldsymbol{\Delta}_1(\zeta)\right|^4 &= \mathbb{E}\left|\sum_{i=1}^{n}\gamma_i\right|^4 \\ &\leq K_4\mathbb{E}\left(\sum_{i=1}^{n}|\gamma_i|^2\right)^2 = K_4\frac{|\zeta|^4}{n_1^4}\mathbb{E}\left\{\sum_{i=1}^{n}\left|(\mathbb{E}_{i-1} - \mathbb{E}_i)(\tilde{\gamma}_i)\right|^2\right\}^2 \\ &\leq K_4\frac{|\zeta|^4}{n_1^4}\mathbb{E}\left(\sum_{i=1}^{n}\frac{1}{|rs|^2}\right)^2 = K_4\frac{|\zeta|^4}{|rs|^4}\frac{n^2}{n_1^4},\end{aligned} \tag{4.116}$$

where the final inequality follows from (4.115) and the fact that

$$\left|(\mathbb{E}_{i-1} - \mathbb{E}_i)(\tilde{\gamma}_i)\right| \leq |\mathbb{E}_{i-1}(\tilde{\gamma}_i)| + |\mathbb{E}_i(\tilde{\gamma}_i)| \leq \mathbb{E}_{i-1}|\tilde{\gamma}_i| + \mathbb{E}_i|\tilde{\gamma}_i|.$$

Applying the Borel-Cantelli lemma [80, Example 14.14] and recalling that $n_1/n \to p_1$ yields (4.104).

4.11.2.2 Pointwise convergence of the mean

Let $\zeta = r + \imath s \in \mathbb{C}$ with $r, s \neq 0$, and note that

$$\mathbb{E}\frac{1}{n_1}\operatorname{tr}\boldsymbol{\Delta}_1(\zeta) = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbb{E}\{\zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{ii} = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbb{E}\left\{\frac{1}{\zeta - \tilde{\varepsilon}_i^{\mathsf{H}}(\zeta\boldsymbol{\Gamma}_i)\tilde{\varepsilon}_i}\right\}, \tag{4.117}$$

where $\boldsymbol{\Gamma}_i := (\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}})^{-1}$ and the expression for $\{\zeta(\zeta^2\mathbf{I} - \tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}_{ii}$ comes from applying the Sherman-Morrison-Woodbury formula [100, Equation (0.7.4.1)] to the denominator in (4.108). Hence

$$\mathbb{E}\frac{1}{n_1}\operatorname{tr}\boldsymbol{\Delta}_1(\zeta) - \frac{1}{\mu} = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbb{E}\left(\frac{1}{\mu - \xi_i} - \frac{1}{\mu}\right) = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbb{E}\left\{\frac{\xi_i}{\mu(\mu - \xi_i)}\right\}, \tag{4.118}$$

where

$$\mu := \zeta - w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}), \qquad \xi_i := \tilde{\varepsilon}_i^{\mathsf{H}}(\zeta\mathbf{\Gamma}_i)\tilde{\varepsilon}_i - w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}), \tag{4.119}$$

and $\mathbf{\Gamma} := (\zeta^2\mathbf{I} - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}$. Now note that

$$\frac{\xi_i}{\mu(\mu-\xi_i)} = \frac{\xi_i}{\mu(\mu-\xi_i)}\frac{(\mu-\xi_i)+\xi_i}{\mu} = \frac{\xi_i}{\mu^2} + \frac{\xi_i^2}{\mu^2(\mu-\xi_i)}, \tag{4.120}$$

and so

$$\left|\mathbb{E}\left\{\frac{\xi_i}{\mu(\mu-\xi_i)}\right\}\right| \leq \frac{|\mathbb{E}(\xi_i)|}{|\mu|^2} + \mathbb{E}\left(\frac{|\xi_i|^2}{|\mu|^2|\mu-\xi_i|}\right). \tag{4.121}$$

For any $\lambda \in \mathbb{R}$,

$$\frac{\zeta}{\zeta^2-\lambda} = \frac{\zeta((\zeta^*)^2-\lambda)}{|\zeta^2-\lambda|^2} = \frac{\zeta^*|\zeta|^2-\zeta\lambda}{|\zeta^2-\lambda|^2} = r\frac{|\zeta|^2-\lambda}{|\zeta^2-\lambda|^2} - \imath s\frac{|\zeta|^2+\lambda}{|\zeta^2-\lambda|^2}, \tag{4.122}$$

and so

$$\begin{aligned}\operatorname{sign}[\Im\{\operatorname{tr}(\zeta\mathbf{\Gamma})\}] &= \operatorname{sign}\left[\sum_{j=1}^{d}\Im\left\{\frac{\zeta}{\zeta^2-\lambda_j(\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})}\right\}\right] \\ &= \operatorname{sign}\left[\sum_{j=1}^{d}\left\{-s\underbrace{\frac{|\zeta|^2+\lambda_j(\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})}{|\zeta^2-\lambda_j(\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})|^2}}_{>0}\right\}\right] = -\operatorname{sign}(s),\end{aligned} \tag{4.123}$$

where sign denotes the sign of its argument, λ_j denotes the jth eigenvalue of its argument, and we use the fact that $\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}}$ has nonnegative eigenvalues. Hence $|\mu|$ is lower bounded as

$$|\mu| \geq \left|\Im\left\{\zeta - w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma})\right\}\right| = \left|s - w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\Im\{\operatorname{tr}(\zeta\mathbf{\Gamma})\}\right| \geq |s|. \tag{4.124}$$

Likewise, $\operatorname{sign}[\Im\{\tilde{\varepsilon}_i^{\mathsf{H}}(\zeta\mathbf{\Gamma}_i)\tilde{\varepsilon}_i\}] = -\operatorname{sign}(s)$ and $|\mu-\xi_i| \geq |s|$ As a result, (4.121) is further bounded as

$$\left|\mathbb{E}\left\{\frac{\xi_i}{\mu(\mu-\xi_i)}\right\}\right| \leq \frac{|\mathbb{E}(\xi_i)|}{|s|^2} + \frac{\mathbb{E}|\xi_i|^2}{|s|^3} = \frac{|\mathbb{E}(\xi_i)|}{|s|^2} + \frac{|\mathbb{E}(\xi_i)|^2}{|s|^3} + \frac{\mathbb{E}\left|\xi_i - \mathbb{E}(\xi_i)\right|^2}{|s|^3}, \tag{4.125}$$

so it remains to bound the mean and variance of ξ_i. Note that

$$|\mathbb{E}(\xi_i)| = \left|w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}_i) - w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma})\right| \leq \frac{w_1^2\sigma_1^2}{n}\mathbb{E}|\operatorname{tr}(\zeta\mathbf{\Gamma}_i) - \operatorname{tr}(\zeta\mathbf{\Gamma})|, \tag{4.126}$$

since $\tilde{\varepsilon}_i$ and $\mathbf{\Gamma}_i$ are independent and $\mathbb{E}(\tilde{\varepsilon}_i\tilde{\varepsilon}_i^{\mathsf{H}}) = (w_1^2\sigma_1^2/n)\mathbf{I}$ when $i \in [1, n_1]$. Next, observe that

$$|\operatorname{tr}(\zeta\mathbf{\Gamma}_i) - \operatorname{tr}(\zeta\mathbf{\Gamma})| = |\zeta|\frac{|\tilde{\varepsilon}_i^{\mathsf{H}}\mathbf{\Gamma}_i^2\tilde{\varepsilon}_i|}{|1-\tilde{\varepsilon}_i^{\mathsf{H}}\mathbf{\Gamma}_i\tilde{\varepsilon}_i|} \leq \frac{|\zeta|}{2|rs|}, \tag{4.127}$$

where the equality follows from applying the Sherman-Morrison-Woodbury formula [100, Equation (0.7.4.1)] to $\mathbf{\Gamma} = (\zeta^2\mathbf{I} - \tilde{\mathbf{E}}_{-i}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}} - \tilde{\varepsilon}_i\tilde{\varepsilon}_i^{\mathsf{H}})^{-1}$ then simplifying. The inequality follows in a similar way as in [14, Section 3.3.2, Step 1]. Substituting (4.127) into (4.126) yields the bound on the mean:

$$|\mathbb{E}(\xi_i)| \leq \frac{w_1^2\sigma_1^2}{n}\mathbb{E}\left(\frac{|\zeta|}{2|rs|}\right) = \frac{1}{n}\frac{w_1^2\sigma_1^2|\zeta|}{2|rs|}. \tag{4.128}$$

Now note that

$$\begin{aligned}\mathbb{E}\big|\xi_i - \mathbb{E}(\xi_i)\big|^2 &= \mathbb{E}\big|\tilde{\varepsilon}_i^{\mathsf{H}}(\zeta\mathbf{\Gamma}_i)\tilde{\varepsilon}_i - w_1^2\sigma_1^2\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}_i)\big|^2 \\ &= \mathbb{E}\Big|\tilde{\varepsilon}_i^{\mathsf{H}}(\zeta\mathbf{\Gamma}_i)\tilde{\varepsilon}_i - w_1^2\sigma_1^2\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}_i)\Big|^2 + w_1^4\sigma_1^4\mathbb{E}\Big|\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}_i) - \mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\mathbf{\Gamma}_i)\Big|^2,\end{aligned} \tag{4.129}$$

since $\mathbb{E}_{\tilde{\varepsilon}_i}\{\tilde{\varepsilon}_i^{\mathsf{H}}(\zeta\mathbf{\Gamma}_i)\tilde{\varepsilon}_i\} = w_1^2\sigma_1^2(1/n)\operatorname{tr}(\zeta\mathbf{\Gamma}_i)$. Defining $\mathbf{T} := \zeta\mathbf{\Gamma}_i$ and recalling that $\tilde{\varepsilon}_i = (w_1\sigma_1/\sqrt{n})\varepsilon_i$, the first term in (4.129) is

$$\begin{aligned}\frac{w_1^4\sigma_1^4}{n^2}\mathbb{E}\big|\varepsilon_i^{\mathsf{H}}\mathbf{T}\varepsilon_i - \operatorname{tr}\mathbf{T}\big|^2 &= \frac{w_1^4\sigma_1^4}{n^2}\mathbb{E}\Big|\sum_{p,q=1}^{d}\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq} - \sum_{p=1}^{d}\mathbf{T}_{pp}\Big|^2 \\ &= \frac{w_1^4\sigma_1^4}{n^2}\mathbb{E}\Big|\sum_{p\neq q}\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq} + \sum_{p=1}^{d}\mathbf{T}_{pp}(|\mathbf{E}_{pi}|^2 - 1)\Big|^2 \\ &= \frac{w_1^4\sigma_1^4}{n^2}\left(\mathbb{E}\Big|\sum_{p\neq q}\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq}\Big|^2 + \mathbb{E}\Big|\sum_{p=1}^{d}\mathbf{T}_{pp}(|\mathbf{E}_{pi}|^2 - 1)\Big|^2\right. \\ &\qquad \left. + 2\Re\mathbb{E}\left[\Big(\sum_{p\neq q}\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq}\Big)^*\Big\{\sum_{p=1}^{d}\mathbf{T}_{pp}(|\mathbf{E}_{pi}|^2 - 1)\Big\}\right]\right).\end{aligned} \tag{4.130}$$

Since the entries of $\mathbf{E}$ are independent and mean zero,

$$\mathbb{E}\left[\Big(\sum_{p\neq q}\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq}\Big)^*\Big\{\sum_{p=1}^{d}\mathbf{T}_{pp}(|\mathbf{E}_{pi}|^2 - 1)\Big\}\right] = 0, \tag{4.131}$$

so it remains to bound the other two terms in (4.130). Observe that

$$\begin{aligned}\mathbb{E}\Big|\sum_{p\neq q}\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq}\Big|^2 &= \sum_{\substack{p\neq q\\ j\neq k}}\mathbb{E}\Big(\mathbf{E}_{pi}\mathbf{E}_{qi}^*\mathbf{T}_{pq}^*\mathbf{E}_{ji}^*\mathbf{E}_{ki}\mathbf{T}_{jk}\Big) \\ &= \sum_{p\neq q}\mathbb{E}\Big(\mathbf{E}_{pi}\mathbf{E}_{qi}^*\mathbf{T}_{pq}^*\mathbf{E}_{pi}^*\mathbf{E}_{qi}\mathbf{T}_{pq}\Big) + \sum_{p\neq q}\mathbb{E}\Big(\mathbf{E}_{pi}\mathbf{E}_{qi}^*\mathbf{T}_{pq}^*\mathbf{E}_{qi}^*\mathbf{E}_{pi}\mathbf{T}_{qp}\Big) \\ &= \sum_{p\neq q}\mathbb{E}|\mathbf{E}_{pi}|^2\mathbb{E}|\mathbf{E}_{qi}|^2\mathbb{E}|\mathbf{T}_{pq}|^2 + \sum_{p\neq q}\mathbb{E}(\mathbf{E}_{pi})^2\mathbb{E}(\mathbf{E}_{qi}^*)^2\mathbb{E}(\mathbf{T}_{pq}^*\mathbf{T}_{qp}),\end{aligned} \tag{4.132}$$

where the second equality is obtained by dropping terms in the sum with expectation equal to zero, e.g., terms with $p \neq q, j, k$ for which $\mathbb{E}(\mathbf{E}_{pi}) = 0$ can be pulled out by independence. Now note that

$$\begin{aligned}(4.133)\qquad \sum_{p\neq q} \mathbb{E}(\mathbf{E}_{pi})^2 \mathbb{E}(\mathbf{E}_{qi}^*)^2 \mathbb{E}(\mathbf{T}_{pq}^* \mathbf{T}_{qp}) &\leq \sum_{p\neq q} \left| \mathbb{E}(\mathbf{E}_{pi})^2 \mathbb{E}(\mathbf{E}_{qi}^*)^2 \mathbb{E}(\mathbf{T}_{pq}^* \mathbf{T}_{qp}) \right| \\ &\leq \sum_{p\neq q} \mathbb{E}|\mathbf{E}_{pi}|^2 \mathbb{E}|\mathbf{E}_{qi}|^2 \mathbb{E}|\mathbf{T}_{pq}^* \mathbf{T}_{qp}| = \sum_{p\neq q} \mathbb{E}|\mathbf{T}_{pq}^* \mathbf{T}_{qp}| \leq \sum_{p\neq q} \mathbb{E}|\mathbf{T}_{pq}|^2,\end{aligned}$$

where the second inequality follows from Jensen's inequality, the equality holds because $\mathbb{E}|\mathbf{E}_{pi}|^2 = \mathbb{E}|\mathbf{E}_{qi}|^2 = 1$, and the final inequality follows from the arithmetic mean geometric mean inequality as

$$\begin{aligned}\sum_{p\neq q} \mathbb{E}|\mathbf{T}_{pq}^* \mathbf{T}_{qp}| = \sum_{p\neq q} \mathbb{E}\big(|\mathbf{T}_{pq}||\mathbf{T}_{qp}|\big) &= \sum_{p\neq q} \mathbb{E}\Big(\sqrt{|\mathbf{T}_{pq}|^2 |\mathbf{T}_{qp}|^2}\Big) \\ &\leq \sum_{p\neq q} \mathbb{E}\left(\frac{|\mathbf{T}_{pq}|^2 + |\mathbf{T}_{qp}|^2}{2}\right) = \sum_{p\neq q} \mathbb{E}|\mathbf{T}_{pq}|^2.\end{aligned}$$

Combining (4.132) and (4.133), and recalling that $\mathbb{E}|\mathbf{E}_{pi}|^2 = 1$, yields

$$(4.134)\qquad \mathbb{E}\left| \sum_{p\neq q} \mathbf{E}_{pi}^* \mathbf{E}_{qi} \mathbf{T}_{pq} \right|^2 \leq 2 \sum_{p\neq q} \mathbb{E}|\mathbf{T}_{pq}|^2 \leq 2 \sum_{p,q=1}^{d} \mathbb{E}|\mathbf{T}_{pq}|^2.$$

Denoting $\kappa > 1$ for an upper bound to $\mathbb{E}|\mathbf{E}_{pi}|^4 < \infty$, the second term in (4.130) is bounded as

$$\begin{aligned}(4.135)\qquad \mathbb{E}\left| \sum_{p=1}^{d} \mathbf{T}_{pp}(|\mathbf{E}_{pi}|^2 - 1) \right|^2 &= \sum_{p=1}^{d} \mathbb{E}|\mathbf{T}_{pp}|^2 (\mathbb{E}|\mathbf{E}_{pi}|^4 - 1) \\ &\leq (\kappa - 1) \sum_{p=1}^{d} \mathbb{E}|\mathbf{T}_{pp}|^2 \leq (\kappa - 1) \sum_{p,q=1}^{d} \mathbb{E}|\mathbf{T}_{pq}|^2.\end{aligned}$$

where the equality can be obtained by expanding the squared magnitude and dropping terms from the resulting double sum that are equal to zero.

Combining (4.131), (4.134), and (4.135) yields the bound for (4.130),

$$\begin{aligned}(4.136)\qquad \frac{w_1^4 \sigma_1^4}{n^2} \mathbb{E}\left| \varepsilon_i^{\mathsf{H}} \mathbf{T} \varepsilon_i - \operatorname{tr} \mathbf{T} \right|^2 &\leq \frac{w_1^4 \sigma_1^4}{n^2} \left\{ 2 \sum_{p,q=1}^{d} \mathbb{E}|\mathbf{T}_{pq}|^2 + (\kappa - 1) \sum_{p,q=1}^{d} \mathbb{E}|\mathbf{T}_{pq}|^2 \right\} \\ &= \frac{w_1^4 \sigma_1^4}{n^2} (\kappa + 1) \sum_{p,q=1}^{d} \mathbb{E}|\mathbf{T}_{pq}|^2 \leq \frac{w_1^4 \sigma_1^4}{n^2} (\kappa + 1) \frac{d|\zeta|^2}{4|rs|^2} = \frac{d}{n^2} \frac{w_1^4 \sigma_1^4 (\kappa + 1)|\zeta|^2}{4|rs|^2},\end{aligned}$$

where the final inequality holds because

$$\sum_{p,q=1}^{d}\mathbb{E}|\mathbf{T}_{pq}|^2=\mathbb{E}\,\mathrm{tr}(\mathbf{T}\mathbf{T}^{\mathsf{H}})=\mathbb{E}\Big\{\sum_{j=1}^{d}\frac{|\zeta|^2}{|\zeta^2-\lambda_j(\tilde{\mathbf{E}}_{-i}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}})|^2}\Big\}$$
$$\le\mathbb{E}\Big\{\sum_{j=1}^{d}\frac{|\zeta|^2}{|\Im\{\zeta^2-\lambda_j(\tilde{\mathbf{E}}_{-i}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}})\}|^2}\Big\}=\mathbb{E}\Big\{\sum_{j=1}^{d}\frac{|\zeta|^2}{(2|rs|)^2}\Big\}=\frac{d|\zeta|^2}{4|rs|^2},$$

where λ_j denotes the jth eigenvalue of its argument.

To bound the second term in (4.129), consider the expansion

$$\frac{1}{n}\,\mathrm{tr}(\zeta\mathbf{\Gamma}_i)-\mathbb{E}\frac{1}{n}\,\mathrm{tr}(\zeta\mathbf{\Gamma}_i)=\sum_{j=1}^{n}\underbrace{(\mathbb{E}_{j-1}-\mathbb{E}_j)\Big\{\frac{1}{n}\,\mathrm{tr}(\zeta\mathbf{\Gamma}_i)\Big\}}_{=:\nu_j},\tag{4.137}$$

where $\mathbb{E}_j$ denotes expectation over the first j columns of $\tilde{\mathbf{E}}$. Note that $\nu_i=0$ since $(1/n)\,\mathrm{tr}(\zeta\mathbf{\Gamma}_i)$ does not involve $\tilde{\varepsilon}_i$. When $j\neq i$

$$\begin{aligned}|\nu_j|&=\frac{1}{n}\Big|(\mathbb{E}_{j-1}-\mathbb{E}_j)\Big[\,\mathrm{tr}\{\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}})^{-1}\}\\&\qquad\qquad-\mathrm{tr}\{\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i,j}\tilde{\mathbf{E}}_{-i,j}^{\mathsf{H}})^{-1}\}\Big]\Big|\\&\le\frac{1}{n}(\mathbb{E}_{j-1}+\mathbb{E}_j)\Big|\,\mathrm{tr}\{\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i}\tilde{\mathbf{E}}_{-i}^{\mathsf{H}})^{-1}\}\\&\qquad\qquad-\mathrm{tr}\{\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}_{-i,j}\tilde{\mathbf{E}}_{-i,j}^{\mathsf{H}})^{-1}\}\Big|\\&\le\frac{1}{n}(\mathbb{E}_{j-1}+\mathbb{E}_j)\frac{|\zeta|}{2|rs|}=\frac{1}{n}\frac{|\zeta|}{|rs|},\end{aligned}\tag{4.138}$$

where $\tilde{\mathbf{E}}_{-i,j}$ is $\tilde{\mathbf{E}}$ with both the ith and the jth columns removed, and the final inequality follows in a similar way as (4.127). As a result $\nu_1,\ldots,\nu_n$ form a complex martingale difference sequence, and applying the extended Burkholder inequality [14, Lemma 2.12] for the second moment yields

$$\begin{aligned}\mathbb{E}\Big|\frac{1}{n}\,\mathrm{tr}(\zeta\mathbf{\Gamma}_i)-\mathbb{E}\frac{1}{n}\,\mathrm{tr}(\zeta\mathbf{\Gamma}_i)\Big|^2&=\mathbb{E}\Big|\sum_{j=1}^{n}\nu_j\Big|^2\\&\le K_2\mathbb{E}\sum_{j=1}^{n}|\nu_j|^2\le K_2\mathbb{E}\sum_{j=1}^{n}\frac{1}{n^2}\frac{|\zeta|^2}{|rs|^2}=\frac{1}{n}\frac{K_2|\zeta|^2}{|rs|^2}.\end{aligned}\tag{4.139}$$

Substituting (4.136) and (4.139) into (4.129) yields the variance bound for ξ_i:

$$\mathbb{E}\big|\xi_i-\mathbb{E}(\xi_i)\big|^2\le\frac{d}{n^2}\frac{w_1^4\sigma_1^4(\kappa+1)|\zeta|^2}{4|rs|^2}+\frac{1}{n}\frac{w_1^4\sigma_1^4K_2|\zeta|^2}{|rs|^2}.\tag{4.140}$$

Finally, combining (4.118), (4.125), (4.128), and (4.140) yields

$$
\begin{aligned}
(4.141)\qquad \left|\mathbb{E}\frac{1}{n_1}\operatorname{tr}\boldsymbol{\Delta}_1(\zeta)-\frac{1}{\mu}\right| &\le \frac{1}{n_1}\sum_{i=1}^{n_1}\left|\mathbb{E}\left\{\frac{\xi_i}{\mu(\mu-\xi_i)}\right\}\right| \\
&\le \frac{1}{n_1}\sum_{i=1}^{n_1}\left(\frac{1}{n}\frac{w_1^2\sigma_1^2|\zeta|}{2|rs^3|}+\frac{1}{n^2}\frac{w_1^4\sigma_1^4|\zeta|^2}{4|r^2s^5|}\right. \\
&\qquad\left.+\frac{d}{n^2}\frac{w_1^4\sigma_1^4(\kappa+1)|\zeta|^2}{4|r^2s^5|}+\frac{1}{n}\frac{w_1^4\sigma_1^4K_2|\zeta|^2}{|r^2s^5|}\right) \\
&= \frac{1}{n}\left(\frac{w_1^2\sigma_1^2|\zeta|}{2|rs^3|}+\frac{w_1^4\sigma_1^4K_2|\zeta|^2}{|r^2s^5|}\right)+\frac{1}{n^2}\frac{w_1^4\sigma_1^4|\zeta|^2}{4|r^2s^5|}+\frac{d}{n^2}\frac{w_1^4\sigma_1^4(\kappa+1)|\zeta|^2}{4|r^2s^5|} \\
&\to 0,
\end{aligned}
$$

since $1/n, 1/n^2, d/n^2 \to 0$ as $n, d \to \infty$ while $n/d \to c$, and (4.105) follows by observing that

$$
\mathbb{E}\frac{1}{n}\operatorname{tr}(\zeta\boldsymbol{\Gamma}) = \frac{d}{n}\mathbb{E}\frac{1}{d}\operatorname{tr}\{\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathsf{H}})^{-1}\} \to \frac{\varphi_1(\zeta)}{c},
$$

and $|\mu|, |\zeta - w_1^2\sigma_1^2\varphi_1(\zeta)/c| \ge |s| \neq 0$.

4.11.2.3 Almost sure uniform convergence

Let $\tau > 0$ be arbitrary, and consider the (countable) set

$$
(4.142)\qquad \mathbb{C}_0 := \{r + \imath s : r \in \mathbb{Q}, s \in \mathbb{Q}, r > b + \tau, s \neq 0\} \subset \{\zeta \in \mathbb{C} : \Re(\zeta) > b + \tau\},
$$

and observe that for any $\zeta \in \mathbb{C}_0$ it follows from (4.103)–(4.105) that

$$
(4.143)\qquad \frac{1}{n}\operatorname{tr}\zeta\mathbf{W}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W} \xrightarrow{\text{a.s.}} \sum_{\ell=1}^{L}\frac{p_\ell w_\ell^2}{\zeta-\tilde{\sigma}_\ell^2\varphi_1(\zeta)/c}.
$$

More precisely

$$
(4.144)\qquad \forall_{\zeta\in\mathbb{C}_0}\ \Pr\left\{\frac{1}{n}\operatorname{tr}\zeta\mathbf{W}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W} \to \sum_{\ell=1}^{L}\frac{p_\ell w_\ell^2}{\zeta-\tilde{\sigma}_\ell^2\varphi_1(\zeta)/c}\right\} = 1,
$$

but since $\mathbb{C}_0$ is countable, it follows that

$$
(4.145)\qquad \Pr\left\{\forall_{\zeta\in\mathbb{C}_0}\ \frac{1}{n}\operatorname{tr}\zeta\mathbf{W}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W} \to \sum_{\ell=1}^{L}\frac{p_\ell w_\ell^2}{\zeta-\tilde{\sigma}_\ell^2\varphi_1(\zeta)/c}\right\} = 1.
$$

Now consider $\zeta \in \mathbb{C}$ with $\Re(\zeta) > b+\tau$, and observe that eventually $\Re(\zeta)$ for all such ζ exceed all the singular values of $\tilde{\mathbf{E}}$ by at least $\tau/2$ since the largest singular value

of $\tilde{\mathbf{E}}$ converges to b. Thus, eventually

$$\begin{aligned}(4.146)\qquad \left|\frac{1}{n}\operatorname{tr}\zeta\mathbf{W}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W}\right|^2 &= \left|\frac{1}{n}\operatorname{tr}\{\mathbf{W}^2\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\}\right|^2 \\ &\le \left(\frac{1}{n}\|\mathbf{W}^2\|_F^2\right)\left\{\frac{1}{n}\|\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\|_F^2\right\} \le \frac{4}{\tau^2}\sum_{\ell=1}^{L}p_\ell w_\ell^4,\end{aligned}$$

where the first inequality follows from the Cauchy-Schwarz inequality, and the second inequality holds because $\|\mathbf{W}^2\|_F^2/n = p_1w_1^4+\cdots+p_Lw_L^4$ and

$$\begin{aligned}(4.147)\qquad &\frac{1}{n}\|\zeta(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\|_F^2 \\ &\quad= \frac{1}{n}\sum_{j=1}^{n}\left|\frac{\zeta}{\zeta^2-\nu_j^2(\tilde{\mathbf{E}})}\right|^2 = \frac{1}{n}\sum_{j=1}^{n}\left\{\frac{1}{|\zeta-\nu_j(\tilde{\mathbf{E}})|}\frac{|\zeta|}{|\zeta+\nu_j(\tilde{\mathbf{E}})|}\right\}^2 \\ &\quad\le \frac{1}{n}\sum_{j=1}^{n}\frac{1}{|\zeta-\nu_j(\tilde{\mathbf{E}})|^2} \le \frac{1}{n}\sum_{j=1}^{n}\frac{1}{|\Re(\zeta)-\nu_j(\tilde{\mathbf{E}})|^2} \le \frac{4}{\tau^2}\end{aligned}$$

where ν_j denotes the jth largest singular value of its argument, and we use the fact that $\Re(\zeta), \nu_j(\tilde{\mathbf{E}}) \ge 0$. Applying [14, Lemma 2.14] yields, almost surely, the uniform convergence (4.25) and the derivative convergence (4.27).

4.11.2.4 Properties

This section concludes the proof by verifying the following properties (4.26) of φ_2:

a) For any $\zeta > b$, almost surely eventually ζ^2 exceeds all the square singular values of $\tilde{\mathbf{E}}$, so $(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1} \succeq (1/\zeta^2)\mathbf{I}$ and

$$(4.148)\qquad \frac{1}{n}\operatorname{tr}\zeta\mathbf{W}(\zeta^2\mathbf{I}-\tilde{\mathbf{E}}^{\mathsf{H}}\tilde{\mathbf{E}})^{-1}\mathbf{W} \ge \frac{1}{\zeta}\sum_{\ell=1}^{L}\frac{n_\ell}{n}w_\ell^2 > 0.$$

Thus $\varphi_2(\zeta) > 0$ for all $\zeta > b$.

b) As $|\zeta| \to \infty$, $|\zeta - w_\ell^2\sigma_\ell^2\varphi_1(\zeta)/c| \to \infty$ for each $\ell \in \{1,\dots,L\}$ since $\varphi_1(\zeta) \to 0$ as shown in Section 4.11.1.1. Thus $\varphi_2(\zeta) \to 0$ as $|\zeta| \to \infty$.

c) As shown in Section 4.11.1.1, $\Im\{\varphi_1(\zeta)\}$ is zero if $\Im(\zeta)$ is zero and has the opposite sign of $\Im(\zeta)$ otherwise. As a result,

$$\Im\{\zeta - w_\ell^2\sigma_\ell^2\varphi_1(\zeta)/c\} = \Im(\zeta) - (w_\ell^2\sigma_\ell^2/c)\Im\{\varphi_1(\zeta)\}$$

is zero if $\Im(\zeta)$ is zero and has the same sign as $\Im(\zeta)$ otherwise. Thus we conclude that $\varphi_2(\zeta) \in \mathbb{R} \Leftrightarrow \zeta \in \mathbb{R}$.

4.11.3 Proof of Lemma 4.15

Let $Q \subset P$ be the set of points that maximize f, and note that it is nonempty by assumption. Since every level set of f is a flat, there exists some matrix A and vector b such that

$$Q = P \cap \{x \in \mathbb{R}^n : Ax = b\}, \tag{4.149}$$

so Q is also a polyhedron. Since P has at least one extreme point, Q must also have at least one extreme point.

Let x be an extreme point of Q. Next we show that x is an extreme point of P by contradiction. Suppose x is not an extreme point of P. Then there exist points $y, z \in P$, both different from x, that have convex combination equal to x. Without loss of generality, let $f(y) \leq f(z)$. Recalling that $x \in Q$ maximizes f yields

$$f(y) \leq f(z) \leq f(x). \tag{4.150}$$

By the intermediate value theorem, there exists some $\tilde{z}$ between y and x for which $f(\tilde{z}) = f(z)$. Namely, z and $\tilde{z}$ lie in the same level set, as do their affine combinations because the level sets are flats. In particular, both y and x are affine combinations of z and $\tilde{z}$, and as a result

$$f(y) = f(\tilde{z}) = f(x) = f(z), \tag{4.151}$$

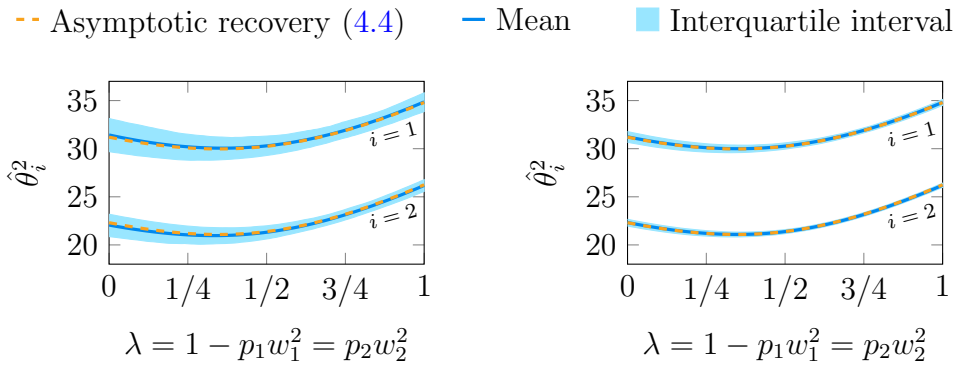
and so $y, z \in Q$, implying that x is not an extreme point of Q and producing a contradiction. Thus x is an extreme point of P that maximizes f. □

4.11.4 Additional Numerical Simulations

This section provides additional numerical simulations to demonstrate that the asymptotic results of Theorem 4.3 provide meaningful predictions for finitely many samples in finitely many dimensions. In particular, this section provides analogous plots to Fig. 4.9 in Section 4.9 for the:

- amplitudes $\hat{\theta}_i^2$ in Fig. 4.11,
- weighted score recoveries $|\langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle_{\mathbf{W}^2}|^2$ in Fig. 4.12,
- products $\langle \hat{u}_i, u_i \rangle \langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle^*_{\mathbf{W}^2}$ in Fig. 4.13.

As in Section 4.9, data are generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise



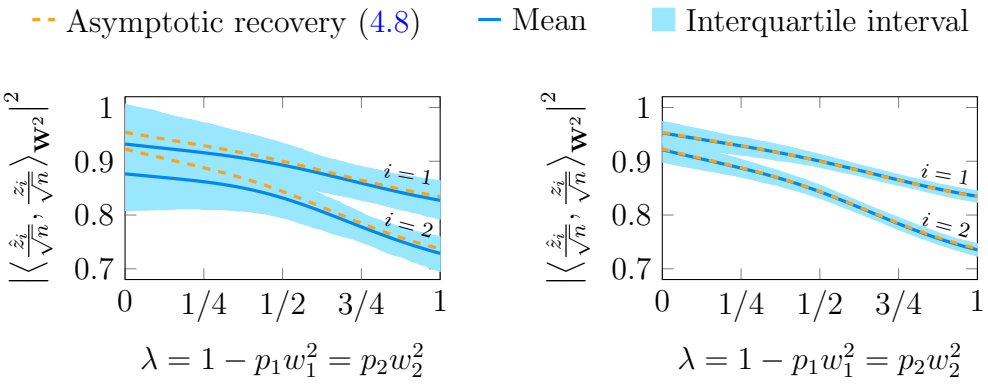
(a) 10^3 samples in 10^3 dimensions.

(b) 10^4 samples in 10^4 dimensions.

Figure 4.11: Simulated amplitudes $\hat{\theta}_i^2$ for data generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise variance $\sigma_2^2 = 4$. Weights are set as $w_1^2 = (1 - \lambda)/p_1$ and $w_2^2 = \lambda/p_2$. Simulation mean (blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic prediction (4.4) of Theorem 4.3 (orange dashed curve). Increasing the data size from (a) to (b) shrinks the interquartile intervals, indicating concentration to the mean, which is itself converging to the asymptotic recovery.

variance $\sigma_2^2 = 4$. Underlying scores and unscaled noise entries are both generated from the standard normal distribution, i.e., $z_{ij}, \varepsilon_{ij} \sim \mathcal{N}(0, 1)$, and the weights are set to $w_1^2 = (1 - \lambda)/p_1$ and $w_2^2 = \lambda/p_2$ where λ is swept from zero to one.

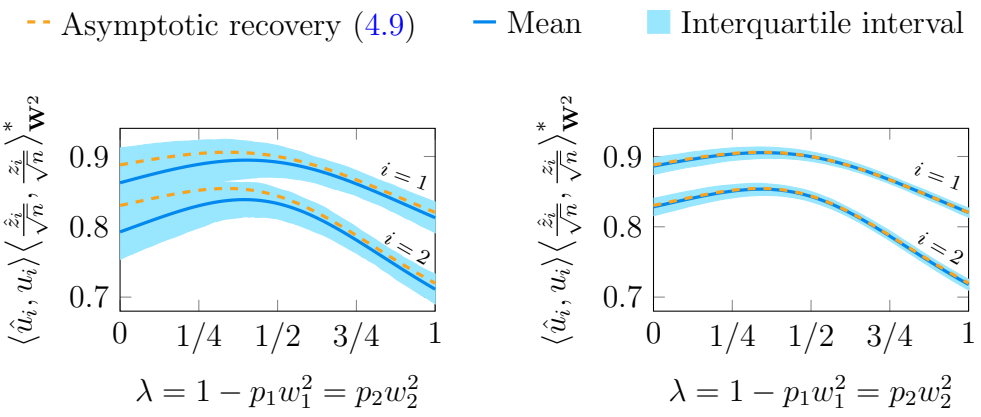
Two simulations are shown: the first has $n = 10^3$ samples in $d = 10^3$ dimensions, and the second increases these to $n = 10^4$ samples in $d = 10^4$ dimensions. Both are repeated for 500 trials. As in Fig. 4.9, the first simulation illustrates general agreement in behavior between the non-asymptotic recovery and its asymptotic prediction, and the second simulation shows what happens when the number of samples and dimensions are increased. The interquartile intervals shrink dramatically, indicating concentration of each recovery (a random quantity) around its mean, and each mean converges to the corresponding limit.



(a) 10^3 samples in 10^3 dimensions.

(b) 10^4 samples in 10^4 dimensions.

Figure 4.12: Simulated weighted score recoveries $|\langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n}\rangle_{\mathbf{W}^2}|^2$ for data generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise variance $\sigma_2^2 = 4$. Weights are set as $w_1^2 = (1 - \lambda)/p_1$ and $w_2^2 = \lambda/p_2$. Simulation mean (blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic prediction (4.8) of Theorem 4.3 (orange dashed curve). Increasing the data size from (a) to (b) shrinks the interquartile intervals, indicating concentration to the mean, which is itself converging to the asymptotic recovery.



(a) 10^3 samples in 10^3 dimensions.

(b) 10^4 samples in 10^4 dimensions.

Figure 4.13: Simulated products $\langle \hat{u}_i, u_i \rangle \langle \hat{z}_i/\sqrt{n}, z_i/\sqrt{n} \rangle^*_{\mathbf{W}^2}$ for data generated according to the model (4.3) with $c = 1$ sample per dimension, underlying amplitudes $\theta_1^2 = 25$ and $\theta_2^2 = 16$, and $p_1 = 20\%$ of samples having noise variance $\sigma_1^2 = 1$ with the remaining $p_2 = 80\%$ of samples having noise variance $\sigma_2^2 = 4$. Weights are set as $w_1^2 = (1 - \lambda)/p_1$ and $w_2^2 = \lambda/p_2$. Simulation mean (blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic prediction (4.9) of Theorem 4.3 (orange dashed curve). Increasing the data size from (a) to (b) shrinks the interquartile intervals, indicating concentration to the mean, which is itself converging to the asymptotic recovery.

CHAPTER V

Generalized canonical polyadic tensor decomposition for non-Gaussian data

Tensor decomposition is a fundamental unsupervised machine learning method in data science. It generalizes matrix decomposition methods to multiway data and has numerous applications ranging from network analysis to sensor signal processing. Standard tensor decompositions seek to minimize the squared residuals between the low-rank approximation and data. This chapter develops a generalized canonical polyadic (GCP) low-rank tensor decomposition that allows for other loss functions. For instance, the logistic loss or the generalized Kullback-Leibler divergence [122, Equation (3)] can be used to enable tensor decomposition for binary or count data. We present a variety of statistically-motivated loss functions for various scenarios. We provide a generalized framework for computing gradients and handling missing data that enables the use of standard optimization methods for fitting the model. Finally, we demonstrate the flexibility of GCP on several real-world examples including interactions in a social network, neural activity in a mouse, and monthly rainfall measurements in India.

This chapter presents joint work with Dr. Tamara G. Kolda, Dr. Cliff Anderson-Bergman, and Dr. Jed Duersch that began during a summer internship at Sandia National Labs under the mentorship of Dr. Kolda and Dr. Anderson-Bergman and that led to the recently accepted journal paper:

> [97] David Hong, Tamara G. Kolda, and Jed A. Duersch. Generalized Canonical Polyadic Tensor Decomposition. *SIAM Review*, 2019. To appear. arXiv: 1808.07452v2.

5.1 Introduction

As discussed in Section 2.3, many data sets are naturally represented as higher-order tensors. The CANDECOMP/PARAFAC or canonical polyadic (CP) tensor decomposition builds a low-rank tensor decomposition model and is a standard tool for unsupervised multiway data analysis [41, 83, 93, 118]. Structural features in the dataset are represented as rank-1 tensors, which reduces the size and complexity of the data. This form of dimensionality reduction has many applications including data decomposition into explanatory factors, filling in missing data, and data compression. It has been used to analyze multiway data sets in a variety of domains including neuroscience [2, 51, 212], chemistry [106, 144], cybersecurity [139], network analysis and link prediction [62, 116, 153], machine learning [6, 25, 173], hyperspectral imaging [68, 223], function approximation [26, 27, 81, 174], and so on. In this chapter, we consider generalizing the *loss function* for determining the low-rank model.

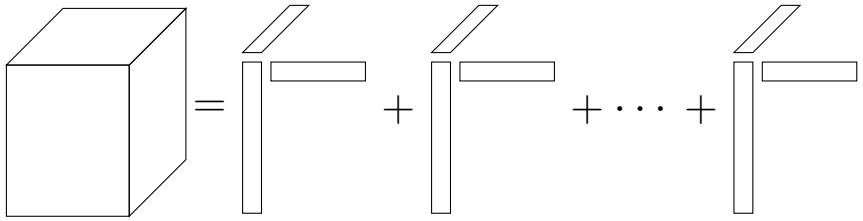


Figure 5.1: Illustration of CP-structured tensor. The tensor is the sum of r components, and each component is the outer product of d vectors, also known as a rank-1 tensor (here we show $d = 3$). The rank of such a tensor that has r components is bounded above by r, so it is low-rank if r is small.

Given a d-way data tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$, we propose a generalized CP (GCP) decomposition that approximates $\mathcal{X}$ as measured by the sum of elementwise losses specified by a generic function $f : \mathbb{R} \otimes \mathbb{R} \to \mathbb{R}$, i.e.,

$$\min F(\mathcal{M}; \mathcal{X}) := \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} f(x_i, m_i) \quad \text{subject to} \quad \mathcal{M} \text{ is low rank.} \tag{5.1}$$

Here, $\mathcal{M}$ is a low-rank model tensor that has CP structure, as illustrated in Fig. 5.1. For the usual CP decomposition, the elementwise loss is $f(x, m) = (x - m)^2$. While this loss function is suitable for many situations, it implicitly assumes the data is normally distributed. Many datasets of interest, however, do not satisfy this hidden assumption. Such data can be nonnegative, discrete, or boolean.

