The Dissertation Committee for Megasthenis Asteris
certifies that this is the approved version of the following dissertation:

Quadratic Maximization Under Combinatorial Constraints and Related Applications

Committee:

Alexandros G. Dimakis, Supervisor

Constantine Caramanis

Eric Price

Sanjay Shakkottai

Sujay Sanghavi

Quadratic Maximization Under Combinatorial Constraints and Related Applications

by

Megasthenis Asteris, B.E.; M.S.E.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
December 2016

Quadratic Maximization Under Combinatorial Constraints and Related Applications

Publication No. _____

Megasthenis Asteris, Ph.D.
The University of Texas at Austin, 2016

Supervisor: Alexandros G. Dimakis

Motivated primarily by restricted variants of Principal Component Analysis (PCA), we study quadratic maximization problems subject to sparsity, nonnegativity and other combinatorial constraints. Intuitively, a key technical challenge is determining the support of the optimal solution. We develop a method that can surprisingly solve the maximization exactly when the argument matrix of the quadratic objective is positive semidefinite and has constant rank. Our approach relies on a hyper-spherical transformation of the low-rank space and has complexity that scales exponentially in the rank of the input, but polynomially in the ambient dimension. Extending these observations, we describe a simpler approximation algorithm based on exploring the low-rank space with an ϵ-net, drastically improving the dependence on the ambient dimension, implying a Polynomial Time Approximation Scheme (PTAS) for inputs with rank scaling up to logarithmically in the dimension or sufficiently

sharp spectral decay. We discuss extensions of our approach to jointly computing multiple principal components under combinatorial constraints, such as the problem of extracting multiple orthogonal nonnegative components, or sparse components with common or disjoint supports, and related approximate matrix factorization problems. We further extend our quadratic maximization framework to bilinear optimization problems and employ it in the context of specific applications, *e.g.*, to develop a provable approximation algorithm for the NP-hard problem of Bipartite Correlation Clustering (BCC).

Real datasets will typically produce covariance matrices that have full rank, rendering our algorithms not applicable. Our approach is to first obtain a low-rank approximation of the input data and subsequently solve the low-rank problem using our framework. Although this approach is not always suitable, from an optimization perspective it yields provable, data-dependent performance bounds that rely on the spectral decay of the input and the employed approximation technique. Interestingly, most real matrices can be well approximated by low-rank surrogates since the eigenvalues display a significant decay. Empirical evaluation shows that our algorithms have excellent performance and in many cases outperform the previous state of the art. Finally, utilizing our framework, we develop algorithms with interesting theoretical guarantees in the context of specific applications, such as approximate Orthogonal Nonnegative Matrix Factorization or Bipartite Correlation Clustering.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Identifying structure, extracting interpretable information, or more ambitiously generating knowledge from massive datasets is a central goal in data analytics. It is often synonymous to dimensionality reduction, *i.e.*, the process of determining a few degrees of freedom that capture most of the data variability: documents in a large and seemingly diverse text corpus may be coarsely summarized as combinations of few topics; users and items may be clustered into a small number of groups to build recommendation systems, and many more. A simplified representation of the data may be itself the end goal, for example to assist scientific discovery, or the means to improve the performance of machine learning and data mining algorithms in subsequent tasks.

Principal Component Analysis (PCA) is one of the most popular dimensionality reduction tools; it extracts orthogonal directions that capture most of the variability in the observed samples and reduces the dimensionality of the data by projecting it on the extracted directions. These directions coincide with the principal eigenvectors of the data covariance matrix $\mathbf{A}$. In particular, the leading principal component is the solution to

$$\text{(PCA)} \qquad \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}} \quad \mathbf{x}^\top \mathbf{A}\mathbf{x}, \tag{1.1}$$

where

$$\mathcal{X} = \Big\{ \mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1 \Big\}, \tag{1.2}$$

and d is the ambient dimension of the input data. The optimization problem (1.1) is not a convex problem since it involves *maximizing* a convex function over a convex set. Interestingly, it can be solved efficiently via the Singular Value Decomposition (SVD).

The extracted principal component aligns with the direction of maximum data variability. More formally, it can be used to optimally represent the input data among all one-dimensional subspaces under the ℓ_2 error. The quadratic objective (1.1) is referred to as the *explained variance*; it measures the variance of the projections of the input data-points along $\mathbf{x}$. Equivalently, it measures the variance preserved upon reducing the description of each multi-variate observation to a single variable, a new *feature*, formed as a linear combination of the d original variables. In turn, the extracted component effectively captures how the original variables *interact* with each other.

The principal component obtained through (1.1) is optimal with respect to the representation of the input data and can be computed efficiently, but may fall short in other aspects. From a statistical point of view, the goal is typically to recover the direction of maximum variability of the population from which the observed samples originate. If the ambient dimension of the data is comparable to or larger than the sample size, (1.1) will not recover the desired direction [78]. Similarly, from a machine learning perspective, although the

principal component optimally represents the observed data, the large number of parameters may result in overfit models, which generalize poorly to unseen data. Moreover, since the extracted feature is typically an arbitrary linear combination of all original variables, it may not be *interpretable*, *i.e.*, it may be difficult to attribute any physical interpretation. This is sometimes mentioned informally and is frequently not as important: if a computer vision classifier can correctly identify human faces, *why* it works is considered a secondary question. However, when machine learning is used for the scientific understanding of an underlying system, interpretability is perhaps the single most important property.

To address the aforementioned shortcomings, succinctly described as the effect of dimensionality, we often resort to incorporating prior knowledge or enforcing structure to the extracted component. For example, from a statistical standpoint, if the desired component is known to be sparse, *i.e.*, have a limited number of nonzero entries, then recovery is theoretically possible using fewer samples [142]. In machine learning, enforcing structure such as sparsity can result in simpler and more robust models with improved generalization properties. Finally, a new feature formed by linearly combining only a small number of variables can be easier to interpret. For example, under the common "bag-of-words" model, document vectors and eigenvectors are supported on words. A sparse component can be interpreted as a *topic* defined by a few relevant words, while the projection of a document on a principal component indicates how strong the specific topic is in that document.

To extract a principal component satisfying the desired structural properties, we can appropriately modify the feasible region (1.2) of the vanilla PCA problem (1.1). The additional constraints, however, render the corresponding optimization problem NP-hard, and hence intractable in general.

In this thesis, motivated primarily by restricted variants of PCA, we study quadratic maximization problems as (1.1) under sparsity, nonnegativity and other combinatorial constraints. We develop a method that can surprisingly solve (1.1) *exactly* when the argument matrix $\mathbf{A}$ of the quadratic objective is positive semidefinite and has constant rank. The approach relies on a hyper-spherical transformation of the low-rank space and has complexity that scales exponentially in the rank of the input, but polynomially in the ambient dimension. This novel approach allows us to handle constraints like integrality, sparsity and nonnegativity. Further, we describe a simpler *approximation* algorithm based on exploring the low-rank space with an ϵ-net, drastically improving the dependence on the ambient dimension, implying a Polynomial Time Approximation Scheme (PTAS) for rank scaling up to logarithmically in the dimension. We discuss extensions of our approach to jointly computing multiple principal components under combinatorial constraints, such as the problem of extracting multiple orthogonal nonnegative components, and sparse components with common or disjoint supports, and related approximate matrix factorization problems. We further adapt our quadratic maximization framework to bilinear optimization problems and employ it in the context of specific applications, *e.g.*, to develop an approximation algorithm for Bipartite

Correlation Clustering (BCC) (Chapter 7).

Our framework can solve quadratic problems when the involved matrix has constant rank. Of course, real datasets will typically produce covariance matrices that have full rank, rendering our algorithms not applicable. Our natural approach is to first obtain a low-rank approximation of the input data and subsequently solve the low-rank problem using our framework. For full rank matrices we establish approximation guarantees that depend on the decay of the eigenvalues: when a covariance matrix is close to low rank, our approximation error is very small. We obtain provable, data-dependent performance bounds relying on various low-rank approximation methods ranging from simple Singular Value Decomposition (SVD) to more sophisticated techniques. Interestingly, most real matrices can be well approximated by low-rank surrogates since the eigenvalues display a significant decay. Empirically, our preliminary work shows that the algorithms obtained from our framework have excellent performance and in many cases outperform the previous state of the art, at least with respect to the objective function of the optimization problem. Further, in the context of specific applications, even though the involved matrices may have full rank, we can exploit properties of the problem to obtain interesting theoretical guarantees; in Chapter 4, for example, we describe an algorithm for computing an approximate orthogonal nonnegative matrix factorization problem on arbitrary input and obtain provable guarantees, while in Chapter 7, we exploit the "density" of the BCC problem to obtain a PTAS for the problem of maximizing the number of agreements between the input

graph and the output clustering.

It is noteworthy, however, that a low-rank approximation step is not always suitable. For example, in the context of sparse PCA and in particular under the spiked covariance model [78] studied in statistics, such low-rank approximation step nullifies the original motivation of seeking a sparse principal component.

We will now show that some quadratic optimization problems of the form of (1.1) can be hard even when the input argument $\mathbf{A}$ has rank equal to 1. It is a common misconception that optimization problems lying in a space with low intrinsic dimension are inherently easy, especially if we can afford computational complexity that scales exponentially in the intrinsic dimension. As an example, consider the SubsetSum problem: given a multiset of d integers, we ask whether there exists a non-empty subset whose elements sum to zero. For instance, for the set $\{-9, -2, 1, 4, 5\}$, the answer would be affirmative since the nonempty subset $\{-9, 4, 5\}$ has a zero sum. SubsetSum can be solved utilizing the optimization (1.1) with a rank-1 matrix argument over sparsity and integrality constraints as follows. Given a multiset of d integers, we arbitrarily arrange its elements in an d-dimensional vector $\mathbf{y}$, construct $\mathbf{A} = -\mathbf{y}\mathbf{y}^\top$. Then, for $s = 1, \ldots, d$, we invoke an oracle to solve $\max \mathbf{x}^\top \mathbf{A}\mathbf{x}$ over all $\mathbf{x} \in \{0, 1\}^n$ such that $\|\mathbf{x}\|_0 \leq s$. SubsetSum has a solution if and only if the maximum value of the quadratic is zero for at least one value of the parameter $s = 1, \ldots, d$. From this simple reduction it immediately follows that the quadratic maximization is NP-hard in general, even for a rank-1 input

argument.[1]

Similar hardness observations arise in Nonnegative Matrix Factorization (NMF), which is broadly related to the Orthogonal NMF problem considered in Chapter 4. The *nonnegative rank* of an $m \times d$ nonnegative matrix $\mathbf{M}$, denoted by $\text{rank}_+(\mathbf{M})$, is the smallest integer s for which there exist nonnegative matrices $\mathbf{U} \in \mathbb{R}^{m \times k}$ and $\mathbf{V} \in \mathbb{R}^{k \times n}$ such that $\mathbf{M} = \mathbf{UV}$. Determining whether $\text{rank}_+(\mathbf{M}) = \text{rank}(\mathbf{M})$ is NP-hard for arbitrary input [139], while questions pertaining to the complexity of computing the nonnegative rank remain unanswered to date, even for input matrices with fixed rank $r > 2$. In [61], Gillis defined the *restricted nonnegative rank* (RNR) of a nonnegative matrix, a variant of the aforementioned problem with the additional restriction $\text{rank}(\mathbf{U}) = \text{rank}(\mathbf{M})$ (see [61], Sec 3.3), and shows that computing the RNR is NP-hard even if the input matrix has constant rank $r \geq 4$.

1.1 Outline

We describe our contributions in the context of specific applications.

Chapter 2 We consider the problem of computing the leading nonnegative and sparse principal component. Sparse PCA is the most extensively studied variant of PCA in machine learning and statistics; the extracted component

[1] SubsetSum is considered one of the "easiest" NP-Complete problems; an FPTAS was provided by [86]. For our purposes, however, it is noteworthy that the quadratic maximization cannot be solved exactly in polynomial time even for a rank-1 input argument.

is restricted to have at most s nonzero entries for a given parameter s. Restricting those nonzero entries to be positive, yields directions of maximum data variability which depend only on positive interactions of the observed variables. In the form of an optimization, we seek

$$\underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}} \quad \mathbf{x}^\top \mathbf{A}\mathbf{x},$$

where

$$\text{(NNSPCA)} \qquad \mathcal{X} = \Big\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_0 = s,\ \mathbf{x} \geq 0\Big\}. \tag{1.3}$$

The ℓ_0-norm constraint restricts the cardinality of the support of $\mathbf{x}$. As expected, both the sparsity and the nonnegativity constraints individually render the problem NP-hard and hence computationally intractable in general.

In our preliminary work [20, 18], we showed that if the matrix $\mathbf{A}$ is positive semidefinite and has *constant* rank r then (1.3) can be solved *exactly* in polynomial time (but time exponential in the rank r). Our algorithm relied on an hyperspherical transformation of the low-dimensional range of $\mathbf{A}$, which allowed as to efficiently collect a large but tractable number of candidate supports for the sparse principal component. Once the candidate supports are collected, the optimal solution can be determined via exhaustive search. In this chapter, we describe our algorithm in the context of nonnegative and sparse PCA [16]; nonnegativity constraints incur additional technical challenges. For the case where the input matrix has full rank, we develop approximation guarantees that depend on the decay of the eigenvalues; for example,

for a power-law (or faster) decay, we can provably approximate the optimal objective within any desired accuracy factor. Finally, we describe an alternative approximate algorithm for the low-rank problem using ϵ-nets, which drastically improves the computational complexity with respect to the ambient dimension of our data. As a corollary, our approximation algorithm implies a PTAS for (1.3) when the rank of the input scales at most logarithmically in the ambient dimension.

Chapter 3 In many cases, one is not interested in finding only the first principal component, but rather the leading k, where k is the reduced dimension where the data will be projected. Contrary to the single-component problem, there has been very limited work on computing multiple sparse components. The scarcity is partially attributed to conventional wisdom stemming from PCA: multiple components can be computed one by one, repeatedly solving the single-component sparse PCA problem and *deflating* [103] the input data to remove information captured by previously extracted components.

We extend the algorithmic ideas and approximation guarantees of the previous chapter to the problem of jointly extracting multiple sparse components. Our algorithm can extract orthogonal sparse components that either share a common support or have pairwise disjoint supports. Here, we focus on the latter case and present an algorithm for sparse PCA that jointly optimizes multiple disjoint components. The extracted features capture variance that lies within a multiplicative factor arbitrarily close to 1 from the optimal. Our

algorithm is combinatorial and computes the desired components by solving multiple instances of the bipartite maximum weight matching problem. We evaluate our algorithm on real datasets and empirically demonstrate that in many cases it outperforms existing, deflation-based approaches.

Chapter 4 We describe an algorithm to jointly compute multiple orthogonal nonnegative components. Our developments are similar to those in Chapter 3, substituting appropriate subroutines to handle the nonnegativity constraints. Further, utilizing our NNPCA algorithm, we develop an algorithm for Orthogonal Nonnegative Matrix Factorization (ONMF); given a nonnegative matrix, ONMF aims to approximate it as the product of two k-dimensional nonnegative factors, one of which has orthonormal columns. This yields potentially useful data representations as superposition of disjoint parts. Effectively, we present a new ONMF algorithm with provable approximation guarantees. For any constant dimension k, we obtain an additive EPTAS on the relative Frobenious error without any assumptions on the input beyond it being nonnegative. We evaluate our algorithms on several real and synthetic datasets and show that their performance matches or outperforms the state of the art.

Chapter 5 We derail from our algorithmic developments and revisit the single-component sparse PCA problem. While sparsity results in succinct models, it may fall short in capturing the true interactions in a physical system, especially when the number of samples is limited. Incorporating higher-order

prior structural information can improve interpretability of the extracted component. In this chapter, we introduce the idea of (sparse) structure captured by an underlying graph. In particular, motivated by a problem in neuroscience, we introduce a novel problem referred to as Path PCA: given a covariance matrix and a separate directed graph defined on the same variables, we seek a principal component whose support coincides with some path on the graph. The covariance matrix $\mathbf{A}$ and the graph are independent inputs in this problem: the matrix captures data correlations, while the graph is a mathematical tool to efficiently describe interpretable supports.

From a statistical perspective, information on the underlying network may potentially reduce the number of observations required to recover the population principal component. We introduce a simple and natural network and analyze the the canonical estimator which optimally exploits the prior knowledge by solving a non-convex quadratic maximization on the empirical covariance (as in (1.1)) under the spiked covariance model. We show that the additional side information can reduce the statistical complexity of recovering the true principal component.

Chapter 6 Up to this point, we have considered only quadratic maximization problems of the form (1.1) where the argument matrix was positive semidefinite. We describe extensions of our algorithm to approximately solve

bilinear maximization problems of the form

$$\underset{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}} \quad \mathbf{x}^\top \mathbf{B}\mathbf{y}, \tag{1.4}$$

over combinatorial constraints and an arbitrary argument matrix $\mathbf{B}$. In Chapter 6, we present our developments in the context of (an approximate formulation [148] of) sparse Canonical Correlation Analysis (CCA): given two sets of variables derived from a common set of samples, CCA seeks linear combinations of a small number of variables in each set, such that the induced *canonical* variables are maximally correlated. In contrast to the majority of existing approaches, our algorithm administers precise control on the sparsity of either or both extracted canonical vectors. Similar to the symmetric quadratic case, the algorithm comes with theoretical data-dependent global approximation guarantees, that hinge on the spectrum of the input data, it can be straightforwardly adapted to other constrained variants of CCA, enforcing additional structure beyond sparsity. We empirically evaluate the proposed scheme.

Chapter 7 We study the NP-hard problem of Bipartite Correlation Clustering (BCC): given a complete bipartite graph G with $+$ and $-$ edges, find a vertex clustering that maximizes the number of *agreements*, *i.e.*, the number of all $+$ edges within clusters plus all $-$ edges cut across clusters.

We present a novel approximation algorithm for k-BCC, a variant of BCC with an upper bound k on the number of clusters. Our algorithm outputs a k-clustering that provably achieves a number of agreements within a

multiplicative $(1-\delta)$-factor from the optimal, for any desired accuracy δ. It relies on solving a combinatorially constrained bilinear maximization on the bi-adjacency matrix $\mathbf{B}$ of G, *i.e.*, an optimization of the form

$$\underset{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}}{\operatorname{argmax}} \quad \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{A}\mathbf{Y}\Big), \tag{1.5}$$

where the feasible region comprises all binary valid cluster assignment matrices. The algorithm runs in time exponential in s and $1/\delta$, but linear in the size of the input. Further, we show that, in the (unconstrained) BCC setting, an $(1-\delta)$-approximation can be achieved by $O(\delta^{-1})$ clusters regardless of the size of the graph. In turn, our k-BCC algorithm implies an Efficient PTAS for the BCC objective of maximizing agreements.

Chapter 2

Nonnegative Sparse PCA

We introduce a novel algorithm to compute nonnegative sparse principal components of positive semidefinite (PSD) matrices. Our algorithm comes with approximation guarantees contingent on the spectral profile of the input data: the sharper the spectrum decay, the better the quality of the approximation.

If the eigenvalues decay like any asymptotically vanishing function, we can approximate nonnegative sparse PCA within any accuracy ϵ in time polynomial in the matrix dimension n and desired sparsity s, but not in $1/\epsilon$. Further, we obtain a data dependent bound that is computed by executing an algorithm on a given data set. This bound is significantly tighter than *a-priori* bounds and can be used to show that for all tested datasets our algorithm is provably within $40\% - 90\%$ from the unknown optimum.

Our algorithm is combinatorial and explores a subspace defined by the

Chapter 2 is based on material from Reference [16]: Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis, *"Nonnegative Sparse PCA With Provable Guarantees"*, Proceedings of the 31st International Conference on Machine Learning (ICML), pp. 1728–1736, 2014. The author of this dissertation is the lead author of [16], and contributed to the conception of the research problem, the theoretical and analytical developments, the experimental design and implementation, the writing of the manuscript and its revisions.

leading eigenvectors of $\mathbf{A}$. We test our scheme on several data sets, showing that it matches or outperforms the previous state of the art.

2.1 Introduction

Given a data matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$ comprising m zero-mean vectors on n features, the first principal component (PC) is

$$\arg \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A} \mathbf{x}, \tag{2.1}$$

where $\mathbf{A} = {}^1\!/\!_m \cdot \mathbf{S}\mathbf{S}^\top$ is the $n \times n$ positive semidefinite (PSD) empirical covariance matrix. Subsequent PCs can be computed after $\mathbf{A}$ has been appropriately deflated to remove the first eigenvector. PCA is arguably the workhorse of high dimensional data analysis and achieves dimensionality reduction by computing the directions of maximum variance. Typically, all n features affect positively or negatively these directions resulting in dense PCs, which explain the largest possible data variance, but are often not interpretable.

Enforcing nonnegativity on the computed principal components can aid interpretability. This is particularly true in applications where features interact only in an additive manner. For instance, in bioinformatics, chemical concentrations are nonnegative [87], or the expression level of genes is typically attributed to positive or negative influences of those genes, but not both [23]. Here, enforcing nonnegativity, in conjunction with sparsity on the computed components can assist the discovery of local patterns in the data. In computer vision, where features may coincide with non negatively valued image pixels,

nonnegative sparse PCA pertains to the extraction of the most informative image parts [94]. In other applications, nonnegative weights admit a meaningful probabilistic interpretation.

Sparsity emerges as an additional desirable trait of the computed components because it further helps interpretability [160, 50], even independently of nonnegativity. From a machine learning perspective, enforcing sparsity serves as an unsupervised feature selection method: the active coordinates in an optimal l_0-norm constrained PC should correspond to the most informative subset of features. Although nonnegativity inherently promotes sparsity, an explicit sparsity constraint enables precise control on the number of selected features.

Nonnegative Sparse PC Nonnegativity and sparsity can be directly enforced on the principal component optimization by adding constraints to (2.1). The s-sparse nonnegative principal component of $\mathbf{A}$ is

$$\mathbf{x}_\star = \arg\max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}\mathbf{x}, \tag{2.2}$$

where

$$\mathbb{S}_s^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1, \|\mathbf{x}\|_0 \leq s, \mathbf{x} \geq 0\}, \tag{2.3}$$

for a desired sparsity parameter $s \in [n]$.

The problem of computing the first eigenvector (2.1) is easily solvable, but with the additional sparsity and nonnegativity constraints problem (2.2)

becomes computationally intractable. The cardinality constraint alone renders sparse PCA NP-hard [110]. Even if the l_0-norm constraint is dropped, we show that problem (2.2) remains computationally intractable by reducing it to checking matrix copositivity, a well known co-NP complete decision problem [112, 116]. Therefore, each of the constraints $\mathbf{x} \geq \mathbf{0}$ and $\|\mathbf{x}\|_0 \leq s$ individually makes the problem intractable.

Our Contributions We introduce a novel algorithm for approximating the nonnegative s-sparse principal component with provable approximation guarantees.

Given any PSD matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, sparsity parameter s, and accuracy parameter $r \in [n]$, our algorithm outputs a nonnegative, s-sparse, unit norm vector $\mathbf{x}_r$ that achieves at least ρ_r fraction of the maximum objective value in (2.2), *i.e.*,

$$\mathbf{x}_r^\top \mathbf{A} \mathbf{x}_r \;\geq\; \rho_r \cdot \mathbf{x}_\star^\top \mathbf{A} \mathbf{x}_\star, \tag{2.4}$$

where

$$\rho_r \;\geq\; \max\left\{ \frac{s}{2n},\; \frac{1}{1 + 2\frac{n}{s}\lambda_{r+1}/\lambda_1} \right\}. \tag{2.5}$$

Here, λ_i is the ith largest eigenvalue of $\mathbf{A}$, and the accuracy parameter r specifies the rank of the approximation used and controls the running time. Specifically, our algorithm runs in time $O(n^r s^r + n^{r+1})$. As can be seen our result depends on the spectral profile of $\mathbf{A}$: the faster the eigenvalue decay, the tighter the approximation.

Near-Linear Time Approximation Our algorithm has a running time $O(n^r s^r + n^{r+1})$, which in the linear sparsity regime can be as high as $O(n^{2r})$. This can be non-practical for large data sets, even if we set the rank parameter r to be two or three. We present a modification of our algorithm that can provably approximate the result of the first in near-linear time. Specifically, for any desired accuracy $\epsilon \in (0, 1]$ it computes a nonnegative, s-sparse, unit norm vector $\widehat{\mathbf{x}}_r$ such that

$$\widehat{\mathbf{x}}_r^\top \mathbf{A} \widehat{\mathbf{x}}_r \;\geq\; (1 - \epsilon) \cdot \rho_r \cdot \mathbf{x}_\star{}^\top \mathbf{A} \mathbf{x}_\star, \tag{2.6}$$

where ρ_r is as described in (2.5). We show that the running time of our approximate algorithm is $O(\epsilon^{-r} \cdot n \log n)$, which is near-linear in n for any fixed accuracy parameters r and ϵ.

Our approximation theorem has several implications.

Exact Solution for Low-Rank Input Observe that if the matrix $\mathbf{A}$ has rank r, our algorithm returns the optimal s-sparse PC for any target sparsity s. The same holds in the case of the rank-r update matrix $\mathbf{A} = \sigma\mathbf{I} + \mathbf{C}$, with $\text{rank}(\mathbf{C}) = r$ and arbitrary constant σ, since the algorithm can be equivalently applied on $\mathbf{C}$.

PTAS for any Spectral Decay Consider the linear sparsity regime $s = c \cdot n$ and assume that the eigenvalues follow a decay law $\lambda_i \leq \lambda_1 \cdot f(i)$ for any decay function $f(i)$ which vanishes: $f(i) \to 0$ as $i \to \infty$. Special

cases include power law decay $f(i) = 1/i^{\alpha}$ or even very slow decay functions like $f(i) = 1/\log\log i$. For all these cases, we can solve nonnegative sparse PCA for any desired accuracy ϵ in time polynomial in n and s, but not in $1/\epsilon$. Therefore, we obtain a polynomial-time approximation scheme (PTAS) for any spectral decay behavior.

Computable Data-Dependent Bounds In addition to these theoretical guarantees, our method yields a data dependent upper bound on the maximum value of (2.2), that can be computed by running our algorithm. As it can be seen in Fig. 2.4-2.6, the obtained upper bound, combined with our achievable point, sandwiches the unknown optimum within a narrow region. Using this upper bound we are able to show that our solutions are within $40 - 90\%$ from the optimal in all the datasets that we examine. To the best of our knowledge, this framework of data dependent bounds has not been considered in the previous literature.

2.2 Related Work

There is a substantial volume of work on sparse PCA, spanning a rich variety of approaches: from early heuristics in [80], to the LASSO based techniques in [81], the elastic net l_1-regression in [160], a greedy branch-and-bound technique in [110], or semidefinite programming approaches [48, 159, 51]. This line of work does not consider or enforce nonnegativity constraints.

When nonnegative components are desired, fundamentally different ap-

proaches have been used. Nonnegative matrix factorization [94] and its sparse variants [71, 87] fall within that scope: data is expressed as (sparse) nonnegative linear combinations of (sparse) nonnegative parts. These approaches are interested in finding a lower dimensionality representation of the data that reveals latent structure and minimizes a reconstruction error, but are not explicitly concerned with the statistical significance of individual output vectors.

Nonnegativity as an additional constraint on (sparse) PCA first appeared in [157]. The authors suggested a coordinate-descent scheme that jointly computes a set of nonnegative sparse principal components, maximizing the cumulative explained variance. An l_1-penalty promotes sparsity of computed components on average, but not on each component individually. A second convex penalty is incorporated to favor orthogonal components.

Similar convex optimization approaches for nonnegative PCA have been subsequently proposed in the literature. In [5] for instance, the authors suggest an alternating maximization scheme for the computation of the first nonnegative PC, allowing the incorporation of known structural dependencies.

A competitive algorithm for nonnegative sparse PCA was established in [127], with the development of a framework stemming from Expectation-Maximization (EM) for a probabilistic generative model of PCA. The proposed algorithm, which enforces hard sparsity, or nonnegativity, or both constraints simultaneously, computes the first approximate PC in $O(n^2)$, *i.e.*, time quadratic in the number of features.

To the best of our knowledge, no prior works provide provable approximation guarantees for the nonnegative sparse PCA optimization problem. Further, no data dependent upper bounds have been present in the previous literature.

Differences from SPCA work Our work is closely related to [85, 20, 113] that introduced the ideas of solving low-rank quadratic combinatorial optimization problems on low-rank PSD matrices using hyperspectral transformations. Such transformations are called spannograms and follow a similar architecture. Here, we extend the spannogram framework to nonnegative sparse PCA. The most important technical issue compared to [20, 113] is introducing nonnegativity constraints in spannogram algorithms.

To understand how this changes the problem, notice that in the original sparse PCA problem without nonnegativity constraints, if the support is known, the optimal principal component supported on that set can be easily found. However, under nonnegativity constraints, the problem is hard even if the optimal support is known. This is the fundamental technical problem that we address in this chapter. We show that if the involved subspace is low-dimensional, it is possible to solve this problem.

2.3 Algorithm Overview

Given an $n \times n$ PSD matrix $\mathbf{A}$, the desired sparsity s, and an accuracy parameter $r \in [n]$, our algorithm computes a *nonnegative, s-sparse, unit norm*

vector $\mathbf{x}_r$ approximating the nonnegative, s-sparse PC of $\mathbf{A}$. We begin with a high-level description of the main steps of the algorithm.

Step 1. *Compute $\mathbf{A}_r$, the rank-r approximation of $\mathbf{A}$.* We compute $\mathbf{A}_r$, the best rank-r approximation of $\mathbf{A}$, zeroing out the $n-r$ trailing eigenvalues of $\mathbf{A}$, that is,

$$\mathbf{A}_r = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top,$$

where λ_i is the ith largest eigenvalue of $\mathbf{A}$ and $\mathbf{u}_i$ the corresponding eigenvector.

Step 2. *Compute $\mathcal{S}_r$, a set of $O(n^r)$ candidate supports.* Enumerating the $\binom{n}{s}$ possible supports for s-sparse vectors in $\mathbb{R}^n$ is computationally intractable. Using our *Spannogram* technique described in Section 2.5, we efficiently determine a collection $\mathcal{S}_r$ of support sets, with cardinality $|\mathcal{S}_r| \leq 2^r \binom{n+1}{r}$, that provably contains the support of the nonnegative, s-sparse PC of $\mathbf{A}_r$.

Step 3. *Compute $\mathcal{X}_r$, a set of candidate solutions.* For each candidate support set $\mathcal{I} \in \mathcal{S}_r$, we compute a candidate solution $\mathbf{x}$ supported only in $\mathcal{I}$:

$$\arg \max_{\substack{\|\mathbf{x}\|_2=1, \mathbf{x}\geq \mathbf{0}, \\ \operatorname{supp}(\mathbf{x})\subseteq \mathcal{I}}} \mathbf{x}^\top \mathbf{A}_r \mathbf{x}. \tag{2.7}$$

The constant rank of $\mathbf{A}_r$ is essential in solving (2.7): the constrained quadratic maximization is in general NP-hard, even for a given support.

Step 4. *Output the best candidate solution in $\mathcal{X}_r$, i.e.*, the candidate that maximizes the quadratic form.

Algorithm 2.1 Spannogram Nonnegative Sparse PCA

input $\mathbf{A}$ – $n \times n$ real PSD matrix
s – parameter controlling target sparsity. Takes values in $[n]$.
r – parameter controlling the rank of approximation of $\mathbf{A}$ to be used.

output $\mathbf{x}_r$ – n-dimensional real vector with at most s nonzero entries. See Thm. 2.1 for guarantees.

1: $\mathbf{U}, \mathbf{\Lambda} \leftarrow \texttt{svd}(\mathbf{A}, r)$
2: $\mathbf{V} = \mathbf{U}\mathbf{\Lambda}^{1/2}$ $\{\mathbf{A}_r = \mathbf{V}\mathbf{V}^\top\}$
3: $\mathcal{S}_r \leftarrow \text{Spannogram}(\mathbf{V}, s)$ {Algo. 2.2}
4: $\mathcal{X}_r \leftarrow \{\}$ $\{|\mathcal{S}_r| \leq O(n^r)\}$
5: **for all** $\mathcal{I} \in \mathcal{S}_r$ **do**
6: $\quad \mathbf{c}^{(\mathcal{I})} \leftarrow \arg\max_{\substack{\|\mathbf{c}\|_2=1 \\ \mathbf{V}_\mathcal{I}\mathbf{c} \geq \mathbf{0}}} \|(\mathbf{V}_\mathcal{I}\mathbf{c})\|_2^2$ {Sec. 2.6}
7: $\quad \mathbf{x}_\mathcal{I}^{(\mathcal{I})} \leftarrow |\mathbf{V}_\mathcal{I}\mathbf{c}|/\|\mathbf{V}_\mathcal{I}\mathbf{c}\|, \quad \mathbf{x}_{\mathcal{I}^c}^{(\mathcal{I})} \leftarrow \mathbf{0}$
8: $\quad \mathcal{X}_r \leftarrow \mathcal{X}_r \cup \{\mathbf{x}^{(\mathcal{I})}\}$
9: **end for** $\{|\mathcal{X}_r| \leq |\mathcal{S}_r|\}$
10: $\mathbf{x}_r \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}_r} \mathbf{x}^\top \mathbf{A}_r \mathbf{x}$

If multiple components are desired, the procedure is repeated after an appropriate deflation has been applied on $\mathbf{A}_r$ [103]. The steps are formally presented in Algorithm 2.1. A detailed description is the subject of subsequent sections.

2.3.1 Approximation Guarantees

Instead of the nonnegative, s-sparse, principal component $\mathbf{x}_\star$ of $\mathbf{A}$, which attains the optimal value $\text{OPT} = \mathbf{x}_\star{}^\top \mathbf{A}\mathbf{x}_\star$, our algorithm outputs a nonnegative, s-sparse, unit norm vector $\mathbf{x}_r$. We measure the quality of $\mathbf{x}_r$ as a surrogate of $\mathbf{x}_\star$ by the approximation factor $\mathbf{x}_r{}^\top \mathbf{A}\mathbf{x}_r/\text{OPT}$. Clearly, the approximation factor takes values in $(0, 1]$, with higher values implying tighter

approximation.

Theorem 2.1. *For any $n \times n$ PSD matrix $\mathbf{A}$, sparsity parameter s, and accuracy parameter $r \in [n]$, Alg. 2.1 outputs a nonnegative, s-sparse, unit norm vector $\mathbf{x}_r$ such that*

$$\mathbf{x}_r{}^\top \mathbf{A} \mathbf{x}_r \;\geq\; \rho_r \cdot \mathbf{x}_\star{}^\top \mathbf{A} \mathbf{x}_\star,$$

where

$$\rho_r \;\geq\; \max\left\{ \frac{s}{2n},\; \frac{1}{1 + 2\frac{n}{s}\lambda_{r+1}/\lambda_1} \right\},$$

in time $O(n^{r+1} + n^r s^r)$.

The approximation guarantee of Theorem 2.1 relies on establishing connections among the eigenvalues of $\mathbf{A}$, and the quadratic forms $\mathbf{x}_r{}^\top \mathbf{A} \mathbf{x}_r$ and $\mathbf{x}_r{}^\top \mathbf{A}_r \mathbf{x}_r$. The proof can be found in the supplemental material. The complexity of Algorithm 2.1 follows upon its detailed description.

2.4 Proposed Scheme

Our algorithm approximates the nonnegative, s-sparse PC of a PSD matrix $\mathbf{A}$ by computing the corresponding PC of $\mathbf{A}_r$, a rank-r surrogate of the input argument $\mathbf{A}$:

$$\mathbf{A}_r = \sum_{i=1}^{r} \mathbf{v}_i \mathbf{v}_i{}^\top = \mathbf{V}\mathbf{V}^\top, \tag{2.8}$$

where $\mathbf{v}_i = \sqrt{\lambda_i}\mathbf{u}_i$ is the scaled eigenvector corresponding to the ith largest eigenvalue of $\mathbf{A}$, and $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_r] \in \mathbb{R}^{n \times r}$. In this section, we delve into the

details of our algorithmic developments and describe how the low rank of $\mathbf{A}_r$ unlocks the computation of the desired PC.

2.4.1 Rank-1: A simple case

We begin with the rank-1 case because, besides its motivational simplicity, it is a fundamental component of the algorithmic developments for the rank-r case.

In the rank-1 case, $\mathbf{V}$ reduces to a single vector in $\mathbb{R}^n$ and $\mathbf{x}_1$, the nonnegative s-sparse PC of $\mathbf{A}_1$, is the solution to

$$\max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_1 \mathbf{x} = \max_{\mathbf{x}\in\mathbb{S}_s^n} \left(\mathbf{v}^\top \mathbf{x}\right)^2. \tag{2.9}$$

That is, $\mathbf{x}_1$ is the nonnegative, s-sparse, unit length vector that maximizes $(\mathbf{v}^\top \mathbf{x})^2$. Let $\mathcal{I} = \text{supp}(\mathbf{x}_1)$, $|\mathcal{I}| \leq s$, be the unknown support of $\mathbf{x}_1$. Then, $(\mathbf{v}^\top \mathbf{x})^2 = (\sum_{i\in\mathcal{I}} v_i \cdot x_i)^2$. Since $\mathbf{x}_1 \geq \mathbf{0}$, it should not be hard to see that the active entries of $\mathbf{x}_1$ must correspond to nonnegative or nonpositive entries of $\mathbf{v}$, but not a combination of both. In other words, $\mathbf{v}_\mathcal{I}$, the entries of $\mathbf{v}$ indexed by $\mathcal{I}$, must satisfy $\mathbf{v}_\mathcal{I} \geq \mathbf{0}$ or $\mathbf{v}_\mathcal{I} \leq \mathbf{0}$. In either case, by the Cauchy-Schwarz inequality,

$$\left(\mathbf{v}^\top \mathbf{x}\right)^2 = \left(\mathbf{v}_\mathcal{I}{}^\top \mathbf{x}_\mathcal{I}\right)^2 \leq \|\mathbf{v}_\mathcal{I}\|_2^2 \|\mathbf{x}_\mathcal{I}\|_2^2, = \|\mathbf{v}_\mathcal{I}\|_2^2. \tag{2.10}$$

Equality in (2.10) can always be achieved by setting $\mathbf{x}_\mathcal{I} = \mathbf{v}_\mathcal{I}/\|\mathbf{v}_\mathcal{I}\|_2$ if $\mathbf{v}_\mathcal{I} \geq \mathbf{0}$, and $\mathbf{x}_\mathcal{I} = -\mathbf{v}_\mathcal{I}/\|\mathbf{v}_\mathcal{I}\|_2$ if $\mathbf{v}_\mathcal{I} \leq \mathbf{0}$. The support of the optimal solution $\mathbf{x}_1$ is the set $\mathcal{I}$ for which $\|\mathbf{v}_\mathcal{I}\|_2^2$ in (2.10) is maximized under the restriction that the entries of $\mathbf{v}_\mathcal{I}$ do not have mixed signs.

Def. 2.4.1. *Let* $\mathcal{I}_s^+(\mathbf{v})$, $1 \leq s \leq n$ *denote the set of indices of the (at most)* s *largest nonnegative entries in* $\mathbf{v} \in \mathbb{R}^n$.

Proposition 2.4.1. *Let* $\mathbf{x}_1$ *be the solution to problem* (2.9). *Then, supp*$(\mathbf{x}_1) \in \mathcal{S}_1 = \{\mathcal{I}_s^+(\mathbf{v}),\ \mathcal{I}_s^+(-\mathbf{v})\}$.

The collection $\mathcal{S}_1$ and the associated candidate vectors via (2.10) are constructed in $O(n)$. The solution $\mathbf{x}_1$ is the candidate that maximizes the quadratic.

2.4.2 Rank-r case

In the rank-r case, $\mathbf{x}_r$, the nonnegative, s-sparse PC of $\mathbf{A}_r$ is the solution to the following problem:

$$\max_{\mathbf{x} \in \mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r \mathbf{x} = \max_{\mathbf{x} \in \mathbb{S}_s^n} \|\mathbf{V}^\top \mathbf{x}\|_2^2. \tag{2.11}$$

Consider an auxiliary vector $\mathbf{c} \in \mathbb{R}^r$, with $\|\mathbf{c}\|_2 = 1$. From the Cauchy-Schwarz inequality,

$$\|\mathbf{V}^\top \mathbf{x}\|_2^2 = \|\mathbf{c}\|_2^2 \|\mathbf{V}^\top \mathbf{x}\|_2^2 \geq \left|\mathbf{c}^\top \left(\mathbf{V}^\top \mathbf{x}\right)\right|^2. \tag{2.12}$$

Equality in (2.12) is achieved if and only if $\mathbf{c}$ is colinear to $\mathbf{V}^\top \mathbf{x}$. Since $\mathbf{c}$ spans the entire unit sphere, such a $\mathbf{c}$ exists for every $\mathbf{x}$, yielding an alternative description for the objective function in (2.11):

$$\|\mathbf{V}^\top \mathbf{x}\|_2^2 = \max_{\mathbf{c} \in \mathbb{S}^r} \left|(\mathbf{V}\mathbf{c})^\top \mathbf{x}\right|^2, \tag{2.13}$$

where $\mathbb{S}^r = \{\mathbf{c} \in \mathbb{R}^r : \|\mathbf{c}\|_2 = 1\}$ is the r-dimensional unit sphere. The maximization in (2.11) becomes

$$\begin{aligned}\max_{\mathbf{x}\in\mathbb{S}_s^n} \|\mathbf{V}^\top \mathbf{x}\|_2^2 &= \max_{\mathbf{x}\in\mathbb{S}_s^n} \max_{\mathbf{c}\in\mathbb{S}^r,} |(\mathbf{V}\mathbf{c})^\top \mathbf{x}|^2 \\ &= \max_{\mathbf{c}\in\mathbb{S}^r,\ \mathbf{x}\in\mathbb{S}_s^n} \max |(\mathbf{V}\mathbf{c})^\top \mathbf{x}|^2. \end{aligned} \tag{2.14}$$

The Set of Candidate Supports A first key observation is that for fixed $\mathbf{c}$, the product $(\mathbf{V}\mathbf{c})$ is a vector in $\mathbb{R}^n$. Maximizing $|(\mathbf{V}\mathbf{c})^\top \mathbf{x}|^2$ over all vectors $\mathbf{x} \in \mathbb{S}_s^n$ is a rank-1 instance of the optimization problem, as in (2.9). Let $(\mathbf{c}_r, \mathbf{x}_r)$ be the optimal solution of (2.11). By Proposition 2.4.1, the support of $\mathbf{x}_r$ coincides with either $\mathcal{I}_s^+(\mathbf{V}\mathbf{c}_r)$ or $\mathcal{I}_s^+(-\mathbf{V}\mathbf{c}_r)$. Hence, we can safely claim that $\mathrm{supp}(\mathbf{x}_r)$ appears in

$$\mathcal{S}_r = \bigcup_{\mathbf{c}\in\mathbb{S}^r} \left\{\mathcal{I}_s^+(\mathbf{V}\mathbf{c})\right\}. \tag{2.15}$$

Naively, one might think that $\mathcal{S}_r$ can contain as many as $\binom{n}{s}$ distinct support sets. In Section 2.5, we show that $|\mathcal{S}_r| \leq 2^r\binom{n+1}{r}$ and present our Spannogram technique (Alg. 2.2) for efficiently constructing $\mathcal{S}_r$ in $O(n^{r+1})$. Each support in $\mathcal{S}_r$ corresponds to a candidate principal component.

Solving for a Given Support We seek a pair $(\mathbf{x}, \mathbf{c})$ that maximizes (2.14) under the additional constraint that $\mathbf{x}$ is supported only on a given set $\mathcal{I}$. By the Cauchy-Schwarz inequality, the objective in (2.14) satisfies

$$|(\mathbf{V}\mathbf{c})^\top \mathbf{x}|^2 = |(\mathbf{V}_\mathcal{I}\mathbf{c})^\top \mathbf{x}_\mathcal{I}|^2 \leq \|(\mathbf{V}_\mathcal{I}\mathbf{c})\|_2^2, \tag{2.16}$$

where $\mathbf{V}_{\mathcal{I}}$ is the matrix formed by the rows of $\mathbf{V}$ indexed by $\mathcal{I}$. Equality in (2.16) is achieved if and only if $\mathbf{x}_{\mathcal{I}}$ is colinear to $\mathbf{V}_{\mathcal{I}}\mathbf{c}$. However, it is not achievable for arbitrary $\mathbf{c}$, as $\mathbf{x}_{\mathcal{I}}$ must be nonnegative. From Proposition 2.4.1, we infer that $\mathbf{x}$ being supported in $\mathcal{I}$ implies that all entries of $\mathbf{V}_{\mathcal{I}}\mathbf{c}$ have the same sign. Further, whenever the last condition holds, a nonnegative $\mathbf{x}_{\mathcal{I}}$ colinear to $\mathbf{V}_{\mathcal{I}}\mathbf{c}$ exists and equality in (2.16) can be achieved. Under the additional constraint that $\operatorname{supp}(\mathbf{x}) = \mathcal{I} \in \mathcal{S}_r$, the maximization in (2.14) becomes

$$\max_{\mathbf{c}\in\mathbb{S}^r} \max_{\substack{\mathbf{x}\in\mathbb{S}^n_s \\ \operatorname{supp}(\mathbf{x})\subseteq\mathcal{I}}} |(\mathbf{V}\mathbf{c})^\top \mathbf{x}|^2 = \max_{\substack{\mathbf{c}\in\mathbb{S}^r \\ \mathbf{V}_{\mathcal{I}}\mathbf{c}\geq\mathbf{0}}} \|(\mathbf{V}_{\mathcal{I}}\mathbf{c})\|_2^2. \tag{2.17}$$

The constraint $\mathbf{V}_{\mathcal{I}}\mathbf{c} \geq \mathbf{0}$ in (2.17), is equivalent to requiring that all entries in $\mathbf{V}_{\mathcal{I}}\mathbf{c}$ have the same sign, since $\mathbf{c}$ and $-\mathbf{c}$ achieve the same objective value.

The optimization problem in (2.17) is NP-hard. In fact, it encompasses the original nonnegative PCA problem as a special case. Here, however, the constant dimension $r = \Theta(1)$ of the unknown variable $\mathbf{c}$ permits otherwise intractable operations. In Section 2.6, we outline an $O(s^r)$ algorithm for solving this constrained quadratic maximization.

The Algorithm The previous discussion suggests a two-step algorithm for solving the rank-r optimization problem in (2.11). First, run the Spannogram algorithm to construct $\mathcal{S}_r$, the collection of $O(n^r)$ candidate supports for $\mathbf{x}_r$, in $O(n^{r+1})$. For each $\mathcal{I} \in \mathcal{S}_r$, solve (2.17) in $O(s^r)$ to obtain a candidate solution $\mathbf{x}^{(\mathcal{I})}$ supported on $\mathcal{I}$. Output the candidate solution that max-

imizes the quadratic $\mathbf{x}^\top \mathbf{A}_r \mathbf{x}$. Efficiently combining the previous steps yields an $O(n^{r+1} + n^r s^r)$ procedure for approximating the nonnegative sparse PC, outlined in Alg. 2.1.

2.5 The Nonnegative Spannogram

In this section, we describe how to construct $\mathcal{S}_r$, the collection of candidate supports, defined in (2.15) as

$$\mathcal{S}_r = \bigcup_{\mathbf{c} \in \mathbb{S}^r} \left\{ \mathcal{I}_s^+(\mathbf{V}\mathbf{c}) \right\},$$

for a given $\mathbf{V} \in \mathbb{R}^{n \times r}$. $\mathcal{S}_r$ comprises all support sets induced by vectors in the range of $\mathbf{V}$. The *Spannogram* of $\mathbf{V}$ is a *visualization* of its range, and a valuable tool in efficiently collecting those supports.

2.5.1 Constructing $\mathcal{S}_2$

We describe the $r = 2$ case, the simplest nontrivial case, to facilitate a gentle exposure to the Spannogram technique. The core ideas generalize to arbitrary r and a detailed description is provided in the supplemental material.