Our goal in this chapter is to develop a general framework for fitting GCP models with generic loss functions, enabling the user to adapt the model to the nature of the data. For example, we later see that a natural elementwise loss function for binary

tensors, which have all entries in $\{0, 1\}$, is $f(x_i, m_i) = \log(m_i + 1) - x_i \log m_i$. We show that the GCP gradient has an elegant form that uses the same computational kernels as the standard CP gradient. The formula handles the case where the sum in (5.1) is only over a subset of entries, so it also covers the case of incomplete tensors, where some data is missing due to either collection issues or an inability to make measurements. This is a common issue for real-world datasets, and it can be easily handled in the GCP framework.

5.1.1 Contributions of this chapter

We develop the GCP algorithmic framework for computing the CP tensor decomposition with an arbitrary elementwise loss function.

- The main difference between GCP and standard CP is the choice of loss function, so we discuss loss function choices and their statistical connections in Section 5.3.
- We describe fitting the GCP model in Section 5.4. We derive the gradient for GCP with respect to the model components, along with a straightforward way of handling missing data. We explain how to add regularization and use a standard optimization method.
- In Section 5.5, we demonstrate the flexibility of GCP on several real-world examples with corresponding applications including inference of missing entries, and unsupervised pattern extraction over a variety of data types.

5.1.2 Relationship to previous works

Applications of the CP tensor decomposition date back to the 1970 work of Carrol and Chang [41] and Harshman [83], though its mathematical origins date back to Hitchcock in 1927 [93]. Many surveys exist on CP and its applications; see, for instance, [6, 31, 118, 161]. Our proposed GCP framework uses so-called direct or all-at-once optimization, in contrast to the alternating approach that is popular for computing CP known as CP-ALS. The direct optimization approach has been considered for CP by Acar, Dunlavy, and Kolda [62] and Phan, Tichavský, and Cichocki [167]. The later case showed that the Hessians have special structure, and similar structure applies in the case of GCP though we do not discuss it here. The GCP framework can incorporate many of the computational improvements for CP, such as tree-based MTTKRP computations [166] and ADMM for constraints [102]. Our approach for handling missing data is essentially the same as that proposed for

standard CP by Acar, Dunlavy, Kolda, and Mørup [5]; the primary difference is that we now have a more elegant and general theory for the derivatives.

There have been a wide variety of papers that have considered alternative loss functions, so here we mention some of the most relevant. The famous nonnegative matrix factorization paper of Lee and Seung [121] considered KL divergence in the matrix case, and Welling and Weber [211] and others [44, 82, 180] considered it in the tensor case. This equates to Poisson with identity link (5.21) in our framework. Cichocki, Zdunek, Choi, Plemmons, and Amari [48] have considered loss functions based on alpha- and beta-divergences for nonnegative CP [47]; both these divergences fit into the GCP framework and we explicitly discuss beta-divergence. GCP unifies these varied loss functions into a single algorithmic framework that can fit them all.

To the best of our knowledge, no general loss function frameworks have been proposed in the tensor case, but several have been proposed in the matrix case. Collins, Dasgupta, and Schapire [50] developed a generalized version of matrix principal component analysis (PCA) based on loss functions from the exponential family (Gaussian, Poisson with exponential link, Bernoulli with logit link). Gordon [78] considers a "Generalized^2 Linear^2 Model" for matrix factorization that allows different loss functions and nonlinear relationships between the factors and the low-rank approximation. Udell et al. [202] develop a general framework for matrix factorization that allows for the loss function to be different for each column; several of their proposed loss functions overlap with ours (e.g., their "Poisson PCA" is equivalent to Poisson with the log link).

5.2 Background and notation

Before we continue, we review some basic tensor notation and concepts; see Kolda and Bader [118] for a full review. The number of ways or dimensions of the tensor is called the *order*. Each way is referred to as a *mode*. The Khatri-Rao product of two matrices $\mathbf{A} \in \mathbb{R}^{m\times p}$ and $\mathbf{B} \in \mathbb{R}^{n\times p}$ is the columnwise Kronecker product, i.e.,

$$\begin{aligned}\mathbf{A} \odot \mathbf{B} &= \Big(\mathbf{A}(:,1) \otimes \mathbf{B}(:,1), \ldots, \mathbf{A}(:,p) \otimes \mathbf{B}(:,p)\Big) \\ &= \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1p}b_{1p} \\ a_{11}b_{21} & a_{12}b_{22} & \cdots & a_{1p}b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{n1} & a_{m2}b_{n2} & \cdots & a_{mp}b_{np} \end{pmatrix} \in \mathbb{R}^{mn\times p}.\end{aligned} \tag{5.2}$$

In the remainder of this chapter, we assume all tensors are real-valued d-way arrays of size $n_1 \times n_2 \times \cdots \times n_d$. We define n and $\bar{n}$ to be the geometric and arithmetic

means of the sizes, i.e.,

$$n = \sqrt[d]{\prod_{k=1}^{d} n_k} \quad \text{and} \quad \bar{n} = \frac{1}{d}\sum_{k=1}^{d} n_k. \tag{5.3}$$

In this way, n^d is the total number of elements in the tensor and $d\bar{n}$ is the sum of the sizes of all the modes. As shown above, modes are typically indexed by $k \in \{1, \ldots, d\}$.

Tensors are indexed using i as shorthand for the *multiindex* $(i_1, i_2, \ldots, i_d)$, so that $x_i := x(i_1, i_2, \ldots, i_d)$. We let $\mathcal{I}$ denote the set of all possible indices, i.e.,

$$\mathcal{I} := \{1, \ldots, n_1\} \otimes \{1, \ldots, n_2\} \otimes \cdots \otimes \{1, \ldots, n_d\}. \tag{5.4}$$

It may be the case that some entries of $\mathcal{X}$ are *missing*, i.e., were not observed due to measurement problems. We let $\Omega \subseteq \mathcal{I}$ denote the set of *observed* entries, and then $\mathcal{I} \setminus \Omega$ is the set of missing entries.

The *vectorization* of $\mathcal{X}$ rearranges its elements into a vector of size n^d and is denoted by x. Tensor element $x(i_1, i_2, \ldots, i_d)$ is mapped to $x(i')$ in x where the *linear index* $i' \in \{1, \ldots, n^d\}$ is given by $i' = 1 + \sum_{k=1}^{d}(i_k - 1)n'_k$ with $n'_1 = 1$ and $n'_k = \prod_{\ell=1}^{k-1} n_\ell$ otherwise. The mode-k *unfolding* or *matricization* of $\mathcal{X}$ rearranges its elements into a matrix of size $n_k \times (n^d/n_k)$ and is denoted as $\mathbf{X}_k$, where the subscript indicates the mode of the unfolding. Element $(i_1, \ldots, i_d) \in \mathcal{I}$ maps to matrix entry (i_k, i'_k) where

$$i'_k = 1 + \sum_{\ell=1}^{k-1}(i_\ell - 1)n'_\ell + \sum_{\ell=k+1}^{d}(i_\ell - 1)(n'_\ell/n_k) \tag{5.5}$$

We assume the model tensor $\mathcal{M}$ in (5.1) has low-rank CP structure as illustrated in Fig. 5.1. Following Bader and Kolda [12], we refer to this type of tensor as a *Kruskal tensor*. Specifically, it is defined by a set of d *factor matrices*, $\mathbf{A}_k$ of size $n_k \times r$ for $k = 1, \ldots, d$, such that

$$m_i := m(i_1, i_2 \ldots, i_d) = \sum_{j=1}^{r} a_1(i_1, j)a_2(i_2, j)\cdots a_d(i_d, j) \qquad \text{for all } i \in \mathcal{I}. \tag{5.6}$$

The number of columns r is the same for all factor matrices and equal to the number of *components* (d-way outer products) in the model. In Fig. 5.1, the jth component is the outer product of the jth column vectors of the factor matrices, i.e., $\mathbf{A}_1(:, j)$, $\mathbf{A}_2(:, j)$, etc. We denote (5.6) in shorthand as $\mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$. The mode-$k$ unfolding of a Kruskal tensor has a special form that depends on the Khatri-Rao products of the factor matrices, i.e.,

$$\mathbf{M}_k = \mathbf{A}_k\mathbf{Z}_k^\top \quad \text{where} \quad \mathbf{Z}_k := \mathbf{A}_d \odot \cdots \odot \mathbf{A}_{k+1} \odot \mathbf{A}_{k-1} \odot \cdots \odot \mathbf{A}_1. \tag{5.7}$$

If r is relatively small (e.g., $r \leq \mathcal{O}(n)$), then we say $\mathcal{M}$ has *low rank*. The advantage of finding a low-rank structure is that it is more parsimonious. The model $\mathcal{M}$ has n^d entries but the number of values to define it is only

$$r \sum_{k=1}^{d} n_k = dr\bar{n} \ll n^d.$$

It is sometimes convenient to normalize the columns of the factor matrices and have an explicit weight for each component. For clarity of presentation, we omit this from our main discussion but do provide this alternative form and related results in Section 5.7.1.

5.3 Choice of loss function

The difference between GCP and the standard CP formulation is flexibility in the choice of loss function. This section motivates alternative loss functions by looking at the statistical likelihood of a model for a given data tensor.

In statistical modeling, we often want to maximize the *likelihood* of a model that parameterizes the distribution; see, e.g., [85, section 8.2.2]. We assume that we have a parameterized probability density function (PDF) or probability mass function (PMF) that gives the likelihood of each entry, i.e.,

$$x_i \sim p(x_i \,|\, \theta_i) \quad \text{where} \quad \ell(\theta_i) = m_i,$$

where x_i is an observation of a random variable and $\ell(\cdot)$ is an invertible *link function* that connects the model parameter m_i and the corresponding *natural parameter* of the distribution, θ_i. The link function is oftentimes just the identity function, but we show the utility of a nontrivial link function in Section 5.3.2. Link functions are a common statistical concept and have been used for generalized matrix factorizations [50, 78].

Our goal is to find the model $\mathcal{M}$ that is the *maximum likelihood estimate* (MLE) across all entries. Conditional independence of observations[1] means that the overall likelihood is just the product of the likelihoods, so the MLE is the solution to

$$\max_{\mathcal{M}} \; L(\mathcal{M}; \mathcal{X}) := \prod_{i \in \Omega} p(x_i \,|\, \theta_i) \quad \text{with} \quad \ell(\theta_i) = m_i \text{ for all } i \in \Omega. \tag{5.8}$$

We are trying to estimate the parameters θ_i, but we only have *one* observation per random variable x_i. Nevertheless, we are able to make headway because of the low-

[1] The independence is conditioned on $\mathcal{M}$. Although there are dependencies between the entries of $\mathcal{M}$ since indeed the entire purpose of the GCP decomposition is to discover these dependencies, the observations themselves remain conditionally independent.

rank structure of $\mathcal{M}$ and corresponding interdependences of the θ_i's. Recall that we have n^d observations but only $dr\bar{n}$ free variables.

For a variety of reasons, expression (5.8) is awkward for optimization. Instead we take the negative logarithm to convert the product into a sum. Since the log is monotonic, it does not change the maximizer. Negation simply converts the maximization problem into a minimization problem which is common for optimization. Eliminating θ_i as well, we arrive at the minimization problem

$$\min F(\mathcal{M}; \mathcal{X}) := \sum_{i \in \Omega} f(x_i, m_i) \quad \text{where} \quad f(x, m) := -\log p(x \,|\, \ell^{-1}(m)). \tag{5.9}$$

In the remainder of this section, we discuss how specific choices of distributions (and corresponding p's) lead naturally to specific choices for the elementwise loss function f. Each distribution has its own standard notation for the generic parameter θ, e.g., the Poisson distribution in Section 5.3.5 refers to its natural parameter as λ. Although our focus here is on statistically-motivated choices for the loss function, other options are possible as well. We mention two, the Huber loss and β-divergence, explicitly in Section 5.3.7.

5.3.1 Gaussian distribution and the standard formulation

This subsection reviews the fact that the standard squared error loss function, $f(x, m) = (x-m)^2$, comes from an assumption that the data is Gaussian distributed. A usual assumption is that the data has low-rank structure but is contaminated by "white noise," i.e.,

$$x_i = m_i + \epsilon_i \quad \text{with} \quad \epsilon_i \sim \mathcal{N}(0, \sigma) \quad \text{for all} \quad i \in \Omega. \tag{5.10}$$

Here $\mathcal{N}(\mu, \sigma)$ denotes the normal or Gaussian distribution with mean μ and standard deviation σ. We assume σ is *constant across all entries*. We can rewrite (5.10) to see that the data is Gaussian distributed:

$$x_i \sim \mathcal{N}(\mu_i, \sigma) \quad \text{with} \quad \mu_i = m_i \quad \text{for all} \quad i \in \Omega.$$

In this case, the link function between the natural parameter μ_i and the model m_i is simply the identity, i.e., $\ell(\mu) = \mu$.

From standard statistics, the PDF for the normal distribution $\mathcal{N}(\sigma, \mu)$ is

$$p(x \,|\, \mu, \sigma) = e^{-(x-\mu)^2 \,/\, 2\sigma^2} \,/\, \sqrt{2\pi\sigma^2}.$$

Following the framework in (5.9), the elementwise loss function is

$$f(x, m) = (x - m)^2 \,/\, (2\sigma^2) + \tfrac{1}{2} \log(2\pi\sigma^2).$$

Since σ is constant, it has no impact on the optimization, so we remove those terms to arrive at the standard form

$$f(x, m) = (x - m)^2 \quad \text{for} \quad x, m \in \mathbb{R}.$$

Note that this final form is no longer strictly a likelihood which has implications for, e.g., using Akaike information criterion (AIC) or the Bayesian information criterion (BIC) to choose the number of parameters. In the matrix case, the maximum likelihood derivation can be found in [218].

It is not uncommon to add a nonnegativity assumption on $\mathcal{M}$ [121, 154–156, 211], which may correspond to some prior knowledge about the means being nonnegative.

5.3.2 Bernoulli distribution and connections to logistic regression

This subsection describes a loss function for binary data. A binary random variable $x \in \{0, 1\}$ is Bernoulli distributed with parameter $\rho \in [0, 1]$ if ρ is the probability of a 1 and, consequently, $(1 - \rho)$ is the probability of a zero. We denote this by $x \sim \text{Bernoulli}(\rho)$. Clearly, the PMF is given by

$$p(x \mid \rho) = \rho^x (1 - \rho)^{(1-x)} \quad x \in \{0, 1\}. \tag{5.11}$$

A reasonable model for a binary data tensor $\mathcal{X}$ is

$$x_i \sim \text{Bernoulli}(\rho_i) \quad \text{where} \quad \ell(\rho_i) = m_i. \tag{5.12}$$

If we choose ℓ to be the identity link, then we need to constrain $m_i \in [0, 1]$ which is a complex nonlinear constraint, i.e.,

$$0 \leq \sum_{j=1}^{r} a_1(i_1, j) a_2(i_2, j) \cdots a_d(i_d, j) \leq 1 \quad \text{for all} \quad i \in \mathcal{I}, \tag{5.13}$$

Instead, we can use a different link function.

One option for the link function is to work with the *odds* ratio, i.e.,

$$\ell(\rho) = \rho \,/\, (1 - \rho). \tag{5.14}$$

It is arguably even easier to think in terms of odds ratios than the probability, so this is a natural transformation. For any $\rho \in [0, 1)$, we have $\ell(\rho) \geq 0$. Hence, using (5.14) as the link function means that we need only constrain $m_i \geq 0$. This constraint can be enforced by requiring the factor matrices to be nonnegative, which is a bound constraint and much easier to handle than the nonlinear constraint (5.13). With some algebra, it is easy to show that we can write the log of (5.11) as

$$-\log\big(p(x \mid \rho)\big) = \log\big(1 \,/\, (1 - \rho)\big) - x \log\big(\rho \,/\, (1 - \rho)\big).$$

Plugging this and the link function (5.14) into our general framework in (5.9) yields the elementwise loss function

$$f(x, m) = \log(1 + m) - x \log m \quad \text{for} \quad x \in \{\, 0, 1 \,\}, m \geq 0.$$

For a given odds $m \geq 0$, the associated probability is $\rho = m/(1 + m)$. Note that $f(1, 0) = -\infty$ because this represents a statistically impossible situation. In practice, we replace $\log m$ with $\log(m + \epsilon)$ for some small $\epsilon > 0$ to prevent numerical issues.

Another common option for the link function is to work with the log-odds, i.e.,

$$\ell(\rho) = \log\big(\, \rho \,/\, (1 - \rho) \,\big). \tag{5.15}$$

It is so common that it has a special name: *logit*. The loss function then becomes

$$f(x, m) = \log(1 + e^m) - xm \quad \text{for} \quad x \in \{\, 0, 1 \,\}, m \in \mathbb{R},$$

and the associated probability is $\rho = e^m/(1 + e^m)$. In this case, m is completely unconstrained and can be any real value. This is the transformation commonly used in logistic regression. A form of logistic tensor decomposition for a different type of decomposition called DEDICOM was proposed by Nickel and Tresp [151].

We contrast the odds and logit link functions in terms of the interpretation of the components. An advantage of odds with nonnegative factors is that each component can only *increase* the probability of a 1. The disadvantage is that it requires a nonnegativity constraint. The logit link is common in statistics and has the advantage that it does not require any constraints. A potential disadvantage is that it may be harder to interpret components since they can counteract one another. Moreover, depending on the signs of its factors, an individual component can simultaneously increase the probability of a 1 for some entries while reducing it for others. As such, interpretations may be nuanced.

5.3.3 Gamma distribution for positive continuous data

There are several distributions for handling nonnegative continuous data. As mentioned previously, one option is to assume a Gaussian distribution but impose a nonnegativity constraint. Another option is a Rayleigh distribution, discussed in the next subsection. Yet another is the gamma distribution (for strictly positive data), with PDF

$$p(x \,|\, k, \theta) = \big(x^{k-1} \,/\, (\Gamma(k)\, \theta^k)\big)\, e^{-x/\theta} \quad \text{for} \quad x > 0, \tag{5.16}$$

where $k > 0$ and $\theta > 0$ are called the shape and scale parameters respectively and $\Gamma(\cdot)$ is the Gamma function.[2] We assume k is *constant across all entries and given*, in

[2]The Gamma distribution may alternatively by parameterized by $\alpha = k$ and $\beta = 1/\theta$.

which case this is a member of the exponential family of distributions. For example, $k = 1$ and $k = 2$ are the exponential and chi-squared distributions, respectively. If we use the link function $\ell(\theta) = k\theta$ which induces a positivity constraint $m > 0$ as a byproduct,[3] and plug this and (5.16) into (5.9) and remove all constant terms (i.e., terms involving only k), the elementwise loss function is

$$f(x, m) = \log(m) + x/m \quad \text{for} \quad x > 0, m > 0. \tag{5.17}$$

In practice, we use the constraint $m \geq 0$ and replace m with $m + \epsilon$ (with small ϵ) in the loss function (5.17).

5.3.4 Rayleigh distribution for nonnegative continuous data

As alluded to in the previous subsection the *Rayleigh* distribution is a distribution for nonnegative data. The PDF is

$$p(x \,|\, \sigma) = \left(x \,/\, \sigma^2\right) e^{-x^2/(2\sigma^2)} \quad \text{for} \quad x \geq 0, \tag{5.18}$$

where $\sigma > 0$ is called the *scale* parameter. The link $\ell(\sigma) = \sqrt{\pi/2}\,\sigma$ (corresponding to the mean) induces a positivity constraint on m. Plugging this link and (5.18) into (5.9) and removing the constant terms yields the loss function

$$f(x, m) = 2\log(m) + \tfrac{\pi}{4}\,(x/m)^2 \quad \text{for} \quad x \geq 0, m > 0. \tag{5.19}$$

We again replace $m > 0$ with $m \geq 0$ and replace m with $m + \epsilon$ (with small ϵ) in the loss function (5.19).

5.3.5 Poisson distribution for count data

If the tensor values are counts, i.e., *natural* numbers ($\mathbb{N} = \{\,0, 1, 2, \ldots\,\}$), then they can be modelled as a Poisson distribution, a *discrete* probability distribution commonly used to describe the number of events that occurred in a specific window in time, e.g., emails per month. The PMF for a Poisson distribution with mean λ is given by

$$p(x \,|\, \lambda) = e^{-\lambda}\lambda^x \,/\, x! \quad \text{for} \quad x \in \mathbb{N}. \tag{5.20}$$

If we use the identity link function ($\ell(\lambda) = \lambda$) and (5.20) in (5.9) and drop constant terms, we have

$$f(x, m) = m - x\log m \quad \text{for} \quad x \in \mathbb{N}, m \geq 0. \tag{5.21}$$

[3] This also means that we set m to be the expected mean value, i.e., $m = \mathbb{E}[x] = k\theta$.

This loss function has been studied previously by Welling and Weber [211] and Chi and Kolda [44] in the case of tensor decomposition; Lee and Seung introduced it in the context of matrix factorizations [121]. As in the Bernoulli case, we have a statistical impossibility if $x > 0$ and $m = 0$, so we make the same correction of adding a small ϵ inside the log term.

Another option for the link function is the *log link*, i.e., $\ell(\lambda) = \log \lambda$. In this case, the loss function becomes

$$f(x, m) = e^m - xm \quad \text{for} \quad x \in \mathbb{N}, m \in \mathbb{R}. \tag{5.22}$$

The advantage of this approach is that m is unconstrained.

5.3.6 Negative binomial for count data

Another option for count data is the negative binomial (NegBinom) distribution. This distribution models the number of trials required before we experience $r \in \mathbb{N}$ failures, given that the probability of failure is $\rho \in [0, 1]$. The PMF is given by

$$p(x \,|\, r, \rho) = \binom{x + r - 1}{k} \rho^x (1 - \rho)^r \quad \text{for} \quad x \in \mathbb{N}. \tag{5.23}$$

If we use the odds link (5.14) with the probability of failure ρ, then the loss function for a given number of failures r is

$$f(x, m) = (r + x) \log(1 + m) - x \log m \quad \text{for} \quad x \in \mathbb{N}, m > 0.$$

We could also use a logit link (5.15). This is sometimes used as an alternative when Poisson is overdispersed.

5.3.7 Choosing the loss function

Our goal is to give users flexibility in the choice of loss function. In rare cases where the generation of the data is well understood, the loss function may be easily prescribed. In most real-world scenarios, however, some guesswork is required. The choice of fit function corresponds to an assumption on how the data is generated (e.g., according to a Bernoulli distribution) and we further assume that the parameters for the data generation form a low-rank tensor. Generally, users would experiment with several different fit functions and several choices for the model rank.

An overview of the statistically-motivated loss functions that we have discussed is presented in Table 5.1. The choices of Gaussian, Poisson with log link, Bernoulli with logit link, and Gamma with given k are part of the *exponential family* of loss functions, explored by Collins et al. [50] in the case of matrix factorization. We

note that some parameters are assumed to be constant (denoted in blue). For the normal and Gamma distributions, the constant terms (σ and k, respectively) do not even appear in the loss function. The situation is different for the negative binomial, where r does show up in the loss function. We have modified the positivity constraints ($m > 0$) to instead be nonnegativity constraints ($m \geq 0$) by adding a small $\epsilon = 10^{-10}$ in appropriate places inside the loss functions; these changes are indicated in red. This effectively converts the constraint to $m \geq \epsilon$. The modification is pragmatic since otherwise finite-precision arithmetic yields in $\pm\infty$ gradient and/or function values. In the sections that follow, nonnegativity of $\mathcal{M}$ is enforced by requiring that the factor matrices ($\{ \mathbf{A}_k \mid k = 1, \ldots, d \}$) be nonnegative.

Table 5.1: Statistically-motivated loss functions. Parameters in blue are assumed to be constant. Numerical adjustments are indicated in red.

Distribution	Link function	Loss function	Constraints
$\mathcal{N}(\mu, \sigma)$	$m = \mu$	$(x-m)^2$	$x, m \in \mathbb{R}$
Gamma(k, θ)	$m = k\theta$	$x/(m+\epsilon) + \log(m+\epsilon)$	$x > 0, m \geq 0$
Rayleigh(σ)	$m = \sqrt{\pi/2}\,\sigma$	$2\log(m+\epsilon) + (\pi/4)(x/(m+\epsilon))^2$	$x > 0, m \geq 0$
Poisson(λ)	$m = \lambda$	$m - x\log(m+\epsilon)$	$x \in \mathbb{N}, m \geq 0$
	$m = \log\lambda$	$e^m - xm$	$x \in \mathbb{N}$, $m \in \mathbb{R}$
Bernoulli(ρ)	$m = \rho\,/\,(1-\rho)$	$\log(m+1) - x\log(m+\epsilon)$	$x \in \{0, 1\}, m \geq 0$
	$m = \log(\rho\,/\,(1-\rho))$	$\log(1+e^m) - xm$	$x \in \{0, 1\}$, $m \in \mathbb{R}$
NegBinom(r, ρ)	$m = \rho\,/\,(1-\rho)$	$(r+x)\log(1+m) - x\log(m+\epsilon)$	$x \in \mathbb{N}$, $m \geq 0$

In terms of choosing the loss function from this list, the choice may be dictated by the form of the data. If the data is binary, for instance, then one of the Bernoulli choices may be preferred. Count data may indicate a Poisson or NB distribution. There are several choices for strictly positive data: Gamma, Rayleigh, and even Gaussian with nonnegativity constraints.

The list of possible loss functions and constraints in Table 5.1 is by no means comprehensive, and many other choices are possible. For instance, we might want to use the *Huber loss* [103], which is quadratic for small values of $|x - m|$ and linear for larger values. This is a robust loss function [85]. The Huber loss is

$$f(x, m; \Delta) = \begin{cases} (x-m)^2 & \text{if } |x-m| \leq \Delta, \\ 2\Delta|x-m| - \Delta^2 & \text{otherwise.} \end{cases} \tag{5.24}$$

This formulation has continuous first derivatives and so can be used in the GCP framework. Another option is to consider β-divergences, which have been popular in

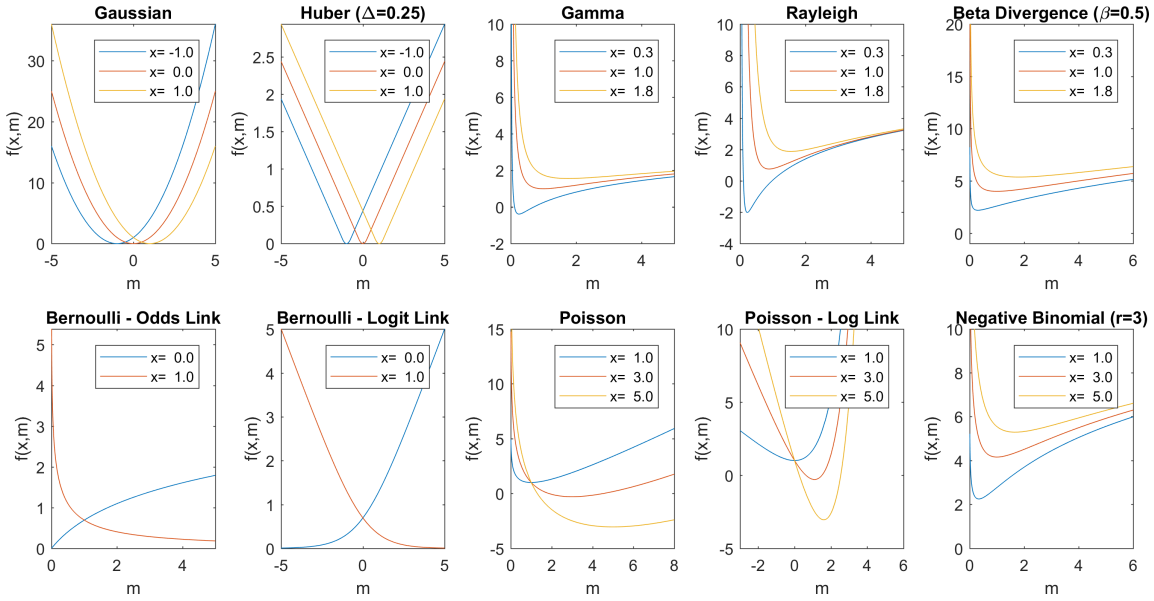


Figure 5.2: Graphical comparison of different loss functions. Note that some are only defined for binary or integer values of x (bottom row) and that some are only defined for nonnegative values of x and/or m.

matrix and tensor factorizations [46, 47, 69]. We give the formulas with the constant terms (depending only on x) omitted:

$$f(x,m;\beta) = \begin{cases} \frac{1}{\beta}m^{\beta} - \frac{1}{\beta-1}xm^{\beta-1} & \text{if } \beta \in \mathbb{R} \setminus \{0,1\}, \\ m - x\log m & \text{if } \beta = 1, \\ \frac{x}{m} + \log m & \text{if } \beta = 0. \end{cases}$$

Referring to Table 5.1, $\beta = 1$ is the same as Poisson loss with the identity link, and $\beta = 0$ is the same as the Gamma loss with the linear link.

Figure 5.2 shows a graphical summary of all the loss functions. The top row is for continuous data, and the bottom row is for discrete data. The Huber loss can be thought of as a smooth approximation of an L1 loss. Gamma, Rayleigh, and β-divergence are similar, excepting the sharpness of the dip near the minimum.

5.4 GCP decomposition

We now consider how to compute the GCP for a given elementwise loss function. The majority of this section focuses on dense tensors. Section 5.4.3 discusses both sparse tensors, i.e., tensors with many entries equal to zero, and scarce tensors, i.e., tensors with many entries missing/unknown.

Recall that we have a data tensor $\mathcal{X}$ of size $n_1 \times n_2 \times \cdots \times n_d$ and that $\Omega \subseteq \mathcal{I}$ is the set of indices where the values of $\mathcal{X}$ are known. For a given r, the objective for GCP decomposition is to find the factor matrices $\mathbf{A}_k \in \mathbb{R}^{n_k \times r}$ for $k = 1, \ldots, d$ that solve

$$\min \; F(\mathcal{M}; \mathcal{X}, \Omega) := \frac{1}{|\Omega|} \sum_{i \in \Omega} f(x_i, m_i) \quad \text{subject to} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]. \tag{5.25}$$

We sum only over the known entries, i.e., $i \in \Omega$; the same approach to missing data has been used for the CP decomposition [4, 5]. We scale by the constant $1/|\Omega|$ so that we are working with the *mean*. This is simply a convenience that makes it easier to compare function values for tensors with different sizes or different amounts of missing data. This is an optimization problem, and we propose to solve it using an off-the-shelf optimization method, which has been successful for the standard CP decomposition [3, 167] and is amenable to missing data [4, 5]. In contrast to an alternating approach, we do not have to solve a series of optimization problems. The main advantage of the alternating least squares in the solution of the standard CP decomposition is that the subproblems have closed-form solutions [118]; in contrast, the GCP subproblems do not have general closed-form solutions so we do not use an alternating method.

We focus on first-order methods, so we need to calculate the gradient of F with respect to the factor matrices. This turns out to have an elegant formulation as shown in Section 5.4.1. The GCP formulation (5.25) can also be augmented in various ways. We might add constraints on the factor matrices such as nonnegativity. Another option is to add L2-regularization on the factor matrices to handle the scale ambiguity [3], and we explain how to do this in Section 5.4.2. We might alternatively want to use L1-regularization on the factor matrices to encourage sparsity. The special structure for sparse and scarce tensors is discussed in Section 5.4.3.

5.4.1 GCP gradient

We need the gradient of F in (5.25) with respect to the factor matrices, and this is our main result in Theorem 5.3. The importance of this result is that it shows that the gradient can be calculated via a standard tensor operation called the matricized tensor times Khatri-Rao product (MTTKRP), allowing us to take advantage of existing optimized implementations for this key tensor operation. Before we get to that, we establish some useful results in the matrix case. These will be applied to mode-k unfoldings of $\mathcal{M}$ in the proof of Theorem 5.3. The next result is standard in matrix calculus and left as an exercise for the reader.

Lemma 5.1. *Let* $\mathbf{M} = \mathbf{A}\mathbf{B}^\top$ *where* $\mathbf{A}$ *is a matrix of size* $n \times r$ *and* $\mathbf{B}$ *is a matrix of size* $p \times r$*. Then*

$$\frac{\partial m_{i\ell}}{\partial a_{i'j}} = \begin{cases} b_{\ell j} & \text{if } i = i', \\ 0 & \text{if } i \neq i' \end{cases} \quad \text{for all} \quad i, i' \in \{1, \ldots, n\}, j \in \{1, \ldots, r\}, \ell \in \{1, \ldots, p\}.$$

Next, we consider the problem of generalized *matrix* factorization in Lemma 5.2, which is our linchpin result. This keeps the index notation simple but captures exactly what we need for the main result in Theorem 5.3. In Lemma 5.2, the matrix $\mathbf{W}$ is an arbitrary matrix of weights for the terms in the summation, and the matrix $\mathbf{Y}$ (which depends on $\mathbf{W}$) is a matrix of derivatives of the elementwise loss function with respect to the model.

Lemma 5.2. *Let* $\mathbf{X}, \mathbf{W}, \mathbf{A}, \mathbf{B}$ *be matrices of size* $n \times p$*,* $n \times p$*,* $n \times r$*, and* $p \times r$*, respectively. Let* $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *be a function that is continuously differentiable w.r.t. its second argument. Define the real-valued function* $\tilde{F}$ *as*

$$\tilde{F}(\mathbf{M}; \mathbf{X}, \mathbf{W}) = \sum_{i=1}^{n} \sum_{\ell=1}^{p} w_{i\ell} \, f(x_{i\ell}, m_{i\ell}) \quad \text{subject to} \quad \mathbf{M} = \mathbf{A}\mathbf{B}^\top. \tag{5.26}$$

Then the first partial derivative of $\tilde{F}$ *w.r.t.* $\mathbf{A}$ *is*

$$\frac{\partial \tilde{F}}{\partial \mathbf{A}} = \mathbf{Y}\mathbf{B} \quad \in \mathbb{R}^{n \times r}$$

where we define the $n \times p$ *matrix* $\mathbf{Y}$ *as*

$$y_{i\ell} = w_{i\ell} \, \frac{\partial f}{\partial m_{i\ell}}(x_{i\ell}, m_{i\ell}) \quad \text{for all} \quad i \in \{1, \ldots, n\}, \ell \in \{1, \ldots, p\}. \tag{5.27}$$

Proof. Consider the derivative of $\tilde{F}$ with respect to matrix element a_{ij}. We have

$$\begin{aligned} \frac{\partial \tilde{F}}{\partial a_{ij}} &= \sum_{i'=1}^{n} \sum_{\ell=1}^{p} w_{i'\ell} \, \frac{\partial f}{\partial a_{ij}}(x_{i'\ell}, m_{i'\ell}) && \text{by definition of } F \\ &= \sum_{i'=1}^{n} \sum_{\ell=1}^{p} w_{i'\ell} \, \frac{\partial f}{\partial m_{i'\ell}}(x_{i'\ell}, m_{i'\ell}) \frac{\partial m_{i'\ell}}{\partial a_{ij}} && \text{by chain rule,} \\ &= \sum_{\ell=1}^{p} y_{i\ell} b_{\ell j} && \text{by Lemma 5.1 and (5.27).} \end{aligned}$$

Rewriting this in matrix notation produces the desired result. □

Now we can consider the tensor of the GCP problem (5.25) in Theorem 5.3. For simplicity, we replace Ω with an indicator tensor $\mathcal{W}$ such that $w_i = \delta_{i \in \Omega}$ and rewrite F using $\mathcal{W}$. Although this result specifies a specific $\mathcal{W}$, it could be extended to incorporate general weights such as the relative importance of each entry; see Section 5.6 for further discussion on this topic.

Theorem 5.3 (GCP Gradients). *Let $\mathcal{X}$ be a tensor of size $n_1 \times n_2 \times \cdots \times n_d$ and Ω be the indices of known elements of $\mathcal{X}$. Let $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a function that is continuously differentiable w.r.t. its second argument. Define $\mathcal{W}$ to be an indicator tensor such that $w_i = \delta_{i\in\Omega}/|\Omega|$. Then we can rewrite the GCP problem (5.25) as*

$$\min \; F(\mathcal{M}; \mathcal{X}, \mathcal{W}) := \sum_{i\in\mathcal{I}} w_i \, f(x_i, m_i) \quad \textit{subject to} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]. \tag{5.28}$$

Here $\mathbf{A}_k$ is a matrix of size $n_k \times r$ for $k \in \{1, \ldots, d\}$. For each mode k, the first partial derivative of F w.r.t. $\mathbf{A}_k$ is given by

$$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{Y}_k \mathbf{Z}_k \tag{5.29}$$

where $\mathbf{Z}_k$ is defined in (5.7) and $\mathbf{Y}_k$ is the mode-k unfolding of a tensor $\mathcal{Y}$ defined by

$$y_i = w_i \, \frac{\partial f}{\partial m_i}(x_i, m_i) \quad \textit{for all} \quad i \in \mathcal{I}. \tag{5.30}$$

Proof. For a given k, recall that $\mathbf{M}_k = \mathbf{A}_k \mathbf{Z}_k^\top$. Hence, we can write F in (5.28) as

$$F(\mathcal{M}; \mathcal{X}, \mathcal{W}) = \tilde{F}(\mathbf{A}_k \mathbf{Z}_k^\top; \mathbf{X}_k, \mathbf{W}_k),$$

where $\tilde{F}$ is from (5.26). The result follows from Lemma 5.2 with the substitutions used in the following table:

Matrix Case	$\mathbf{X}$	$\mathbf{W}$	$\mathbf{A}$	$\mathbf{B}$	$\mathbf{Y}$	n	p	r
Tensor Case	$\mathbf{X}_k$	$\mathbf{W}_k$	$\mathbf{A}_k$	$\mathbf{Z}_k$	$\mathbf{Y}_k$	n_k	n^d/n_k	r

We note that the definition of $\mathcal{Y}$ is consistent across all k. □

Theorem 5.3 generalizes several previous results: the gradient for CP [3, 180], the gradient for CP in the case of missing data [5], and the gradient for Poisson tensor factorization [44].

Consider the gradient in (5.29). The $\mathbf{Z}_k$ has no dependence on $\mathcal{X}$, Ω, or the loss function; it depends only on the structure of the model. Conversely, $\mathcal{Y}$ has no dependence on the structure of the model. The *elementwise derivative tensor* $\mathcal{Y}$ is the same size as $\mathcal{X}$ and is zero wherever $\mathcal{X}$ is missing data. The structure of Ω determines the structural sparsity of $\mathcal{Y}$, and this will be important in Section 5.4.3. The form of the derivative is a matricized tensor times Khatri-Rao product (MTTKRP) with the tensor $\mathcal{Y}$ and the Khatri-Rao product $\mathbf{Z}_k$. The MTTKRP is the dominant kernel in the standard CP computation in terms of computation time and has optimized high-performance implementations [12, 86, 128, 186]. In the dense case, the MTTKRP costs $\mathcal{O}(rn^d)$.

Algorithm 5.1 computes the GCP loss function and gradient. On Line 2 we compute elementwise values at known data locations. If all or most elements are known, we can compute the full model using (5.7) at a cost of rn^d. However, if only a few elements are known, it may be more efficient to compute model values only at the locations in Ω using (5.6) at a cost of $2r|\Omega|$. We compute the elementwise derivative tensor $\mathcal{Y}$ in Line 4; here the quantity $\delta_{i\in\Omega}$ is 1 if $i \in \Omega$ and 0 otherwise. The cost of Lines 3 and 4 is $O(|\Omega|)$. Lines 5 to 7 compute the gradient with respect to each factor matrix, and the cost is $O(drn^d)$. Communication lower bounds as well as a parallel implementation for MTTKRP for dense tensors are covered in [16]. Since this is a *sequence* of MTTKRP operations, we can also consider reusing intermediate computations as has been done [166] and reduces the d part of the expense. Hence, the cost is dominated by the MTTKRP, just as for the standard CP-ALS. We revisit this method in the case of sparse or large-scale tensors in Section 5.4.3.

Algorithm 5.1 GCP loss function and gradient

1: **function** GCP_FG($\mathcal{X}$,Ω,$\{\mathbf{A}_k \mid k = 1,\dots,d\}$)
2: $\quad m_i \leftarrow \text{ENTRY}(\{\mathbf{A}_k \mid k = 1,\dots,d\}, i)$ for all $i \in \Omega$ ▷ Model entries
3: $\quad F \leftarrow \frac{1}{|\Omega|}\sum_{i\in\Omega} f(x_i, m_i)$ ▷ Loss function
4: $\quad y_i \leftarrow (\delta_{i\in\Omega}/|\Omega|)\frac{\partial f}{\partial m_i}(x_i, m_i)$ for all $i \in \mathcal{I}$ ▷ Elementwise derivative tensor
5: $\quad$ **for** $k = 1,\dots,d$ **do** ▷ Full sequence of MTTKRPs
6: $\quad\quad \mathbf{G}_k \leftarrow \text{MTTKRP}(\mathcal{Y}, [\![\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_d]\!], k)$ ▷ Gradients w.r.t. $\mathbf{A}_k$
7: $\quad$ **end for**
8: $\quad$ **return** F and $\{\mathbf{G}_k \mid k = 1,\dots,d\}$
9: **end function**

5.4.2 Regularization

It is straightforward to add regularization to the GCP formulation. This may especially be merited when there is a large proportion of missing data, in which case some of the factor elements may not be constrained due to lack of data. As an example, consider simple L2 regularization. We modify the GCP problem in (5.25) to be

$$\min F(\mathcal{M}; \mathcal{X}, \Omega, \{\eta_k\}) := \frac{1}{|\Omega|}\sum_{i\in\Omega} f(x_i, m_i) + \sum_{k=1}^{d}\frac{\eta_k}{2}\|\mathbf{A}_k\|_2^2 \quad \text{subject to} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_d]\!]. \tag{5.31}$$

In this case, the gradients are given by

$$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{Y}_k\mathbf{Z}_k + \eta_k\mathbf{A}_k, \tag{5.32}$$

where $\mathbf{Y}_k$ and $\mathbf{Z}_k$ are the same as in (5.30). The difficulty is in picking the regularization parameters, $\{\eta_k\}$. These can all be equal or different, and can be selected by cross-validation using prediction of held out elements.

5.4.3 GCP Decomposition for Sparse or Scarce Tensors

Sparse and scarce tensors can be efficiently stored by keeping only nonzero/known values and the corresponding indices. If s is the number of nonzero/known values, the required storage is $s(d+1)$ rather than n^d for the dense tensor where every zero or unknown value is stored explicitly.