Spherical Variables Up to scaling, all vectors $\mathbf{v}$ in the range of $\mathbf{V} \in \mathbb{R}^{n \times 2}$, $\mathcal{R}(\mathbf{V})$, can be written as $\mathbf{v} = \mathbf{V}\mathbf{c}$ for some $\mathbf{c} \in \mathbb{R}^2 : \|\mathbf{c}\| = 1$. We introduce a variable $\phi \in \Phi = (-\pi/2, \pi/2]$, and set $\mathbf{c}$ to be the following function of ϕ:

$$\mathbf{c}(\phi) = \begin{bmatrix} \sin(\phi) & \cos(\phi) \end{bmatrix}^\top.$$

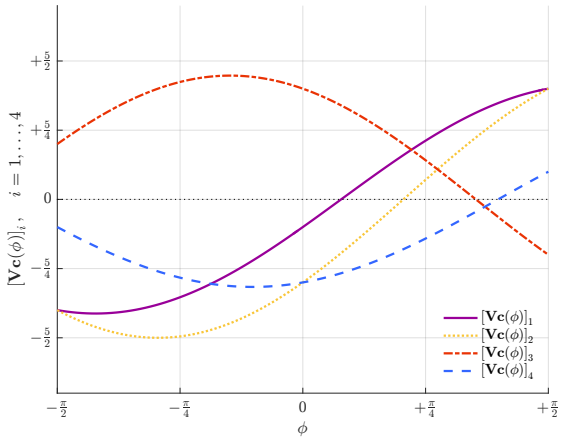


Figure 2.1: Example of rank-2 spannogram. The figure depicts the spannogram of an arbitrary rank-2 matrix $\mathbf{V} \in \mathbb{R}^{4\times 2}$. Each curve is associated with (generated by) a row of $\mathbf{V}$. At a point ϕ of the horizontal axis, the values of the curves correspond to the entries of a vector $\mathbf{v}(\phi) = \mathbf{Vc}(\phi)$ in the range of $\mathbf{V}$ and vice versa.

The range of $\mathbf{V}$, $\mathcal{R}(\mathbf{V}) = \{\pm\mathbf{v}(\phi) = \pm\mathbf{Vc}(\phi), \phi \in \Phi\}$, is also a function of ϕ, and in turn $\mathcal{S}_2$ can be expressed as

$$\mathcal{S}_2 = \bigcup_{\phi\in\Phi} \left\{\mathcal{I}_s^+(\mathbf{v}(\phi)),\ \mathcal{I}_s^+(-\mathbf{v}(\phi))\right\}.$$

Spannogram The ith entry of $\mathbf{v}(\phi)$ is a continuous function of ϕ generated by the ith row of $\mathbf{V}$: $[\mathbf{v}(\phi)]_i = \mathbf{V}_{i,1}\sin(\phi) + \mathbf{V}_{i,2}\cos(\phi)$. Fig. 2.1 depicts the functions corresponding to the rows of an arbitrary matrix $\mathbf{V} \in \mathbb{R}^{4\times 2}$. We call this a *spannogram*, because at each ϕ, the values of the curves coincide with the entries of a vector in the range of $\mathbf{V}$. A key observation is that the sorting of the curves at some ϕ is locally invariant for most points in Φ. In fact, due to the continuity of the curves, as we move along the ϕ-axis, the set $\mathcal{I}_s^+(\mathbf{v}(\phi))$

can only change at points where a curve intersects with i) another curve, or ii) the zero axis; a change in either the sign of a curve or the relative order of two curves is necessary, although not sufficient, for $\mathcal{I}_s^+(\mathbf{v}(\phi))$ to change.

Appending a zero $(n+1)$th row to $\mathbf{V}$, the two aforementioned conditions can be merged into one: $\mathcal{I}_s^+(\mathbf{v}(\phi))$ can change only at the points where two of the $n+1$ curves intersect. Finding the unique intersection point of two curves $[\mathbf{v}(\phi)]_i$ and $[\mathbf{v}(\phi)]_j$ for all pairs $\{i, j\}$ is the key to discovering all possible candidate support sets. There are exactly $\binom{n+1}{2}$ such points partitioning Φ into $\binom{n+1}{2}+1$ intervals within which the set of largest s nonnegative entries of $\mathbf{v}(\phi)$ and $-\mathbf{v}(\phi)$ are invariant.

Constructing $\mathcal{S}_2$ The point ϕ_{ij} where the ith and jth curves intersect, corresponds to a vector $\mathbf{v}(\phi_{ij}) \in \mathcal{R}(\mathbf{V})$ whose ith and jth entries are equal. To find it, it suffices to compute a $\mathbf{c} \neq \mathbf{0}$ such that $(\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{V}\mathbf{c} = 0$, *i.e.*, a unit norm vector $\mathbf{c}_{ij}$ in the one-dimensional nullspace of $(\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{V}$. Then, $\mathbf{v}(\phi_{ij}) = \mathbf{V}\mathbf{c}_{ij}$.

We compute the candidate support $\mathcal{I}_s^+(\mathbf{v}(\phi_{ij}))$ at the intersection. Assuming for simplicity that only the ith and jth curves intersect at ϕ_{ij}, the sorting of all curves is unchanged in a small neighborhood of ϕ_{ij}, except the ith and jth curves whose order changes over ϕ_{ij}. If both the ith and jth entries of $\mathbf{v}(\phi_{ij})$ or none of them is included in the s largest nonnegative entries, then the set $\mathcal{I}_s^+(\mathbf{v}(\phi))$ in the two intervals incident to ϕ_{ij} is identical. Otherwise, the ith and jth curve occupy the s^{th} and $(s+1)^{\text{th}}$ order at ϕ_{ij}, and the change in their relative order implies that one leaves and one joins the set of s largest

nonnegative curves at ϕ_{ij}. The support sets associated with the two adjacent intervals differ only in one element (one contains index i and the other contains index j instear), while the remaining $s-1$ common indices correspond to the $s-1$ largest curves at the intersection point ϕ_{ij}. We include both in $\mathcal{S}_2$ and repeat the above procedure for $\mathcal{I}_s^+(-\mathbf{v}(\phi_{ij}))$.

Each pairwise intersection is computed in $O(1)$ and the at most 4 associated candidate supports in $O(n)$. In total, the collection $\mathcal{S}_2$ comprises $|\mathcal{S}_2| \leq 4\binom{n+1}{2} = O(n^2)$ candidate supports and can be constructed in $O(n^3)$.

The generalized Spannogram algorithm for constructing $\mathcal{S}_r$ runs in $O(n^{r+1})$ and is formally presented in Alg. 2.2. A detailed description is provided in the supplemental material.

2.6 Quadratic Maximization over Hyper-Spherical Cap

Each support set $\mathcal{I}$ in $\mathcal{S}_r$ yields a candidate nonnegative, s-sparse PC, which can be obtained by solving (2.17), a quadratic maximization over the intersection of halfspaces and the unit sphere:

$$\mathbf{c}_\star = \arg\max_{\substack{\mathbf{c}\in\mathbb{S}^r \\ \mathbf{R}\mathbf{c}\geq 0}} \mathbf{c}^\top \mathbf{Q}\mathbf{c}, \qquad (\mathrm{P}_r)$$

where $\mathbf{Q} = \mathbf{V}_\mathcal{I}^\top \mathbf{V}_\mathcal{I}$ is a $r \times r$ matrix and $\mathbf{R}$ is a $s \times r$ matrix. Problem (P_r) is NP-hard: for $\mathbf{Q}$ PSD and $\mathbf{R} = \mathbf{I}_{r\times r}$, it reduces to the original problem (2.2). Here, however, we are interested in the case where the dimension r is a constant. We

Algorithm 2.2 Spannogram algorithm for constructing $\mathcal{S}_r$

input $\mathbf{V}$ – $n \times r$ real matrix.
s – parameter controlling target sparsity; takes values in $[n]$.
output $\mathcal{S}_r$ – Collection of candidate support sets.

$\mathcal{S}_r \leftarrow \{\}$ {Set of candidate supports in $\mathcal{R}(\mathbf{V})$}
$\widehat{\mathbf{V}} \leftarrow \begin{bmatrix}\mathbf{V}^\top \ \mathbf{0}_r\end{bmatrix}^\top$ {Append zero row; $\widehat{\mathbf{V}} \in \mathbb{R}^{n+1 \times r}$}
for all $\binom{n+1}{r}$ sets $\{i_1, \ldots, i_r\} \subseteq [n+1]$ **do**
  $\mathbf{c} \leftarrow \text{Nullspace}\left(\begin{bmatrix}\mathbf{e}_{i_1}^\top - \mathbf{e}_{i_2}^\top \\ \vdots \\ \mathbf{e}_{i_1}^\top - \mathbf{e}_{i_r}^\top\end{bmatrix}\widehat{\mathbf{V}} \in \mathbb{R}^{r-1 \times r}\right)$
  for $\alpha \in \{+1, -1\}$ **do**
    $\mathcal{I} \leftarrow \mathcal{I}_s^+(\alpha \mathbf{V}\mathbf{c})$ {Entries $\geq$ than the s^{th} one}
    if $|\mathcal{I}| \leq s$ **then**
      $\mathcal{S}_r \leftarrow \mathcal{S}_r \cup \{\mathcal{I}\}$ {No ambiguity}
    else
      $\mathcal{A} \leftarrow \{i_1, \ldots, i_r\} \backslash \{n+1\}$ {Ambiguous set}
      $\mathcal{T} \leftarrow \mathcal{I} \backslash \{\{i_1, \ldots, i_r\} \cup \{n+1\}\}$
      $r \leftarrow s - |\mathcal{T}|$
      $\widehat{r} \leftarrow |\mathcal{A}|$
      for all $\binom{\widehat{r}}{r}$ r-subsets $\mathcal{A}^{(r)} \subseteq \mathcal{A}\}$ **do**
        $\widehat{\mathcal{I}} \leftarrow \mathcal{T} \cup \mathcal{A}^{(r)}$
        $\mathcal{S}_r \leftarrow \mathcal{S}_r \cup \{\widehat{\mathcal{I}}\}$ {$\leq 2^r$ new candidates}
      end for
    end if
  end for
end for

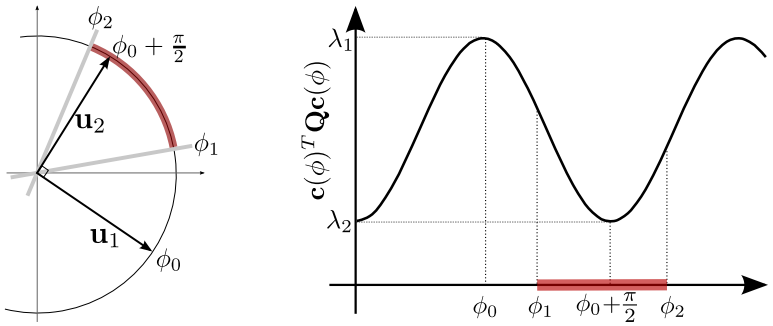
outline an $O(s^r)$ algorithm, *i.e.*, polynomial in the number of linear constraints, for solving (P_r). A detailed proof is available in the supplemental material.

The objective of (P_r) is maximized by $\mathbf{u}_1 \in \mathbb{R}^r$, the leading eigenvector of $\mathbf{Q}$. If $\mathbf{u}_1$ or $-\mathbf{u}_1$ is feasible, *i.e.*, if it satisfies all linear constraints, then $\mathbf{c}_\star = \pm\mathbf{u}_1$. It can be shown that if none of $\pm\mathbf{u}_1$ is feasible, at least one of the s linear

constraints is active at the optimal solution $\mathbf{c}_\star$, that is, there exists $1 \leq i \leq s$ such that $\mathbf{R}_{i,:}\mathbf{c}_\star = 0$.

Fig. 2.2 depicts an example for $r = 2$. The leading eigenvector of $\mathbf{Q}$ lies outside the feasible region, an arc of the unit-circle in the intersection of s halfspaces. The optimal solution coincides with one of the two endpoints of the feasible region, where a linear inequality is active, motivating a simple algorithm for solving (P_2): *i*) for each linear inequality determine a unit length point where the inequality becomes active, and *ii*) output the point that is feasible and maximizes the objective.

Back to the general (P_r) problem, if a linear inequality $\mathbf{R}_{i,:}\mathbf{c} \geq 0$ for some $i \in [s]$ is enforced with equality, the modified problem can be written as



(a) Feasible ϕ set. (b) $Q(\phi) = \mathbf{c}(\phi)^\top \mathbf{Q}_2 \mathbf{c}(\phi)$.

Figure 2.2: Visualization of an instance of (P_2); a rank-2 quadratic maximization over a hyperspherical cap. The point ϕ_0, corresponding to the leading eigenvector $\mathbf{u}_1$ of the argument matrix $\mathbf{Q} \in \mathbb{S}^2$, lies outside the feasible region, denoted by a highlighted red arc in Fig. 2.2(a). Fig. 2.2(b) depicts the quadratic objective function $\mathbf{c}(\phi)^\top \mathbf{Q}\mathbf{c}(\phi)$ as a function of ϕ. The maximum value of the constrained maximization is attained at ϕ_1, an endpoint of the feasible ϕ-interval. The feasible interval has length at most equal to π.

a quadratic maximization in the form of (P_r), with dimension reduced to $r-1$ and $s-1$ linear constraints. This observation suggests a recursive algorithm for solving (P_r): If $\pm\mathbf{u}_1$ is feasible, it is also the optimal solution. Otherwise, for $i = 1, \ldots, s$, set the ith inequality constraint active, solve recursively, and collect candidate solutions. Finally, output the candidate that maximizes the objective. The $O(s^r)$ recursive algorithm is formally presented in the supplemental material.

2.7 A Near-Linear Time Algorithm for NN-SPCA

Alg. 2.1 approximates the nonnegative, s-sparse PC of a PSD matrix $\mathbf{A}$ by solving the nonnegative sparse PCA problem exactly on $\mathbf{A}_r$, the best rank-r approximation of $\mathbf{A}$. Albeit polynomial in n, the running time of Alg. 2.1 can be impractical even for moderate values of n.

Instead of pursuing the exact solution to the low-rank nonnegative sparse PCA problem $\max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r \mathbf{x}$, we can compute an approximate solution in near-linear time, with performance arbitrarily close to optimal. The suggested procedure is outlined in Algorithm 2.3, and a detailed discussion is provided in the supplemental material. Alg. 2.3 relies on randomly sampling points from the range of $\mathbf{A}_r$ and efficiently solving rank-1 instances of the nonnegative sparse PCA problem as described in Section 2.4.1.

Theorem 2.2. *For any $n \times n$ PSD matrix $\mathbf{A}$, sparsity parameter s, and accuracy parameters $r \in [n]$ and $\epsilon \in (0, 1]$, Alg. 2.3 outputs a nonnegative, s-sparse,*

Algorithm 2.3 Near-Linear Time Algorithm for Nonnegative Sparse PCA

input $\mathbf{A}$ – $n \times n$ real PSD matrix
s – parameter controlling the target sparsity, *i.e.*, the number of nonzero entries in the output.
r – parameter controlling the rank of the approximation of $\mathbf{A}$ to be used.
ϵ – Accuracy parameter in $(0, 1)$.

output $\hat{\mathbf{x}}_r$ – n-dimensional real vector with at most s nonzero entries. See Thm 2.2 for guarantees.

$\mathbf{U}, \mathbf{\Lambda} \leftarrow \texttt{svd}(\mathbf{A}, r)$
$\mathbf{V} \leftarrow \mathbf{U}\mathbf{\Lambda}^{1/2}$ $\{\mathbf{A}_r = \mathbf{V}\mathbf{V}^\top\}$
$\mathcal{X}_r \leftarrow \{\}$
for $i = 1, \ldots, O(\epsilon^{-r} \cdot \log n)$ **do**
  $\mathbf{c} \leftarrow \texttt{randn}(r, 1)$
  $\mathbf{a} \leftarrow \mathbf{V}\mathbf{c}/\|\mathbf{c}\|_2$
  $\mathbf{x} \leftarrow \texttt{rank1solver}(\mathbf{a})$ {Section 2.4.1}
  $\mathcal{X}_r \leftarrow \mathcal{X}_r \cup \{\mathbf{x}\}$
end for
$\widehat{\mathbf{x}}_r \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}_r} \|\mathbf{V}^\top \mathbf{x}\|_2^2$

unit norm vector $\widehat{\mathbf{x}}_r$ *such that*

$$\widehat{\mathbf{x}}_r^\top \mathbf{A} \widehat{\mathbf{x}}_r \;\geq\; (1-\epsilon) \cdot \rho_r \cdot \mathbf{x}_\star{}^\top \mathbf{A} \mathbf{x}_\star,$$

with probability at least $1 - 1/n$*, in time* $O(\epsilon^{-r} \cdot n \log n)$ *plus* $\mathrm{T}_{\mathsf{EIG}}(\mathbf{A}, r)$*, i.e., the time required to compute the* r *leading eigenvectors of* $\mathbf{A}$*.*

2.8 Experiments

We empirically evaluate the performance of our algorithm on various datasets and compare it to the EM algorithm[1] for sparse and nonnegative PCA of [127] which is known to outperform previous algorithms.

CBCL Face Dataset The CBCL face image dataset [130], with 2429 gray scale images of size 19×19 pixels, has been used in the performance evaluation of both the NSPCA [157] and EM [127] algorithms.

Fig. 2.3 depicts samples from the dataset, as well as six orthogonal, nonnegative, s-sparse components ($s = 40$) successively computed by i) Alg. 2.3 configured with parameters $r = 3$ and $\epsilon = 0.1$, and ii) the EM algorithm. Features active in one component are removed from the dataset prior to computing subsequent PCs to ensure orthogonality. Fig. 2.3 reveals the ability of nonnegative sparse PCA to extract significant parts.

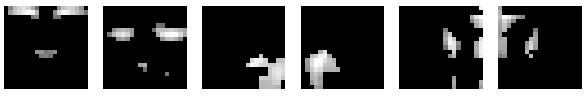
In Fig. 2.4, we plot the variance explained by the computed approximate nonnegative, s-sparse PC (normalized by the leading eigenvalue) versus

[1] Matlab implementation available by the author.

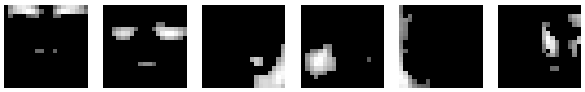
the sparsity parameter s. Alg. 2.3 for $r = 3$ and $\epsilon = 0.1$, and the EM algorithm exhibit nearly identical performance. For this dataset, we also compute the leading component using the NSPCA algorithm of [157]. Note that NSPCA does not allow for a precise control of the sparsity of its output; an appropriate sparsity penalty β was determined via binary search for each target sparsity s. We plot the explained variance only for those values of s for which a s-sparse component was successfully extracted. Finally, note that both the EM and NSPCA algorithms are randomly initialized. All depicted values are the best results over multiple random restarts.

Our theory allows us to obtain provable approximation guarantees: based on Theorem 2.2 and the output of Alg. 2.3, we compute a data dependent upper bound on the maximum variance, which provably lies in the shaded area. For instance, for $s = 180$, the extracted component explains at

(a) Example images from the CBCL Face dataset [130]



(b) Output of Alg. 2.3 ($r = 3, \epsilon = 0.1$).



(c) Output of EM algorithm.

Figure 2.3: Visualization of the nonnegative sparse principal components extracted from the CBCL Face image dataset [130]. The figure depicts (a) example images from the dataset, as well as the six leading orthogonal, nonnegative, s-sparse PCs for $s = 40$ extracted by (b) Alg. 2.3 configured with parameters $r = 3$ and $\epsilon = 0.1$, and (c) the EM algorithm of [127]. The components were extracted one by one. The variables selected by a component were removed from the dataset prior to extracting subsequent components.

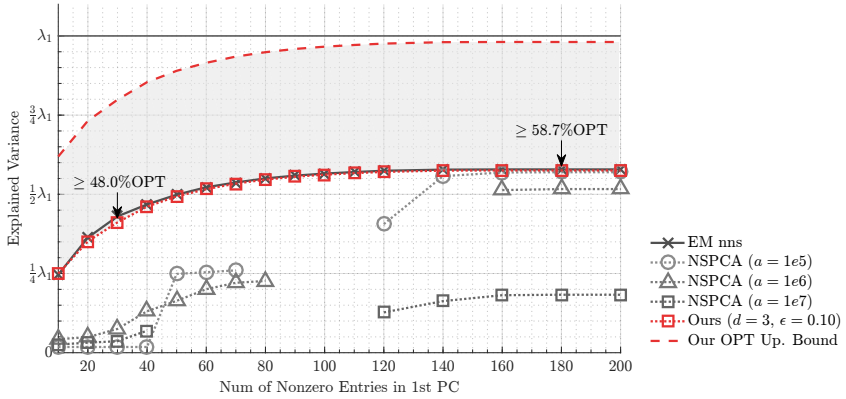


Figure 2.4: Explained variance achieved by the top s-sparse nonnegative component on the CBCL Face image dataset [130] as a function of the sparsity parameter s. We plot the variance explained by the approximate nonnegative, s-sparse PC computed by various algorithms as a function of the number s of nonzero entries. Our theory yields a provable data dependent approximation guarantee: for any value of s, the true unknown optimum provably lies in the shaded area.

least 58% of the variance explained by the true nonnegative, s-sparse PC. The quality of the bound depends on the accuracy parameters r and ϵ, and the eigenvalue decay of the empirical covariance matrix of the data. There exist datasets on which our algorithm provably achieves 70% or even 90% of the optimal.

Leukemia Dataset The Leukemia dataset [10] contains 72 samples, each consisting of expression values for 12582 probe sets. The dataset was used in the evaluation of [127]. In Fig. 2.5, we plot the normalized variance explained by the computed nonnegative, s-sparse PC versus the sparsity parameter s. For low values of s, Alg. 2.3 outperforms the EM algorithm in terms

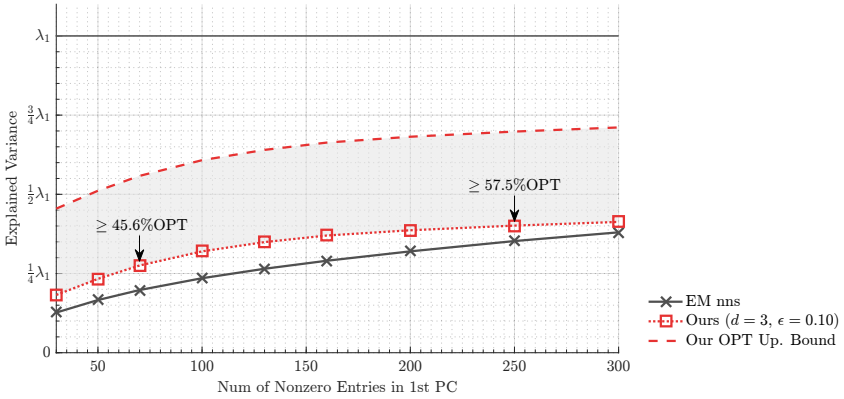


Figure 2.5: Explained variance achieved by the top s-sparse nonnegative component on the Leukemia dataset [10] as a function of the sparsity parameter s. This dataset was also used as a benchmark in [127]. We plot the variance explained by the output of Alg. 2.3 ($r = 3$, $\epsilon = 0.1$), and compare with the output of the EM algorithm of [127]. By our approximation guarantees, the maximum variance provably lies in the shaded area.

of explained variance. For larger values, the two algorithms exhibit similar performance.

The approximation guarantees accompanying our algorithm allow us to upper bound the optimal performance. For s as small as 50, which roughly amounts to 0.4% of the features, the extracted component captures at least 44.6% of the variance corresponding to the true nonnegative s-sparse PC. The obtained upper bound is a significant improvement compared to the trivial bound given by λ_1.

Low Resolution Spectrometer Dataset The Low Resolution Spectrometer (LRS) dataset, available in [22], originates from the Infra-Red Astronomy

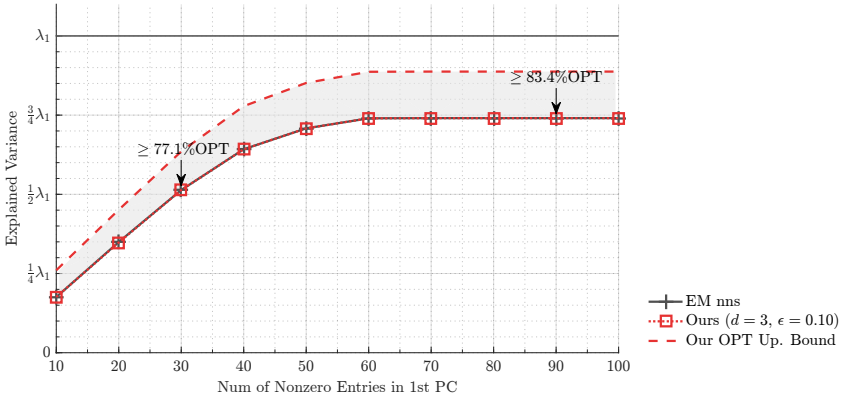


Figure 2.6: Explained variance achieved by the top s-sparse nonnegative component on the Low Resolution Spectrometer (LRS) dataset [22] as a function of the sparsity parameter s. Alg. 2.3 ($r = 3$, $\epsilon = 0.1$) and the EM algorithm [127] exhibit similar performance. The optimum value of the objective in (2.2) provably lies in the shaded area, which in this case is particularly tight.

Satellite Project. It contains 531 high quality spectra (samples) measured in 93 bands. Fig. 2.6 depicts the normalized variance explained by the computed nonnegative, s-sparse PC versus the sparsity parameter s. The empirical covariance matrix of this dataset exhibits sharper decay in the spectrum than the previous examples, yielding tighter approximation guarantees according to our theory. For instance, for $s = 20$, the extracted nonnegative component captures at least 86% of the maximum variance. For values closer to $s = 90$, where the computed PC is nonnegative but no longer sparse, this value climbs to nearly 93%.

2.9 Conclusions

We introduced a novel algorithm for nonnegative sparse PCA, expanding the spannogram theory to nonnegative quadratic optimization. We observe that the performance of our algorithm often matches and sometimes outperforms the previous state of the art [127]. Even though the theoretical running time of Alg. 2.3 scales better than EM, in practice we observed similar speed, both in the order of a few seconds. Our approach has the benefit of provable approximation, giving both theoretical a-priori guarantees and data dependent bounds that can be used to estimate the variance explained by nonnegative sparse PCs, as shown in our experiments.

Chapter 3

Sparse PCA – Components with Disjoint Supports

We consider the following multi-component sparse PCA problem: given a set of data points, we seek to extract a small number of sparse components with *disjoint* supports that jointly capture the maximum possible variance. Such components can be computed one by one, repeatedly solving the single-component problem and deflating the input data matrix, but this greedy procedure is suboptimal. We present a novel algorithm for sparse PCA that jointly optimizes multiple disjoint components. The extracted features capture variance that lies within a multiplicative factor arbitrarily close to 1 from the optimal. Our algorithm is combinatorial and computes the desired components by solving multiple instances of the bipartite maximum weight matching problem. Its complexity grows as a low order polynomial in the ambient dimension of the input data, but exponentially in its rank. However, it can be effectively applied on a low-dimensional sketch of the input data. We

Chapter 3 is based on material from Reference [19]: Megasthenis Asteris, Dimitris Papailiopoulos, Anastasios Kyrillidis, and Alexandros G Dimakis, *"Sparse PCA via Bipartite Matchings"*, In Advances in Neural Information Processing Systems, pp. 766–774, 2015. The author of this dissertation is the lead author of [19], and contributed to the conception of the research problem, the theoretical and analytical developments, the experimental design and implementation, and the writing of the manuscript and its revisions.

evaluate our algorithm on real datasets and empirically demonstrate that in many cases it outperforms existing, deflation-based approaches.

3.1 Introduction

Principal Component Analysis (PCA) reduces data dimensionality by projecting it onto principal subspaces spanned by the leading eigenvectors of the sample covariance matrix. It is one of the most widely used algorithms with applications ranging from computer vision, document clustering to network anomaly detection (see *e.g.* [108, 145, 51, 76, 160]). Sparse PCA is a useful variant that offers higher data interpretability [83, 80, 81] a property that is sometimes desired even at the cost of statistical fidelity [160]. Furthermore, when the obtained features are used in subsequent learning tasks, sparsity potentially leads to better generalization error [31].

Given a real $n \times d$ data matrix $\mathbf{S}$ representing n centered data points in d variables, the first sparse principal component is the sparse vector that maximizes the explained variance:

$$\mathbf{x}_\star \triangleq \operatorname*{argmax}_{\|\mathbf{x}\|_2=1, \|\mathbf{x}\|_0=s} \mathbf{x}^\top \mathbf{A} \mathbf{x}, \tag{3.1}$$

where $\mathbf{A} = {}^1\!/_n \cdot \mathbf{S}^\top \mathbf{S}$ is the $d \times d$ empirical covariance matrix. Unfortunately, the directly enforced sparsity constraint makes the problem NP-hard and hence computationally intractable in general. A significant volume of prior work has focused on various algorithms for approximately solving this optimization problem [80, 81, 160, 82, 110, 49, 159, 51, 155, 127, 114, 18], while some theoret-

ical results have also been established under statistical or spectral assumptions on the input data.

In most cases one is not interested in finding only the first sparse eigenvector, but rather the first k, where k is the reduced dimension where the data will be projected. Contrary to the single-component problem, there has been very limited work on computing multiple sparse components. The scarcity is partially attributed to conventional wisdom stemming from PCA: multiple components can be computed one by one, repeatedly solving the single-component sparse PCA problem (3.1) and *deflating* [103] the input data to remove information captured by previously extracted components. In fact, multi-component sparse PCA is not a uniquely defined problem in the literature. Deflation-based approaches can lead to different output depending on the type of deflation [103]; extracted components may or may not be orthogonal, while they may have disjoint or overlapping supports. In the statistics literature, where the objective is typically to recover a "true" principal subspace, a branch of work has focused on the "subspace row sparsity" [142], an assumption that leads to sparse components all supported on the same set of variables. While in [106] the authors discuss an alternative perspective on the fundamental objective of the sparse PCA problem.

We focus on the multi-component sparse PCA problem with disjoint supports, *i.e.*, the problem of computing a small number of sparse components

with non-overlapping supports that jointly maximize the explained variance:

$$\mathbf{X}_\star \triangleq \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X}_k} \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{A}\mathbf{X}\Big), \tag{3.2}$$

where $\mathcal{X}_k \triangleq \mathcal{X}_k^{(1)} \cap \mathcal{X}_k^{(2)}$ with

$$\mathcal{X}_k^{(1)} \triangleq \Big\{\mathbf{X}\in\mathbb{R}^{d\times k} : \|\mathbf{X}^j\|_2 = 1, \|\mathbf{X}^j\|_0 = s,\ \forall\, j\in[k]\Big\},$$
$$\mathcal{X}_k^{(2)} \triangleq \Big\{\mathbf{X}\in\mathbb{R}^{d\times k} : \operatorname{supp}(\mathbf{X}^i)\cap\operatorname{supp}(\mathbf{X}^j) = \emptyset,\ \forall\, i,j\in[k], i\neq j\Big\},$$

and $\mathbf{X}^j$ denoting the jth column of $\mathbf{X}$. The number k of the desired components is considered a small constant. Contrary to the greedy sequential approach that repeatedly uses deflation, our algorithm *jointly* computes all the vectors in $\mathbf{X}$ and comes with theoretical approximation guarantees. Note that even if we could solve the single-component sparse PCA problem (3.1) exactly, the greedy approach could be highly suboptimal. We show this with a simple example in Sec. B.1 of the appendix.

Our Contributions

1. We develop an algorithm that provably approximates the solution to the sparse PCA problem (3.2) within a multiplicative factor arbitrarily close to optimal. Our algorithm is the first that jointly optimizes multiple components with disjoint supports and operates by recasting the sparse PCA problem into multiple instances of the bipartite maximum weight matching problem.

2. The computational complexity of our algorithm grows as a low order polynomial in the ambient dimension d, but is exponential in the intrinsic dimension of the input data, *i.e.*, the rank of $\mathbf{A}$. To alleviate the impact of this dependence, our algorithm can be applied on a low-dimensional sketch of the input data to obtain an approximate solution to (3.2). This extra level of approximation introduces an additional penalty in our theoretical approximation guarantees, which naturally depends on the quality of the sketch and, in turn, the spectral decay of $\mathbf{A}$.We show how these bounds further translate to an additive PTAS (polynomial-time approximation scheme) for sparse PCA. Our additive PTAS outputs an approximate solution with explained variance of at least $\text{OPT} - \epsilon \cdot s$, for any sparsity $s \in [n]$, any constant error $\epsilon > 0$ and any $k = O(1)$ number of orthogonal components.[1]

3. We empirically evaluate our algorithm on real datasets, and compare it against state-of-the-art methods for the single-component sparse PCA problem (3.1) in conjunction with the appropriate deflation step. In many cases, our algorithm—as a result of jointly optimizing over multiple components—leads to significantly improved results, and outperforms deflation-based approaches.

[1]Here, OPT is the explained variance captured by the optimal set of k components that are s sparse and have disjoint supports.

3.2 Related Work

A significant volume of work has focused on the single-component sparse PCA problem (3.1); we scratch the surface and refer the reader to citations therein. Representative examples range from early heuristics in [80], to the LASSO based techniques in [81], the elastic net ℓ_1-regression in [160], ℓ_1 and ℓ_0 regularized optimization methods such as GPower in [82], a greedy branch-and-bound technique in [110], or semidefinite programming approaches [49, 159, 51]. Many focus on a statistical analysis that pertains to specific data models and the recovery of a "true" sparse component. In practice, the most competitive results in terms of the maximization in (3.1) seem to be achieved by *i*) the simple and efficient truncated power (TPower) iteration of [155], *ii*) the approach of [127] stemming from an expectation-maximization (EM) formulation, and *iii*) the (SpanSPCA) framework of [114] which solves the sparse PCA problem through low rank approximations based on [18].

We are not aware of any algorithm that explicitly addresses the multi-component sparse PCA problem (3.2). Multiple components can be extracted by repeatedly solving (3.1) with one of the aforementioned methods. To ensure disjoint supports, variables "selected" by a component are removed from the dataset. However, this greedy approach can result in highly suboptimal objective value (see Sec. B.1). More generally, there has been relatively limited work in the estimation of principal subspaces or multiple components under sparsity constraints. Non-deflation-based algorithms include extensions of the diagonal [79] and iterative thresholding [102] approaches, while [143] and [146]

propose methods that rely on the "row sparsity for subspaces" assumption of [142]. These methods yield components supported on a common set of variables, and hence solve a problem different from (3.2). In [106], the authors discuss the multi-component sparse PCA problem, propose an alternative objective function and for that problem obtain interesting theoretical guarantees. In [74] they consider a structured variant of sparse PCA where higher-order structure is encoded by an atomic norm regularization. Finally, [123] develops a framework for sparse matrix factorizaiton problems, based on an atomic norm. Their framework captures sparse PCA (although not explicitly the constraint of disjoint supports) but the resulting optimization problem, albeit convex, is NP-hard.

3.3 Our Sparse PCA Algorithm

We present a novel algorithm for the sparse PCA problem with multiple disjoint components. Our algorithm approximately solves the constrained maximization (3.2) on a $d \times d$ rank-r Positive Semi-Definite (PSD) matrix $\mathbf{A}$ within a multiplicative factor arbitrarily close to 1. It operates by recasting the maximization into multiple instances of the bipartite maximum weight matching problem. Each instance ultimately yields a feasible solution to the original sparse PCA problem; a set of k s-sparse components with disjoint supports. Finally, the algorithm exhaustively determines and outputs the set of components that maximizes the explained variance, *i.e.*, the quadratic objective in (3.2).

The computational complexity of our algorithm grows as a low order polynomial in the ambient dimension d of the input, but exponentially in its rank r. Despite the unfavorable dependence on the rank, it is unlikely that a substantial improvement can be achieved in general [105]. However, decoupling the dependence on the ambient and the intrinsic dimension of the input has an interesting ramification; instead of the original input $\mathbf{A}$, our algorithm can be applied on a low-rank surrogate to obtain an approximate solution, alleviating the dependence on r. We discuss this in Section 3.4. In the sequel, we describe the key ideas behind our algorithm, leading up to its guarantees in Theorem 3.3.

Let $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ denote the truncated eigenvalue decomposition of $\mathbf{A}$; $\mathbf{\Lambda}$ is a diagonal $r \times r$ whose ith diagonal entry is equal to the ith largest eigenvalue of $\mathbf{A}$, while the columns of $\mathbf{U}$ coincide with the corresponding eigenvectors. By the Cauchy-Schwartz inequality, for any $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \left\|\mathbf{\Lambda}^{1/2}\mathbf{U}^\top \mathbf{x}\right\|_2^2 \geq \left\langle \mathbf{\Lambda}^{1/2}\mathbf{U}^\top \mathbf{x},\, \mathbf{c}\right\rangle^2, \quad \forall\, \mathbf{c} \in \mathbb{R}^r : \|\mathbf{c}\|_2 = 1. \tag{3.3}$$

In fact, equality in (3.3) can always be achieved for $\mathbf{c}$ colinear to $\mathbf{\Lambda}^{1/2}\mathbf{U}\mathbf{x} \in \mathbb{R}^r$ and in turn

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \max_{\mathbf{c} \in \mathbb{S}_2^{r-1}} \left\langle \mathbf{x},\, \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{c}\right\rangle^2,$$

where $\mathbb{S}_2^{r-1}$ denotes the ℓ_2-unit sphere in r dimensions. More generally, for any $\mathbf{X} \in \mathbb{R}^{d\times k}$,

$$\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{X}\right) = \sum_{j=1}^{k} \mathbf{X}^{j\top} \mathbf{A}\mathbf{X}^j = \max_{\mathbf{C}:\mathbf{C}^j \in \mathbb{S}_2^{r-1} \forall j} \sum_{j=1}^{k} \left\langle \mathbf{X}^j,\, \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{C}^j\right\rangle^2. \tag{3.4}$$

Under the variational characterization of the trace objective in (3.4), the sparse PCA problem (3.2) can be re-written as a joint maximization over the variables $\mathbf{X}$ and $\mathbf{C}$ as follows:

$$\max_{\mathbf{X}\in\mathcal{X}_k} \textsc{Tr}\Big(\mathbf{X}^\top\mathbf{A}\mathbf{X}\Big) \;=\; \max_{\mathbf{X}\in\mathcal{X}_k}\max_{\mathbf{C}:\mathbf{C}^j\in\mathbb{S}_2^{r-1}\forall j} \sum_{j=1}^{k}\Big\langle \mathbf{X}^j,\, \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{C}^j\Big\rangle^2. \tag{3.5}$$

The alternative formulation of the sparse PCA problem in (3.5) may be seemingly more complicated than the original one in (3.2). However, it takes a step towards decoupling the dependence of the optimization on the ambient and intrinsic dimensions d and r, respectively. The motivation behind the introduction of the auxiliary variable $\mathbf{C}$ will become more clear in the sequel.

For a given $\mathbf{C}$, the value of $\mathbf{X}\in\mathcal{X}_k$ that maximizes the objective in (3.5) for that $\mathbf{C}$ is

$$\widehat{\mathbf{X}} \triangleq \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X}_k} \sum_{j=1}^{k}\Big\langle \mathbf{X}^j, \mathbf{W}^j\Big\rangle^2, \tag{3.6}$$

where $\mathbf{W}\triangleq\mathbf{U}\mathbf{L}^{1/2}\mathbf{C}$ is a real $d\times k$ matrix. The constrained, non-convex maximization (3.6) plays a central role in our developments. We will later describe a combinatorial $O(d\cdot(s\cdot k)^2)$ procedure to efficiently compute $\widehat{\mathbf{X}}$, reducing the maximization to an instance of the bipartite maximum weight matching problem. For now, however, let us assume that such a procedure exists.

Let $\mathbf{X}_\star$, $\mathbf{C}_\star$ be the pair that attains the maximum in (3.5); in other words, $\mathbf{X}_\star$ is the desired solution to the sparse PCA problem. If the optimal value $\mathbf{C}_\star$ of the auxiliary variable were known, then we would be able to recover $\mathbf{X}_\star$ by solving the maximization (3.6) for $\mathbf{C}=\mathbf{C}_\star$. Of course, $\mathbf{C}_\star$ is

Algorithm 3.4 Sparse PCA (Multiple disjoint components)

input $\mathbf{A}$ – $d \times d$ real PSD matrix of rank r
ϵ – Accuracy parameter in $(0, 1)$
k – target number of components
output $\overline{\mathbf{X}}$ – $d \times k$ matrix in $\mathcal{X}_k$. See Theorem 3.3 for guarantees.

1: $\mathcal{C} \leftarrow \{\}$
2: $\mathbf{U}, \mathbf{L} \leftarrow \texttt{eig}(\mathbf{A})$
3: **for each** $\mathbf{C} \in [\mathcal{N}_{\epsilon/2}(\mathbb{S}_2^{r-1})]^{\otimes k}$ **do**
4: $\quad \mathbf{W} \leftarrow \mathbf{U}\mathbf{L}^{1/2}\mathbf{C}$ $\qquad \{\mathbf{W} \in \mathbb{R}^{d \times k}\}$
5: $\quad \widehat{\mathbf{X}} \leftarrow$Output of Alg. 3.5 for input $\mathbf{W}$
6: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \left\{\widehat{\mathbf{X}}\right\}$
7: **end for**
8: $\overline{\mathbf{X}} \leftarrow \operatorname{argmax}_{\mathbf{X} \in \mathcal{C}} \textsc{Tr}\left(\mathbf{X}^\top \mathbf{A} \mathbf{X}\right)$

not known, and it is not possible to exhaustively consider all possible values in the domain of $\mathbf{C}$. Instead, we examine only a finite number of possible values of $\mathbf{C}$ over a fine discretization of its domain. In particular, let $\mathcal{N}_{\epsilon/2}(\mathbb{S}_2^{r-1})$ denote a finite $\epsilon/2$-net of the r-dimensional ℓ_2-unit sphere; for any point in $\mathbb{S}_2^{r-1}$, the net contains a point within an $\epsilon/2$ radius from the former. There are several ways to construct such a net. Further, let $[\mathcal{N}_{\epsilon/2}(\mathbb{S}_2^{r-1})]^{\otimes k} \subset \mathbb{R}^{d \times k}$ denote the kth Cartesian power of the aforementioned $\epsilon/2$-net. By construction, this collection of points contains a matrix $\mathbf{C}$ that is column-wise close to $\mathbf{C}_\star$. In turn, it can be shown using the properties of the net, that the candidate solution $\mathbf{X} \in \mathcal{X}_k$ obtained through (3.6) at that point $\mathbf{C}$ will be approximately as good as the optimal $\mathbf{X}_\star$ in terms of the quadratic objective in (3.2). All above observations yield a procedure for approximately solving the sparse PCA problem (3.2). The steps are outlined in Algorithm 3.4. Given the desired number of components k and an accuracy parameter $\epsilon \in (0, 1)$, the algorithm

generates a net $[\mathcal{N}_{\epsilon/2}(\mathbb{S}_2^{r-1})]^{\otimes k}$ and iterates over its points. At each point $\mathbf{C}$, it computes a feasible solution for the sparse PCA problem – a set of k s-sparse components – by solving maximization (3.6) via a procedure (Alg. 3.5) that will be described in the sequel. The algorithm collects the candidate solutions identified at the points of the net. The best among them achieves an objective in (3.2) that provably lies close to optimal. More formally,

Theorem 3.3. *For any real $d \times d$ rank-r PSD matrix $\mathbf{A}$, desired number of components k, number s of nonzero entries per component, and accuracy parameter $\epsilon \in (0, 1)$, Algorithm 3.4 outputs $\overline{\mathbf{X}} \in \mathcal{X}_k$ such that*

$$\textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\Big) \;\geq\; (1-\epsilon) \cdot \textsc{Tr}\Big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\Big),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\Big((4/\epsilon)^{r \cdot k} \cdot d \cdot (s \cdot k)^2\Big)$. Here, $\mathbf{X}_\star$ denotes an optimal feasible solution, i.e., $\mathbf{X}_\star \triangleq \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}_k} \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{A} \mathbf{X}\Big)$, .

Algorithm 3.4 is the first nontrivial algorithm that provably approximates the solution of the sparse PCA problem (3.2). According to Theorem 3.3, it achieves an objective value that lies within a multiplicative factor from the optimal, arbitrarily close to 1. Its complexity grows as a low-order polynomial in the dimension d of the input, but exponentially in the intrinsic dimension r. Note, however, that it can be substantially better compared to the $O(d^{s \cdot k})$ brute force approach that exhaustively considers all candidate supports for the k sparse components. The complexity of our algorithm follows from the cardinality of the net and the complexity of Algorithm 3.5, the subroutine that solves the constrained maximization (3.6). The latter is a key

ingredient of our algorithm, and is discussed in detail in the next subsection. A formal proof of Theorem 3.3 is provided in Section B.3.2.

3.3.1 Sparse Components via Bipartite Matchings

In the core of Alg. 3.4 lies a procedure that solves the constrained maximization (3.6) (Alg. 3.5). The latter breaks down the maximization into two stages. First, it identifies the support of the optimal solution $\widehat{\mathbf{X}}$ by solving an instance of the maximum weight matching problem on a bipartite graph G. Then, it recovers the exact values of its nonzero entries based on the Cauchy-Schwarz inequality. In the sequel, we provide a brief description of Alg. 3.5, leading up to its guarantees in Lemma 3.3.1.

Let $\mathcal{I}_j \triangleq \text{supp}(\widehat{\mathbf{X}}^j)$ be the support of the jth column of $\widehat{\mathbf{X}}$, $j = 1, \ldots, k$. The objective in (3.6) becomes

$$\sum_{j=1}^{k} \left\langle \widehat{\mathbf{X}}^j, \mathbf{W}^j \right\rangle^2 = \sum_{j=1}^{k} \left(\sum_{i \in \mathcal{I}_j} \widehat{X}_{ij} \cdot W_{ij} \right)^2 \leq \sum_{j=1}^{k} \sum_{i \in \mathcal{I}_j} W_{ij}^2. \tag{3.7}$$

The inequality is due to Cauchy-Schwarz and the constraint $\|\mathbf{X}^j\|_2 = 1, \forall j \in [k]$. In fact, if an oracle reveals the supports $\mathcal{I}_j$, $j = 1, \ldots, k$, the upper bound in (3.7) can always be achieved by setting the nonzero entries of $\widehat{\mathbf{X}}$ as in Algorithm 3.5 (Line 6). Therefore, the key in solving (3.6) is determining the collection of supports to maximize the right-hand side of (3.7).

By constraint, the sets $\mathcal{I}_j$ must be pairwise disjoint, each with cardinality s. These structural constraints can be captured by a bipartite graph.

Consider a weighted bipartite graph $G = (U = \{U_1, \ldots, U_k\}, V, E)$ constructed as follows[2] (Fig. 3.1):

- V is a set of d vertices $v_1, \ldots, v_d$, corresponding to the d variables, *i.e.*, the d rows of $\widehat{\mathbf{X}}$.
- U is a set of $k \cdot s$ vertices, conceptually partitioned into k disjoint subsets $U_1, \ldots, U_k$, each of cardinality s. The jth subset, U_j, is associated with the support $\mathcal{I}_j$; the s vertices $u_\alpha^{(j)}$, $\alpha = 1, \ldots, s$ in U_j serve as placeholders for the variables/indices in $\mathcal{I}_j$.
- Finally, the edge set is $E = U \times V$. The edge weights are determined by

[2]The construction is formally outlined in Algorithm B.21 in Section B.2.

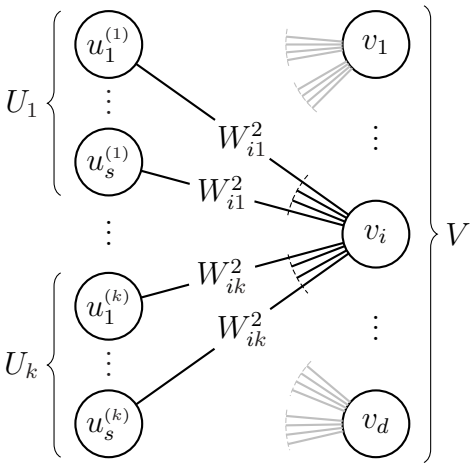


Figure 3.1: The complete bipartite graph $G = (\{U_1, \ldots, U_k\}, V, E)$ on $k \cdot s + d$ vertices, generated by Alg. 3.5 with input an $d \times k$ real matrix $\mathbf{W}$. Solving the maximum weight matching problem on G determines the support of the solution $\widehat{\mathbf{X}}$ of the maximization (3.6).

Algorithm 3.5 Compute Candidate Solution

input $\mathbf{W}$ – $d \times k$ real

output $\widehat{\mathbf{X}}$ – $d \times k$ real matrix that maximizes $\sum_{j=1}^{k} \langle \mathbf{X}^j, \mathbf{W}^j \rangle^2$ over all $\mathbf{X} \in \mathcal{X}_k$. See Lemma 3.3.1.