The fact that $\mathcal{X}$ is sparse does not imply that the $\mathcal{Y}$ tensor needed to compute the gradient (see Theorem 5.3) is sparse. This is because $\frac{\partial f}{\partial m_i}(0, m_i) \neq 0$ for general values of m_i. There are two cases where the gradient has a structure that allows us to avoid explicitly calculating $\mathcal{Y}$:

- Standard Gaussian formulation; see Section 5.7.2 for details.
- Poisson formulation with the identity link; see [44] for details.

Otherwise, we have to calculate the dense $\mathcal{Y}$ explicitly to compute the gradients. For many large-scale tensors, this is infeasible. The fact that $\mathcal{X}$ is scarce, however, does imply that the tensor $\mathcal{Y}$ is sparse. This is because all missing elements in $\mathcal{X}$ correspond to zeros in $\mathcal{Y}$.

Let us take a moment to contrast the implication of sparse versus scarce. Recall that a sparse tensor is one where the vast majority of elements are zero, whereas a scarce tensor is one where the vast majority of elements are missing. The elementwise gradient tensor $\mathcal{Y}$ for a sparse tensor is structurally dense, but it is sparse for a scarce tensor. To put it another way, if $\mathcal{X}$ is sparse, then the MTTKRP calculation in Line 6 of Algorithm 5.1 has a dense $\mathcal{Y}$; but if $\mathcal{X}$ is scarce, then the MTTKRP calculation uses a sparse $\mathcal{Y}$. Further discussion of sparse versus scarce in the matrix case can be found in a blog post by Kolda [117]. We summarize the situation in Fig. 5.3.

Dense $\mathcal{X} \Rightarrow$ Dense $\mathcal{Y}$
Sparse $\mathcal{X} \Rightarrow$ Dense $\mathcal{Y}$
Scarce $\mathcal{X} \Rightarrow$ Sparse $\mathcal{Y}$

Figure 5.3: Contrasting sparsity and scarcity in GCP.

The idea that scarcity yields sparsity in the gradient calculation suggests several possible approaches for handling large-scale tensors. One possibility is to simply leave out some of the data, i.e., impose scarcity. Consider that we have a vastly

overdetermined problem because we have n^d observations but only need to determine $rd\bar{n}$ parameters. Special care needs to be taken if the tensor is sparse, since leaving out the vast majority of the nonzero entries would clearly degrade the solution. Another option is to consider stochastic gradient descent, where the batch at each iteration can be considered as a scarce tensor, leading again to a sparse $\mathcal{Y}$ in the gradient calculation. These are topics that we will investigate in detail in future work.

5.5 Experimental results

The goal of GCP is to give data analysts the flexibility to try out different loss functions. This section shows examples that illustrate the differences in the tensor factorization from using different loss functions. We do not claim that any particular loss function is better than any other; instead, we want to highlight the ability to easily use different loss functions. Along the way, we also show the general utility of tensor decomposition, which includes:

- **Data decomposition into explanatory factors**: We can directly visualize the resulting components and oftentimes use this for interpretation. This is analogous to matrix decompositions such as principal component analysis, independent component analysis, nonnegative matrix factorization, etc.
- **Compressed object representations**: Object i_k in mode k corresponds to row i_k in factor matrix $\mathbf{A}_k$, which is a length-r vector. This can be used as input to regression, clustering, visualization, machine learning, etc.

We focus primarily on these types of activities. However, we could also consider filling in missing data, data compression, etc.

All experiments are conducted in MATLAB. The method is implemented as `gcp_opt` in the Tensor Toolbox for MATLAB [11, 13]. For the optimization, we use limited-memory BFGS with bound constraints (L-BFGS-B) [36] that requires only the objective function and its gradient.[4] To initialize, we generate random factors with i.i.d. entries uniform on $(0, 1)$, then re-scale them to make the Frobenius norm of the corresponding model tensor match that of the data tensor. First-order optimization methods such as L-BFGS-B typically expect a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}$ and a corresponding vector-valued gradient, but the optimization variables in GCP are matrix-valued; see Section 5.7.3 for discussion of how we practically

[4] We specifically use the MATLAB-compatible translation by Stephen Becker, available at https://github.com/stephenbeckr/L-BFGS-B-C.

handle the required reshaping. For simplicity, we choose a rank that works reasonably well for the purposes of illustration. Generally, however, the choice of model rank is a complex procedure. It might be selected based on model consistency across multiple runs, cross-validation for estimation of hold-out data, or some prediction task using the factors. Likewise, we choose an arbitrary "run" for the purposes of illustration. These are nonconvex optimization problems, and so we are not guaranteed that every run will find the global minimum. In practice, a user would do a few runs and usually choose the one with the lowest objective value.

5.5.1 Social network

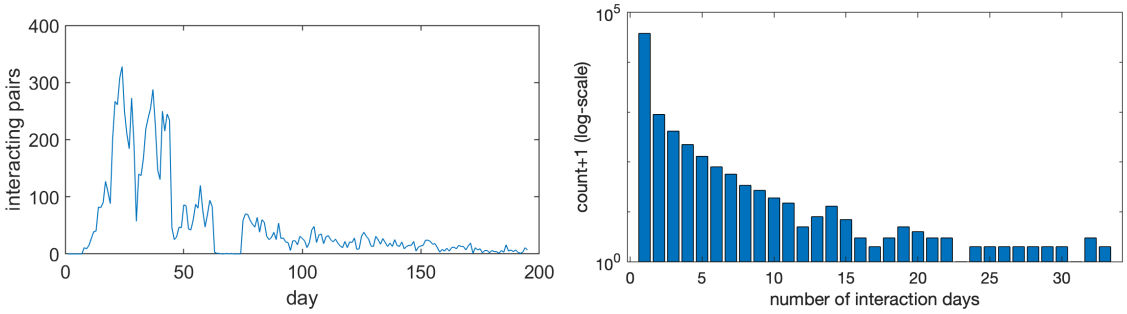
We consider the application of GCP to a social network dataset. Specifically, we use a chat network from students at UC Irvine [152, 153, 159]. It contains transmission times and sizes of 59,835 messages sent among 1899 anonymized users over 195 days from April to October 2004. Because many of the users included in the dataset sent few messages, we select only the 200 most prolific senders in this analysis. We consider a three-way binary tensor of size $200 \times 200 \times 195$ of the following form:

$$x(i_1, i_2, i_3) = \begin{cases} 1 & \text{if student } i_1 \text{ sent a message to student } i_2 \text{ on day } i_3, \\ 0 & \text{otherwise.} \end{cases}$$

It has 9764 nonzeros, so it is only 0.13% dense though we treat it as dense in this work. The number of interacting pairs per day is shown in Fig. 5.4a, and there is clearly more activity earlier in the study. To give a sense of how many days any given pair of students interact, we consider the histogram in Fig. 5.4b. The vast majority of students that interacted had only one interaction, i.e., 4×10^4 of the interactions were for only one day. The maximum number of interaction days was 33, which occurred for only one pair.

5.5.1.1 Explanatory factors for social network

We compare the explanatory GCP factors using three different loss functions in Fig. 5.5. Recall that each *component* is the outer product of three vectors; these vectors are what we plot to visualize the model. In all cases, we use $r = 7$ components because it seemed to be adequately descriptive. To visualize the factorization, components are shown as "rows", numbered on the left, and ordered by magnitude. We show all three modes as bar plots. The first two modes correspond to students, as senders and receivers. They are ordered from greatest to least total activity and



(a) Number of interacting pairs per day. Note the gap around day 70 and the decrease in activity toward the end of the experiment.

(b) Histogram of number of interactions per pair where count is in the log scale. Most students only interact once. The greatest number of interaction days is 33.

Figure 5.4: Statistics for a social network tensor where $x(i_1, i_2, i_3) = 1$ if student i_1 sends a message to student i_2 on day i_3.

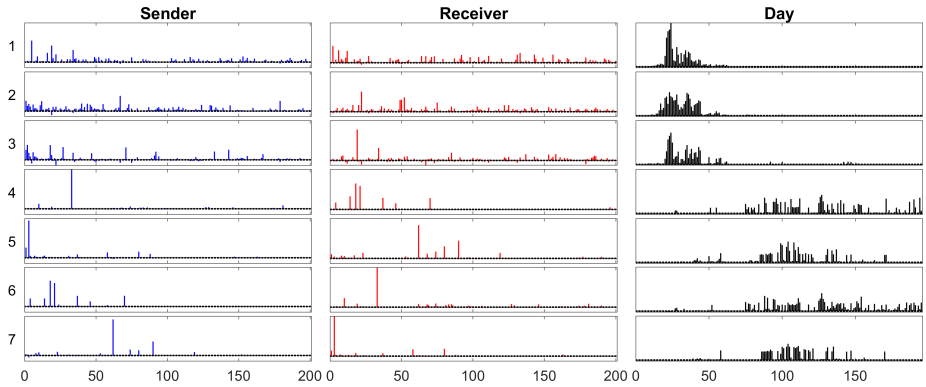
normalized to unit length. The third mode is day, and it is normalized to the magnitude of the component. Each component groups students that are messaging one another along with the dates of activity.

For the standard CP in Fig. 5.5a, we did not add a nonnegative constraint on the factors, but there are only a few small negative entries (see, e.g., the third component). There is a clear temporal locality in the first three factors. The remaining four are more diffuse. A few sender/receiver factors capture only a few large magnitude entries: sender factor 4, receiver factor 6, and both sender/receiver factors 7.

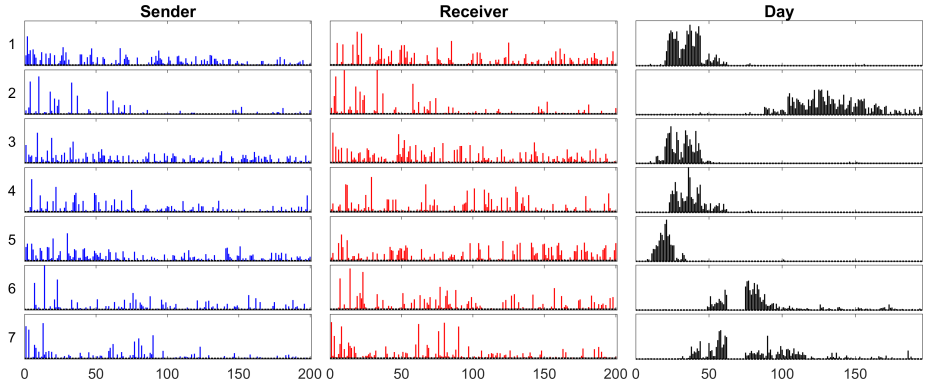
For Bernoulli with an odds link in Fig. 5.5b, the factor matrices are constrained to be nonnegative. We see even more defined temporal locality in this version. In particular, components 6 and 7 do not really have an analogue in the Gaussian version. The sender and receiver factors are correlated with one another in components 2, 6, and 7, which is something that we did not really see in the Gaussian case. Such correlations are indicative of a group talking to itself. The factors in this case seem to do a better job capturing the activity on the most active days per Fig. 5.4a.

For Bernoulli with a logit link in Fig. 5.5c, the interpretation is very different. Recall that negative values correspond to observing zeros. The first component is roughly inversely correlated with the activity per day, i.e., most entries are zeros and this is what is picked up. It is only really in components 5 and 7 where there is some push toward positive values, i.e., interactions.

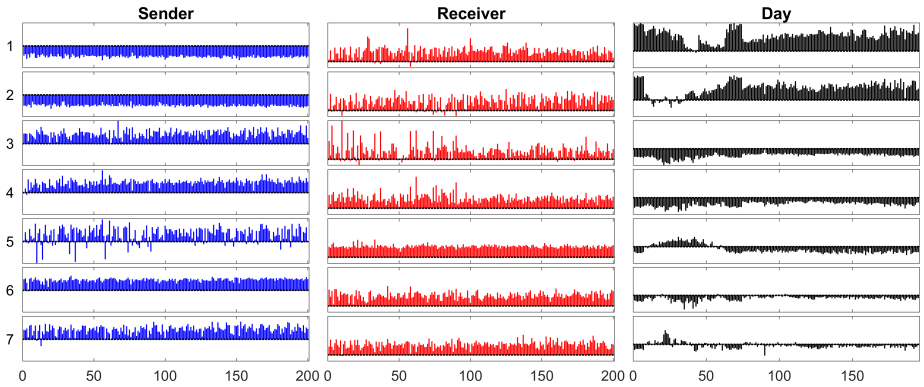
Overall, the three loss functions produce fairly different decompositions and each choice exposes a different aspect of the data. For example, standard CP tended to find factors with smaller groups of senders and receivers while Bernoulli with an odds link tended to find factors that were generally more temporally localized.



(a) Gaussian (standard CP). Some factors only pick up one or two students as senders or receivers.



(b) Bernoulli-odds (with nonnegativity constraints). Compared with CP-ALS, many students are identified with each component and more emphasis is placed on the heavier traffic days.



(c) Bernoulli-logit. A negative product means the likely result is a zero, i.e., no communication. The first few factors are focused primarily on the zeros.

Figure 5.5: GCP tensor decomposition of $200 \times 200 \times 195$ binary (0/1) social network tensor using different loss functions and $r = 7$. The three loss functions group senders and receivers in different ways, exposing different aspects of the data; selecting the most appropriate will depend on the context.

Bernoulli with a logit link seemed to most clearly identify overall activity over time. In this case it is not immediately clear which decomposition to prefer, highlighting the benefit of a generic framework like GCP that allows data analysts to try several decompositions and combine the insights obtained from them all.

5.5.1.2 Prediction for social network

To show the benefit of using a different loss function, we consider the problem of predicting missing values. We run the same experiment as before but hold out 50 ones and 50 zeros at random when fitting the model. We then use the model to predict the held out values. Let Ω denote the set of known values, so $i \notin \Omega$ means that the entry was held out. We measure the accuracy of the prediction using the log-likelihood under a Bernoulli assumption, i.e., we compute

$$\text{log-likelihood} = \sum_{\substack{x_i=1 \\ i \notin \Omega}} \log p_i + \sum_{\substack{x_i=0 \\ i \notin \Omega}} \log(1 - p_i),$$

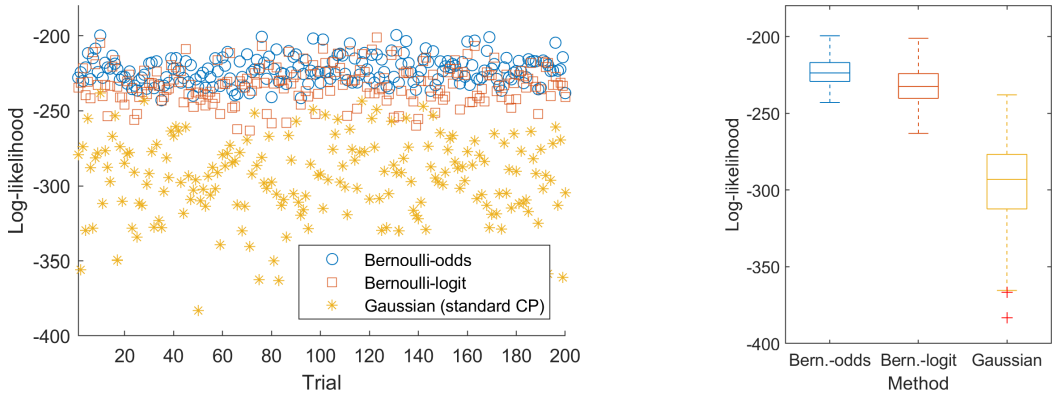
where p_i is the *probability* of a one as predicted by the model. A higher log-likelihood indicates a more accurate prediction. We convert the predicted values m_i, computed from (5.6), to probabilities p_i (truncated to the range $[10^{-16}, 1 - 10^{-16}]$) as follows:

- **Gaussian.** Let $p_i = m_i$, truncating to the range (0,1).
- **Bernoulli-odds.** Convert from the odds ratio: $p_i = m_i/(1 + m_i)$.
- **Bernoulli-logit.** Convert from the log-odds ratio: $p_i = e^{m_i}/(1 + e^{m_i})$.

We repeat the experiment two hundred times, each time holding out a different set of 100 entries. The results are shown in Fig. 5.6. This is a difficult prediction problem since ones are extremely rare; the differences in prediction performance were negligible for predicting the zeros but predicting the ones was much more difficult. Both Bernoulli-odds and Bernoulli-logit consistently outperform the standard approach based on a Gaussian loss function. We also note that the Gaussian-based predictions were outside of the range $[0, 1]$ for 11% of the predictions, making it tricky to interpret the Gaussian-based predictions.

5.5.2 Neural activity of a mouse

In recent work, Williams et al. [212] consider the application of CP tensor decomposition to analyze the neural activity of a mouse completing a series of trials. They have provided us with a reduced version of their dataset to illustrate the utility of the GCP framework. In the dataset we study, the setup is as follows. A mouse runs a



(a) Prediction of 100 missing entries for 200 trials.

(b) Box plot of prediction results.

Figure 5.6: Log-likelihood for GCP with different loss functions. Each trial holds out 50 ones and 50 zeros at random. The GCPs are computed and used to estimate each held-out value. A higher log-likelihood indicates a better prediction. In the box plot, the box represents 25th–75th percentiles with a horizontal midline at the 50th percentile, i.e., the median. The whiskers extend to the most extreme data points that are not considered outliers, and then outliers are indicated with plus-symbols.

maze over and over again, for a total of 300 trials. The maze has only one junction, at which point the mouse must turn either right or left. The mouse is forced to learn which way to turn to receive a reward. For the first 75 trials, the mouse gets a reward if it turns right; for the next 125 trials, it gets a reward if it turns left; and for the final 100 trials, it gets a reward if it turns right. Data was recorded from the prefrontal cortex of a mouse using calcium imaging; specifically, the activity of 282 neurons was recorded and processed so that all data values lie between 0 and 1. The neural activity in time for a few sample neurons is shown in Fig. 5.7; we plot each of the 300 different trials and the average value. From this image, we can see that different neurons have distinctive patterns of activity. Additionally, we see an example of at least one neuron that is clearly active for some trials and not for others (Neuron 117).

This is large and complex multiway data. We can arrange this data as a three-way nonnegative tensor as follows: 282 (neurons) $\times$ 110 (time points) $\times$ 300 trials. Applying GCP tensor decomposition reduces the data into explanatory factors, as we discuss in Section 5.5.2.1. We show how the factors can be used in a regression task in Section 5.5.2.2.

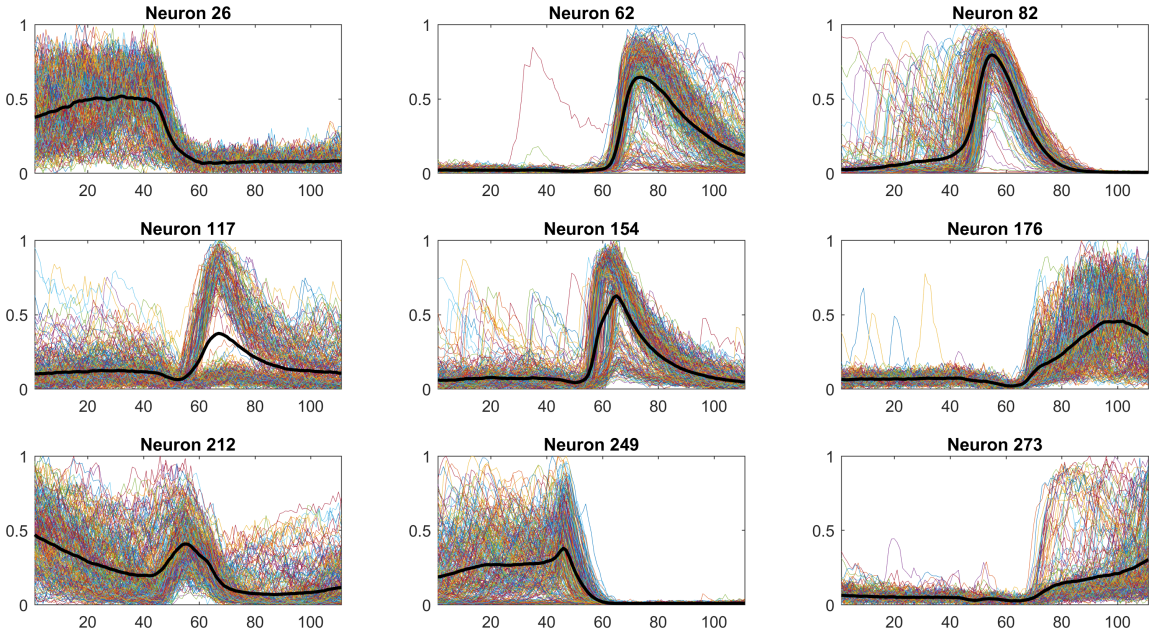


Figure 5.7: Example neuron activity across all trials. Each thin line (randomly colored) is the time profile for a single trial, and the single dark line is the average over all 300 trials. Different neurons have distinctive temporal patterns. Moreover, some have markedly different activity for different trials, like Neuron 117.

5.5.2.1 Explanatory factors for mouse neural activity

We compare the results of using different loss functions in terms of explanatory factors. In all cases, we use $r = 8$ components. The first mode corresponds to the neurons and is normalized to the size of the component, The second and third modes are, respectively, within-trial time and trial, each normalized to length 1. The neuron factors are plotted as bar graphs, showing the activation level of each neuron. The example neurons in Fig. 5.7 are highlighted as red bars; the rest are gray. The time factors are plotted as lines, and turn out to be continuous because *that is an inherent feature of the data itself*. We did nothing to enforce continuity in those factors. The trial factors are scatter plots, color coded to indicate which way the mouse turned. The dot is filled in if the mouse received a reward. When the rules changed (at trial 75 and 200, indicated by vertical dotted lines), the mouse took several trials to figure out the new way to turn for the reward.

The result of a standard CP analysis is shown in Fig. 5.8a. Several components are strongly correlated with the trial conditions, indicating the power of the CP analysis. For instance, component 3 correlates with receiving a reward (filled). Components 5, 6, and 8 correlate to turning left (orange) and right (green). Their time profiles

align with when these activities are happening (e.g., end of trial for reward and mid-trial for turn). The problem with the standard CP model is that interpretation of the negative values is difficult. Consider that neuron 212 has a significant score for nearly every component, making it hard to understand its role. Indeed, several of the example neurons have high magnitude scores for multiple components, and so it might be hard to hypothesize which neurons correspond to which trial conditions.

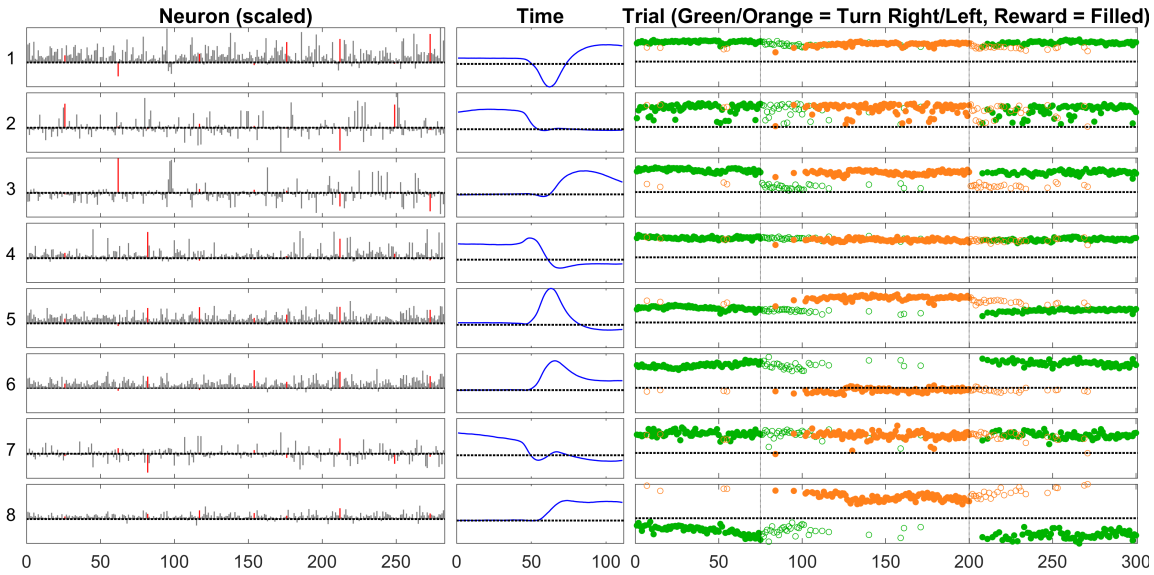
In contrast, consider Fig. 5.8b which shows the results of GCP with β-divergence with $\beta = 0.5$. The factorization is arguably easier to interpret since it has only nonnegative values. As before, we see that several components clearly correlate with the trial conditions. Components 3 and 6 correlate with reward conditions. Components 5 and 7 correlate to the turns. In this case, the example neurons seem to have clearer identities with the factors. Neuron 176 is strongest for factor 3 (reward), whereas neuron 273 is strongest for factor 6 (no reward). Some of the components do not correspond to the reward or turn, and we do not always know how to interpret them. They may have to do with external factors that are not recorded in the experimental metadata. We might also hypothesize interpretations for some components. For instance, the second component is active mid-trial and may have to do with detecting the junction in the maze. The fourth component also seems to capture similar behavior but slightly shifted in time, suggesting that aligning the temporal traces could yield an even more parsimonious decomposition.

For further comparison, we include the results of using Rayleigh, Gamma, and Huber loss functions in Fig. 5.9. These capture many of the same trends.

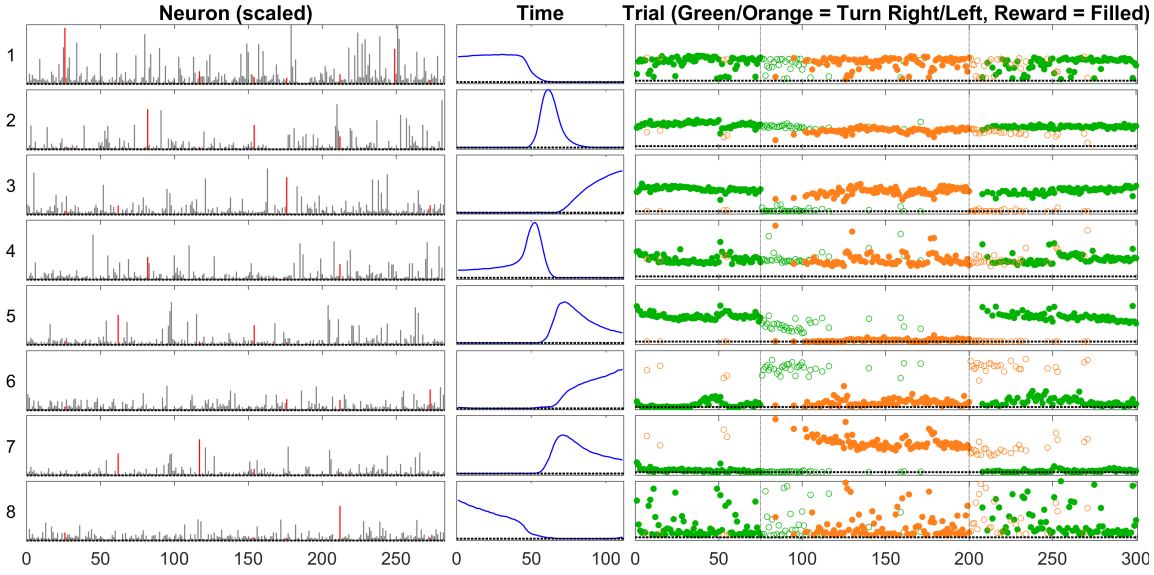
5.5.2.2 Regression task for mouse neural activity

Recall that the tensor factorization has no knowledge of the experimental conditions, i.e., which way the mouse turned or whether or not it received a reward. Suppose that the experimental logs were corrupted in such a way that we lost 50% of the trial indicators (completely at random rather than in a sequence). For instance, we might not know whether the mouse turned left or right in Trial 87. We can use the results of the GCP tensor factorization to recover that information. Observe that each trial is represented by 8 values, i.e., a score for each component. These vectors can be used for regression.

Our experimental setup is as follows. We randomly selected 50% of the 300 trials as training data and use the remainder for testing. We do simple linear regression. Specifically, we let $\mathbf{A}_3^{\text{train}}$ be the rows of $\mathbf{A}_3$ corresponding to the training trials and y^{train} be the corresponding binary responses (e.g., 1 for left turn and 0 for right turn).

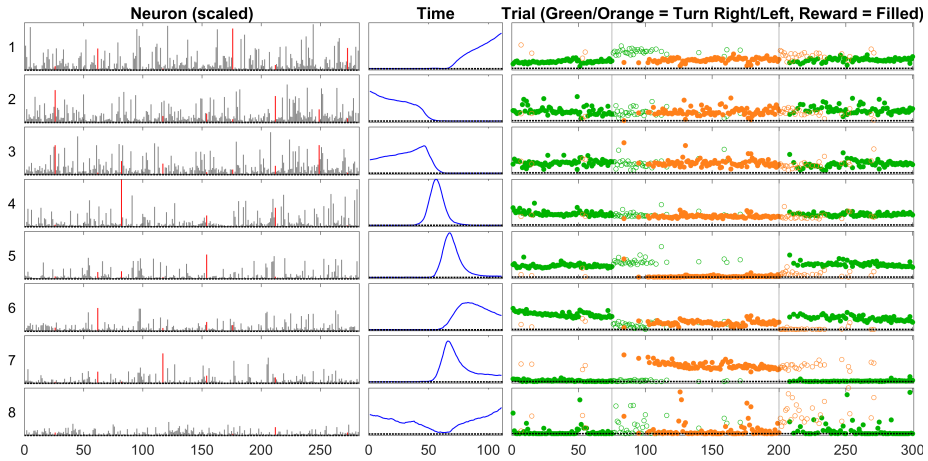


(a) Gaussian (standard CP) is difficult to interpret because of negative values

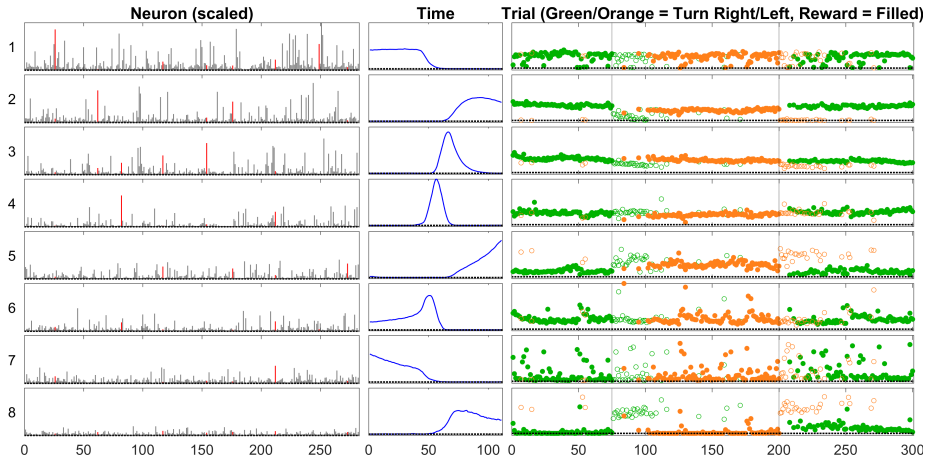


(b) Beta-divergence with β=0.5 has no negative factor values and so is easier to interpret

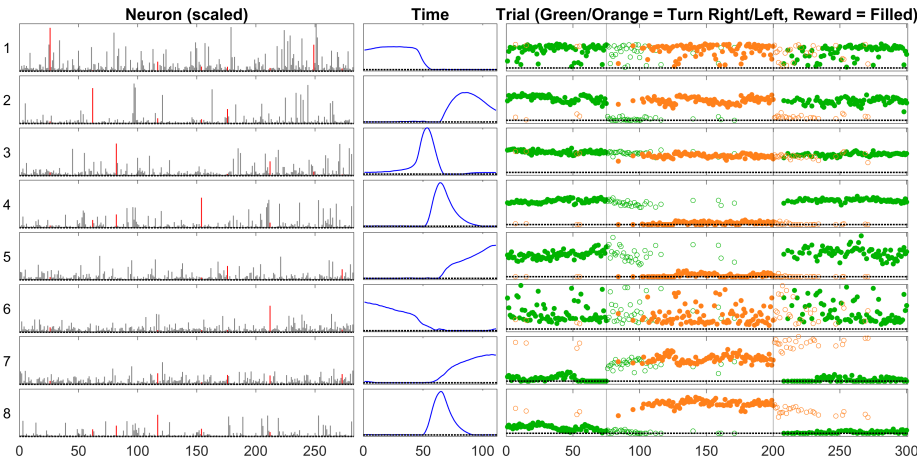
Figure 5.8: GCP tensor decomposition of mouse neural activity. Components ordered by size (top to bottom). Example neurons (26, 62, 82, 117, 154, 176, 212, 249, 273) from Fig. 5.7 are highlighted in red. Trial symbols are coded by conditions: color indicates turn and filled indicates a reward. The rule changes are denoted by vertical dotted lines. Some factors split the trials by turn (green versus orange) and others split by reward (open versus filled), even though the tensor decomposition has no knowledge of the trial conditions.



(a) Rayleigh with nonnegativity constraints



(b) Gamma with nonnegativity constraints



(c) Huber with Δ=0.25 and nonnegativity constraints

Figure 5.9: Additional GCP tensor decompositions of mouse neural activity (cf. Fig. 5.8).

Loss Type	Regression Coefficients 1	2	3	4	5	6	7	8	Max Std. Dev.	Incorrect out of 15000
Gaussian	-9.6	2.1	0.5	-0.8	3.7	15.9	3.5	1.3	2.2e+00	0
Beta Div.	5.5	5.4	-4.6	3.0	5.9	-1.8	-5.6	1.9	1.2e+00	0
Rayleigh	2.7	1.9	1.2	0.9	5.6	3.7	-5.3	-0.4	1.2e+00	0
Gamma	-15.1	22.4	6.2	4.3	-0.3	-7.6	-8.2	10.5	3.0e+00	1454
Huber	2.8	-1.3	3.4	9.7	-0.6	1.4	-1.5	-2.7	7.1e-01	0

(a) Turn

Loss Type	Regression Coefficients 1	2	3	4	5	6	7	8	Max Std. Dev.	Incorrect out of 15000
Gaussian	11.6	-0.5	18.7	-2.1	-6.9	-8.6	0.0	-3.2	3.6e+00	37
Beta Div.	5.1	-0.8	7.4	-0.1	2.8	-3.8	2.6	2.4	1.1e+00	0
Rayleigh	-6.3	8.5	8.1	1.0	-1.6	5.1	1.9	-3.0	1.3e+00	520
Gamma	10.7	1.9	0.5	0.3	-2.1	3.6	5.6	-6.4	1.3e+00	172
Huber	3.0	13.5	-9.0	2.3	2.5	2.2	-1.0	4.0	1.3e+00	62

(b) Reward

Table 5.2: Regression coefficients and prediction performance for different loss functions

We solve the regression problem:

$$\min_{\beta} \|\mathbf{A}_3^{\text{train}}\beta - y^{\text{train}}\|.$$

We let $\mathbf{A}_3^{\text{test}}$ be the rows of $\mathbf{A}_3$ corresponding to the testing trials. Using the optimal β, we make predictions for y^{test} by computing

$$\hat{y}^{\text{test}} = \left[\mathbf{A}_3^{\text{test}}\beta \geq 0.5\right].$$

We did this 100 times, both for determining the turn direction (left or right) and the reward (yes or no).

The results are shown in Table 5.2. We caution that these are merely for illustrative purposes as changing the ranks and other parameters might impact the relative performance of the methods. For the turn results, shown in Table 5.2a, only the Gamma loss failed to achieve perfect classification. We can see which factors were most important based on the regression coefficients. For instance, the sixth component is clearly the most important for Gaussian, whereas the fifth and seventh are key for β-divergence. The reward was harder to predict, per the results in Table 5.2b. This is likely due to the fact that there were relatively few times when the reward was not received. For instance, the Rayleigh method performed worst, in contrast to its perfect classification for the turn direction. Only the β-divergence achieved *perfect* regression with the third component being the most important predictor.

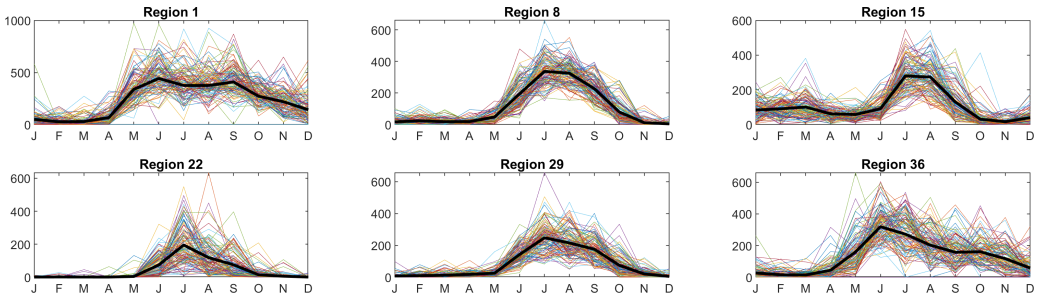


Figure 5.10: Rainfall totals per month in several regions in India. Each colored thin line represents a single year. The average is shown as a thick black line. Monsoon season is June – September.

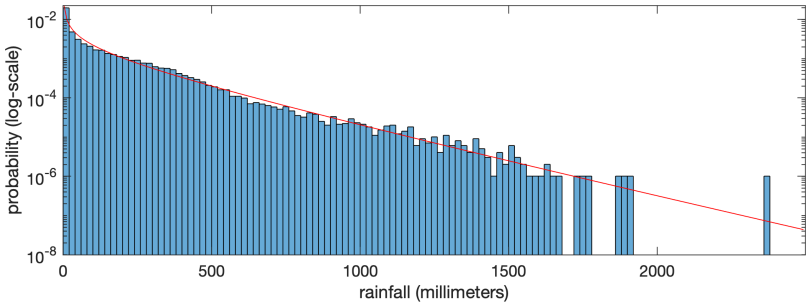


Figure 5.11: Histogram of monthly rainfall totals for 36 regions in India over 115 years. The estimated gamma distribution is shown in red.

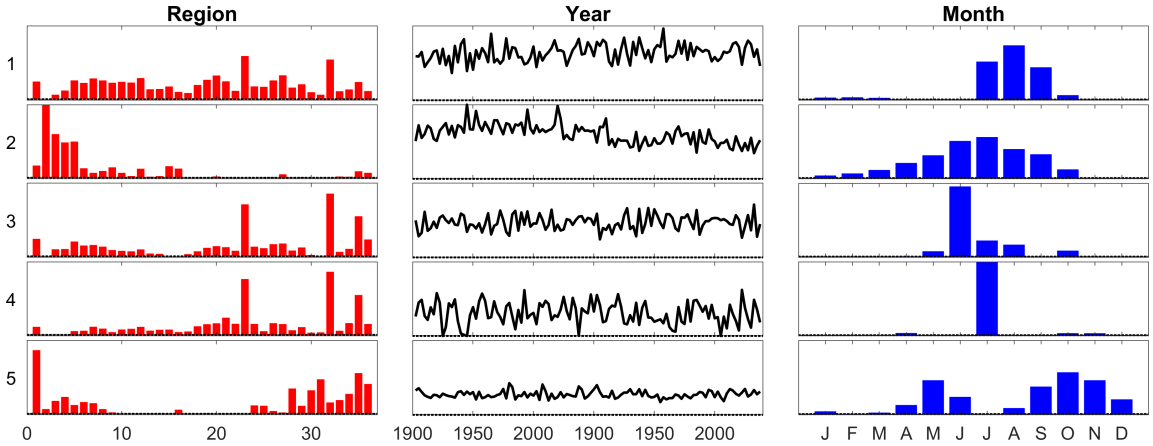
5.5.3 Rainfall in India

We consider monthly rainfall data for different regions in India for the period 1901–2015, available from Kaggle.[5] For each of 36 regions, 12 months, and 115 years, we have the total rainfall in millimeters. There is a small amount of missing data (0.72%), which GCP handles explicitly. We show example monthly rainfalls for 6 regions in Fig. 5.10.

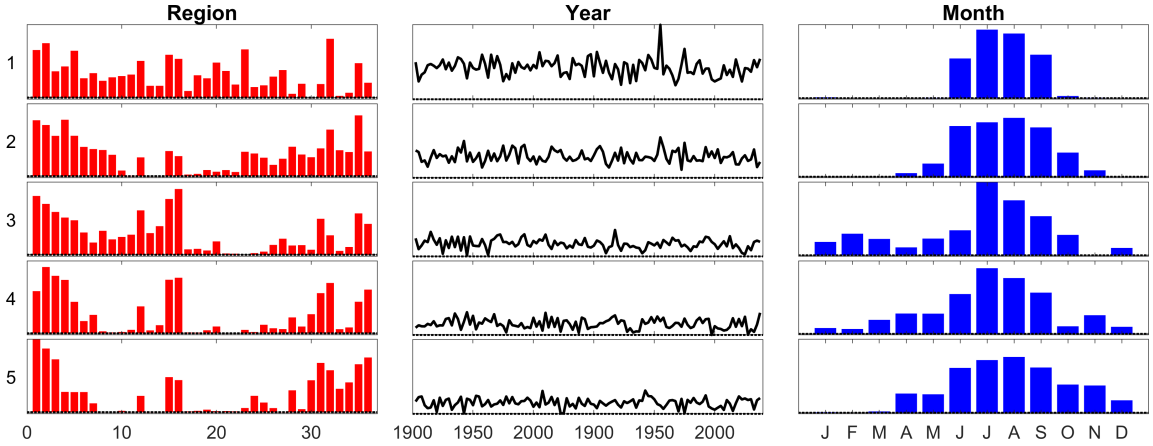
Oftentimes the gamma distribution is used to model rainfall. A histogram of all monthly values is shown in Fig. 5.11 along with the estimated gamma distribution (in red), and it seems as though a gamma distribution is potentially a reasonable model. Most rainfall totals are very small (the smallest nonzero value is 0.1mm, which is presumably the precision of the measurements), but the largest rainfall in a month exceeds 2300mm. For this reason, we consider the GCP tensor decomposition with gamma loss.

A comparison of two GCP tensor decompositions is shown in Fig. 5.12. Factors in the first two modes (region and year) are normalized to length one, and the monthly factor is normalized by the size of the component. The rainfall from year to year

[5] https://www.kaggle.com/rajanand/rainfall-in-india



(a) Gaussian (standard CP) with nonnegativity constraints, which separates July into its own component.



(b) Gamma (with nonnegativity constraints), which picks up the monsoon in the first component.

Figure 5.12: GCP tensor decomposition of India rainfall data, organized into a tensor of 36 regions, 115 years, and 12 months. The first two modes are normalized to length 1

follows no clear pattern, and this is consistent with the general understanding of these rainfall patterns. India is known for its monsoons, which occur in June–September of each year.

The GCP with standard Gaussian error loss and nonnegative constraints is shown in Fig. 5.12a. The first component captures the period July–September, which is the main part of the monsoon season. Components 3, 4, and 5 are dominated by a few regions. It is well known that Gaussian fitting can be swamped by outliers, and this may be the case here.

The GCP with the gamma distribution loss function is shown in Fig. 5.12b. This captures the monsoon season primarily in the first two components. There are no particular regions that dominate the factors.

5.6 Discussion

We have presented the GCP tensor decomposition framework which allows the use of an arbitrary differentiable elementwise loss function, generalizing previous works and enabling some extensions. GCP includes standard CP tensor decomposition and Poisson tensor decomposition [44], as well as decompositions based on beta divergences [48]. Using the GCP framework, we are able to define Bernoulli tensor decomposition for binary data, which is something like the tensor decomposition version of logistic regression and is derived via maximum likelihood. Alternatively, GCP can also handle a heuristic loss function such as Huber loss. We do not claim that any particular loss function is necessarily better than any other. Rather, for data analysis, it is often useful to have a variety of tools available, and GCP provides flexibility in terms of choosing among different loss functions to fit the needs of the analyst. Additionally, the GCP framework *efficiently* manages missing data, which is a common difficulty in practice. Our main theorem (Theorem 5.3) generalizes prior results for the gradient in the case of standard least squares, Poisson tensor factorization, and for missing data. It further reveals that the gradient takes the form of an MTTKRP, enabling the use of efficient implementations for this key tensor operation.

In our framework, we have proposed that the weights w_i be used as indicators for missingness and restricted as $w_i \in \{0, 1\}$. To generalize this, we can easily incorporate nonnegative elementwise weights $w_i \geq 0$. For instance, we might give higher or lower weights depending on the confidence in the data measurements. In recommender systems, there is also an idea that missing data may not be entirely missing at random. In this case, it may be useful to treat missing data elements as zeros but with low weights; see, e.g., [190].

For simplicity, our discussion also focused on using the same elementwise loss function $f(x_i, m_i)$ for all entries of the tensor. However, we could easily define a different loss function for every entry, i.e., $f_i(x_i, m_i)$. The only modification is to the definition (5.30) of the elementwise derivative tensor $\mathcal{Y}$. If we have a heterogeneous mixture of data types, this may be appropriate. In the matrix case, Udell, Horn, Zadeh, and Boyd [202] have proposed generalized low-rank models (GLRMs) which use a different loss function for each column in matrix factorization. We have also assumed our loss functions are continuously differentiable with respect to m_i, but that can potentially be relaxed as well in the same way as done by Udell et al. [202].

In our discussion of scarcity in Section 5.4.3, we alluded to the potential utility of imposing scarcity for scaling up to larger scale tensors. In stochastic gradient descent, for example, we impose scarcity by selecting only a few elements of the tensor at each

iteration. Another option is to purposely omit most of the data, depending on the inherent redundancies in the data (assuming it is sufficiently incoherent). These are topics that we will investigate in detail in future work.

Lastly, it may also be of interest to extend the GCP framework to functional tensor decomposition. Garcke [73], e.g., has used hinge and Huber losses for fitting a functional version of the CP tensor decomposition.

5.7 Supplementary material

5.7.1 Kruskal tensors with explicit weights

It is sometimes convenient to write (5.6) with explicit positive weights $\lambda \in \mathbb{R}^r_+$, i.e.,

$$m(i_1, i_2 \ldots, i_d) = \sum_{j=1}^{r} \lambda(j)\, a_1(i_1, j)\, a_2(i_2, j) \cdots a_d(i_d, j), \tag{5.33}$$

with shorthand $\mathcal{M} = [\![\lambda; \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$. In this case, the mode-k unfolding in (5.7) is instead given by

$$\mathbf{M}_k = \mathbf{A}_k \operatorname{diag}(\lambda) \mathbf{Z}_k^T.$$

We can also define the vectorized form

$$\mathcal{M} = [\![\lambda; \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!] \Rightarrow m = \mathbf{Z}\boldsymbol{\lambda}, \tag{5.34}$$

where

$$\mathbf{Z} := \mathbf{A}_d \odot \mathbf{A}_{d-1} \odot \cdots \odot \mathbf{A}_1 \in \mathbb{R}^{n^d \times r}. \tag{5.35}$$

Using these definitions, it is a straightforward exercise to extend Theorem 5.3 to the case $\mathcal{M} = [\![\lambda; \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$.

Corollary 5.4. *Let the conditions of Theorem 5.3 hold except that the model has an explicit weight vector so that* $\mathcal{M} = [\![\lambda; \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$. *In this case, the partial derivatives of* F *w.r.t.* $\mathbf{A}_k$ *and* $\boldsymbol{\lambda}$ *are*

$$\frac{\partial F}{\partial \mathbf{A}_k} = \mathbf{Y}_k \mathbf{Z}_k \operatorname{diag}(\boldsymbol{\lambda}) \quad \textit{and} \quad \frac{\partial F}{\partial \boldsymbol{\lambda}} = \mathbf{Z}^T y, \tag{5.36}$$

where $\mathbf{Y}_k$ *and* y *are, respectively, the mode-k unfolding and vectorization of the tensor* $\mathcal{Y}$ *defined in* (5.30), $\mathbf{Z}_k$ *is defined in* (5.7), *and* $\mathbf{Z}$ *is defined in* (5.35).

Algorithm 5.2 Wrapper for using first-order optimization method

```
function GCP_FG_WRAPPER(a)
    { A_k | k = 1,...,d } ← VEC2KT(a)
    [F, { G_k | k = 1,...,d }] ← GCP_FG(X, Ω, { A_k | k = 1,...,d })
    g ← KT2VEC({ G_k | k = 1,...,d })
    return [F, g]
end function
```

5.7.2 Special structure of standard CP gradient

In standard CP, which uses $f(x,m) = (x-m)^2$, the gradient has special structure that can be exploited when $\mathcal{X}$ is sparse. Leaving out the constant, $\frac{\partial f}{\partial m} = -x + m$; therefore, $\mathcal{Y} = -\mathcal{X} + \mathcal{M}$. From (5.29), the CP gradient is

$$\frac{\partial F}{\partial \mathbf{A}_k} = -(\mathbf{X}_k - \mathbf{M}_k)\mathbf{Z}_k = -\mathbf{X}_k\mathbf{Z}_k + \mathbf{A}_k(\mathbf{Z}_k^\top \mathbf{Z}_k). \tag{5.37}$$

The first term is an MTTKRP with the original tensor, and so it can exploit sparsity if $\mathcal{X}$ is sparse, reducing the cost from $\mathcal{O}(rn^d)$ to $\mathcal{O}(r^2 d \cdot \text{nnz}(\mathcal{X}))$ and avoiding forming $\mathbf{Z}_k$ explicitly. The second term can also avoid forming $\mathbf{Z}_k$ explicitly since its gram matrix is given by

$$\mathbf{Z}_k^\top \mathbf{Z}_k = (\mathbf{A}_1^\top \mathbf{A}_1) * \cdots * (\mathbf{A}_{k-1}^\top \mathbf{A}_{k-1}) * (\mathbf{A}_{k+1}^\top \mathbf{A}_{k+1}) * \cdots * (\mathbf{A}_d^\top \mathbf{A}_d), \tag{5.38}$$

where $*$ is the Hadamard (elementwise) product. This means that $\mathbf{Z}_k^\top \mathbf{Z}_k$ is trivial to compute, requiring only $\mathcal{O}(r^2 d\bar{n})$ operations. (5.37) is a well-known result; see, e.g., [3]. Computation of MTTKRP with a sparse tensor is discussed further in [12].

5.7.3 GCP optimization

First-order optimization methods expect a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}$ and a corresponding vector-valued gradient, but our variable is the set of d factor matrices. Because it may not be immediately obvious, we briefly explain how to make the conversion. We define the function KT2VEC to convert a Kruskal tensor, i.e., a set of factor matrices, as follows:

$$a \leftarrow \text{KT2VEC}(\{ \mathbf{A}_k \mid k = 1,\dots,d \}) := [\text{VEC}(\mathbf{A}_1); \text{VEC}(\mathbf{A}_2); \dots; \text{VEC}(\mathbf{A}_d)] .$$

The VEC operator converts a matrix to a column vector by stacking its columns, and we use MATLAB-like semicolon notation to say that the KT2VEC operator stacks all those vectors on top of each other. We can define a corresponding inverse operator, VEC2KT. The number of variables in the set of factor matrices $\{ \mathbf{A}_k \mid k = 1,\dots,d \}$ is $dr\bar{n}$, and this is exactly the same number in the vector version a because it is

just a rearrangement of the entries in the factor matrices. Since the entries of the gradient matrices correspond to the same entries in the factor matrices, we use the same transformation function for them. The wrapper that would be used to call an optimization method is shown in Algorithm 5.2. The optimization method would input a vector optimization variable, this is converted to a sequence of matrices, we compute the function and gradient using Algorithm 5.1, we turn the gradients into a vector, and we return this along with the function value.

CHAPTER VI

Ensemble K-subspaces for data from unions of subspaces

Subspace clustering is the unsupervised grouping of points lying near a union of low-dimensional linear subspaces. Algorithms based directly on geometric properties of such data tend to either provide poor empirical performance, lack theoretical guarantees, or depend heavily on their initialization. This chapter presents a novel geometric approach to the subspace clustering problem that leverages ensembles of the K-subspaces (KSS) algorithm via the evidence accumulation clustering framework. We derive general recovery guarantees for algorithms that form an affinity matrix with entries close to a monotonic transformation of pairwise absolute inner products, and show that a specific instance of our Ensemble K-subspaces (EKSS) method has this property, yielding recovery guarantees under similar conditions to state-of-the-art algorithms. The finding is, to the best of our knowledge, the first recovery guarantee for evidence accumulation clustering and for a K-subspaces based algorithm. Synthetic and real data experiments show excellent performance for a broad range of setups.

This chapter presents joint work with Dr. John Lipor, Dr. Yan Shuo Tan, and Dejiao Zhang that began when Dr. Lipor proposed we work together on the challenge of analyzing the Ensemble K-subspaces algorithm he was developing. The analysis ended up having some interesting and involved features, and our joint work led to the submitted journal paper that this chapter presents:

> [98] David Hong*, John Lipor*, Yan Shuo Tan, and Laura Balzano. Subspace Clustering using Ensembles of K-Subspaces, 2018. Submitted. (*equal contribution). arXiv: 1709.04744v2.

6.1 Introduction

In modern computer vision problems such as face recognition [20] and object tracking [198], researchers have found success applying the union of subspaces (UoS) model, in which data vectors lie near one of several low-rank subspaces. This model can be viewed as a generalization of principal component analysis (PCA) to the case of multiple subspaces, or alternatively a generalization of clustering models where the clusters have low-rank structure. The modeling goal is therefore to simultaneously identify these underlying subspaces and cluster the points according to their nearest subspace. Algorithms designed for this task are called *subspace clustering* algorithms. This topic has received a great deal of attention in recent years [207] due to various algorithms' efficacy on real-world problems such as face recognition [74], handwritten digit recognition [119], and motion segmentation [198].

One approach to subspace clustering is to leverage self-expressiveness [67, 132, 135, 216, 217], i.e., the fact that points lying on a UoS are often most efficiently represented by other points in the same subspace. Several state-of-the-art algorithms take this approach, but these methods can degrade when subspaces get very close as shown for sparse subspace clustering (SSC) in Fig. 6.2. Geometric methods [7, 30, 76, 91, 107, 162, 200, 225] take a different approach by more directly utilizing the properties of data lying on a UoS. For many geometric methods, the inner product between points is a fundamental tool used in algorithm design and theoretical analysis. In particular, the observation that the inner product between points on the same subspace is often greater than that between points on different subspaces plays a key role. This idea motivates the thresholded subspace clustering (TSC) algorithm [91], appears in the recovery guarantees of the conic subspace clustering algorithm [107], and has been shown to be an effective method of outlier rejection in both robust PCA [169] and subspace clustering [76]. However, despite directly leveraging the UoS structure in the data, geometric methods tend to either exhibit poor empirical performance, lack recovery guarantees, or depend heavily on their initialization.

In this work, we aim to overcome these issues through a set of general recovery guarantees as well as a novel geometric algorithm that achieves state-of-the-art performance across a variety of benchmark datasets. We develop recovery guarantees that match the state-of-the-art and apply to *any* algorithm that builds an affinity matrix $\mathbf{A}$ with entries close to a monotonic transformation of pairwise absolute inner products, i.e., for which

$$|\mathbf{A}_{i,j} - f\left(|\langle x_i, x_j \rangle|\right)| < \tau, \tag{6.1}$$

where f is a monotonic function, x_i, x_j are data points, and $\tau > 0$ is the maximum

deviation. Such affinity matrices arise in settings where only approximate inner products are practically available (e.g., dimensionality-reduced data), as well as in settings where deviating from pairwise inner products produces better empirical performance (e.g., by incorporating higher-order structure). We propose the Ensemble K-subspaces (EKSS) algorithm, which builds its affinity matrix by combining the outputs of many instances of the well-known K-subspaces (KSS) algorithm [7, 30] via the *evidence accumulation* clustering framework [72]. We show that the affinity matrix obtained from the first iteration of KSS fits the observation model (6.1) and consequently enjoys strong theoretical guarantees. To the best of our knowledge, these results are the first theoretical guarantees characterizing an affinity matrix resulting from evidence accumulation, as well as the first recovery guarantees for any variant of the KSS algorithm. Finally, we demonstrate that EKSS achieves excellent empirical performance on several canonical benchmark datasets.

The remainder of this chapter is organized as follows. In Section 6.2 we define the subspace clustering problem in detail and give an overview of the related work. In Section 6.3 we propose the Ensemble K-subspaces algorithm. Section 6.4 contains the theoretical contributions of this chapter. We demonstrate the strong empirical performance of EKSS on a variety of datasets in Section 6.5. Conclusions and future work are described in Section 6.6.

6.2 Problem Formulation & Related Work

Consider a collection of points $\mathcal{X} = \{ x_1, \ldots, x_N \}$ in $\mathbb{R}^D$ lying near a union of K subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_K$ having dimensions $d_1, \ldots, d_K$. Let $\mathbf{X} \in \mathbb{R}^{D \times N}$ denote the matrix whose columns are the points in $\mathcal{X}$. The goal of subspace clustering is to label points in the unknown union of K subspaces according to their nearest subspace. Once the clusters have been obtained, the corresponding subspace bases can be recovered using principal components analysis (PCA).

6.2.1 Self-expressive approaches

Most state-of-the-art approaches to subspace clustering rely on a *self-expressive* property of UoS data: informally stated, each point in a UoS is often most efficiently represented by other points in the same subspace. These methods typically use a self-expressive data cost function that is regularized to encourage efficient representation as follows:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \lambda \|\mathbf{Z}\| \\ \text{subject to} \quad & \operatorname{diag}(\mathbf{Z}) = 0, \end{aligned} \tag{6.2}$$

where λ balances the regression and penalization terms and $\|\mathbf{Z}\|$ may be the 1-norm as in sparse subspace clustering (SSC) [67], nuclear norm as in low-rank representation (which omits the constraint on $\mathbf{Z}$) [132], or a combination of these and other norms. The solution to (6.2) is used to form the affinity/similarity matrix $|\mathbf{Z}| + |\mathbf{Z}|^T$, and spectral clustering on this matrix concludes these methods. Other terms can be added to (6.2) for robustness to noise and outliers, and numerous recent papers follow this framework [135, 181, 187, 206]. Solving (6.2) can be prohibitive for large datasets; algorithms such as [216, 217] employ orthogonal matching pursuit and elastic-net formulations to reduce computational complexity and improve connectivity.

Self-expressive approaches typically have theoretical results guaranteeing no false connections (NFC), i.e., that points lying in different subspaces have zero affinity. These guarantees depend on a notion of distance between subspaces called the *subspace affinity* (6.9). Roughly stated, the closer any pair of underlying subspaces is, the more difficult the subspace clustering problem becomes. An excellent overview of these results is given in [210].

6.2.2 Geometric approaches

One class of geometric approaches, broadly speaking, applies spectral clustering on an affinity matrix built by finding a set of q "nearest neighbors" for each point. An early example of this type of algorithm is the Spectral Local Best-Fit Flats (SLBF) algorithm [226], in which neighbors are selected in terms of Euclidean distance, with the optimal number of neighbors estimated via the introduced local best-fit heuristic. While this heuristic is theoretically motivated, no clustering guarantees accompany this approach, and its performance on benchmark datasets lags significantly behind that of self-expressive methods. The greedy subspace clustering (GSC) algorithm [162] greedily builds subspaces by adding points with largest projection to form an affinity matrix, with the number of neighbors fixed. This algorithm has strong theoretical guarantees, and while its performance is still competitive, it lags behind that of self-expressive methods. Thresholded subspace clustering (TSC) [91] chooses neighbors based on the largest absolute inner product, and this simple approach obtains correct clustering under assumptions similar to those considered in the analysis of SSC. However, empirical results show that TSC performs poorly on a number of benchmark datasets. Our proposed algorithm possesses the same theoretical guarantees of TSC while also achieving excellent empirical performance.

In contrast to the above methods, the K-subspaces (KSS) algorithm [7, 30] seeks

to minimize the sum of residuals of points to their assigned subspace, i.e.,

$$\min_{\mathcal{C},\mathcal{U}} \sum_{k=1}^{K} \sum_{i:x_i \in c_k} \left\| x_i - \mathbf{U}_k \mathbf{U}_k^T x_i \right\|_2^2, \tag{6.3}$$

where $\mathcal{C} = \{ c_1, \ldots, c_K \}$ denotes the set of estimated clusters and $\mathcal{U} = \{ \mathbf{U}_1, \ldots, \mathbf{U}_K \}$ denotes the corresponding set of orthonormal subspace bases. We claim that this is a "natural" choice of objective function for the subspace clustering problem since its value is zero if a perfect UoS fit is obtained. Further, in the case of noiseless data, the optimal solution to (6.3) does not depend on how close any pair of subspaces is, indicating that a global solution to (6.3) may be more robust than other objectives to subspaces with high affinity.

However, (6.3) was recently shown to be even more difficult to solve than the K-means problem in the sense that it is NP-hard to *approximate* within a constant factor [76] in the worst case. As a result, researchers have turned to the use of alternating algorithms to obtain an approximate solution. Beginning with an initialization of K candidate subspace bases, KSS alternates between (i) clustering points by nearest subspace and (ii) obtaining new subspace bases by performing PCA on the points in each cluster. The algorithm is computationally efficient and guaranteed to converge to a local minimum [30, 200], but as with K-means, the KSS output is highly dependent on initialization. It is typically applied by performing many restarts and choosing the result with minimum cost (6.3) as the output.

This idea was extended to minimize the median residual (as opposed to mean) in [225], where a heuristic for intelligent initialization is also proposed. In [17], the authors use an alternating method based on KSS to perform online subspace clustering in the case of missing data. In [89], the authors propose a novel initialization method based on ideas from [226], and then perform the subspace update step using gradient steps along the Grassmann manifold. While this method is computationally efficient and improves upon the previous performance of KSS, it lacks theoretical guarantees. Most recently, the authors of [76] show that the subspace estimation step in KSS can be cast as a robust subspace recovery problem that can be efficiently solved using the Coherence Pursuit (CoP) algorithm [169]. The authors motivate the use of CoP by proving that it is capable of rejecting outliers from a UoS and demonstrate that replacing PCA with CoP results in strong empirical performance when there are many points per subspace. However, performance is limited when there are few points per subspace, and the algorithm performance is still highly dependent on the initialization. Moreover, CoP can be easily integrated into our proposed algorithm to provide improved performance.

Our method is based on the observation that the partially correct clustering in-

formation from each random initialization of KSS can be leveraged using *consensus clustering* in such a way that the consensus is much more informative than even the best single run. Unlike the above-mentioned variations on KSS, our proposed approach has cluster recovery guarantees, and its empirical performance is significantly stronger.

6.2.3 Consensus Clustering, Evidence Accumulation and Stability Selection

Ensemble methods have been used in the context of general clustering for some time and fall within the topic of *consensus clustering*, with an overview of the benefits and techniques given in [75]. The central idea behind these methods is to obtain many clusterings from a simple base clusterer, such as K-means, and then combine the results intelligently. To obtain different base clusterings, diversity of some sort must be incorporated. This is typically done by obtaining bootstrap samples of the data [125, 142], subsampling the data to reduce computational complexity [201], or performing random projections of the data [197]. Alternatively, the authors of [70, 71] use the randomness in different initializations of K-means to obtain diversity. We take this approach here for subspace clustering. After diversity is achieved, the base clustering results must be combined.

The *evidence accumulation clustering* framework laid out in [72] combines results by voting, i.e., creating a co-association matrix $\mathbf{A}$ whose (i, j)th entry is equal to the number of times two points are clustered together.[1] A theoretical framework for this approach is laid out in [34], where the entries of the co-association matrix are modeled as Binomial random variables. This approach is studied further in [133, 134], in which the clustering problem is solved as a Bregman divergence minimization. These models result in improved clustering performance over previous work but are not accompanied by any theoretical guarantees with regard to the resulting co-association matrix. Further, in our experiments, we did not find the optimization-based approach to perform as well as simply running spectral clustering on the resulting co-association matrix.

Subspace clustering can also be viewed through the lens of variable selection by restating the goal as follows: select from among all pairs of samples those that came from the same subspace. Each variable corresponds to a pair of samples and selection corresponds to clustering the two together. In this framework, the co-associations in evidence accumulation [72, Section 3.2] correspond to the selection probabilities of individual variables in stability selection [140, Definition 1]. This connection may be

[1] In the context of consensus clustering, we use the terms *affinity matrix* and *co-association matrix* interchangeably.

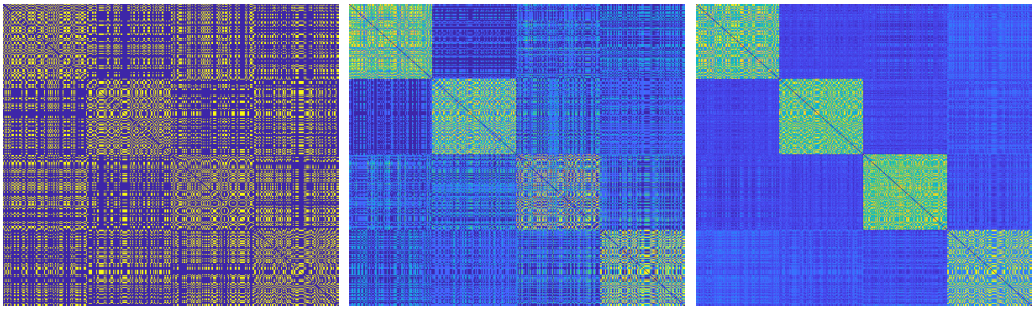


Figure 6.1: Co-association matrix of EKSS for $B = 1, 5, 50$ base clusterings. Data generation parameters are $D = 100$, $d = 3$, $K = 4$, $N = 400$, and the data is noise-free; the algorithm uses $\bar{K} = 4$ candidate subspaces of dimension $\bar{d} = 3$ and no thresholding. Resulting clustering errors are 61%, 25%, and 0%.

a promising avenue for further investigation, especially in the aspects where stability selection differs from our current approach. For example, [140] selects variables based on their stability paths, i.e., their selection probabilities over a sweep of regularization parameters, and using this to handle some of the tuning parameters in EKSS could be an interesting extension of our current work. Furthermore, [140] obtains diversity primarily by sub-sampling the data. The random initializations of EKSS are more closely related to the randomized lasso discussed in [140, Section 3.1].

The remainder of this chapter applies ideas from consensus clustering to the subspace clustering problem. We describe our Ensemble KSS algorithm and its guarantees and demonstrate the algorithm's state-of-the-art performance on both synthetic and real datasets.

6.3 Ensemble K-subspaces

This section describes our method for subspace clustering using ensembles of the K-subspaces algorithm, which we refer to as Ensemble K-subspaces (EKSS). Our key insight leading to EKSS is the fact that the partially-correct clustering information from each random initialization of KSS can be combined to form a more accurate clustering of the data. We therefore run several random initializations of KSS and form a co-association matrix combining their results that becomes the affinity matrix used in spectral clustering to obtain cluster labels.

In more technical detail, our EKSS algorithm proceeds as follows. For each of $b = 1, \ldots, B$ base clusterings, we obtain an estimated clustering $\mathcal{C}^{(b)}$ from a single run of KSS with a random initialization of candidate bases. The (i, j)th entry of

Algorithm 6.1 ENSEMBLE K-SUBSPACES (EKSS)

Input: $\mathcal{X} = \{ x_1, x_2, \ldots, x_N \} \subset \mathbb{R}^D$: data, $\bar{K}$: number of candidate subspaces, $\bar{d}$: candidate dimension, K: number of output clusters, q: threshold parameter, B: number of base clusterings, T: number of KSS iterations
Output: $\mathcal{C} = \{ c_1, \ldots, c_K \}$: clusters of $\mathcal{X}$
for $b = 1, \ldots, B$ (in parallel) **do**
    $\mathbf{U}_1, \ldots, \mathbf{U}_{\bar{K}} \overset{iid}{\sim} \text{Unif}(\text{St}(D, \bar{d}))$ Draw $\bar{K}$ random subspace bases
    $c_k \leftarrow \{ x \in \mathcal{X} \ : \ \forall j \ \left\|\mathbf{U}_k^T x\right\|_2 \geq \left\|\mathbf{U}_j^T x\right\|_2 \}$ for $k = 1, \ldots, \bar{K}$ Cluster by projection
    for $t = 1, \ldots, T$ (in sequence) **do**
        $\mathbf{U}_k \leftarrow \text{PCA}\left(c_k, \bar{d}\right)$ for $k = 1, \ldots, \bar{K}$ Estimate subspaces
        $c_k \leftarrow \{ x \in \mathcal{X} \ : \ \forall j \ \left\|\mathbf{U}_k^T x\right\|_2 \geq \left\|\mathbf{U}_j^T x\right\|_2 \}$ for $k = 1, \ldots, \bar{K}$ Cluster by projection
    end for
    $\mathcal{C}^{(b)} \leftarrow \{ c_1, \ldots, c_{\bar{K}} \}$
end for
$\mathbf{A}_{i,j} \leftarrow \frac{1}{B} \left| \{ b : x_i, x_j \text{ are co-clustered in } \mathcal{C}^{(b)} \} \right|$ for $i, j = 1, \ldots, N$ Form co-association matrix
$\bar{\mathbf{A}} \leftarrow \text{THRESH}(\mathbf{A}, q)$ Keep top q entries per row/column
$\mathcal{C} \leftarrow \text{SPECTRALCLUSTERING}(\bar{\mathbf{A}}, K)$ Final Clustering

the co-association matrix is the number of base clusterings for which x_i and x_j are clustered together. We then threshold the co-association matrix as in [91] by taking the top q values from each row/column. Once this thresholded co-association matrix is formed, cluster labels are obtained using spectral clustering. Algorithm 6.1 gives pseudocode for EKSS, where THRESH sets all but the top q entries in each row/column to zero as in [91] (pseudocode in Algorithm 6.2) and SPECTRALCLUSTERING [150] clusters the data points based on the co-association matrix $\mathbf{A}$. Note that the number of candidates $\bar{K}$ and candidate dimension $\bar{d}$ need not match the number K and dimension d of the true underlying subspaces. Figure 6.1 shows the progression of the co-association matrix as $B = 1, 5, 50$ base clusterings are used for noiseless data from $K = 4$ subspaces of dimension $d = 3$ in an ambient space of dimension $D = 100$ using $\bar{K} = 4$ candidates of dimension $\bar{d} = 3$. Section 6.3.2 discusses the choice of parameters for EKSS.

6.3.1 Computational Complexity

Recall the relevant parameters: K is the number of output clusters, $\bar{K}$ is the number of candidate subspaces in EKSS, $\bar{d}$ is the dimension of those candidates, N is the number of points, D is the ambient dimension, B is the number of KSS base clusterings to combine, and T is the number of iterations within KSS. To form the

Algorithm 6.2 AFFINITY THRESHOLD (THRESH)

Input: $\mathbf{A} \in [0,1]^{N \times N}$: affinity matrix, q: threshold parameter
Output: $\bar{\mathbf{A}} \in [0,1]^{N \times N}$: thresholded affinity matrix
for $i = 1, \ldots, N$ **do**
    $\mathbf{Z}^{\text{row}}_{i,:} \leftarrow \mathbf{A}_{i,:}$ with the smallest $N - q$ entries set to zero. Threshold rows
    $\mathbf{Z}^{\text{col}}_{:,i} \leftarrow \mathbf{A}_{:,i}$ with the smallest $N - q$ entries set to zero. Threshold columns
end for
$\bar{\mathbf{A}} \leftarrow \frac{1}{2}\left(\mathbf{Z}^{\text{row}} + \mathbf{Z}^{\text{col}}\right)$ Average

co-association matrix, the complexity of EKSS is $O(BT(\bar{K}D^2\bar{d} + \bar{K}D\bar{d}N))$. We run the KSS base clusterings in parallel and use very few iterations, making the functional complexity of EKSS $O(\bar{K}D^2\bar{d} + \bar{K}D\bar{d}N)$, which is competitive with existing methods. In comparison, TSC has complexity $O(DN^2)$ and SSC-ADMM has complexity $O(TN^3)$, where T is the number of ADMM iterations. Note that typically $N > D$ and sometimes much greater. We have not included the cost of spectral clustering, which is $O(KN^2)$. For most modern subspace clustering algorithms (except SSC-ADMM), this dominates the computational complexity as N grows.

6.3.2 Parameter Selection

EKSS requires six input parameters, whose selection we now discuss. As stated in Section 6.3.1, we use a small number of KSS iterations, setting $T = 3$ in all experiments. Typically, B should be chosen as large as computation time allows. In our experiments on real data, we choose $B = 1000$. The number of output clusters K is required for all subspace clustering algorithms, and methods such as those described in [90] can be used to estimate this value. Hence, the relevant parameters for selection are the candidate parameters $\bar{K}$ and $\bar{d}$ and the thresholding parameter q.

When possible, the candidate parameters should be chosen to match the true UoS parameters. In particular, it is advised to set $\bar{K} = K$ and $\bar{d} = d$ when they are known. In practice, a good approximating dimension for the underlying subspace is often known. For example, images of a Lambertian object under varying illumination are known to lie near a subspace with $d = 9$ [20] and moving objects in video are known to lie near an affine subspace with $d = 3$ [196]. However, as we will show in the following section, our theoretical guarantees hold even if there is model mismatch. Namely, the choice of $\bar{K} = 2$ and $\bar{d} = 1$ still provably yields correct clustering, though this results in a degradation of empirical performance.

The thresholding parameter q can be chosen according to data-driven techniques as in [90], or following the choice in [91]. In our experiments on real data, we select q (or the relevant thresholding parameter in the case of SSC) by sweeping over a large

range of values and choosing the value corresponding to the lowest clustering error. Note that q is applied to the co-association matrix $\mathbf{A}$, and hence the computational complexity of performing model selection is much lower than that of running the entire EKSS algorithm numerous times.

We briefly consider the parameters required by existing algorithms. SSC [67] and EnSC [216] both require two parameters to be selected when solving the sparse regression problem (6.2). SSC also performs thresholding on the affinity matrix, which in our experiments appears critical to performance on real data. See the author code of [67] for details. TSC requires the thresholding parameter q to be selected. To the best of our knowledge, no principled manner of selecting these parameters has been proposed in the literature, and we consider this an important issue for future study.

6.3.3 Base Clustering Accuracy

A natural heuristic to improve the clustering performance of EKSS is to add larger values to the co-association matrix for base clusterings believed to be more accurate, and smaller values for those believed to be less accurate. Here, we briefly describe one such approach. Note that Step 12 in EKSS is equivalent to adding a unit weight to each entry corresponding to co-clustered points, i.e., $\mathbf{A} \leftarrow \frac{1}{B}\sum_{b=1}^{B} \mathbf{A}^{(b)} w(b)$, where $\mathbf{A}^{(b)}_{i,j} := \mathbb{1}\left\{x_i, x_j \text{ are clustered together in } \mathcal{C}^{(b)}\right\}$ and $w(b) = 1$. The key idea is that this weight $w(b)$ can instead be chosen to reflect some estimation of the quality of the bth clustering; we propose using the KSS cost function as a measure of clustering quality. Let $\mathcal{C}^{(b)} = \{\, c_1^{(b)}, \ldots, c_K^{(b)} \,\}$ denote the bth base clustering, and let $\mathcal{U}^{(b)} = \{\, \mathbf{U}_1^{(b)}, \ldots, \mathbf{U}_K^{(b)} \,\}$ denote the set of subspace bases estimated by performing PCA on the points in the corresponding clusters. The clustering quality can then be measured as

$$w(b) = 1 - \sum_{k=1}^{K} \sum_{i:x_i \in c_k^{(b)}} \left\| x_i - \mathbf{U}_k^{(b)} \mathbf{U}_k^{(b)^T} x_i \right\|_2^2 / \left\|\mathbf{X}\right\|_F^2 , \tag{6.4}$$

a value between 0 and 1 that decreases as the KSS cost increases. We employ this value of $w(b)$ in all experiments on real data.

6.3.4 Alternative Ensemble Approaches

As KSS is known to perform poorly in many cases, one may wonder whether better performance can be obtained by applying the evidence accumulation framework to more recent algorithms such as SSC and GSC. We attempted such an approach by subsampling the data to obtain diversity in SSC-OMP [217] and EnSC [216]. However, the resulting clustering performance did not always surpass that of the base

algorithm run on the full dataset. Similar behavior occurred for ensembles of the GSC algorithm [162] as well as the Fast Landmark Subspace Clustering algorithm [209]. We also experimented with MKF as a base clustering algorithm but found little or no benefit at a significant increase in computational complexity. Hence, it seems that the success of our proposed approach depends both on the evidence accumulation framework *and* the use of KSS as a base clustering algorithm. Toward this end, we found that EKSS did benefit from the recent CoP-KSS algorithm [76] as a base clusterer for larger benchmark datasets, as discussed in Section 6.5. The appropriate combination of ensembles of other algorithms is nontrivial and an exciting open topic for future research.

6.4 Recovery Guarantees

Recovery guarantees for KSS remain elusive despite nearly twenty years of use since its introduction. Intelligent initialization methods based on probabilistic farthest insertion are used in [89, 225], but these too lack theoretical guarantees. This section provides a first step toward recovery guarantees for EKSS (Alg. 6.1). In particular, we show that (a) any "angle preserving" affinity matrix can be used to obtain clustering with guarantees matching those of state-of-the-art subspace clustering methods, and (b) EKSS has such an affinity matrix after the first KSS clustering step with high probability. Put together, these findings provide state-of-the-art guarantees for EKSS in the case where only the first KSS iteration is performed (i.e., $T = 0$ in Alg. 6.1). We refer to this parameter choice as *EKSS-0* and include explicit pseudocode for this specialization in Algorithm 6.3. To the best of our knowledge, our work is the first to provide any recovery guarantees for a KSS algorithm as well as the first characterization of a co-association matrix in the context of consensus clustering.

Section 6.4.1 presents the notion of an angle preserving affinity matrix and extends the guarantees of [91] to all algorithms that use such affinity matrices. Though developed here to analyze EKSS, these results apply broadly and provide a promising approach for analyzing other geometrically-based subspace clustering algorithms in the future. Section 6.4.2 shows that the affinity/co-association matrix of EKSS with $T = 0$ is angle preserving with high probability, and presents the resulting recovery guarantees: correct clustering for noiseless data and no false connections (NFC) for noisy data or data with missing entries.

We use N_{max} (N_{min}) throughout to refer to the maximum (minimum) number of points on any single subspace and d_{max} to refer to the maximum subspace dimension. The proofs of all results in this section are in Section 6.7.

Algorithm 6.3 EKSS-0

Input: $\mathcal{X} = \{ x_1, x_2, \ldots, x_N \} \subset \mathbb{R}^D$: data, $\bar{K}$: number of candidate subspaces, $\bar{d}$: candidate dimension, K: number of output clusters, q: threshold parameter, B: number of base clusterings,
Output: $\mathcal{C} = \{ c_1, \ldots, c_K \}$: clusters of $\mathcal{X}$
for $b = 1, \ldots, B$ (in parallel) **do**
    $\mathbf{U}_1, \ldots, \mathbf{U}_{\bar{K}} \overset{iid}{\sim} \mathrm{Unif}(\mathrm{St}(D, \bar{d}))$ Draw $\bar{K}$ random subspace bases
    $c_k \leftarrow \{ x \in \mathcal{X} \ : \ \forall j \ \left\|\mathbf{U}_k^T x\right\|_2 \geq \left\|\mathbf{U}_j^T x\right\|_2 \}$ for $k = 1, \ldots, \bar{K}$ Cluster by projection
    $\mathcal{C}^{(b)} \leftarrow \{ c_1, \ldots, c_{\bar{K}} \}$
end for
$\mathbf{A}_{i,j} \leftarrow \frac{1}{B} \left| \{ b : x_i, x_j \text{ are co-clustered in } \mathcal{C}^{(b)} \} \right|$ for $i, j = 1, \ldots, N$ Form affinity matrix
$\bar{\mathbf{A}} \leftarrow \textsc{Thresh}(\mathbf{A}, q)$ Keep top q entries per row/column
$\mathcal{C} \leftarrow \textsc{SpectralClustering}(\bar{\mathbf{A}}, K)$ Final Clustering

6.4.1 Recovery Guarantees for Angle Preserving Affinity Matrices

This section extends the NFC and connectedness guarantees of [91] to any algorithm that uses angle preserving affinity matrices. The key idea is that these affinity matrices sufficiently capture the information contained in pairwise angles and obtain good recovery when the angles differentiate the clusters well. Observe that using angles need not be a "goal" of such methods; deviating may in fact produce better performance in broader regimes, e.g., by incorporating higher order structure. Nevertheless, so long as the relative angles among points are sufficiently captured, the method immediately enjoys the guarantees of this section.

Definition 6.1 (Angle Preserving)**.** An affinity matrix $\mathbf{A}$ is *τ-angle preserving* for a set of points $\mathcal{X}$ with respect to a strictly increasing function $f : \mathbb{R}_+ \to \mathbb{R}_+$ if

$$|\mathbf{A}_{i,j} - f\left(|\langle x_i, x_j \rangle|\right)| \leq \tau, \quad i, j \in [N], \tag{6.5}$$

where we note that $\cos^{-1}\left(|\langle x_i, x_j \rangle|\right)$ is the angle between the points x_i and x_j.

Note that f is an arbitrary monotonic transformation that takes small angles (large absolute inner products) to large affinities and takes large angles (small absolute inner products) to small affinities, and τ quantifies how close the affinity matrix is to such a transformation. Taking $f(\alpha) = \alpha$ and $\tau = 0$ recovers the absolute inner product.

To guarantee correct clustering (as opposed to NFC only), it is sufficient to show that the thresholded affinity matrix has both NFC and exactly K connected components [91, Appendix A]. We formalize this fact for clarity in the proposition below.

Proposition 6.2 (NFC and connectedness give correct clustering [91, Equation (15)]). *Assume that the thresholded affinity matrix formed by an algorithm satisfies NFC with probability at least* $1-\varepsilon_1$ *and given NFC satisfies the connectedness condition with probability at least* $1-\varepsilon_2$. *Then spectral clustering correctly identifies the components with probability at least* $1-\varepsilon_1-\varepsilon_2$.

Thus, we study conditions under which NFC and connectedness are guaranteed; conditions for correct clustering follow. In particular, we provide upper bounds on τ that guarantee NFC (Theorem 6.4) and connectedness (Theorem 6.5). The upper bound for NFC is given by a property of the data that we call the *q-angular separation*, defined as follows. We later bound this quantity in a variety of contexts.

Definition 6.3 (Angular Separation). The *q-angular separation* ϕ_q of the points $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$ with respect to a strictly increasing function $f : \mathbb{R}_+ \to \mathbb{R}_+$ is

$$\phi_q = \min_{l\in[K], i\in[N_l]} \frac{f\left(\left|\left\langle x_i^{(l)}, x_{\neq i}^{(l)}\right\rangle\right|_{[q]}\right) - f\left(\max_{k\neq l, j\in[N_k]}\left|\left\langle x_i^{(l)}, x_j^{(k)}\right\rangle\right|\right)}{2}, \tag{6.6}$$

where $x_i^{(l)}$ denotes the ith point of $\mathcal{X}_l$, and $\left|\left\langle x_i^{(l)}, x_{\neq i}^{(l)}\right\rangle\right|_{[q]}$ denotes the q^{th} largest absolute inner product between the point $x_i^{(l)}$ and other points in subspace l.

In words, the q-angular separation quantifies how far apart the clusters are, as measured by the transformed absolute inner products. When this quantity is positive and large, pairwise angles differentiate clusters well. The following theorem connects this data property to angle preserving affinity matrices.

Theorem 6.4 (No false connections (NFC)). *Suppose* $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$ *have q-angular separation* ϕ_q *with respect to a strictly increasing function* f. *Then the q-nearest neighbor graph for any* ϕ_q*-angle preserving affinity matrix (with respect to* f*) has no false connections.*

Theorem 6.4 states that sufficiently small deviation τ guarantees NFC as long as the data has positive q-angular separation. The next theorem provides an upper bound on τ that guarantees connectedness within a cluster with high probability given NFC. Under NFC, the q-nearest neighbors of any point (with respect to the affinity matrix) are in the same subspace, and so the theorem is stated with respect to only points within a single subspace. In particular, we restrict to the d-dimensional subspace and so consider the q-nearest neighbor graph $\tilde{G}$ for points $a_1, \ldots, a_n$ uniformly distributed on the sphere $\mathbb{S}^{d-1}$.

Theorem 6.5 (Connectedness). *Let $a_1, \dots, a_n \in \mathbb{R}^d$ be i.i.d. uniform on $\mathbb{S}^{d-1}$ and let $\tilde{G}$ be their corresponding q-nearest neighbor graph formed from a τ-angle preserving affinity matrix. Let $\gamma \in (1, n/\log n)$ be arbitrary, and let θ be the spherical radius of a spherical cap covering $\gamma \log n/n$ fraction of the area of $\mathbb{S}^{d-1}$. Suppose $q \in [4(24\pi)^{d-1}\gamma \log n, n]$ and $\theta < \pi/48$. Then if $\tau < C_3$,*

$$\mathbb{P}\{\tilde{G} \text{ is connected } \} \geq 1 - \frac{2}{n^{\gamma-1}\gamma \log n}, \tag{6.7}$$

where C_3 depends only on d, n, f, and γ and is defined in the proof.