1: Generate a bipartite graph $G(\{U_1, \ldots, U_k\}, V, E)$ based on $\mathbf{W}$ as described in Alg. B.21.
2: Compute a maximum weight matching $\mathcal{M} \subset E$ on G.
3: $\widehat{\mathbf{X}} \leftarrow \mathbf{0}_{d \times k}$
4: **for** $j = 1, \ldots, k$ **do**
5: $\quad \mathcal{I}_j \leftarrow \{i \in \{1, \ldots, d\} : (u, v_i) \in \mathcal{M}, u \in U_j\}$
6: $\quad [\widehat{\mathbf{X}}^j]_{\mathcal{I}_j} \leftarrow [\mathbf{W}^j]_{\mathcal{I}_j} / \|[\mathbf{W}^j]_{\mathcal{I}_j}\|_2$
7: **end for**

the $d \times k$ matrix $\mathbf{W}$ in (3.6). In particular, the weight of edge $(u_\alpha^{(j)}, v_i)$ is equal to W_{ij}^2. Note that all vertices in U_j are effectively identical; they all share a common neighborhood and edge weights.

Any feasible support $\{\mathcal{I}_j\}_{j=1}^k$ corresponds to a *perfect matching* in G and vice-versa. Recall that a matching is a subset of the edges containing no two edges incident to the same vertex, while a perfect matching, in the case of an unbalanced bipartite graph $G = (U, V, E)$ with $|U| \leq |V|$, is a matching that contains at least one incident edge for each vertex in U. Given a perfect matching $\mathcal{M} \subseteq E$, the disjoint neighborhoods of U_js under $\mathcal{M}$ yield a support $\{\mathcal{I}_j\}_{j=1}^k$. Conversely, any valid support yields a unique perfect matching in G (taking into account that all vertices in U_j are isomorphic). Moreover, due to the choice of weights in G, the right-hand side of (3.7) for a given support $\{\mathcal{I}_j\}_{j=1}^k$ is equal to the weight of the matching $\mathcal{M}$ in G induced by the former, *i.e.*, $\sum_{j=1}^k \sum_{i \in \mathcal{I}_j} W_{ij}^2 = \sum_{(u,v) \in \mathcal{M}} w(u, v)$. It follows that determining

the support of the solution in (3.6), reduces to solving the maximum weight matching problem on the bipartite graph G.

Algorithm 3.5 readily follows. Given $\mathbf{W} \in \mathbb{R}^{d\times k}$, the algorithm generates a weighted bipartite graph G as described, and computes its maximum weight matching. Based on the latter, it first recovers the desired support of $\widehat{\mathbf{X}}$ (Line 5), and subsequently the exact values of its nonzero entries (Line 6). The running time is dominated by the computation of the matching, which can be done in $O(|E||U| + |U|^2 \log |U|)$ using a variant of the Hungarian algorithm [122]. Hence,

Lemma 3.3.1. *For any $\mathbf{W} \in \mathbb{R}^{d\times k}$, Algorithm 3.5 computes the solution to (3.6), in time $O(d \cdot (s \cdot k)^2)$.*

A more formal analysis and proof of Lemma 3.3.1 is available in Sec. B.3.1. This completes the description of our sparse PCA algorithm (Alg. 3.4) and the proof sketch of Theorem 3.3.

3.4 Sparse PCA on Low-Dimensional Sketches

Algorithm 3.4 approximately solves the sparse PCA problem (3.2) on a $d \times d$ rank-r PSD matrix $\mathbf{A}$ in time that grows as a low-order polynomial in the ambient dimension d, but depends exponentially on r. This dependence can be prohibitive in practice. To mitigate its effect, we can apply our sparse PCA algorithm on a low-rank sketch of $\mathbf{A}$. Intuitively, the quality of the extracted components should depend on how well that low-rank surrogate

Algorithm 3.6 Sparse PCA on Low Dim. Sketch

input $\mathbf{S}$ – $n \times d$ real matrix (Input data matrix.)
r – parameter controlling the rank of approximation of $\mathbf{S}$ to be used. Takes values in $[\min n, d]$.
ϵ – Accuracy parameter taking values in $(0, 1)$.
k – the number of desired componenets.

output $\overline{\mathbf{X}}_{(\mathrm{r})}$ – $d \times k$ real matrix in $\in \mathcal{X}_k$. See Thm. 3.4.

1: $\overline{\mathbf{S}} \leftarrow \textsc{Sketch}(\mathbf{S}, r)$
2: $\overline{\mathbf{A}} \leftarrow \overline{\mathbf{S}}^\top \overline{\mathbf{S}}$
3: $\overline{\mathbf{X}}_{(\mathrm{r})} \leftarrow$ Output of Algorithm 3.4 for input $(\overline{\mathbf{A}}, \epsilon, k)$

approximates the original input.

More formally, let $\mathbf{S}$ be the real $n \times d$ data matrix representing n (potentially centered) datapoints in d variables, and $\mathbf{A}$ the corresponding $d \times d$ covariance matrix. Further, let $\overline{\mathbf{S}}$ be a low-dimensional sketch of the original data; an $n \times d$ matrix whose rows lie in an r-dimensional subspace, with r being an accuracy parameter. Such a sketch can be obtained in several ways, including for example exact or approximate SVD, or online sketching methods [66]. Finally, let $\overline{\mathbf{A}} = {}^1\!/_n \cdot \overline{\mathbf{S}}^\top \overline{\mathbf{S}}$ be the covariance matrix of the sketched data. Then, instead of $\mathbf{A}$, we can approximately solve the sparse PCA problem by applying Algorithm 3.4 on the low-rank surrogate $\overline{\mathbf{A}}$. The above are formally outlined in Algorithm 3.6. We note that the covariance matrix $\overline{\mathbf{A}}$ does not need to be explicitly computed; Algorithm 3.4 can operate directly on the (sketched) input data matrix.

Theorem 3.4. *For any $n \times d$ input data matrix $\mathbf{S}$, with corresponding empirical covariance matrix $\mathbf{A} = {}^1\!/_n \cdot \mathbf{S}^\top \mathbf{S}$, any desired number of components k, and*

accuracy parameters $\epsilon \in (0,1)$ *and* r*, Algorithm 3.6 outputs* $\mathbf{X}_{(r)} \in \mathcal{X}_k$ *such that*

$$\textsc{Tr}\Big(\mathbf{X}_{(r)}^\top \mathbf{A}\mathbf{X}_{(r)}\Big) \;\geq\; (1-\epsilon)\cdot \textsc{Tr}\Big(\mathbf{X}_\star^\top \mathbf{A}\mathbf{X}_\star\Big) - 2\cdot k\cdot \lambda_{1,s}(\mathbf{A}-\overline{\mathbf{A}}),$$

in time $\mathrm{T}_{\mathsf{SKETCH}}(r) + \mathrm{T}_{\mathsf{SVD}}(r) + O\Big((4/\epsilon)^{r\cdot k}\cdot d\cdot (s\cdot k)^2\Big)$*. Here,* $\mathbf{X}_\star \in \mathcal{X}_k$ *denotes the point that maximizes* $\textsc{Tr}\Big(\mathbf{X}^\top\mathbf{A}\mathbf{X}\Big)$ *over all* $\mathbf{X}_\star \in \mathcal{X}_k$*, and* $\lambda_{1,s}(\mathbf{A})$ *denotes the sparse eigenvalue, i.e., the eigenvalue that corresponds to the principal* s*-sparse eigenvector of* $\mathbf{A}$*.*

The error $\lambda_{1,s}(\mathbf{A}-\overline{\mathbf{A}})$ and in turn the tightness of the approximation guarantees hinges on the quality of the sketch $\overline{\mathbf{A}}$. Higher values of the parameter r (the rank of the sketch) can allow for a more accurate solution and tighter guarantees. That is the case, for example, when the sketch is obtained through exact SVD. In that sense, Theorem 3.4 establishes a natural trade-off between the running time of Algorithm 3.6 and the quality of the approximation guarantees. A formal proof of Theorem 3.4 is provided in Section B.3.3. Observe that the error term itself is a sparse eigenvalue that is hard to approximate, however even loose bounds provide tight conditional approximation results, as we see next.

Using the main matrix approximation result of [6], the next theorem establishes that Algorithm 3.6 can be turned into an additive PTAS.

Theorem 3.5. *Let* $\mathbf{A}$ *be a* $d\times d$ *positive semidefinite matrix with entries in* $[-1,1]$*,* $\mathbf{V}$ *be a* $d\times d$ *matrix such that* $\mathbf{A}=\mathbf{V}\mathbf{V}^\top$*. Further, let* $\mathbf{R}$ *be a random*

$d \times r$ matrix with entries drawn i.i.d. according to $\mathcal{N}(0, 1/r)$, and define

$$\overline{\mathbf{A}} \triangleq \mathbf{V}\mathbf{R}\mathbf{R}^\top\mathbf{V}^\top.$$

For any constant $\epsilon \in (0, 1]$, let $r = O(\epsilon^{-2} \log d)$. Then, for any desired sparsity s, and number of components $k = O(1)$, Algorithm 3.4 with input argument $\overline{\mathbf{A}}$ and accuracy parameter ϵ, outputs $\mathbf{X}_{(r)} \in \mathcal{X}_k$ such that

$$\textsc{Tr}\left(\mathbf{X}_{(r)}^\top \mathbf{A}\mathbf{X}_{(r)}\right) \geq \textsc{Tr}\left(\mathbf{X}_\star^\top \mathbf{A}\mathbf{X}_\star\right) - \epsilon \cdot s$$

with probability at least $1 - 1/poly(d)$, in time $n^{O(\log(1/\epsilon)/\epsilon^2))}$.

Proof. See Appendix Sec. B.3.3. □

Remark 3.4.1. *Note that $\lambda_1(\mathbf{A} - \overline{\mathbf{A}})$ serves as another elementary upper bound on $\lambda_{1,s}(\mathbf{A} - \overline{\mathbf{A}})$. If $\overline{\mathbf{A}}$ is a the rank-d SVD approximation of $\mathbf{A}$, then—similar to [16]—we can obtain a multiplicative PTAS for sparse PCA, under the assumption of a decaying spectrum (e.g., under a power-law decay), and for $s = \Omega(n)$.*

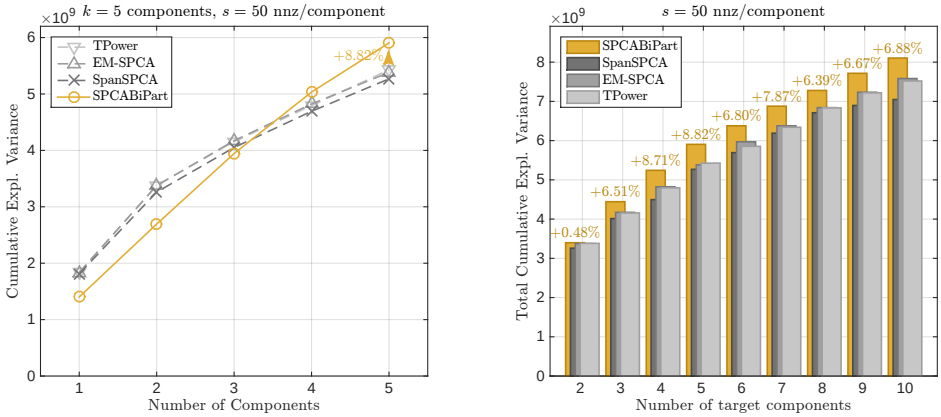
3.5 Experiments

We evaluate our algorithm on a series of real datasets, and compare it to deflation-based approaches for sparse PCA using TPower [155], EM [127], and SpanSPCA [114]. The latter are representative of the state of the art for the single-component sparse PCA problem (3.1). Multiple components are computed one by one. To ensure disjoint supports, the deflation step effectively amounts to removing from the dataset all variables used by previously

extracted components. For algorithms that are randomly initialized, we depict best results over multiple random restarts.

Our experiments are conducted in a Matlab environment. Due to its nature, our algorithm is easily parallelizable; its prototypical implementation utilizes the Parallel Pool Matlab feature to exploit multicore (or distributed cluster) capabilities. Recall that our algorithm operates on a low-rank approximation of the input data. Unless otherwise specified, it is configured for a rank-4 approximation obtained via truncated SVD. Finally, we note that our algorithm is slower than the deflation-based methods. We set a barrier on the execution time of our algorithm at the cost of the theoretical approximation guarantees; the algorithm returns the best result at the time of termination. This "early termination" can only hurt the performance of our algorithm.

Leukemia Dataset We evaluate our algorithm on the Leukemia dataset [22]. The dataset comprises 72 samples, each consisting of expression values for 12582 probe sets. We extract $k = 5$ sparse components, each active on $s = 50$ features. In Fig. 3.2(a), we plot the cumulative explained variance versus the number of components. Deflation-based approaches are greedy: the leading components capture high values of variance, but subsequent ones contribute less. On the contrary, our algorithm jointly optimizes the $k = 5$ components and achieves higher *total* cumulative variance; one cannot identify a top component. We repeat the experiment for multiple values of k. Fig. 3.2(b) depicts the total cumulative variance capture by each method,



(a) (b)

Figure 3.2: Cumulative variance captured by k s-sparse extracted components; Leukemia dataset [22]. The dataset comprises 72 samples in 12582 dimensions. We arbitrarily set the sparsity to $s = 50$ nonzero entries per component. Fig. 3.2(a) depicts the cumulative variance versus the number of components, for $k = 5$. Deflation-based approaches are greedy; first components capture high variance, but subsequent contribute less. Our algorithm jointly optimizes the k components and achieves higher objective. Fig. 3.2(b) depicts the cumul. variance achieved for various values of k.

for each value of k.

Additional Datasets We repeat the experiment on multiple datasets, arbitrarily selected from [22]. Table 3.1 lists the total cumulative variance captured by $k = 5$ components, each with $s = 40$ nonzero entries, extracted using the four methods. Our algorithm achieves the highest values in most cases.

Bag of Words (BoW) Dataset This is a collection of text corpora [22] stored under the "bag-of-words" model. For each text corpus, a vocabulary of d

words is extracted upon tokenization, and the removal of stopwords and words appearing fewer than ten times in total. Each document is then represented as a vector in that d-dimensional space, with the ith entry corresponding to the number of appearances of the ith vocabulary entry in the document.

We solve the sparse PCA problem (3.2) on the word-by-word cooccurrence matrix, and extract $k = 8$ sparse components, each with cardinality $s = 10$. We note that the latter is not explicitly constructed; our algorithm can operate directly on the input word-by-document matrix. Table 3.2 lists the variance captured by each method; our algorithm consistently outperforms the other approaches.

Finally, note that here each sparse component effectively *selects* a small set of words. In turn, the k extracted components can be interpreted as a set of well-separated *topics*. In Table 3.3, we list the topics extracted from the NY Times corpus (part of the Bag of Words dataset). The corpus consists

		TPower	EM sPCA	SpanSPCA	SPCABiPart
Amzn Com Rev	(1500×10000)	$7.31e+03$	$7.32e+03$	$7.31e+03$	$\mathbf{7.79e+03}$
Arcence Train	(100×10000)	$1.08e+07$	$1.02e+07$	$1.08e+07$	$\mathbf{1.10e+07}$
CBCL Face Train	(2429×361)	$5.06e+00$	$5.18e+00$	$5.23e+00$	$\mathbf{5.29e+00}$
Isolet-5	(1559×617)	$3.31e+01$	$3.43e+01$	$3.34e+01$	$\mathbf{3.51e+01}$
Leukemia	(72×12582)	$5.00e+09$	$5.03e+09$	$4.84e+09$	$\mathbf{5.37e+09}$
Pems Train	(267×138672)	$\mathbf{3.94e+00}$	$3.58e+00$	$3.89e+00$	$3.75e+00$
Mfeat Pix	(2000×240)	$5.00e+02$	$5.27e+02$	$5.08e+02$	$\mathbf{5.47e+02}$

Table 3.1: Total cumulative variance captured by $k = 5$ 40-sparse extracted components on various datasets [22]. For each dataset, we list the size (#samples×#variables) and the value of variance captured by each method. Our algorithm operates on a rank-4 sketch in all cases.

		TPower	EM sPCA	SpanSPCA	SPCABiPart
BoW:NIPS	(1500×12419)	$2.51e+03$	$2.57e+03$	$2.53e+03$	$\mathbf{3.34e+03}$ (+29.98%)
BoW:KOS	(3430×6906)	$4.14e+01$	$4.24e+01$	$4.21e+01$	$\mathbf{6.14e+01}$ (+44.57%)
BoW:Enron	(39861×28102)	$2.11e+02$	$2.00e+02$	$2.09e+02$	$\mathbf{2.38e+02}$ (+12.90%)
BoW:NyTimes	(300000×102660)	$4.81e+01$	–	$4.81e+01$	$\mathbf{5.31e+01}$ (+10.38%)

Table 3.2: Total variance captured by $k = 8$ extracted components, each with $s = 15$ nonzero entries – Bag of Words dataset [22]. For each corpus, we list the size (#documents×#vocabulary-size) and the explained variance. Our algorithm operates on a rank-5 sketch in all cases.

of $3 \cdot 10^5$ news articles and a vocabulary of $d = 102660$ words.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
1:	percent	zzz_united_states	zzz_bush	company	team	cup	school	zzz_al_gore
2:	million	zzz_u_s	official	companies	game	minutes	student	zzz_george_bush
3:	money	zzz_american	government	market	season	add	children	campaign
4:	high	attack	president	stock	player	tablespoon	women	election
5:	program	military	group	business	play	oil	show	plan
6:	number	palestinian	leader	billion	point	teaspoon	book	tax
7:	need	war	country	analyst	run	water	family	public
8:	part	administration	political	firm	right	pepper	look	zzz_washington
9:	problem	zzz_white_house	american	sales	home	large	hour	member
10:	com	games	law	cost	won	food	small	nation

Table 3.3: BoW:NyTimes dataset [22]. The table lists the words corresponding to the $s = 10$ nonzero entries of each of the $k = 8$ extracted components (topics). Words corresponding to higher magnitude entries appear higher in the topic.

3.6 Conclusions

We considered the sparse PCA problem for multiple components with disjoint supports. Existing methods for the single component problem can be used along with an appropriate deflation step to compute multiple compo-

nents one by one, leading to potentially suboptimal results. We presented a novel algorithm for jointly computing multiple sparse and disjoint components with provable approximation guarantees. Our algorithm is combinatorial and exploits interesting connections between the sparse PCA and the bipartite maximum weight matching problems. Its running time grows as a low-order polynomial in the ambient dimension of the input data, but depends exponentially on its rank. To alleviate this dependency, we can apply the algorithm on a low-dimensional sketch of the input, at the cost of an additional error in our theoretical approximation guarantees. Empirical evaluation showed that in many cases our algorithm outperforms deflation-based approaches.

Chapter 4

Nonnegative Components and Orthogonal NMF

Orthogonal Nonnegative Matrix Factorization (ONMF) aims to approximate a nonnegative matrix as the product of two k-dimensional nonnegative factors, one of which has orthonormal columns. It yields potentially useful data representations as superposition of disjoint parts, while it has been shown to work well for clustering tasks where traditional methods underperform. Existing algorithms rely mostly on heuristics, which despite their good empirical performance, lack provable performance guarantees.

We present a new ONMF algorithm with provable approximation guarantees. For any constant dimension k, we obtain an additive EPTAS without any assumptions on the input. Our algorithm relies on a novel approximation to the related Nonnegative Principal Component Analysis (NNPCA) problem;

Chapter 4 is based on material from Reference [17]: Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis, *"Orthogonal NMF Through Subspace Exploration"*, In Advances in Neural Information Processing Systems, pp. 343–351, 2015. The author of this dissertation is the lead author of [17], and contributed to the conception of the research problem, the theoretical and analytical developments, the experimental design and implementation, and the writing of the manuscript and its revisions.

given an arbitrary data matrix, NNPCA seeks k nonnegative components that jointly capture most of the variance. Our NNPCA algorithm is of independent interest and generalizes previous work that could only obtain guarantees for a single component.

We evaluate our algorithms on several real and synthetic datasets and show that their performance matches or outperforms the state of the art.

4.1 Introduction

Orthogonal NMF The success of Nonnegative Matrix Factorization (NMF) in a range of disciplines spanning data mining, chemometrics, signal processing and more, has driven an extensive practical and theoretical study [95, 32, 126, 100, 43, 28, 62, 73]. Its power lies in its potential to generate meaningful decompositions of data into non-subtractive combinations of a few nonnegative parts.

Orthogonal NMF (ONMF) [55] is a variant of NMF with an additional orthogonality constraint: given a real *nonnegative* $m \times n$ matrix $\mathbf{M}$ and a target dimension k, typically much smaller than m and n, we seek to approximate $\mathbf{M}$ by the product of an $m \times k$ nonnegative matrix $\mathbf{W}$ with orthogonal (w.l.o.g, orthonormal) columns, and an $n \times k$ nonnegative matrix $\mathbf{H}$. In the form of an optimization,

$$\text{(ONMF)} \qquad \mathcal{E}_\star \triangleq \min_{\substack{\mathbf{W}\geq\mathbf{0},\, \mathbf{H}\geq\mathbf{0} \\ \mathbf{W}^\top\mathbf{W}=\mathbf{I}_k}} \|\mathbf{M} - \mathbf{W}\mathbf{H}^\top\|_{\mathrm{F}}^2. \tag{4.1}$$

Since $\mathbf{W}$ is nonnegative, its columns are orthogonal if and only if they have disjoint supports. In turn, each row of $\mathbf{M}$ is approximated by a scaled version of a single (transposed) column of $\mathbf{H}$.

Despite the admittedly limited representational power compared to NMF, ONMF yields sparser part-based representations that are potentially easier to interpret, while it naturally lends itself to certain applications. In a clustering setting, for example, $\mathbf{W}$ serves as a cluster membership matrix and the columns of $\mathbf{H}$ correspond to k cluster centroids [97, 55, 119]. Empirical evidence shows that ONMF performs remarkably well in certain clustering tasks, such as document classification [41, 153, 89, 152, 119, 28]. In the analysis of textual data where $\mathbf{M}$ is a words by documents matrix, the orthogonal columns of $\mathbf{W}$ can be interpreted as topics defined by disjoint subsets of words. In the case of an image dataset, with each column of $\mathbf{M}$ corresponding to an image evaluated on multiple pixels, each of the orthogonal base vectors highlights a disjoint segment of the image area.

Nonnegative PCA For any given factor $\mathbf{W} \geq \mathbf{0}$ with orthonormal columns, the second ONMF factor $\mathbf{H}$ is readily determined: $\mathbf{H} = \mathbf{M}^\top \mathbf{W} \geq \mathbf{0}$. This follows from the fact that $\mathbf{M}$ is by assumption nonnegative. Based on the above, it can be shown that the ONMF problem (4.1) is equivalent to

$$\text{(NNPCA)} \qquad \mathcal{V}_\star \triangleq \max_{\mathbf{W} \in \mathcal{W}_k} \|\mathbf{M}^\top \mathbf{W}\|_{\mathrm{F}}^2, \tag{4.2}$$

where

$$\mathcal{W}_k \triangleq \Big\{ \mathbf{W} \in \mathbb{R}^{m \times k} : \ \mathbf{W} \geq \mathbf{0}, \ \mathbf{W}^\top \mathbf{W} = \mathbf{I}_k \Big\}.$$

For arbitrary —*i.e.*, not necessarily nonnegative— matrices $\mathbf{M}$, the non-convex maximization (4.2) coincides with the Nonnegative Principal Component Analysis (NNPCA) problem [157]. Similarly to vanilla PCA, NNPCA seeks k orthogonal components that jointly capture most of the variance of the (centered) data in $\mathbf{M}$. The nonzero entries of the extracted components, however, must be positive, which renders the problem NP-hard even in the case of a single component ($k = 1$) [16].

Our Contributions We present a novel algorithm for NNPCA. Our algorithm approximates the solution to (4.2) for any real input matrix and is accompanied with global approximation guarantees. Using the above as a building block, we develop an algorithm to approximately solve the ONMF problem (4.1) on any *nonnegative* matrix. Our algorithm outputs a solution that strictly satisfies both the nonnegativity and the orthogonality constraints. Our main results are as follows:

Theorem 1. (NNPCA) *For any $m \times n$ matrix $\mathbf{M}$, desired number of components k, and accuracy parameter $\epsilon \in (0, 1)$, our NNPCA algorithm computes $\overline{\mathbf{W}} \in \mathcal{W}_k$ such that*

$$\left\|\mathbf{M}^\top \overline{\mathbf{W}}\right\|_{\mathrm{F}}^2 \;\geq\; (1-\epsilon) \cdot \mathcal{V}_\star - k \cdot \sigma_{r+1}^2(\mathbf{M}),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\Big((1/\epsilon)^{r \cdot k} \cdot k \cdot m\Big)$. Here, $\mathrm{T}_{\mathsf{SVD}}(r)$ denotes the time required to compute a rank-r approximation $\overline{\mathbf{M}}$ of the input $\mathbf{M}$ using the truncated singular value decomposition (SVD), and $\sigma_{r+1}(\mathbf{M})$ is the $(r+1)$th singular value of $\mathbf{M}$.

Our NNPCA algorithm operates on the low-rank matrix $\overline{\mathbf{M}}$. The parameter r controls a natural trade-off; higher values of r lead to tighter guarantees, but impact the running time of our algorithm. Finally, note that despite the exponential dependence in r and k, the complexity scales polynomially in the ambient dimension of the input.

If the input matrix $\mathbf{M}$ is nonnegative, as in any instance of the ONMF problem, we can compute an approximate orthogonal nonnegative factorization in two steps: first obtain an orthogonal factor $\mathbf{W}$ by (approximately) solving the NNPCA problem on $\mathbf{M}$, and subsequently set $\mathbf{H} = \mathbf{M}^\top \mathbf{W}$.

Theorem 2. (ONMF) *For any $m \times n$ nonnegative matrix $\mathbf{M}$, target dimension k, and desired accuracy $\epsilon \in (0, 1)$, our ONMF algorithm computes an ONMF pair $\mathbf{W}, \mathbf{H}$, such that*

$$\left\|\mathbf{M} - \mathbf{W}\mathbf{H}^\top\right\|_{\mathrm{F}}^2 \leq \mathcal{E}_\star + \epsilon \cdot \|\mathbf{M}\|_{\mathrm{F}}^2,$$

in time $\mathrm{T}_{\mathsf{SVD}}(k/\epsilon) + O\Big((1/\epsilon)^{k^2/\epsilon} \cdot k \cdot m\Big)$.

For any constant dimension k, Theorem 4.7 implies an additive EPTAS for the relative ONMF approximation error. This is, to the best our knowledge, the first general ONMF approximation guarantee since we impose no assumptions on $\mathbf{M}$ beyond nonnegativity.

We evaluate our NNPCA and ONMF algorithms on synthetic and real datasets. As we discuss in Section 4.5, for several cases we show improvements compared to the previous state of the art.

4.2 Related Work

ONMF as a variant of NMF first appeared implicitly in [156]. The formulation in (4.1) was introduced in [55]. Several algorithms in a subsequent line of work [96, 34, 98, 41, 153, 37] approximately solve variants of that optimization problem. Most rely on modifying approaches for NMF to accommodate the orthogonality constraint; either exploiting the additional structural properties in the objective [153], introducing a penalization term [55], or updating the current estimate in suitable directions [41], they typically reduce to a multiplicative update rule which attains orthogonality only in a limit sense. In [119], the authors suggest two alternative approaches: an EM algorithm motivated by connections to spherical k-means, and an augmented Lagrangian formulation that explicitly enforces orthogonality, but only achieves nonnegativity in the limit. Despite their good performance in practice, existing methods only guarantee local convergence.

A significant body of work [60, 11, 12, 90] has focused on Separable NMF, a variant of NMF partially related to ONMF. Sep. NMF seeks to decompose $\mathbf{M}$ into the product of two nonnegative matrices $\mathbf{W}$ and $\mathbf{H}^\top$ where $\mathbf{W}$ contains a permutation of the $k \times k$ identity matrix. Intuitively, the geometric picture of Sep. NMF should be quite different from that of ONMF: in the former, the rows of $\mathbf{H}^\top$ are the extreme rays of a convex cone enclosing all rows of $\mathbf{M}$, while in the latter they should be scattered in the interior of that cone so that each row of $\mathbf{M}$ has one representative in small angular distance. Algebraically, ONMF factors approximately satisfy the structural requirement

of Sep. NMF, but the converse is not true: a Sep. NMF solution is *not* a valid ONMF solution (Fig. 4.1).

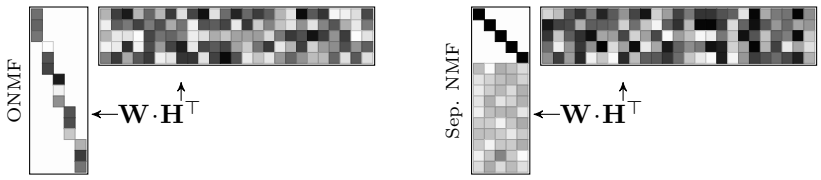


Figure 4.1: ONMF and Separable NMF, upon appropriate permutation of the rows of $\mathbf{M}$. In the first case, each row of $\mathbf{M}$ is approximated by a single row of $\mathbf{H}^\top$, while in the second, by a nonnegative combination of all k rows of $\mathbf{H}^\top$.

In the NNPCA front, nonnegativity as a constraint on PCA first appeared in [157], which proposed a coordinate-descent scheme on a penalized version of (4.2) to compute a set of nonnegative components. In [127], the authors developed a framework stemming from Expectation-Maximization (EM) on a generative model of PCA to compute a nonnegative (and optionally sparse) component. In [16], the authors proposed an algorithm based on sampling points from a low-dimensional subspace of the data covariance and projecting them on the nonnegative orthant. [127] and [16] focus on the single-component problem; multiple components can be computed sequentially employing a heuristic deflation step. Our main theoretical result is a generalization of the analysis of [16] for multiple components. Finally, note that despite the connection between the two problems, existing algorithms for ONMF are not suitable for NNPCA as they only operate on nonnegative matrices.

4.3 Algorithms and Guarantees

4.3.1 Overview

We first develop an algorithm to approximately solve the NNPCA problem (4.2) on any arbitrary —*i.e.*, not necessarily nonnegative— $m \times n$ matrix $\mathbf{M}$. The core idea is to solve the NNPCA problem not directly on $\mathbf{M}$, but a rank-r approximation $\overline{\mathbf{M}}$ instead. Our main technical contribution is a procedure that approximates the solution to the constrained maximization (4.2) on a rank-r matrix within a multiplicative factor arbitrarily close to 1, in time exponential in r, but polynomial in the dimensions of the input. Our Low Rank NNPCA algorithm relies on generating a large number of candidate solutions, one of which provably achieves objective value close to optimal.

The k nonnegative components $\overline{\mathbf{W}} \in \mathcal{W}_k$ returned by our Low Rank NNPCA algorithm on the sketch $\overline{\mathbf{M}}$ are used as a surrogate for the desired components of the original input $\mathbf{M}$. Intuitively, the performance of the extracted nonnegative components depends on how well $\mathbf{M}$ is approximated by the low rank sketch $\overline{\mathbf{M}}$; a higher rank approximation leads to better results. However, the complexity of our low rank solver depends exponentially in the rank of its input. A natural trade-off arises between the quality of the extracted components and the running time of our NNPCA algorithm.

Using our NNPCA algorithm as a building block, we propose a novel algorithm for the ONMF problem (4.1). In an ONMF instance, we are given an $m \times n$ *nonnegative* matrix $\mathbf{M}$ and a target dimension $k < m, n$, and seek to approximate $\mathbf{M}$ with a product $\mathbf{W}\mathbf{H}^\top$ of two nonnegative matrices, where $\mathbf{W}$

Algorithm 4.7 NNPCA

input $\overline{\mathbf{M}}$ – $m \times n$ real rank-r matrix
ϵ – Accuracy parameter in $(0, 1)$
k – target number of components

output $\mathbf{W}$ – $m \times k$ real nonnegative matrix in the feasible region $\mathcal{W}_k$. See Lemma 4.3.2.

1: $\mathcal{C} \leftarrow \{\}$ {Candidate solutions}
2: $\overline{\mathbf{U}}, \overline{\mathbf{\Sigma}}, \overline{\mathbf{V}} \leftarrow \mathtt{svd}(\overline{\mathbf{M}}, r)$ {Trunc. SVD}
3: **for each** $\mathbf{C} \in \mathcal{N}_{\epsilon/2}^{\otimes k}\left(\mathbb{S}_2^{r-1}\right)$ **do**
4: $\quad \mathbf{A} \leftarrow \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\mathbf{C}$ {$\mathbf{A} \in \mathbb{R}^{m \times k}$}
5: $\quad \widehat{\mathbf{W}} \leftarrow$ Output of Algorithm 4.9 with input $\mathbf{A}$
6: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \left\{\widehat{\mathbf{W}}\right\}$
7: **end for**
8: $\overline{\mathbf{W}} \leftarrow \arg\max_{\mathbf{W} \in \mathcal{C}} \left\|\overline{\mathbf{M}}^\top \mathbf{W}\right\|_{\mathrm{F}}^2$

additionally has orthonormal columns. Computing such a factorization is equivalent to solving the NNPCA problem on the nonnegative matrix $\mathbf{M}$. (See Appendix C.1.1 for a formal argument.) Once a nonnegative orthogonal factor $\mathbf{W}$ is obtained, the second ONMF factor is readily determined: $\mathbf{H} = \mathbf{M}^\top \mathbf{W}$ minimizes the Frobenius approximation error in (4.1) for a given $\mathbf{W}$. Under an appropriate configuration of the accuracy parameters, for any nonnegative $m \times n$ input $\mathbf{M}$ and constant target dimension k, our algorithm yields an additive EPTAS for the relative approximation error, without any additional assumptions on the input data.

4.3.2 Main Results

Low Rank NNPCA We develop an algorithm to approximately solve the NNPCA problem on an $m \times n$ real rank-r matrix $\overline{\mathbf{M}}$:

$$\overline{\mathbf{W}}_\star \triangleq \operatorname*{argmax}_{\mathbf{W} \in \mathcal{W}_k} \left\| \overline{\mathbf{M}}^\top \mathbf{W} \right\|_{\mathrm{F}}. \tag{4.3}$$

The procedure, which lies in the core of our subsequent developments, is encoded in Alg. 4.7. We describe it in detail in Section 4.4. The key observation is that irrespectively of the dimensions of the input, the maximization in (4.3) can be reduced to $k \cdot r$ unknowns. The algorithm generates a large number of k-tuples of r-dimensional points; the collection of tuples is denoted by $\mathcal{N}_{\epsilon/2}^{\otimes k}\big(\mathbb{S}_2^{r-1}\big)$, the kth Cartesian power of an $\epsilon/2$-net of the r-dimensional unit sphere. Using these points, we effectively sample the column-space of the input $\overline{\mathbf{M}}$. Each tuple yields a feasible solution $\mathbf{W} \in \mathcal{W}_k$ through a computationally efficient subroutine (Alg. 4.9). The best among those candidate solutions is provably close to the optimal $\overline{\mathbf{W}}_\star$ with respect to the objective in (4.2). The approximation guarantees are formally established in the following lemma.

Lemma 4.3.2. *For any real $m \times n$ matrix $\overline{\mathbf{M}}$ with rank r, desired number of components k, and accuracy parameter $\epsilon \in (0, 1)$, Algorithm 4.7 outputs $\overline{\mathbf{W}} \in \mathcal{W}_k$ such that*

$$\left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}} \right\|_{\mathrm{F}}^2 \;\geq\; (1 - \epsilon) \cdot \left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}}_\star \right\|_{\mathrm{F}}^2,$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\Big((2/\epsilon)^{r \cdot k} \cdot k \cdot m\Big)$. Here, $\overline{\mathbf{W}}_\star$ denotes the optimal solution defined in (4.3).

Proof. (See Appendix C.1.2.) □

Nonnegative PCA Given an arbitrary real $m \times n$ matrix $\mathbf{M}$, we can generate a rank-r sketch $\overline{\mathbf{M}}$ and solve the low rank NNPCA problem on $\overline{\mathbf{M}}$ using Algorithm 4.7. The output $\overline{\mathbf{W}} \in \mathcal{W}_k$ of the low rank problem can be used as a surrogate for the desired components of the original input $\mathbf{M}$. For simplicity, here we consider the case where $\overline{\mathbf{M}}$ is the rank-r approximation of $\mathbf{M}$ obtained by the truncated SVD. Intuitively, the performance of the extracted components on the original data matrix $\mathbf{M}$ will depend on how well the latter is approximated by $\overline{\mathbf{M}}$, and in turn by the spectral decay of the input data. For example, if $\mathbf{M}$ exhibits a sharp spectral decay, which is frequently the case in real data, a moderate value of r suffices to obtain a good approximation. This leads to our first main theorem which formally establishes the guarantees of our NNPCA algorithm.

Theorem 4.6. *For any real $m \times n$ matrix $\mathbf{M}$, let $\overline{\mathbf{M}}$ be its best rank-r approximation. Algorithm 4.7 with input $\overline{\mathbf{M}}$, and parameters k and $\epsilon \in (0,1)$, outputs $\overline{\mathbf{W}} \in \mathcal{W}_k$ such that*

$$\left\|\mathbf{M}^\top \overline{\mathbf{W}}\right\|_{\mathrm{F}}^2 \;\geq\; (1-\epsilon) \cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 - k \cdot \left\|\mathbf{M} - \overline{\mathbf{M}}\right\|_2^2,$$

where $\mathbf{W}_\star \triangleq \operatorname{argmax}_{\mathbf{W} \in \mathcal{W}_k} \left\|\mathbf{M}^\top \mathbf{W}\right\|_{\mathrm{F}}^2$, in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\Big((1/\epsilon)^{r \cdot k} \cdot k \cdot m\Big)$.

Proof. The proof follows from Lemma 4.3.2. It is formally provided in Appendix C.1.3. □

Theorem 4.6 establishes a trade-off between the computational complexity of the proposed NNPCA approach and the tightness of the approximation guarantees; higher values of r imply smaller $\|\mathbf{M} - \overline{\mathbf{M}}\|_2^2$ and in turn a

tighter bound (assuming that the singular values of $\mathbf{M}$ decay), but have an exponential impact on the running time. Despite the exponential dependence on r and k, our approach is polynomial in the dimensions of the input $\mathbf{M}$, dominated by the truncated SVD.

In practice, Algorithm 4.7 can be terminated early returning the best computed result at the time of termination, sacrificing the theoretical approximation guarantees. In Section 4.5 we empirically evaluate our algorithm on real datasets and demonstrate that even for small values of r, our NNPCA algorithms significantly outperforms existing approaches.

Orthogonal NMF The NNPCA algorithm straightforwardly yields an algorithm for the ONMF problem (4.1). In an ONMF instance, the input matrix $\mathbf{M}$ is by assumption nonnegative. Given any $m \times k$ orthogonal nonnegative factor $\mathbf{W}$, the optimal choice for the second factor is $\mathbf{H} = \mathbf{M}^\top \mathbf{W}$. Hence, it suffices to determine $\mathbf{W}$, which can be obtained by solving the NNPCA problem on $\mathbf{M}$.

The proposed ONMF algorithm is outlined in Alg. 4.8. Given a nonnegative $m \times n$ matrix $\mathbf{M}$, we first obtain a rank-r approximation $\overline{\mathbf{M}}$ via the truncated SVD, where r is an accuracy parameter. Using Alg. 4.7 on $\overline{\mathbf{M}}$, we compute an orthogonal nonnegative factor $\overline{\mathbf{W}} \in \mathcal{W}_k$ that approximately maximizes (4.3) within a desired accuracy. The second ONMF factor $\overline{\mathbf{H}}$ is readily determined as described earlier.

The accuracy parameter r once again controls a trade-off between the

Algorithm 4.8 ONMFS

input $\mathbf{M}$ – $m \times n$ real nonnegative matrix
ϵ – Accuracy parameter in $(0, 1)$
r – Accuracy parameter controlling the rank of the approximation to be used.
k – target number of components

output $\overline{\mathbf{W}}, \overline{\mathbf{H}}$ – A pair of nonnegative matrices with $\overline{\mathbf{W}}$ satisfying orthogonality constraints. See Thm. 4.7

1: $\overline{\mathbf{M}} \leftarrow \mathtt{svd}(\mathbf{M}, r)$
2: $\overline{\mathbf{W}} \leftarrow$ Output of Algorithm 4.7 with input $\left(\overline{\mathbf{M}}, k, \epsilon\right)$
3: $\overline{\mathbf{H}} \leftarrow \mathbf{M}^\top \overline{\mathbf{W}}$

quality of the ONMF factors and the complexity of the algorithm. We note, however, that for any target dimension k and desired accuracy parameter ϵ, setting $r = \lceil k/\epsilon \rceil$ suffices to achieve an additive ϵ error on the relative approximation error of the ONMF problem. More formally,

Theorem 4.7. *For any $m \times n$ real nonnegative matrix $\mathbf{M}$, target dimension k, and desired accuracy $\epsilon \in (0, 1)$, Algorithm 4.8 with parameter $r = \lceil k/\epsilon \rceil$ outputs an ONMF pair $\mathbf{W}, \mathbf{H}$, such that*

$$\|\mathbf{M} - \mathbf{W}\mathbf{H}^\top\|_{\mathrm{F}}^2 \;\leq\; \mathcal{E}_\star + \varepsilon \cdot \|\mathbf{M}\|_{\mathrm{F}}^2,$$

in time $\mathrm{T}_{\mathsf{SVD}}(k/\epsilon) + O\Big((1/\epsilon)^{k^2/\epsilon} \cdot (k \cdot m)\Big)$.

Proof. (See Appendix C.1.4.) □

Theorem 4.7 implies an additive EPTAS[1] for the relative approximation error in the ONMF problem for any *constant* target dimension k; Algorithm 4.8

[1] Additive EPTAS (Efficient Polynomial Time approximation Scheme [33, 35]) refers

runs in time polynomial in the dimensions of the input $\mathbf{M}$. Finally, note that it did not require any assumption on $\mathbf{M}$ beyond nonnegativity.

4.4 The Low Rank NNPCA Algorithm

In this section, we re-visit Alg. 4.7, which plays a central role in our developments, as it is the key piece of our NNPCA and in turn our ONMF algorithm. Alg. 4.7 approximately solves the NNPCA problem (4.3) on a rank-r, $m \times n$ matrix $\overline{\mathbf{M}}$. It operates by producing a large, but tractable number of candidate solutions $\mathbf{W} \in \mathcal{W}_k$, and returns the one that maximizes the objective value in (4.2). In the sequel, we provide a brief description of the ideas behind the algorithm.

We are interested in approximately solving the low rank NNPCA problem (4.3). Let $\overline{\mathbf{M}} = \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\overline{\mathbf{V}}^\top$ denote the truncated SVD of $\overline{\mathbf{M}}$. For any $\mathbf{W} \in \mathbb{R}^{m \times k}$,

$$\|\overline{\mathbf{M}}^\top \mathbf{W}\|_{\mathrm{F}}^2 = \|\overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{W}\|_{\mathrm{F}}^2 = \sum_{j=1}^{k} \|\overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{w}_j\|_2^2 = \sum_{j=1}^{k} \max_{\mathbf{c}_j \in \mathbb{S}_2^{r-1}} \left\langle \mathbf{w}_j,\, \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\mathbf{c}_j \right\rangle^2, \quad (4.4)$$

where $\mathbb{S}_2^{r-1}$ denotes the r-dimensional ℓ_2-unit sphere. Let $\mathbf{C}$ denote the $r \times k$ variable formed by stacking the unit-norm vectors $\mathbf{c}_j$, $j = 1, \ldots, k$. The key observation is that for a given $\mathbf{C}$, we can efficiently compute a $\mathbf{W} \in \mathcal{W}_k$ that

to an algorithm that can approximate the solution of an optimization problem within an arbitrarily small additive error ϵ and has complexity that scales polynomially in the input size n, but possibly exponentially in $1/\epsilon$. EPTAS is more efficient than a PTAS because it enforces a polynomial dependency on n for any ϵ, *i.e.*, a running time $f(1/\epsilon) \cdot p(n)$, where $p(n)$ is polynomial. For example, a running time of $O(n^{1/\epsilon})$ is considered PTAS, but not EPTAS.

maximizes the right-hand side of (4.4). The procedure for that task is outlined in Alg. 4.9. Hence, the NNPCA problem (4.3) is reduced to determining the optimal value of the low-dimensional variable $\mathbf{C}$. But, first let us we provide a brief description of Alg. 4.9.

Algorithm 4.9 Projection on $\mathcal{W}_k$

input $\mathbf{A}$ – $m \times k$ real matrix

output $\widehat{\mathbf{W}}$ – $m \times k$ real nonnegative matrix with orthonormal columns, *i.e.*, $\widehat{\mathbf{W}} \in \mathcal{W}_k$. See Lemma 4.3.2.

$\mathcal{C}_W \leftarrow \{\}$
for each $\mathbf{s} \in \{\pm 1\}^k$ **do**
  $\mathbf{A}' \leftarrow \mathbf{A} \cdot \mathrm{diag}(\mathbf{s})$
  $\mathcal{I}_j \leftarrow \{\}, \quad j = 1, \ldots, k$
  for $i = 1 \ldots, m$ **do**
    $j_\star \leftarrow \arg\max_j A'_{ij}$
    if $A'_{ij_\star} \geq 0$ **then**
      $\mathcal{I}_{j_\star} \leftarrow \mathcal{I}_{j_\star} \cup \{i\}$
    end if
  end for
  $\mathbf{W} \leftarrow \mathbf{0}_{m \times k}$
  for $j = 1, \ldots, k$ **do**
    $[\mathbf{w}_j]_{\mathcal{I}_j} \leftarrow [\mathbf{a}'_j]_{\mathcal{I}_j} / \|[\mathbf{a}'_j]_{\mathcal{I}_j}\|$
  end for
  $\mathcal{C}_W \leftarrow \mathcal{C}_W \cup \mathbf{W}$
end for
$\widehat{\mathbf{W}} \leftarrow \mathrm{argmax}_{\mathbf{W} \in \mathcal{C}_W} \sum_{j=1}^k \langle \mathbf{w}_j, \mathbf{a}_j \rangle^2$

For a fixed $r \times k$ matrix $\mathbf{C}$, Algorithm 4.9 computes

$$\widehat{\mathbf{W}} \triangleq \operatorname*{argmax}_{\mathbf{W} \in \mathcal{W}_k} \sum_{j=1}^{k} \langle \mathbf{w}_j, \mathbf{a}_j \rangle^2, \tag{4.5}$$

where $\mathbf{A} \triangleq \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\mathbf{C}$. The challenge is to determine the support of the optimal solution $\widehat{\mathbf{W}}$; if an oracle revealed the optimal supports $\mathcal{I}_j$, $j = 1, \ldots, k$ of its

columns, then the exact value of the nonzero entries would be determined by the Cauchy-Schwarz inequality, and the contribution of the jth summand in (4.5) would be equal to $\sum_{i \in \mathcal{I}_j} A_{ij}^2$. Due to the nonnegativity constrains in $\mathcal{W}_k$, the optimal support $\mathcal{I}_j$ of the jth column must contain indices corresponding to only nonnegative or nonpositive entries of $\mathbf{a}_j$, but not a combination of both. Algorithm 4.9 considers all 2^k possible sign combinations for the support sets implicitly by solving (4.5) on all 2^k matrices $\mathbf{A}' = \mathbf{A} \cdot \text{diag}(\mathbf{s})$, $\mathbf{s} \in \{\pm 1\}^k$. Hence, we may assume without loss of generality that all support sets correspond to nonnegative entries of $\mathbf{A}$. Moreover, if index $i \in [m]$ is assigned to $\mathcal{I}_j$, then the contribution of the entire ith row of $\mathbf{A}$ to the objective is equal to A_{ij}^2. Based on the above, Algorithm 4.9 constructs the collection of the support sets by assigning index i to $\mathcal{I}_j$ if and only if A_{ij} is nonnegative and the largest among the entries of the ith row of $\mathbf{A}$. The algorithm runs in time[2] $O(2^k \cdot k \cdot m)$ and guarantees that the output is the optimal solution to (4.5). A more formal analysis of the Alg. 4.9 is provided in Section C.1.5.

Thus far, we have seen that any given value of $\mathbf{C}$ can be associated with a feasible solution $\mathbf{W} \in \mathcal{W}_k$ via the maximization (4.5) and Alg. 4.9. If we could efficiently consider all possible values in the (continuous) domain of $\mathbf{C}$, we would be able to recover the pair that maximizes (4.4) and, in turn, the optimal solution of (4.3). However, that is not possible. Instead, we consider a fine discretization of the domain of $\mathbf{C}$ and settle for an approximate solution.

[2] When used as a subroutine in Alg. 4.7, Alg. 4.9 can be simplified into an $O(k \cdot m)$ procedure (lines 4-14).