We now provide explicit high-probability lower bounds on the q-angular separation ϕ_q from (6.6) in some important settings relevant to subspace clustering. These results can be used to guarantee NFC by bounding the deviation level τ. Consider first the case where there is no intersection between any pair of subspaces but there are potentially unobserved entries, i.e., missing data. Lemma 6.6 bounds ϕ_q from below in such a setting; the bound depends on a variant of the minimum principal angle between subspaces that accounts for missing data.

Lemma 6.6 (Angular separation for missing data). *Let $\mathcal{S}_k$, $k = 1, \dots, K$ be subspaces of dimension $d_1, \dots, d_K$ in $\mathbb{R}^D$. Let the N_k points in $\mathcal{X}_k$ be drawn as $x_j^{(k)} = \mathbf{U}^{(k)} a_j^{(k)}$, where $a_j^{(k)}$ are i.i.d. uniform on $\mathbb{S}^{d_k-1}$ and $\mathbf{U}^{(k)} \in \mathbb{R}^{D \times d_k}$ has (not necessarily orthonormal) columns that form a basis for $\mathcal{S}_k$. In each $x_j \in \mathcal{X}$, up to s entries are then unobserved, i.e., set to zero. Let $\rho \in [0, 1)$ be arbitrary and suppose that $N_{min} > N_0$, where N_0 is a constant that depends only on d_{max} and ρ. Suppose that $q < N_{min}^{\rho}$ and*

$$r_s = \frac{\max_{k,l:k\neq l,\mathcal{D}:|\mathcal{D}|\leq 2s} \left\| {\mathbf{U}_{\mathcal{D}}^{(k)}}^{\top} \mathbf{U}^{(l)} \right\|_2}{\min_{l,\mathcal{D}:|\mathcal{D}|\leq 2s,\|a\|=1} \left\| {\mathbf{U}_{\mathcal{D}}^{(l)}}^{\top} \mathbf{U}^{(l)} a \right\|_2} < 1, \tag{6.8}$$

where $\mathbf{U}_{\mathcal{D}}^{(l)}$ is the matrix obtained from $\mathbf{U}^{(l)}$ by setting the rows indexed by $\mathcal{D} \subset \{1, \dots, D\}$ to zero. Then $\phi_q > C_1$ with probability at least $1 - \sum_{k=1}^{K} N_k e^{-c_1(N_k-1)}$, where $c_1 > 0$ is a numerical constant that depends on N_{min}^{ρ}, and $C_1 > 0$ depends only on r_s and f. Both c_1 and C_1 are defined in the proof.

To gain insight to the above lemma, note that for full data $s = 0$, and r_s simplifies to $\max_{k,l:k\neq l} ||{\mathbf{U}^{(k)}}^{\top} \mathbf{U}^{(l)}||_2$, which is less than one if and only if there is no intersection between subspaces. In this case, Lemma 6.6 states that ϕ_q is positive (i.e., NFC is achievable) as long as there is no intersection between any pair of subspaces. We next consider the case where the subspaces are allowed to intersect and points may be corrupted by additive noise. Lemma 6.7 bounds ϕ_q from below in such a setting;

it requires the subspaces to be sufficiently far apart with respect to their affinity, which is defined as [91, 222]

$$\text{aff}(\mathcal{S}_k, \mathcal{S}_l) = \frac{1}{\sqrt{d_k \wedge d_l}} \left\| \mathbf{U}_k^T \mathbf{U}_l \right\|_F , \tag{6.9}$$

where $\mathbf{U}_k$ and $\mathbf{U}_l$ form orthonormal bases for the d_k- and d_l-dimensional subspaces $\mathcal{S}_k$ and $\mathcal{S}_l$. Note that $\text{aff}(\mathcal{S}_k, \mathcal{S}_l)$ is a measure of how close two subspaces are in terms of their principal angles and takes the value 1 if two subspaces are equivalent and 0 if they are orthogonal.

Lemma 6.7 (Angular separation for noisy data)**.** *Let the points in $\mathcal{X}_k$ be the set of N_k points $x_i^{(k)} = y_i^{(k)} + e_i^{(k)}$, where the $y_i^{(k)}$ are drawn i.i.d. from the set $\{ y \in \mathcal{S}_k : \|y\| = 1 \}$, independently across k, and the $e_i^{(k)}$ are i.i.d. $\mathcal{N}(0, \frac{\sigma^2}{D} I_D)$. Let $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$ and $q < N_{min}/6$. Suppose that*

$$\max_{k,l:k \neq l} \text{aff}(\mathcal{S}_k, \mathcal{S}_l) + \frac{\sigma(1+\sigma)}{\sqrt{\log N}} \frac{\sqrt{d_{max}}}{\sqrt{D}} \leq \frac{1}{15 \log N}, \tag{6.10}$$

with $D > 6 \log N$. Then $\phi_q > C_2$ with probability at least $1 - \frac{10}{N} - \sum_{k=1}^{K} N_k e^{-c_2(N_k - 1)}$, where $c_2 > 0$ is a numerical constant, and $C_2 > 0$ depends only on σ, D, d_{max}, N, $\max_{k,l:k \neq l} \text{aff}(\mathcal{S}_k, \mathcal{S}_l)$, and f. Both c_2 and C_2 are defined in the proof.

Lemmas 6.6 and 6.7 state that under certain conditions on the arrangement of subspaces and points, the separation ϕ_q defined in (6.6) is positive with high probability and with given lower bounds. In the following section, we show that taking sufficiently many base clusterings B in EKSS-0 guarantees the affinity matrix is sufficiently angle preserving with high probability.

6.4.2 EKSS-0 Recovery Guarantees

This section shows that the co-association/affinity matrix formed by EKSS-0 is angle preserving, leading to a series of recovery guarantees for the problem of subspace clustering. We say that two points are *co-clustered* if they are assigned to the same candidate subspace in line 5 of Algorithm 6.1 (note that lines 6-9 are not computed for EKSS-0). The key to our guarantees lies in the fact that for points lying on the unit sphere, the probability of co-clustering is a monotonically increasing function of the absolute value of their inner product, as shown in Lemma 6.9 below. For EKSS-0, the entries of the affinity matrix $\mathbf{A}$ are empirical estimates of these probabilities, and hence the deviation level τ is appropriately bounded with high probability by taking sufficiently many base clusterings B. These results allow us to apply Theorems 6.4 and 6.5 from the previous section. We remind the reader that

the parameters $\bar{K}$ and $\bar{d}$ are the number and dimension of the *candidate* subspaces in EKSS, and need not be related to the data being clustered.

Theorem 6.8 (EKSS-0 is angle preserving)**.** *Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ be the affinity matrix formed by EKSS-0 (line 12, Alg. 6.1) with parameters $\bar{K}, \bar{d}$ and B. Let $\tau > 0$. Then with probability at least $1 - N(N-1)e^{-c_3\tau^2 B}$, the matrix $\mathbf{A}$ is τ-angle preserving, where the increasing function $f_{\bar{K},\bar{d}}$ is defined in the proof of Lemma 6.9, $c_3 = 2\sqrt{\log 2}$, and the probability is taken with respect to the random subspaces drawn in EKSS-0 (line 4, Alg. 6.1).*

In the context of the previous section, Theorem 6.8 states that the affinity matrix formed by EKSS-0 is τ-angle preserving and hence satisfies the main condition required for Theorems 6.4 and 6.5. We refer to the transformation function as $f_{\bar{K},\bar{d}}$ to denote the dependence on the EKSS-0 parameters, noting that $f_{\bar{K},\bar{d}}$ is increasing for *any* natural numbers $\bar{K}$ and $\bar{d}$. A consequence of Theorem 6.8 is that by increasing the number of base clusterings B, we can reduce the deviation level τ to be arbitrarily small while maintaining a fixed probability that the model holds. This fact allows us to apply the results of the previous section to provide recovery guarantees for EKSS-0. The major nontrivial aspect of proving Theorem 6.8 lies in establishing the following lemma.

Lemma 6.9. *The (i,j)th entry of the affinity matrix $\mathbf{A}$ formed by EKSS-0 (line 12, Alg. 6.1) has expected value*

$$\mathbb{E}\mathbf{A}_{i,j} = f_{\bar{K},\bar{d}}(|\langle x_i, x_j\rangle|) \tag{6.11}$$

where $f_{\bar{K},\bar{d}} : \mathbb{R}_+ \to \mathbb{R}_+$ is a strictly increasing function (defined in the proof), and the expectation is taken with respect to the random subspaces drawn in EKSS-0 (line 4, Alg. 6.1). The subscripts $\bar{K}$ and $\bar{d}$ indicate the dependence of $f_{\bar{K},\bar{d}}$ on those EKSS-0 parameters.

Proof. We provide a sketch of the proof here; the full proof can be found in Section 6.7. For notational compactness, we instead prove that the probability of two points being co-clustered is a *decreasing* function of the angle θ between them. Denote this probability by $p_{\bar{K},\bar{d}}(\theta)$. Let $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_{\bar{K}} \in \mathbb{R}^{D\times\bar{d}}$ be the K candidate bases. Let $\tilde{p}(\theta)$ be the probability that any two points with corresponding angle θ are assigned to the candidate $\mathbf{U}_1$. Then by symmetry we have $p_{\bar{K},\bar{d}}(\theta) = K\tilde{p}(\theta)$, and it suffices to prove that $\tilde{p}$ is strictly decreasing. Without loss of generality, let $x_i = e_1$ and $x_j = \cos(\theta)e_1 + \sin(\theta)e_2$, where $e_m \in \mathbb{R}^D$ is the mth standard basis vector. We then have that

$$\tilde{p}(\theta) = \mathbb{P}\left\{\, \mathbf{Q}x_i, \mathbf{Q}x_j \text{ both assigned to } \mathbf{U}_1 \,\right\},$$

where $\mathbf{Q}$ is an arbitrary orthogonal transformation of $\mathbb{R}^D$. Let E denote the event of interest and L denote the span of e_1 and e_2. The event E can then be written as

$$z^T\mathbf{Q}\mathbf{P}_L(\mathbf{P}_1 - \mathbf{P}_k)\mathbf{P}_L\mathbf{Q}z > 0, \quad \text{for} \quad 1 < k \leq K \quad \text{and} \quad z = x_i, x_j, \tag{6.12}$$

where $\mathbf{P}_L$ denotes the orthogonal projection onto the subspace L and $\mathbf{P}_k$ denotes the orthogonal projection onto the subspace spanned by $\mathbf{U}_k$. By restricting to L, (6.12) can be reduced to a two-dimensional quadratic form, and we can compute in closed form $\mathbb{P}\{\, E \mid \mathbf{U}_1, \ldots, \mathbf{U}_{\bar{K}} \,\}$. Differentiating shows that this term is decreasing and hence (by the law of total probability) so is $\tilde{p}(\theta)$. □

Another approach to proving Lemma 6.9 and Theorem 6.8 for $\bar{K} = 2$ may be to observe that in this case the co-association $\mathbf{A}_{i,j}$ is closely related to the Kendall rank correlation between the random variables $X_i = \|\mathbf{U}^T x_i\|_2$ and $X_j = \|\mathbf{U}^T x_j\|_2$ since

$$\mathbb{E}\mathbf{A}_{i,j} = \frac{1 + \mathbb{E}\Big\{ \operatorname{sign}\left(\left\|\mathbf{U}_1^T x_i\right\|_2 - \left\|\mathbf{U}_2^T x_i\right\|_2 \right) \operatorname{sign}\left(\left\|\mathbf{U}_1^T x_i\right\|_2 - \left\|\mathbf{U}_2^T x_j\right\|_2 \right) \Big\}}{2}. \tag{6.13}$$

This connection makes it possible to draw on ideas and insights from the analysis of Kendall rank correlations. For example, as discussed in [19, Section 2], the Kendall rank correlation is monotonically related to the Pearson correlation for transelliptical distributions, and [19, Section 4.3] provides related error bounds for an empirical estimator. Using these results would involve either establishing that (X_i, X_j) is transelliptical or extending [19] to other distributions.

Note that the result of Lemma 6.9 does not depend on the underlying data distribution, i.e., the number or arrangement of subspaces, but instead says that clustering with EKSS-0 is (in expectation) a function of the absolute inner product between points, regardless of the parameters. Thus, the results of this section all hold even with the simple parameter choice of $\bar{K} = 2$ and $\bar{d} = 1$ in Algorithm 6.1. Our empirical results suggest that decreasing $\bar{K}$ and increasing $\bar{d}$ increases the probability of co-clustering. However, when running several iterations of KSS (EKSS with $T > 0$), we find that it is advantageous to choose $\bar{K}$ and $\bar{d}$ to match the true parameters of the data as closely as possible, allowing KSS to more accurately model the underlying subspaces.

Combined with the results of Section 6.4.1, Theorem 6.8 enables us to quickly obtain recovery guarantees for EKSS-0, which we now present. We first consider the case where the data are noiseless, i.e., lie perfectly on a union of K subspaces. Theorems 6.10 and 6.11 provide sufficient conditions on the arrangement of subspaces such that EKSS-0 achieves *correct clustering* with high probability.

Theorem 6.10 (EKSS-0 provides correct clustering for disjoint subspaces)**.** *Let $\mathcal{S}_k$, $k = 1, \dots, K$ be subspaces of dimension $d_1, \dots, d_K$ in $\mathbb{R}^D$. Let the N_k points in $\mathcal{X}_k$ be drawn as $x_j^{(k)} = \mathbf{U}^{(k)} a_j^{(k)}$, where $a_j^{(k)}$ are i.i.d. uniform on $\mathbb{S}^{d_k - 1}$ and $\mathbf{U}^{(k)} \in \mathbb{R}^{D \times d_k}$ has orthonormal columns that form a basis for $\mathcal{S}_k$. Let $\rho \in [0, 1)$ be arbitrary and suppose that $N_{min} > N_0$, where N_0 is a constant that depends only on d_{max} and ρ. Suppose that $q \in [c_4 \log N_{max}, N_{min}^{\rho}]$ and*

$$r_0 = \max_{k,l:k \neq l} \left\| {\mathbf{U}^{(k)}}^{\top} \mathbf{U}^{(l)} \right\|_2 < 1, \tag{6.14}$$

where $c_4 = 12(24\pi)^{d_{max}-1}$. Then $\bar{\mathbf{A}}$ obtained by EKSS-0 results in correct clustering of the data with probability at least $1 - \sum_{k=1}^{K} \left(N_k e^{-c_1 (N_k - 1)} + 2N_k^{-2} \right) - N(N-1) e^{-c_3 B \min\{ C_1, C_3 \}^2}$, where $c_1, c_3 > 0$ are numerical constants, $C_1 > 0$ depends on r_0 and the function $f_{\bar{K},\bar{d}}$ defined in Theorem 6.8, and $C_3 > 0$ depends on d_{max}, N_{min}, and $f_{\bar{K},\bar{d}}$.

Theorem 6.11 (EKSS-0 provides correct clustering for subspaces with bounded affinity)**.** *Let $\mathcal{S}_k$, $k = 1, \dots, K$ be subspaces of dimension $d_1, \dots, d_K$ in $\mathbb{R}^D$. Let the points in $\mathcal{X}_k$ be a set of N_k points drawn uniformly from the unit sphere in subspace k, i.e., from the set $\{ x \in \mathcal{S}_k : \|x\| = 1 \}$. Let $\mathcal{X} = \mathcal{X}_1 \cup \dots \cup \mathcal{X}_K$ and $N = \sum_k N_k$. Let $q \in [c_4 \log N_{max}, N_{min}/6)$, where $c_4 = 12(24\pi)^{d_{max}-1}$. If*

$$\max_{k,l:k \neq l} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l) \leq \frac{1}{15 \log N},$$

then $\bar{\mathbf{A}}$ obtained by EKSS-0 results in correct clustering of the data with probability at least $1 - \frac{10}{N} - \sum_{k=1}^{K} \left(N_k e^{-c_2 (N_k - 1)} - 2N_k^{-2} \right) - N(N-1) e^{-c_3 B \min\{ C_2, C_3 \}^2}$, where $c_2, c_3 > 0$ are numerical constants, $C_2 > 0$ depends only on $\max_{k,l:k \neq l} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l)$, D, d_{max}, N, and the function $f_{\bar{K},\bar{d}}$ defined in Theorem 6.8, and $C_3 > 0$ depends on d_{max}, N_{min}, and $f_{\bar{K},\bar{d}}$.

We next consider two forms of data corruption. Theorem 6.12 shows that the affinity matrix built by EKSS-0 has NFC in the presence of data corrupted by additive Gaussian noise. Theorem 6.13 shows that EKSS-0 maintains NFC even in the presence of a limited number of missing (unobserved) entries.

Theorem 6.12 (EKSS-0 has NFC with noisy data)**.** *Let $\mathcal{S}_k$, $k = 1, \dots, K$ be subspaces of dimension $d_1, \dots, d_K$ in $\mathbb{R}^D$. Let the points in $\mathcal{X}_k$ be the set of N_k points $x_i^{(k)} = y_i^{(k)} + e_i^{(k)}$, where the $y_i^{(k)}$ are drawn i.i.d. from the set $\{ y \in \mathcal{S}_k : \|y\| = 1 \}$, independently across k, and the $e_i^{(k)}$ are i.i.d. $\mathcal{N}(0, \frac{\sigma^2}{D} \mathbf{I}_D)$. Let $\mathcal{X} = \mathcal{X}_1 \cup \dots \cup \mathcal{X}_K$ and $q < N_{min}/6$. If*

$$\max_{k,l:k \neq l} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l) + \frac{\sigma(1+\sigma)}{\sqrt{\log N}} \frac{\sqrt{d_{max}}}{\sqrt{D}} \leq \frac{1}{15 \log N},$$

with $D > 6 \log N$, then $\bar{\mathbf{A}}$ obtained from running EKSS-0 has no false connections with probability at least $1 - \frac{10}{N} - \sum_{k=1}^{K} N_k e^{-c_2(N_k-1)} - N(N-1)e^{-c_3 C_2^2 B}$, where $c_2, c_3 > 0$ are numerical constants, and $C_2 > 0$ depends only on $\max_{k \neq l} aff(\mathcal{S}_k, \mathcal{S}_l)$, σ, D, d_{max}, N and the function $f_{\bar{K},\bar{d}}$ defined in Theorem 6.8.

Theorem 6.13 (EKSS-0 has NFC with missing data)**.** *Let the n points in $\mathcal{X}_k$ be drawn as $x_j^{(k)} = \mathbf{U}^{(k)} a_j^{(k)}$, where $a_j^{(k)}$ are i.i.d. uniform on $\mathbb{S}^{d-1}$ and the entries of $\mathbf{U}^{(k)} \in \mathbb{R}^{D \times d}$ are i.i.d. $\mathcal{N}(0, \frac{1}{D})$. Let $\rho \in [0, 1)$ be arbitrary and suppose that $n > N_0$, where N_0 is a constant that depends only on d and ρ. Suppose that $q < n^\rho$, and assume that in each $x_j \in \mathcal{X}$ up to s arbitrary entries are unobserved, i.e., set to 0. Let $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$. If*

$$D - 3c_5 d - c_5 \log K \geq s \left(c_5 \log \left(\frac{De}{2s} \right) + c_6 \right), \tag{6.15}$$

then $\bar{\mathbf{A}}$ obtained by EKSS-0 has no false connections with probability at least $1 - Ne^{-c_1(n-1)} - N(N-1)e^{-c_3 C_1^2 B} - 4e^{-c_7 D}$, where $c_1, c_3, c_5, c_6, c_7 > 0$, are numerical constants and $C_1 > 0$ depends only on the ratio r_s defined in (6.8) and the function $f_{\bar{K},\bar{d}}$ defined in Theorem 6.8.

6.4.3 Discussion of Results

The data model considered in Theorems 6.10-6.13 is known as the "semi-random" model [187], due to the fixed arrangement of subspaces with randomly-drawn points, and has been analyzed widely throughout the subspace clustering literature [91, 92, 187, 188, 210]. Our guarantees under this model are identical (up to constants and log factors) to those for TSC and SSC (see [91, Section VII] for further discussion of their guarantees). The key difference between our results and those of TSC is that we pay a $N(N-1)/2e^{-c_3 \tau^2 B}$ penalty in recovery probability due to the approximate observations of the transformed inner products. Although our experiments indicate that EKSS-0 appears to have no benefits over TSC, we do find that by running a small number of KSS iterations, significant performance improvements are achieved. While the above analysis holds only for the case of $T = 0$, letting $T > 0$ is guaranteed to not increase the KSS cost function [30]. In our experiments, we found that setting $T > 0$ uniformly improved clustering performance, and our empirical results indicate that EKSS is in fact more robust (than EKSS-0 and TSC) to subspaces with small principal angles.

While the explicit choice of B is tied to the unknown function $f_{\bar{K},\bar{d}}$, our results provide intuition for setting this value; namely, the closer the underlying subspaces (in terms of principal angles), the more base clusterings required. The inverse dependence on $\log N$ in Theorems 6.11 and 6.12 indicates a tension as the problem

size grows. On one hand, points from the same subspace are more likely to be close when N is large, improving the angular separation. On the other hand, points are also more likely to fall near the intersection of subspaces, potentially degrading the angular separation. In all experimental results, we see that both EKSS and TSC perform better with larger N. Finally, we note that the leading $O\left(N^2\right)$ coefficient in the above probabilities results from applying a union bound and is likely conservative.

6.5 Experimental Results

This section demonstrates the performance in terms of clustering error (defined in Section 6.8.1) of EKSS on both synthetic and real datasets. We first show the performance of our algorithm as a function of the relevant problem parameters and verify that EKSS-0 exhibits the same empirical performance as TSC, as expected based on our theoretical guarantees. We also show that EKSS can recover subspaces that either have large intersection or are extremely close. We then demonstrate on benchmark datasets that EKSS not only improves over previous geometric methods, but that it achieves state-of-the-art results competitive with those obtained by self-expressive methods. Unless otherwise specified, we use $T = 3$ iterations in EKSS and $B = 1000$ base clusterings in EKSS-0 and EKSS, as described in Section 6.3.2. The experiments in this section were produced by Dr. John Lipor.

6.5.1 Synthetic Data

For all experiments in this section, we take $q = \max(3, \lceil N_k/20 \rceil)$ for EKSS-0 and TSC and $q = \max(3, \lceil N_k/6 \rceil)$ for EKSS, where $\lceil \cdot \rceil$ denotes the largest integer greater than or equal to its argument. We set $B = 10000$ for EKSS-0 and EKSS. When the angles between subspaces are not explicitly specified, it is assumed that the subspaces are drawn uniformly at random from the set of all d-dimensional subspaces of $\mathbb{R}^D$. For all experiments, we draw points uniformly at random from the unit sphere in the corresponding subspace and show the mean error over 100 random problem instances. We use the code provided by the authors for TSC and SSC. We employ the ADMM implementation of SSC and choose the parameters that result in the best performance in each scenario.

We explore the influence of some relevant problem parameters on the EKSS algorithm in Fig. 6.2. We take the ambient dimension to be $D = 100$, the number of subspaces to be $K = 3$, and generate noiseless data. We first consider the dependence on subspace dimension and the number of points per subspace. The top row of Fig. 6.2 shows the misclassification rate as the number of points per subspace ranges from $10 - 500$ and the subspace dimension ranges from $1 - 75$. When $2d > D$

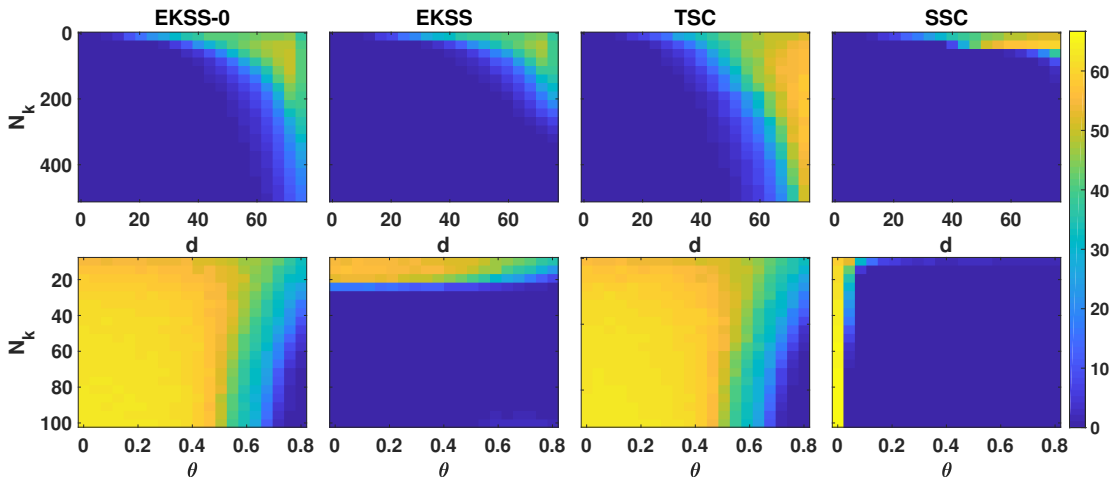


Figure 6.2: Clustering error (%) for proposed and state-of-the-art subspace clustering algorithms as a function of problem parameters N_k, number of points per subspace, and true subspace dimension d or angle between subspaces θ. Fixed problem parameters are $D = 100$, $K = 3$.

(i.e., $d \geq 51$), pairs of subspaces necessarily have intersection, and the intersection dimension grows with d. First, the figures demonstrate that EKSS-0 achieves roughly the same performance as TSC, resulting in correct clustering even in the case of subspaces with large intersection. Second, we see that EKSS can correctly cluster for subspace dimensions larger than that of TSC as long as there are sufficiently many points per subspace. For large subspace dimensions with a moderate number of points per subspace, SSC achieves the best performance.

We next explore the clustering performance as a function of the distance between subspaces, as shown in the second row of Fig. 6.2. We set the subspace dimension to $d = 10$ and generate $K = 3$ subspaces such that the principal angles between subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, as well as those between $\mathcal{S}_1$ and $\mathcal{S}_3$ are θ, for 20 values in the range $[0.001, 0.8]$. Most strikingly, EKSS is able to resolve subspaces with even the smallest separation. This stands in contrast to TSC; it fails in this regime because when the subspaces are extremely close, the inner products between points on different subspaces can be nearly as large as those within the same subspace. Similarly, in the case of SSC, points on different subspaces can be used to regress any given point with little added cost, and so it fails at very small subspace angles. However, as long as there is still some separation between subspaces, EKSS is able to correctly cluster all points. The theory presented here does not capture this phenomenon, and recovery guarantees that take into account multiple iterations of KSS are an important topic for future work.

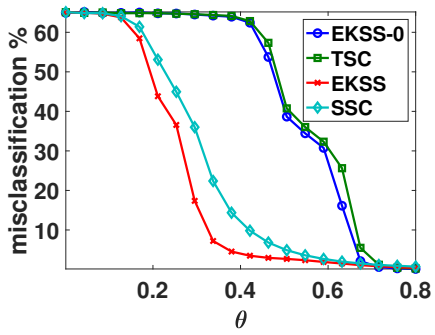


Figure 6.3: Clustering error (%) as a function of subspace angles with noisy data. Problem parameters are $D = 100$, $d = 10$, $K = 3$, $N_k = 500$, $\sigma^2 = 0.05$.

As a final comparison, we show the clustering performance with noisy data. Fig. 6.3 shows the clustering error as a function of the angle between subspaces for the case of $K = 3$ subspaces of dimension $d = 10$, with $N_k = 500$ points corrupted by zero-mean Gaussian noise with covariance $0.05\mathbf{I}_D$. We again consider 20 values of the angle θ between 0.001 and 0.08. EKSS-0 and TSC obtain similar performance, and more importantly EKSS is more robust to small subspace angles than SSC, even in the case of noisy data.

6.5.2 Benchmark Data

This section shows that EKSS achieves competitive subspace clustering performance on a variety of datasets commonly used as benchmarks in the subspace clustering literature. We consider the Hopkins-155 dataset [198], the cropped Extended Yale Face Database B [74, 123], COIL-20 [149] and COIL-100 [148] object databases, the USPS dataset provided by [37], and 10,000 digits of the MNIST handwritten digit database [119], where we obtain features using a scattering network [32] as in [216]. Descriptions of these datasets and the relevant problem parameters are included in Section 6.8.2. We compare the performance of EKSS to several benchmark algorithms: KSS [30], CoP-KSS [76], Median K-Flats (MKF) [225], TSC [91], the ADMM implementation of SSC [67], SSC-OMP [217], and Elastic Net Subspace Clustering (EnSC) [216]. For all algorithms, we selected the parameters that yielded the lowest clustering error, performing extensive model selection where possible. We point out that this method of parameter selection requires knowledge of the ground truth labels, which are typically unavailable in practice. For the larger USPS and MNIST datasets, we obtained a small benefit by replacing PCA (line 7, Alg. 6.1) with the more robust Coherence Pursuit, i.e., we use CoP-KSS as a base clustering algorithm instead of KSS. Further implementation details, including parameter selection and

Algorithm	Hopkins	Yale B	COIL-20	COIL-100	USPS	MNIST-10k
EKSS	**0.26**	14.31	13.47	**28.57**	**15.84**	**2.39**
KSS	0.35	54.28	33.12	66.04	18.31	2.60
CoP-KSS	0.69	52.59	29.10	51.38	**7.73**	**2.57**
MKF	**0.24**	41.32	35.69	59.50	28.49	28.17
TSC	2.07	22.20	15.28	29.82	31.57	15.98
SSC-ADMM	1.07	**9.83**	**13.19**	44.06	56.61	19.17
SSC-OMP	25.25	**13.28**	27.29	34.79	77.94	19.19
EnSC	9.75	18.87	**8.26**	**28.75**	33.66	17.97

Table 6.1: Clustering error (%) of subspace clustering algorithms for a variety of benchmark datasets. The lowest two clustering errors are given in bold. Note that EKSS is among the best three for all datasets, but no other algorithm is in the top five across the board.

data preprocessing, can be found in Section 6.8.2.

The clustering error for all datasets and algorithms is shown in Table 6.1, with the lowest two errors given in bold. First, note that EKSS outperforms its base clustering algorithm (KSS or CoP-KSS) in all cases except the USPS dataset, and sometimes by a very large margin. This result emphasizes the importance of leveraging all clustering information from the B base clusterings, as opposed to simply choosing the best single clustering. While CoP-KSS achieves lower clustering error than EKSS on the USPS dataset, a deeper investigation of the performance of CoP-KSS revealed that only 17 of the 1000 individual clusterings achieved an error lower than the 15.84% obtained by EKSS. A more sophisticated weighting scheme than that described in Section 6.3.3 could be employed to add more significant weights for the small number of base clusterings corresponding to low error. Alternative measures of clustering quality based on subspace margin [130] or novel internal clustering validation metrics [131] may provide improved performance. Next, the results show that EKSS is among the top performers in all datasets considered, achieving nearly perfect clustering of the Hopkins-155 dataset, which is known to be well approximated by the UoS model. Scalable algorithms such as SSC-OMP and EnSC perform poorly on this dataset, likely due to the small number of points. For the larger COIL-100, USPS, and MNIST datasets, EKSS also achieves strong performance, demonstrating its flexibility to perform well in both the small and large sample regimes. The self-expressive methods outperform EKSS on the Yale and COIL-20 datasets, likely due to the fact that they do not explicitly rely on the UoS model in building the affinity matrix. However, EKSS still obtains competitive performance on both datasets, making it a strong choice for a general-purpose algorithm for subspace clustering.

6.6 Discussion

In this work, we presented the first known theoretical guarantees for both evidence accumulation clustering and the KSS algorithm. We showed that with a given choice of parameters, the EKSS algorithm can provably cluster data from a union of subspaces under the same conditions as existing algorithms. The theoretical guarantees presented here match existing guarantees in the literature, and our experiments on synthetic data indicate that the iterative approach of KSS provides a major improvement in robustness to small angles between subspaces. Further, our results generalize those in the existing literature, yielding the potential to inform future algorithm design and analysis. We demonstrated the efficacy of our approach on both synthetic and real data, and showed that our method achieves excellent performance on several real datasets.

A number of important open problems remain. First, extending our analysis to the general case of Alg. 6.1 (i.e., $T > 0$) is an important next step that is difficult because of the alternating nature of KSS. In selecting tuning parameters, we chose the combination that resulted in the lowest clustering error, which is not known in practice. Methods for unsupervised model selection are an important practical consideration for EKSS and subspace clustering in general. Drawing connections to stability selection [140] and its extensions, e.g., subspace stability selection [193], may yield some interesting new approaches and insights. Random-projection ensemble classification with screened projections [40, Section 3] also bears some similarity to EKSS; considering connections in the theoretical analysis might provide some new perspectives. Finally, further attempts at effective ensembles of state-of-the-art algorithms such as SSC could yield improved empirical performance.

6.7 Proofs of Theoretical Results

The results of this section make use of the following notation. We define the absolute inner product between points $x_i \in \mathcal{S}_l$ and $x_j \in \mathcal{S}_k$ as

$$z_{i,j}^{(l,k)} = \left| \left\langle x_i^{(l)}, x_j^{(k)} \right\rangle \right|,$$

where k may be equal to l. We denote the qth largest absolute inner product between $x_i^{(l)}$ and other points in the subspaces $\mathcal{S}_l$ as $z_{(i,q)}^{(l)}$, i.e., we have

$$z_{(i,q)}^{(l)} = \left| \left\langle x_i^{(l)}, x_{\neq i}^{(l)} \right\rangle \right|_{[q]}$$

in the context of Definition 6.3.

6.7.1 Proof of Theorem 6.4

We first prove the statement for a fixed $x_i \in \mathcal{S}_l$. The statement of the theorem can be written as

$$\hat{f}^{(l)}_{(i,q)} > \max_{k \neq l, j} \hat{f}^{(l,k)}_{i,j}, \tag{6.16}$$

where $\hat{f}^{(l)}_{(i,q)}$ denotes the qth largest value in the set $\{ \hat{f}^{(l,l)}_{i,j} \}$. We first bound $\hat{f}$ in terms of f. Let $x_\iota \in \mathcal{S}_{k^*}$ be such that $\max_{k \neq l, j} \hat{f}^{(l,k)}_{i,j} = \hat{f}^{(l,k^*)}_{i,\iota}$ and note that $z^{(l,k^*)}_{i,\iota} \leq \max_{k \neq l, j} z^{(l,k)}_{i,j}$. Then we have

$$\begin{aligned} \max_{k \neq l, j} \hat{f}^{(l,k)}_{i,j} = \hat{f}^{(l,k^*)}_{i,\iota} &\leq f\left(z^{(l,k^*)}_{i,\iota}\right) + \tau \\ &\leq f\left(\max_{k \neq l, j} z^{(l,k)}_{i,j}\right) + \tau, \end{aligned}$$

where the second line follows by monotonicity of f. To lower bound $\hat{f}^{(l)}_{(i,q)}$, let x_κ be such that $\hat{f}^{(l)}_{(i,q)} = \hat{f}^{(l,l)}_{i,\kappa}$. If $z^{(l,l)}_{i,\kappa} \geq z^{(l)}_{(i,q)}$, then $f\left(z^{(l,l)}_{i,\kappa}\right) \geq f\left(z^{(l)}_{(i,q)}\right)$ by monotonicity of f. For the case where $z^{(l,l)}_{i,\kappa} < z^{(l)}_{(i,q)}$, define $x_\lambda \in \mathcal{S}_l$ such that $z^{(l)}_{(i,q)} = z^{(l,l)}_{i,\lambda}$ and note that

$$\hat{f}^{(l,l)}_{i,\kappa} > \hat{f}^{(l,l)}_{i,\lambda} \geq f\left(z^{(l,l)}_{i,\lambda}\right) - \tau = f\left(z^{(l)}_{(i,q)}\right) - \tau.$$

Therefore

$$\hat{f}^{(l)}_{(i,q)} \geq f\left(z^{(l)}_{(i,q)}\right) - \tau,$$

and (6.16) holds as long as

$$f\left(z^{(l)}_{(i,q)}\right) - \tau > f\left(\max_{k \neq l, j} z^{(l,k)}_{i,j}\right) + \tau,$$

or equivalently if

$$\tau < \frac{f\left(z^{(l)}_{(i,q)}\right) - f\left(\max_{k \neq l, j} z^{(l,k)}_{i,j}\right)}{2}. \tag{6.17}$$

Taking the minimum right-hand side of (6.17) among all $x \in \mathcal{X}$ completes the proof.

6.7.2 Proof of Theorem 6.5

To prove Theorem 6.5, we first prove a slightly more general result that we will then apply.

Lemma 6.14. *Let $a_1, \ldots, a_n \in \mathbb{R}^d$ be i.i.d. uniform on $\mathbb{S}^{d-1}$ and let $\tilde{G}$ be the corresponding q-nearest neighbor graph with respect to the (transformed and noisy) inner products*

$$\hat{f}_{ij} = f(|\langle a_i, a_j \rangle|) + \tau_{ij}, \quad i, j \in 1, \ldots, n \tag{6.18}$$

where $f : \mathbb{R}_+ \to \mathbb{R}_+$ is a strictly increasing function and $\tau_{ij} \in [-\tau, \tau]$ are bounded measurement errors. Let $\delta \geq 0$ and $\gamma \in (1, n/\log n)$ be arbitrary, and let θ be the spherical radius of a spherical cap covering $\gamma \log n/n$ fraction of the area of $\mathbb{S}^{d-1}$. Then if $q \in [3(24\pi)^{d-1}\gamma \log n + 3\frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{n}{d-1}(2\delta)^{d-1}, n]$, $\theta \leq (\pi/2 - \delta)/24$ and $\tau \leq \{f(\cos(16\theta)) - f(\cos(16\theta + \delta))\}/2$, we have

$$\mathbb{P}\{\tilde{G} \text{ is connected }\} \geq 1 - \frac{2}{n^{\gamma-1}\gamma \log n}, \tag{6.19}$$

where $\mathcal{L}$ denotes the Lebesgue measure of its argument.

Proof of Lemma 6.14. Following the approach taken in [91, Appendix A.B], we partition the unit sphere $\mathbb{S}^{d-1}$ into $M := n/(\gamma \log n)$ non-overlapping regions $R_1, \ldots, R_M$ of equal area with spherical diameters upper bounded as

$$\sup_{x,y \in R_m} \arccos(\langle x, y \rangle) \leq 8\theta =: \theta^*$$

for all m; the existence of such a partition was shown in [126, Lemma 6.2]. Consider the events

$$A_m := R_m \text{ contains at least one of } a_1, \ldots, a_n$$
$$B_m := \text{Fewer than } q/2 \text{ samples are within } 3\theta^* + \delta \text{ of } c_m \text{ in spherical distance}$$

where $c_1, \ldots, c_M$ are arbitrarily chosen points in $R_1, \ldots, R_M$, respectively, and the spherical distance between two points x and y is $\arccos(\langle x, y \rangle)$. The proof proceeds as in [91, Appendix A.B] by first showing that $\tilde{G}$ is connected if A_m and B_m hold for all $m = 1, \ldots, M$. It then follows that

$$\mathbb{P}\{\tilde{G} \text{ is connected}\} \geq \mathbb{P}\{\forall m\ A_m \wedge B_m\} \geq 1 - \sum_{m=1}^{M} \mathbb{P}\{\neg A_m\} - \sum_{m=1}^{M} \mathbb{P}\{\neg B_m\} \tag{6.20}$$

where $\wedge$ is conjunction, $\neg$ is negation, and the second inequality follows from a union bound. The proof concludes by upper bounding $\mathbb{P}\{\neg A_m\}$ and $\mathbb{P}\{\neg B_m\}$; substituting the bounds into (6.20) yields the final result (6.19).

Implication. We show that $\tilde{G}$ is connected if A_m and B_m hold for all $m = 1, \ldots, M$, by showing that all samples in neighboring regions are connected when B_m holds for all m. Since each region contains at least one sample when A_m holds for all m, it then follows that any pair of samples is connected via a chain of connections through neighboring regions and so $\tilde{G}$ is connected.

Let a_i and a_ℓ be arbitrary samples in neighboring regions R_m and R_n. Then a_ℓ is within $2\theta^*$ of a_i in spherical distance and thus $\hat{f}_{i\ell} \geq \tilde{f}(2\theta^*) - \tau$, where we define

$\tilde{f}(\alpha) = f(\cos(\alpha))$ for convenience and note that it is decreasing on $[0, \pi/2]$. Any sample a_j for which $\hat{f}_{ij} \geq \tilde{f}(2\theta^*) - \tau$ must satisfy

$$\begin{aligned}\tilde{f}(\arccos|\langle a_i, a_j\rangle|) &= \hat{f}_{ij} - \tau_{ij} \geq \hat{f}_{ij} - \tau \\ &\geq \tilde{f}(2\theta^*) - 2\tau = \tilde{f}(16\theta) - 2\tau \\ &\geq \tilde{f}(16\theta + \delta) = \tilde{f}(2\theta^* + \delta)\end{aligned} \tag{6.21}$$

and so must also satisfy $\arccos|\langle a_i, a_j\rangle| \leq 2\theta^* + \delta$ because $\tilde{f}$ is decreasing. Namely, any such sample must be within $2\theta^* + \delta$ of either a_i or $-a_i$, and must hence be within $3\theta^* + \delta$ of either c_m or $c_{m'}$ where $R_{m'}$ is the region containing $-a_i$. Under B_m and $B_{m'}$, there are fewer than q such samples and so all must be connected to a_i. In particular, a_ℓ must be connected to a_i, and all samples in neighboring regions are connected when B_m holds for all m.

Upper bound on $\mathbb{P}\{\neg A_m\}$. As in [91, Equations (27)–(28)], we use the fact that each sample falls outside of R_m with probability $1 - 1/M$ since the samples are drawn uniformly from $\mathbb{S}^{d-1}$ and the M regions have equal area. The samples are furthermore drawn independently, and so

$$\mathbb{P}\{\neg A_m\} = \left(1 - \frac{1}{M}\right)^n \leq e^{-n/M} = \frac{1}{M}\frac{1}{n^{\gamma-1}\gamma\log n}. \tag{6.22}$$

Upper bound on $\mathbb{P}\{\neg B_m\}$. For convenience let $\mathcal{C}_m := \{x : \arccos(\langle x, c_m\rangle) \leq 3\theta^* + \delta\}$ denote the spherical cap of spherical radius $3\theta^* + \delta$ around c_m, and let N_m denote the number of samples in $\mathcal{C}_m$. In this notation, B_m is the event that $N_m \leq q/2$. As in [91, Appendix A.B], we note that N_m is a binomially distributed random variable with n trials and probability $p := \mathcal{L}(\mathcal{C}_m)/\mathcal{L}(\mathbb{S}^{d-1})$, where $\mathcal{L}$ is the area (Lebesgue measure) of a set.