In particular, let $\mathcal{N}_\epsilon(\mathbb{S}_2^{r-1})$ denote a finite ϵ-net of the r-dimensional ℓ_2-unit sphere; for any point in $\mathbb{S}_2^{r-1}$, the net contains a point within distance ϵ from the former. (see Appendix C.3 for the construction of such a net). Further, let $[\mathcal{N}_\epsilon(\mathbb{S}_2^{r-1})]^{\otimes k}$ denote the kth Cartesian power of the previous net; the latter is a collection of $r \times k$ matrices $\mathbf{C}$. Alg. 4.7 operates on this collection: for each $\mathbf{C}$ it identifies a candidate solution $\mathbf{W} \in \mathcal{W}_k$ via the maximization (4.5) using Algorithm 4.9. By the properties of the ϵ-nets, it can be shown that at least one of the computed candidate solutions must attain an objective value close to the optimal of (4.3).

The guarantees of Alg. 4.7 are formally established in Lemma 4.3.2. A detailed analysis of the algorithm is provided in the corresponding proof in Appendix C.1.2. This completes the description of our algorithmic developments.

4.5 Experiments

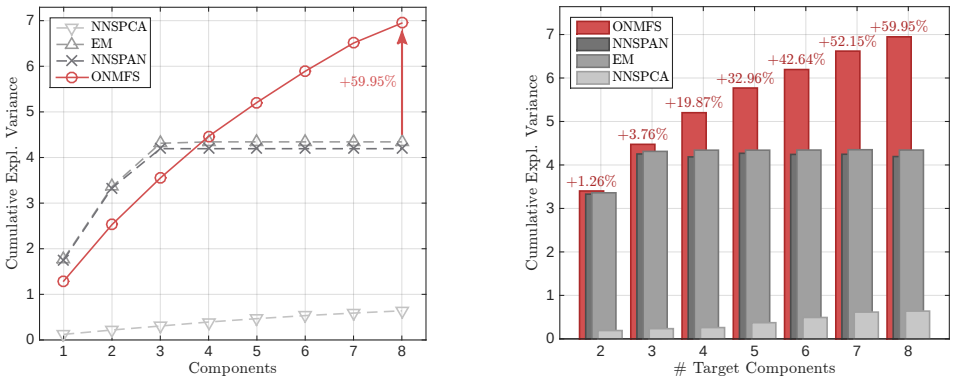
4.5.1 Experiments on NNPCA

We compare our NNPCA algorithm against three existing approaches: NSPCA [157], EM [127] and NNSPAN [16] on real datasets. NSPCA computes multiple nonnegative, but not necessarily orthogonal components; a parameter α penalizes the overlap among their supports. We set a high penalty ($\alpha = 1e10$) to promote orthogonality. EM and NNSPAN compute only a single nonnegative component. Multiple components are computed consecutively, interleaving an appropriate deflation step. To ensure orthogonality, the de-

flation step effectively zeroes out the variables used in previously extracted components. Finally, note that both the EM and NSPCA algorithms are randomly initialized. All depicted values are the best results over multiple random restarts. For our algorithm, we use a sketch of rank $r = 4$ of the (centered) input data. Further we apply an early termination criterion; execution is terminated if no improvement is observed in a number of consecutive iterations (samples). This can only hurt the performance of our algorithm.

CBCL Dataset The CBCL dataset [130] contains 2429, 19×19 pixel, gray scale face images. It has been used in the evaluation of all three methods [157, 127, 16]. We extract k orthogonal nonnegative components using all methods and compare the total explained variance, *i.e.*, the objective in (4.2). We note that input data has been centered and it is hence not nonnegative.

Fig. 4.2(a) depicts the cumulative explained variance versus the number of components for $k = 8$. EM and NNSPAN extract components greedily with a deflation step; the first component achieves high value, but subsequent ones contribute less to the total variance. On the contrary, our algorithm jointly optimizes the $k = 8$ components, achieving an approximately 60% increase in the total variance compared to the second best method. We repeat the experiment for $k = 2, \ldots, 8$. Fig. 4.2(b) depicts the total variance captured by each method for each value of k. Our algorithm significantly outperforms the existing approaches.



(a) (b)

Figure 4.2: Cumulative variance captured by k nonnegative components; CBCL Face image dataset [130]. In Fig. 4.2(a), we set $k = 8$ and plot the cumul. variance versus the number of components. EM and NNSPAN extract components greedily; first components achieve high value, but subsequent ones contribute less to the objective. Our algorithm jointly optimizes the $k = 8$ components, achieving a 59.95% improvement over the second best method. Fig. 4.2(b) depicts the cumul. variance for various values of k. We note the percentage improvement of our algorithm over the second best method.

		NSPCA	EM	NNSPAN	Ours
Amzn Com. Rev	(1500×10000)	$5.44e+01$	$7.32e+03$	$7.32e+03$	$\mathbf{7.86e+03}$ (+7.37%)
Arcence Train	(100×10000)	$4.96e+04$	$3.01e+07$	$3.00e+07$	$\mathbf{3.80e+07}$ (+26.7%)
Isolet-5	(1559×617)	$5.83e-01$	$3.54e+01$	$3.55e+01$	$\mathbf{4.55e+01}$ (+28.03%)
Leukemia	(72×12582)	$3.02e+07$	$7.94e+09$	$8.02e+09$	$\mathbf{1.04e+10}$ (+29.57%)
Mfeat Pix	(2000×240)	$2.00e+01$	$3.20e+02$	$3.25e+02$	$\mathbf{5.24e+02}$ (+61.17%)
Low Res. Spec.	(531×100)	$3.98e+06$	$2.29e+08$	$2.29e+08$	$\mathbf{2.41e+08}$ (+5.34%)
BoW:KOS	(3430×6906)	$4.96e-02$	$2.96e+01$	$3.00e+01$	$\mathbf{4.59e+01}$ (+52.95%)

Table 4.1: Total variance captured by $k = 5$ nonnegative components on various datasets [22]. For each dataset, we list (#samples × #variables) and the variance captured by each method; higher values are better. Our algorithm operates on a rank-4 sketch in all cases, and consistently achieves the best results. We note the percentage improvement over the second best method.

Additional Datasets We solve the NNPCA problem on various datasets obtained from [22]. We arbitrarily set the target number of components to $k = 5$ and configure our algorithm to use a rank-4 sketch of the input. Table 4.1 lists the total variance captured by the extracted components for each method. Our algorithm consistently outperforms the other approaches.

4.5.2 Experiments on ONMF

We compare our algorithm with several state-of-the-art ONMF algorithms *i*) the O-PNMF algorithm of [153] (for 1000 iterations), and *ii*) the more recent ONP-MF [120], and *iii*) EM-ONMF algorithms of [119] (for 1000 iterations). We also compare to clustering methods, namely vanilla and spherical k-means, since such algorithms also yield an approximate ONMF.

Synthetic data We generate a synthetic dataset as follows. We select five base vectors $\mathbf{c}_j$, $j = 1, \ldots, 5$ randomly and independently from the unit hypercube in 100 dimensions. Then, we generate data points $\mathbf{x}_i = a_i \cdot \mathbf{c}_j + p \cdot \mathbf{n}_i$, for some $j \in \{1, \ldots, 5\}$, where $a_i \sim U([0.1, 1])$, $\mathbf{n}_i \sim N(\mathbf{0}, \mathbf{I})$, and p is a parameter controlling the noise variance. Any negative entries of $\mathbf{x}_i$ are set to zero.

We vary p in $[10^{-2}, 1]$. For each p value, we compute an approximate ONMF on 10 randomly generated datasets and measure the relative Frobenius approximation error. For the methods that involved random initialization, we run 10 averaging iterations per Monte-Carlo trial. Our algorithm is configured to operate on a rank-5 sketch. Figure 4.3 depicts the relative error achieved

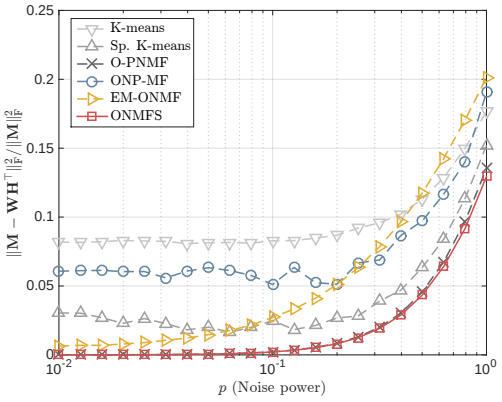


Figure 4.3: ONMF approximation error on synthetic data. The figure depicts the relative Frobenius approximation error achieved by the nonnegative factors obtained through various algorithms. Data points (samples) are generated by randomly scaling and adding noise to one of five base points that have been randomly selected from the unit hypercube in 100 dimensions. We run ONMF methods with target dimension $k = 5$. Our algorithm is labeled as ONMFS.

by each method (averaged over the random trials) versus the noise variance p. Our algorithm, labeled ONMFS achieves competitive or higher accuracy for most values in the range of p.

Real Data We apply the ONMF algorithms on various nonnegative datasets obtained from [22]. We arbitrarily set the target number of components to $k = 6$. Table 4.2 lists the relative Frobenius approximation error achieved by each algorithm. We note that on the text datasets (*e.g.*, Bag of Words [22]) we run the algorithms on the uncentered word-by-document matrix. Our algorithm performs competitively compared to other methods.

		K-MEANS	O-PNMF	ONP-MF	EM-ONMF	ONMFS
Amzn Com. Rev	(10000×1500)	0.0547	0.1153	0.1153	0.0467	**0.0462**(5)
Arcence Train	(100×10000)	0.0837	–	0.1250	0.0856	**0.0788**(4)
Mfeat Pix	(2000×240)	0.2489	0.2974	0.3074	**0.2447**	0.2615 (4)
Pems Train	(267×138672)	0.1441	0.1439	0.1380	**0.1278**	0.1283 (5)
BoW:KOS	(3430×6906)	0.8193	0.7692	0.7671	0.7671	**0.7609**(4)
BoW:Enron	(28102×39861)	0.9946	–	0.6728	0.7148	**0.6540**(4)
BoW:NIPS	(1500×12419)	0.8137	0.7277	0.7277	0.7375	**0.7252**(5)
BoW:NyTimes	(102660×3 · 10^5)	–	–	**0.9199**	0.9238	**0.9199**(5)

Table 4.2: ONMF approximation error on nonnegative matrices obtained from real datasets [22]. For each dataset, we list the size (#samples × #variables) and the relative Frobenius approximation error achieved by each method; lower values are better. We arbitrarily set the target dimension $k = 6$. Dashes (-) denote an invalid solution/non-convergence. Lower values are better. For our method, we note in parentheses the approximation rank r used.

4.6 Conclusions

We presented a novel algorithm for approximately solving the ONMF problem on a nonnegative matrix. Our algorithm relied on a new method for solving the NNPCA problem. The latter jointly optimizes multiple orthogonal nonnegative components and provably achieves an objective value close to optimal. Our ONMF algorithm is the first one to be equipped with theoretical approximation guarantees; for a constant target dimension k, it yields an additive EPTAS for the relative approximation error. Empirical evaluation on synthetic and real datasets demonstrates that our algorithms outperform or match existing approaches in both problems.

Chapter 5

PCA Along Graph Paths

We introduce a variant of (sparse) PCA in which the set of feasible support sets is determined by a graph. In particular, we consider the following special case: given input data in p dimensions and a directed acyclic graph G on p vertices corresponding to the observed variables, we seek to extract a (sparse) principal component such that the non-zero entries of the extracted principal component coincide with vertices lying along a simple path in G.

From a statistical perspective, information on the underlying network may potentially reduce the number of observations required to recover the population principal component. We consider the canonical minimax estimator which optimally exploits the prior knowledge by solving a non-convex quadratic maximization on the empirical covariance. We introduce a simple network and analyze the estimator under the spiked covariance model. We

Chapter 5 is based on material from Reference [13]: Megasthenis Asteris, Anastasios Kyrillidis, Alex Dimakis, Han-Gyol Yi, and Bharath Chandrasekaran, *"Stay On Path: PCA Along Graph Paths"*, In Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 1728–1736, 2015. The author of this dissertation is the lead author of [13], and contributed to the conception of the research problem, the theoretical and analytical developments, the experimental design and implementation, and the writing of the manuscript and its revisions. The interpretation of the experimental results on real data was provided by Han-Gyol Yi and Bharath Chandrasekaran.

show that side information potentially improves the statistical complexity.

Finally, we propose two simple algorithms to approximate the solution of the estimator and recover a component supported on the graph, and empirically evaluate them on synthetic and real datasets.

5.1 Introduction

Principal Component Analysis (PCA) is an invaluable tool in data analysis and machine learning. For a set of n centered datapoints lying in p-dimensional ambient space, represented as the columns of a matrix $\mathbf{Y} \in \mathbb{R}^{p \times n}$, the leading principal component is the first eigenvector of the empirical covariance matrix $\widehat{\mathbf{\Sigma}} = 1/n \cdot \mathbf{Y}\mathbf{Y}^\top$, or equivalently, the solution to

$$\text{(PCA)} \qquad \underset{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\|_2 = 1}{\operatorname{argmax}} \ \mathbf{x}^\top \widehat{\mathbf{\Sigma}} \mathbf{x}. \tag{5.1}$$

The principal component spans the direction of maximum data variability. This direction is a linear combination of all p observed variables; in other words, the PC vectors are typically non-sparse. However, it is often desirable to obtain a principal component with specific structure, for example limiting the support of non-zero entries. From a statistical viewpoint, in the high dimensional regime $n = O(p)$, the recovery of the true (population) principal component is only possible if additional side-information is available; for example, we may know that the leading principal component has a limited number of nonzero entries, *i.e.*, it is a sparse vector [7, 142].

The leading sparse principal component is the solution to

$$\underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\, \mathbf{x}^\top \widehat{\mathbf{\Sigma}}\mathbf{x}, \tag{5.2}$$

where

$$\mathcal{X} \triangleq \{\mathbf{x}\in\mathbb{R}^p : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_0 \leq s\}, \tag{5.3}$$

with the parameter s controlling the number of nonzero entries in the extracted component. The above estimator is optimal in terms of the statistical rate of convergence [142] (with respect to a natural loss function defined by the Frobenious distance of the projection matrices on the principal direction), *i.e.*, it optimally exploits the side information on the sparsity. However, it is NP-hard (by a reduction from maximum clique problem), and hence computationally intractable in general.

Path PCA In this chapter, we move beyond sparsity, to higher order structure. We consider the case where the desired principal component is not only sparse, but its support satisfies additional restrictions captured by an underlying network, given as side-information along with the observed data. In particular, motivated primarily by a neuroscience application, we consider the following special case, hereafter referred to as *Graph Path PCA*, or simply *Path PCA*. Consider a directed acyclic graph (DAG) $G = (V, E)$ on p vertices corresponding to the ambient dimension of the input data. Let S and T be two additional special vertices and consider all simple paths from S to T on the graph G. Ignoring the order of vertices along a path, let $\mathcal{P}(G)$ denote

the collection of all S-T paths in G. We seek a sparse principal component whose support corresponds to vertices lying along a simple path in G, *i.e.*, the solution to

$$\text{(PathPCA)} \qquad \underset{\mathbf{x}\in\mathcal{X}(G)}{\operatorname{argmax}}\, \mathbf{x}^\top \widehat{\mathbf{\Sigma}}\mathbf{x}, \tag{5.4}$$

where

$$\mathcal{X}(G) \triangleq \{\mathbf{x}\in\mathbb{R}^p : \|\mathbf{x}\|_2 = 1,\ \text{supp}(\mathbf{x}) \in \mathcal{P}(G)\}. \tag{5.5}$$

A natural question arises regarding the usefulness of the above Path PCA formulation. We will argue that it can be used to impose several interesting types of structure. Note that the data covariance matrix and the underlying graph can be arbitrary: the matrix captures data correlations, while the graph is a mathematical tool to efficiently describe the possible supports of interest. We illustrate this through a few applications.

Multiple Choice PCA: The Financial Model Example Consider a dataset formed by observing the stock prices of the companies listed in the S&P500 index over a large time window. We wish to identify a small number of companies that can form a direction of maximum data variability. Running Sparse PCA with a sparsity parameter s will select s companies that maximize explained variance under this cardinality constraint. However, it may be useful to enforce more structure. For example, the listed companies can be conceptually grouped into a few business sectors (*e.g.*, Energy, Health

Care,*etc.*). If we are forced to choose at most one company from each sector, *how could we identify the best representatives?*

More abstractly, given a partition of the observed variables, we may wish to identify a sparse principal component in which the nonzero variables correspond to distinct parts. In other words, each segment of the partition is represented in the principal component by a single variable; this is a *Multiple Choice PCA* problem.

In Section 5.5, we show that Multiple Choice PCA can be encoded using our graph path framework. We compare our variable selection with that of Sparse PCA in the context of the aforementioned finance model selection example and show that it leads to interpretable results.

Biological and fMRI Networks Several problems involve variables that are naturally connected in a network. In these cases, Path PCA can enforce interpretable sparsity structure, especially when the starting and ending points are manually selected by domain experts. In section 5.5, we apply our algorithm on fMRI data using a graph on regions of interest (ROIs) based on the Harvard-Oxford brain structural atlas [54].

We emphasize that our applications to brain data is preliminary: the directional graphs we extract are simply based on distance and should not be interpreted as causality; rather, simply as a way of encoding desired supports.

What can we say about the tractability of (5.4)*?* Note that despite the additional constraints on the sparsity patterns, the number of admissible sup-

port sets (*i.e.*, the collection of S-T paths in the graph G) can be exponential in p, the number of variables. For example, consider the following graph G: all vertices except S and T are grouped in $(p/2-1)$ pairs; the source vertex S is connected to the two vertices of the first pair, both of which are in turn connected to the two vertices of the second pair and so on, and finally, the two vertices of the last pair are connected to the terminal vertex T. Clearly, there are by construction $2^s \approx 2^{p/2}$ S-T paths and therefore a brute-force search over all admissible support sets is not tractable. It turns out that the Path PCA optimization (5.4) is NP-hard. In Section D.1, we provide a proof based on a multi-step reduction from the k-clique problem. The proof implies the same hardness result for Multiple Choice PCA, which is a special case of Path PCA.

Our Contributions

1. From a statistical viewpoint, we show that side information on the underlying graph G can potentially improve the statistical complexity of recovering the true principal component $\mathbf{x}_\star \in \mathcal{X}(G)$ via (5.4). For our analysis, we introduce a simple, sparsity-inducing network model: the network is defined on p vertices partitioned into s layers, with edges from one layer to the next, and maximum out-degree upper bounded by a parameter d (see path:appendix:pathpca-hardnessFig. 5.1). We show that $n = O(\log {}^{p}/_{s} + s \log d)$ observations $\mathbf{y}_i \sim N(\mathbf{0}, \mathbf{\Sigma})$, suffice to obtain an arbitrarily good estimate via (5.4) with respect to a natural choice for

the loss function. Our analysis follows the steps of [142] for the sparse PCA minimax estimator.

2. We complement the above with an information-theoretic lower bound on the minimax estimation error under the spiked covariance model. Here, the latent signal is assumed to be a vector supported along a path of the given graph. The lower bound matches the aforementioned upper bound on the statistical complexity.

3. We propose two algorithms for approximating the solution of (5.4), based on those of [155] and [113, 16] for the sparse PCA problem. We empirically evaluate our algorithms on synthetic and real datasets.

5.2 Related Work

There is a large volume of work on algorithms and the statistical analysis of sparse PCA [77, 160, 49, 51, 77, 142, 7]. On the contrary, there is limited work that considers additional structure on the sparsity patterns. Motivated by a face recognition application, [75] introduce *structured sparse PCA* using a regularization that encodes higher-order information about the data. The authors design sparsity inducing norms that further promote a pre-specified set of sparsity patterns. Of course, we note that the idea of pursuing additional structure on top of sparsity is not limited to PCA: Model-based compressive sensing seeks sparse solutions under a restricted family of sparsity patterns [24, 26, 92], while structure induced by an underlying network is found in [107] for sparse

linear regression.

Finally, we remark that on a technical level, our subsequent statistical analysis of the minimax estimation error relies heavily on the work of Vu and Lei [142], which analyzes the minimax rates of estimation for sparse PCA in high dimensions.

5.3 Minimax Estimation Error for Path PCA

We analyze the minimax estimation error for Path PCA, *i.e.*, the problem of recovering the leading eigenvector of the covariance matrix of the population from which the observed data samples originate under the assumption that the leading eigenvector has a support belonging to the collection of supports induced by a given graph G as described in (5.5). The objective is to quantify the difficulty of the estimation problem and the limitations of statistical inference imposed on any estimator, as well as characterize the performance of specific estimators with respect to these fundamental statistical limits. Our subsequent analysis follows closely the steps of Vu and Lei [142] for the analysis of minimax rates of estimation in the case of sparse PCA.

To analyze the minimax estimation error, we need to first specify two main components. The first component is a generative model for the observed data. In turn, this requires specifying the collection of the parameter to be estimated, *i.e.*, the leading eigenvector of the population matrix, as well as the random process (or more generally a class of random distributions) that generates the observed samples for a given parameter. The second component

is of course the loss function that captures the estimation error.

5.3.1 Loss Function

If $\mathbf{x}_\star$ denotes the true leading eigenvector of the population covariance matrix, then a natural choice for the evaluation of any estimator $\widehat{\mathbf{x}}$ is the ℓ_2-distance $\|\mathbf{x}_\star - \widehat{\mathbf{x}}\|_2$. This loss function, however, has a two-fold weakness. First, it fails to capture the sign ambiguity for $\mathbf{x}_\star$; in other words, $\widehat{\mathbf{x}}$ and $-\widehat{\mathbf{x}}$ should be equally good estimators. Secondly, although beyond the scope or our analysis, it cannot be straightforwardly extended to the problem of estimating a principal subspace of dimension $k > 1$ due to the fact that the basis of a k-dimensional subspace is not unique, and the non-uniqueness is not limited to a sign ambiguity.

To mitigate these issues, we resort to the loss function

$$\ell(\mathbf{x}_\star, \widehat{\mathbf{x}}) = \left\| \mathbf{x}_\star \mathbf{x}_\star^\top - \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top \right\|_{\mathrm{F}}, \tag{5.6}$$

which captures the distance between the projection matrices onto the corresponding one-dimensional subspaces spanned by $\mathbf{x}_\star$ and $\widehat{\mathbf{x}}$, respectively. It removes the sign ambiguity for the one-dimensional case and generalizes to the problem of estimating high-dimensional subspaces since the projection matrices are unique. Finally, note that for the one-dimensional estimation problem, that loss function lies within a small constant factor from the ℓ_2-distance (See [142], Lemma A.1.2).

5.3.2 Data Model

The second component required for our analysis is a data generative model, which in turn, this requires specifying the collection of the parameter to be estimated, *i.e.*, the leading eigenvector of the population matrix, as well as the random process (or more generally a class of random distributions) that generates the observed samples for a given parameter. In our Path PCA problem, the set of permissible vectors is induced by a given DAG G defined on the p variables: it contains all vectors with unit ℓ_2-norm whose support corresponds to a path on the graph. Hence, to specify the parameter space, we need to adopt a model for the underlying network.

The Layer Graph Consider a directed acyclic graph $G = (V, E)$ on p vertices, with the following structure:

1. The set of vertices V contains a special source vertex S and a terminal vertex T. The remaining $p-2$ vertices in $V \backslash \{S, T\}$ are partitioned into s disjoint subsets (*layers*) $\mathcal{L}_1, \ldots, \mathcal{L}_s$, *i.e.*, $\bigcup_i \mathcal{L}_i = \widehat{V}$, and $\mathcal{L}_i \cap \mathcal{L}_j = \emptyset$, $\forall i, j \in [s]$, $i \neq j$. For simplicity, we will further assume that $p - 2$ is a multiple of s and in turn all layers have exactly the same size.

2. The source vertex S is connected to all vertices of the first layer, *i.e.*, $\Gamma_{\text{out}}(S) = \mathcal{L}_1$, where $\Gamma_{\text{out}}(v)$ denotes the out-neighborhood of v. Similarly, all vertices of the sth layer are in the in-neighborhood of the terminal node T, *i.e.*, $\Gamma_{\text{in}}(S) = \mathcal{L}_s$.

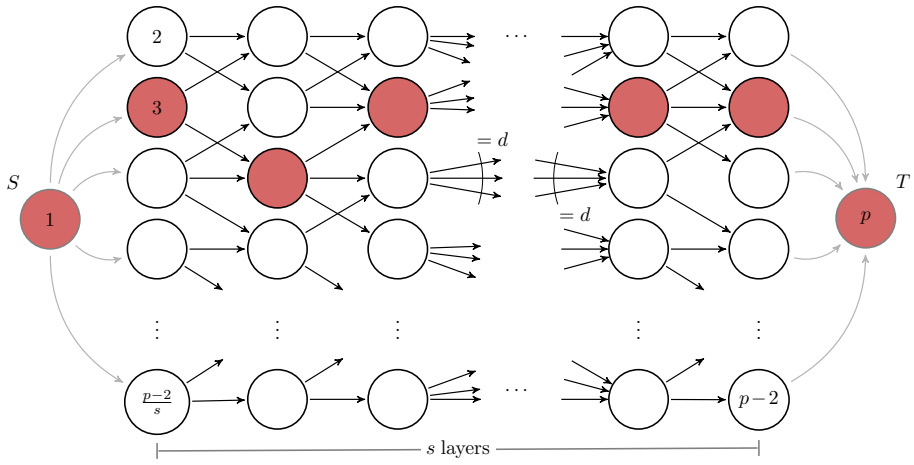


Figure 5.1: The $((p, s, d)$-layer graph $G = (V, E)$ used for the theoretical analysis of Path PCA. G is a directed acyclic graph p vertices. One vertex is considered a source vertex S and another is a terminal vertex T. The remaining $p - 2$ vertices are partitioned into s disjoint sets (layers) $\mathcal{L}_1, \ldots, \mathcal{L}_s$. S is connected to all vertices in the first layer, and all vertices of the last layer are connected to T. Each of the remaining intermediate vertices is connected to d vertices of the subsequent layer. The in-degree of each vertex is also bounded by d. The highlighted vertices form an S-T path.

3. Each of the remaining $p-2$ vertices is connected to (at most) d vertices of the subsequent layer, that is $\Gamma_{\text{out}}(v) \subset \mathcal{L}_{i+1}$, $\forall v \in \mathcal{L}_i$, for $i = 1, \ldots, s-1$. Further, for simplicity, we assume that both the in-degree and out-degree of each of the intermediate vertices is upper bounded by a parameter d, that is, $|\Gamma_{\text{out}}(v)| \leq d$, $\forall v \in \mathcal{L}_i$, $i = 1, \ldots, s-1$, and $|\Gamma_{\text{in}}(v)| \leq d$, $\forall v \in \mathcal{L}_i$, $i = 2, \ldots, s$. In words, the edges from one layer are maximally spread across the vertices of the next.

We refer to G as a $((p, s, d)$-*layer graph*. The graph is illustrated in Fig. 5.1. The highlighted vertices in Fig. 5.1 form an S-T path. Observe that by con-

struction any path from the source to the terminval (excluding those two vertices) has length exactly equal to s. Recall that in our Path PCA problem, the feasible region contains vectors whose supports corresponds to S-T paths in G. Hence, the number s of layers directly controls the number of nonzero entries in the solution of Path PCA problem defined on that graph. Similarly, the parameter d, which upper bounds the (in- and out-) degree of intermediate vertices, effectively controls the size of the collection $\mathcal{P}(G)$ of S-T paths in the graph. In particular,

$$|\mathcal{P}(G)| = |\mathcal{L}_1| \cdot d^{s-1} = \frac{p-2}{s} \cdot d^{s-1}. \tag{5.7}$$

Finally, note that since $d \leq (p-2)/s$, it follows from (5.7) that $|\mathcal{P}(G)| \leq \binom{p-2}{s}$, *i.e.*, the number of all possible supports of cardinality s. Intuitively, this is the reason why the underlying graph structure may improve the statistical complexity compared to vanilla sparse PCA.

The motivation behind the choice of the $(p,\ s,\ d)$-layer graph for the data generative model is multi-fold. It is a simple and natural model. In several physical or biological systems, variables are arranged in a grid in the 3-dimensional space and variable interactions tend to be spatially local. Such simple grid structures are straightforwardly captured by our model. The model further allows a straightforward comparison with sparse PCA. As explained above, the network induces a feasible region containing s-sparse vectors. At the same time, it does not make the problem easy; In fact, our proof of the NP-hardness of the Path PCA problem, is based on an instance defined on a (p, s, d)-layer graph.

Spiked Covariance Model Having established the parameter space, we need to determine the class of distributions under which our observed samples are generated. We consider (a generalizaton of) the *spiked covariance model*, as in the sparse PCA literature [77, 8], except instead of sparsity, we impose the path constraints encoded by a (known) underlying graph G.

Let $\mathbf{x}_\star$ be a unit ℓ_2-norm p-dimensional vector. For $\lambda_1, \lambda_2 \in \mathbb{R}$, with $\lambda_1 > \lambda_2 \geq 0$, let

$$\mathbf{\Sigma} = \lambda_1 \mathbf{x}_\star \mathbf{x}_\star^\top + \lambda_2 \mathbf{\Sigma}_0 \tag{5.8}$$

for some $p \times p$ PSD matrix $\mathbf{\Sigma}_0$ with $\|\mathbf{\Sigma}_0\|_2 = 1$. We will consider the class of distributions whose second moment is of the form (5.8) for a latent signal $\mathbf{x}_\star$ belonging to $\mathcal{X}(G)$ for a (p, s, d)-layer graph G. In particular, we will assume that the n observations (samples) $\{\mathbf{y}_i\}_{i=1}^n \in \mathbb{R}^p$ are indepedently drawn according to such a distribution from a population with covariance matrix $\mathbf{\Sigma}$.

Especially for the second part of our analysis, *i.e.*, the upper bound on the minimax estimation error, we will need a stricter technical assumption, in order to bound the tails of the norm of the observations.

Assumption 1. *There exist i.i.d. random vectors* $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^p$*, such that* $\mathbb{E}\mathbf{z}_i = \mathbf{0}$ *and* $\mathbb{E}\mathbf{z}_i\mathbf{z}_i^\top = \mathbf{I}_p$*,*

$$\mathbf{y}_i = \mu + \mathbf{\Sigma}^{1/2}\mathbf{z}_i \quad \textit{and} \quad \sup_{\mathbf{x} \in \mathbb{S}_2^{p-1}} \|\mathbf{z}_i^\top \mathbf{x}\|_{\psi_2} \leq K, \tag{5.9}$$

where $\mu \in \mathbb{R}^p$ *and* $K > 0$ *is a constant depending on the distribution of* $\mathbf{z}_i$*s.*

Here, $\|\cdot\|_{\psi_\alpha}$ denote the Orlicz ψ_α-norm; for a random variable Y, the Orlicz ψ_α-norm is $\|Y\|_{\psi_\alpha} = \inf\{c > 0 : \mathbb{E}\exp(|Y/c|^\alpha) \leq 2\}$. Random variables with finite ψ_α-norm correspond to those whose tails are bounded by $f(x) = \exp(-Cx^\alpha)$.

5.3.3 Main Results

Our objective is to characterize the minimax estimation error

$$\min_{\widehat{\mathbf{x}}} \max_{\mathbf{x}_\star, \mathcal{D}(\mathbf{x}_\star)} \mathbb{E}_{\mathcal{D}(\mathbf{x}_\star)} \left\| \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top \right\|_{\mathrm{F}} \tag{5.10}$$

where the minimum is taken over all estimators $\widehat{\mathbf{x}}$ which rely exclusively on the collection of the n observed datapoints, while the maximum is over all permissible values of the parameter $\mathbf{x}_\star$, and more precisely over all distributions that satisfy the criteria described in the previous subsection.

In particular, we develop (tight) non-asymptotic upper and lower bounds on the minimax estimation error, which explicitly depend on the number n of observed samples, the ambient dimension p and of course the remaining parameters of the model (e.g., the parameters of the underlying network). The lower bound is information theoretic and relies on Fano's inequality. Note that to obtain a lower bound, it suffices to focus on a specific distribution in the allowed class. The upper bound is based on the analysis of a specific estimator (but must hold for all distributions in the class). Namely, we consider the estimator obtained by solving the constrained quadratic maximization (5.4) with argument the empirical covariance matrix of the available observations, over

a feasible set induced by the aforementioned layer graph. Although that estimato is computationally intractable, we show that it yields an upper bound on the minimax error that nearly matches the information theoretic lower bound.

5.3.3.1 Lower Bound

Theorem 5.8 (Lower Bound)**.** *Consider a (p, s, d)-layer graph G defined on p vertices, with $s \geq 4$ layers, and degree bound d such that $\log d \geq 4H(3/4)$ (Note that by construction the parameters must also satisfy $p - s \cdot d \geq 2$), and let $\mathbf{x}_\star$ be a vector in the feasible set $\mathcal{X}(G)$ induced by G as in (5.5). Let $\{\mathbf{y}_i\}_{i=1}^n$ be a sequence of n random observations, independently drawn according to probability density function*

$$\mathcal{D}_p(\mathbf{x}_\star) = \mathcal{N}\Big(\mathbf{0}, \mathbf{I}_p + \beta \cdot \mathbf{x}_\star \mathbf{x}_\star^\top\Big),$$

for some $\beta > 0$. Let $\mathcal{D}_p^{(n)}(\mathbf{x}_\star)$ denote the product measure over the n independent draws. Consider the problem of estimating $\mathbf{x}_\star$ from the n observations, given G as side-information. There exists $\mathbf{x}_\star \in \mathcal{X}(G)$ such that for every estimator $\widehat{\mathbf{x}}$,

$$\mathbb{E}_{\mathcal{D}_p^{(n)}(\mathbf{x}_\star)}[\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)] \geq \frac{1}{2\sqrt{2}} \cdot \min\left\{1, \left(\frac{C' \cdot (1+\beta)}{\beta^2} \cdot \frac{1}{n}\Big(\log \frac{p-2}{s} + \frac{s}{4}\log d\Big)\right)^{1/2}\right\}. \quad (5.11)$$

Theorem 5.8 states that for some latent signal $\mathbf{x}_\star \in \mathcal{X}(G)$, and observations generated according to the spiked covariance model, the minimax

estimation error is bounded away from zero, unless the number of samples is

$$n = \Omega\Big(\log \frac{p}{s} + s \log d\Big). \tag{5.12}$$

Note that although the lower bound is derived based on a specific distribution for the observed data, it is a lower bound for the entire class described in the previous section, since the distribution described in the theorem is a member of that class. Finally, we compare that with the $n = \Theta\Big(s \log \frac{p}{s}\Big)$ samples that are necessary and sufficient to recover the true s-sparse leading eigenvector in the case of Sparse PCA [142]. In conjuction with the fact that the bound in (5.12) is tight as we will show subsequently, depending on the scaling of the parameters s (sparsity/number of layers) and d (degree upper bound) the side-information of the underlying network G can potentially reduce the statistical complexity of recovering the desired eigenvector.

In the sequel, we provide a sketch proof of Theorem 5.8, following the steps of [142]. The key idea is to discretize the feasible space $\mathcal{X}(G)$ (*i.e.*, the parameter space) and utilize the Generalized Fano Inequality [154] to derive the desired lower bound on the minimax estimation error. The next lemma summarizes Fano's Inequality for the special case in which the n observations are distibuted according to the n-fold product measure $\mathcal{D}_p^{(n)}(\mathbf{x}_\star)$:

Lemma 5.3.3 (Generalized Fano [154])**.** *Let $\mathcal{X}_\epsilon \subset \mathcal{X}(G)$ be a finite set of points $\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{X}_\epsilon|} \in \mathcal{X}(G)$. Each point $\mathbf{x}_i$ implicitly yields a probability measure $\mathcal{D}_p^{(n)}(\mathbf{x}_i)$ on the n observations. If for all pair of points $\mathbf{x}_i$, $\mathbf{x}_j$, $i \neq j$, it*

holds that

$$d(\mathbf{x}_i, \mathbf{x}_j) \geq \alpha \tag{5.13}$$

for some pseudo-metric[1] $d(\cdot, \cdot)$ *and some* $\alpha > 0$*, and further Kullback-Leibler divergence between the corresponding measures* $\mathcal{D}_p^{(n)}(\mathbf{x}_i)$ *and* $\mathcal{D}_p^{(n)}(\mathbf{x}_j)$ *satisfies*

$$\mathrm{KL}\Big(\mathcal{D}_p^{(n)}(\mathbf{x}_i) \parallel \mathcal{D}_p^{(n)}(\mathbf{x}_j)\Big) \leq \gamma,$$

for some $\gamma > 0$*, then for any estimator* $\widehat{\mathbf{x}}$

$$\max_{\mathbf{x}_i \in \mathcal{X}_\epsilon} \mathbb{E}_{\mathcal{D}_p^{(n)}(\mathbf{x}_i)}[d(\widehat{\mathbf{x}}, \mathbf{x}_i)] \geq \frac{\alpha}{2} \cdot \left(1 - \frac{\gamma + \log 2}{\log |\mathcal{X}_\epsilon|}\right). \tag{5.14}$$

Theorem 5.8 will be a result of applying the Generalized Fano's Inequality with

$$d(\widehat{\mathbf{x}}, \mathbf{x}) \triangleq \left\| \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}\mathbf{x}^\top \right\|_{\mathrm{F}}, \tag{5.15}$$

i.e., using our loss function as a pseudo-metric in Lemma 5.3.3. But first, we need to show the existence of a sufficiently large set $\mathcal{X}_\epsilon \subseteq \mathcal{X}(G)$ such that *i*) the points in $\mathcal{X}_\epsilon$ are well separated under the above pseudo-metric $d(\cdot, \cdot)$, while *ii*) the KL divergence of the induced probability measures is upper appropriately bounded.

Lemma 5.3.4 (Local Packing)**.** *Consider a* (p, s, d)*-layer graph* G *on* p *vertices with* $s \geq 4$ *and* $\log d \geq 4 \cdot H(3/4)$ *(i.e., the same assumptions as in*

[1] A pseudometric on a set $\mathcal{X}$ is a function $d : \mathcal{Q}^2 \to \mathbb{R}$ that satisfies all properties of a distance (non-negativity, symmetry, triangle inequality) except the identity of indiscernibles: $d(\mathbf{q}, \mathbf{q}) = 0$, $\forall \mathbf{q} \in \mathcal{Q}$ but possibly $d(\mathbf{q}_1, \mathbf{q}_2) = 0$ for some $\mathbf{q}_1 \neq \mathbf{q}_2 \in \mathcal{Q}$.

Thm. 5.8). For any $\epsilon \in (0,1]$, there exists a set $\mathcal{X}_\epsilon \subset \mathcal{X}(G)$ such that

$$\epsilon/\sqrt{2} \;<\; \|\mathbf{x}_i - \mathbf{x}_j\|_2 \;\leq\; \sqrt{2}\cdot\epsilon,$$

for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_\epsilon$, $\mathbf{x}_i \neq \mathbf{x}_j$, and

$$\log|\mathcal{X}_\epsilon| \geq \log\frac{p-2}{s} + \frac{1}{4}\cdot s\cdot\log d.$$

Proof. The proof of the lemma relies on developing a modified version of the Varshamov-Gilbert Lemma adapted to our specific model, *i.e.*, the set of characteristic vectors of the S-T paths of a (p,s,d)-layer graph G. The proof is given in Appendix D.2. □

Consider a set of points $\mathcal{X}_\epsilon \subset \mathcal{X}(G)$ with the properties described in Lemma 5.3.4. Taking into account the fact that for any two unit ℓ_2-norm vectors $\mathbf{x}$, $\mathbf{y}$ such that $\|\mathbf{x}-\mathbf{y}\|_2 \leq \sqrt{2}$ it holds that $\|\mathbf{x}\mathbf{x}^\top - \mathbf{y}\mathbf{y}^\top\|_{\mathrm{F}} \geq \|\mathbf{x}-\mathbf{y}\|_2$ (See [142], Lemma A.1.2), we have that for any two distinct points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_\epsilon$,

$$d^2(\mathbf{x}_i,\mathbf{x}_j) = \|\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{x}_j\mathbf{x}_j^\top\|_{\mathrm{F}}^2 > \frac{\epsilon^2}{2} \tag{5.16}$$

Moreover, for the KL divergence between the distributions $\mathcal{D}_p(\mathbf{x}_i)$ and $\mathcal{D}_p(\mathbf{x}_j)$ for two distinct points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_\epsilon$, we have

$$\begin{aligned}
&\mathrm{KL}(\mathcal{D}_p(\mathbf{x}_i) \parallel \mathcal{D}_p(\mathbf{x}_j)) \\
&\quad= \frac{\beta}{2(1+\beta)}\cdot\Big[(1+\beta)\times \mathrm{Tr}\Big(\Big(\mathbf{I}-\mathbf{x}_j\mathbf{x}_j^\top\Big)\mathbf{x}_i\mathbf{x}_i^\top\Big) - \mathrm{Tr}\Big(\mathbf{x}_j\mathbf{x}_j^\top(\mathbf{I}-\mathbf{x}_i\mathbf{x}_i^\top)\Big)\Big] \\
&\quad= \frac{\beta^2}{4(1+\beta)}\cdot\|\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{x}_j\mathbf{x}_j^\top\|_{\mathrm{F}}^2 \\
&\quad\leq \frac{\beta^2}{(1+\beta)}\cdot\|\mathbf{x}_i-\mathbf{x}_j\|_2^2,
\end{aligned}$$

and in turn, for the n-fold product distribution,

$$\mathrm{KL}\Big(\mathcal{D}_p^{(n)}(\mathbf{x}_i) \parallel \mathcal{D}_p^{(n)}(\mathbf{x}_j)\Big) \leq \frac{n\beta^2}{(1+\beta)}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \leq \frac{2n\beta^2\epsilon^2}{(1+\beta)}, \tag{5.17}$$

where the last inequality follows by the fact that $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \sqrt{2} \cdot \epsilon$.

The lower bound (5.16) on the pseudo-metric $d(\cdot, \cdot)$ and the upper bound (5.17) on the KL-divergence implicitly determine a value for the parameters α and γ of Lemma 5.3.3. Substituting those into (5.14), we find

$$\max_{\mathbf{x}_i \in \mathcal{X}_\epsilon} \mathbb{E}_{\mathcal{D}_p^{(n)}(\mathbf{x}_i)}[d(\widehat{\mathbf{x}}, \mathbf{x}_i)] \geq \frac{\epsilon}{2\sqrt{2}}\left[1 - \frac{n\frac{2\epsilon^2\beta^2}{(1+\beta)} + \log 2}{\log|\mathcal{X}_\epsilon|}\right], \tag{5.18}$$

for any $\epsilon \in (0, 1]$.

Inequality (5.18) resembles substantially the desired lower bound (5.11). To get that desired result, it suffices to choose the appropriate value of ϵ. For convenience, let $B = \log((p-2)/s) + 1/4 \cdot s \log d$ denote the lower bound on the log-cardinality of $\mathcal{X}_\epsilon$ given in Lemma 5.8, that is, $\log|\mathcal{X}_\epsilon| \geq B$. We set

$$\epsilon^2 = \min\left\{1, \ \frac{1}{8} \cdot \frac{1}{n} \cdot \frac{(1+\beta)}{\beta^2} \cdot B\right\}. \tag{5.19}$$

For this choice of ϵ, we have

$$n \cdot \frac{2\epsilon^2\beta^2}{(1+\beta)} \frac{1}{\log|\mathcal{X}_\epsilon|} \leq \frac{1}{4}. \tag{5.20}$$

To verify that, observe that regardless of which of the two quantities in the RHS of (5.19) is the smallest, we have that

$$\epsilon^2 \leq \frac{1}{8} \cdot \frac{1}{n} \cdot \frac{(1+\beta)}{\beta^2} \cdot B,$$

from which (5.20) immediately follows taking into account that $\log|\mathcal{X}_\epsilon| \geq B$. Furthermore, by Lemma 5.3.4, and under the assumptions of Thm. 5.8 on the parameters s and d (note that $p-2 \geq k \cdot d$ by the structure of G), we have

$$\log|\mathcal{X}_\epsilon| \geq \log\frac{p-2}{s} + \frac{s}{4}\cdot\log d \geq 4\cdot H(3/4) \geq 4\log 2. \tag{5.21}$$

Combining (5.20) and (5.21), inequality in (5.18) implies that

$$\max_{\mathbf{x}_i\in\mathcal{X}_\epsilon} \mathbb{E}_{\mathcal{D}_p^{(n)}(\mathbf{x}_i)}[d(\widehat{\mathbf{x}},\mathbf{x}_i)] \geq \frac{1}{2\sqrt{2}}\cdot\epsilon. \tag{5.22}$$

Substituting ϵ in the RHS of the last inequality according to (5.19) implies the desired result (5.11), completing the proof of Theorem 5.8.

5.3.3.2 Upper bound

Up to this point, we have developed an information theoretic lower bound on the minimax estimation error. That bound is oblivious to the estimator; in principle, it may be unatainable by any estimator. In this section, we derive an upper bound on the minimax estimation error, by analyzing the (statistical) performance of a specific estimator: the one obtained by solving the constrained quadratic maximization (5.4) on a constraint set induced by the given (p,s,d)-layer graph G. This upper bound effectively implies that there exists an estimator (although in our case computationally intractable) which can recover the true eigenvector of the population covariance matrix using approximately as few samples as required by the funamental statistical limit. In particular, we will show the following.

Theorem 5.9 (Upper bound)**.** *Consider a (p, s, d)-layer graph G and a vector $\mathbf{x}_\star \in \mathcal{X}(G)$. Let $\{\mathbf{y}_i\}_{i=1}^n$ be a sequence of n indepdendent observations, identically distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} \succeq \mathbf{0}$ with eigenvalues $\lambda_1 > \lambda_2 \geq \ldots$, and principal eigenvector $\mathbf{x}_\star$. Further, let $\widehat{\mathbf{\Sigma}}$ be the empirical covariance of those n samples and $\widehat{\mathbf{x}}$ be the estimate of $\mathbf{x}_\star$ obtained via (5.4) for the given graph G. For the loss function $\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star) \triangleq \|\widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star\mathbf{x}_\star^\top\|_{\mathrm{F}}$, we have*

$$\mathbb{E}[\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)] \leq C \cdot \frac{\lambda_1}{\lambda_1 - \lambda_2} \cdot \frac{1}{n} \cdot \max\Big\{\sqrt{nA},\, A\Big\},$$

where C is a positive constant and

$$A = O(\log((p-2)/s) + s \log d).$$

Here, the expectation is over the randomness of the observations.

Theorem 5.9 effectively determines the scaling of n –the number of samples– with respect to the remaining parameters that suffices to obtain an arbitrarily small expected error. In the sequel, we provide a sketch proof of Theorem 5.9. Our proof relies heavily on the proof of [142] for a similar upper bound for the sparse PCA problem, but we reproduce most of the steps for completeness.

Lemma 5.3.5 (Lemma 3.2.1 [142])**.** *Let $\mathbf{\Sigma}$ be a $p \times p$ symmetric, positive semidefinite matrix whose principal eigenvector is $\mathbf{x}_\star$. For any $\widetilde{\mathbf{x}} \in \mathbb{R}^p$ such that $\|\widetilde{\mathbf{x}}\|_2 = 1$, we have*

$$\frac{\lambda_1 - \lambda_2}{2} \cdot \|\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top - \mathbf{x}_\star\mathbf{x}_\star^\top\|_{\mathrm{F}}^2 \;\leq\; \Big\langle \mathbf{\Sigma},\, \mathbf{x}_\star\mathbf{x}_\star^\top - \widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top \Big\rangle.$$

Let $\widehat{\mathbf{x}}$ be an estimator of $\mathbf{x}_\star$ obtained via solving the constrained quadratic maximization (5.4). The solution $\widehat{\mathbf{x}}$ belongs to the feasible region $\mathcal{X}(G)$ and is a unit ℓ_2-norm vector. By Lemma 5.3.5, it follows (see [142]; Sec 3.2) that

$$\begin{aligned}\frac{\lambda_1 - \lambda_2}{2} \cdot \ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)^2 &\leq \left\langle \boldsymbol{\Sigma},\, \mathbf{x}_\star \mathbf{x}_\star^\top - \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top \right\rangle \\ &\leq \left\langle \widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma},\, \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top \right\rangle. \end{aligned} \tag{5.23}$$

Recall that both $\mathbf{x}_\star$ and its estimate $\widehat{\mathbf{x}}$ are supported on (potentially different) sets that belong to $\mathcal{P}(G)$, the set of $S-T$ paths in G. Hence, for the inner produt in the RHS of (5.23), the only entries of $\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}$ that are relevant, are those in the rows and columns indexed by the union of the two supports. More formally, let $S = \mathrm{supp}(\mathbf{x}_\star) \cap \mathrm{supp}(\widehat{\mathbf{x}})$ and Π_S be the diagonal matrix that takes the value 1 in the ith diagonal entry if $i \in S$. Further, define $\mathcal{P}^2(G)$ to be the collection of sets formed by the union of two sets in $\mathcal{P}(G)$; for example, $S \in \mathcal{P}^2(G)$. Similarly, define $\mathcal{X}^2(G)$ as the set of unit ℓ_2-norm vectors supported in $\mathcal{P}^2(G)$. Then, continuing from (5.23), we have

$$\begin{aligned}\frac{\lambda_1 - \lambda_2}{2} \cdot \ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)^2 &\leq \left\langle \widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma},\, \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top \right\rangle \\ &= \left\langle \Pi_S\big(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\big)\Pi_S,\, \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top \right\rangle \\ &\leq \left\| \Pi_S\big(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\big)\Pi_S \right\|_2 \cdot \mathrm{Tr}\big(\widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top\big) \\ &\leq \left\| \Pi_S\big(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\big)\Pi_S \right\|_2 \cdot \sqrt{2} \cdot \left\| \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star \mathbf{x}_\star^\top \right\|_{\mathrm{F}} \\ &\leq \sup_{\mathbf{x} \in \mathcal{X}^2(G)} \left| \mathbf{x}^\top \big(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\big)\mathbf{x} \right| \cdot \sqrt{2} \cdot \ell(\widehat{\mathbf{x}}, \mathbf{x}_\star), \end{aligned} \tag{5.24}$$

where the second inequality is due to Von Neumann's trace inequality, the third by the ℓ_1/ℓ_2 norm inquality and the fact that the matrix in the trace

has rank equal to 2 and hence at most two nonzero singular values. The last inequality is follows by the defintion of the spectral norm and the fact that $S \in \mathcal{P}^2(G)$. In summary,

$$\mathbb{E}[\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)] \leq 2\sqrt{2} \cdot \frac{1}{\lambda_1 - \lambda_2} \cdot \mathbb{E}\Big[\sup_{\mathbf{x} \in \mathcal{X}^2} \Big|\mathbf{x}^\top \Big(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\Big)\mathbf{x}\Big|\Big]. \tag{5.25}$$

Subsequently, [142] ultize a very interesting result by Mendelson [109] to upper bound the constrained quadratic in the RHS of (5.25). This is the only part where the additional assumptions (see Ass. 1) are needed. We note that in their proof the constraint set is defined on sparse vectors, but the argument is oblivious to the exact strcuture of that set and extends to our case. Based on the above, we are able to show that

$$\mathbb{E}\left[\sup_{\mathbf{x} \in \mathcal{X}^2} \Big|\mathbf{x}^\top \Big(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\Big)\mathbf{x}\Big|\right] \leq c \cdot K^2 \cdot \frac{\lambda_1}{n} \cdot \max\Big\{\sqrt{n}A,\, A^2\Big\}, \tag{5.26}$$

where

$$A \triangleq \mathbb{E}_{\mathbf{Y} \sim N(\mathbf{0}, \mathbf{I}_p)}\Big[\sup_{\mathbf{x} \in \mathcal{X}^2(G)} \langle \mathbf{Y}, \mathbf{x} \rangle\Big]. \tag{5.27}$$

Here, c and K constants depending on the input distribution (Ass.1).

Inequality (5.26) (in conjuction with (5.25)), reduces the problem of upper bounding $\mathbb{E}[\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)]$ to upper bounding the supremum of a Gaussian process (5.27). To achieve that, we use a simple discretization of the relevant set $\mathcal{X}^2(G)$ with a δ-covering net. In particular, we show the following.

Lemma 5.3.6. *Let $\mathcal{N}_\delta \subset \mathcal{X}^2(G)$ be a δ-covering set of $\mathcal{X}^2(G)$ in the Euclidean metric, with the following property: $\forall \mathbf{x} \in \mathcal{X}^2(G)$, $\exists \mathbf{y} \in \mathcal{N}_\delta$ such that $\|\mathbf{x} -$*

$\mathbf{y}\|_2 \leq \delta$ *and* $\mathrm{supp}(\mathbf{x}-\mathbf{y}) \in \mathcal{P}^2(G)$. *Then,*

$$A \triangleq \mathbb{E}_{\mathbf{Y}\sim N(\mathbf{0},\mathbf{I}_p)}\left[\sup_{\mathbf{x}\in\mathcal{X}^2(G)}\langle \mathbf{Y},\mathbf{x}\rangle\right] \leq (1-\delta)^{-1}\cdot\sqrt{2\log|\mathcal{N}_\delta|}. \tag{5.28}$$

Proof. Let $\mathbf{y}_\star \in \mathcal{X}^2(G)$ be such that $\langle \mathbf{Y},\mathbf{y}_\star\rangle = \sup_{\mathbf{x}\in\mathcal{X}^2}\langle \mathbf{Y},\mathbf{x}\rangle$. The set $\mathcal{N}_\delta$, by construction contains a point $\mathbf{y}_\sharp$ such that $\|\mathbf{y}_\star-\mathbf{y}_\sharp\|_2 \leq \delta$ and $\mathrm{supp}(\mathbf{y}_\star-\mathbf{y}_\sharp) \in \mathcal{P}^2(G)$. It follows that

$$\begin{aligned}\sup_{\mathbf{x}\in\mathcal{X}^2}\langle \mathbf{Y},\mathbf{x}\rangle &= \langle \mathbf{Y},\mathbf{y}_\star-\mathbf{y}_\sharp\rangle + \langle \mathbf{Y},\mathbf{y}_\sharp\rangle \\ &\leq \|\mathbf{y}_\star-\mathbf{y}_\sharp\|_2\cdot\sup_{\mathbf{y}\in\mathcal{X}^2(G)}\langle \mathbf{Y},\mathbf{y}\rangle + \langle \mathbf{Y},\mathbf{y}_\sharp\rangle \\ &\leq \delta\cdot\sup_{\mathbf{y}\in\mathcal{X}^2(G)}\langle \mathbf{Y},\mathbf{y}\rangle + \max_{\mathbf{y}\in\mathcal{N}_\delta}\langle \mathbf{Y},\mathbf{y}\rangle. \end{aligned}\tag{5.29}$$

In turn,

$$\sup_{\mathbf{x}\in\mathcal{X}^2}\langle \mathbf{Y},\mathbf{x}\rangle \leq (1-\delta)^{-1}\max_{\mathbf{y}\in\mathcal{N}_\delta}\langle \mathbf{Y},\mathbf{y}\rangle. \tag{5.30}$$

The inner product $\langle \mathbf{Y},\mathbf{y}\rangle$ in the RHS of the last inequality is a standard Gaussian random variable for every $\mathbf{y}\in\mathcal{N}_\delta$, since $\mathbf{y}$'s have unit ℓ_2-norm and $\mathbf{Y}$ consists of jointly Gaussian and uncorrelated (hence independent) random variables. The expected value of the maximum of $|\mathcal{N}_\delta|$ Gaussian random variables with unit variance, can be upper bounded as follows:

$$\mathbb{E}_{\mathbf{Y}\sim N(\mathbf{0},\mathbf{I}_p)}\left[\max_{\mathbf{y}\in\mathcal{N}_\delta}\langle \mathbf{Y},\mathbf{y}\rangle\right] \leq \sqrt{2\log|\mathcal{N}_\delta|} \tag{5.31}$$

(see Lemma D.3.49). This implies the desired result. □

The final remaining ingredient to obtain the upper bound of Thm. 5.9, is to show that such a δ-covering of $\mathcal{X}^2(G)$ exists for the case where G is a (p,s,d)-layer graph, and show that the covering is not too large.

Lemma 5.3.7. *For any $\delta \in (0,1)$, there exists a δ-covering $\mathcal{N}_\delta$ of $\mathcal{X}^2(G)$, where G is a (p,s,d)-layer graph in the Euclidean metric, with the following properties:*

1. *$\forall \mathbf{x} \in \mathcal{X}^2(G)$, $\exists \mathbf{y} \in \mathcal{N}_\delta$ such that $\|\mathbf{x}-\mathbf{y}\|_2 \leq \delta$ and $\operatorname{supp}(\mathbf{x}-\mathbf{y}) \in \mathcal{P}^2(G)$, and*

2. *$|\mathcal{N}_\delta| \leq c(\log((p-2)/s) + s\log d)$, for some constant $c > 0$ that depends potentially only on δ.*

Proof. We construct the δ-covering with the desired properties by associating isometric copies of $\mathbb{S}_2^{2s+1}$ with each support set in $\mathcal{P}^2(G)$ (recall that supports in $\mathcal{P}^2(G)$ have cardinality equal to $2s+2$), and subsequently using a separate δ-covering for each copy. The union of the local δ-nets forms a set $\mathcal{N}_\delta$ with the desired properties.

To verify that, observe that for any point $\mathbf{x} \in \mathcal{X}^2(G)$, the support of $\mathbf{x}$ has cardinality at most $2s+2$ and it is associated with one of the local copies. That local copy by definition contains a point $\mathbf{y}$ that lies δ-close to $\mathbf{x}$ in Euclidean distance. Furthermore, $\mathbf{y}$ has the same support as $\mathbf{x}$ and in turn, so does their difference. Finally, it is known [140] that there exists a minimal

δ-covering for $\mathbb{S}_2^{2s+1}$ with cardinality at most $(1 + 2/\delta)^{2s+2}$. Then,

$$\begin{aligned}
\log|\mathcal{N}_\delta| &\leq \log|\mathcal{P}^2(G)| + 2(s+1)\log(1+2/\delta) \\
&\leq 2\log|\mathcal{P}(G)| + 2(s+1)\log(1+2/\delta) \\
&\leq 2(\log((p-2)/s) + s\log d) + 2(s+1)\log(1+2/\delta) \\
&\leq 2\log(1+2/\delta)\cdot(\log((p-2)/s) + s\log d),
\end{aligned}$$

which proves the second property. □

Substituting the cardinality of $\mathcal{N}_\delta$ in (5.28) as Lemma 5.3.7 yields the desired bound on $\mathbb{E}[\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star)]$, completing the proof of Theorem 5.9.

5.4 Algorithmic approaches

We propose two algorithms for approximating the solution of the constrained quadratic maximization in (5.4): The first is an adaptation of the *truncated power iteration* method of [155] for the problem of computing sparse eigenvectors. The second relies on approximately solving (5.4) on a low rank approximation of $\widehat{\mathbf{\Sigma}}$, similar to [113, 16]. Both algorithms rely on a projection operation from $\mathbb{R}^p$ onto the feasible set $\mathcal{X}(G)$, for a given graph $G = (V, E)$. Besides the projection step, the algorithms are oblivious to the specifics of the constraint set,[2] and can adapt to different constraints by modifying the projection operation.

[2]For Alg. 5.11, the observation holds under mild assumptions: $\mathcal{X}(G)$ must be such that $\|\mathbf{x}\|_2 = \Theta(1)$, while $\pm\mathbf{x} \in \mathcal{X}(G)$ should both achieve the same objective value.

5.4.1 Graph-Truncated Power Method

We consider a simple iterative procedure, similar to the truncated power method of [155] for the problem of computing sparse eigenvectors. Our algorithm produces sequence of vectors $\mathbf{x}_i \in \mathcal{X}(G)$, $i \geq 0$, that serve as intermediate estimates of the desired solution of (5.4).

The procedure is summarized in Algorithm 5.10. In the ith iteration, the current estimate $\mathbf{x}_i$ is multiplied by the empirical covariance $\widehat{\mathbf{\Sigma}}$, The product $\mathbf{w}_i \in \mathbb{R}^p$ is projected back to the feasible set $\mathcal{X}(G)$, yielding the next estimate $\mathbf{x}_{i+1}$. The core of Algorithm 5.10 lies in the projection operation,

$$\mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \triangleq \operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}(G)} \frac{1}{2}\|\mathbf{x} - \mathbf{w}\|_2^2, \tag{5.32}$$

which is analyzed separately in Section 5.4.3. The initial estimate $\mathbf{x}_0$ can be selected randomly or based on simple heuristics, *e.g.*, the projection on $\mathcal{X}(G)$ of the column of $\widehat{\mathbf{\Sigma}}$ corresponding to the largest diagonal entry. The algorithm terminates when some convergence criterion is satisfied.

The computational complexity (per iteration) of Algorithm 5.10 is dominated by the cost of matrix-vector multiplication and the projection step. The former is $O(s \cdot p)$, where s is cardinality of the largest support in $\mathcal{X}(G)$. The projection operation for the particular set $\mathcal{X}(G)$, boils down to solving the longest path problem on a weighted variant of the DAG G (see Section 5.4.3), which can be solved in time $O(|V| + |E|)$, *i.e.*, linear in the size of G.

Algorithm 5.10 Graph-Truncated Power Method

input $\widehat{\mathbf{\Sigma}}$ – $p \times p$ real PSD covariance matrix
G – DAG $G = (V, E)$ on p vertices corresponding to the dimensions
$\mathbf{x}_0$, – p-dimensional real vector; the initial point of the iterative procedure.

1: $i \leftarrow 0$
2: **repeat**
3: $\quad \mathbf{w}_i \leftarrow \widehat{\mathbf{\Sigma}}\mathbf{x}_i$
4: $\quad \mathbf{x}_{i+1} \leftarrow \mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$
5: $\quad i \leftarrow i + 1$
6: **until** Convergence/Stop Criterion

output $\mathbf{x}_i$

5.4.2 Low-Dimensional Sample and Project

The second algorithm outputs an estimate of the desired solution of (5.4) by (approximately) solving the constrained quadratic maximization *not* on the original matrix $\widehat{\mathbf{\Sigma}}$, but on a low rank approximation $\widehat{\mathbf{\Sigma}}_r$ of $\widehat{\mathbf{\Sigma}}$, instead:

$$\widehat{\mathbf{\Sigma}}_r = \sum_{i=1}^{r} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top = \sum_{i=1}^{r} \mathbf{v}_i \mathbf{v}_i^\top = \mathbf{V}\mathbf{V}^\top, \tag{5.33}$$

where λ_i is the ith largest eigenvalue of $\widehat{\mathbf{\Sigma}}$, $\mathbf{q}_i$ is the corresponding eigenvector, $\mathbf{v}_i \triangleq \sqrt{\lambda_i} \cdot \mathbf{q}_i$, and $\mathbf{V}$ is the $p \times r$ matrix whose ith column is equal to $\mathbf{v}_i$. The approximation rank r is an accuracy parameter; typically, $r \ll p$.

Our algorithm operates[3] on $\widehat{\mathbf{\Sigma}}_r$ and seeks

$$\mathbf{x}_r \triangleq \operatorname*{argmax}_{\mathbf{x} \in \mathcal{X}(G)} \mathbf{x}^\top \widehat{\mathbf{\Sigma}}_r \mathbf{x}. \tag{5.34}$$

[3] Under the spiked covariance model, this approach may be asymptotically unsuitable; as the ambient dimension increases, it with fail to recover the latent signal. Empirically, however, if the spectral decay of $\widehat{\mathbf{\Sigma}}$ is sharp, it yields very competitive results.

Algorithm 5.11 Low-Dimensional Sample and Project

input $\widehat{\mathbf{\Sigma}}$ – $p \times p$ real PSD covariance matrix
G – DAG $G = (V, E)$ on p vertices corresponding to the dimensions
r – Accuracy parameter in $[p]$
ϵ – Accuracy parameter in $(0, 1)$
output $\widehat{\mathbf{x}}_r$ – p-dimensional real vector supported on an $S - T$ path of G.

$[\mathbf{Q}, \mathbf{\Lambda}] \leftarrow \texttt{svd}(\widehat{\mathbf{\Sigma}}, r)$
$\mathbf{V} \leftarrow \mathbf{Q}\mathbf{\Lambda}^{1/2}$ $\{\widehat{\mathbf{\Sigma}}_r \triangleq \mathbf{V}\mathbf{V}^\top\}$
$\mathcal{C} \leftarrow \{\}$ {Set of candidate solutions}
for $i = 1 : O(\epsilon^{-r} \cdot \log p)$ **do**
  $\mathbf{c}_i \leftarrow$ uniformly sampled from $\mathbb{S}^{r-1}$
  $\mathbf{w}_i \leftarrow \mathbf{V}\mathbf{c}_i$
  $\mathbf{x}_i \leftarrow \text{Proj}_{\mathcal{X}(G)}(\mathbf{w}_i)$
  $\mathcal{C} = \mathcal{C} \cup \{\mathbf{x}_i\}$
end for
$\widehat{\mathbf{x}}_r \leftarrow \text{argmax}_{\mathbf{x} \in \mathcal{C}} \|\mathbf{V}^\top \mathbf{x}\|_2^2$

The motivation is that an (approximate) solution for the low-rank problem in (5.34) can be efficiently computed. Intuitively, if $\widehat{\mathbf{\Sigma}}_r$ is a sufficiently good approximation of the original matrix $\widehat{\mathbf{\Sigma}}$, then $\mathbf{x}_r$ would perform similarly to the solution $\mathbf{x}_\star$ of the original problem (5.4). Our algorithm samples points from the low-dimensional principal subspace of $\widehat{\mathbf{\Sigma}}$, and projects them on the feasible set $\mathcal{X}(G)$, producing a set of candidate estimates for $\mathbf{x}_r$. It outputs the candidate that maximizes the objective in (5.34). The exact steps are formally presented in Algorithm 5.11. The following paragraphs delve into the details of Algorithm 5.11.

5.4.2.1 The Low Rank Problem

The rank-r maximization in (5.34) can be written as

$$\max_{\mathbf{x}\in\mathcal{X}(G)} \mathbf{x}^\top \widehat{\mathbf{\Sigma}}_r \mathbf{x} = \max_{\mathbf{x}\in\mathcal{X}(G)} \|\mathbf{V}^\top \mathbf{x}\|_2^2, \tag{5.35}$$

and in turn (see [16] for details), as a double maximization over the variables $\mathbf{c} \in \mathbb{S}^{r-1}$ and $\mathbf{x} \in \mathbb{R}^p$:

$$\max_{\mathbf{x}\in\mathcal{X}(G)} \|\mathbf{V}^\top \mathbf{x}\|_2^2 = \max_{\mathbf{c}\in\mathbb{S}^{r-1}} \max_{\mathbf{x}\in\mathcal{X}(G)} \Big((\mathbf{V}\mathbf{c})^\top \mathbf{x}\Big)^2. \tag{5.36}$$

The rank-1 case Let $\mathbf{w} \triangleq \mathbf{V}\mathbf{c}$; $\mathbf{w}$ is only a vector in $\mathbb{R}^p$. For given $\mathbf{c}$ and $\mathbf{w}$, the $\mathbf{x}$ that maximizes the objective in (5.36) (as a function of $\mathbf{c}$) is

$$\mathbf{x}(\mathbf{c}) \in \operatorname*{argmax}_{\mathbf{x}\in\mathcal{X}(G)} \Big(\mathbf{w}^\top \mathbf{x}\Big)^2. \tag{5.37}$$

The maximization in (5.37) is nothing but a rank-1 instance of the maximization in (5.35). Observe that if $\mathbf{x} \in \mathcal{X}(G)$, then $-\mathbf{x} \in \mathcal{X}(G)$, and the two vectors attain the same objective value. Hence, (5.37) can be simplified:

$$\mathbf{x}(\mathbf{c}) \in \operatorname*{argmax}_{\mathbf{x}\in\mathcal{X}(G)} \mathbf{w}^\top \mathbf{x}. \tag{5.38}$$

Further, since $\|\mathbf{x}\|_2 = 1$, $\forall \mathbf{x} \in \mathcal{X}(G)$, the maximization in (5.38) is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}-\mathbf{x}\|_2^2$. In other words, $\mathbf{x}(\mathbf{c})$ is just the projection of $\mathbf{w} \in \mathbb{R}^p$ onto $\mathcal{X}(G)$:

$$\mathbf{x}(\mathbf{c}) \;\in\; \mathrm{Proj}_{\mathcal{X}(G)}(\mathbf{w}). \tag{5.39}$$

The projection operator is described in Section 5.4.3.

Multiple rank-1 instances Let $(\mathbf{c}_r, \mathbf{x}_r)$ denote a pair that attains the maximum value in (5.36). If $\mathbf{c}_r$ was known, then $\mathbf{x}_r$ would coincide with the projection $\mathbf{x}(\mathbf{c}_r)$ of $\mathbf{w} = \mathbf{V}\mathbf{c}_r$ on the feasible set, according to (5.39).

Of course, the optimal value $\mathbf{c}_r$ of the auxiliary variable is not known. Recall, however, that $\mathbf{c}_r$ lies on the low dimensional manifold $\mathbb{S}^{r-1}$. Consider an ϵ-net $\mathcal{N}_\epsilon$ covering the r-dimensional unit sphere $\mathbb{S}^{r-1}$; Algorithm 5.11 constructs such a net by random sampling. By definition, $\mathcal{N}_\epsilon$ contains at least one point, call it $\widehat{\mathbf{c}}_r$, in the vicinity of $\mathbf{c}_r$. It can be shown that the corresponding solution $\mathbf{x}(\widehat{\mathbf{c}}_r)$ in (5.37) will perform approximately as well as the optimal solution $\mathbf{x}_r$, in terms of the quadratic objective in (5.36), for a large, but tractable, number of points in the ϵ-net of $\mathbb{S}^{r-1}$.

5.4.3 Projection on the Feasible Set

Algorithms 5.10, and 5.11 rely on a projection operation from $\mathbb{R}^p$ onto the feasible set $\mathcal{X}(G)$ dfined in (5.5). We show that the projection effectively reduces to solving the *longest path problem* on (a weighted variant of) the side-information graph G.

The projection operation, defined in Eq. (5.32), can be equivalently[4] written as

$$\text{Proj}_{\mathcal{X}(G)}(\mathbf{w}) \triangleq \underset{\mathbf{x}\in\mathcal{X}(G)}{\text{argmax}}\, \mathbf{w}^\top \mathbf{x}.$$

[4] It follows from expanding the quadratic $\frac{1}{2}\|\mathbf{x} - \mathbf{w}\|_2^2$ and the fact that $\|\mathbf{x}\|_2 = 1$, $\forall \mathbf{x} \in \mathcal{X}(G)$.

For any $\mathbf{x} \in \mathcal{X}(G)$, $\text{supp}(\mathbf{x}) \in \mathcal{P}(G)$. For a given set π, by the Cauchy-Schwarz inequality,

$$\mathbf{w}^\top \mathbf{x} = \sum_{i \in \pi} w_i x_i \;\leq\; \sum_{i \in \pi} w_i^2 = \widehat{\mathbf{w}}^\top \mathbf{1}_\pi, \tag{5.40}$$

where $\widehat{\mathbf{w}} \in \mathbb{R}^p$ is the vector obtained by squaring the entries of $\mathbf{w}$, *i.e.*, $\widehat{w}_i = w_i^2$, $\forall i \in [n]$, and $\mathbf{1}_\pi \in \{0,1\}^p$ denotes the characteristic of π. Letting $\mathbf{x}[\pi]$ denote the subvector of $\mathbf{x}$ supported on π, equality in (5.40) can be achieved by $\mathbf{x}$ such that $\mathbf{x}[\pi] = \mathbf{w}[\pi]/\|\mathbf{w}[\pi]\|_2$, and $\mathbf{x}[\pi^c] = \mathbf{0}$.

Hence, the problem in (5.40) reduces to determining

$$\pi(\mathbf{w}) \in \operatorname*{argmax}_{\pi \in \mathcal{P}(G)} \widehat{\mathbf{w}}^\top \mathbf{1}_\pi. \tag{5.41}$$

Consider a weighted graph $G_{\mathbf{w}}$, obtained from $G = (V, E)$ by assigning weight $\widehat{w}_v = w_v^2$ on vertex $v \in V$. The objective function in (5.41) equals the *weight of the path* π in $G_{\mathbf{w}}$, *i.e.*, the sum of weights of the vertices along π. Determining the optimal support $\pi(\mathbf{w})$ for a given $\mathbf{w}$, is equivalent to solving the *longest (weighted) path problem*[5] on $G_{\mathbf{w}}$.

The longest (weighted) path problem is NP-hard on arbitrary graphs. In the case of DAGs, however, it can be solved using standard algorithms relying on topological sorting in time $O(|V| + |E|)$ [45], *i.e.*, linear in the size of the graph. Hence, the projection $\mathbf{x}$ can be determined in time $O(p + |E|)$.

[5] The longest path problem is commonly defined on graphs with weighted edges instead of vertices. The latter is trivially transformed to the former: set $w(u,v) \leftarrow w(v)$, $\forall (u,v) \in E$, where $w(u,v)$ denotes the weight of edge (u,v), and $w(v)$ that of vertex v. Auxiliary edges can be introduced for source vertices.

5.5 Experiments

5.5.1 Synthetic Data

We evaluate Alg. 5.10 and 5.11 on synthetic data, generated according to the model of Sec. 5.3.2. We consider two metrics: *i*) the loss function $\ell(\widehat{\mathbf{x}}, \mathbf{x}_\star) = \|\widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top - \mathbf{x}_\star\mathbf{x}_\star\|_{\mathrm{F}}$, and *ii*) the Jaccard distance $J(\widehat{\mathbf{x}}, \mathbf{x}_\star)$ between the supports of the true signal $\mathbf{x}_\star$ and the estimate $\widehat{\mathbf{x}}$.

For dimension p, we generate a (p, s, d)-layer graph G, with $s = \log p$ layers and a degree upper bound $d = p/s$, *i.e.*, each vertex is connected to all vertices of the following layer. We augment the graph with auxiliary source and terminal vertices S and T with edges to the original vertices as in Fig. 5.1.

Per random realization, we first construct a signal $\mathbf{x}_\star \in \mathcal{X}(G)$ as follows: we randomly select an S-T path π in G, and assign random zero-mean Gaussian values to the entries of $\mathbf{x}_\star$ indexed by π. The signal is scaled to unit length. Given $\mathbf{x}_\star$, we generate n independent samples according to (5.8).

Fig. 5.2 depicts the aforementioned distance metrics as a function of the number n of observations. Results are the average of 100 independent realizations. We repeat the procedure for multiple values of the ambient dimension p.

Comparison with Sparse PCA We compare the performance of Alg. 5.10 and Alg. 5.11 with their sparse PCA counterparts: the Truncated Power Method of [155] and the Spannogram Alg. of [113], respectively.

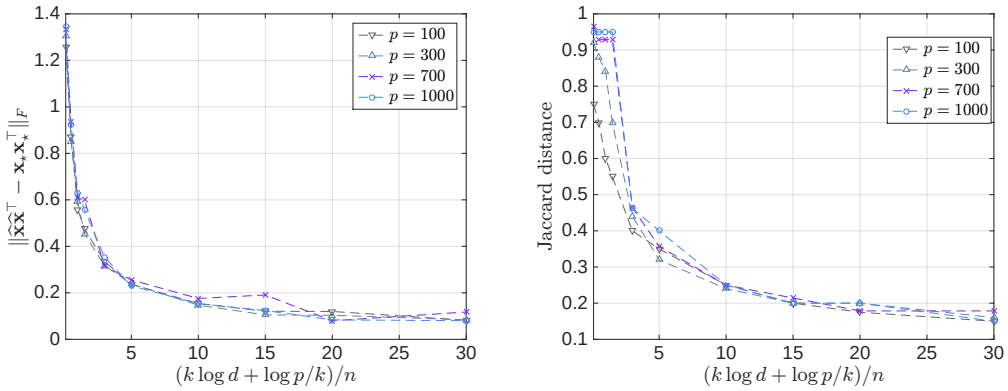


Figure 5.2: Metrics on the estimate $\widehat{\mathbf{x}}$ produced by Alg. 5.10 (Alg. 5.11 is similar) as a function of the sample number (average of 100 realizations). Samples are generated according to the spiked covariance model with signal $\mathbf{x}_\star \in \mathcal{X}(G)$ for a (p, k, d)-layer graph G. Here, $k = \log p$ and $d = p/s$. We repeat for multiple values of p.

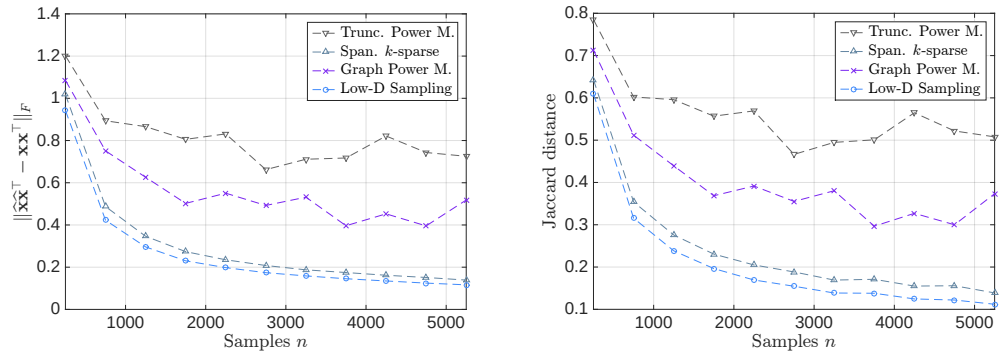


Figure 5.3: Estimation error between true signal $\mathbf{x}_\star$ and estimate $\widehat{\mathbf{x}}$ from n samples. (average of 100 realizations). Samples generated *i.i.d.* $\sim N(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ has eigenvalues $\lambda_i = i^{-1/4}$ and principal eigenvector $\mathbf{x}_\star \in \mathcal{X}(G)$, for a (p, s, d)-layer graph G. ($p = 10^3$, $s = 50$, $d = 10$).

Fig. 5.3 depicts the metrics of interest as a function of the number of samples, for all four algorithms. Here, samples are drawn *i.i.d* from $N(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ has principal eigenvector equal to $\mathbf{x}_\star$, and power law spectral de-

cay: $\lambda_i = i^{-1/4}$. Results are an average of 100 realizations.

The side information on the structure of $\mathbf{x}_\star$ assists the recovery: both algorithms achieve improved performance compared to their sparse PCA counterparts. Here, the power method based algorithms exhibit inferior performance, which may be attributed to poor initialization. We note, though, that at least for the size of these experiments, the power method algorithms are significantly faster.

5.5.2 Finance Data

This dataset contains daily closing prices for 425 stocks of the S&P 500 Index, over a period of 1259 days (5-years): 02.01.2010 – 01.28.2015, collected from Yahoo! Finance[6]. Stocks are classified, according to the *Global Industry Classification Standard*[7] (GICS), into 10 business *sectors e.g.*, Energy, Health Care, Information Technology, etc (see Fig. 5.4 for the complete list).

We seek a set of stocks comprising a single representative from each GICS sector, which captures most of the variance in the dataset. Equivalently, we want to compute a structured principal component constrained to have exactly 10 nonzero entries; one for each GICS sector.

Consider a layer graph $G = (V, E)$ (similar to the one depicted in Fig. 5.1) on $p = 425$ vertices corresponding to the 425 stocks, partitioned into $k = 10$ groups (layers) $\mathcal{L}_1, \ldots, \mathcal{L}_{10} \subseteq V$, corresponding to the GICS sec-

[6] http://finance.yahoo.com
[7] http://www.msci.com/products/indexes/sector/gics/gics_structure.html

tors. Each vertex in layer $\mathcal{L}_i$ has outgoing edges towards all (and only the) vertices in layer $\mathcal{L}_{i+1}$. Note that (unlike Fig. 5.1) layers do *not* have equal sizes, and the vertex out-degree varies across layers. Finally, we introduce auxiliary vertices S and T connected with the original graph as in Fig. 5.1.

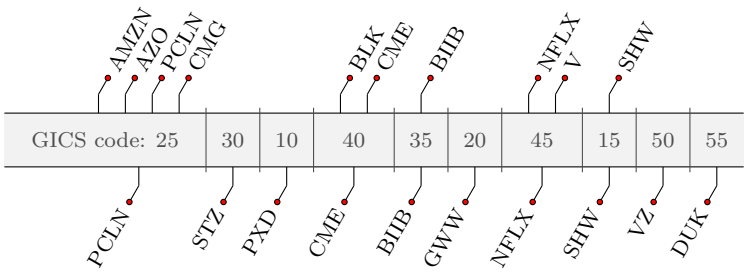
Observe that any set of sector-representatives corresponds to an S-T path in G, and vice versa. Hence, the desired set of stocks can be obtained by finding a *structured* principal component constrained to be supported along an S-T path in G. Note that the order of layers in G is irrelevant.

Fig. 5.4 depicts the subset of stocks selected by the proposed structure PCA algorithms (Alg. 5.10, 5.11). A single representative is selected from each sector. For comparison, we also run two corresponding algorithms for sparse PCA, with sparsity parameter $k = 10$, equal to the number of sectors. As expected, the latter yield components achieving higher values of explained variance, but the selected stocks originate from only 5 out of the 10 sectors.

5.5.3 Neuroscience Data

We use a single-session/single-participant resting state functional magnetic resonance imaging (resting state fMRI) dataset. The participant was not instructed to perform any explicit cognitive task throughout the scan [137]. Data was provided by the Human Connectome Project, WU-Minn Consortium.[8]

[8](Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience



GICS Sector Codes			
10	Energy	35	Health Care
15	Materials	40	Financials
20	Industrials	45	Information Tech.
25	Consumer Discretionary	50	Telecom. Services
30	Consumer Staples	55	Utilities

Figure 5.4: Output of Sparse PCA and Path PCA algorithms on the Finance Dataset. The dataset comprises daily closing prices for 425 stocks of the S&P 500 Index, over a period of 1259 days (crawled from Yahoo! Finance). The figure depicts the sets of 10 stocks extracted by Sparse PCA and our Path PCA approach. Sparse PCA (for $k = 10$), selects 10 stocks from 5 GICS sectors (above). On the contrary, our structured PCA algorithms yield a set of 10 stocks containing a representative from each sector (below) as desired.

Mean timeseries of $n = 1200$ points for $p = 111$ regions-of-interest (ROIs) are extracted based on the Harvard-Oxford Atlas [54]. The timescale of analysis is restricted to 0.01–0.1Hz. Based on recent results on resting state fMRI neural networks, we set the posterior cingulate cortex as a source node S, and the prefrontal cortex as a target node T [64]. Starting from S, we construct a layered graph with $s = 4$, based on the physical (Euclidean) distances between the center of mass of the ROIs: *i.e.,* given layer $\mathcal{L}_i$, we construct $\mathcal{L}_{i+1}$ from non-selected nodes that are close in the Euclidean sense.

Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

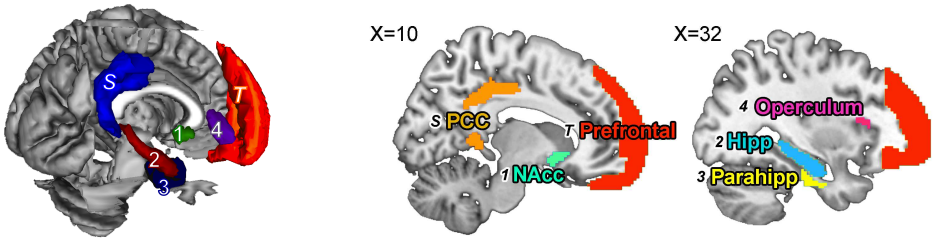


Figure 5.5: We highlight the nodes extracted for the neuroscience example. Source node set to the posterior cingulate cortex (S: PCC), and target to the prefrontal cortex (T: Prefrontal). The directed path proceeded from the nucleus accumbens (1: NAcc), hippocampus (2: Hipp), parahippocampal gyrus (3: Parahipp), and to the frontal operculum (4: Operculum). Here, X coordinates (in mm) denote how far from the midline the cuts are.

Here, $|\mathcal{L}_1| = 34$ and $|\mathcal{L}_i| = 25$ for $i = 2, 3, 4$. Each layer is fully connected with its previous one. No further assumptions are derived from neurobiology.

The extracted component suggests a directed pathway from the posterior cingulate cortex (S) to the prefrontal cortex (T), through the hippocampus (1), nucleus accumbens (2), parahippocampal gyrus (3), and frontal operculum (4) (Fig. 5.5). Hippocampus and the parahippocampal gyrus are critical in memory encoding, and have been found to be structurally connected to the posterior cingulate cortex and the prefrontal cortex [64]. The nucleus accumbens receives input from the hippocampus, and plays an important role in memory consolidation [150]. It is noteworthy that our approach has pinpointed the core neural components of the memory network, given minimal information.

5.6 Conclusions

We introduced a new problem: sparse PCA where the set of feasible support sets is determined by a graph on the variables. We focused on the special case where feasible sparsity patterns coincide with paths on the underlying graph. We provided an upper bound on the statistical complexity of the constrained quadratic maximization estimator (5.4), under a simple graph model, complemented with a lower bound on the minimax error. Finally, we proposed two algorithms to extract a component accommodating the graph constraints and applied them on real data from finance and neuroscience.

A potential future direction is to expand the set of graph-induced sparsity patterns (beyond paths) that can lead to interpretable solutions and are computationally tractable. We hope this work triggers future efforts to introduce and exploit such underlying structure in diverse research fields.

Chapter 6

A Simple Algorithm for Sparse CCA (SVD)

Given two sets of variables derived from a common set of samples, sparse Canonical Correlation Analysis (CCA) seeks linear combinations of a small number of variables in each set, such that the induced *canonical* variables are maximally correlated. Sparse CCA is NP-hard.

We propose a novel combinatorial algorithm for sparse CCA. Our algorithm operates on a low rank approximation of the input data and its computational complexity scales linearly with the ambient dimension, which makes it suitable for large scale settings. It is simple to implement, and embarrassingly parallelizable. In contrast to the majority of existing approaches, our algorithm administers precise control on the sparsity of the extracted canonical vectors, and comes with theoretical data-dependent global approximation guarantees, that hinge on the spectrum of the input data. Finally, it can

Chapter 6 is based on material from Reference [14]: Megasthenis Asteris, Anastasios Kyrillidis, Oluwasanmi Koyejo, and Russell Poldrack, *"A Simple and Provable Algorithm for Sparse Diagonal CCA"*, Proceedings of The 33rd International Conference on Machine Learning, Volume 48, JMLR Workshop and Conference Proceedings, 2016. The author of this dissertation is the lead author of [14], and contributed to the conception of the research problem, the theoretical and analytical developments, the experimental design and implementation, and the writing of the manuscript and its revisions. The interpretation of the experimental results on real data was provided by Oluwasanmi Koyejo and Russell Poldrack.

be straightforwardly adapted to other constrained variants of CCA, enforcing additional structure beyond sparsity.

We empirically evaluate the proposed scheme and apply it on a real neuroimaging dataset to investigate associations between brain activity and behavior measurements.

6.1 Introduction

One of the key objectives in cognitive neuroscience is to localize cognitive processes in the brain, and understand their role in human behavior, as measured by psychological scores and physiological measurements [121]. This mapping may be investigated by using functional neuroimaging techniques to measure brain activation during carefully designed experimental tasks [117]. Following the experimental manipulation, a joint analysis of brain activation and behavioral measurements across subjects can reveal associations that exist between the two [27].

Similarly, in genetics and molecular biology, several studies involve the joint analysis of multiple assays performed on a single group of patients [118, 111, 129]. If DNA variants and gene expression measurements are simultaneously available for a set of tissue samples, a natural objective is to identify correlations between the expression levels of gene subsets and variation in the related genes.

Canonical Correlation Analysis (CCA) [70] is a classic method for dis-

covering such linear relationships across two sets of variables and has been extensively used to investigate association between multiple views of the same set of observations; *e.g.*, see [52, 99, 128] in neuroscience. Given two datasets $\mathbf{X}$ and $\mathbf{Y}$ of dimensions $k \times m$ and $k \times n$, respectively, on k common samples, CCA seeks linear combinations of the original variables of each type that are maximally correlated. More formally, the objective is to compute a pair of *canonical vectors (or weights)* $\mathbf{x}$ and $\mathbf{y}$ such that the *canonical variables* $\mathbf{Xx}$ and $\mathbf{Yy}$ achieve the maximum possible correlation[1]

$$\mathbf{x}^\top \mathbf{\Sigma}_{\text{XY}} \mathbf{y} / \left(\mathbf{x}^\top \mathbf{\Sigma}_{\text{XX}} \mathbf{x}\right)^{1/2} \left(\mathbf{y}^\top \mathbf{\Sigma}_{\text{YY}} \mathbf{y}\right)^{1/2}. \tag{6.1}$$

The optimal canonical pair can be computed via a generalized eigenvalue decomposition involving the empirical estimates of the (cross-) covariance matrices in (6.1).

Imaging and behavioral measurements in cognitive neuroscience, similar to genomic data in bioinformatics, typically involve hundreds of thousands of variables with only a limited number of samples. In that case, the CCA objective in (6.1) is ill-posed; it is always possible to design canonical variables for which the factors in the denominator vanish, irrespective of the data. Model regularization via constraints such as sparsity, not only improves the interpretability of the extracted canonical vectors, but is critical for enabling the recovery of meaningful results.

[1] We assume that the variables in $\mathbf{X}$ and $\mathbf{Y}$ are standardized, *i.e.*, each column has zero mean and has been scaled to have unit standard deviation.

Sparse CCA seeks to maximize the correlation between subsets of variables of each type while performing variable selection. Here, we consider the following formulation of Sparse CCA, similar to [148][2]:

$$\text{(Sparse CCA)} \qquad \max_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} \mathbf{x}^\top \mathbf{A}\mathbf{y}, \tag{6.2}$$

where

$$\begin{aligned} \mathcal{X} &= \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x}\|_2 = 1, \|\mathbf{x}\|_0 \leq s_x\}, \\ \mathcal{Y} &= \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_2 = 1, \|\mathbf{y}\|_0 \leq s_y\}, \end{aligned} \tag{6.3}$$

for given parameters s_x and s_y. The $m \times n$ argument matrix $\mathbf{A} = \mathbf{X}^\top\mathbf{Y}$ is the input of the optimization problem. Note that besides the introduced sparsity requirement, the optimization problem in (6.2) is obtained from (6.1) treating the covariance matrices for $\mathbf{\Sigma}_{\text{XX}}$ and $\mathbf{\Sigma}_{\text{YY}}$ as idenity matrices, which is common in high dimensions [57, 135].

Disregarding the ℓ_0 cardinality constraint, although the objective is non-convex, the optimal solution of (6.2) can be easily computed as it coincides with the leading singular vectors of input matrix $\mathbf{A}$. The constraint on the number of nonzero entries of $\mathbf{x}$ and $\mathbf{y}$, however, renders the problem NP-hard as it can be shown by a reduction to the closely related sparse PCA problem; see Appendix E.1. Several heuristics have been developed to obtain an approximate solution.

[2] [148] consider a relaxation of (6.2) where the ℓ_0 cardinality constraint on $\mathbf{x}$ and $\mathbf{y}$ is replaced by a threshold on the sparsity inducing ℓ_1-norm. The important aspect here is the objective function, which is derived from (6.1) treating the covariance matrices of each individual dataset as an identity matrix.

Finally, we note that sparsity alone may be insufficient to obtain interpretable results; genes participate in groups in biological pathways, and brain activity tends to be localized forming connected components over an underlying network. If higher order structural information is available on a physical system, it is meaningful incorporate that in the optimization (6.2). We can then consider *Structured* variants of CCA (6.2), by appropriately modifying the feasible region in (6.3) to reflect the desired structure.

Our contributions We present a novel and efficient combinatorial algorithm for sparse CCA. The main idea is to reduce the exponentially large search space of candidate supports of the canonical vectors, by exploring a low-dimensional principal subspace of the input data. Our algorithm runs in polynomial time –in fact linear– in the dimension of the input. It administers precise control over the sparsity of the extracted canonical vectors and can extract components for multiple sparsity values on a single run. It is simple and embarrassingly parallelizable; we empirically demonstrate that it achieves an almost-linear speedup factor in the number of available processing units.

The algorithm is accompanied with theoretical data-dependent global approximation guarantees with respect to the CCA objective (6.2); this is the first approach with this kind of global guarantees. The latter depend on the rank r of the low-dimensional space that is explored by the algorithm and the spectral decay of the input matrix $\mathbf{A}$. The main weakness is an exponential dependence of the computational complexity on the accuracy parameter r. In

practice, however, disregarding the theoretical approximation guarantees, our algorithm can be executed for any allowable time window.

Beyond sparsity, our algorithm can be straightforwardly extended to other constrained variants of CCA; we only require the existence of suitable subroutines for "projecting" a vector onto the non-convex feasible sets $\mathcal{X}$ and $\mathcal{Y}$. Similar to the vanilla sparsity case, such exact or approximate projections exist for several constraints of interest such as non-negativity, smooth sparsity, group sparsity, or sparsity with patterns induced by underlying graph models. Our theoretical approximation guarantees are oblivious to the type of constraints and immediately extend to these cases.

Finally, we note that our approach is similar to that of [16] for sparse PCA. The latter has a similar formulation with (6.2) but is restricted to a positive semidefinite argument $\mathbf{A}$. Our main technical contribution is extending those algorithmic ideas and developing theoretical approximation guarantees for the bilinear maximization (6.2) where the input matrix can be arbitrary.

6.2 Related Work

Sparse CCA is closely related to sparse PCA; the latter can be formulated as in (6.2) but on a positive semidefinite argument $\mathbf{A}$ and a single multivariate $\mathbf{x}$. There is a large volume of work on sparse PCA –see [160, 8] and references therein– but these methods cannot be trivially generalized to the CCA problem. One exception is the work of [51] where the authors discuss extensions to the "non-square case". Their approach relies on a semidefinite

relaxation.

References to sparsity in CCA date back to [134] and [133] who identified the importance of sparsity regularization to obtain meaningful results [3]. However, no specific algorithm was proposed. Several subsequent works considered a penalized version of the CCA problem in (6.1), typically under a Langrangian formulation involving a convex relaxation of the ℓ_0 cardinality constraint [136, 67, 68]. [42] characterize the solutions of the unconstrained problem and formulate convex ℓ_1 minimization problems to seek sparse solutions in that set. [147] proposed an efficient greedy procedure that gradually expands the supports of the canonical vectors. Unlike other methods, this greedy approach allows precise control of the sparsity of the extracted components.

[148, 115] formulate sparse CCA as the optimization (6.2) and in particular considered an ℓ_1 relaxation of the ℓ_0 cardinality constraint. They suggest an alternating minization approach exploiting the bi-convex nature of the relaxed problem, solving a lasso regression in each step. The same approach is followed in [144] combining ℓ_2 and ℓ_1 regularizers similarly to the elastic net approach for sparse PCA [161]. A common weakness in these apporaches is the lack of precise control over sparsity: the mapping between the reguralization parameters and the number of nonzero entries in the extracted components is

[3] [134]: "[...], the fewer variables there are in a canonical analysis which yields a correlation of a given magnitude, the greater is the likelihood that that correlation is due to real, population-wide sources of co-variation, rather than to sample-specific sources.".

highly nonlinear. Further, such methods usually lack provable approximation guarantees. Beyond sparsity, [148, 149] discuss alternative penalizations such as fused lasso to impose additional structure, while [39] introduce a group-lasso to promote sparsity with structure.

Finally, although a review of CCA applications is beyond the scope of this manuscript, we simply note that CCA is considered a promising approach for scientific research as evidenced by several recent works in the literature, *e.g*, in neuroscience [125, 52, 99, 101, 128].

6.3 An Algorithm for Sparse CCA

We present SpanCCA, a novel and simple algorithm for sparse CCA (6.2) with global approximation guarantees. We begin this section with a brief discussion of the problem and the key ideas behind our approach. Next, we provide an overview of SpanCCA and the accompanying approximation guarantees and conclude with a short analysis.

6.3.1 Intuition

The hardness of the sparse CCA (6.2) lies in the detection of the optimal supports for the canonical vectors. In the *unconstrained* problem, where only a unit ℓ_2-norm constraint is imposed on $\mathbf{x}$ and $\mathbf{y}$, the optimal CCA pair coincides with the top singular vectors of the input argument $\mathbf{A}$. In the sparse variant, we restrict the feasible region to unit ℓ_2-norm vectors $\mathbf{x}$ and $\mathbf{y}$ with at most s_x and s_y nonzero entries, respectively. If the optimal supports for $\mathbf{x}$ and $\mathbf{y}$

were known, computing the optimal solution would be straightforward: the nonzero subvectors of $\mathbf{x}$ and $\mathbf{y}$ would coincide with the leading singular vectors of the $s_x \times s_y$ submatrix of $\mathbf{A}$, indexed by the two support sets. Hence, the bottleneck lies in determining the optimal supports for $\mathbf{x}$ and $\mathbf{y}$.

Exhaustive search A straightforward, brute-force approach to sparse CCA is to exhaustively consider all possible supports for $\mathbf{x}$ and $\mathbf{y}$: for each candidate pair solve the unconstrained CCA problem on the restricted input, and determine the supports for which the objective (6.2) is maximized. Albeit optimal, this procedure is intractable as the number of candidate supports $\binom{m}{s_x}\binom{n}{s_y}$ is overwhelming even for small values of s_x and s_y.