We begin by bounding $q/2$ below by $3np$; this will make applying a binomial tail bound more convenient. By assumption, $3\theta^* + \delta = 24\theta + \delta \leq \pi/2$ and so we can apply [126, Equation (5.2)] as in [91] to bound p as

$$p := \frac{\mathcal{L}(\mathcal{C}_m)}{\mathcal{L}(\mathbb{S}^{d-1})} \leq \frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{(3\theta^* + \delta)^{d-1}}{d-1} \leq \frac{1}{2}\left(\frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{(6\theta^*)^{d-1}}{d-1} + \frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{(2\delta)^{d-1}}{d-1}\right) \tag{6.23}$$

where the second inequality follows from the convexity of x^{d-1} (when $x > 0$) applied to the convex combination $x = 3\theta^* + \delta = 1/2(6\theta^*) + 1/2(2\delta)$. The first term can be further bounded since

$$\theta^* \leq 4\pi\left((d-1)\frac{\mathcal{L}(\mathbb{S}^{d-1})}{\mathcal{L}(\mathbb{S}^{d-2})}\frac{\gamma\log n}{n}\right)^{1/(d-1)} \tag{6.24}$$

as in [91, Equation (31)]; the proof is the same with $3(24\pi)^{d-1}$ in place of $6(12\pi)^{d-1}$. Substituting into (6.23) yields

$$p \leq \frac{1}{2}\left((24\pi)^{d-1}\frac{\gamma \log n}{n} + \frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{(2\delta)^{d-1}}{d-1}\right) \tag{6.25}$$

and thus

$$3np \leq \frac{1}{2}\left(3(24\pi)^{d-1}\gamma \log n + 3\frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{n}{d-1}(2\delta)^{d-1}\right) \leq \frac{q}{2}. \tag{6.26}$$

Applying the binomial tail bound [109, Theorem 1] as done in [91, Equation (29)] now yields

(6.27)

$$\mathbb{P}\{\neg B_m\} = \mathbb{P}\{N_m > q/2\} \leq \mathbb{P}\{N_m > 3np\} \leq e^{-np} \leq e^{-n/M} = \frac{1}{M}\frac{1}{n^{\gamma-1}\gamma \log n}.$$

The last inequality holds since $R_m \subset \mathcal{C}_m$ and so $p = \mathcal{L}(\mathcal{C}_m)/\mathcal{L}(\mathbb{S}^{d-1}) \geq \mathcal{L}(R_m)/\mathcal{L}(\mathbb{S}^{d-1}) = 1/M$. □

Remark 6.15. An alternative bound on $(\alpha+\beta)^{d-1}$ could have been used in the proof of Lemma 6.14 to shift the constants more heavily on the δ term. For example,

$$(\alpha+\beta)^{d-1} \leq \lambda\left(\frac{\alpha}{\lambda}\right)^{d-1} + (1-\lambda)\left(\frac{\beta}{1-\lambda}\right)^{d-1} \tag{6.28}$$

for any $\lambda \in (0,1)$ and taking $\lambda \approx 1$ shifts the constants heavily onto the second term. The proof of Lemma 6.14 uses $\lambda = 1/2$.

We are now prepared to prove Theorem 6.5 by applying Lemma 6.14 with a particular choice of δ.

Proof of Theorem 6.5. Take

$$C_3 = \frac{f(\cos(16\theta)) - f(\cos(16\theta+\delta))}{2} > 0, \tag{6.29}$$

where we note that θ is implicitly a function of n, d and γ, and we define

$$\delta = \min\left\{12\pi\left(\frac{d-1}{3}\frac{\mathcal{L}(\mathbb{S}^{d-1})}{\mathcal{L}(\mathbb{S}^{d-2})}\frac{\gamma \log n}{n}\right)^{1/(d-1)}, \frac{\pi}{2} - 24\theta\right\} > 0, \tag{6.30}$$

which is also implicitly a function of n, d and γ. Now we need only to verify that the conditions of Theorem 6.5 satisfy Lemma 6.14. Note first that by construction $\delta \leq \pi/2 - 24\theta$ and so $\theta \leq (\pi/2-\delta)/24$. Furthermore

$$3\frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{n}{d-1}(2\delta)^{d-1} \leq (24\pi)^{d-1}\gamma \log n \tag{6.31}$$

and so

$$q \geq 4(24\pi)^{d-1}\gamma \log n = 3(24\pi)^{d-1}\gamma \log n + (24\pi)^{d-1}\gamma \log n \tag{6.32}$$

$$\geq 3(24\pi)^{d-1}\gamma \log n + 3\frac{\mathcal{L}(\mathbb{S}^{d-2})}{\mathcal{L}(\mathbb{S}^{d-1})}\frac{n}{d-1}(2\delta)^{d-1}. \tag{6.33}$$

Hence all conditions of Lemma 6.14 are satisfied and the conclusion follows. □

6.7.3 Proof of Lemma 6.6

We again prove the statement for a fixed $x_i \in \mathcal{S}_l$, taking a union bound to show the condition holds for all points. First define

$$\alpha = \min_{l,\mathcal{D}:|\mathcal{D}|\leq 2s,\|a\|=1} \left\| {\mathbf{U}_{\mathcal{D}}^{(l)}}^\top \mathbf{U}^{(l)} a \right\|_2,$$

and note that by the assumption of the lemma, there exists an $\eta > 0$ such that

$$\max_{k,l:k\neq l,\mathcal{D}:|\mathcal{D}|\leq 2s} \left\| {\mathbf{U}_{\mathcal{D}}^{(k)}}^\top \mathbf{U}^{(l)} \right\|_2 = \alpha - \eta. \tag{6.34}$$

Equation (6.34) implies that

$$\max_{k\neq l,j} z_{i,j}^{(l,k)} \leq \alpha - \eta \tag{6.35}$$

deterministically. Next, we show that

$$z_{(i,q)}^{(l)} \geq \alpha - \frac{\eta}{2} \tag{6.36}$$

with high probability. The proof is nearly identical to [91, Lemma 1]. First, we have that

$$\begin{aligned} z_{i,j}^{(l,l)} &\sim \left\| {\mathbf{U}_{\mathcal{D}}^{(l)}}^\top \mathbf{U}_{\mathcal{E}}^{(l)} a_i^{(l)} \right\|_2 \left| \left\langle a_i^{(l)}, a_j^{(l)} \right\rangle \right| \\ &\geq \min_{l,\mathcal{D}:|\mathcal{D}|\leq 2s,\|a\|=1} \left\| {\mathbf{U}_{\mathcal{D}}^{(l)}}^\top \mathbf{U}^{(l)} a \right\|_2 \left| \left\langle a_i^{(l)}, a_j^{(l)} \right\rangle \right|, \end{aligned}$$

where the sets $\mathcal{D}, \mathcal{E} \subset [D]$ are the indices of the unobserved entries of $x_j^{(l)}$ and $x_i^{(l)}$, respectively. Letting $\tilde{z}_{i,j}^{(l,l)} = \left| \left\langle a_i^{(l)}, a_j^{(l)} \right\rangle \right|$, we see that

$$\begin{aligned} \mathbb{P}\{ z_{i,j}^{(l,l)} \leq z \} &\leq \mathbb{P}\{ \min_{l,\mathcal{D}:|\mathcal{D}|\leq 2s,\|a\|=1} \left\| {\mathbf{U}_{\mathcal{D}}^{(l)}}^\top \mathbf{U}^{(l)} a \right\|_2 \tilde{z}_{i,j}^{(l,l)} \leq z \} \\ &= \mathbb{P}\{ \tilde{z}_{i,j}^{(l,l)} \leq \frac{z}{\alpha} \}. \end{aligned}$$

We can bound the probability that (6.36) does not hold as

$$\begin{aligned} \mathbb{P}\{ z_{(i,q)}^{(l)} \leq \alpha - \frac{\eta}{2} \} &\leq \mathbb{P}\{ \tilde{z}_{(i,q)}^{(l)} \leq 1 - \frac{\eta}{2\alpha} \} \\ &\leq \left(e\frac{N_l - 1}{q-1} \right)^{q-1} p^{N_l - q}, \end{aligned}$$

where $p = \mathbb{P}\{\, \tilde{z}_{i,j}^{(l,l)} \leq 1 - \frac{\eta}{2\alpha} \,\}$. Setting $\xi = \frac{N_l - 1}{N_l^\rho - 1}$, we obtain

$$\begin{aligned}
\mathbb{P}\{\, z_j^{(l)} \leq 1 - \frac{\eta}{2\alpha} \,\} &\leq (e\xi)^{\frac{N_l-1}{\xi}} \, p^{(N_l-1)\left(1-\frac{1}{\xi}\right)} \\
&= \left((e\xi)^{\frac{1}{\xi}} \, p^{1-\frac{1}{\xi}} \right)^{N_l - 1} \\
&\leq e^{-(N_l-1)c_1},
\end{aligned}$$

where the last inequality holds for a constant $c_1 > 0$ as long as

$$(e\xi)^{\frac{1}{\xi}} \, p^{1-\frac{1}{\xi}} < 1 \Leftrightarrow (e\xi)^{-\frac{1}{\xi - 1}} > p.$$

This inequality can be satisfied for every $p < 1$ by taking N_0, and consequently ξ, sufficiently large. By inspection, we have $p < 1$ as long as $\eta > 0$, which is true by assumption of the lemma.

By monotonicity of f, (6.35) implies that

$$f\left(\max_{k \neq l, j} z_{i,j}^{(l,k)} \right) \leq f\left(\alpha - \eta\right)$$

and (6.36) implies that

$$f\left(z_{(i,q)}^{(l)} \right) \geq f\left(\alpha - \frac{\eta}{2} \right).$$

Finally, we have that

$$\begin{aligned}
C_{i,l} &:= f\left(z_{(i,q)}^{(l)} \right) - f\left(\max_{k \neq l, j} z_{i,j}^{(l,k)} \right) \\
&\geq f\left(\alpha - \frac{\eta}{2} \right) - f\left(\alpha - \eta\right) > 0,
\end{aligned}$$

where the second line follows by monotonicity of f, noting that $\alpha - \eta/2 > \alpha - \eta$. Taking $C_1 = \min_{l \in [K], i \in [N_l]} C_{i,l}/2$ and a union bound completes the proof.

6.7.4 Proof of Lemma 6.7

We again prove the statement for a fixed $x_i \in \mathcal{S}_l$, with a union bound completing the proof. Let $\nu = 2/3$, $N_l \geq 6q$, and $c_2 > 1/20$. From [91, Appendix C], we have that

$$z_{(i,q)}^{(l)} \geq \frac{\nu}{\sqrt{d_l}} - \varepsilon \tag{6.37}$$

and

$$\max_{k \neq l, j} z_j^{(k)} \leq \alpha + \varepsilon \tag{6.38}$$

with probability at least $1 - e^{-c_2(N_l-1)} - 10Ne^{-\beta^2/2}$, where

$$\alpha = \frac{\beta(1+\beta)}{\sqrt{d_l}} \max_{k \neq l} \frac{1}{\sqrt{d_k}} \left\| \mathbf{U}^{(k)^\top} \mathbf{U}^{(l)} \right\|_F,$$

$$\varepsilon = \frac{2\sigma(1+\sigma)}{\sqrt{D}} \beta$$

and $\frac{1}{\sqrt{2\pi}} \leq \beta \leq \sqrt{D}$. Let $\beta = \sqrt{6 \log N}$ and note that $D \geq 6 \log N$ implies $\beta \leq \sqrt{D}$. Noting that $q < N_{min}/6$ implies $N > 6$, we have $(1+\beta) < 4\sqrt{\log N}$. These are sufficient to guarantee that $\alpha + \varepsilon < \frac{\nu}{\sqrt{d_l}} - \varepsilon$. By monotonicity of f, (6.38) implies that

$$f\left(\max_{k \neq l, j} z_{i,j}^{(l,k)} \right) \leq f\left(\alpha + \varepsilon\right)$$

and (6.37) implies that

$$f\left(z_{(i,q)}^{(l)} \right) \geq f\left(\frac{\nu}{\sqrt{d_l}} - \varepsilon \right).$$

Finally, we have that

$$\begin{aligned} C_{i,l} &:= f\left(z_{(i,q)}^{(l)} \right) - f\left(\max_{k \neq l, j} z_{i,j}^{(l,k)} \right) \\ &\geq f\left(\frac{\nu}{\sqrt{d_l}} - \varepsilon \right) - f\left(\alpha + \varepsilon\right) > 0, \end{aligned}$$

where the second line follows by monotonicity of f. Taking $C_2 = \min_{l \in [K], i \in [N_l]} C_{i,l}/2$ and a union bound completes the proof.

6.7.5 Proof of Theorem 6.8

By Lemma 6.9, the expected entries of the co-association matrix obtained by EKSS-0 are an increasing function of the inner product between points. It remains to show how tightly these values concentrate around their mean. This concentration allows us to bound the noise level τ via the following lemma.

Lemma 6.16. *Let* $\mathbf{A}$ *be the affinity matrix formed by EKSS-0 (line 12, Alg. 6.1). For two points* $x_i, x_j \in \mathcal{X}$*, let*

$$f_{\bar{K},\bar{d}}\left(|\langle x_i, x_j \rangle|\right) = \mathbb{E}\mathbf{A}_{i,j} = \mathbb{P}\left\{ x_i, x_j \text{ co-clustered} \right\}$$

and

$$\hat{f}_{i,j} = \mathbf{A}_{i,j} = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\left\{ x_i, x_j \text{ co-clustered in } \mathcal{C}^{(b)} \right\}.$$

Then for all $\tau > 0$

$$\mathbb{P}\left\{ \left| \hat{f}_{i,j} - f_{\bar{K},\bar{d}}\left(|\langle x_i, x_j \rangle|\right) \right| > \tau \right\} < 2e^{-c_3 \tau^2 B}, \tag{6.39}$$

where $c_3 = 2\sqrt{\log 2}$ and the randomness is with respect to the subspaces drawn in EKSS-0 (line 4, Alg. 6.1).

Proof. The proof relies on sub-Gaussian concentration. The measurements $\hat{f}$ are bounded and hence sub-Gaussian with parameter $\frac{1}{\sqrt{\log 2}}$. Note that $\hat{f}_{i,j}$ is the empirical estimate of $f_{\bar{K},\bar{d}}(|\langle x_i, x_j\rangle|)$, and thus $\mathbb{E}\hat{f}_{i,j} = f_{\bar{K},\bar{d}}(|\langle x_i, x_j\rangle|)$. Therefore, by the General form of Hoeffding's inequality [204, Theorem 2.6.2]

$$\mathbb{P}\left\{ \left| \hat{f}_{i,j} - \mathbb{E}\hat{f}_{i,j} \right| > \tau \right\} \leq 2e^{-c_3\tau^2 B},$$

where $c_3 = 2\sqrt{\log 2}$. □

Combining the results of Theorem 6.9 and Lemma 6.16 shows that the (i,j)th entry of the affinity matrix is τ-angle preserving with high probability for a single point. A union bound over all $N(N-1)/2$ unique pairs completes the proof.

6.7.6 Proof of Lemma 6.9

For notational compactness, we instead prove that the probability is a *decreasing* function of the angle θ between points and note that $z = \cos(\theta)$. Let $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_K \in \mathbb{R}^{D\times d}$ be the K candidate bases. Let $\tilde{p}(\theta)$ be the probability that two points that are at angle θ apart are assigned to the candidate $\mathbf{U}_1$. Then we clearly have $p_{K,D}(\theta) = K\tilde{p}(\theta)$, and it suffices to prove that $\tilde{p}$ is strictly decreasing.

Let $e_1, \ldots, e_D$ be the standard basis vectors in $\mathbb{R}^D$. For a given θ, set $x_i := e_1$, and $x_j = x_j(\theta) := \cos(\theta)e_1 + \sin(\theta)e_2$. By definition, for any orthogonal transformation $\mathbf{Q}$ of $\mathbb{R}^D$,

$$\tilde{p}(\theta) = \mathbb{P}\left\{ \mathbf{Q}x_i, \mathbf{Q}x_j \text{ both assigned to } \mathbf{U}_1 \right\}.$$

We may average out this equation over a choice subgroup of orthogonal matrices. Indeed, let $\mathcal{L}$ denote the span of e_1 and e_2, and let $\mathbf{Q}$ be a random matrix uniformly distributed over the set of orthogonal matrices that decompose into a rotation on $\mathcal{L}$ and the identity on $\mathcal{L}^\perp$. We take expectations with respect to $\mathbf{Q}$ and exchange the order of integration to get

$$\begin{aligned}\tilde{p}(\theta) &= \mathbb{E}_{\mathbf{Q}}\mathbb{P}_{\mathbf{U}_1,\ldots,\mathbf{U}_K}\left\{ \mathbf{Q}x_i, \mathbf{Q}x_j \text{ both assigned to } \mathbf{U}_1 \right\} \\ &= \mathbb{E}_{\mathbf{U}_1,\ldots,\mathbf{U}_K}\mathbb{P}\left\{ \mathbf{Q}x_i, \mathbf{Q}x_j \text{ both assigned to } \mathbf{U}_1 \;\middle|\; \mathbf{U}_1, \ldots, \mathbf{U}_K \right\}.\end{aligned}$$

Now fix $\mathbf{U}_1, \ldots, \mathbf{U}_K$. Let $A = A(\theta)$ be the event that $\mathbf{Q}x_i$ and $\mathbf{Q}x_j(\theta)$ are both assigned to $\mathbf{U}_1$. We claim that $\mathbb{P}\{ A(\theta) \mid \mathbf{U}_1, \ldots, \mathbf{U}_K \}$ is non-increasing in θ. To see this, let us examine the event more closely. By the definition of candidate

assignment, A occurs when $\mathbf{U}_1$ is the *closest* candidate to both x_i and x_j. More mathematically, this is when

$$\|\mathbf{P}_{\mathbf{U}_1}\mathbf{Q}z\|_2^2 > \|\mathbf{P}_{\mathbf{U}_k}\mathbf{Q}z\|_2^2, \quad \text{for} \quad 1 < k \leq K, \quad \text{and} \quad z = x_i, x_j. \tag{6.40}$$

Here, we use $\mathbf{P}_{\mathcal{F}}$ to denote the orthogonal projection onto a subspace $\mathcal{F}$.

We shall attempt to rewrite (6.40) in a more useful form. First, observe that

$$\begin{aligned} \|\mathbf{P}_{\mathbf{U}_1}\mathbf{Q}z\|_2^2 - \|\mathbf{P}_{\mathbf{U}_k}\mathbf{Q}z\|_2^2 &= z^\top \mathbf{Q}^\top \mathbf{P}_{\mathbf{U}_1}^\top \mathbf{P}_{\mathbf{U}_1} z - z^\top \mathbf{P}_{\mathbf{U}_k}^\top \mathbf{P}_{\mathbf{U}_k}\mathbf{Q}z \\ &= z^\top \mathbf{Q}^\top \mathbf{P}_{\mathcal{L}} \left(\mathbf{P}_{\mathbf{U}_1}^\top \mathbf{P}_{\mathbf{U}_1} - \mathbf{P}_{\mathbf{U}_k}^\top \mathbf{P}_{\mathbf{U}_k}\right) \mathbf{P}_{\mathcal{L}}^\top \mathbf{Q}z. \end{aligned} \tag{6.41}$$

Let us also introduce some new notation. We use $\tilde{x}_i$ and $\tilde{x}_j$ to denote the two-dimensional coordinate vectors of x_i and x_j with respect to e_1 and e_2, we let $\tilde{\mathbf{Q}}$ denote the restriction of $\mathbf{Q}$ to $\mathcal{L}$, and similarly let $\tilde{\mathbf{P}}_{\mathcal{L}}$ be the projection $\mathbf{P}_{\mathcal{L}}$ treated as a map from $\mathbb{R}^D$ to $\mathbb{R}^2$. We therefore have

$$z^\top \mathbf{Q}^\top \mathbf{P}_{\mathcal{L}} \left(\mathbf{P}_{\mathbf{U}_1}^\top \mathbf{P}_{\mathbf{U}_1} - \mathbf{P}_{\mathbf{U}_k}^\top \mathbf{P}_{\mathbf{U}_k}\right) \mathbf{P}_{\mathcal{L}}^\top \mathbf{Q}z = \tilde{z}^\top \tilde{\mathbf{Q}}^\top \mathbf{M}_k \tilde{\mathbf{Q}}\tilde{z},$$

where $\mathbf{M}_k := \tilde{\mathbf{P}}_{\mathcal{L}} \left(\mathbf{P}_{\mathbf{U}_1}^\top \mathbf{P}_{\mathbf{U}_1} - \mathbf{P}_{\mathbf{U}_k}^\top \mathbf{P}_{\mathbf{U}_k}\right) \tilde{\mathbf{P}}_{\mathcal{L}}^\top$. Following these calculations, we see that (6.40) is equivalent to

$$\tilde{z}^\top \tilde{\mathbf{Q}}^\top \mathbf{M}_k \tilde{\mathbf{Q}}\tilde{z} > 0, \quad \text{for} \quad 1 < k \leq K, \quad \text{and} \quad \tilde{z} = \tilde{x}_i, \tilde{x}_j. \tag{6.42}$$

When $\tilde{\mathbf{Q}}$ is fixed, denote by $A_{\tilde{\mathbf{Q}}}$ the event over which (6.42) holds.

Observe that $\mathbf{M}_k$ is a 2 by 2 real symmetric matrix. As such, the set S_k of points $\tilde{z}$ in $\mathbb{R}^2$ for which $\tilde{z}^T \mathbf{M}_k \tilde{z} > 0$ comprises the union of two (possibly degenerate) antipodal *sectors*. The same is true for the intersection $S := \cap_{k>1} S_k$. Let $\phi = \phi(\mathbf{U}_1, \ldots, \mathbf{U}_K)$ denote the angle spanned by one of the two sectors comprising S, and note that $0 \leq \phi \leq \pi$. Furthermore, let T be the union of the sector spanned by $\tilde{x}_i$ and $\tilde{x}_j$ with its antipodal reflection. Then $A_{\tilde{\mathbf{Q}}}$ holds if and only if $\tilde{\mathbf{Q}}T \subset S$ or $S^c \subset \tilde{\mathbf{Q}}T$. It is a simple exercise to compute

$$\mathbb{P}\left\{ \tilde{\mathbf{Q}}T \subset S \;\middle|\; \mathbf{U}_1, \ldots, \mathbf{U}_K \right\} = \frac{(\phi - \theta)_+}{\pi},$$

$$\mathbb{P}\left\{ S^c \subset \tilde{\mathbf{Q}}T \;\middle|\; \mathbf{U}_1, \ldots, \mathbf{U}_K \right\} = \frac{(\theta - \pi + \phi)_+}{\pi}.$$

Since A is the disjoint union of these events, we have

$$\mathbb{P}\left\{ A(\theta) \;\middle|\; \mathbf{U}_1, \ldots, \mathbf{U}_K \right\} = \frac{(\phi - \theta)_+}{\pi} + \frac{(\theta - \pi + \phi)_+}{\pi}. \tag{6.43}$$

Differentiating at any point other than the obvious discontinuities, we have

$$\begin{aligned}\frac{d}{d\theta}\mathbb{P}\{\, A(\theta) \;\mid\; \mathbf{U}_1, \ldots, \mathbf{U}_K \,\} &= \frac{d}{d\theta}\frac{(\phi-\theta)_+}{\pi} + \frac{(\theta-\pi+\phi)_+}{\pi} \\ &= -\frac{1}{\pi}1_{(0,\phi)}(\theta) + \frac{1}{\pi}1_{(\pi-\phi,\pi/2)}(\theta) \\ &= -\frac{1}{\pi} + \frac{1}{\pi}1_{(\phi,\pi/2)}(\theta) + \frac{1}{\pi}1_{(\pi-\phi,\pi/2)}(\theta) \\ &\leq 0.\end{aligned}$$

Here, the last inequality follows from the fact that either $\phi \geq \pi/2$ or $\pi - \phi > \pi/2$, thereby completing the proof of the claim. Recalling that $\tilde{p}(\theta) = \mathbb{E}_{\mathbf{U}_1,\ldots,\mathbf{U}_K}\mathbb{P}\{\, A(\theta) \;\mid\; \mathbf{U}_1, \ldots, \mathbf{U}_K \,\}$, we have thus proved that $\tilde{p}$ is non-increasing. To see that it is strictly decreasing, simply note that $\frac{d}{d\theta}\mathbb{P}\{\, A(\theta) \;\mid\; \mathbf{U}_1, \ldots, \mathbf{U}_K \,\} < 0$ whenever $\phi(\mathbf{U}_1, \ldots, \mathbf{U}_K) < \pi/2$. This occurs on a set of positive measure.

6.7.7 Proof of Theorem 6.10

By Theorem 6.8, the co-association matrix $\bar{\mathbf{A}}$ is τ-angle preserving with high probability. Applying Lemma 6.6 with $s = 0$, we obtain $C_1 > 0$ that lower bounds the separation ϕ_q defined in (6.6) with high probability. Applying Lemma 6.5 with $\gamma = 3$, we obtain $C_3 > 0$ such that the components corresponding to each subspace are connected with high probability. Setting $\tau = \min\{\, C_1, C_3 \,\}$ in Theorem 6.8 completes the proof.

6.7.8 Proof of Theorem 6.11

By Theorem 6.8, the co-association matrix $\bar{\mathbf{A}}$ is τ-angle preserving with high probability. Applying Lemma 6.7 with $\sigma = 0$, we obtain $C_2 > 0$ that lower bounds the separation ϕ_q defined in (6.6) with high probability. Applying Lemma 6.5 with $\gamma = 3$, we obtain $C_3 > 0$ such that the components corresponding to each subspace are connected with high probability. Setting $\tau = \min\{\, C_1, C_3 \,\}$ in Theorem 6.8 completes the proof.

6.7.9 Proof of Theorem 6.12

By Theorem 6.8, the co-association matrix $\bar{\mathbf{A}}$ is τ-angle preserving with high probability. Applying Lemma 6.7, we obtain $C_2 > 0$ that lower bounds the separation ϕ_q defined in (6.6) with high probability. Setting $\tau = \min\{\, C_1, C_3 \,\}$ in Theorem 6.8 completes the proof.

6.7.10 Proof of Theorem 6.13

By Theorem 6.8, the co-association matrix $\bar{\mathbf{A}}$ is τ-angle preserving with high probability. By [91, Lemma 4], the condition (6.8) holds with probability at least $1 - 4e^{-c_7 D}$ as long as (6.15) is satisfied. Thus, applying Lemma 6.6 with the parameters $N_k = n$, $d_k = d$ for all k, the result holds with the specified probability.

6.8 Implementation Details

This section includes implementation details beyond those included in the main body. We define the clustering error precisely and describe the preprocessing steps and parameters used for our experiments on real data.

6.8.1 Clustering Error

The clustering error, which is the metric used for all experimental results, is computed by matching the true labels and the labels output by a given clustering algorithm,

$$\text{err} = \frac{100}{N}\left(1 - \max_{\pi} \sum_{i,j} Q^{\text{out}}_{\pi(i)j} Q^{\text{true}}_{ij}\right),$$

where π is a permutation of the cluster labels, and Q^{out} and Q^{true} are the output and ground-truth labelings of the data, respectively, where the (i, j)th entry is one if point j belongs to cluster i and is zero otherwise.

6.8.2 Experiments on Benchmark Data

Dataset	N	K	D
Hopkins-155	39-556	2-3	30-200
Yale	2432	38	2016
COIL-20	1440	20	1024
COIL-100	7200	100	1024
USPS	9298	10	256
MNIST-10k	10000	10	500

Table 6.2: Datasets used for experiments with relevant parameters; N: total number of samples, K: number of clusters, D: ambient dimension.

This section describes the benchmark datasets used in our experiments, as well as any preprocessing steps and the parameters selected for all algorithms. All datasets are normalized so that each column lies on the unit sphere in the corresponding

ambient dimension, as is common in the literature [89, 91, 187]. Table 6.2 gives a summary of all datasets considered.

The Hopkins-155 dataset [198] consists of 155 motion sequences with $K = 2$ in 120 of sequences and $K = 3$ in the remaining 35. In each sequence, objects moving along different trajectories each lie near their own affine subspace of dimension at most 3. We perform no preprocessing steps on this dataset.

The Extended Yale Face Database B [74, 123] consists of 64 images of each of 38 different subjects under a variety of lighting conditions. Each image is of nominal size 192×168 and is known to lie near a 9-dimensional subspace [20]. We downsample so that each image is of size 48×42, as in [67]. For EKSS, KSS, CoP-KSS, MKF, and TSC, we perform an initial whitening as in [91, 226] by removing the first two singular components of the dataset and then project the data onto its first 500 principal components to reduce the computational complexity of these methods. Whitening resulted in worse performance for all other algorithms, so we omitted this step.

Algorithm	Hopkins	Yale	COIL-20	COIL-100	USPS	MNIST-10k
EKSS	$d = 3, q = 2$	$d = 2, q = 6$	$d = 2, q = 6$	$d = 8, q = 7$	$d = 13, q = 3$	$d = 13, q = 72$
KSS	$d = 3$	$d = 3$	$d = 1$	$d = 5$	$d = 9$	$d = 13$
CoP-KSS	$d = 4$	$d = 6$	$d = 9$	$d = 1$	$d = 7$	$d = 18$
MKF	$d = 3$	$d = 17$	$d = 19$	$d = 18$	$d = 20$	$d = 20$
TSC	$q = 3$	$q = 3$	$q = 4$	$q = 4$	$q = 3$	$q = 3$
SSC-ADMM	$\rho = 0.1, \alpha = 226.67$	$\rho = 0.1, \alpha = 670$	$\rho = 0.8, \alpha = 5$	$\rho = 1, \alpha = 20$	$\rho = 1, \alpha = 20$	$\rho = 1, \alpha = 20$
SSC-OMP	$\varepsilon = 2^{-52}, k_{max} = 2$	$\varepsilon = 2^{-52}, k_{max} = 2$	$\varepsilon = 2^{-52}, k_{max} = 2$	$\varepsilon = 2^{-52}, k_{max} = 2$	$\varepsilon = 2^{-52}, k_{max} = 29$	$\varepsilon = 2^{-52}, k_{max} = 17$
EnSC	$\lambda = 0.01, \alpha = 98$	$\lambda = 0.88, \alpha = 3$	$\lambda = 0.99, \alpha = 3$	$\lambda = 0.95, \alpha = 3$	$\lambda = 0.95, \alpha = 50$	$\lambda = 0.95, \alpha = 3$

Table 6.3: Parameters used in experiments on real datasets for all algorithms considered.

The COIL-20 [149] and COIL-100 [148] datasets consist of 72 images of 20 and 100 distinct objects (respectively) under a variety of rotations. All images are of size 32×32. On both datasets, we whiten by removing the first singular component when it improves algorithm performance.

The USPS dataset provided by [37] contains 9,298 total handwritten digits of size 16×16 with roughly even label distribution. No preprocessing is performed on this dataset.

The MNIST dataset [119] contains a total of 70,000 handwritten digits, of which we consider only the 10,000 "test" images. The images have nominal size 29×29, and we use the output of the scattering convolutional network [32] of size 3,472 and then project onto the first 500 principal components as in [216].

For all algorithms, we set K to be the correct number of clusters. For EKSS, we set $B = 1000$ and $T = 3$ for all datasets except MNIST, for which we set $T = 30$. Due to the benefits demonstrated in [76], we employed CoP-KSS instead of KSS as a base clustering algorithm for the USPS and MNIST datasets. For a fair comparison

to KSS, CoP-KSS, and MKF, we ran 1000 trials of each and use the clustering result that achieves the lowest clustering error. The parameters used for all experiments are shown in Table 6.3, with the most common parameters given among the 155 datasets for the Hopkins database. For the Hopkins, Yale, and COIL-20 datasets, we performed extensive model sweeps over a wide range of values for each parameter for each algorithm. For the larger COIL-100, USPS, and MNIST-10k datasets, this was infeasible for SSC-ADMM and EnSC, so the values were instead chosen from an intelligently-selected subset of parameters.

CHAPTER VII

Sequences of unions of subspaces for data with heterogeneous complexity

In important applications ranging from medical imaging [137] to multi-band signal processing [143] and genetics [191], signals of interest are well-approximated by sparse linear combinations of atomic signals from a dictionary. Equivalently, such signals are well-approximated by a union of low-dimensional subspaces generated by the dictionary where each sparsity level has an associated union of subspaces (UoS) generated by sparse combinations of correspondingly many atoms. Considering a sequence of sparsity levels yields a sequence of unions of subspaces (SUoS) of increasing dimension. This chapter considers the problem of learning such an SUoS from data.

While each UoS is combinatorially large with respect to sparsity level, our learning approach exploits the fact that sparsity is structured for many signals of interest, i.e., that certain collections of atoms are more frequently used together than others. This is known as group sparsity structure and has been studied extensively when the structure is known a priori. This chapter instead supposes that the structure is unknown, and we seek to learn it from training data. We also adapt the subspaces we obtain to improve representation and parsimony, similar to the goal of adapting atoms in dictionary learning. Finally, a denoising example illustrates the benefits of learning a dictionary-based SUoS; using a more parsimonious and representative SUoS results in improved recovery of complicated structures and edges.

This work was conducted jointly with (then undergraduate) Robert Malinas, and led to the conference paper that this chapter presents:

[99] David Hong, Robert P. Malinas, Jeffrey A. Fessler, and Laura Balzano. Learning Dictionary-Based Unions of Subspaces for Image Denoising. In *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, September 2018. doi: 10.23919/eusipco.2018.8553117.

7.1 Introduction

Consider signals $x \in \mathbb{C}^m$ that are well-approximated by sparse linear combinations from a large (over-complete) set of dictionary atoms, i.e., suppose that

$$\min_{z:\|z\|_0 \leq k} \|x - \mathbf{D}z\|_2 \leq \varepsilon\sqrt{m}, \tag{7.1}$$

where

- $\mathbf{D} \in \mathbb{C}^{m \times n}$ is a *dictionary* with n unit norm columns $d_1, \ldots, d_n \in \mathbb{C}^m$ referred to as *atoms*,
- k is the *sparsity level* (typically much smaller than n), and
- ε is the approximation root mean square error (RMSE).

Any signal that is exactly k-sparse in the dictionary $\mathbf{D}$, i.e., that satisfies (7.1) with $\varepsilon = 0$, lies in the span of the atoms identified by the support of its k-sparse coefficient vector z. Hence, it has often been noted that such signals lie in a union of $\binom{n}{k}$ subspaces, each of dimension k.

Since the results of [39, 60] showed that it is possible to efficiently recover these signals from only $O(k \log n)$ measurements using ℓ_1 optimization, this model has been applied widely for signal denoising and inverse problems. It has also been widely recognized that not all $\binom{n}{k}$ possible k-sparse supports are equally likely, resulting in an extensive literature on group sparsity or structured sparsity constraints for signal representation and recovery. Identifying groups corresponds exactly to selecting a subset in the union of k-dimensional subspaces (7.2) and hence this model is also called a *structured* union of subspaces [66, 213]. In the vast majority of research, however, the group structure is assumed known. This chapter considers learning the group structure from data.

In general, learning which of the combinatorially many possible supports are most relevant for a given dataset is challenging. Our key insight is that the lowest-dimensional models represent the bulk of the signals for some datasets. Hence, we first learn 1-sparse supports, then 2-sparse supports and so on, where at each stage we seek to represent only the data not already well approximated. The training data associated with each support can then be collected and used to learn an even lower-dimensional subspace. We can also discard subspaces associated with only a few signals; doing so further simplifies the model and increases overall representation error only slightly. Organizing the remaining subspaces by dimension yields a sequence of unions of subspaces (SUoS) of increasing dimension.

This chapter proposes an algorithm based on this intuition for learning a parsimonious and representative SUoS from data. We demonstrate the benefit of the learned model over unstructured sparsity by applying it to image denoising.

7.2 Related Work

7.2.1 Hidden Markov Models for Wavelet coefficients

Motivated by the observation that wavelet coefficients are typically correlated within and across scales, [53, 175] propose learning hidden Markov models with tree-structure to capture the correlations among the coefficients then using them to improve signal estimation and classification. Capturing these correlations provides rich information about the sparse coefficients and their relationships. In contrast, our proposed approach learns only the structure of supports, but immediately applies without extension to dictionaries without tree-structure.

7.2.2 Structured Sparsity and Group Lasso

Numerous applications ranging from multi-band signal processing [143] to genetics [191] motivated extensive work in the past decade on both theory and algorithms that exploit known group structure in supports to improve signal/subspace recovery from compressive measurements [18, 61, 66, 171, 213] and classification [170]. Example structures include non-overlapping groups [101], overlapping groups [105, 165], tree-structured groups [115], and even groups with internal sparsity [185]. This chapter is largely inspired by the benefits of capturing structured sparsity that they demonstrate, but we focus instead on how to learn unknown structures from data.

7.2.3 Learning the structure for structured sparsity

The authors of [182] propose a statistical model for structured sparsity and an inference scheme for its hyperparameters. In contrast to [182] and the works discussed above, our proposed approach learns new subspaces that need not be generated from atoms of the dictionary and so may provide more parsimonious representations. Namely, we focus more on learning arbitrary sequences of unions of subspaces than on sparsity structure for a given dictionary. Still, we use sparsity structure to first cluster the data; incorporating ideas from [182] in that step would be an interesting avenue for future work.

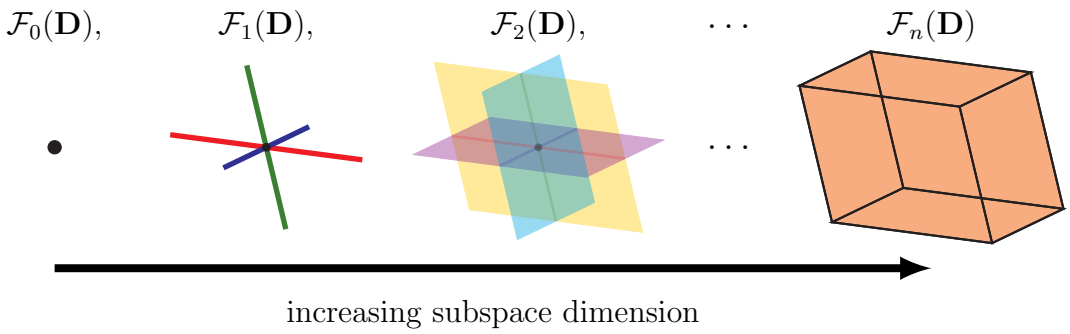


Figure 7.1: The sequence of unions of subspaces (SUoS) generated by a dictionary.

7.2.4 Subspace clustering

As discussed in Section 2.4, subspace clustering [205] groups data to their nearest-subspaces and can be used to learn a union of subspaces by simply learning a subspace for each cluster. Likewise, our proposed approach clusters data then learns a subspace for each cluster. However, it generally differs from other subspace clustering approaches by exploiting structured sparsity with respect to an initial dictionary to select the number of clusters. Additionally, while many subspace clustering techniques can learn subspaces of different dimensions, an SUoS may further have higher-dimensional subspaces that entirely contain lower-dimensional subspaces. Subspace clustering techniques do not typically learn this type of structure, but this feature is critical for SUoS models to generalize dictionary sparsity.

7.2.5 Dictionary learning

Dictionary learning adapts dictionary atoms to more parsimoniously represent data [8, 129] and our proposal shares this trait. As with any collection of subspaces, one can also obtain a learned dictionary from our proposal by using the subspace basis vectors as atoms and assuming the corresponding non-overlapping group sparsity. In contrast to dictionary learning approaches, however, we first cluster the data and then learn subspaces for them. A notable consequence is that the (effective) number of atoms and the sparsity structure are learned from data.

7.3 Learning a Sequence of Unions of Subspaces

The set of all k-sparse signals in a given dictionary $\mathbf{D}$ forms a union of $\binom{n}{k}$ many k-dimensional subspaces, defined as:

$$\mathcal{F}_k(\mathbf{D}) := \{\mathbf{D}z : z \in \mathbb{C}^n, \|z\|_0 \leq k\} = \bigcup_{\mathcal{I} \in \Omega_k} \mathcal{R}(\mathbf{D}_\mathcal{I}), \tag{7.2}$$

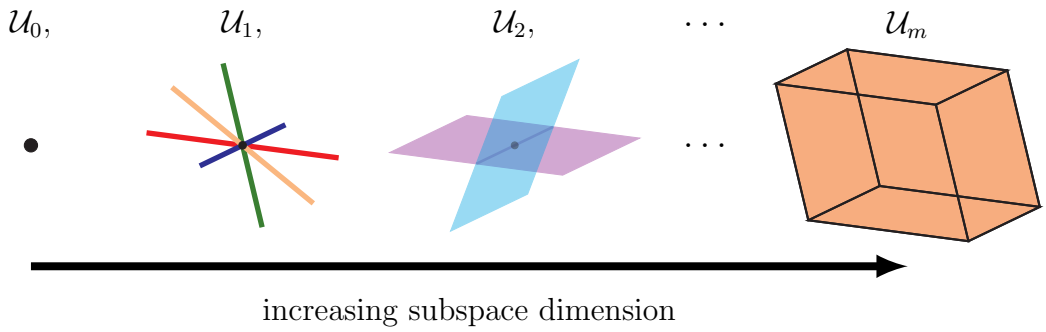


Figure 7.2: A general sequence of unions of subspaces (SUoS). This one has no generating dictionary.

where $\mathcal{R}(\cdot)$ is the column span of its argument, $\mathbf{D}_{\mathcal{I}}$ is a matrix formed from the columns of $\mathbf{D}$ indexed by $\mathcal{I}$, and the union is carried out over the $\binom{n}{k}$ index sets in

$$\Omega_k := \{\mathcal{I} \subset \{1, \dots, n\} : |\mathcal{I}| = k\}.$$

Taking all sparsity levels yields the sequence of unions of subspaces (SUoS) in Fig. 7.1. Note that this sequence is distinct from a union of unions; $\mathcal{F}_0(\mathbf{D}) \cup \cdots \cup \mathcal{F}_n(\mathbf{D})$ is actually just $\mathcal{F}_n(\mathbf{D})$ since it contains the rest.

7.3.1 Goal: Learn a "parsimonious" SUoS from data

We aim to learn a general SUoS $\mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_m \subseteq \mathbb{C}^m$, e.g., as shown in Fig. 7.2, that closely approximates a given collection of T training vectors $x_1, \dots, x_T \in \mathbb{C}^m$. Each $\mathcal{U}_k$ is a (potentially empty) union of N_k many k-dimensional subspaces

$$\mathcal{U}_k := \bigcup_{i=1}^{N_k} \mathcal{R}(\mathbf{U}_{k,i}) = \bigcup_{i=1}^{N_k} \{\mathbf{U}_{k,i} z : z \in \mathbb{C}^k\}, \tag{7.3}$$

where the columns of $\mathbf{U}_{k,i} \in \mathbb{C}^{m \times k}$ span a k-dimensional subspace. We consider $\{0\}$ to be a zero-dimensional subspace.