Thresholding On the other hand, a feasible pair of sparse canonical vectors $\mathbf{x}$ and $\mathbf{y}$ can be extracted by hard-thresholding the unconstrained CCA solution, *i.e.*, computing the leading singular vectors $\mathbf{u}$ and $\mathbf{v}$ of $\mathbf{A}$, suppressing to zero all but the s_x and s_y largest in magnitude entries, respectively, and rescaling to obtain a unit ℓ_2-norm solution. Essentially, this heuristic resorts to unconstrained CCA for a *guided* selection of the sparse support.

Proposed method Our sparse CCA algorithm covers the ground between these two approaches. Instead of relying on the unconstrained CCA solution for the choice of the sparse support pair, it explores a principal subspace of the input matrix $\mathbf{A}$, spanned by its leading $r \geq 1$ singular vector pairs.

For $r = 1$, its output coincides with that of the thresholding approach, while for $r = \min\{m, n\}$ it approximates that of exhaustive search.

Effectively, we solve the sparse CCA problem (6.2) on a rank-r approximation of the original input $\mathbf{A}$. The key observation is that the low inner dimension of the argument matrix can be exploited to substantially reduce the search space: our algorithm identifies an (approximately) optimal pair of supports for the low rank sparse CCA problem, without considering the entire collection of possible supports of cardinalities s_x and s_y.

6.3.2 Overview and Guarantees

SpanCCA is outlined in Algorithm 6.12. The first step is to compute a rank-r approximation $\mathbf{B}$ of the input $\mathbf{A}$, where r is an accuracy input parameter. The low rank surrogate matrix can be easily obtained via the truncated singular value decomposition (SVD) of $\mathbf{A}$, or in the case of very high dimensionality, where standard Lanczos-based methods may be computationally prohibitive, using faster randomized approaches; see [66]. Here, for simplicity, we consider the exact case.

From that point on, the algorithm operates exclusively on $\mathbf{B}$ effectively solving a low-rank sparse CCA problem:

$$\max_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} \quad \mathbf{x}^\top \mathbf{B}\mathbf{y}, \tag{6.4}$$

where $\mathcal{X} \subseteq \mathbb{S}_2^{m-1}$ and $\mathcal{Y} \subseteq \mathbb{S}_2^{n-1}$ are defined in (6.3). As we discuss in the next section, we can consider other constrained variants of CCA on potentially

arbitrary, non-convex sets. We do require, however, that there exist procedures to (at least approximately) solve the maximizations

$$\mathsf{P}_{\mathcal{X}}(\mathbf{a}) \triangleq \operatorname*{argmax}_{\mathbf{x}\in\mathcal{X}} \ \mathbf{a}^\top \mathbf{x}, \tag{6.5}$$

$$\mathsf{P}_{\mathcal{Y}}(\mathbf{b}) \triangleq \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \ \mathbf{b}^\top \mathbf{y}, \tag{6.6}$$

for any given vectors $\mathbf{a} \in \mathbb{R}^{m\times 1}$ and $\mathbf{b} \in \mathbb{R}^{n\times 1}$. Fortunately, this is the case for the sets of sparse unit ℓ_2-norm vectors. Algorithm 6.13 outlines an efficient $O(m)$ procedure that given $\mathbf{a} \in \mathbb{R}^{m\times 1}$ computes an exact solution to (6.5) with at most $s \leq m$ nonzero entries: first it determines the s largest (in magnitude) entries of $\mathbf{a}$ (breaking ties arbitrarily), it zeroes out the remaining entries and re-scales the output to meet the ℓ_2-norm requirement.

The main body of Algorithm 6.12 consists of a single iteration. In the ith round, it independently samples a point, or equivalently direction, $\mathbf{c}_i$ from the r-dimensional unit ℓ_2 sphere and uses it to appropriately sample a point $\mathbf{a}_i$ in the range of $\mathbf{B}$. The latter is then used to compute a feasible solution pair $\mathbf{x}_i$, $\mathbf{y}_i$ via a two-step procedure: first the algorithm computes $\mathbf{x}_i$ by "projecting" $\mathbf{a}_i$ onto $\mathcal{X}$ invoking Alg. 6.13 as a subroutine to solve maximization (6.5), and then computes $\mathbf{y}_i$ by projecting $\mathbf{b}_i = \mathbf{B}^\top \mathbf{x}_i$ onto $\mathcal{Y}$ in a similar fashion. The algorithm repeats this procedure for T rounds and outputs the pair that achieves the maximum objective value in (6.4). We emphasize that consecutive rounds are completely independent and can be executed in parallel.

For a sufficiently large number T of rounds (or samples) the procedure

Algorithm 6.12 SpanCCA

input $\mathbf{A}$ – $m \times n$ real matrix.
r – Parameter controlling the rank of the approximation to be used.
T – Parameter controlling the number samples/iterations.

output $(\mathbf{x}_\sharp, \mathbf{y}_\sharp)$ – Pair of feasible vectors $\mathbf{x}_\sharp \in \mathcal{X}$, $\mathbf{y}_\sharp \in \mathcal{Y}$. See Thm. 6.11.

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \texttt{svd}(\mathbf{A}, r)$ $\quad \{\ \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\ \}$
for $i = 1, \ldots, T$ **do**
  $\mathbf{c}_i \leftarrow \texttt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r\times r})\}$
  $\mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
  $\mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$
  $\mathbf{x}_i \leftarrow \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \mathbf{a}_i^\top \mathbf{x}$ $\quad \{\mathsf{P}_{\mathcal{X}}(\cdot)\}$
  $\mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{x}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$
  $\mathbf{y}_i \leftarrow \operatorname{argmax}_{\mathbf{y}\in\mathcal{Y}} \mathbf{b}_i^\top \mathbf{y}$ $\quad \{\mathsf{P}_{\mathcal{Y}}(\cdot)\}$
  $\text{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{y}_i$
end for
$i_0 \leftarrow \arg\max_{i\in[T]} \text{obj}_i$
$(\mathbf{x}_\sharp, \mathbf{y}_\sharp) \leftarrow (\mathbf{x}_{i_0}, \mathbf{y}_{i_0})$

Algorithm 6.13 Maximization Oracle $\mathsf{P}_{\mathcal{X}}(\cdot)$ for $\mathcal{X} \triangleq \left\{\mathbf{x} \in \mathbb{S}_2^{m-1} : \|\mathbf{x}\|_0 \le s\right\}$

input $\mathbf{a}$ – d-dimensional real vector.
output $\mathbf{x}_0$ – d-dimensional vector that maximizes $\langle \mathbf{a}, \mathbf{x}\rangle$ over all $\mathbf{x} \in \mathcal{X}$.

$\mathbf{x}_0 \leftarrow \mathbf{0}_{d\times 1}$
$t \leftarrow$ index of sth order element of $\texttt{abs}(\mathbf{a})$
$\mathcal{I} \leftarrow \{i : |a_i| \ge |a_t|\}$
$\mathbf{x}_0[i] \leftarrow \mathbf{a}[i], \forall i \in \mathcal{I}$
$\mathbf{x}_0 \leftarrow \mathbf{x}_0 / \|\mathbf{x}_0\|_2$

guarantees that the output pair will be approximately optimal in terms of the low-rank CCA objective (6.4). That, it turn, translates to approximation guarantees for the sparse CCA problem on the original input $\mathbf{A}$:

Theorem 6.10. *For any real $m \times n$ matrix $\mathbf{A}$, $\epsilon \in (0, 1)$, and $r \le \max\{m, n\}$, Algorithm 6.12 with input $\mathbf{A}$, r, and $\mathrm{T} = \widetilde{O}\left(2^{r\cdot \log_2(2/\epsilon)}\right)$ outputs $\mathbf{x}_\sharp \in \mathcal{X}$ and*

$\mathbf{y}_\sharp \in \mathcal{Y}$ *such that*

$$\mathbf{x}_\sharp^\top \mathbf{A}\mathbf{y}_\sharp \geq \mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star - \epsilon \cdot \sigma_1(\mathbf{A}) - 2\sigma_{r+1}(\mathbf{A}),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{X}} + \mathrm{T}_{\mathcal{Y}} + r \cdot \max\{m, n\}))$.

Here, $\mathbf{x}_\star$ and $\mathbf{y}_\star$ denote the unknown optimal pair of canonical vectors satisfying the desired constraints. $\mathrm{T}_{\mathsf{SVD}}(r)$ denotes the time to compute the rank-r truncated SVD of the input $\mathbf{A}$, while $\mathrm{T}_{\mathcal{X}}$ and $\mathrm{T}_{\mathcal{Y}}$ denote the time required to compute the maximizations (6.5) and (6.6), respectively, which in the case of Alg. 6.13 are linear in the dimensions m and n.

The first term in the additive error is due to the sampling approach of Alg. 6.12. The second term is due to the fact that the algorithm operates on the rank-r surrogate matrix $\mathbf{B}$. Theorem 6.10 establishes a trade-off between the computational complexity of Alg. 6.12 and the quality of the approximation guarantees: the latter decreases as r increases, but the former depends exponentially in r.

Finally, in the special case where we impose sparsity constraints on only one of the two variables, say $\mathbf{x}$, while allowing the second variable, here $\mathbf{y}$, to be any vector with unit ℓ_2 norm, we obtain stronger guarantees.

Theorem 6.11. *If* $\mathcal{Y} = \{\mathbf{y} : \|\mathbf{y}\|_2 = 1\}$*, i.e., if no constraint is imposed on variable* $\mathbf{y}$ *besides unit length, then Algorithm 6.12 under the same configuration as that in Theorem 6.10 outputs* $\mathbf{x}_\sharp \in \mathcal{X}$ *and* $\mathbf{y}_\sharp \in \mathcal{Y}$ *such that*

$$\mathbf{x}_\sharp^\top \mathbf{A}\mathbf{y}_\sharp \geq (1 - \epsilon) \cdot \mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star - 2 \cdot \sigma_{r+1}(\mathbf{A}).$$

Theorem 6.11 implies that due to the flexibility in the choice of the canonical vector $\mathbf{y}$, Alg. 6.12 solves the low-rank sparse CCA problem (6.4) within a multiplicative $(1-\epsilon)$-factor from the optimal; the extra additive error term is once again due to the fact that the algorithm operates on the rank-r approximation $\mathbf{B}$ instead of the original input $\mathbf{A}$. In this case, the optimal choice of $\mathbf{y}$ in (6.6) is just a scaled version of the argument $\mathbf{b}$. A formal proof for Theorem 6.11 is provided in the Appendix, Sec. E.2.

Overall, SpanCCA is simple to implement and is trivially parallelizable: the main iteration can be split across and arbitrary number of processing units achieving a potentially linear speedup. It is the first algorithm for sparse CCA with data-dependent global approximation guarantees. As discussed in Theorem 6.10, the input accuracy parameter r establishes a trade-off between the running time and the tightness of the theoretical guarantees. Its complexity scales linearly in the dimensions of the input for any constant r, but admittedly becomes prohibitive even for moderate values of r. In practice, however, the spectrum of the data exhibits sharp decay leading to useful approximation guarantees even for small values of r such as 2 or 3. Moreover, disregarding the theoretical guarantees, the algorithm can always be executed for any rank r and an arbitrary number of iterations T in the allowable time window.

6.3.3 Analysis

Let $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, and $\mathbf{x}_{(\mathrm{B})}, \mathbf{y}_{(\mathrm{B})}$ be a pair that maximizes –not necessarily uniquely– the objective $\mathbf{x}^\top\mathbf{B}\mathbf{y}$ in (6.4) over all feasible solutions. We assume that $\mathbf{x}_{(\mathrm{B})}^\top\mathbf{B}\mathbf{y}_{(\mathrm{B})} > 0$.[4] Define the $r \times 1$ vector $\mathbf{c}_{(\mathrm{B})} \triangleq \mathbf{V}^\top\mathbf{y}_{(\mathrm{B})}$ and let ρ denote its ℓ_2-norm. Then, $0 < \rho \leq 1$; the upper bound follows from the fact that the r columns of $\mathbf{V}$ are orthonormal and $\|\mathbf{y}_{(\mathrm{B})}\|_2 = 1$, while the lower follows by the aforementioned assumption. . Finally, define $\overline{\mathbf{c}}_{(\mathrm{B})} = \mathbf{c}_{(\mathrm{B})}/\rho$, the projection of $\mathbf{c}_{(\mathrm{B})}$ on the unit ℓ_2-sphere $\mathbb{S}_2^{r-1}$.

Def. 6.3.2. *For any $\epsilon \in (0,1)$, an ϵ-net of $\mathbb{S}_2^{r-1}$ is a finite collection N of points in $\mathbb{R}$ such that for any $\mathbf{c} \in \mathbb{S}_2^{r-1}$, N contains a point $\mathbf{c}\prime$ such that $\|\mathbf{c}\prime - \mathbf{c}\| \leq \epsilon$.*

Lemma 6.3.8 ([140], Lemma 5.2)**.** *For any $\epsilon \in (0,1)$, there exists an ϵ-net of $\mathbb{S}_2^{r-1}$ equipped with the Euclidean metric, with at most $(1 + 2/\epsilon)^r$ points.*

Algorithm 6.12 runs in an iteration with T rounds. In each round, it independently samples a point $\mathbf{c}_i$ from $\mathbb{S}_2^{r-1}$, by randomly generating a vector according to a spherical Gaussian distribution and appropriately scaling its length. Based on Lemma 6.3.8 and elementary counting arguments, for sufficiently large T the collection of sampled points forms a ϵ-net of $\mathbb{S}_2^{r-1}$ with high probability:

[4] Observe that this is always true for any nonzero argument $\mathbf{B}$ as long as at least one of the two variables $\mathbf{x}$ and $\mathbf{y}$ can take arbitrary signs. It is hence true under vanilla sparsity constraints.

Lemma 6.3.9. *For any $\epsilon, \delta \in (0,1)$, a set of $\mathrm{T} = O(r(\epsilon/4)^{-r} \cdot \ln 4/\epsilon \cdot \delta)$ randomly and independently drawn points uniformly distributed on $\mathbb{S}_2^{r-1}$ suffices to construct an $\epsilon/2$-net of $\mathbb{S}_2^{r-1}$ with probability at least $1-\delta$.*

It follows that there exists $i_\star \in [\mathrm{T}]$ such that

$$\|\mathbf{c}_{i_\star} - \overline{\mathbf{c}}_{(\mathrm{B})}\|_2 \leq \epsilon/2. \tag{6.7}$$

In the $i_\star$th round, the algorithm samples the point $\mathbf{c}_{i_\star}$ and computes a feasible pair $(\mathbf{x}_{i_\star}, \mathbf{y}_{i_\star})$ via the two step maximization procedure, that is,

$$\mathbf{x}_{i_\star} \triangleq \operatorname*{argmax}_{\mathbf{x}\in\mathcal{X}} \mathbf{x}^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_{i_\star} \quad \text{and} \quad \mathbf{y}_{i_\star} \triangleq \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \mathbf{x}_{i_\star}^\top \mathbf{A}\mathbf{y}.$$

We have:

$$\begin{aligned}
\mathbf{x}_{(\mathrm{B})}^\top \mathbf{B}\mathbf{y}_{(\mathrm{B})} &= \rho \cdot \mathbf{x}_{(\mathrm{B})}^\top \mathbf{U}\boldsymbol{\Sigma}\overline{\mathbf{c}}_{(\mathrm{B})} \\
&= \rho \cdot \mathbf{x}_{(\mathrm{B})}^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_{i_\star} + \rho \cdot \mathbf{x}_{(\mathrm{B})}^\top \mathbf{U}\boldsymbol{\Sigma}(\overline{\mathbf{c}}_{(\mathrm{B})} - \mathbf{c}_{i_\star}) \\
&\leq \rho \cdot \mathbf{x}_{i_\star}^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_{i_\star} + \rho \cdot \mathbf{x}_{(\mathrm{B})}^\top \mathbf{U}\boldsymbol{\Sigma}(\overline{\mathbf{c}}_{(\mathrm{B})} - \mathbf{c}_{i_\star}) \\
&\leq \rho \cdot \mathbf{x}_{i_\star}^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_{i_\star} + \tfrac{\epsilon}{2} \cdot \sigma_1(\mathbf{A}).
\end{aligned} \tag{6.8}$$

The first step follows by the definition of $\overline{\mathbf{c}}_{(\mathrm{B})}$ and the second by linearity. The first inequality follows from the fact that $\mathbf{x}_{i_\star}$ maximizes the first term over all $\mathbf{x} \in \mathcal{X}$. The last inequality follows straightforwardly from the fact that $\|\mathbf{x}_{(\mathrm{B})}\|_2 = 1$ and $\rho \leq 1$ (see Lemma E.3.55). Using similar arguments,

$$\begin{aligned}
\rho \cdot \mathbf{x}_{i_\star}^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_{i_\star} &= \mathbf{x}_{i_\star}^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \mathbf{y}_{(\mathrm{B})} + \rho \cdot \mathbf{x}_{i_\star}^\top \mathbf{U}\boldsymbol{\Sigma}(\mathbf{c}_{i_\star} - \overline{\mathbf{c}}_{(\mathrm{B})}) \\
&\leq \mathbf{x}_{i_\star}^\top \mathbf{B}\mathbf{y}_{i_\star} + \tfrac{\epsilon}{2} \cdot \sigma_1(\mathbf{A}).
\end{aligned} \tag{6.9}$$

The inequality follows by the fact that $\mathbf{y}_{i_\star}$ maximizes the inner product with $\mathbf{B}^\top \mathbf{x}_{i_\star}$ over all $\mathbf{y} \in \mathcal{Y}$, as well as that $\|\mathbf{x}_{(\mathrm{B})}\|_2 = 1$ and $\rho \leq 1$. Combining (6.8) and (6.9), we obtain

$$\mathbf{x}_{i_\star}^\top \mathbf{B} \mathbf{y}_{i_\star} \geq \mathbf{x}_{(\mathrm{B})}^\top \mathbf{B} \mathbf{y}_{(\mathrm{B})} - \epsilon \cdot \sigma_1(\mathbf{A}). \tag{6.10}$$

Algorithm 6.12 computes multiple candidate solution pairs and outputs the pair $(\mathbf{x}_\sharp, \mathbf{y}_\sharp)$ that achieves the maximum objective value. The latter is at least as high as that achieved by $(\mathbf{x}_{i_\star}, \mathbf{y}_{i_\star})$.

Inequality (6.10) establishes an approximation guarantee for the low-rank sparse CCA problem (6.4). Those can be translated to guarantees on the original problem with input argument $\mathbf{A}$. Let $\mathbf{x}_\star$ and $\mathbf{y}_\star$ denote the (unknown) optimal solution of the sparse CCA problem (6.2). By the definition of $\mathbf{x}_{(\mathrm{B})}$ and $\mathbf{y}_{(\mathrm{B})}$, it follows that

$$\begin{aligned} \mathbf{x}_{(\mathrm{B})}^\top \mathbf{B} \mathbf{y}_{(\mathrm{B})} \geq \mathbf{x}_\star^\top \mathbf{B} \mathbf{y}_\star &= \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \mathbf{x}_\star^\top (\mathbf{A} - \mathbf{B}) \mathbf{y}_\star \\ &\geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \sigma_{r+1}(\mathbf{A}). \end{aligned} \tag{6.11}$$

Similarly,

$$\begin{aligned} \mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp &= \mathbf{x}_\sharp^\top \mathbf{B} \mathbf{y}_\sharp - \mathbf{x}_\sharp^\top (\mathbf{B} - \mathbf{A}) \mathbf{y}_\sharp \\ &\geq \mathbf{x}_\sharp^\top \mathbf{B} \mathbf{y}_\sharp - \sigma_{r+1}(\mathbf{A}). \end{aligned} \tag{6.12}$$

Combining (6.11) and (6.12) with (6.10), we obtain the approximation guarantees of Theorem 6.10.

The running time of Algorithm 6.12 follows straightforwardly by inspection. The algorithm first computes the truncated singular value decomposition of inner dimension r in time denoted by $\mathrm{T}_{\mathtt{svd}}(r)$. Subsequently, it performs T iterations. The cost of each iteration is determined by the cost of the matrix-vector multiplications and the running times $\mathrm{T}_{\mathcal{X}}$ and $\mathrm{T}_{\mathcal{Y}}$ of the operators $\mathsf{P}_{\mathcal{X}}(\cdot)$ and $\mathsf{P}_{\mathcal{Y}}(\cdot)$. Note that matrix multiplications can exploit the available matrix decomposition and are performed in time $r \cdot \max\{m, n\}$. Substituting the value of T with that specified in Lemma 6.3.9 completes the proof of Thm. 6.10. The proof of Theorem 6.11 follows a similar path; see Appendix Sec. E.2.

6.4 Beyond Sparsity: Structured CCA

While enforcing sparsity results in succinct models, the latter may fall short in capturing the true interactions in a physical system, especially when the number of samples is limited. Incorporating additional prior structural information can improve interpretability[5]; *e.g.*, [56] argue that a structure-aware sparse CCA incorporating group-like structure obtains biologically more meaningful results, while [101] demonstrated that group prior knowledge improved performance compared to standard sparse CCA in a task of identifying brain regions susceptible to schizophrenia. Several works suggest using structure-inducing regularizers to promote smoothness [148, 38, 88] or group sparse structure [42] in CCA.

[5]This is a shared insight in the broader area of sparse approximations [26, 72, 21, 92].

Our sparse CCA algorithm and its theoretical approximation guarantees in Theorems 6.10 and 6.11 extend straightforwardly to constraints beyond sparsity. In the overview of Algorithm 6.12 and its guarantees, we only made two assumptions on the feasible sets $\mathcal{X}$ and $\mathcal{Y}$: *i*) we seek canonical vectors with unit ℓ_2-norm, and *ii*) there exists a tractable procedure $\mathsf{P}_{\mathcal{X}}$ that solves the constrained maximization in (6.5), and similarly a procedure $\mathsf{P}_{\mathcal{Y}}$ for (6.6). The specific structure of the feasible sets only manifests itself in the implementation of those subroutines, *e.g.*, Alg. 6.13 for the case of sparsity constraints. Therefore, Alg. 6.12 can be straightforwardly adapted to any structural constraint for which the aforementioned conditions are satisfied.

In fact, observe that under the unit ℓ_2 restriction on the feasible vectors, the maximizations in (6.5) and (6.6) are equivalent to computing the Euclidean projection of a given real vector on the (nonconvex) sets $\mathcal{X}$ and $\mathcal{Y}$. Such exact or approximate projection procedures exist for several interesting constraints beyond sparsity such as smooth or group sparsity [72, 24, 91], sparsity constraints onto norm balls [93], or even sparsity patterns guided by underlying graphs [69, 13]. Of course, in the case where the projection is approximate, our theoretical guarantees would have to be adjusted accordingly.

6.5 Experiments

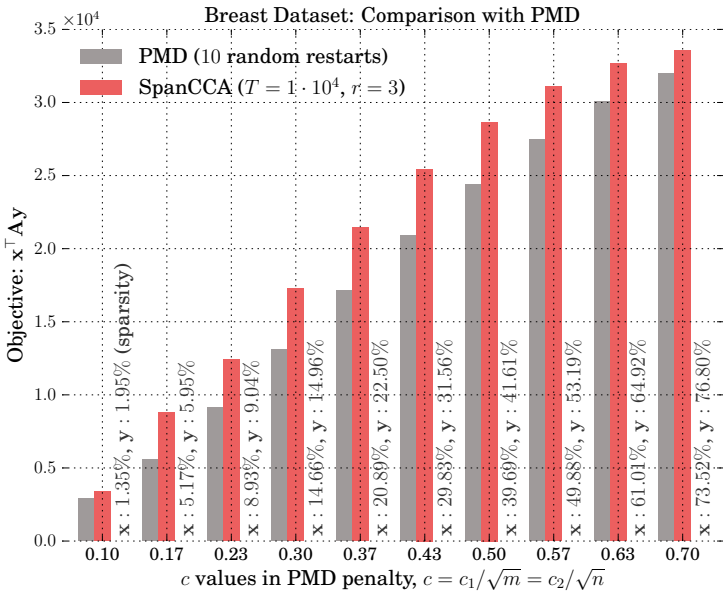
We empirically evaluate our sparse CCA algorithm on two real datasets: *i*) a publicly available breast cancer dataset [40], also used in the evaluation of the sparse CCA algorithm of [148], and *ii*) a neuroimaging dataset obtained

from the Human Connectome Project [137] on which we investigate associations between brain activation and behavior measurements.

6.5.1 Breast Cancer Dataset

The breast cancer dataset [40] consists of gene expression and DNA copy number measurements on a set of 89 tissue samples. Among others, it contains a 89×2149 matrix (DNA) with CGH spots for each sample and a 89×19672 matrix (RNA) of genes, along with information for the chromosomal locations of each CGH spot and each gene. As described in [148], this dataset can be used to perform integrative analysis of gene expression and DNA copy number data, and in particular to identify sets of genes that have expression that is correlated with a set of chromosomal gains or losses.

We run our sparse CCA algorithm on the breast cancer dataset and compare the output with the PMD algorithm of [148]; PMD is regarded as state of the art by practitioners and has been used –in its original form or slightly modified– in several neuroscience and biomedical applications; see also Section 6.2. The input to both algorithms is the $m \times n$ matrix $\mathbf{A} = \mathbf{X}^\top \mathbf{Y}$ ($m = 2149$, $n = 19672$), where $\mathbf{X}$ and $\mathbf{Y}$ are obtained from the aforementioned DNA and RNA matrices upon feature standardization. Recall that PMD is an iterative, alternating optimization scheme, where the sparsity of the extracted components $\mathbf{x}$ and $\mathbf{y}$ is implicitly controlled by enforcing upper bounds c_1 and c_2 on their ℓ_1-norm, respectively, with $1 \leq c_1 \leq \sqrt{m}$ and $1 \leq c_2 \leq \sqrt{n}$. Here, for simplicity, we set $c_1 = c\sqrt{m}$ and $c_2 = c\sqrt{n}$ and consider multiple val-



Algorithm	Avg Exec. Time	Configuration
PMD	$\sim$ 44 seconds	10 rand. restarts
SpanCCA	$\sim$ 24 seconds	T = 10^4, $r = 3$.

Figure 6.1: Comparison of SpanCCA and the PMD algorithm [148] on the Breast Cancer dataset [40]. We configure PMD with ℓ_1-norm thresholds $c_1 = c\cdot\sqrt{m}$ and $c_2 = c\cdot\sqrt{n}$, and consider various values of the constant $c \in (0, 1)$. For each c, we run PMD 10 times, select the canonical vectors $\mathbf{x}$, $\mathbf{y}$ that achieve the highest objective value and count their nonzero entries (depicted as percentage of the corresponding dimension). Finally, we run our SpanCCA algorithm with $T = 10^4$ and $r = 3$, using the latter as target sparsities, and compare the objective values achieved by the two methods. We also list the execution time for each algorithm for the aforementioned configurations; execution times remain approximately the same for all target sparsity values (equiv. all c).

ues of the constant c in $(0, 1)$. Note that under this configuration, for any given value of c, we expect that the extracted components will be approximately equally sparse, relatively to their dimension.

For each c, we first run the PMD algorithm 10 times with random initializations, determine the pair of components $\mathbf{x}$ and $\mathbf{y}$ that achieves the highest objective value, and count the number of nonzero entries of both components as a percentage of their corresponding dimension. Subsequently, we run SpanCCA (Alg. 6.12) with parameters $T = 10^4$, $r = 3$, and target sparsity equal to that of the former PMD output. Recall that our algorithm administers precise control on the number of nonzero entries of the extracted components.

Figure 6.1 depicts the CCA objective value achieved by the two algorithms, as well as the corresponding sparsity level of the extracted components. SpanCCA achieves a higher CCA objective value in all cases. Finally, note that under the above configuration, both algorithms run for a few seconds per target sparsity, with SpanCCA running approximately half the time of PMD.

6.5.2 Brain Imaging Dataset

We analyzed functional statistical maps and behavioral variables from 497 subjects available from the Human Connectome Project (HCP) [137]. The HCP consists of high-quality imaging and behavioral data, collected from a large sample of healthy adult subjects, motivated by the goal of advancing knowledge between human brain function and its association to behavior. We apply our sparse CCA algorithm to investigate the shared co-variation between patterns of brain activity as measured by the experimental tasks, and behavioral variables. We selected the same subset of behavioral variables examined by [128], which include scores from psychological tests, physiologi-

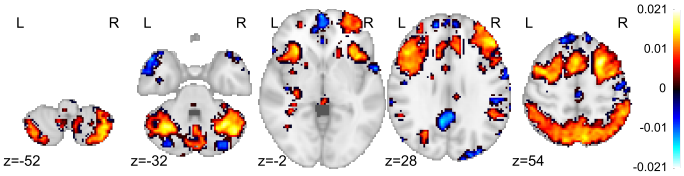
cal measurements, and self reported behavior questionnaires ($\mathbf{Y}$ dataset with dimensions 497×38).

For each subject, we collected statistical maps corresponding to 2-back task and the reasoning task. These statistical maps summarize the activation of each voxel in response to the experimental manipulation. In the "n-back" task, designed to measure working memory, items are presented one at a time and subjects identify each item that repeats relative to the item that occurred n items before. Further details on all tasks and variables are available in the HCP documentation [137].

We used the pre-computed 2-back / 0-back statistical contrast maps provided by the HCP. Standard preprocessing included motion correction, image registration to the MNI template (for comparison across subjects), and general linear model analysis, resulting in $91 \times 109 \times 91$ voxels. Voxels are then resampled to $61 \times 73 \times 61$ using the `nilearn` python package[6] and applying standard brain masks, resulting in 65598 voxels after masking non-grey matter regions. ($\mathbf{X}$ dataset with dimensions 497×65598).

We apply our SpanCCA algorithm on the HCP data with arbitrarily selected parameters $\mathrm{T} = 10^6$ and $r = 5$. We set the target sparsity at 15% for each canonical vector. Figure 6.2 depicts the brain regions and the behavioral factors corresponding to the nonzero weights of the extracted canonical pair. The map identifies a set of fronto-parietal regions known to be involved

[6] http://nilearn.github.io/



Behavioral Factor	Weight
PMAT24_A_CR	0.487
PicVocab_AgeAdj	0.448
ReadEng_AgeAdj	0.440
PicVocab_Unadj	0.433
ReadEng_Unadj	0.426

Figure 6.2: Brain regions and behavioral factors selected by the sparse left and right canonical vectors extracted by our SpanCCA algorithm. Target sparsity is set at 15% for each canonical vector and SpanCCA is configured to run for $T = 10^6$ samples operating on a rank $r = 5$ approximation of the input data. The map identifies a set of fronto-parietal regions known to be involved in executive function and working memory and deactivation in the default mode areas (medial prefrontal and parietal), which is also associated with engagement of difficult cognitive functions. The behavioral variables identified to be positively correlated with the activation of this network are all related to various aspects of intelligence.

in executive function and working memory, which are the major functions isolated by the 2-back / 0-back contrast. In addition, it identifies deactivation in the default mode areas (medial prefrontal and parietal), which is also associated with engagement of difficult cognitive functions. The behavioral variables associated with activation of this network are all related to various aspects of intelligence; the Penn Matrix Reasoning Test (a measure of fluid intelligence), picture vocabulary (a measure of language comprehension), and reading ability.

Parallelization To speed up execution, our prototypical `Python` implementation of SpanCCA exploits the `multiprocessing` module: N independent worker processes are spawned, and each one independently performs T/N

rounds of the main iteration of Alg. 6.12 returning a single canonical vector pair. The main process collects and compares the candidate pairs to determine the final output.

To demonstrate the parallelizability of our algorith, we run SpanCCA for the aforementioned task on the brain imaging data for various values of the number N of workers on a single server with 36 physical processing cores[7] and approximately 250Gb of main memory. In Figure 6.3 (top panel), we plot the run time with respect to the number of workers used. The bottom panel depicts the achieved speedup factor: using the execution time on 5 worker processes as a reference value, the speedup factor is the ratio of the execution time on 5 processes over that on N. As expected, the algorithm achieved a speedup factor that grows almost linearly in the number of available processors.

6.6 Discussion

We presented a novel combinatorial algorithm for sparse CCA and other constrained variants with provable data-dependent global approximation guarantees and several attractive properties: the algorithm is simple, embarrassingly parallelizable, with complexity that scales linearly in the dimension of the input data, while it administers precise control on the sparsity of the extracted canonical vectors. Further it can accommodate additional structural

[7]Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz

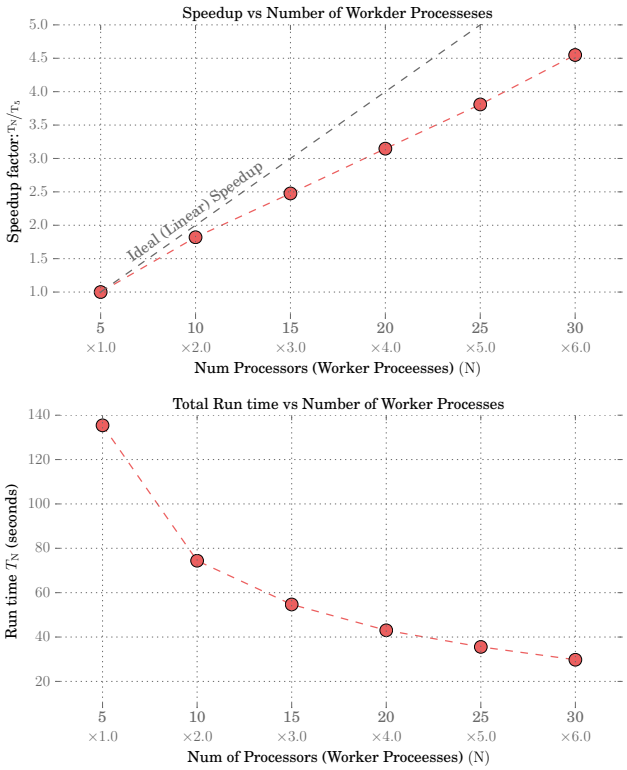


Figure 6.3: Speedup factors and corresponding total execution time, achieved by the prototypical parallel implementation of SpanCCA (Alg. 6.12) as a function of the number of worker processes or equivalently the number of processors used. Depicted values are medians over 20 executions, each with $T = 10^5$ and $r = 5$, on the 65598×38 example discussed in section 6.5.2. A speedup factory approximately linear in the number of workers is achieved.

constraints by plugging in a suitable "projection" subroutine.

Several directions remain open. We addressed the question of computing a single pair of sparse canonical vectors. Numerically, multiple pairs can be computed successively employing an appropriate deflation step. However, determining the kind of deflation most suitable for the application at hand, as

well as the sparsity level of each component can be a challenging task, leaving a lot of room for research. As evidenced by the several works in scientific research (*e.g.*, in neuroscience or bioinformatics) that resort to CCA for investigating associations across datasets, exploring algorithmic solutions that yield meaningful results becomes increasingly important.

Chapter 7

Bipartite Correlation Clustering

In *Bipartite Correlation Clustering* (BCC) we are given a complete bipartite graph G with $+$ and $-$ edges, and we seek a vertex clustering that maximizes the number of *agreements*: the number of all $+$ edges within clusters plus all $-$ edges cut across clusters. BCC is known to be NP-hard [9].

We present a novel approximation algorithm for k-BCC, a variant of BCC with an upper bound k on the number of clusters. Our algorithm outputs a k-clustering that provably achieves a number of agreements within a multiplicative $(1-\delta)$-factor from the optimal, for any desired accuracy δ. It relies on solving a combinatorially constrained bilinear maximization on the bi-adjacency matrix of G. It runs in time exponential in k and $1/\delta$, but linear in the size of the input.

Further, we show that, in the (unconstrained) BCC setting, an $(1-\delta)$-approximation can be achieved by $O(\delta^{-1})$ clusters regardless of the size of the

Chapter 7 is based on material from Reference [15]: Megasthenis Asteris, Anastasios Kyrillidis, Dimitris Papailiopoulos, and Alexandros Dimakis, *"Bipartite Correlation Clustering: Maximizing Agreements"*, In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, pp. 121–129, 2016. The author of this dissertation is the lead author of [15], and contributed to the conception of the research problem, the theoretical and analytical developments, the experimental design and implementation, and the writing of the manuscript and its revisions.

graph. In turn, our k-BCC algorithm implies an Efficient PTAS for the BCC objective of maximizing agreements.

7.1 Introduction

Correlation Clustering (CC) [25] considers the task of partitioning a set of objects into clusters based on their pairwise relationships. It arises naturally in several areas such as in network monitoring [1], document clustering [25] and textual similarity for data integration [44]. In its simplest form, objects are represented as vertices in a complete graph whose edges are labeled $+$ or $-$ to encode *similarity* or *dissimilarity* among vertices, respectively. The objective is to compute a vertex partition that maximizes the number of *agreements* with the underlying graph, *i.e.*, the total number of $+$ edges in the interior of clusters plus the number of $-$ edges across clusters. The number of output clusters is itself an optimization variable –not part of the input. It may be meaningful, however, to restrict the number of output clusters to be at most k. The constrained version is known as k-CC and similarly to the unconstrained problem, it is NP-hard [25, 63, 84]. A significant volume of work has focused on approximately solving the problem of maximizing the number of agreements (MaxAgree), or the equivalent –yet more challenging in terms of approximation– objective of minimizing the number of *disagreements* (MinDisagree) [25, 53, 36, 3, 138, 4].

Bipartite Correlation Clustering (BCC) is a natural variant of CC on bipartite graphs. Given a complete bipartite graph $G = (U, V, E)$ with

edges labeled + or −, the objective is once again to compute a clustering of the vertices that maximizes the number of agreements with the labeled pairs. BCC is a special case of the *incomplete* CC problem [36], where only a subset of vertices are connected and non-adjacent vertices do not affect the objective function. The output clusters may contain vertices from either one or both sides of the graph. Finally, we can define the k-BCC variant that enforces an upper bound on the number of output clusters, similar to k-CC for CC.

The task of clustering the vertices of a bipartite graph is common across many areas of machine learning. Applications include recommendation systems [132, 141], where analyzing the structure of a large sets of pairwise interactions (*e.g.*, among users and products) allows useful predictions about future interactions, gene expression data analysis [9, 104] and graph partitioning problems in data mining [59, 158].

Despite the practical interest in bipartite graphs, there is limited work on BCC and it is focused on the theoretically more challenging MinDisagree objective: [9] established an 11-approximation algorithm, while [2] achieved a 4-approximation, the currently best known guarantee. Algorithms for incomplete CC [53, 36, 131] can be applied to BCC, but they do not leverage the structure of the bipartite graph. Moreover, existing approaches for incomplete CC rely on LP or SDP solvers which scale poorly.

Our contributions We develop a novel approximation algorithm for k-BCC with provable guarantees for the MaxAgree objective. Further,

we show that under an appropriate configuration, our algorithm yields an Efficient Polynomial Time Approximation Scheme (EPTAS[1]) for the unconstrained BCC problem. Our contributions can be summarized as follows:

1. k-BCC: Given a bipartite graph $G = (U, V, E)$, a parameter k, and any constant accuracy parameter $\delta \in (0, 1)$, our algorithm computes a clustering of $U \cup V$ into at most k clusters and achieves a number of agreements that lies within a $(1 - \delta)$-factor from the optimal. It runs in time exponential in k and $1/\delta$, but linear in the size of G.

2. BCC: In the unconstrained BCC setting, the optimal number of clusters may be anywhere from 1 to $|U| + |V|$. We show that if one is willing to settle for a $(1 - \delta)$-approximation of the MaxAgree objective, it suffices to use at most $O(\delta^{-1})$ clusters, regardless of the size of G. In turn, under an appropriate configuration, our k-BCC algorithm yields an EPTAS for the (unconstrained) BCC problem.

3. Our algorithm relies on formulating the k-BCC/ MaxAgree problem as a combinatorially constrained bilinear maximization

$$\max_{\mathbf{X} \in \mathcal{X}, \mathbf{Y} \in \mathcal{Y}} \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B} \mathbf{Y}\Big), \tag{7.1}$$

[1] EPTAS refers to an algorithm that approximates the solution of an optimization problem within a multiplicative $(1 - \epsilon)$-factor, for any constant $\epsilon \in (0, 1)$, and has complexity that scales arbitrarily in $1/\epsilon$, but as a constant order polynomial (independent of ϵ) in the input size n. EPTAS is more efficient than a PTAS; for example, a running time of $O(n^{1/\epsilon})$ is considered a PTAS, but not an EPTAS.

where $\mathbf{B}$ is the bi-adjacency matrix of G, and $\mathcal{X}$, $\mathcal{Y}$ are the sets of cluster assignment matrices for U and V, respectively. In Sec. 7.4, we briefly describe our approach for approximately solving (7.1) and its guarantees under more general constraints.

We note that our k-BCC algorithm and its guarantees can be extended to *incomplete*, but dense BCC instances, where the input G is *not* a complete bipartite graph, but $|E| = \Omega(|U| \cdot |V|)$. For simplicity, we restrict the description to the complete case. Finally, we supplement our theoretical findings with experimental results on synthetic and real datasets.

7.2 Related work

There is extensive literature on CC; see [29] for a list of references. Here, we focus on bipartite variants.

BCC was introduced by Amit in [9] along with an 11-approximation algorithm for the MinDisagree objective, based on a linear program (LP) with $O(|E|^3)$ constraints. In [2], Ailon et al. proposed two algorithms for the same objective: *i*) a deterministic 4-approximation based on an LP formulation and de-randomization arguments by [138], and *ii*) a randomized 4-approximation combinatorial algorithm. The latter is computationally more efficient with complexity scaling linearly in the size of the input, compared to the LP that has $O\Big((|V| + |U|)^3\Big)$ constraints.

For the *incomplete* CC problem, which encompasses BCC as a special

case, [53] provided an LP-based $O(\log n)$-approximation for MinDisagree. A similar result was achieved by [36]. For the MaxAgree objective, [2, 131] proposed an SDP relaxation, similar to that for MAX k-CUT, and achieved a 0.7666-approximation. We are not aware of any results explicitly on the MaxAgree objective for either BCC or k-BCC. For comparison, in the k-CC setting [25] provided a simple 3-approximation algorithm for $k = 2$, while for $k \geq 2$ [63] provided a PTAS for MaxAgree and one for MinDisagree. [84] improved on the latter utilizing approximations schemes to the Gale-Berlekamp switching game. Table 7.1 summarizes the aforementioned results.

Finally, we note that our algorithm relies on adapting ideas from [19] for approximately solving a combinatorially constrained quadratic maximization.

7.3 k-BCC MaxAgree as a Bilinear Maximization

We describe the k-BCC problem and show that it can be formulated as a combinatorially constrained bilinear maximization on the bi-adjacency matrix of the input graph. Our k-BCC algorithm relies on approximately solving that bilinear maximization.

Maximizing Agreements An instance of consists of an undirected, complete, bipartite graph $G = (U, V, E)$ whose edges have binary ± 1 weights, and an integer parameter $1 \leq k \leq |U| + |V|$. The objective, hereafter referred to as MaxAgree$[k]$, is to compute a clustering of the vertices into at most k clusters, such that the total number of positive edges in the interior of clusters

Ref.	Setting	Objective	Guarantee	Complexity	D/R
[63]	k-CC	MaxAgree	(E)PTAS	$n/\delta \cdot k^{O\left(\delta^{-2}\log k \log(1/\delta)\right)}$	R
[63]	k-CC	MinDisagree	PTAS	$n^{O(100^k/\delta^2)} \cdot \log n$	R
[84]	k-CC	MinDisagree	PTAS	$n^{O(9^k/\delta^2)} \cdot \log n$	R
[53, 36]	Inc. CC	MinDisagree	$\times O(\log n)$	LP	D
[131]	Inc. CC	MaxAgree	$\times 0.766$	SDP	D
[9]	BCC	MinDisagree	$\times 11$	LP	D
[2]	BCC	MinDisagree	$\times 4$	LP	D
[2]	BCC	MinDisagree	$\times 4$	$\lvert E\rvert$	R
Ours	k-BCC	MaxAgree	(E)PTAS	$2^{O(k/\delta^2 \cdot \log \sqrt{k}/\delta)} \cdot (\delta^{-2}+k) \cdot n + \mathrm{T}_{\mathsf{SVD}}(\delta^{-2})$	R
	BCC	MaxAgree	(E)PTAS	$2^{O(\delta^{-3} \cdot \log \delta^{-3})} \cdot O(\delta^{-2}) \cdot n + \mathrm{T}_{\mathsf{SVD}}(\delta^{-2})$	R

Table 7.1: Summary of existing results on BCC and related literature. For each scheme, we indicate the problem setting, the objective (MaxAgree/ MinDisagree), the guarantees ($\times c$ implies a multiplicative factor approximation), and its computational complexity (n denotes the total number of vertices, LP/SDP denotes the complexity of a linear/semidefinite program, and $\mathrm{T}_{\mathsf{SVD}}(r)$ the time required to compute a rank-r truncated SVD of a $n \times n$ matrix). The D/R column indicates whether the scheme is deterministic or randomized.

plus the number of negative edges across clusters is maximized.

Let E^+ and E^- denote the sets of positive and negative edges, respectively. Also, let $m = |U|$ and $n = |V|$. The bipartite graph G can be represented by its weighted bi-adjacency matrix $\mathbf{B} \in \{\pm 1\}^{m \times n}$, where B_{ij} is equal to the weight of edge (i, j).

Consider a clustering $\mathcal{C}$ of the vertices $U \cup V$ into *at most* k clusters $C_1, \ldots, C_k$. Let $\mathbf{X} \in \{0,1\}^{m \times k}$ be the cluster assignment matrix for the vertices of U associated with $\mathcal{C}$, that is, $X_{ij} = 1$ if and only if vertex $i \in U$ is assigned to cluster C_j. Similarly, let $\mathbf{Y} \in \{0,1\}^{n \times k}$ be the cluster assignment for V.

Lemma 7.3.10. *For any instance $G = (U, V, E)$ of the k-*BCC *problem with bi-adjacency matrix $\mathbf{B} \in \{\pm 1\}^{|U| \times |V|}$, and for any clustering $\mathcal{C}$ of $U \cup V$ into k clusters, the number of agreements achieved by $\mathcal{C}$ is*

$$\textsc{Agree}(\mathcal{C}) = \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B} \mathbf{Y}\Big) + |E^-|,$$

where $\mathbf{X} \in \{0,1\}^{|U| \times k}$ and $\mathbf{Y} \in \{0,1\}^{|V| \times k}$ are the cluster assignment matrices for the vertex sets U and V, respectively, corresponding to the clustering $\mathcal{C}$.

Proof. Let $\mathbf{B}^+ \in \{0,1\}^{|U| \times |V|}$ be the indicator for the set of positive edges E^+, and similarly, $\mathbf{B}^- \in \{0,1\}^{|U| \times |V|}$ be the indicator for E^-. Then, $\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^-$. Given a clustering $\mathcal{C} = \{C_j\}_{j=1}^k$, or equivalently the assignment matrices $\mathbf{X} \in \{0,1\}^{|U| \times k}$ and $\mathbf{Y} \in \{0,1\}^{|V| \times k}$, the number of pairs of similar vertices assigned to the same cluster is equal to $\textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B}^+ \mathbf{Y}\Big)$. Similarly, the number of pairs of dissimilar vertices assigned to different clusters is equal to

$|E^-| - \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B}^- \mathbf{Y}\Big)$. The total number of agreements achieved by $\mathcal{C}$ is

$$\begin{aligned}\textsc{Agree}(\mathcal{C}) &= \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B}^+ \mathbf{Y}\Big) + |E^-| - \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B}^- \mathbf{Y}\Big) \\ &= \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B} \mathbf{Y}\Big) + |E^-|,\end{aligned}$$

which is the desired result. □

It follows that computing a k-clustering that achieves the maximum number of agreements, boils down to a constrained bilinear maximization:

$$\textsc{MaxAgree}[k] = \max_{\mathbf{X}\in\mathcal{X}, \mathbf{Y}\in\mathcal{Y}} \textsc{Tr}\Big(\mathbf{X}^\top \mathbf{B} \mathbf{Y}\Big) + |E^-|, \tag{7.2}$$

where

$$\begin{aligned}\mathcal{X} &\triangleq \Big\{\mathbf{X} \in \{0,1\}^{m\times k} : \|\mathbf{X}\|_{\infty,1} = 1\Big\}, \\ \mathcal{Y} &\triangleq \Big\{\mathbf{Y} \in \{0,1\}^{n\times k} : \|\mathbf{Y}\|_{\infty,1} = 1\Big\}.\end{aligned} \tag{7.3}$$

Here, $\|\mathbf{X}\|_{\infty,1}$ denotes the maximum of the ℓ_1-norm of the rows of $\mathbf{X}$. Since $\mathbf{X} \in \{0,1\}^{|U|\times k}$, the constraint ensures that each row of $\mathbf{X}$ has exactly one nonzero entry. Hence, $\mathcal{X}$ and $\mathcal{Y}$ describe the sets of valid cluster assignment matrices for U and V, respectively.