Note that $\mathcal{U}_0$ must be either $\{0\}$ or $\emptyset$, and likewise $\mathcal{U}_m$ must be either $\mathbb{C}^m$ or $\emptyset$. Beyond that, however, there are infinitely many choices for each of $\mathcal{U}_1, \dots, \mathcal{U}_{m-1}$ that produce perfect representation of the training vectors. Two such choices are always:

$$\mathcal{U}_0, \mathcal{U}_2, \dots, \mathcal{U}_m = \emptyset \qquad \mathcal{U}_1 = \bigcup_{i=1}^{T} \mathcal{R}(x_i), \tag{7.4}$$

$$\mathcal{U}_0, \dots, \mathcal{U}_{m-1} = \emptyset \qquad \mathcal{U}_m = \mathbb{C}^m. \tag{7.5}$$

However, (7.4) is undesirable because it does not generalize from the data; only scaled training vectors appear in the SUoS. It is not parsimonious because $\mathcal{U}_1$ contains many

subspaces. On the other hand, (7.5) has only one subspace but is not parsimonious because it is not low-dimensional.

We seek low-dimensional subspaces, where each represents nontrivially many training vectors. This requires balancing the trade-off between using low-dimensional subspaces and using subspaces expressive enough to represent diverse data vectors. Formulating this goal precisely and cleanly is challenging and ongoing work.

7.3.2 Proposal: A dictionary-based SUoS learning algorithm

We consider learning an SUoS $\mathcal{U}_0, \ldots, \mathcal{U}_m$ where each subspace approximates a subset of the training vectors $x_1, \ldots, x_N$ identified by their *structured sparsity* in a dictionary. Given a dictionary $\mathbf{D}$, approximation tolerances $\varepsilon_{\mathrm{s}}, \varepsilon_{\mathrm{u}}$ and a threshold number of training vectors τ, the proposed method has the following steps:

1. Sparsely approximate each training vector x_t with the dictionary $\mathbf{D}$ by sparse coding: for $t = 1, \ldots, T$ solve

   $$\hat{z}_t = \operatorname*{argmin}_{z_t \in \mathbb{C}^n} \|z_t\|_0 \quad \text{s.t.} \quad \|x_t - \mathbf{D}z_t\|_2 \leq \varepsilon_{\mathrm{s}}\sqrt{m}. \tag{7.6}$$

   Under mild conditions on the dictionary [199], orthogonal matching pursuit (OMP) solves (7.6) efficiently and reliably. OMP solves (7.6) exactly for orthogonal atoms.

2. Cluster the training vectors by the atoms in their sparse approximation, i.e., by the supports $\operatorname{supp}(\hat{z}_t)$.

3. Discard clusters containing fewer than τ training vectors, obtaining L clusters $\mathcal{X}_1, \ldots, \mathcal{X}_L \subset \{x_1, \ldots, x_T\}$.

4. Learn an orthonormal subspace basis $\mathbf{U}_\ell \in \mathbb{C}^{m \times k_\ell}$ for each cluster $\mathcal{X}_\ell$ by minimizing the root mean square approximation error

   $$\rho(\mathbf{U}_\ell, \mathcal{X}_\ell) := \sqrt{\frac{1}{|\mathcal{X}_\ell|} \sum_{x \in \mathcal{X}_\ell} \|x - \mathbf{U}_\ell \mathbf{U}_\ell^{\mathsf{H}} x\|_2^2}, \tag{7.7}$$

   where k_ℓ is the smallest dimension that results in a $\mathbf{U}_\ell$ within the approximation tolerance $\rho(\mathbf{U}_\ell, \mathcal{X}_\ell) \leq \varepsilon_{\mathrm{u}}\sqrt{m}$.

   We find the dimension k_ℓ and associated $\mathbf{U}_\ell \in \mathbb{C}^{m \times k_\ell}$ via the singular value decomposition by noting that k_ℓ is the smallest value for which

   $$\frac{1}{\sqrt{|\mathcal{X}_\ell|}} \sqrt{\sum_{j > k_\ell} \lambda_j^2(\mathbf{X}_\ell)} \leq \varepsilon_{\mathrm{u}}\sqrt{m}, \tag{7.8}$$

where $\lambda_j(\mathbf{X}_\ell)$ is the jth singular value of the matrix $\mathbf{X}_\ell \in \mathbb{C}^{m\times|\mathcal{X}_\ell|}$ whose columns are the $|\mathcal{X}_\ell|$ training vectors in $\mathcal{X}_\ell$. The columns of $\mathbf{U}_\ell$ are simply the first k_ℓ left singular vectors of $\mathbf{X}_\ell$ [63].

5. Collect the subspace bases $\mathbf{U}_1, \dots, \mathbf{U}_L$ by their dimensions $k_1, \dots, k_L$, obtaining the unions of subspaces:

$$\mathcal{U}_k = \bigcup_{\ell:k_\ell=k} \mathcal{R}(\mathbf{U}_\ell), \qquad k \in \{0, \dots, m\}. \tag{7.9}$$

Steps 1–3 exploit structured sparsity to form clusters of training signals that we hope lie near low-dimensional subspaces that are learned in steps 4–5. In this way, the approach combines learning sparsity structure like [182] with adaptation to the data like in dictionary learning [8].

Note that the approach automatically chooses how many subspaces of each dimension to include, and encourages a parsimonious SUoS with low-dimensional subspaces that all represent nontrivially many training vectors. Furthermore, the approach is efficient; the primary sources of computational cost are sparse coding, which is done efficiently via OMP, and the singular value decomposition of each cluster.

7.4 Denoising with a general sequence of unions of subspaces

This section describes how to use an SUoS for denoising. Denoising a vector $y \in \mathbb{C}^m$ using (unstructured) sparsity can be accomplished by solving the sparse coding problem:

$$\hat{z} \in \operatorname*{argmin}_{z\in\mathbb{C}^n} \|z\|_0 \quad \text{s.t.} \quad \|y - \mathbf{D}z\|_2 \le \varepsilon\sqrt{m}, \tag{7.10}$$

then returning the "denoised" vector $\hat{x} = \mathbf{D}\hat{z}$. We propose a generalization of this scheme to arbitrary SUoS models as follows: given an SUoS $\mathcal{U}_0, \dots, \mathcal{U}_m$ solve the low-dimensional coding problem:

$$(\hat{z}, \hat{\mathbf{U}}) \in \operatorname*{argmin}_{\substack{z\in\mathbb{C}^k, \mathbf{U}\in\mathcal{U}_k \\ k\in\{0,\dots,m\}}} k \quad \text{s.t.} \quad \|y - \mathbf{U}z\|_2 \le \varepsilon\sqrt{m}, \tag{7.11}$$

then returning $\hat{x} = \hat{\mathbf{U}}\hat{z}$. In essence, we seek to project y onto the lowest dimensional subspace that approximates it with RMSE within ε. For a dictionary-generated SUoS $\mathcal{F}_0(\mathbf{D}), \dots, \mathcal{F}_n(\mathbf{D})$, (7.11) is precisely a restatement of the sparsity approach; the subspace dimension k in (7.11) corresponds exactly to the sparsity $\|z\|_0$ in (7.10). One might solve the general denoising objective (7.11) with the following procedure:

1. Initialize $k = 0$.

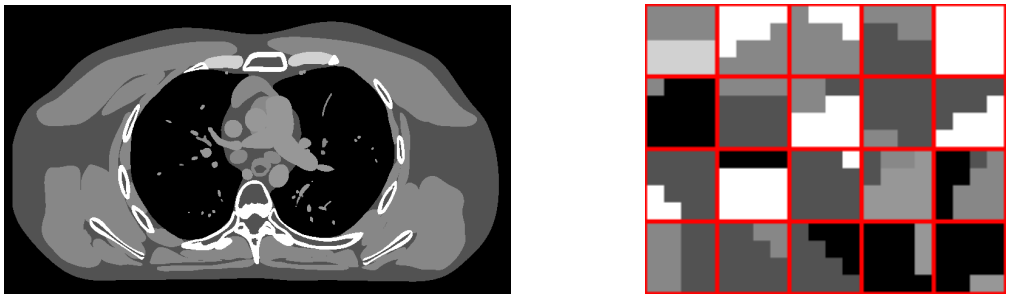


Figure 7.3: Training slice (475×835) of the XCAT digital phantom [177, 178] and a set of randomly selected 4×4 patches. The display window for both is $[900, 1100]$ HU.

2. Select the subspace basis $\mathbf{U}$ among those in $\mathcal{U}_k$ that maximizes the projection length $\|\mathbf{U}^{\mathsf{H}} y\|_2 = \|\mathbf{U}\mathbf{U}^{\mathsf{H}} y\|_2$ since that minimizes the residual

$$\min_{z \in \mathbb{C}^k} \|y - \mathbf{U}z\|_2 = \|y - \mathbf{U}\mathbf{U}^{\mathsf{H}} y\|_2 = \sqrt{\|y\|_2 - \|\mathbf{U}\mathbf{U}^{\mathsf{H}} y\|_2},$$

   by Pythagorean theorem.

3. If $\|y - \mathbf{U}\mathbf{U}^{\mathsf{H}} y\|_2 \leq \varepsilon\sqrt{m}$, return $\mathbf{U}\mathbf{U}^{\mathsf{H}} y$ as $\hat{x}$. Otherwise, increment k and go back to step 2.

The exhaustive search in step 2 may appear worrisome, but for parsimonious SUoS we hope to have relatively few subspaces in each union. Moreover, we hope that most signals are close to low-dimensional subspaces and can exit early in the algorithm.

Varying ε trades off between model error and noise; larger choices allow approximation by lower-dimensional subspaces that further suppress noise but that are also less likely to be representative. Adapting the dimension in this way is desirable for diverse signal classes such as image patches where some are nearly constant while others may be highly textured.

7.5 Experiments on an X-ray CT digital phantom

This section illustrates learning an SUoS for patches of an axial slice of the XCAT digital phantom [177, 178] then using it for denoising.

7.5.1 Learning an SUoS

We learn an SUoS for 4×4 patches extracted from a 475×835 slice of the XCAT phantom, shown in Figure 7.3 with a display window of 900 to 1100 modified Hounsfield units (HU). Extracting all overlapping 4×4 patches yields $T = 392704$ training samples in $\mathbb{R}^{16}$, 53444 of which are not constant. We use 2D orthogonal Haar

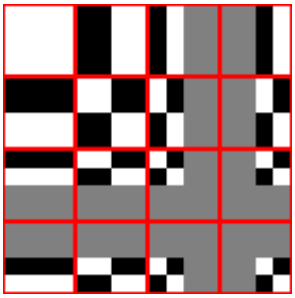


Figure 7.4: Atoms of the 2D orthogonal Haar wavelet dictionary.

Table 7.1: Number of unique supports at each sparsity level for XCAT patches ($\varepsilon_{\text{s}} = 5$ HU).

Sparsity	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Total
$\tau = 1$	1	1	7	10	11	15	29	50	67	89	98	101	63	74	32	4	0	652
$\tau = 25$	1	1	2	4	1	0	8	11	11	16	6	16	11	10	2	4	0	104
# possible	1	16	120	560	1820	4368	8008	11440	12870	11440	8008	4368	1820	560	120	16	1	65536

wavelets (Figure 7.4) as the input dictionary $\mathbf{D} \in \mathbb{R}^{16\times 16}$ since the XCAT phantom is piecewise constant, and we set the approximation tolerances to $\varepsilon_{\text{s}} = \varepsilon_{\text{u}} = 5$ HU based on a rough desired precision. The threshold number of training vectors $\tau = 25$ is chosen to remove sufficiently rare subspaces; note that $\tau/T = 25/392704 \approx 0.006\%$ of the training data.

Table 7.1 shows the number of unique supports obtained at each sparsity level k after step 3 of the learning algorithm for both $\tau = 1$ (i.e., no clusters discarded) and $\tau = 25$, in addition to the number of possible supports $\binom{16}{k}$. Discarding small clusters ($\tau = 25$) discards 2113 patches (approximately 0.5% of the training data) and reduces the number of unique supports from 652 to 104, but even before discarding any ($\tau = 1$), there are already many fewer supports than the $2^{16} = 65536$ possible. The patches are also sparsely representable by the 2D Haar wavelets overall, with an average of $(1/T)\sum_{t=1}^{T} \|z_t\|_0 = 1.6$ nonzero coefficients per patch. However, a nontrivial number of patches are not easily represented by sparse combinations of these wavelets, evidenced by the dense supports containing over eight atoms found both when $\tau = 1$ and when $\tau = 25$. The second stage of learning finds lower-dimensional subspace representations for these patches.

Table 7.2: Number of subspaces learned at each dimension for XCAT patches ($\varepsilon_{\text{s}} = \varepsilon_{\text{u}} = 5$ HU).

Dimension	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Total
$\tau = 1$	1	291	106	79	40	49	22	16	10	11	13	8	4	0	2	0	0	652
$\tau = 25$	1	1	11	13	10	15	11	4	6	11	7	8	4	0	2	0	0	104

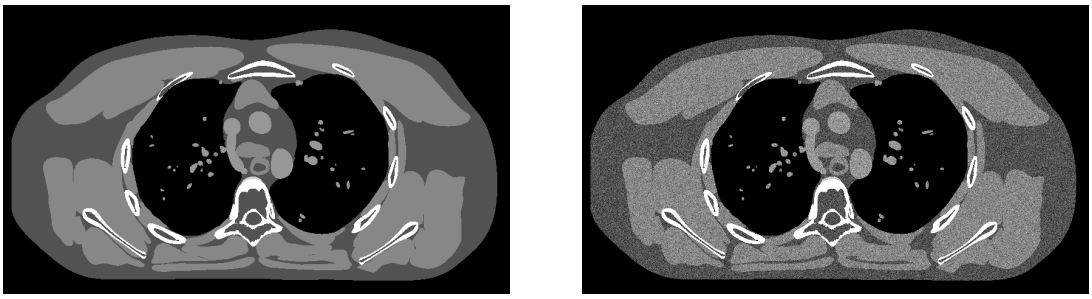


Figure 7.5: Test slice (475×835) of the XCAT digital phantom [177, 178] on left with a noisy version on right (noise std. dev. of 20 HU). Display window is $[900, 1100]$ HU.

Table 7.2 shows the number of subspaces obtained at each dimension after completion of the learning algorithm for both $\tau = 1$ and $\tau = 25$. As each subspace is formed from a cluster identified in steps 1–3, there are once again 652 subspaces when $\tau = 1$ and 104 when $\tau = 25$. Compared with the sparsity of supports in Table 7.1, however, the dimensions of the final learned subspaces tend to be significantly smaller. Adapting a subspace to each cluster allows for low-dimensional subspaces when the cluster contains signals that are similar but not sparse in the input dictionary, and seeking an *average* approximation error within ε_{u} allows for even lower-dimensional subspaces that only approximate the cluster overall.

The learning algorithm automatically avoids the trivial solutions (7.4) and (7.5) by exploiting structured sparsity in the 2D Haar wavelets to cluster and by adapting the subspaces to obtain 104 generally low-dimensional subspaces that all represent nontrivially many training vectors. Using a laptop with an Intel Core i5-6300U CPU (2.40 GHz, 2.50 GHz) and 8 GB of RAM, learning the subspaces from the 392704 patches takes around 15 seconds with unoptimized code written in Julia.

7.5.2 Denoising using the learned SUoS

We denoised a 475×835 test slice extracted from another portion of the XCAT phantom that has additive zero-mean Gaussian noise with a standard deviation of 20 HU as shown in Figure 7.5. We first denoised all 4×4 patches extracted from the noisy image and then combined the denoised patches back into a denoised image, averaging where patches overlap. Figure 7.6 shows absolute error maps for the denoised images obtained when patches are denoised by: a) solving (unstructured) sparse coding (7.10) with 2D Haar wavelets, or b) using the learned SUoS. We chose a tolerance of $\varepsilon = 27$ HU for both; it seemed to produce the best sparse coding performance in our experiments. Using a laptop with an Intel Core i5-6300U CPU (2.40 GHz, 2.50 GHz) and 8 GB of RAM, denoising the 475×835 test slice takes

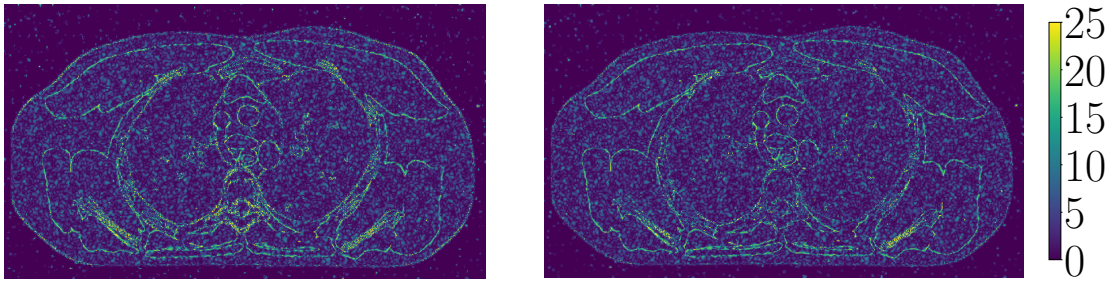


Figure 7.6: Absolute error maps in $[0, 25]$ HU range for images denoised using unstructured sparse coding (left) and the learned SUoS (right) with a tolerance of $\varepsilon = 27$ HU.

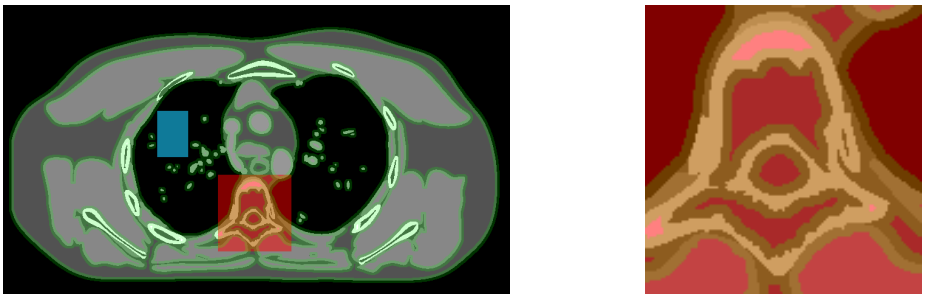


Figure 7.7: Color overlays (zoomed in on right), showing locations of the regions of interest: edge vicinity (green), spine (red), their intersection (yellow), and lung (cyan).

around 4 seconds for both sparse coding and the learned SUoS with unoptimized Julia code.

Comparing against the true (noiseless) test slice, sparse coding obtains an overall RMSE of 5.1 HU and the learned SUoS obtains a slight improvement to 4.6 HU. However, edge detail is important for these images, and the error maps reveal that the learned SUoS generally recovers edges more accurately, especially around the spine. To investigate further, we consider four regions of interest (ROI) shown in Figure 7.7: a) an edge ROI obtained by dilating the edge map provided by a Canny edge detector, b) a spine ROI, c) their intersection, and d) a lung ROI. The RMSE's (in HU) on all ROI's are:

	Edge	Spine	Intersection	Lung
Sparse coding	8.6	8.5	11.1	**3.6**
Learned SUoS	**7.5**	**6.1**	**7.5**	**3.6**

There is practically no improvement in the lung ROI, where the XCAT phantom is nearly constant and the two models provide equally parsimonious representations. However, the learned SUoS better recovers detailed regions, most significantly seen in the intersection ROI, i.e., around edges in the spine.

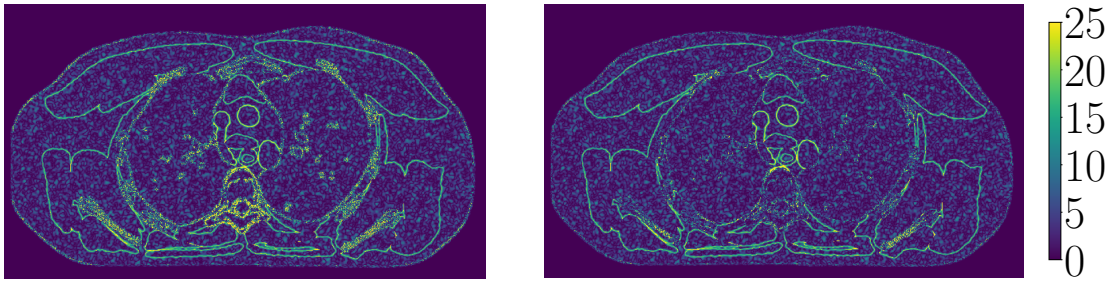


Figure 7.8: Absolute error maps in $[0, 25]$ HU range for images denoised using unstructured sparse coding (left) and the learned SUoS (right) with a larger tolerance of $\varepsilon = 50$ HU.

Choosing the best ε can be a challenge in practice. Denoising again but with a larger tolerance $\varepsilon = 50$ HU yields error maps shown in Figure 7.8 and the following ROI RMSE's (in HU):

	Edge	Spine	Intersection	Lung
Sparse coding	10.7	12.7	16.9	**3.4**
Learned SUoS	**8.4**	**7.0**	**8.9**	**3.4**

Since the learned SUoS captures more of the structure, it is significantly more robust to overestimating ε.

7.6 Conclusion

This chapter introduced the sequence of unions of subspaces (SUoS) model that unifies and generalizes union of subspace models and dictionary sparse models. We proposed a method for learning a dictionary-based SUoS and illustrated the benefits of the learned model with image denoising. Interesting avenues of future work include understanding how to choose the approximation tolerances and threshold number of atoms in learning as well as understanding the impact of the seed dictionary used, especially if it was learned or over-complete. Formulating a precise and clean learning objective is also an interesting and important challenge that might provide a more principled foundation for further work on this model. A final avenue of future work is the application of this model to inverse problems such as image reconstruction.

CHAPTER VIII

Conclusion and open problems

This dissertation studied low-dimensional modeling for the high-dimensional and heterogenous data that are increasingly common in modern data analysis. In these new regimes, some traditional intuitions and techniques break down. New theory and techniques are needed to unlock the full potential for discoveries facilitated by the scale and diversity of modern data. Chapter III analyzed the asymptotic performance of the standard and ubiquitous Principal Component Analysis (PCA) when samples have heterogeneous noise variance, characterizing how this heteroscedasticity harms PCA performance. Chapter IV analyzed a weighted variant of PCA that gives noisier samples less influence and found optimal weights that turn out to more aggressively downweight noisy samples than the typical choice of inverse noise variance weights. Chapter V considered data with heterogeneous statistical properties and generalized the increasingly standard Canonical Polyadic (CP) tensor decomposition to provide a unified algorithmic framework for many general notions of fit beyond the traditional least-squares. The final two chapters, Chapters VI and VII, considered unions of subspaces that model samples of heterogeneous type by combining several subspace models where each subspace models one of the classes. Chapter VI proposed and analyzed an ensemble method for clustering samples by associated subspace, and Chapter VII proposed an extension of unions of subspaces to a sequence of unions of subspaces that more systematically captures samples with heterogeneous complexity. Much work remains and one can expect that the need for theory and techniques suitable for large and heterogeneous data will only grow in the future. The conclusions of Chapters III to VII each describe avenues for future work in context; the remainder of this final chapter organizes and highlights a few.

8.1 Open problems in asymptotic (weighted) PCA analysis

The analyses of PCA and weighted PCA in Chapters III and IV leave several questions unanswered. First is the conjectured phase transition when $A(\beta_i) = 0$, stated as Conjecture 3.5 in the context of PCA and as Conjecture 4.8 in the more general context of weighted PCA. The claim is that the asymptotic recovery of the ith component is zero when $A(\beta_i) \leq 0$, which has been shown for unweighted PCA in the special cases where:

- the noise is homoscedastic and Gaussian [163], or
- there is only one component, i.e., $k = 1$ [94].

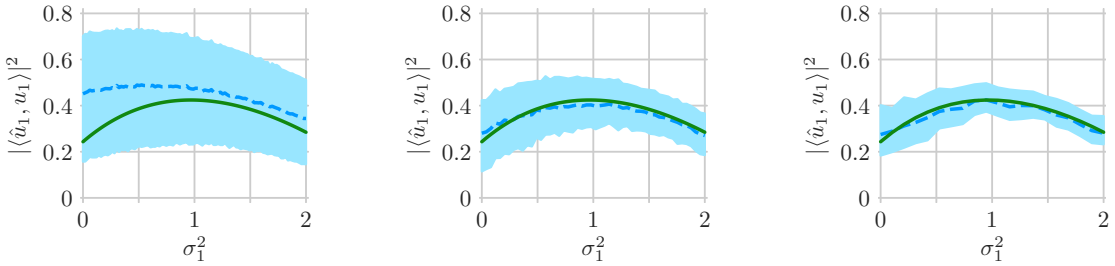
The numerical experiments in Section 3.4 suggest the conjecture holds in general, but proving this claim may involve showing that the noise singular values exhibit repulsion; see [22, Remark 2.13].

A second open problem is to characterize the unweighted recovery of the scores in weighted PCA. Theorem 4.3 characterizes the weighted score recovery

$$\sum_{j:\theta_j=\theta_i} \left| \left\langle \frac{\hat{z}_i}{\sqrt{n}}, \frac{z_j}{\sqrt{n}} \right\rangle_{\mathbf{M}} \right|^2,$$

with weighted Euclidean metric $\mathbf{M} = \mathbf{W}^2$ in (4.8), resulting in weighted aggregate score recovery (4.14) and weighted mean square error (4.15) in Corollary 4.7. Characterizing the unweighted recovery, i.e., $\mathbf{M} = \mathbf{I}_n$, is important because it would enable the optimization of weights for (unweighted) mean square error. Choosing weights to optimize the weighted mean square error is conceptually peculiar because the performance metric changes with the weights in this case. Numerical simulations in Fig. 4.10 suggest that unweighted recovery also concentrates in high dimensions; the challenge is in finding ways to extend our existing analysis tools to find the limit.

A third open problem is to explain a surprising phenomenon of unweighted PCA predicted by Theorem 3.4: adding noise can improve the performance of PCA, e.g., when there is extreme imbalance in noise levels. Plotting the asymptotic recovery (3.4) of unweighted PCA in Fig. 3.3 reveals this behavior. Increasing σ_1^2 from zero while keeping $\sigma_2^2 = 4$ fixed initially improves recovery; see the discussion in Section 3.3.3. Figure 8.1 shows numerical simulations further illustrating the behavior for data generated according to (3.1) with $c = 10$ samples per dimension, an underlying amplitude $\theta_1^2 = 1$, and $p_2 = 1\%$ of samples having noise variance $\sigma_2^2 = 7.5$ with the remaining $p_1 = 99\%$ of samples having noise variance σ_1^2 swept from 0 to 2. The interquartile interval (light blue ribbon) concentrates around the mean (dashed blue curve) as the data size increases from (a) to (b) and (c) while the mean approaches



(a) 1000 samples in 100 dims, 10000 trials.

(b) 10000 samples in 1000 dims, 1000 trials.

(c) 25000 samples in 2500 dims, 200 trials.

Figure 8.1: Simulated component recoveries $|\langle \hat{u}_1, u_1 \rangle|^2$ of (unweighted) PCA for data generated according to (3.1) with $c = 10$ samples per dimension, an underlying amplitude $\theta_1^2 = 1$, and $p_2 = 1\%$ of samples having noise variance $\sigma_2^2 = 7.5$ with the remaining $p_1 = 99\%$ of samples having noise variance σ_1^2 swept from 0 to 2. Simulation mean (dashed blue curve) and interquartile interval (light blue ribbon) are shown with the asymptotic prediction (3.4) of Theorem 3.4 (green curve). Increasing the data size from (a) to (b) and (c) shrinks the interquartile intervals, indicating concentration to the mean, which is itself converging to the asymptotic recovery. The performance of unweighted PCA does indeed sometimes improve with additional noise.

the asymptotic prediction (3.4) of Theorem 3.4 (green curve), indicating that Theorem 3.4 correctly predicts the limit. Moreover, the empirical recoveries themselves seem to initially improve as the noise level σ_1^2 increases, indicating that this behavior does not only occur in the limit, i.e., it is not an artifact of the asymptotic analysis. Nevertheless, the phenomenon is puzzling because degrading data quality by adding noise typically harms performance; neither inverse noise variance weighted PCA nor optimally weighted PCA exhibit this behavior as discussed in Section 4.7.3. Understanding why adding noise can aid unweighted PCA may provide new and valuable insight into PCA.

Further directions for future work on the analysis of (weighted) PCA with heteroscedastic noise are described in Sections 3.7 and 4.10. For example, one might consider a continuum of noise variances by supposing that the empirical noise variance distribution converges $(\delta_{\eta_1^2} + \cdots + \delta_{\eta_n^2}) \to \nu$ as $n \to \infty$; this would, e.g., allow the number of noise variances L to grow with n. Section 3.7 states our conjecture for unweighted PCA. Analyzing more general types of heterogeneity and associated forms of weighted PCA, e.g., heterogeneous noise variances both across variables and across samples with weights applied across both, is another important avenue, but the ability to first characterize unweighted recoveries will likely be crucial in this set-

ting. Finally, obtaining tight non-asymptotic analyses remains a large open problem. Results along this line of work would provide precise and rigorous characterizations of the empirical concentration seen in Sections 3.4 and 4.9.

8.2 Extensions and applications of weighted PCA

In Chapters III and IV, the asymptotic analyses of PCA and weighted PCA were used to characterize and gain various insights about:

- the positive bias of PCA amplitudes (Section 3.2.4)
- the impact of heteroscedasticity in PCA (Section 3.2.6)
- optimal weighting in weighted PCA (Sections 4.5 and 4.6)
- optimal sampling under linear sampling constraints (Section 4.8)
- the impact of data properties such as sample-to-dimension ratio, underlying amplitudes, proportions, and noise variances (Sections 3.3 and 4.7), and
- the impact of including noisier samples (Sections 3.3.4 and 4.7.4).

Many such opportunities remain to exploit the simple algebraic descriptions for asymptotic recovery (Theorems 3.4 and 4.3) to probe the behavior of (weighted) PCA. For example, the study of how data properties impact recovery in Sections 3.3 and 4.7 forms largely qualitative conclusions; more carefully quantifying the insights gained would be an interesting avenue for further work that the analyses in Chapters III and IV bring within reach. Another interesting direction is to characterize or bound the benefit of optimal weighting over inverse noise variance or square inverse noise variance weights, and identify the regimes where optimal weighting significantly improves asymptotic recovery.

An important avenue for extending weighted PCA is to develop a data driven approach that estimates the underlying amplitudes and noise variances directly from the data for use in an optimally weighted PCA. This extension would make optimally weighted PCA practical for a broader set of applications. Underlying amplitudes might be estimated from an initial PCA using (4.4) in Theorem 4.3. Estimating the noise variances might be done by observing that the normalized squared norm $\|y_i\|_2^2/d$ of any single sample should concentrate around its noise variance in high dimensions since the signal component has asymptotically zero relative energy. Grouping samples into clusters of similar noise variances could also be used to improve the estimates, though this clustering can become challenging if the number of groups

L grows with the number of samples n. Incorporating spectrum estimation methods such as [120, 141] is another promising approach, and one can further exploit knowledge of which samples share a noise variance by considering the spectrums of subsets of data. The noise spectrum might be isolated by dropping the first few singular values or by permuting the data as done in parallel analysis [57]; alternating between estimating components with weighted PCA and estimating noise variances can help mitigate interference from large principal components. Investigating these various approaches and analyzing their performance would be significant contributions to the theory and practice of weighted PCA. The analysis of weighted PCA in Chapter IV can quantify how much performance degrades when weights deviate from optimal, so it may help characterize the impact of errors in estimates of the underlying amplitudes and noise variances.

Another extension of weighted PCA is to other types of heterogeneity. For example, the current work assumes that only the noise level is heterogeneous, with all samples having the same underlying amplitudes. However, sometimes samples also reflect the underlying components in heterogeneous ways. For example, some samples may be more informative about the first component, while other samples may be more informative about the second component. Understanding how weighted PCA behaves in these settings, and determining how to modify PCA appropriately is an exciting challenge. Even more sophisticated forms of heterogeneity arise in real data, providing ample opportunity for further work along this direction.

Finally, it may be interesting to apply weighted PCA for two problems in MRI. The first problem is coil compression, where measurements from several physical coils are combined to form a smaller set of "virtual" coils that capture much of the signal. This dimensionality reduction is currently done in some settings [35, 45, 224] via (unweighted) PCA. However, one expects coils further from an area of interest to have more noise relative to the signal, so weighted PCA may be a natural choice here. The second problem is finding navigators. The goal here is to identify a low-frequency signal corresponding to motion, e.g., due to breathing, that can be used to compensate for motion in dynamic MRI. Once again, some approaches [158] use unweighted PCA, but one expects some heterogeneity (even heteroscedasticity), so weighted PCA may be an appropriate choice here as well.

8.3 Probabilistic PCA as an alternative to weighted PCA

Weighted PCA is a natural way of handling heteroscedastic noise across samples, but it is not clear that this approach is optimal. For example, [221] showed that an alternative iterative method has minimax optimal rate for noise that is heteroscedas-

tic within samples, i.e., across the entries of each sample. An alternative approach to weighted PCA for heteroscedastic noise across samples is to take a probabilistic PCA [195] approach by modeling the samples $y_1, \dots, y_n \in \mathbb{C}^d$ as

$$y_i = \mathbf{M} z_i + \eta_i \varepsilon_i, \qquad i \in \{1, \dots, n\}, \tag{8.1}$$

where $\mathbf{M} \in \mathbb{C}^{d \times k}$ contains latent factors, η_i^2 is the ith noise variance, and the coefficient vector $z_1, \dots, z_n \in \mathbb{C}^k$ and noise vectors $\varepsilon_1, \dots, \varepsilon_n \in \mathbb{C}^d$ are modeled as

$$z_1, \dots, z_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I}_k), \qquad \varepsilon_1, \dots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I}_d). \tag{8.2}$$

Namely, $y_i \sim \mathcal{N}(0, \mathbf{M}\mathbf{M}^{\mathsf{H}} + \eta_i^2 \mathbf{I}_d)$ for $i \in \{1, \dots, n\}$, so the maximum likelihood estimate of $\mathbf{M}$ is given by maximizing the log-likelihood

$$\mathcal{L}(\mathbf{M}) := \frac{1}{2} \sum_{i=1}^n \log \det(\mathbf{M}\mathbf{M}^{\mathsf{H}} + \eta_i^2 \mathbf{I}_d)^{-1} - \frac{1}{2} \sum_{i=1}^n y_i^{\mathsf{H}} (\mathbf{M}\mathbf{M}^{\mathsf{H}} + \eta_i^2 \mathbf{I}_d)^{-1} y_i, \tag{8.3}$$

where (8.3) drops an irrelevant $(2\pi)^{d/2}$ constant. An alternative is to optimize $\mathcal{L}(\mathbf{U}\mathbf{\Theta})$ with respect to $\mathbf{U} \in \mathbb{C}^{d \times k}$ with orthonormal columns and diagonal matrix $\mathbf{\Theta} \in \mathbb{R}^{k \times k}$, effectively working with the eigendecomposition $\mathbf{M}\mathbf{M}^{\mathsf{H}} = \mathbf{U}\mathbf{\Theta}^2\mathbf{U}^{\mathsf{H}}$ of $\mathbf{M}\mathbf{M}^{\mathsf{H}}$.

When the noise variances are homogeneous, i.e., $\eta_1^2 = \cdots = \eta_n^2 = \sigma^2$, this problem is solved by taking the principal eigenvectors and eigenvalues (with shrinkage) of the sample covariance matrix $(y_1 y_1^{\mathsf{H}} + \cdots + y_n y_n^{\mathsf{H}})/n$ as in PCA [195, Section 3.2], but the same is not true in general. Heterogeneous noise variances complicate the optimization problem, making algorithm development for this approach an important avenue for exploration. One approach is expectation maximization (EM) to optimize with respect to $\mathbf{M}$ in the style of [195, Appendix B]. Another approach is to alternate between optimizing with respect to $\mathbf{\Theta}$ and $\mathbf{U}$. The optimization with respect to $\mathbf{\Theta}$ turns out to separate into k univariate problems, each solvable by root-finding. The optimization with respect to $\mathbf{U}$ has several candidates, such as a minorize-maximize (MM) approach inspired by [23, 192] or gradient ascent on the Stiefel manifold [1, 64].

An interesting connection to optimally weighted PCA arises by applying the matrix inversion lemma to obtain

$$(\mathbf{U}\mathbf{\Theta}^2\mathbf{U}^{\mathsf{H}} + \eta_i^2 \mathbf{I}_d)^{-1} = \frac{1}{\eta_i^2}\mathbf{I}_d - \mathbf{U} \underbrace{\operatorname{diag}\left(\frac{1}{\eta_i^2} \frac{\theta_1^2}{\theta_1^2 + \eta_i^2}, \dots, \frac{1}{\eta_i^2} \frac{\theta_k^2}{\theta_k^2 + \eta_i^2} \right)}_{=: \mathbf{W}_i} \mathbf{U}^{\mathsf{H}}, \tag{8.4}$$

where the entries of $\mathbf{W}_i \in \mathbb{R}_+^{k \times k}$ are the (scaled) optimal weights in Theorem 4.10. Using (8.4) to simplify $\mathcal{L}(\mathbf{U}\mathbf{\Theta})$ yields

$$\mathcal{L}(\mathbf{U}\mathbf{\Theta}) = C + \frac{1}{2} \sum_{i=1}^n \log \det(\mathbf{\Theta}^2 + \eta_i^2 \mathbf{I}_k)^{-1} + \frac{1}{2} \sum_{i=1}^n y_i^{\mathsf{H}} \mathbf{U} \mathbf{W}_i \mathbf{U}^{\mathsf{H}} y_i, \tag{8.5}$$

with constant

$$C := \frac{1}{2}\sum_{i=1}^{n} \log\det(\eta_i^2 \mathbf{I}_{d-k})^{-1} - \frac{1}{2}\sum_{i=1}^{n} \frac{y_i^{\mathsf{H}} y_i}{\eta_i^2}.$$

Thus optimizing $\mathcal{L}(\mathbf{U}\boldsymbol{\Theta})$ with respect to $\mathbf{U}$ amounts to the maximization problem

$$\underset{\mathbf{U}\in\mathbb{C}^{d\times k}}{\operatorname{argmax}} \ \frac{1}{2}\sum_{i=1}^{n} y_i^{\mathsf{H}} \mathbf{U}\mathbf{W}_i\mathbf{U}^{\mathsf{H}} y_i \qquad \text{s.t.} \quad \mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{I}_k, \tag{8.6}$$

coinciding exactly with an optimally weighted PCA when $k = 1$.

8.4 Efficient algorithms for GCP tensor decomposition

The optimization problem involved in fitting GCP tensor models presents new challenges for developing efficient algorithms. General loss functions can destroy the structure typically exploited by fast alternating minimization approaches to CP tensor decomposition, making it difficult to fit GCP for large tensors. One approach might be to try to find quadratic majorizers for each subproblem in the alternating minimization that can then be efficiently minimized similarly to the alternating least squares used in CP tensor decomposition. The key challenge is finding a majorizer that is *generic* enough to easily handle general loss functions. Finding appropriate places to sketch the data could also be an interesting approach; see [21] for a recent work that does so for CP tensor decomposition. The challenge is again in handling general loss functions.

Another promising approach is to replace the gradients used in Chapter V with stochastic gradients formed from random subsets of data tensor entries. Doing so raises numerous interesting design questions. For example, one must choose how many entries to use for each stochastic gradient. Using too many entries eliminates the benefit; using too few entries yields noisier gradients and makes it challenging to efficiently reuse intermediately computed results. Another challenge in sampling entries to calculate stochastic gradients is in properly handling tensors where a few entries are highly informative. For example, the nonzero entries of a sparse tensor may be critical but might account for a small percentage of the tensor. Uniform sampling will likely miss these entries, and an approach for non-uniform sampling with general losses in GCP is an open problem. A third challenge is in efficiently choosing the step sizes for gradient descent. Theory to guide step size selection is an area of incredibly active research, and GCP fitting would be a natural setting for further work. Finally, assessing when to stop iterating often involves evaluating the objective function at each iterate, but this computation is impractical for large tensors and a natural approach is to form a stochastic estimate. In this case,

choosing how many entries to use in the estimate is another fundamental question. Intuitively, few samples are needed when the iterate is far from optimal since further iterations will descend the objective function rapidly, and many samples are needed close to optimality where greater "resolution" is needed to determine whether to stop. Making this qualitative intuition quantitative is an exciting challenge.

8.5 GCP tensor decompositions for heterogeneous data

A natural combination of the ideas in Chapters IV and V is to develop tensor decomposition techniques for heteroscedastic data. GCP allows for weighting already, and the core question again becomes how to choose the weights. Less is known about the recovery of underlying latent factors by tensor decompositions, so this question presents a challenging and important arena for fundamental work. Our previous work on PCA suggests that weighting samples more aggressively than inverse noise variance may be a good strategy, and developing new theory to understand if these intuitions carry over to tensors would be fascinating to work on. In many cases, tensor decompositions behave differently from their matrix counterparts, so new theory and insights are needed.

For simplicity, our discussion of GCP also focused on using a single elementwise loss function $f(x_i, m_i)$ for all entries of the tensor. However, the algorithmic framework of Chapter V easily allows for a different loss function for each entry, i.e., $f_i(x_i, m_i)$. The only modification is to the definition (5.30) of the elementwise derivative tensor $\mathcal{Y}$. Different loss functions may be appropriate, e.g., if the tensor contains a heterogeneous mixture of data types as studied for matrices in [202]. Further investigation of these possibilities would be exciting avenues for testing out the full potential of the framework in Chapter V.

8.6 Extended analysis of Ensemble K-subspaces

Our analysis of the Ensemble K-subspaces algorithm in Chapter VI characterized only the first iteration of K-subspaces, showing that the first iteration effectively reproduces thresholding-based subspace clustering (TSC) [91] and enjoys the same recovery guarantees. One might expect that allowing K-subspaces to instead iterate until convergence improves recovery by incorporating higher order correlations among samples, and we do indeed see an improvement in practice. However, analyzing this setting is challenging because the alternating nature of K-subspaces makes it difficult to track the statistics of the resulting affinity matrix. Extending the existing analysis to the general case of multiple iterations is an important next step. Another direction

for extension is in making the analysis tighter by better characterizing the geometry that drives, e.g., how many nearest-neighbors must be used in the thresholding step. In general, selecting the tuning parameters of the algorithm in practice remains an open question; better theory to guide this choice would be a significant contribution.