In the sequel, we briefly describe our approach for approximately solving the maximization in (7.2) under more general constraints, and subsequently apply it to the k-BCC/MaxAgree problem.

7.4 An Algorithm for the Bilinear Maximization

We describe a simple algorithm for computing an approximate solution to the constrained maximization

$$\max_{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}} \quad \textsc{Tr}\Big(\mathbf{X}^\top\mathbf{A}\mathbf{Y}\Big), \tag{7.4}$$

where the input argument $\mathbf{A}$ is a real $m \times n$ matrix, and $\mathcal{X}$, $\mathcal{Y}$ are norm-bounded sets. Our approach is exponential in the rank of the argument $\mathbf{A}$, which can be prohibitive in practice. To mitigate this effect, we solve the maximization on a low-rank approximation $\mathbf{A}_r$ of $\mathbf{A}$, instead. The quality of the output depends on the spectrum of $\mathbf{A}$ and the rank r of the surrogate matrix. Alg. 7.14 outlines our approach for solving (7.4), operating directly on the rank-r matrix $\mathbf{A}_r$.

If we knew the optimal value for variable $\mathbf{Y}$, then the optimal value for variable $\mathbf{X}$ would be the solution to

$$\mathrm{P}_{\mathcal{X}}(\mathbf{L}) \triangleq \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X}} \textsc{Tr}\Big(\mathbf{X}^\top\mathbf{L}\Big), \tag{7.5}$$

for $\mathbf{L} = \mathbf{A}_r\mathbf{Y}$. Similarly, with $\mathbf{R} = {\mathbf{A}_r}^\top\mathbf{X}$ for a given $\mathbf{X}$,

$$\mathrm{P}_{\mathcal{Y}}(\mathbf{R}) \triangleq \operatorname*{argmax}_{\mathbf{Y}\in\mathcal{Y}} \textsc{Tr}\Big(\mathbf{R}^\top\mathbf{Y}\Big) \tag{7.6}$$

is the optimal value of $\mathbf{Y}$ for that $\mathbf{X}$. Our algorithm requires that such a linear maximization oracles $\mathrm{P}_{\mathcal{X}}(\cdot)$ and $\mathrm{P}_{\mathcal{Y}}(\cdot)$ exist.[2] Of course, the optimal

[2] In the case of the k-BCC problem (7.2), where $\mathcal{X}$ and $\mathcal{Y}$ correspond to the sets of cluster assignment matrices defined in (7.3), such optimization oracles exist (see Alg. 7.15).

value for either of the two variables is not known. It is known, however, that the columns of the $m \times k$ matrix $\mathbf{L} = \mathbf{A}_r\mathbf{Y}$ lie in the r-dimensional range of $\mathbf{A}_r$ for all feasible $\mathbf{Y} \in \mathcal{Y}$. Alg. 7.14 effectively operates by sampling candidate values for $\mathbf{L}$. More specifically, it considers a large (exponential in $r \cdot k$) collection of $r \times k$ matrices $\mathbf{C}$ whose columns are points of an ϵ-net of the unit ℓ_2-ball $\mathbb{B}_2^{r-1}$. Each matrix $\mathbf{C}$ is used to generate a matrix $\mathbf{L}$ whose k columns lie in the range of the input matrix $\mathbf{A}_r$ and in turn to produce a feasible solution pair $\mathbf{X}, \mathbf{Y}$ by successively solving (7.5) and (7.6). Due to the properties of the ϵ-net, one of the computed feasible pairs is guaranteed to achieve an objective value in (7.4) close to the optimal one (for argument $\mathbf{A}_r$).

Lemma 7.4.11. *For any real $m \times n$, rank-r matrix $\mathbf{A}_r$, and sets $\mathcal{X} \subset \mathbb{R}^{m\times k}$ and $\mathcal{Y} \subset \mathbb{R}^{n\times k}$ (bounded under the Frobenious norm), let*

$$(\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}) \triangleq \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}} \mathrm{TR}\Big(\mathbf{X}^\top \mathbf{A}_r \mathbf{Y}\Big).$$

Assuming the existence of linear maximization oracles $\mathrm{P}_\mathcal{X}$ and $\mathrm{P}_\mathcal{Y}$ as in (7.5) and (7.6), then Alg. 7.14 with input $\mathbf{A}_r$ and accuracy $\epsilon \in (0, 1)$ outputs $\mathbf{X}^{(r)} \in \mathcal{X}$ and $\mathbf{Y}^{(r)} \in \mathcal{Y}$ such that

$$\mathrm{TR}\Big(\mathbf{X}^{(r)\top} \mathbf{A}_r \mathbf{Y}^{(r)}\Big) \geq \mathrm{TR}\Big(\mathbf{X}_\star^{(r)\top} \mathbf{A}_r \mathbf{Y}_\star^{(r)}\Big) - 2\epsilon\sqrt{k} \cdot \|\mathbf{A}_r\|_2 \cdot \mu_\mathcal{X} \cdot \mu_\mathcal{Y},$$

in time $O\Big((2\sqrt{r}/\epsilon)^{r\cdot k} \cdot (\mathrm{T}_\mathcal{X} + \mathrm{T}_\mathcal{Y} + (m+n)r)\Big) + \mathrm{T}_{\mathsf{SVD}}(r)$. Here, $\mu_\mathcal{X}$ and $\mu_\mathcal{Y}$ are such that $\sup_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\|_\mathrm{F} \leq \mu_\mathcal{X}$ and $\sup_{\mathbf{Y}\in\mathcal{Y}} \|\mathbf{Y}\|_\mathrm{F} \leq \mu_\mathcal{Y}$. $\mathrm{T}_{\mathsf{SVD}}(r)$ denotes the time required to compute the truncated SVD of the rank-r matrix $\mathbf{A}_r$, while $\mathrm{T}_\mathcal{X}$ and $\mathrm{T}_\mathcal{Y}$ denote the running time of the maximization oracles $\mathrm{P}_\mathcal{X}$ and $\mathrm{P}_\mathcal{Y}$, respectively.

Algorithm 7.14 BilinearMaximization

input $\mathbf{A}_r$ – $m \times n$ real matrix of rank r
ϵ – Accuracy parameter in $(0, 1)$
output $\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}$ – Pair of matrices in $\mathcal{X} \times \mathcal{Y}$. See Lemma 7.4.11 for guarantees

$\mathcal{C} \leftarrow \{\}$ {Candidate solutions}
$\widetilde{\mathbf{U}}, \widetilde{\mathbf{\Sigma}}, \widetilde{\mathbf{V}} \leftarrow \mathtt{svd}(\mathbf{A}_r)$ $\{\ \widetilde{\mathbf{\Sigma}} \in \mathbb{R}^{r \times r}\ \}$
for each $\mathbf{C} \in (\epsilon\text{-net of } \mathbb{B}_2^{r-1})^{\otimes k}$ **do**
  $\mathbf{L} \leftarrow \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\mathbf{C}$ $\{\mathbf{L} \in \mathbb{R}^{m \times k}\}$
  $\mathbf{X} \leftarrow \mathrm{P}_{\mathcal{X}}(\mathbf{L})$
  $\mathbf{R} \leftarrow \mathbf{X}^\top \mathbf{A}_r$ $\{\mathbf{R} \in \mathbb{R}^{k \times n}\}$
  $\mathbf{Y} \leftarrow \mathrm{P}_{\mathcal{Y}}(\mathbf{R})$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{X}, \mathbf{Y})\}$
end for
$(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}) \leftarrow \arg\max_{(\mathbf{X},\mathbf{Y}) \in \mathcal{C}} \mathrm{Tr}\Big(\mathbf{X}^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^\top \mathbf{Y}\Big)$

The crux of Lemma 7.4.11 is that if there exists an efficient procedure to solve the simpler maximizations (7.5) and (7.6), then we can approximately solve the bilinear maximization (7.4) in time that depends exponentially on the intrinsic dimension r of the input $\mathbf{A}_r$, but polynomially on its dimensions. A formal proof is given in Appendix Sec. F.1.

Recall that $\mathbf{A}_r$ is only a low-rank approximation of the potentially full rank original argument $\mathbf{A}$. The guarantees of Lemma 7.4.11 can be translated to guarantees for the original matrix introducing an additional error term to account for the extra level of approximation.

Lemma 7.4.12. *For any* $\mathbf{A} \in \mathbb{R}^{m \times n}$*, let*

$$(\mathbf{X}_\star, \mathbf{Y}_\star) \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}, \mathbf{Y} \in \mathcal{Y}} \mathrm{Tr}\Big(\mathbf{X}^\top \mathbf{A} \mathbf{Y}\Big),$$

where $\mathcal{X}$ *and* $\mathcal{Y}$ *satisfy the conditions of Lemma 7.4.11. Let* $\mathbf{A}_r$ *be a rank-*r *approximation of* $\mathbf{A}$*, and* $\mathbf{X}^{(r)} \in \mathcal{X}$*,* $\mathbf{Y}^{(r)} \in \mathcal{Y}$ *be the output of Alg. 7.14 with*

input $\mathbf{A}_r$ *and accuracy* ϵ*. Then,*

$$\textsc{Tr}\Big(\mathbf{X}_\star^\top \mathbf{A}\mathbf{Y}_\star\Big) - \textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}\mathbf{Y}^{(r)}\Big) \leq 2\cdot\Big(\epsilon\sqrt{k}\cdot\|\mathbf{A}_r\|_2 + \|\mathbf{A}-\mathbf{A}_r\|_2\Big)\cdot\mu_{\mathcal{X}}\cdot\mu_{\mathcal{Y}}.$$

Lemma 7.4.12 follows from Lemma 7.4.11. A formal proof is deferred to Appendix Sec. F.1. This concludes the brief discussion on our bilinear maximization framework.

7.5 An Efficient PTAS for k-BCC

The k-BCC/MaxAgree problem on a bipartite graph $G = (U, V, E)$ can be written as a constrained bilinear maximization (7.2) on the bi-adjacency matrix $\mathbf{B}$ over the sets of valid cluster assignment matrices (7.3) for V and U. Adopting the bilinear maximization algorithm of the previous section to this particular constraint set we obtain our k-BCC algorithm.

The missing ingredient is a pair of efficient procedures $\mathrm{P}_{\mathcal{X}}(\cdot)$ and $\mathrm{P}_{\mathcal{Y}}(\cdot)$, as described in Lemma 7.4.11 for solving (7.5) and (7.6), respectively, in the case where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of valid cluster assignment matrices (7.3). Such a procedure exists and is outlined in Alg. 7.15.

Algorithm 7.15 Max. Oracle $\mathrm{P}_{\mathcal{X}}(\cdot)$, for $\mathcal{X} = \Big\{\mathbf{X}\in\{0,1\}^{m\times k} : \|\mathbf{X}\|_{\infty,1} = 1\Big\}$

input $\mathbf{L}$ – $m\times k$ real matrix
output $\overline{\mathbf{X}}$ – $m\times k$ matrix in $\mathcal{X}$ that maximizes $\langle\mathbf{X},\mathbf{L}\rangle$ over all $\mathbf{X}\in\mathcal{X}$

$\overline{\mathbf{X}} \leftarrow \mathbf{0}_{m\times k}$
for $i = 1,\ldots,m$ **do**
  $j_i \leftarrow \mathrm{argmax}_{j\in[k]} L_{ij}$
  $\overline{X}_{ij_i} \leftarrow 1$
end for

Lemma 7.5.13. *For any* $\mathbf{L} \in \mathbb{R}^{m \times k}$*, and* $\mathcal{X}$ *as defined in* (7.3)*, Algorithm 7.15 outputs* $\mathbf{X} \in \mathcal{X}$ *that maximizes* $\langle \mathbf{X}, \mathbf{L} \rangle$ *in time* $O(k \cdot m)$*.*

Proof. Appendix, Sec. F.1. □

Note that in our case, Alg. 7.15 is used as both $\mathrm{P}_{\mathcal{X}}(\cdot)$ and $\mathrm{P}_{\mathcal{Y}}(\cdot)$. Putting the pieces together, we obtain the core of our k-BCC algorithm, outlined in Alg. 7.16. Given a bipartite graph G, with bi-adjacency matrix $\mathbf{B}$, we first compute a rank-r approximation $\widetilde{\mathbf{B}}$ of $\mathbf{B}$ via the truncated SVD. Using Alg. 7.14, equipped with Alg. 7.15 as a subroutine, we approximately solve the bilinear maximization (7.2) with argument $\widetilde{\mathbf{B}}$. The output is a pair of valid cluster assignment matrices for U and V.

Alg. 7.16 exposes the configuration parameters r and ϵ that control the performance of our bilinear solver, and in turn the quality of the output k-clustering. To simplify the description, we also create a k-BCC "wrapper" procedure outlined in Alg. 7.17: given a bound k on the number of clusters and a single accuracy parameter $\delta \in (0, 1)$, Alg. 7.17 configures and invokes Alg. 7.16.

Theorem 7.12. (k-BCC.) *For any instance* $(G = (U, V, E), k)$ *of the* k*-BCC problem and for any desired accuracy parameter* $\delta \in (0, 1)$*, Algorithm 7.17 computes a clustering* $\mathcal{C}_k^{(r)}$ *of* $U \cup V$ *into at most* k *clusters, such that*

$$\textsc{Agree}\left(\mathcal{C}_k^{(r)}\right) \geq (1 - \delta) \cdot \textsc{Agree}(\mathcal{C}_{k\star}),$$

Algorithm 7.16 k-BCC core algorithm via low-rank bilinear maximization

input G – bipartite graph $G = (U, V, E)$
k – the target number of clusters
ϵ – Accuracy parameter in $(0, 1)$
r – Accuracy parameter controlling the rank of approximation to be used in the bilinear maximization; will be set as a function of ϵ.

output $\mathcal{C}_k^{(r)}$ – Clustering of $U \cup V$ into k clusters
$(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)})$ – (Equivalently, cluster assignment matrices for U and V, respectively; $\mathbf{X}^{(r)} \in \{0,1\}^{|U| \times k}$ and $\mathbf{Y}^{(r)} \in \{0,1\}^{|V| \times k}$).

1: $\mathbf{B} \leftarrow$ bi-adjacency matrix of $G = (U, V, E)$.
2: $\widetilde{\mathbf{B}} \leftarrow \mathtt{svd}(\mathbf{B}, r)$ {Truncated SVD.}
3: Invoke Alg. 7.14 to approximately solve (7.2) with argument $\widetilde{\mathbf{B}}$, over the sets $\mathcal{X}$, $\mathcal{Y}$ of valid cluster assignment matrices defined in (7.3); Use Alg. 7.15 as a subroutine for both linear maximization oracles $\mathrm{P}_{\mathcal{X}}(\cdot)$ and $\mathrm{P}_{\mathcal{Y}}(\cdot)$:
$\mathbf{X}^{(r)}, \mathbf{Y}^{(r)} \leftarrow \textsc{BilinearMaximization}\left(\widetilde{\mathbf{B}}, \epsilon, r\right)$

Algorithm 7.17 k-BCC/MaxAgree

input G – Bipartite graph $G = (U, V, E)$.
k – Target number of clusters.
δ – Accuracy parameter in $(0, 1)$.

output $\mathcal{C}_k^{(r)}$ – A clustering of $U \cup V$ into (at most) k clusters. See Thm. 7.12.

1: Set up parameters ϵ and r as a function of k and δ:
$\epsilon \leftarrow 2^{-3} \cdot \delta \cdot k^{-1/2}, \quad r \leftarrow 2^6 \cdot \delta^{-2} - 1.$
2: Return output of Alg. 7.16 for input (G, k, r, ϵ).

in time $2^{O(k/\delta^2 \cdot \log \sqrt{k}/\delta)} \cdot (\delta^{-2} + k) \cdot (|U| + |V|) + \mathrm{T}_{\mathsf{SVD}}(\delta^{-2})$. *Here,* $\mathcal{C}_{k\star}$ *denotes the optimal clustering into at most* k *clusters.*

In the remainder of this section, we prove Theorem 7.12. We begin with the core Alg. 7.16, which invokes the low-rank bilinear solver Alg. 7.14 with accuracy parameters ϵ and r on the rank-r matrix $\widetilde{\mathbf{B}}$ and computes a

pair of valid cluster assignment matrices $\mathbf{X}^{(r)}$, $\mathbf{Y}^{(r)}$. By Lemma 7.4.11, and taking into account that $\sigma_1(\widetilde{\mathbf{B}}) = \sigma_1(\mathbf{B}) \leq \|\mathbf{B}\|_{\mathrm{F}} \leq \sqrt{mn}$, and the fact that $\|\mathbf{X}\|_{\mathrm{F}} = \sqrt{m}$ and $\|\mathbf{Y}\|_{\mathrm{F}} = \sqrt{n}$ for all valid cluster assignment matrix pairs, the output pair satisfies

$$\textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top}\widetilde{\mathbf{B}}\mathbf{Y}_\star^{(r)}\Big) - \textsc{Tr}\Big(\mathbf{X}^{(r)\top}\widetilde{\mathbf{B}}\mathbf{Y}^{(r)}\Big) \leq 2\cdot\epsilon\cdot\sqrt{k}\cdot mn,$$

where $(\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}) \triangleq \max_{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}} \textsc{Tr}\Big(\mathbf{X}^\top\widetilde{\mathbf{B}}\mathbf{Y}\Big)$.

In turn, by Lemma 7.4.12 taking into account that $\|\mathbf{B} - \widetilde{\mathbf{B}}\|_2^2 \leq mn/r$ (see Cor. 6), on the original bi-adjacency matrix $\mathbf{B}$ the output pair satisfies

$$\begin{aligned}\textsc{Tr}\Big(\mathbf{X}_{k\star}{}^\top\mathbf{B}\mathbf{Y}_{k\star}\Big) - \textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{B}\mathbf{Y}^{(r)}\Big)& \\ \leq 2\epsilon\sqrt{k}\cdot mn + 2(r+1)^{-1/2}\cdot mn.&\end{aligned} \tag{7.7}$$

Here, $\mathbf{X}_{k\star}, \mathbf{Y}_{k\star}$ denotes the unknown optimal pair of cluster assignment matrices representing the optimal k-clustering $\mathcal{C}_{k\star}$, *i.e.*, the clustering that achieves the maximum number of agreements using at most k clusters. Note that $(\mathbf{X}_{k\star}, \mathbf{Y}_{k\star}) \triangleq \max_{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}} \textsc{Tr}\Big(\mathbf{X}^\top\mathbf{B}\mathbf{Y}\Big)$ for the constraint sets defined in (7.3). Similarly, let $\mathcal{C}_k^{(r)}$ be the clustering induced by the computed pair $\mathbf{X}^{(r)}$, $\mathbf{Y}^{(r)}$; this is also a clustering into (at most) k clusters. From (7.7) and Lemma 7.3.10, it immediately follows that

$$\textsc{Agree}(\mathcal{C}_{k\star}) - \textsc{Agree}\Big(\mathcal{C}_k^{(r)}\Big) \leq 2\cdot(\epsilon\sqrt{k} + \sqrt{r+1})\cdot mn. \tag{7.8}$$

Eq. (7.8) establishes the guarantee of Alg. 7.16 as a function of its accuracy parameters r and ϵ. Substituting those parameters by the values assigned by

Alg. 7.17, we obtain

$$\textsc{Agree}(\mathcal{C}_{k\star}) - \textsc{Agree}\Big(\mathcal{C}_k^{(r)}\Big) \leq 2^{-1} \cdot \delta \cdot mn. \tag{7.9}$$

Fact 1. *For any* BCC *instance (or* k*-*BCC *instance with* $k \geq 2$*), the optimal clustering achieves at least* $mn/2$ *agreements.*

Proof. If more than half of the edges are labeled $+$, a single cluster containing all vertices achieves at least $nm/2$ agreements. Otherwise, two clusters corresponding to U and V achieve the same result. □

In other words, Fact 1 states that

$$\textsc{Agree}(\mathcal{C}_{k\star}) \geq mn/2, \quad \forall k \geq 2. \tag{7.10}$$

Combining (7.10) with (7.9), we obtain

$$\textsc{Agree}\Big(\mathcal{C}_k^{(r)}\Big) \geq (1-\delta) \cdot \textsc{Agree}(\mathcal{C}_{k\star}),$$

which is the desired result. Finally, the computational complexity is obtained substituting the appropriate values in that of Alg. 7.14 as in Lemma 7.4.11, and the time complexity of subroutine Alg. 7.15 given in Lemma 7.5.13. This completes the proof of Theorem 7.12.

7.6 An Efficient PTAS for BCC

We provide an efficient polynomial time approximation scheme (EPTAS) for BCC/MaxAgree, *i.e.*, the unconstrained version with no upper

bound on the number of output clusters. We rely on the following key observation: *any constant factor approximation of the* MaxAgree *objective, can be achieved by constant number of clusters.*[3] Hence, the desired approximation algorithm for BCC can be obtained by invoking the k-BCC algorithm under the appropriate configuration.

Recall that in the unconstrained BCC setting, the number of output clusters is an optimization variable; the optimal clustering may comprise any number of clusters, from a single cluster containing all vertices of the graph, up to $|U|+|V|$ singleton clusters. In principle, one could solve MaxAgree$[k]$ for all possible values of the parameter k to identify the best clustering, but this approach is computationally intractable.

On the contrary, if one is willing to settle for an approximately optimal solution, then a constant number of clusters may suffice. More formally,

Lemma 7.6.14. *For any* BCC *instance, and* $0 < \epsilon \leq 1$*, there exists a clustering* $\mathcal{C}$ *with at most* $k = 2 \cdot \epsilon^{-1} + 2$ *clusters such that* $\textsc{Agree}(\mathcal{C}) \geq \textsc{Agree}(\mathcal{C}_\star) - \epsilon \cdot nm$*, where* $\mathcal{C}_\star$ *denotes the optimal clustering.*

We defer the proof of Lemma 7.6.14 to the end of the section.

In conjunction with Fact 1, Lemma 7.6.14 suggests that in order to obtain a constant factor approximation for the unconstrained BCC/MaxAgree problem, for any constant arbitrarily close to 1, it suffices to solve a k-BCC

[3] This is an extension of a similar observation for CC/MaxAgree established in [25].

Algorithm 7.18 (E)PTAS for BCC/MaxAgree

input G – Bipartite graph $G = (U, V, E)$.
δ – Accuracy parameter in $(0, 1)$.
output $\widetilde{\mathcal{C}}$ – A clustering of $U \cup V$ into (at most) $2^3 \cdot \delta^{-1}$ clusters. See Thm. 7.13.
1: Set up parameters k, ϵ and r as a function δ:
$k \leftarrow 2^3 \cdot \delta^{-1}$, $\epsilon \leftarrow 2^{-6} \cdot \delta^2$, $r \leftarrow 2^8 \cdot \delta^{-2} - 1$.
2: Return output of Alg. 7.16 for input (G, k, r, ϵ).

instance for a sufficiently large –but constant– number of clusters k. In particular, to obtain an $(1 - \delta)$-factor approximation for any constant $\delta \in (0, 1)$, it suffices to solve the k-BCC problem with an upper bound $k = O(\delta^{-1})$ on the number of clusters.

Alg. 7.18 outlines the approximation scheme for BCC. For a given accuracy parameter δ, it invokes Alg. 7.16 under an appropriate configuration of the parameters k, r and ϵ, yielding the following guarantees:

Theorem 7.13. (PTAS for BCC.) *For any instance $G = (U, V, E)$ of the* BCC *problem and for any desired accuracy parameter $\delta \in (0, 1)$, Algorithm 7.18 computes a clustering $\widetilde{\mathcal{C}}$ of $U \cup V$ into (at most) $2^3 \cdot \delta^{-1}$ clusters, such that*

$$\textsc{Agree}\left(\widetilde{\mathcal{C}}\right) \geq (1 - \delta) \cdot \textsc{Agree}(\mathcal{C}_\star),$$

in time $2^{O(\delta^{-3} \cdot \log \delta^{-3})} \cdot \delta^{-2} \cdot (m + n) + \mathrm{T}_{\mathsf{SVD}}(\delta^{-2})$. Here, $\mathcal{C}_\star$ denotes an optimal clustering (with no constraint on the number of clusters).

Proof. The proof of Theorem 7.13 follows from the guarantees of Alg. 7.16 in (7.8) substituting the values of the parameters k, r and ϵ with the values

specified by Alg 7.18. The core k-BCC Alg. 7.16 returns a clustering $\widetilde{\mathcal{C}}$ with at most $k' = 2^3 \cdot \delta^{-1}$ clusters that satisfies

$$\textsc{Agree}\left(\widetilde{\mathcal{C}}\right) \geq \textsc{Agree}(\mathcal{C}_{k'\star}) - \delta/4 \cdot mn, \tag{7.11}$$

where $\mathcal{C}_{k'\star}$ is the best among the clusterings using at most k' clusters. Also, for $k = k'$, Lemma 7.6.14 implies that $\textsc{Agree}(\mathcal{C}_{k'\star}) \geq \textsc{Agree}(\mathcal{C}_\star) - \delta/4 \cdot nm$, where $\mathcal{C}_\star$ is the optimal clustering without any constraint on the number of output clusters. Hence,

$$\textsc{Agree}\left(\widetilde{\mathcal{C}}\right) \geq \textsc{Agree}(\mathcal{C}_\star) - \delta/2 \cdot mn. \tag{7.12}$$

Continuing from (7.12), and taking into account Fact 1, we conclude that

$$\textsc{Agree}\left(\widetilde{\mathcal{C}}\right) \geq (1-\delta) \cdot \textsc{Agree}(\mathcal{C}_\star), \tag{7.13}$$

which is the quarantee of Thm. 7.13.

Finally, the computational complexity of Alg. 7.18 follows from that of Alg. 7.16 substituting the parameter values in Lemma 7.4.11, and taking into account the running time $\mathrm{T}_{\mathcal{X}} = O(\delta^{-2} \cdot m)$ of Alg. 7.15 used as subroutine $\mathrm{P}_{\mathcal{X}}(\cdot)$ and similarly $\mathrm{T}_{\mathcal{Y}} = O(\delta^{-2} \cdot n)$ for $\mathrm{P}_{\mathcal{Y}}(\cdot)$. This concludes the proof of Theorem 7.13. □

For any desired constant accuracy $\delta \in (0, 1)$, Alg. 7.18 outputs a clustering that achieves a number of agreements within a $(1-\delta)$-factor from the optimal, in time that grows exponentially in δ^{-1}, but linearly in the size mn of the input. In other words, Alg. 7.18 is an EPTAS for BCC/MaxAgree.

It remains to prove Lemma 7.6.14, which stated that a constant number of clusters suffices to obtain an approximately optimal solution.

7.6.1 Proof of Lemma 7.6.14

It suffices to consider $\epsilon > 2/(m+n-2)$ since any meaningful clustering has at most $m+n$ clusters and the lemma holds trivially otherwise. Without loss of generality, we focus on clusterings whose all but at most two clusters contain vertices from both U and V; one of the two remaining clusters can contain vertices only from U and the other only from V. To verify that, consider an arbitrary clustering $\mathcal{C}$ and let $\mathcal{C}'$ be the clustering obtained by merging all clusters of $\mathcal{C}$ containing only vertices of U into a single cluster, and those containing only vertices of V into another. The number of agreements is not affected, *i.e.*, $\textsc{Agree}(\mathcal{C}') = \textsc{Agree}(\mathcal{C})$.

We show that a clustering $\mathcal{C}$ with the properties described in the lemma exists by appropriately modifying $\mathcal{C}_\star$. Let $\mathcal{S}_U \subseteq \mathcal{C}_\star$ be the set of clusters that contain at most $\epsilon m/2$ vertices of U, and $\mathcal{S}_V \subseteq \mathcal{C}_\star$ those containing at most $\epsilon n/2$ vertices of V. Note that that $\mathcal{S}_U$ and $\mathcal{S}_V$ may not be disjoint. Finally, let $\mathcal{B}$ be the set of vertices contained in clusters of $\mathcal{S}_U \cup \mathcal{S}_V$. We construct $\mathcal{C}$ as follows. Clusters of $\mathcal{C}_\star$ not in $\mathcal{S}_U \cup \mathcal{S}_V$ are left intact. Each such cluster has size at least $\epsilon(n+m)/2$. Further, all vertices in $\mathcal{B}$ are rearranged into two clusters: one for the vertices in $\mathcal{B} \cap U$ and one those in $\mathcal{B} \cap V$.

The above rearrangement can reduce the number of agreements by at most ϵnm. To verify that, consider a cluster C in $\mathcal{S}_U$; the cluster contains

at most $\epsilon m/2$ vertices of U. Let $t \triangleq |V \cap C|$. Splitting C into two smaller clusters $C \cap U$ and $C \cap V$ can reduce the number of agreements by $\frac{1}{2}\epsilon m t$, *i.e.*, the total number of edges in C. Note that agreements on $-$ edges are not affected by splitting a cluster. Performing the same operation on all clusters in $\mathcal{S}_U$ incurs a total reduction

$$\tfrac{1}{2}\epsilon m \sum_{C \in \mathcal{S}_U} |V \cap C| \leq \tfrac{1}{2}\epsilon m |V| = \tfrac{1}{2}\epsilon m n.$$

Repeating on $\mathcal{S}_V$ incurs an additional cost of at most $\epsilon mn/2$ agreements, while merging all clusters containing only vertices from U (similarly for V) into a single cluster does not reduce the agreements. We conclude that $\textsc{Agree}(\mathcal{C}) \geq \textsc{Agree}(\mathcal{C}_\star) - \epsilon nm$.

Finally, by construction all but at most two clusters in $\mathcal{C}$ contain at least $\frac{1}{2}\epsilon(n+m)$. In turn,

$$n + m = \sum_{C \in \mathcal{C}} |C| \geq (|\mathcal{C}| - 2) \cdot \frac{1}{2}\epsilon(n+m),$$

from which the desired result of Lemma 7.6.14 follows. □

7.7 Experiments

We evaluate our algorithm on synthetic and real data and compare with PivotBiCluster [2] and the SDP-based approach of [131][4] Although [9, 36, 53] provide algorithms applicable to BCC, those rely on LP or SDP formulations with a high number of constraints which imposes limitations in practice.

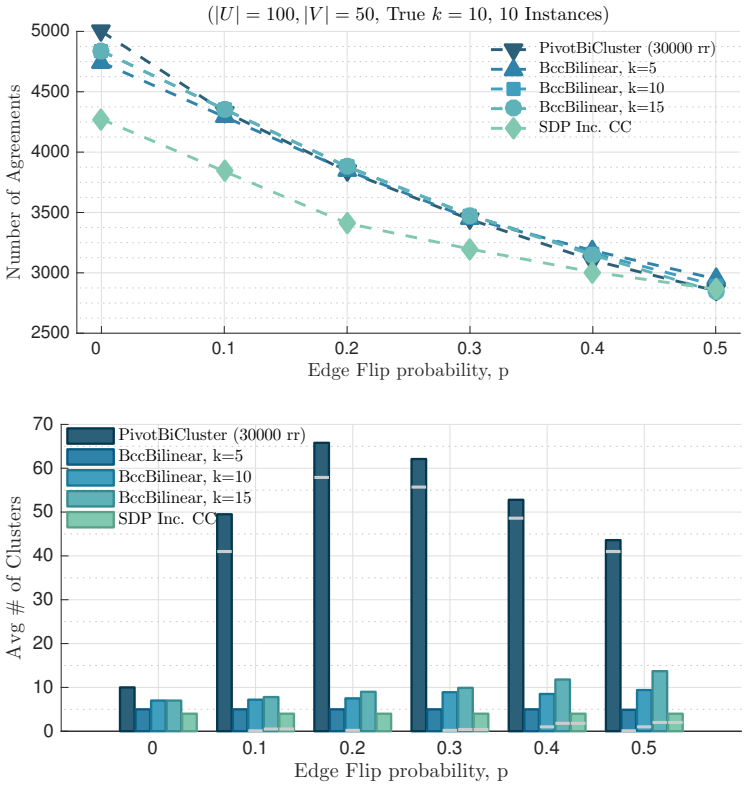
[4] All algorithms are prototypically implemented in Matlab. [131] was implemented using CVX.

We run our core k-BCC algorithm (Alg. 7.16) to obtain a k-clustering. Disregarding the theoretical guarantees, we apply a threshold on the execution time which can only hurt the performance; equivalently, the iterative procedure of Alg. 7.14 is executed for an arbitrary subset of the points in the ϵ-net (Alg. 7.14, step 3).

7.7.1 Synthetic Data

We generate synthetic BCC instances as follows. We arbitrarily set $m = 100$, $n = 50$, and number of clusters $k' = 5$, and construct a complete bipartite graph $G = (U, V, E)$ with $|U| = m$, $|V| = n$ and assign binary ± 1 labels to the edges according to a random k'-clustering of the vertices. Then, we modify the sign of each label independently with probability p. We consider multiple values of p in the range $[0, 0.5]$. For each value, we generate 10 random instances as described above and compute a vertex clustering using all algorithms.

PivotBiCluster returns a clustering of arbitrary size; no parameter k is specified and the algorithm determines the number of clusters in an attempt to maximize the number of agreements. The majority of the output clusters are singletons which can be merged into two clusters (see Subsec. 7.6). The algorithm is substantially faster that the other approaches on examples of this scale. Hence, we run PivotBiCluster with $3 \cdot 10^4$ random restarts on each instance and depict best results to allow for comparable running times. Contrary to PivotBiCluster, for our algorithm, which is reffered to as BccBilinear, we



p	PivotBi-Cluster [2]	SDP Inc CC [131]	Our BccBlinear Alg. $k = 5$	$k = 10$	$k = 15$
0.0	124	198	69	73	78
0.1	83	329	70	74	79
0.2	66	257	70	75	79
0.3	54	258	71	75	80
0.4	45	266	71	75	80
0.5	39	314	71	75	80

Average Runtimes/Instance (in seconds).

Figure 7.1: Maximizing agreements on synthetic instances of BCC. We generate an $m \times n$ bipartite graph with edge weights ± 1 according to a random vertex k-clustering and subsequently flip the sign of each edge with probability p. We plot the average number of agreements achieved by each method over 10 random instances for each p value. The bar plot depicts the average number of output clusters for each scheme/p-value pair. Within each bar, a horizontal line marks the number of singleton clusters.

need to specify the target number k of clusters. We run it for $k = 5, 10$ and 15 and arbitrarily set the parameter $r = 5$ and terminate our algorithm after 10^4 random samples/rounds. Finally, the SDP-based approach returns 4 clusters by construction, and is configured to output best results among 100 random pivoting steps.

Fig. 7.1 depicts the number of agreements achieved by each algorithm for each value of p, the number of output clusters, and the execution times. All numbers are averages over multiple random instances.

We note that in some cases and contrary to intuition, our algorithm performs better for lower values of k, the target number of clusters. We attribute this phenomenon to the fact that for higher values of k, we should typically use higher approximation rank and consider larger number of samples.

7.7.2 Real Data (MovieLens Dataset)

The MovieLens datasets [65] are sets of movie ratings: each of m users assigns scores in $\{1, \ldots, 5\}$ to a small subset of the n movies. Table 7.2 lists the dimensions of the datasets. From each dataset, we generate an instance of the (incomplete) BCC problem as follows: we construct the bipartite graph on the user-movie pairs using the ratings as edge weights, compute the average weight and finally set weights higher than the average to $+1$ and the others to -1.

We run our algorithm to obtain a clustering of the vertices. For a reference, we compare to PivotBiCluster with 50 random restarts. Note, how-

Dataset	m (Users)	n (Movies)	Ratings
MovieLens100K	1000	1700	10^5
MovieLens1M	6000	4000	10^6
MovieLens10M	72000	10000	10^7

Table 7.2: Summary of datasets of the MovieLens collection [65].

ever, that PivotBiCluster is not designed for the incomplete BCC problem; to apply the algorithm, we effectively treat missing edges as edges of negative weight. Finally, the SDP approach of [131], albeit suitable for the incomplete CC problem, does not scale to this size of input. Table 7.3 lists the number of agreements achieved by each method on each one of the three datasets and the corresponding execution times.

7.8 Conclusions

We presented the first algorithm with provable approximation guarantees for k-BCC/MaxAgree. Our approach relied on formulating k-BCC as a

MovieLens	100K	1M	10M
PivotBiCluster	46134	429277	5008577
	(27.95 sec)	(651.13 sec)	($1.5 \cdot 10^5$ sec)
BccBilinear	68141	694366	6857509
	(6.65 sec)	(19.50 sec)	($1.2 \cdot 10^3$ sec)

Table 7.3: Number of agreements achieved by the two algorithms on incomplete k-BCC instances obtained from the MovieLens datasets [65]. For PivotBiCluster we present best results over 50 random restarts. Our algorithm was arbitrarily configured with $r = 4$, $k = 10$ and a limit of 10^4 samples (iterations). We also note in parenteses the runtimes in seconds.

constrained bilinear maximization over the sets of cluster assignment matrices and developing a simple framework to approximately solve that combinatorial optimization.

In the unconstrained BCC setting, with no bound on the number of output clusters, we showed that any constant multiplicative factor approximation for the MaxAgree objective can be achieved using a constant number of clusters. In turn, under the appropriate configuration, our k-BCC algorithm yields an Efficient PTAS for BCC/MaxAgree.

Appendices

Appendix A

Appendix for Chapter 2

A.1 NP-Hardness of Nonnegative PCA

We show that Nonnegative PCA (and in turn Nonnegative Sparse PCA) is NP-hard via a reduction from the problem of checking whether a real symmetric matrix is *copositive*. By definition,

$$\mathbf{M} \text{ is copositive } \Leftrightarrow \mathbf{M} \in \mathbb{S}^n : \mathbf{x}^\top \mathbf{M}\mathbf{x} \geq 0, \ \forall \mathbf{x} \geq 0.$$

Note that the set of copositive matrices forms a convex cone contained in the cone of Positive Semidefinite matrices.

Checking whether a matrix is copositive is a co-NP Complete decision problem [112]. Any vector $\mathbf{x}$ for which $\mathbf{x}^\top \mathbf{M}\mathbf{x} < 0$ serves as a certificate to verify in polynomial time that $\mathbf{M}$ is *not* copositive. In order to check whether a matrix $\mathbf{M}$ is copositive, it suffices to minimize the quadratic $\mathbf{x}^\top \mathbf{M}\mathbf{x}$ over all $\mathbf{x} \geq \mathbf{0}$; $\mathbf{M}$ is not copositive if and only if the minimum value is negative. Since scaling $\mathbf{x}$ does not affect the sign of the quadratic, it suffices to check the sign of the minimum value of the quadratic over the unit ℓ_2-norm vectors $\mathbf{x}$, *i.e.*.,

$$q = \min_{\substack{\mathbf{x} \geq \mathbf{0} \\ \|\mathbf{x}\|=1}} \mathbf{x}^\top \mathbf{M}\, \mathbf{x}.$$

Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $\mathbf{M}$ in decreasing order. The matrix $\overline{\mathbf{M}} = \lambda_1 \mathbf{I} - \mathbf{M}$ is positive semidefinite: its eigenvalues are $\lambda_1 - \lambda_i \geq 0$, for $1 \leq i \leq n$. Moreover, $\mathbf{x}^\top \overline{\mathbf{M}} \mathbf{x} = \lambda_1 - \mathbf{x}^\top \mathbf{M} \mathbf{x}$, $\forall \mathbf{x} : \|\mathbf{x}\|_2 = 1$. Hence, to check whether $\mathbf{M}$ is copositive, it suffices to solve

$$\max_{\substack{\mathbf{x} \geq 0 \\ \|\mathbf{x}\| = 1}} \mathbf{x}^\top \overline{\mathbf{M}} \mathbf{x},$$

on the PSD matrix $\overline{\mathbf{M}}$. The maximum value of the latter is greater than λ_1, if and only if $q < 0$. Noting that the last optimization problem coincides with Nonnegative PCA implies the desired hardness result.

A.2 Approximation Guarantees

In this section, we develop a series of Lemmata that establish the approximation guarantees of Theorem 2.1. First, recall that

$$\mathrm{OPT} = \max_{\mathbf{x} \in \mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A} \mathbf{x}$$

corresponds to the optimal value of the quadratic objective function with argument $\mathbf{A}$, and let $\mathbf{x}_\star$ be the optimal solution, *i.e.*, the nonnegative, s-sparse, unit norm vector achieving value OPT. Similarly, OPT_r denotes the optimal value of the quadratic with argument $\mathbf{A}_r$, the best rank-r approximation of $\mathbf{A}$, that is

$$\mathrm{OPT}_r = \max_{\mathbf{x} \in \mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r \mathbf{x},$$

and $\mathbf{x}_r$ is the corresponding optimal solution. Alg. 2.1 with input $\mathbf{A}$ and accuracy parameter r computes and outputs $\mathbf{x}_r$ as a surrogate for the desired

vector $\mathbf{x}_\star$. We show that

$$\mathbf{x}_r{}^\top \mathbf{A}\mathbf{x}_r \geq \rho_r \cdot \mathbf{x}_\star{}^\top \mathbf{A}\mathbf{x}_\star,$$

where

$$\rho_r \geq \max\left\{\frac{s}{2n},\ \frac{1}{1+2\frac{n}{s}\lambda_{r+1}/\lambda_1}\right\}.$$

Lemma A.2.15. *Let* $\mathbf{x}_r$ *denote the nonnegative* s*-sparse principal component of* $\mathbf{A}_r$*, i.e.,* $\mathbf{x}_r = \arg\max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r \mathbf{x}$*, achieving value* $\mathrm{OPT}_r = \mathbf{x}_r{}^\top \mathbf{A}_r \mathbf{x}_r$*. Then,*

$$\mathbf{x}_r{}^\top \mathbf{A}\mathbf{x}_r \ \geq\ \mathrm{OPT}_r.$$

Proof. The lemma is a consequence of the fact that $\mathbf{A}$ is a positive semidefinite matrix:

$$\begin{aligned}
\mathbf{x}_r{}^\top \mathbf{A}\mathbf{x}_r &= \mathbf{x}_r{}^\top \left(\sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top\right)\mathbf{x}_r \\
&= \mathbf{x}_r{}^\top \mathbf{A}_r \mathbf{x}_r + \sum_{i=d+1}^{n} \left|\sqrt{\lambda_i}\mathbf{q}_i^\top \mathbf{x}_r\right|^2 \\
&\geq \mathrm{OPT}_r, \quad \forall r \in [n],
\end{aligned}$$

which is the desired result. □

Lemma A.2.16. *The optimal value* OPT_r *of the rank-*r *nonnegative* s*-sparse PCA problem satisfies*

$$\mathrm{OPT} - \lambda_{r+1} \ \leq\ \mathrm{OPT}_r \ \leq\ \mathrm{OPT}.$$

Proof. The upper bound is due to the fact that $\mathbf{A}$ is a positive semidefinite matrix:

$$\begin{aligned}
\mathrm{OPT} &= \max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}\mathbf{x} \\
&\geq {\mathbf{x}_r}^\top \mathbf{A}\mathbf{x}_r \\
&= {\mathbf{x}_r}^\top \mathbf{A}_r\mathbf{x}_r + {\mathbf{x}_r}^\top(\mathbf{A}-\mathbf{A}_r)\mathbf{x}_r \\
&= \mathrm{OPT}_r + \sum_{i=d+1}^{n} \lambda_i |{\mathbf{q}_i}^\top \mathbf{x}_r|^2 \\
&\geq \mathrm{OPT}_r.
\end{aligned}$$

For the lower bound,

$$\begin{aligned}
\mathrm{OPT} &= \max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}\mathbf{x} \\
&= \max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \left(\sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top\right)\mathbf{x} \\
&\leq \max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r\mathbf{x} + \max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \sum_{i=d+1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top \mathbf{x} \\
&\leq \mathrm{OPT}_r + \max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \sum_{i=d+1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top \mathbf{x} \\
&\leq \mathrm{OPT}_r + \max_{\|\mathbf{x}\|=1} \mathbf{x}^\top \sum_{i=d+1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top \mathbf{x} \\
&= \mathrm{OPT}_r + \lambda_{r+1},
\end{aligned}$$

which completes the proof. □

Lemma A.2.17.

$$\frac{\mathrm{OPT}_r}{\mathrm{OPT}} \;\geq\; \max\left\{\frac{\mathrm{OPT}_r}{\lambda_1},\; \frac{1}{1+\frac{\lambda_{r+1}}{\mathrm{OPT}_r}}\right\},\; \forall r \in [n].$$

Proof. It suffices to show that $\mathrm{OPT}_r/\mathrm{OPT}$ is lower bounded by both quantities on the right-hand side. The first lower bound follows trivially from the fact that

$$\mathrm{OPT} = \max_{\mathbf{x}\in\mathbb{S}^n_s} \mathbf{x}^\top \mathbf{A}\mathbf{x} \;\leq\; \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A}\mathbf{x} = \lambda_1.$$

For the second lower bound, note that by Lemma A.2.16, $\mathrm{OPT} \leq \mathrm{OPT}_r + \lambda_{r+1}$, which in turn implies

$$\frac{\mathrm{OPT}_r}{\mathrm{OPT}} \;\geq\; \frac{1}{1+\lambda_{r+1}/\mathrm{OPT}_r}.$$

□

Lemma A.2.18. *The optimal value* OPT_1 *of the nonnegative, s-sparse PCA problem* $\max_{\mathbf{x}\in\mathbb{S}^n_s} \mathbf{x}^\top \mathbf{A}_1\mathbf{x}$ *on the rank-*1 *matrix* $\mathbf{A}_1$ *satisfies*

$$\mathrm{OPT}_1 \;\geq\; \frac{1}{2}\frac{s}{n}\lambda_1.$$

Proof. Let $(\mathbf{v})^+_s$ denote the vector obtained by setting to zero all but the (at most) s largest nonnegative entries of $\mathbf{v}$. By definition,

$$\begin{aligned}
\mathrm{OPT}_1 &= \max_{\mathbf{x}\in\mathbb{S}^n_s} \mathbf{x}^\top \mathbf{A}_1\mathbf{x} \\
&= \lambda_1 \cdot \max_{\mathbf{x}\in\mathbb{S}^n_s} \mathbf{x}^\top \mathbf{q}_1\mathbf{q}_1^\top \mathbf{x} \\
&= \lambda_1 \cdot \max_{\mathbf{x}\in\mathbb{S}^n_s} \left|\mathbf{q}_1^\top \mathbf{x}\right|^2 \\
&= \lambda_1 \cdot \max\left\{ \left|\frac{\mathbf{q}_1^\top (\mathbf{q}_1)^+_s}{\|(\mathbf{q}_1)^+_s\|}\right|^2, \left|\frac{\mathbf{q}_1^\top (-\mathbf{q}_1)^+_s}{\|(-\mathbf{q}_1)^+_s\|}\right|^2 \right\} \\
&= \lambda_1 \cdot \max\Big\{ \|(\mathbf{q}_1)^+_s\|^2,\; \|(-\mathbf{q}_1)^+_s\|^2 \Big\}.
\end{aligned}$$

It holds that

$$\|(\mathbf{q}_1)_s^+\|^2 \;\geq\; \frac{s}{n}\|(\mathbf{q}_1)_n^+\|^2. \tag{A.1}$$

To verify that, let $\mathcal{I}_s$ be the support of $(\mathbf{q}_1)_s^+$ and $\mathcal{I}_n$ the support of $(\mathbf{q}_1)_n^+$. Clearly, $\mathcal{I}_s \subseteq \mathcal{I}_n$. Let u be the value of the smallest non-zero entry in $(\mathbf{q}_1)_s^+$. This implies that

$$\|(\mathbf{q}_1)_s^+\|^2 = \sum_{i\in\mathcal{I}_s}([\mathbf{q}_1]_i)^2 \;\geq\; k\cdot u^2.$$

Further,

$$\begin{aligned}\|(\mathbf{q}_1)_n^+\|^2 &= \sum_{i\in\mathcal{I}_s}([\mathbf{q}_1]_i)^2 + \sum_{i\in\mathcal{I}_n\setminus\mathcal{I}_n}([\mathbf{q}_1]_i)^2\\ &= \|(\mathbf{q}_1)_s^+\|^2 + \sum_{i\in\mathcal{I}_n\setminus\mathcal{I}_s}([\mathbf{q}_1]_i)^2\\ &\leq \|(\mathbf{q}_1)_s^+\|^2 + (n-s)\cdot u^2,\end{aligned}$$

From the two inequalities, it follows that

$$\frac{\|(\mathbf{q}_1)_n^+\|^2}{\|(\mathbf{q}_1)_s^+\|^2} \leq 1 + \frac{(n-k)\cdot u^2}{k\cdot u^2} \leq \frac{n}{s},$$

which in turn implies (A.1). By the same argument,

$$\|(-\mathbf{q}_1)_s^+\|^2 \geq \frac{s}{n}\|(-\mathbf{q}_1)_n^+\|^2. \tag{A.2}$$

Finally, noting that

$$1 = \|\mathbf{q}_1\|^2 = \|(\mathbf{q}_1)_n^+\|^2 + \|(-\mathbf{q}_1)_n^+\|^2,$$

and combining with (A.1) and (A.2), we obtain

$$\begin{aligned}\mathrm{OPT}_1 &\geq \lambda_1 \cdot \max\left\{\frac{s}{n}\|(\mathbf{q}_1)_n^+\|^2,\ \frac{s}{n}\|(-\mathbf{q}_1)_n^+\|^2\right\}\\ &= \frac{s}{n}\lambda_1 \cdot \max\left\{\|(\mathbf{q}_1)_n^+\|^2,\ 1-\|(\mathbf{q}_1)_n^+\|^2\right\}\\ &\geq \frac{1}{2}\frac{s}{n}\lambda_1,\end{aligned}$$

which completes the proof. □

Lemma A.2.19.

$$\frac{\mathrm{OPT}_r}{\mathrm{OPT}} \ \geq\ \max\left\{\frac{s}{2n},\ \frac{1}{1+2\frac{n}{s}\lambda_{r+1}/\lambda_1}\right\}.$$

Proof. $\mathbf{A}_1$ is the best rank-1 approximation of $\mathbf{A}_r$. By Lemmata A.2.16 and A.2.18, we have $\mathrm{OPT}_r \geq \mathrm{OPT}_1 \geq \frac{1}{2}\frac{s}{n}\lambda_1$, for all $r \geq 1$. The desired result follows from Lemma A.2.17 and the previous lower bound on OPT_r. □

Proof of Theorem 2.1. By Lemma A.2.15, ${\mathbf{x}_r}^\top \mathbf{A}\mathbf{x}_r \geq \mathrm{OPT}_r$. Dividing both sides by $\mathrm{OPT} = {\mathbf{x}_\star}^\top \mathbf{A}\mathbf{x}_\star$, we obtain

$${\mathbf{x}_r}^\top \mathbf{A}\mathbf{x}_r \ \geq\ \rho_r \cdot {\mathbf{x}_\star}^\top \mathbf{A}\mathbf{x}_\star,$$

where $\rho_r = \mathrm{OPT}_r/\mathrm{OPT}$. The lower bound on ρ_r given in Theorem 2.1 follows from Lemma A.2.19. The computational complexity of Alg. 2.1 follows from the detailed description of the algorithm and is analyzed separately. ■

A.2.1 Approximation Guarantees - Special Cases

Corollary 1. *If the eigenvalues of* $\mathbf{A}$ *follow a decay law* $\lambda_i \leq \lambda_1 \cdot f(i)$ *for any vanishing function* $f(i)$*, i.e., for* $f(i) \to 0$ *as* $i \to \infty$*, then for* $s = c \cdot n$*, where* c

is constant $0 < c \leq 1$ *(linear sparsity regime), Alg. 2.1 yields a polynomial time approximation scheme (PTAS). That is, for any constant* ϵ*, we can choose a constant accuracy parameter* r *and obtain a solution* $\mathbf{x}_r$ *such that*

$$\mathbf{x}_r{}^\top \mathbf{A}\mathbf{x}_r \;\geq\; (1-\epsilon)\cdot \mathrm{OPT},$$

in time polynomial in n *and* s*, but not in* $1/\epsilon$*.*

Proof. By assumption, $\lambda_i \leq \lambda_1 \cdot f(i)$ for some function $f(i)$ such that $f(i) \to 0$ as $i \to \infty$. For any constants c and ϵ, there must hence exists a finite i such that

$$f(i) \leq \frac{c}{2} \cdot \frac{\epsilon}{1-\epsilon}.$$

Set r equal to the smallest i for which the above holds: r will be some function $g(\epsilon)$ that depends on $f(\cdot)$. By Theorem 2.1, and under the assumption of the corollary we have

$$\begin{aligned} \rho_r \;&\geq\; \frac{1}{1+2\frac{n}{s}\lambda_{r+1}/\lambda_1} \;\geq\; \frac{1}{1+2f(r)/c} \\ &\geq\; \frac{1}{1+\epsilon/(1-\epsilon)} \;\geq\; (1-\epsilon), \end{aligned}$$

which implies that for any ϵ, the output $\mathbf{x}_r$ will be within factor $1-\epsilon$ from the optimal. Alg. 2.1 runs in time $O(n^{2r}) = O(n^{g(\epsilon)})$, which completes the proof. □

A.3 The Spannogram Algorithm for General Rank

In this section, we provide a detailed description of the Spannogram algorithm for the construction of the collection $\mathcal{S}_r$ of candidate support sets in the case of arbitrary r.