8.7 Principled approaches to learning an SUoS

Chapter VII proposed a generalization of the union of subspaces model to a sequence of unions of subspaces (SUoS), where the unions of subspaces in the sequence increase in dimension. Doing so enables the model to distinguish between samples that are simpler and can be well-represented by a low-dimensional subspace and those that require a higher-dimensional representation. The hope is that a few low-dimensional subspaces can adequately describe a bulk of signals of interest, while a few signals of interest lie in higher-dimensional subspaces. Dimension is the SUoS analogue to sparsity in dictionary sparse models. Chapter VII proposed a procedure for learning an SUoS by using the connection to dictionary sparsity to cluster samples then learn a subspace for each cluster. However, it remains unclear how SUoS models should be learned in a principled way. Developing an approach is complicated by the fact that parsimony in an SUoS model results both from having low-dimensional subspaces and from having few subspaces, and it is unclear how these competing objectives should be traded off. Finding principled ways to learn SUoS models, e.g., via an appropriate objective, will be an important but challenging next step.

8.8 Union of subspace and dictionary models for medical imaging.

Union of subspace models remain generally unexplored in settings such as medical imaging, except when they take the form of sparsity models. Developing ways of using more general union of subspaces models, or the sequence of unions of subspaces model proposed in Chapter VII, would be an exciting area of future work. A first step in this direction would be to develop efficient image reconstruction algorithms that use such models as regularization. The relevant optimization problem is combinatorial and typically large enough to make exhaustive search impractical. Interesting avenues include developing greedy approaches and considering models that are structured to ease optimization.

A related direction is to return to the problem of learning dictionary models from heterogeneous images as discussed in Section 2.5.2. This problem, in fact, motivated much of the work in this dissertation on learning models from heterogeneous data.

One path is to connect this question to the work of Chapters III, IV and VI, by connecting dictionary models to union of subspace models, as described in Section 2.5.1. In this context, some of the tools of Chapter VI may provide insights into how image patches can be clustered into subspaces that correspond to different sparse supports. Chapters III and IV may then give some insight into how well each of these subspaces that correspond to spans of dictionary atoms can be learned from data with heterogeneous noise. This approach would mirror the sparse code update and dictionary update steps common in dictionary learning methods. Without an oracle identifying the correct sparse code supports, however, the dictionary update step will likely contain heterogeneity beyond the heteroscedasticity considered in Chapters III and IV. These aspects pose new challenges to extending our current understanding and tools for analysis, and they highlight the many exciting opportunities for important work on these frontiers of modern data analysis.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. url: https://press.princeton.edu/titles/8586.html.

[2] Evrim Acar, Canan Aykut-Bingol, Haluk Bingol, Rasmus Bro, and Bülent Yener. Multiway analysis of epilepsy tensors. *Bioinformatics*, 23(13):i10–i18, July 2007. doi: 10.1093/bioinformatics/btm210.

[3] Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *Journal of Chemometrics*, 25(2):67–86, January 2011. doi: 10.1002/cem.1335.

[4] Evrim Acar, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mørup. Scalable Tensor Factorizations with Missing Data. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, April 2010. doi: 10.1137/1.9781611972801.61.

[5] Evrim Acar, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, March 2011. doi: 10.1016/j.chemolab.2010.08.004.

[6] Evrim Acar and Bülent Yener. Unsupervised Multiway Data Analysis: A Literature Survey. *IEEE Transactions on Knowledge and Data Engineering*, 21(1):6–20, January 2009. doi: 10.1109/tkde.2008.112.

[7] Pankaj K. Agarwal and Nabil H. Mustafa. k-means projective clustering. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '04*. ACM Press, 2004. doi: 10.1145/1055558.1055581.

[8] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006. doi: 10.1109/tsp.2006.881199.

[9] Greg W. Anderson, Alice Guionnet, and Ofer Zeitouni. *An Introduction to Random Matrices*. Cambridge University Press, 2009. doi: 10.1017/cbo9780511801334.

[10] Babak A. Ardekani, Jeff Kershaw, Kenichi Kashikura, and Iwao Kanno. Activation detection in functional MRI using subspace modeling and maximum likelihood estimation. *IEEE Transactions on Medical Imaging*, 18(2):101–114, 1999. doi: 10.1109/42.759109.

[11] Brett W. Bader and Tamara G. Kolda. Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping. *ACM Transactions on Mathematical Software*, 32(4):635–653, December 2006. doi: 10.1145/1186785.1186794.

[12] Brett W. Bader and Tamara G. Kolda. Efficient MATLAB Computations with Sparse and Factored Tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, January 2008. doi: 10.1137/060676489.

[13] Brett W. Bader, Tamara G. Kolda, et al. MATLAB Tensor Toolbox Version 3.0-dev. Available online, October 2017. url: https://www.tensortoolbox.org.

[14] Zhidong Bai and Jack W. Silverstein. *Spectral Analysis of Large Dimensional Random Matrices*. Springer New York, 2010. doi: 10.1007/978-1-4419-0661-8.

[15] Zhidong Bai and Jianfeng Yao. On sample eigenvalues in a generalized spiked population model. *Journal of Multivariate Analysis*, 106:167–177, April 2012. doi: 10.1016/j.jmva.2011.10.009.

[16] Grey Ballard, Nicholas Knight, and Kathryn Rouse. Communication Lower Bounds for Matricized Tensor Times Khatri-Rao Product. In *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, May 2018. doi: 10.1109/ipdps.2018.00065.

[17] Laura Balzano, Arthur Szlam, Benjamin Recht, and Robert Nowak. K-subspaces with missing data. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, August 2012. doi: 10.1109/ssp.2012.6319774.

[18] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-Based Compressive Sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, April 2010. doi: 10.1109/tit.2010.2040894.

[19] Rina Foygel Barber and Mladen Kolar. ROCKET: Robust confidence intervals via Kendall's tau for transelliptical graphical models. *The Annals of Statistics*, 46(6B):3422–3450, December 2018. doi: 10.1214/17-aos1663.

[20] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, February 2003. doi: 10.1109/tpami.2003.1177153.

[21] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. A Practical Randomized CP Tensor Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, January 2018. doi: 10.1137/17m1112303.

[22] Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, October 2012. doi: 10.1016/j.jmva.2012.04.019.

[23] Konstantinos Benidis, Ying Sun, Prabhu Babu, and Daniel P. Palomar. Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation. *IEEE Transactions on Signal Processing*, 64(23):6211–6226, December 2016. doi: 10.1109/tsp.2016.2605073.

[24] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997. url: http://athenasc.com/linoptbook.html.

[25] Gregory Beylkin, Jochen Garcke, and Martin J. Mohlenkamp. Multivariate Regression and Machine Learning with Sums of Separable Functions. *SIAM Journal on Scientific Computing*, 31(3):1840–1857, January 2009. doi: 10.1137/070710524.

[26] Gregory Beylkin and Martin J. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16):10246–10251, July 2002. doi: 10.1073/pnas.112329799.

[27] Gregory Beylkin and Martin J. Mohlenkamp. Algorithms for Numerical Analysis in High Dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, January 2005. doi: 10.1137/040604959.

[28] Peter J. Bickel and Elizaveta Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577–2604, December 2008. doi: 10.1214/08-aos600.

[29] Michael Biehl and Andreas Mietzner. Statistical mechanics of unsupervised structure recognition. *Journal of Physics A: Mathematical and General*, 27(6):1885–1897, March 1994. doi: 10.1088/0305-4470/27/6/015.

[30] P. S. Bradley and O. L. Mangasarian. k-Plane Clustering. *Journal of Global Optimization*, 16(1):23–32, 2000. doi: 10.1023/a:1008324625522.

[31] Rasmus Bro. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38(2):149–171, October 1997. doi: 10.1016/s0169-7439(97)00032-4.

[32] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, August 2013. doi: 10.1109/tpami.2012.230.

[33] Wlodzimierz Bryc. *The Normal Distribution*. Springer New York, 1995. doi: 10.1007/978-1-4612-2560-7.

[34] Samuel Rota Bulò, André Lourenço, Ana Fred, and Marcello Pelillo. Pairwise Probabilistic Clustering Using Evidence Accumulation. In *Structural, Syntactic, and Statistical Pattern Recognition. SSPR /SPR 2010.*, volume 6218 of *Lecture Notes in Computer Science*, pages 395–404. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-14980-1_38.

[35] Mark Bydder, Gavin Hamilton, Takeshi Yokoo, and Claude B. Sirlin. Optimal phased-array combination for spectroscopy. *Magnetic Resonance Imaging*, 26(6):847–850, July 2008. doi: 10.1016/j.mri.2008.01.050.

[36] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, September 1995. doi: 10.1137/0916069.

[37] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, August 2011. doi: 10.1109/tpami.2010.231.

[38] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, May 2011. doi: 10.1145/1970392.1970395.

[39] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006. doi: 10.1109/tit.2005.862083.

[40] Timothy I. Cannings and Richard J. Samworth. Random-projection ensemble classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):959–1035, June 2017. doi: 10.1111/rssb.12228.

[41] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, September 1970. doi: 10.1007/bf02310791.

[42] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-Sparsity Incoherence for Matrix Decomposition. *SIAM Journal on Optimization*, 21(2):572–596, April 2011. doi: 10.1137/090761793.

[43] Sourav Chatterjee. Matrix estimation by Universal Singular Value Thresholding. *The Annals of Statistics*, 43(1):177–214, February 2015. doi: 10.1214/14-aos1272.

[44] Eric C. Chi and Tamara G. Kolda. On Tensors, Sparsity, and Nonnegative Factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, January 2012. doi: 10.1137/110859063.

[45] Alan Chu and Douglas C. Noll. Coil compression in simultaneous multislice functional MRI with concentric ring slice-GRAPPA and SENSE. *Magnetic Resonance in Medicine*, 76(4):1196–1209, October 2015. doi: 10.1002/mrm.26032.

[46] Andrzej Cichocki and Shun ichi Amari. Families of Alpha- Beta- and Gamma- Divergences: Flexible and Robust Measures of Similarities. *Entropy*, 12(6):1532–1568, June 2010. doi: 10.3390/e12061532.

[47] Andrzej Cichocki and Anh-Huy Phan. Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E92-A(3):708–721, 2009. doi: 10.1587/transfun.e92.a.708.

[48] Andrzej Cichocki, Rafal Zdunek, Seungjin Choi, Robert Plemmons, and Shun ichi Amari. Non-Negative Tensor Factorization using Alpha and Beta Divergences. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. IEEE, April 2007. doi: 10.1109/icassp.2007.367106.

[49] Robert N. Cochran and Frederick H. Horne. Statistically weighted principal component analysis of rapid scanning wavelength kinetics experiments. *Analytical Chemistry*, 49(6):846–853, May 1977. doi: 10.1021/ac50014a045.

[50] Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. A Generalization of Principal Components Analysis to the Exponential Family. In *Advances in Neural Information Processing Systems 14*, pages 617–624. MIT Press, 2002. url: http://papers.nips.cc/paper/2078-a-generalization-of-principal-components-analysis-to-the-exponential-family.

[51] Fengyu Cong, Qiu-Hua Lin, Li-Dan Kuang, Xiao-Feng Gong, Piia Astikainen, and Tapani Ristaniemi. Tensor decomposition of EEG signals: A brief review. *Journal of Neuroscience Methods*, 248:59–69, June 2015. doi: 10.1016/j.jneumeth.2015.03.018.

[52] Romain Couillet and Merouane Debbah. *Random Matrix Methods for Wireless Communications*. Cambridge University Press, 2009. doi: 10.1017/cbo9780511994746.

[53] Matthew S. Crouse, Robert D. Nowak, and Richard G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, April 1998. doi: 10.1109/78.668544.

[54] Christophe Croux and Anne Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95(1):206–226, July 2005. doi: 10.1016/j.jmva.2004.08.002.

[55] J.-C. Deville and E. Malinvaud. Data Analysis in Official Socio-Economic Statistics. *Journal of the Royal Statistical Society. Series A (General)*, 146(4):335, 1983. doi: 10.2307/2981452.

[56] S. J. Devlin, R. Gnanadesikan, and J. R. Kettenring. Robust Estimation of Dispersion Matrices and Principal Components. *Journal of the American Statistical Association*, 76(374):354–362, June 1981. doi: 10.1080/01621459.1981.10477654.

[57] Edgar Dobriban. Permutation methods for factor analysis and PCA, 2018. arXiv: 1710.00479v2.

[58] Edgar Dobriban, William Leeb, and Amit Singer. PCA from noisy, linearly reduced data: the diagonal case, 2018. arXiv: 1611.10333v2.

[59] Edgar Dobriban, William Leeb, and Amit Singer. Optimal prediction in the linearly transformed spiked model. *The Annals of Statistics*, 2019. To appear. arXiv: 1709.03393v2.

[60] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006. doi: 10.1109/tit.2006.871582.

[61] Marco F. Duarte and Yonina C. Eldar. Structured Compressed Sensing: From Theory to Applications. *IEEE Transactions on Signal Processing*, 59(9):4053–4085, September 2011. doi: 10.1109/tsp.2011.2161982.

[62] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. Temporal Link Prediction Using Matrix and Tensor Factorizations. *ACM Transactions on Knowledge Discovery from Data*, 5(2):1–27, February 2011. doi: 10.1145/1921632.1921636.

[63] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936. doi: 10.1007/bf02288367.

[64] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, January 1998. doi: 10.1137/s0895479895290954.

[65] Noureddine El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, 36(6):2717–2756, December 2008. doi: 10.1214/07-aos559.

[66] Yonina C. Eldar and Moshe Mishali. Robust Recovery of Signals From a Structured Union of Subspaces. *IEEE Transactions on Information Theory*, 55(11):5302–5316, November 2009. doi: 10.1109/tit.2009.2030471.

[67] Ehsan Elhamifar and René Vidal. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, November 2013. doi: 10.1109/tpami.2013.57.

[68] Leyuan Fang, Nanjun He, and Hui Lin. CP tensor-based compression of hyperspectral images. *Journal of the Optical Society of America A*, 34(2):252–258, January 2017. doi: 10.1364/josaa.34.000252.

[69] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456, September 2011. doi: 10.1162/neco_a_00168.

[70] Ana Fred and Anil K. Jain. Evidence Accumulation Clustering Based on the K-Means Algorithm. In *Structural, Syntactic, and Statistical Pattern Recognition. SSPR /SPR 2002.*, volume 2396 of *Lecture Notes in Computer Science*, pages 442–451. Springer Berlin Heidelberg, 2002. doi: 10.1007/3-540-70659-3_46.

[71] Ana L. N. Fred and Anil K. Jain. Data clustering using evidence accumulation. In *Object recognition supported by user interaction for service robots*. IEEE Comput. Soc, 2002. doi: 10.1109/icpr.2002.1047450.

[72] Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, June 2005. doi: 10.1109/tpami.2005.113.

[73] Jochen Garcke. Classification with Sums of Separable Functions. In *Machine Learning and Knowledge Discovery in Databases*, pages 458–473. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-15880-3_35.

[74] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, June 2001. doi: 10.1109/34.927464.

[75] Joydeep Ghosh and Ayan Acharya. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):305–315, May 2011. doi: 10.1002/widm.32.

[76] Andrew Gitlin, Biaoshuai Tao, Laura Balzano, and John Lipor. Improving K-Subspaces via Coherence Pursuit. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1575–1588, December 2018. doi: 10.1109/jstsp.2018.2869363.

[77] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[78] Geoffrey J. Gordon. Generalized^2 linear^2 models. In *Advances in Neural Information Processing Systems 15*, pages 593–600. MIT Press, 2003. url: http://papers.nips.cc/paper/2144-generalized2-linear2-models.

[79] Michael J. Greenacre. *Theory and Applications of Correspondence Analysis*. London (UK) Academic Press, 1984. url: http://www.carme-n.org/?sec=books5.

[80] John A. Gubner. *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006. doi: 10.1017/cbo9780511813610.

[81] Wolfgang Hackbusch and Boris N. Khoromskij. Tensor-product approximation to operators and functions in high dimensions. *Journal of Complexity*, 23(4-6):697–714, August 2007. doi: 10.1016/j.jco.2007.03.007.

[82] Samantha Hansen, Todd Plantenga, and Tamara G. Kolda. Newton-based optimization for Kullback–Leibler nonnegative tensor factorizations. *Optimization Methods and Software*, 30(5):1002–1029, April 2015. doi: 10.1080/10556788.2015.1009977.

[83] Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970. url: http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf.

[84] Trevor Hastie and Patrice Y. Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 13(1):54–65, February 1998. doi: 10.1214/ss/1028905973.

[85] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

[86] Koby Hayashi, Grey Ballard, Yujie Jiang, and Michael J. Tobia. Shared Memory Parallelization of MTTKRP for Dense Tensors, 2017. arXiv: 1708.08976v1.

[87] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2012. doi: 10.1109/cvpr.2012.6247848.

[88] Jun He, Laura Balzano, and Arthur Szlam. Online Robust Background Modeling via Alternating Grassmannian Optimization. In *Background Modeling and Foreground Detection for Video Surveillance*, pages 16–1–16–26. Chapman and Hall/CRC, July 2014. doi: 10.1201/b17223-20. url: https://www.taylorfrancis.com/books/9780429171116/chapters/10.1201/b17223-24.

[89] Jun He, Yue Zhang, Jiye Wang, Nan Zeng, and Hanyong Hao. Robust K-subspaces recovery with combinatorial initialization. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, December 2016. doi: 10.1109/bigdata.2016.7841021.

[90] Reinhard Heckel, Eirikur Agustsson, and Helmut Bölcskei. Neighborhood selection for thresholding-based subspace clustering. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2014. doi: 10.1109/icassp.2014.6854909.

[91] Reinhard Heckel and Helmut Bölcskei. Robust Subspace Clustering via Thresholding. *IEEE Transactions on Information Theory*, 61(11):6320–6342, November 2015. doi: 10.1109/tit.2015.2472520.

[92] Reinhard Heckel, Michael Tschannen, and Helmut Bölcskei. Subspace clustering of dimensionality-reduced data. In *2014 IEEE International Symposium on Information Theory*. IEEE, June 2014. doi: 10.1109/isit.2014.6875384.

[93] Frank L. Hitchcock. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematics and Physics*, 6(1-4):164–189, April 1927. doi: 10.1002/sapm192761164.

[94] David Hong, Laura Balzano, and Jeffrey A. Fessler. Towards a theoretical analysis of PCA for heteroscedastic data. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, September 2016. doi: 10.1109/allerton.2016.7852272.

[95] David Hong, Laura Balzano, and Jeffrey A. Fessler. Asymptotic performance of PCA for high-dimensional heteroscedastic data. *Journal of Multivariate Analysis*, 167:435–452, September 2018. doi: 10.1016/j.jmva.2018.06.002.

[96] David Hong, Jeffrey A. Fessler, and Laura Balzano. Optimally Weighted PCA for High-Dimensional Heteroscedastic Data, 2018. Submitted. arXiv: 1810.12862v2.

[97] David Hong, Tamara G. Kolda, and Jed A. Duersch. Generalized Canonical Polyadic Tensor Decomposition. *SIAM Review*, 2019. To appear. arXiv: 1808.07452v2.

[98] David Hong*, John Lipor*, Yan Shuo Tan, and Laura Balzano. Subspace Clustering using Ensembles of K-Subspaces, 2018. Submitted. (*equal contribution). arXiv: 1709.04744v2.

[99] David Hong, Robert P. Malinas, Jeffrey A. Fessler, and Laura Balzano. Learning Dictionary-Based Unions of Subspaces for Image Denoising. In *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, September 2018. doi: 10.23919/eusipco.2018.8553117.

[100] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2013. url: https://www.cambridge.org/us/academic/subjects/mathematics/algebra/matrix-analysis-2nd-edition?format=PB&isbn=9780521548236.

[101] Junzhou Huang and Tong Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, August 2010. doi: 10.1214/09-aos778.

[102] Kejun Huang, Nicholas D. Sidiropoulos, and Athanasios P. Liavas. A Flexible and Efficient Algorithmic Framework for Constrained Matrix and Tensor Factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, October 2016. doi: 10.1109/tsp.2016.2576427.

[103] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, March 1964. doi: 10.1214/aoms/1177703732.

[104] Peter J. Huber and Elvezio M. Ronchetti. *Robust Statistics*. John Wiley & Sons, Inc., January 2009. doi: 10.1002/9780470434697.

[105] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. ACM Press, 2009. doi: 10.1145/1553374.1553431.

[106] Rudolf Jaffé, Kaelin M. Cawley, and Youhei Yamashita. Applications of Excitation Emission Matrix Fluorescence with Parallel Factor Analysis (EEM-PARAFAC) in Assessing Environmental Dynamics of Natural Dissolved Organic Matter (DOM) in Aquatic Environments: A Review. In *Advances in the Physicochemical Characterization of Dissolved Organic Matter: Impact on Natural and Engineered Systems*, volume 1160 of *ACS Symposium Series*, pages 27–73. American Chemical Society, January 2014. doi: 10.1021/bk-2014-1160.ch003.

[107] Amin Jalali and Rebecca Willett. Subspace Clustering via Tangent Cones. In *Advances in Neural Information Processing Systems 30*, pages 6744–6753. Curran Associates, Inc., 2017. url: https://papers.nips.cc/paper/7251-subspace-clustering-via-tangent-cones.

[108] Jeroen J. Jansen, Huub C. J. Hoefsloot, Hans F. M. Boelens, Jan van der Greef, and Age K. Smilde. Analysis of longitudinal metabolomics data. *Bioinformatics*, 20(15):2438–2446, April 2004. doi: 10.1093/bioinformatics/bth268.

[109] Svante Janson. On concentration of probability. In *Contemporary Combinatorics*, volume 10 of *Bolyai Society Mathematical Studies*, pages 289—-301. Springer-Verlag Berlin Heidelberg, 2002. url: http://www2.math.uu.se/~svante/papers/sj126.pdf.

[110] Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, April 2001. doi: 10.1214/aos/1009210544.

[111] Iain M. Johnstone and Arthur Yu Lu. On Consistency and Sparsity for Principal Components Analysis in High Dimensions. *Journal of the American Statistical Association*, 104(486):682–693, June 2009. doi: 10.1198/jasa.2009.0121.

[112] Iain M. Johnstone and Debashis Paul. PCA in High Dimensions: An Orientation. *Proceedings of the IEEE*, 106(8):1277–1292, August 2018. doi: 10.1109/jproc.2018.2846730.

[113] Iain M. Johnstone and D. Michael Titterington. Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253, November 2009. doi: 10.1098/rsta.2009.0159.

[114] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002. doi: 10.1007/b98835.

[115] Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping. *The Annals of Applied Statistics*, 6(3):1095–1117, September 2012. doi: 10.1214/12-aoas549.

[116] Tamara Kolda and Brett Bader. The TOPHITS Model for Higher-order Web Link Analysis. In *Proceedings of the Fourth Workshop on Link Analysis, Counterterrorism and Security*, 2006. url: https://archive.siam.org/meetings/sdm06/workproceed/Link%20Analysis/21Tamara_Kolda_SIAMLACS.pdf.

[117] Tamara G. Kolda. Sparse versus scarce. Blog post, November 2017. Accessed May 2018. url: http://www.kolda.net/post/sparse-versus-scarce/.

[118] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, August 2009. doi: 10.1137/07070111x.

[119] Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The MNIST database of handwritten digits. Available online, 2016. url: http://yann.lecun.com/exdb/mnist/.

[120] Olivier Ledoit and Michael Wolf. Spectrum estimation: A unified framework for covariance matrix estimation and PCA in large dimensions. *Journal of Multivariate Analysis*, 139:360–384, July 2015. doi: 10.1016/j.jmva.2015.04.006.

[121] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. doi: 10.1038/44565.

[122] Daniel D. Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001. url: http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.

[123] Kuang-Chih Lee, Jeffrey Ho, and David J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, May 2005. doi: 10.1109/tpami.2005.92.

[124] Jeffrey T. Leek. Asymptotic conditional singular value decomposition for high-dimensional genomic data. *Biometrics*, 67(2):344–352, June 2010. doi: 10.1111/j.1541-0420.2010.01455.x.

[125] Friedrich Leisch. Bagged clustering. Technical Report 51, WU Vienna University of Economics and Business, 1999. url: http://epub.wu.ac.at/1272/.

[126] Paul Leopardi. Diameter bounds for equal area partitions of the unit sphere. *Electronic Transactions on Numerical Analysis*, 35:1–16, 2009. url: http://etna.mcs.kent.edu/volumes/2001-2010/vol35/abstract.php?vol=35&pages=1-16.

[127] Gilad Lerman, Michael B. McCoy, Joel A. Tropp, and Teng Zhang. Robust Computation of Linear Models by Convex Relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, September 2014. doi: 10.1007/s10208-014-9221-0.

[128] Jiajia Li, Yuchen Ma, Chenggang Yan, and Richard Vuduc. Optimizing Sparse Tensor Times Matrix on Multi-core and Many-Core Architectures. In *2016 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3)*. IEEE, November 2016. doi: 10.1109/ia3.2016.010.

[129] Shutao Li, Haitao Yin, and Leyuan Fang. Group-Sparse Representation With Dictionary Learning for Medical Image Denoising and Fusion. *IEEE Transactions on Biomedical Engineering*, 59(12):3450–3459, December 2012. doi: 10.1109/tbme.2012.2217493.

[130] John Lipor and Laura Balzano. Leveraging union of subspace structure to improve constrained clustering. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2130–2139, International Convention Centre, Sydney, Australia, August 2017. PMLR. url: http://proceedings.mlr.press/v70/lipor17a.html.

[131] John Lipor and Laura Balzano. Clustering quality metrics for subspace clustering. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018. url: http://web.cecs.pdx.edu/~lipor/Papers/lipor2018clustering.pdf.

[132] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010. url: https://icml.cc/Conferences/2010/papers/521.pdf.

[133] André Lourenço, Samuel Rota Bulò, Nicola Rebagliati, Ana Fred, Mário A. T. Figueiredo, and Marcello Pelillo. Probabilistic Evidence Accumulation for Clustering Ensembles. In *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods*, pages 58–67. SciTePress - Science and and Technology Publications, 2013. doi: 10.5220/0004267900580067.

[134] André Lourenço, Samuel Rota Bulò, Nicola Rebagliati, Ana L. N. Fred, Mário A. T. Figueiredo, and Marcello Pelillo. Probabilistic consensus clustering using evidence accumulation. *Machine Learning*, 98(1-2):331–357, April 2015. doi: 10.1007/s10994-013-5339-6.

[135] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and Efficient Subspace Segmentation via Least Squares Regression. In *Computer Vision – ECCV 2012*, pages 347–360. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-33786-4_26.

[136] Michael Lustig, David Donoho, and John M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. doi: 10.1002/mrm.21391.

[137] Michael Lustig, David L. Donoho, Juan M. Santos, and John M. Pauly. Compressed Sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, March 2008. doi: 10.1109/msp.2007.914728.

[138] V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, April 1967. doi: 10.1070/sm1967v001n04abeh001994.

[139] Koji Maruhashi, Fan Guo, and Christos Faloutsos. MultiAspectForensics: Pattern Mining on Large-Scale Heterogeneous Networks with Tensor Analysis. In *2011 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, July 2011. doi: 10.1109/asonam.2011.80.

[140] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, July 2010. doi: 10.1111/j.1467-9868.2010.00740.x.

[141] Xavier Mestre. Improved Estimation of Eigenvalues and Eigenvectors of Covariance Matrices Using Their Sample Estimates. *IEEE Transactions on Information Theory*, 54(11):5113–5129, November 2008. doi: 10.1109/tit.2008.929938.

[142] Behrouz Minaei-Bidgoli, Alexander Topchy, and William F. Punch. Ensembles of partitions via data resampling. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.* IEEE, 2004. doi: 10.1109/itcc.2004.1286629.

[143] Moshe Mishali and Yonina C. Eldar. Blind Multiband Signal Reconstruction: Compressed Sensing for Analog Signals. *IEEE Transactions on Signal Processing*, 57(3):993–1009, March 2009. doi: 10.1109/tsp.2009.2012791.

[144] Kathleen R. Murphy, Colin A. Stedmon, Daniel Graeber, and Rasmus Bro. Fluorescence spectroscopy and multi-way techniques. PARAFAC. *Analytical Methods*, 5(23):6557, 2013. doi: 10.1039/c3ay41160e.

[145] Raj Rao Nadakuditi. OptShrink: An Algorithm for Improved Low-Rank Signal Matrix Denoising by Optimal, Data-Driven Singular Value Shrinkage. *IEEE Transactions on Information Theory*, 60(5):3002–3018, May 2014. doi: 10.1109/tit.2014.2311661.

[146] Raj Rao Nadakuditi and Alan Edelman. The Polynomial Method for Random Matrices. *Foundations of Computational Mathematics*, 8(6):649–702, December 2007. doi: 10.1007/s10208-007-9013-x.

[147] Boaz Nadler. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, 36(6):2791–2817, December 2008. doi: 10.1214/08-aos618.

[148] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, Columbia University, February 1996. url: http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php.

[149] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96, Columbia University, February 1996. url: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[150] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002. url: https://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.

[151] Maximilian Nickel and Volker Tresp. Logistic Tensor Factorization for Multi-Relational Data, 2013. arXiv: 1306.2084v1.

[152] Tore Opsahl. Network 1: Facebook-like social network. Available online. Accessed June 2018. url: https://toreopsahl.com/datasets/#online_social_network.

[153] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, May 2009. doi: 10.1016/j.socnet.2009.02.002.

[154] Pentti Paatero. Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37(1):23–35, May 1997. doi: 10.1016/s0169-7439(96)00044-5.

[155] Pentti Paatero. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 38(2):223–242, October 1997. doi: 10.1016/s0169-7439(97)00031-2.

[156] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, June 1994. doi: 10.1002/env.3170050203.

[157] Guangming Pan. Strong convergence of the empirical distribution of eigenvalues of sample covariance matrices with a perturbation matrix. *Journal of Multivariate Analysis*, 101(6):1330–1338, July 2010. doi: 10.1016/j.jmva.2010.02.001.

[158] Jianing Pang, Behzad Sharif, Zhaoyang Fan, Xiaoming Bi, Reza Arsanjani, Daniel S. Berman, and Debiao Li. ECG and navigator-free four-dimensional whole-heart coronary MRA for simultaneous visualization of cardiac anatomy and function. *Magnetic Resonance in Medicine*, 72(5):1208–1217, September 2014. doi: 10.1002/mrm.25450.

[159] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932, May 2009. doi: 10.1002/asi.21015.

[160] Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. Streaming Pattern Discovery in Multiple Time-series. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 697–708. VLDB Endowment, 2005. url: http://dl.acm.org/citation.cfm?id=1083592.1083674.

[161] Evangelos E. Papalexakis, U Kang, Christos Faloutsos, Nicholas D. Sidiropoulos, and Abhay Harpale. Large Scale Tensor Decompositions: Algorithmic Developments and Applications. *IEEE Data Eng. Bull.*, 36(3):59–66, 2013. url: http://sites.computer.org/debull/A13sept/p59.pdf.

[162] Dohyung Park, Constantine Caramanis, and Sujay Sanghavi. Greedy Subspace Clustering. In *Advances in Neural Information Processing Systems 27*, pages 2753–2761. Curran Associates, Inc., 2014. url: https://papers.nips.cc/paper/5308-greedy-subspace-clustering.

[163] Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, 17(4):1617–1642, 2007. url: https://www.jstor.org/stable/24307692.

[164] Henrik Pedersen, Sebastian Kozerke, Steffen Ringgaard, Kay Nehrke, and Won Yong Kim. k-tPCA: Temporally constrainedk-tBLAST reconstruction using principal component analysis. *Magnetic Resonance in Medicine*, 62(3):706–716, September 2009. doi: 10.1002/mrm.22052.

[165] Daniel Percival. Theoretical properties of the overlapping groups lasso. *Electronic Journal of Statistics*, 6(0):269–288, 2012. doi: 10.1214/12-ejs672.

[166] Anh-Huy Phan, Petr Tichavsky, and Andrzej Cichocki. Fast Alternating LS Algorithms for High Order CANDECOMP/PARAFAC Tensor Factorizations. *IEEE Transactions on Signal Processing*, 61(19):4834–4846, October 2013. doi: 10.1109/tsp.2013.2269903.

[167] Anh-Huy Phan, Petr Tichavský, and Andrzej Cichocki. Low Complexity Damped Gauss–Newton Algorithms for CANDECOMP/PARAFAC. *SIAM Journal on Matrix Analysis and Applications*, 34(1):126–147, January 2013. doi: 10.1137/100808034.

[168] Chenlu Qiu, Namrata Vaswani, Brian Lois, and Leslie Hogben. Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise. *IEEE Transactions on Information Theory*, 60(8):5007–5039, August 2014. doi: 10.1109/tit.2014.2331344.

[169] Mostafa Rahmani and George K. Atia. Coherence Pursuit: Fast, Simple, and Robust Principal Component Analysis. *IEEE Transactions on Signal Processing*, 65(23):6260–6275, December 2017. doi: 10.1109/tsp.2017.2749215.

[170] Nikhil Rao, Robert Nowak, Christopher Cox, and Timothy Rogers. Classification With the Sparse Group Lasso. *IEEE Transactions on Signal Processing*, 64(2):448–463, January 2016. doi: 10.1109/tsp.2015.2488586.

[171] Nikhil Rao, Ben Recht, and Robert Nowak. Universal Measurement Bounds for Structured Sparse Signal Recovery. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 942–950, La Palma, Canary Islands, April 2012. PMLR. url: http://proceedings.mlr.press/v22/rao12.html.

[172] Saiprasad Ravishankar and Yoram Bresler. Learning Sparsifying Transforms. *IEEE Transactions on Signal Processing*, 61(5):1072–1086, March 2013. doi: 10.1109/tsp.2012.2226449.

[173] Steffen Rendle. Factorization Machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, 3(3):1–22, May 2012. doi: 10.1145/2168752.2168771.

[174] Matthew J. Reynolds, Alireza Doostan, and Gregory Beylkin. Randomized Alternating Least Squares for Canonical Tensor Decompositions: Application to A PDE With Random Data. *SIAM Journal on Scientific Computing*, 38(5):A2634–A2664, January 2016. doi: 10.1137/15m1042802.

[175] Justin K. Romberg, Hyeokho Choi, and Richard G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. *IEEE Transactions on Image Processing*, 10(7):1056–1068, July 2001. doi: 10.1109/83.931100.

[176] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, November 1992. doi: 10.1016/0167-2789(92)90242-f.

[177] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui. Realistic CT simulation using the 4D XCAT phantom. *Medical Physics*, 35(8):3800–3808, July 2008. doi: 10.1118/1.2955743.

[178] W. P. Segars, G. Sturgeon, S. Mendonca, Jason Grimes, and B. M. W. Tsui. 4D XCAT phantom for multimodality imaging research. *Medical Physics*, 37(9):4902–4915, August 2010. doi: 10.1118/1.3480985.

[179] Nitika Sharma and Kriti Saroha. A novel dimensionality reduction method for cancer dataset using PCA and feature ranking. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, August 2015. doi: 10.1109/icacci.2015.7275954.

[180] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*. ACM Press, 2005. doi: 10.1145/1102351.1102451.

[181] Jie Shen, Ping Li, and Huan Xu. Online Low-Rank Subspace Clustering by Basis Dictionary Pursuit. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 622–631, New York, New York, USA, June 2016. PMLR. url: http://proceedings.mlr.press/v48/shen16.html.

[182] Nino Shervashidze and Francis Bach. Learning the structure for structured sparsity. *IEEE Transactions on Signal Processing*, 63(18):4894–4902, September 2015. doi: 10.1109/tsp.2015.2446432.

[183] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, July 2017. doi: 10.1109/tsp.2017.2690524.

[184] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, 53(17):4777–4807, August 2008. doi: 10.1088/0031-9155/53/17/021.

[185] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A Sparse-Group Lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, April 2013. doi: 10.1080/10618600.2012.681250.

[186] Shaden Smith and George Karypis. Tensor-matrix products with a compressed sparse tensor. In *Proceedings of the 5th Workshop on Irregular Applications Architectures and Algorithms - IA3 '15*. ACM Press, 2015. doi: 10.1145/2833179.2833183.

[187] Mahdi Soltanolkotabi and Emmanuel J. Candés. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, August 2012. doi: 10.1214/12-aos1034.

[188] Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, April 2014. doi: 10.1214/13-aos1199.

[189] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003. url: https://www.aaai.org/Library/ICML/2003/icml03-094.php.

[190] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '10*. ACM Press, 2010. doi: 10.1145/1835804.1835895.

[191] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, September 2005. doi: 10.1073/pnas.0506580102.

[192] Ying Sun, Prabhu Babu, and Daniel P. Palomar. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, February 2017. doi: 10.1109/tsp.2016.2601299.

[193] Armeen Taeb, Parikshit Shah, and Venkat Chandrasekaran. False Discovery and Its Control in Low Rank Estimation, 2018. arXiv: 1810.08595v1.

[194] O. Tamuz, T. Mazeh, and S. Zucker. Correcting systematic effects in a large set of photometric light curves. *Monthly Notices of the Royal Astronomical Society*, 356(4):1466–1470, February 2005. doi: 10.1111/j.1365-2966.2004.08585.x.

[195] Michael E. Tipping and Christopher M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, August 1999. doi: 10.1111/1467-9868.00196.

[196] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992. doi: 10.1007/bf00129684.

[197] Alexander Topchy, Anil K. Jain, and William Punch. Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, December 2005. doi: 10.1109/tpami.2005.237.

[198] Roberto Tron and Rene Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2007. doi: 10.1109/cvpr.2007.382974.

[199] Joel A. Tropp. Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, October 2004. doi: 10.1109/tit.2004.834793.

[200] P. Tseng. Nearest q-Flat to m Points. *Journal of Optimization Theory and Applications*, 105(1):249–252, April 2000. doi: 10.1023/a:1004678431677.

[201] Kagan Tumer and Adrian K. Agogino. Ensemble clustering with voting active clusters. *Pattern Recognition Letters*, 29(14):1947–1953, October 2008. doi: 10.1016/j.patrec.2008.06.011.

[202] Madeleine Udell, Corinne Horn, Reza Zadeh, and Stephen Boyd. Generalized Low Rank Models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016. doi: 10.1561/2200000055.

[203] Namrata Vaswani and Han Guo. Correlated-PCA: Principal Components' Analysis when Data and Noise are Correlated. In *Advances in Neural Information Processing Systems 29*, pages 1768–1776. Curran Associates, Inc., 2016. url: https://papers.nips.cc/paper/6598-correlated-pca-principal-components-analysis-when-data-and-noise-are-correlated.

[204] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, September 2018. doi: 10.1017/9781108231596.

[205] René Vidal. Subspace Clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, March 2011. doi: 10.1109/msp.2010.939739.

[206] René Vidal and Paolo Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47–61, July 2014. doi: 10.1016/j.patrec.2013.08.006.

[207] René Vidal, Yi Ma, and S. S. Sastry. *Generalized Principal Component Analysis*. Springer New York, 2016. doi: 10.1007/978-0-387-87811-9.

[208] Gregory S. Wagner and Thomas J. Owens. Signal detection using multi-channel seismic data. *Bulletin of the Seismological Society of America*, 86(1A):221–231, 1996. url: https://pubs.geoscienceworld.org/ssa/bssa/article-abstract/86/1A/221/120051/signal-detection-using-multi-channel-seismic-data?redirectedFrom=fulltext.

[209] Xu Wang and Gilad Lerman. Fast Landmark Subspace Clustering, 2015. arXiv: 1510.08406v1.

[210] Yining Wang, Yu-Xiang Wang, and Aarti Singh. Graph Connectivity in Noisy Sparse Subspace Clustering. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 538–546, Cadiz, Spain, May 2016. PMLR. url: http://proceedings.mlr.press/v51/wang16b.html.

[211] Max Welling and Markus Weber. Positive tensor factorization. *Pattern Recognition Letters*, 22(12):1255–1261, October 2001. doi: 10.1016/s0167-8655(01)00070-8.

[212] Alex H. Williams, Tony Hyun Kim, Forea Wang, Saurabh Vyas, Stephen I. Ryu, Krishna V. Shenoy, Mark Schnitzer, Tamara G. Kolda, and Surya Ganguli. Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. *Neuron*, 98(6):1099–1115, June 2018. doi: 10.1016/j.neuron.2018.05.015.

[213] Thakshila Wimalajeewa, Yonina C. Eldar, and Pramod K. Varshney. Subspace Recovery From Structured Union of Subspaces. *IEEE Transactions on Information Theory*, 61(4):2101–2114, April 2015. doi: 10.1109/tit.2015.2403260.

[214] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via Outlier Pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, May 2012. doi: 10.1109/tit.2011.2173156.

[215] Jianfeng Yao, Shurong Zheng, and Zhidong Bai. *Large Sample Covariance Matrices and High-Dimensional Data Analysis*. Cambridge University Press, 2015. doi: 10.1017/cbo9781107588080.

[216] Chong You, Chun-Guang Li, Daniel P. Robinson, and René Vidal. Oracle Based Active Set Algorithm for Scalable Elastic Net Subspace Clustering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. doi: 10.1109/cvpr.2016.426.

[217] Chong You, Daniel P. Robinson, and René Vidal. Scalable Sparse Subspace Clustering by Orthogonal Matching Pursuit. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. doi: 10.1109/cvpr.2016.425.

[218] Gale Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6(1):49–53, February 1941. doi: 10.1007/bf02288574.

[219] H. Henry Yue and Masayuki Tomoyasu. Weighted principal component analysis and its applications to improve FDC performance. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. IEEE, 2004. doi: 10.1109/cdc.2004.1429421.

[220] Jinchun Zhan, Brian Lois, Han Guo, and Namrata Vaswani. Online (and Offline) Robust PCA: Novel Algorithms and Performance Guarantees. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1488–1496, Cadiz, Spain, May 2016. PMLR. url: http://proceedings.mlr.press/v51/zhan16.html.

[221] Anru Zhang, T. Tony Cai, and Yihong Wu. Heteroskedastic PCA: Algorithm, Optimality, and Applications, 2018. arXiv: 1810.08316v1.

[222] Dejiao Zhang and Laura Balzano. Global Convergence of a Grassmannian Gradient Descent Algorithm for Subspace Estimation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1460–1468, Cadiz, Spain, May 2016. PMLR. url: http://proceedings.mlr.press/v51/zhang16b.html.

[223] Qiang Zhang, Han Wang, Robert J. Plemmons, and V. Pau'l Pauca. Tensor methods for hyperspectral data analysis: a space object material identification study. *Journal of the Optical Society of America A*, 25(12):3001–3012, November 2008. doi: 10.1364/josaa.25.003001.

[224] Tao Zhang, John M. Pauly, Shreyas S. Vasanawala, and Michael Lustig. Coil compression for accelerated imaging with Cartesian sampling. *Magnetic Resonance in Medicine*, 69(2):571–582, April 2012. doi: 10.1002/mrm.24267.

[225] Teng Zhang, Arthur Szlam, and Gilad Lerman. Median K-Flats for hybrid linear modeling with many outliers. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, September 2009. doi: 10.1109/iccvw.2009.5457695.

[226] Teng Zhang, Arthur Szlam, Yi Wang, and Gilad Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, June 2012. doi: 10.1007/s11263-012-0535-6.