For completeness, we first give a proof for Proposition 2.4.1, which states that the support of $\mathbf{x}_1$, the optimal solution of the rank-1 nonnegative sparse PCA problem

$$\max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_1 \mathbf{x} = \max_{\mathbf{x}\in\mathbb{S}_s^n} \left(\mathbf{v}^\top \mathbf{x}\right)^2,$$

where $\mathbf{A}_1 = \mathbf{v}\mathbf{v}^\top$, coincides with one of the two sets in the collection $\mathcal{S}_1 = \{\mathcal{I}_s^+(\mathbf{v}),\ \mathcal{I}_s^+(-\mathbf{v})\}$.

A.3.1 Proof of Proposition 2.4.1

Let $\mathbf{x}_\star = \arg\max_{\mathbf{x}\in\mathbb{S}_s^n} \left(\mathbf{v}^\top \mathbf{x}\right)^2$. First, assume that $\mathbf{v}^\top \mathbf{x}_\star \geq 0$. We will show that $\mathrm{supp}(\mathbf{x}_1^\star) \subseteq \mathcal{I}_s^+(\mathbf{v})$.

Assume, for the sake of contradiction, that the support of $\mathbf{x}_\star$ does *not* coincide with $\mathcal{I}_s^+(\mathbf{v})$. This implies that there exists an index $j \notin \mathcal{I}_s^+(\mathbf{v})$ such that $j \in \mathrm{supp}(\mathbf{x}_\star)$, *i.e.*, $[\mathbf{x}_\star]_j > 0$. By the definition of $\mathcal{I}_s^+(\mathbf{v})$, $j \notin \mathcal{I}_s^+(\mathbf{v})$ implies that either *i)* $v_j < 0$, or *ii)* there exist at least s nonnegative entries in $\mathbf{v}$ larger than v_j.

In the first case, consider a vector $\widehat{\mathbf{x}}$ that is equal to $\mathbf{x}_\star$ in all entries except the jth entry which is set to zero in $\widehat{\mathbf{x}}$. Then, $\mathbf{y} = \widehat{\mathbf{x}}/\|\widehat{\mathbf{x}}\|_2$, is s-sparse

(at most $s-1$ nonzero entries), nonnegative and unit length. It should not be hard to see that since $v_j < 0$, $\mathbf{v}^\top \mathbf{y} \geq \mathbf{v}^\top \mathbf{x}_\star$, contradicting the optimality of $\mathbf{x}_\star$.

In the second case, let l be the index of one of the s largest nonnegative entries in $\mathbf{v}$ such that $[\mathbf{x}_\star]_l = 0$. Such an entry exists, because otherwise $\mathbf{x}_\star$ would have more than s nonzero entries. Construct a nonnegative, s-sparse, unit length vector $\mathbf{y}$ by swapping the values in the jth and l-th entries of $\mathbf{x}_\star$. Then, $\mathbf{v}^\top \mathbf{y} \geq \mathbf{v}^\top \mathbf{x}_\star$, contradicting the optimality of $\mathbf{x}_\star$.

We conclude that

$$\mathbf{v}^\top \mathbf{x}_\star \geq 0 \quad \Rightarrow \quad \operatorname{supp}(\mathbf{x}_\star) \subseteq \mathcal{I}_s^+(\mathbf{v}).$$

Similarly, if $\mathbf{v}^\top \mathbf{x}_\star < 0$, then $-\mathbf{v}^\top \mathbf{x}_\star > 0$, and

$$\mathbf{v}^\top \mathbf{x}_\star < 0 \quad \Rightarrow \quad \operatorname{supp}(\mathbf{x}_\star) \subseteq \mathcal{I}_s^+(-\mathbf{v}).$$

Since either $\mathbf{v}^\top \mathbf{x}_\star \geq 0$ or $\mathbf{v}^\top \mathbf{x}_\star < 0$ holds, we conclude that $\operatorname{supp}(\mathbf{x}_\star) \in \mathcal{S}_1 = \{\mathcal{I}_s^+(\mathbf{v}),\ \mathcal{I}_s^+(-\mathbf{v})\}$. ■

A.3.2 The general rank-r case

We generalize the developments of Section 2.5.1 to case of arbitrary constant r. More specifically, we will show that for any $\mathbf{V} \in \mathbb{R}^{n \times r}$, the collection

$$\mathcal{S}_r = \bigcup_{\mathbf{c} \in \mathbb{S}^r} \left\{ \mathcal{I}_s^+(\mathbf{V}\mathbf{c}) \right\}$$

contains at most $O(n^r)$ candidate support sets and can be constructed in $O(n^{r+1})$.

Hyperspherical Variables Let $\mathcal{R}(\mathbf{V})$ denote the range of $\mathbf{V} \in \mathbb{R}^{n\times r}$. Up to scaling, all vectors $\mathbf{v}$ in $\mathcal{R}(\mathbf{V})$, can be written as $\mathbf{v} = \mathbf{V}\mathbf{c}$ for some $\mathbf{c} \in \mathbb{R}^r$: $\|\mathbf{c}\| = 1$. We introduce $r-1$ variables $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_{r-1}] \in \Phi^{r-1} = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^{r-1}$, and set $\mathbf{c}$ to be the following function of $\boldsymbol{\phi}$:

$$\mathbf{c}(\boldsymbol{\phi}) = \begin{bmatrix} \sin(\phi_1) \\ \cos(\phi_1)\sin(\phi_2) \\ \vdots \\ \cos(\phi_1)\cos(\phi_2)\cdots\sin(\phi_{r-1}) \\ \cos(\phi_1)\cos(\phi_2)\cdots\cos(\phi_{r-1}) \end{bmatrix} \in \mathbb{R}^r. \tag{A.3}$$

In other words, $\phi_1, \ldots, \phi_{r-1}$ are the spherical coordinates of $\mathbf{c}(\boldsymbol{\phi})$. All unit vectors in $\mathbb{R}^r$ can be mapped to a spherical coordinate vector $\boldsymbol{\phi} \in (-\pi, \pi] \times \Phi^{r-2}$. Restricting variable ϕ_1 to Φ limits $\mathbf{c}(\boldsymbol{\phi})$ to half the r-dimensional unit sphere: for any unit norm vector $\mathbf{c}$, there exists $\boldsymbol{\phi} \in \Phi^{r-1}$ such that $\mathbf{c} = \mathbf{c}(\boldsymbol{\phi})$ or $\mathbf{c} = -\mathbf{c}(\boldsymbol{\phi})$.

Under (A.3), the vectors in $\mathcal{R}(\mathbf{V})$ can be described as a function of $\boldsymbol{\phi}$: $\mathcal{R}(\mathbf{V}) = \{\pm\mathbf{v}(\boldsymbol{\phi}) = \pm\mathbf{V}\mathbf{c}(\boldsymbol{\phi}), \boldsymbol{\phi} \in \Phi^{r-1}\}$. In turn, the set of indices of the s largest nonnegative entries of $\mathbf{v}(\boldsymbol{\phi})$ is itself a function of $\boldsymbol{\phi}$, and

$$\mathcal{S}_r = \bigcup_{\boldsymbol{\phi}\in\Phi^{r-1}} \left\{\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi})),\ \mathcal{I}_s^+(-\mathbf{v}(\boldsymbol{\phi}))\right\}.$$

Spannogram The ith entry of $\mathbf{v}(\boldsymbol{\phi})$ is

$$[\mathbf{v}(\boldsymbol{\phi})]_i = V_{i,1}\sin(\phi_1) + \cdots + V_{i,r}\prod_{l=1}^{r}\cos(\phi_{i_l}),$$

a continuous function of $\boldsymbol{\phi} \in \Phi^{r-1}$; a $(r-1)$-dimensional hypersurface in the r-dimensional space $\Phi^{r-1} \times \mathbb{R}$, for all $i \in [n]$. The collection of hypersur-

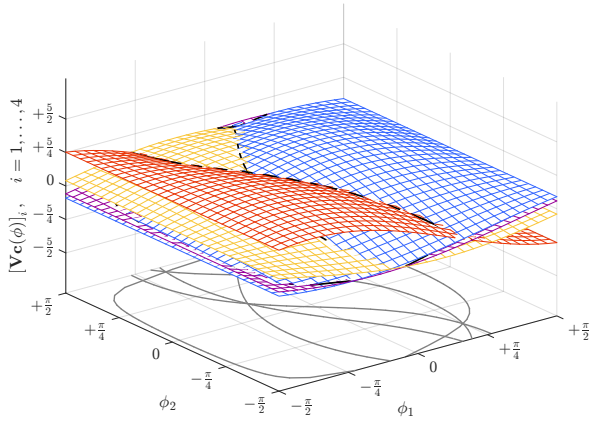


Figure A.1: Example of rank-3 spannogram. The figure depicts the spannogram of an arbitrary rank-3 matrix $\mathbf{V} \in \mathbb{R}^{4\times 3}$. Each surface is associated with (generated by) a row of $\mathbf{V}$. At every point $\boldsymbol{\phi} = [\phi_1, \phi_2]$, the surface values correspond to the entries of a vector in the range of $\mathbf{V}$ and vice versa.

faces constitutes the rank-r spannogram. As an example, Fig. A.1 depicts the spannogram of an arbitrary 4×3 ($r = 3$) matrix $\mathbf{V}$.

At any particular point $\boldsymbol{\phi} \in \Phi^{r-1}$, assuming that no two hypersurfaces intersect at $\boldsymbol{\phi}$, the set $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ can be readily determined: sort the entries of $\mathbf{v}(\boldsymbol{\phi})$ and pick the indices of the at most s largest nonnegative entries. Note, however, that constructing $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ does not require a complete sorting the entries of $\mathbf{v}(\boldsymbol{\phi})$: detecting the s^{th} order entry and the (at most) $s-1$ nonnegative entries larger than that can be done in $O(n)$.

The key observation of our algorithm is that, due to their continuity, the hypersurfaces will retain their sorting *around* $\boldsymbol{\phi}$ and hence, $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ tends to remain invariant. Moving away from $\boldsymbol{\phi}$, $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ can only change if

when either the sign of a hypersurface or its order relative to other hypersurfaces changes. In other words, $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ can only change at points $\boldsymbol{\phi} \in \Phi^{r-1}$ where i) two hypersurfaces intersect, or ii) a hypersurface crosses the zero-hypersurface. Henceforth, we will assume that $\mathbf{V}$ has $n+1$ rows, where the last row is the zero vector, $\mathbf{0}_r$, generating the zero-hypersurface. As a result, the points of interest lie in the intersection of subsets of the $n+1$ hypersurfaces in the spannogram of $\mathbf{V}$.

We have argued that in order to construct $\mathcal{S}_r$, it suffices to consider points corresponding to the intersection of pairs of hypersurfaces. For $r > 2$, pairwise hypersurface intersections no longer correspond to single points. In the sequel, however, we will show that the points of interest can be further reduced to a finite set of points.

Let us examine when the set $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ changes from the perspective of the ith hypersurface. That is, we ask what are the points in Φ^{r-1} where the ith index might join or leave the candidate support set $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$. We know it suffices to examine only those points in Φ^{r-1} at which the ith hypersurface intersects with another of the $n+1$ hypersurfaces. Let us focus on the intersection with the jth hypersurface, $j \in [n+1], j \neq i$. We define

$$\mathcal{H}(i,j) = \Big\{\mathbf{v}(\boldsymbol{\phi}) : [\mathbf{v}(\boldsymbol{\phi})]_i = [\mathbf{v}(\boldsymbol{\phi})]_j, \phi \in \Phi^{r-1}\Big\},$$

as the set of points lying in the intersection of hypersurfaces i and j. These

points form a $(r-2)$-dimensional hypersurface[1]. Further, let

$$\Phi(i,j) = \{\boldsymbol{\phi} : \mathbf{v}(\boldsymbol{\phi}) \in \mathcal{H}(i,j)\},$$

be the corresponding $\boldsymbol{\phi}$'s. By definition, at every $\boldsymbol{\phi} \in \Phi(i,j)$, hypersurfaces i and j have the same values, and in opposite directions over $\boldsymbol{\phi} \in \Phi(i,j)$ the relative order of the two hypersurfaces changes. However, not all of the points in $\Phi(i,j)$ are necessarily points of interest; it is not necessary that $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ changes at every $\boldsymbol{\phi} \in \Phi(i,j)$. We seek to restrict our attention to a smaller subset of points.

If at some $\boldsymbol{\phi} \in \Phi(i,j)$ the ith hypersurface is included or excluded from $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$, we ask what are those points where index i might leave or join the candidate support set. Once again, due to the continuity of the hypersurfaces, the set $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ is locally invariant as we scan $\Phi(i,j)$. The points of interest are those points at which the ith hypersurface intersects another hypersurface. Provided that $\Phi(i,j)$ corresponds to points where the ith and jth hypersurfaces coincide, any intersection of the ith hypersurface with a third hypersurface, say the l-th one, will be a joint intersection of the three hypersurfaces $\{i,j,l\}$. The set of points where the hypersurfaces $\{i,j,l\}$ intersect is

$$\mathcal{H}(i,j,l) \subseteq \mathcal{H}(i,j),$$

for all $l \in [n+1] \backslash \{i,j\}$. Repeating this argument recursively, we conclude that it suffices to examine the intersections of subsets of r hypersurfaces,

[1] In the rank-2 case, the intersection was a single point.

$\mathcal{H}(i_1, i_2, \ldots, i_r)$, for all possible sets $\{i_1, i_2, \ldots, i_r\} \subseteq [n+1]$. Such intersections correspond to single points[2], where r hypersurfaces have the same value. By our perturbation argument, we can assume that exactly (*i.e.*, not more than) r hypersurfaces intersects at that exact $\boldsymbol{\phi}$. If all r intersecting hypersurfaces or none of them are included or excluded from $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$, the candidate set does not change (at least from the perspective of the ith hypersurface) around $\boldsymbol{\phi}$. That is, index i neither leaves nor joins the candidate support set at $\boldsymbol{\phi}$. On the contrary, if the r intersecting hypersurfaces $\{i_1, i_2, \ldots, i_r\}$ are nonnegative and in the s-th order at $\boldsymbol{\phi}$, then there are multiple candidate support sets associated with the area around $\boldsymbol{\phi}$. In each of these candidates, only a subset of $\{i_1, i_2, \ldots, i_r\}$ can be included in $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$, due to the constraint that $|\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))| \leq s$. However, hypersurfaces $\{i_1, i_2, \ldots, i_r\}$ are the only ones that might join or leave the candidate set at that particular point and there are at most 2^{r-1} partitions of $\{i_1, i_2, \ldots, i_r\}$ into two subsets. Hence, at most a constant number of candidates, readily determined, is associated with each such intersection point.

Building $\mathcal{S}_r$ We consider all points where r hypersurfaces intersect, *i.e.*, we find $\boldsymbol{\phi}$ such that

$$[\mathbf{v}(\boldsymbol{\phi})]_{i_1} = [\mathbf{v}(\boldsymbol{\phi})]_{i_2} = \ldots = [\mathbf{v}(\boldsymbol{\phi})]_{i_r},$$

[2]We assume that every r rows of $\mathbf{V}$ are linearly independent. If that is not the case, we can ignore the

for all possible sets $\{i_1, \ldots, i_r\} \subseteq [n+1]$. To that end, it suffices to find $\boldsymbol{\phi}$ where the pairwise equalities

$$[\mathbf{v}(\boldsymbol{\phi})]_{i_1} = [\mathbf{v}(\boldsymbol{\phi})]_{i_2}, \ \ldots \ , [\mathbf{v}(\boldsymbol{\phi})]_{i_1} = [\mathbf{v}(\boldsymbol{\phi})]_{i_r}$$

are jointly satisfied, or, equivalently, to find $\mathbf{c}(\boldsymbol{\phi})$ such that

$$\begin{bmatrix} \mathbf{e}_{i_1}^\top - \mathbf{e}_{i_2}^\top \\ \vdots \\ \mathbf{e}_{i_1}^\top - \mathbf{e}_{i_{r-1}}^\top \end{bmatrix} \mathbf{V}\mathbf{c}(\boldsymbol{\phi}) = \mathbf{0}_{r-1}.$$

In other words, we seek the unique (up to scaling) vector in the nullspace of the $r-1 \times r$ matrix multiplying $\mathbf{c}(\boldsymbol{\phi})$.

At the intersection point, the hypersurfaces indexed by $\{i_1, \ldots, i_r\}$ are all equal. If all r intersecting hypersurfaces are all (or none) included in $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$, then any modification in their sorting does not affect the set, in the sense that none of these r hypersurfaces leaves or joins the set of s largest nonnegative hypersurfaces. On the other hand, if the r hypersurfaces are nonnegative and equal to the s^{th} order hypersurface, then only a subset of them can be included in $\mathcal{I}_s^+(\mathbf{v}(\boldsymbol{\phi}))$ at any point around the intersection point. Further these are the only hypersurfaces that can leave or join the set at that point. If $1 \leq r \leq r-1$ of them can be included, then there must be at most $\binom{r}{r}$ candidates associated with the cells around $\boldsymbol{\phi}$ (or $\binom{r-1}{r}$ if one of them is the artificial zero hypersurface). That is, there can be at most $\binom{r}{\lceil \frac{r}{2} \rceil} \leq 2^{r-1}$ candidate support sets around $\boldsymbol{\phi}$.

We repeat the process for $\mathcal{I}_s^+(-\mathbf{v}(\boldsymbol{\phi}))$. Therefore, a maximum of $2 \cdot 2^{r-1}$

candidates are introduced at each intersection point, and

$$|\mathcal{S}_r| \leq 2^r \binom{n+1}{r} = O(n^r).$$

The candidates at each intersection point are determined in linear time: determining the entries of $\mathbf{v}(\boldsymbol{\phi})$ that are greater than its s-th largest nonnegative entry can be done in linear time, and the algorithm produces at most 2^{r-1} candidates at each intersection point. We conclude that $\mathcal{S}_r$ can be constructed in $O(n^{r+1})$.

A.4 Quadratic Maximization over Hyper-Spherical Cap

We consider the constrained quadratic maximization

$$\mathbf{c}_\star = \arg \max_{\substack{\|\mathbf{c}\|_2=1, \\ \mathbf{R}\mathbf{c}\geq 0}} \mathbf{c}^\top \mathbf{Q}\mathbf{c}, \qquad (\mathrm{P}_r)$$

where $\mathbf{Q}$ is a $r \times r$ symmetric matrix, and $\mathbf{R}$ is a real $s \times r$ matrix. In general, (P_r) is NP-hard: for $\mathbf{Q}$ PSD and $\mathbf{R}$ equal to the identity matrix $\mathbf{I}_r$, (P_r) reduces to the original problem in (2.2). In this section, however, we consider the case where the dimension r of the problem is a constant and develop an $O(s^r)$ algorithm for the non-trivial task of solving (P_r).

The $r = 1$ case. The optimization variable $\mathbf{c}$ is a scalar in $\{+1, -1\}$, and $\mathbf{R}$ is a vector in $\mathbb{R}^s$. If either $\mathbf{R} \geq \mathbf{0}$ or $-\mathbf{R} \geq \mathbf{0}$, the optimal solution is $\mathbf{c}^\star = 1$ or -1, respectively. Otherwise, the problem is infeasible.

The $r = 2$ case. The $r = 2$ case is the simplest nontrivial case. Let $\lambda_1 \geq \lambda_2$, be the two eigenvalues of $\mathbf{Q} \in \mathbb{S}^2$. The corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2$ form an orthonormal basis of $\mathbb{R}^2$. Any unit length vector $\mathbf{c}$ can be expressed as $\mathbf{c}(\phi) = \mathbf{U}[\cos(\phi),\ \sin(\phi)]^\top$, for some $\phi \in [0, 2\pi)$, where $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2]$.

The feasible region is an arc on the unit circle in the intersection of s half-spaces (see Fig. 2.2 for an example). It comprises vectors $\mathbf{c}(\phi)$ with ϕ restricted in some interval $[\phi_1,\ \phi_2]$. Note that ϕ_1 and ϕ_2 are points where at least one linear constraint becomes active. Unless $\mathbf{R}$ is the zero matrix, $0 \leq |\phi_1 - \phi_2| \leq \pi$. If $\pm\mathbf{u}_1$ lies in the feasible region, then $\mathbf{c}_\star = \pm\mathbf{u}_1$: the leading eigenvector is the global unconstrained maximum. The key observation is that if neither $\mathbf{u}_1$ or $-\mathbf{u}_1$ is feasible, the optimal solution coincides with either $\mathbf{c}(\phi_1)$ or $\mathbf{c}(\phi_2)$. To verify that, let

$$Q(\phi) = \mathbf{c}(\phi)^\top \mathbf{Q}\mathbf{c}(\phi) = \cos^2(\phi)\lambda_1 + \sin^2(\phi)\lambda_2$$

denote the quadratic objective in (P_2) as a function of ϕ. $Q(\phi)$ is differentiable with four critical points at $\phi = 0, \pi/2, \pi$, and $3\pi/2$. By assumption, $\phi = 0$ and $\phi = \pi$, which correspond to $\mathbf{c}(\phi) \pm \mathbf{u}_1$, lie outside the feasible interval $[\phi_1,\ \phi_2]$. Since $0 \leq |\phi_1 - \phi_2| \leq \pi$, at most one of the local minima $\phi = \pi/2$ and $\phi = 3\pi/2$ may lie in $[\phi_1, \phi_2]$. We conclude that either *i*) $Q(\phi)$ is monotonically increasing in $[\phi_1, \phi_2]$, *ii*) monotonically decreasing in $[\phi_1, \phi_2]$, or *iii*) has a unique local minimum in (ϕ_1, ϕ_2). In either case, $Q(\phi)$ attains its maximum at one ϕ_1 and ϕ_2.

The above motivate the following steps for solving (P_2):

Algorithm A.19 Compute the solution $\mathbf{c}_\star$ of (P_2)

input $\mathbf{Q}$ – 2×2 real symmetric matrix
$\mathbf{R}$ – $s \times 2$ real symmetric matrix
output $\mathbf{c}_\star$ – 2-dimensional real vector in the feasible region $\mathbf{Rc} \geq 0, \|\mathbf{c}\|_2 = 1$. See Lemma A.4.20.

$\mathbf{u}_1 \leftarrow$ leading eigenvector of $\mathbf{Q}$
if $\pm\mathbf{R}\mathbf{u}_1 \geq 0$ **then**
  $\mathbf{c}_\star \leftarrow \pm\mathbf{u}_1$
else
  $\mathcal{C} = \{\}$
  for $i = 1, \ldots, s$ **do**
    $\mathbf{c}_i \leftarrow [-R_{i,2},\ R_{i,1}]^\top / \|\mathbf{R}_{i,:}\|_2$
    if $\mathbf{R}(\pm\mathbf{c}_i) \geq 0$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\pm\mathbf{c}_i\}$
    end if
  end for $\{\mathcal{C} = \emptyset \Rightarrow (\mathrm{P}_2)$ infeasible$\}$
  $\mathbf{c}_\star \leftarrow \arg\max_{\mathbf{c} \in \mathcal{C}} \mathbf{c}^\top \mathbf{Q} \mathbf{c}$
end if

1. If $\pm\mathbf{R}\mathbf{u}_1 \geq \mathbf{0}$, then $\mathbf{c}_\star = \pm\mathbf{u}_1$.
2. Otherwise, initialize an empty collection $\mathcal{C}$ of candidate solutions. For $i = 1, \ldots, s$:
   - Compute $\mathbf{c}_i = \pm[-R_{i,2},\ R_{i,1}]^\top / \|\mathbf{R}_{i,:}\|$, the unit norm vectors in the direction at which the ith inequality is active. If $\pm\mathbf{c}_i$ is feasible, include $\pm\mathbf{c}_i$ in $\mathcal{C}$.
3. Return $\mathbf{c}_\star = \arg\max_{\mathbf{c} \in \mathcal{C}} \mathbf{c}^\top \mathbf{Q} \mathbf{c}$.

The previous steps are formally presented in Algorithm A.19.

Lemma A.4.20. *Algorithm A.19 computes the optimal solution of* (P_2) *with* s *linear inequality constraints in* $O(s^2)$.

Proof. There exist at most $2s + 2$ candidate solutions, including $\pm\mathbf{u}_1$. Each candidate is computed in $O(1)$, and its feasibility is checked in $O(s)$. In total, the collection $\mathcal{C}$ of feasible candidate solutions is constructed in $O(s^2)$. The optimal solution is determined via exhaustive comparison among the candidates in $\mathcal{C}$ in $O(s)$. □

The arbitrary r case. We demonstrate an algorithm to solve (P_r) for any arbitrary r. Our algorithm relies on generalizing the observations and ideas of the $r = 2$ case. In particular, assuming that the feasible region is non-empty, we will show the following claim:

Claim 1. *Let $\mathbf{u}_1 \in \mathbb{R}^r$ be the leading eigenvector of $\mathbf{Q}$. If $\pm\mathbf{u}_1$ is feasible, $\mathbf{c}_\star = \pm\mathbf{u}_1$ is the optimal solution of (P_r). Otherwise, at least one of the s linear constraints holds with equality at $\mathbf{c}_\star$, i.e., $\exists i \in [s]$ such that $\mathbf{R}_{i,:}\mathbf{c}_\star = 0$.*

Proof. Let $\mathbf{Q} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the eigenvalue decomposition of $\mathbf{Q}$: the diagonal entries of $\mathbf{\Lambda}$ coincide with the real eigenvalues of $\mathbf{Q}$, $\lambda_1, \ldots, \lambda_r$ in decreasing order, and the columns of $\mathbf{U}$ with the corresponding eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$. The latter form an orthonormal basis for $\mathbb{R}^r$.

Clearly, if either of $\pm\mathbf{u}_1$ is feasible, the quadratic objective attains its maximum value at $\mathbf{c}_\star = \pm\mathbf{u}_1$. In the sequel, we are concerned with the case where both $\pm\mathbf{u}_1$ are infeasible.

Consider a feasible point $\mathbf{c}_0$, $\|\mathbf{c}_0\| = 1$, such that $\mathbf{R}\mathbf{c}_0 > \mathbf{0}$. That is, $\mathbf{c}_0$ satisfies all linear constraints with strict inequality. If no such a point exists,

the claim holds trivially. We will show that $\mathbf{c}_0$ cannot be optimal.

In terms of the eigenbasis, we have $\mathbf{c}_0 = \mathbf{U}\boldsymbol{\mu}$, where $\boldsymbol{\mu} = \mathbf{U}^\top \mathbf{c}_0 \in \mathbb{R}^r$, $\|\boldsymbol{\mu}\| = 1$. Let $\widehat{\mathbf{c}}_0$ denote the orthogonal projection of $\mathbf{c}_0$ on the subspace spanned by the trailing eigenvectors $\mathbf{u}_2, \ldots, \mathbf{u}_r$, normalized to unit length. That is, $\widehat{\mathbf{c}}_0 = (1-\mu_1^2)^{-1} \sum_{i=2}^r \mathbf{u}_i \mu_i$. The unit norm vectors in the 2-dimensional span of $\mathbf{u}_1$ and $\widehat{\mathbf{c}}_0$ are all points of the form

$$\boldsymbol{\alpha}(\phi) = \mathbf{U} \begin{bmatrix} 1 & 0 \\ \mathbf{0} & \boldsymbol{\mu}_{2:d}/(1-\mu_1^2) \end{bmatrix} \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix},$$

for $\phi \in [0, 2\pi)$. Note that $\boldsymbol{\alpha}(0) = \mathbf{u}_1$ and $\boldsymbol{\alpha}(\pi) = -\mathbf{u}_1$ are by assumption infeasible. Therefore, points $\alpha(\phi)$ are feasible only for ϕ restricted to some interval $[\phi_1,\ \phi_2]$, with $0 \leq |\phi_1 - \phi_2| \leq \pi$. At the endpoints ϕ_1 and ϕ_2, at least one of the inequality constraints becomes active. Further, there exists a point $\phi_0 = \arccos\Big(\mathbf{u}_1^\top \mathbf{c}_0\Big) \in [\phi_1,\ \phi_2]$, such that $\boldsymbol{\alpha}(\phi_0) = \mathbf{c}_0$. By assumption, $\mathbf{R}_{i,:}\boldsymbol{\alpha}(\phi_0) > 0$, $\forall i \in [s]$.

Let $Q(\phi)$ denote the objective function of (P_r) over the unit norm vectors $\alpha(\phi)$, as a function of ϕ. We will show that $Q(\phi_0) \leq \max\{Q(\phi_1), Q(\phi_2)\}$. We have

$$\begin{aligned} Q(\phi) &= \boldsymbol{\alpha}(\phi)^\top \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top \boldsymbol{\alpha}(\phi) \\ &= \lambda_1 \cdot \cos(\phi)^2 + \frac{1}{1-\mu_1^2} \sum_{i=2}^r \mu_i^2 \lambda_i \cdot \sin(\phi)^2 \\ &= \lambda_1 + \frac{1}{1-\mu_1^2}\Big(\boldsymbol{\mu}^\top \boldsymbol{\Lambda}\boldsymbol{\mu} - \lambda_1\Big) \sin(\phi)^2. \end{aligned}$$

Taking into account that $\lambda_1 \geq \boldsymbol{\mu}^\top \boldsymbol{\Lambda}\boldsymbol{\mu}$, it is straightforward to verify through the first derivative w.r.t. ϕ that $Q(\phi)$ has four critical points at $\phi = 0,\ \pi/2,$

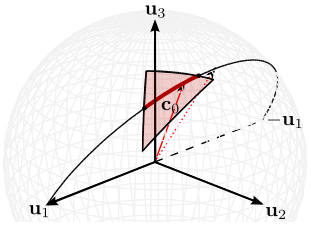


Figure A.2: An instance of (P_3). Feasible solutions lie in the intersection of half-spaces with the unit sphere (highlighted region). The leading eigenvector $\pm\mathbf{u}_1$ of $\mathbf{Q}$ is not feasible. Consider any solution $\mathbf{c}_0$ in the interior of the feasible region. The highlited arc corresponds to unit length points in the span of $\mathbf{c}_0$ and $\mathbf{u}_1$. The objective value at $\mathbf{c}_0$ cannot exceed the value at the endpoints of that arc.

π and $3\pi/2$. One of the following holds: *i*) $Q(\phi)$ is monotonically decreasing in $[\phi_1, \phi_2]$, *ii*) $Q(\phi)$ is monotonically increasing in $[\phi_1, \phi_2]$, or *iii*) a unique local minimum lies in (ϕ_1, ϕ_2). In either case, the maximum value of $Q(\phi)$ over $[\phi_1, \phi_2]$ is achieved at one of ϕ_1 and ϕ_2, which completes the proof. □

According to Claim 1, at least one linear inequality constraint holds with equality at the optimal point $\mathbf{c}_\star$. Assume that the ith linear constraint is such a constraint, *i.e.*, $\mathbf{R}_{i,:}\mathbf{c}_\star = 0$. In the sequel, we investigate how this extra assumption simplifies solving (P_r).

The constraint $\mathbf{R}_{i,:}\mathbf{c} = 0$ enforces a linear dependence on the entries of $\mathbf{c}$. Let $j \in [r]$ be the index of a nonzero entry[3] of $\mathbf{R}_{i,:}$. Let $\mathbf{c}_{\setminus j} \in \mathbb{R}^{r-1}$ and $\mathbf{R}_{i,\setminus j} \in \mathbb{R}^{1\times r-1}$ denote the vectors obtained excluding the jth entry of $\mathbf{c}$

[3]If no such j exists, the ith row of $\mathbf{R}$ is the zero vector. In that case, the ith linear constraint is redundant and can be omitted.

and $\mathbf{R}_{i,:}$, respectively. Then,

$$\mathbf{c} = \mathbf{H}\mathbf{c}_{\setminus j}, \tag{A.4}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{j-1\times j-1} & \mathbf{0}_{j-1\times r-j} \\ -R_{i,j}^{-1}\mathbf{R}_{i,\setminus j} & \\ \mathbf{0}_{r-j\times j-1} & \mathbf{I}_{r-j\times r-j} \end{bmatrix} \in \mathbb{R}^{d\times r-1}.$$

Let $\mathbf{H} = \mathbf{U}_H\mathbf{\Sigma}_H\mathbf{V}_H^\top$ be the compact singular value decomposition of the rank-$(r-1)$ matrix $\mathbf{H}$: $\mathbf{U}_H \in \mathbb{R}^{d\times r-1}$ consists of the $r-1$ leading left singular vectors of $\mathbf{H}$, $\mathbf{V}_H \in \mathbb{R}^{r-1\times r-1}$ is a unitary matrix comprising the right singular vectors, and $\mathbf{\Sigma}_H$ is a diagonal matrix containing the $r-1$ nonzero singular values of $\mathbf{H}$.

Define $\widehat{\mathbf{c}} = \mathbf{\Sigma}_H\mathbf{V}_H^\top\mathbf{c}_{\setminus j} \in \mathbb{R}^{r-1}$. Through (A.4), the original variable $\mathbf{c}$ can be expressed in terms of $\widehat{\mathbf{c}}$: $\mathbf{c} = \mathbf{U}_H\widehat{\mathbf{c}}$. Substituting $\mathbf{c}$ in (P_r) accordingly, we conclude that in order to compute the solution to (P_r), it suffices to compute

$$\widehat{\mathbf{c}}^{(i)} = \arg \max_{\substack{\mathbf{R}\mathbf{U}_H\widehat{\mathbf{c}}\geq\mathbf{0} \\ \|\mathbf{U}_H\widehat{\mathbf{c}}\|=1}} \widehat{\mathbf{c}}^\top\left(\mathbf{U}_H^\top\mathbf{Q}\mathbf{U}_H\right)\widehat{\mathbf{c}}. \tag{A.5}$$

The optimal solution of (P_r) will then be $\mathbf{c}^{(i)} = \mathbf{U}_H\widehat{\mathbf{c}}^{(i)}$, where the superscript is used to remind that $\mathbf{c}^{(i)}$ is optimal under the assumption that the ith linear inequality constraint is active. In practice, it is not known which constraints are active at the globally optimal solution $\mathbf{c}_\star$. However, on principle, we can compute s candidates $\mathbf{c}^{(i)}$, $i \in [s]$, one for each linear inequality constraint in $\mathbf{R}$, and determine $\mathbf{c}_\star$ via exhaustive comparison.

It remains to show that we can efficiently solve (A.5). Due to the fact that the columns of $\mathbf{U}_H$ are orthonormal, the requirement $\|\mathbf{U}_H\widehat{\mathbf{c}}\| = 1$ is equivalent to $\|\widehat{\mathbf{c}}\| = 1$, and (A.5) becomes

$$\widehat{\mathbf{c}}^{(i)} = \arg \max_{\substack{\mathbf{R}^{(i)}\widehat{\mathbf{c}} \geq 0 \\ \|\widehat{\mathbf{c}}\|_2 = 1}} \widehat{\mathbf{c}}^\top \mathbf{Q}^{(i)} \widehat{\mathbf{c}}, \qquad (P_{r-1}^{(i)})$$

with $\mathbf{R}^{(i)} = \mathbf{R}\mathbf{U}_H$ and $\mathbf{Q}^{(i)} = \mathbf{U}_H^\top \mathbf{Q}\mathbf{U}_H$. One can verify that the new optimization is identical in form to (P_r), but the dimension of the unknown variable is reduced to $r-1$. Further, the ith row of $\mathbf{R}^{(i)}$ is the all zero vector, effectively decreasing the number of linear constraints to $s-1$.

The Algorithm The above discussion motivates a recursion for solving (P_r). The procedure is outlined in the following steps and is formally presented in Algorithm A.20.

1. If $r = 2$, compute and return the optimal solution according Algorithm A.19.
2. Compute $\mathbf{u}_1 \in \mathbb{R}^r$, the leading eigenvector of $\mathbf{Q}$. If $\pm\mathbf{u}_1$ is feasible, return $\mathbf{c}_\star = \pm\mathbf{u}_1$.
3. Otherwise, for $i = 1, \ldots, s$:
   - Form the $(r-1)$-dimensional problem $(P_{r-1}^{(i)})$, setting the ith linear inequality constraint active.
   - Solve $(P_{r-1}^{(i)})$ recursively and obtain a candidate solution $\mathbf{c}^{(i)}$ of (P_r). Include $\mathbf{c}^{(i)}$ in $\mathcal{C}$, the collection of candidate solutions.

Algorithm A.20 Compute the solution $\mathbf{c}_\star$ of (P_r)

input $\mathbf{Q}$ – $r \times r$ real symmetric matrix
$\mathbf{R}$ – $s \times r$ real matrix
output $\mathbf{c}_\star$ – r-dimensional real vector in the feasible region $\mathbf{Rc} \geq 0, \|\mathbf{c}\|_2 = 1$. See Lemma A.4.21.

if $r = 2$ **then**
  $\mathbf{c}_\star \leftarrow$ solve $(P_2[\mathbf{Q}, \mathbf{R}])$
end if
$\mathbf{u}_1 \leftarrow$ leading eigenvector of $\mathbf{Q}$
if $\mathbf{R}(\pm\mathbf{u}_1) \geq 0$ **then**
  $\mathbf{c}_\star \leftarrow \pm\mathbf{u}_1$
else
  $\mathcal{C} = \{\}$ {Set of candidate solutions}
  for $i = 1$ **to** k **do**
    $j = \min\{l \in [r] : R_{il} \neq 0\}$
    $\mathbf{H} \leftarrow \begin{bmatrix} \mathbf{I}_{j-1\times j-1} & \mathbf{0}_{j-1\times r-j} \\ -R_{i,j}^{-1}\mathbf{R}_{i,\setminus j} & \\ \mathbf{0}_{r-j\times j-1} & \mathbf{I}_{r-j\times r-j} \end{bmatrix}$
    $\mathbf{U}_H, \mathbf{\Sigma}_H, \mathbf{V}_H \leftarrow \mathtt{svd}(\mathbf{H})$
    $\mathbf{Q}^{(i)} \leftarrow {\mathbf{U}_H}^\top \mathbf{Q}\mathbf{U}_H$ $\{\in \mathbb{S}^{r-1}\}$
    $\mathbf{R}^{(i)} \leftarrow \mathbf{R}_{\setminus i,:}\mathbf{U}_H$ $\{\in \mathbb{R}^{k-1\times r-1}\}$
    $\widehat{\mathbf{c}}^{(i)} \leftarrow \mathsf{solve}\left(P_{r-1}^{(i)}\left[\mathbf{Q}^{(i)},\ \mathbf{R}^{(i)}\right]\right)$ $\{\in \mathbb{R}^{r-1}\}$
    $\mathbf{c}^{(i)} \leftarrow \mathbf{U}_H\widehat{\mathbf{c}}^{(i)}$ $\{\in \mathbb{R}^r\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{c}^{(i)}\}$ $\{|\mathcal{C}| \leq s\}$
  end for
  $\mathbf{c}_\star \leftarrow \arg\max_{\mathbf{c}\in\mathcal{C}}\left(\mathbf{c}^\top\mathbf{Q}\mathbf{c}\right)$
end if

4. Return $\mathbf{c}_\star = \arg\max_{c \in \mathcal{C}} \mathbf{c}^\top \mathbf{Q} \mathbf{c}$.

Lemma A.4.21. *Algorithm A.20 solves* (P_r) *in* $O(s^r)$.

Proof. Let $C(r, s)$ denote the complexity of solving (P_r) with s linear inequality constraints using Algorithm A.20. The leading eigenvector of $\mathbf{Q}$ is computed in $O(r^3)$ and its feasibility can be verified in $O(rs)$. Each of the s sub-problems $(P_{r-1}^{(i)})$ of dimension $r-1$ with $s-1$ inequalities can be formulated in $O(r^3 s)$ and solved recursively in $C(r-1, s-1)$. The maximum recursion depth is $r-2$ and the base problem (P_2) is solved in $O(s^2)$ by Alg. A.19. In total, $C(r, s) = k \cdot C(r-1, s-1) + O(r^3 s^2)$, which in turn yields $C(r, s) = O(s^r)$. □

A.5 Near-Linear Time Nonnegative SPCA

Alg. 2.1 approximates the nonnegative, s-sparse principal component of an $n \times n$ PSD matrix $\mathbf{A}$,

$$\mathbf{x}_\star = \arg\max_{\mathbf{x} \in \mathbb{S}_s^n} \quad \mathbf{x}^\top \mathbf{A} \mathbf{x},$$

by efficiently solving the nonnegative sparse PCA problem on $\mathbf{A}_r$, the best rank-r approximation of $\mathbf{A}$. More precisely, Alg. 2.1 computes and outputs

$$\mathbf{x}_r = \arg\max_{\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r \mathbf{x}, \tag{A.6}$$

in time polynomial in n, for any constant r. The output $\mathbf{x}_r$ is a surrogate for the desired vector $\mathbf{x}_\star$.

Albeit polynomial in n, the computational complexity of Alg. 2.1 can be impractical even for moderate values of n. In this section, we develop Algorithm 2.3, a simple randomized procedure for approximating the nonnegative, s-sparse principal component of a PSD matrix in time almost linear in n. Alg. 2.3 relies on the same core ideas as Alg. 2.1: solve the nonnegative sparse PCA problem on a rank-r matrix $\mathbf{A}_r$ recasting the maximization in (A.6) into a series of simpler problems. But instead of computing the exact solution $\mathbf{x}_r$ of the rank-r nonnegative PCA problem, Alg. 2.3 settles for an approximate solution $\widehat{\mathbf{x}}_r$ computed in near-linear time. This second level of approximation introduces an additional error: $\widehat{\mathbf{x}}_r$ may be a slightly worse approximation of $\mathbf{x}_\star$ compared to $\mathbf{x}_r$. That extra approximation error, however, can be made arbitrarily small.

Lemma A.5.22. *Let $\mathbf{A}$ be an $n \times n$ PSD matrix given as input to Alg. 2.3, along with sparsity parameter $s \in [n]$ and accuracy parameters $r \in [n]$ and $\epsilon \in (0, 1]$. Let $\mathbf{A}_r$ be the best rank-r approximation of $\mathbf{A}$, and $\mathbf{x}_r$ its nonnegative, s-sparse principal component. Alg. 2.3 outputs a nonnegative, s-sparse, unit norm vector $\widehat{\mathbf{x}}_r$ such that*

$$\widehat{\mathbf{x}}_r^\top \mathbf{A}_r \widehat{\mathbf{x}}_r \;\geq\; (1-\epsilon) \cdot \mathbf{x}_r{}^\top \mathbf{A}_r \mathbf{x}_r,$$

with probability at least $1 - 1/n$, in time $O(\epsilon^{-r} \cdot n \log n)$ plus the time required to compute the r leading eigenvectors of $\mathbf{A}$.

The lemma follows from the analysis of Alg. 2.3, which is the focus of Section A.5.1, and its proof is deferred until the end of that section. According

to the lemma, the output $\widehat{\mathbf{x}}_r$ of Alg. 2.3 is a factor $(1-\epsilon)$ approximation of $\mathbf{x}_r$, the nonnegative, s-sparse principal component of $\mathbf{A}_r$, in terms of explained variance on the rank-r matrix $\mathbf{A}_r$. Our ultimate goal, however, is to characterize the quality of $\widehat{\mathbf{x}}_r$ as a surrogate for $\mathbf{x}_\star$, the nonnegative, s-sparse principal component of $\mathbf{A}$. The approximation guarantees of Alg. 2.3 are established in the following theorem.

Theorem 2.2. *For any $n \times n$ PSD matrix $\mathbf{A}$, sparsity parameter s, and accuracy parameters $r \in [n]$ and $\epsilon \in (0, 1]$, Alg. 2.3 outputs a nonnegative, s-sparse, unit norm vector $\widehat{\mathbf{x}}_r$ such that*

$$\widehat{\mathbf{x}}_r^\top \mathbf{A} \widehat{\mathbf{x}}_r \;\geq\; (1-\epsilon) \cdot \rho_r \cdot \mathbf{x}_\star{}^\top \mathbf{A} \mathbf{x}_\star,$$

with probability at least $1 - 1/n$, in time $O(\epsilon^{-r} \cdot n \log n)$ plus the time required to compute the r leading eigenvectors of $\mathbf{A}$.

Proof. For the output $\widehat{\mathbf{x}}_r$ of Alg. 2.3, we have

$$\begin{aligned}
\widehat{\mathbf{x}}_r^\top \mathbf{A} \widehat{\mathbf{x}}_r &= \widehat{\mathbf{x}}_r^\top \mathbf{A}_r \widehat{\mathbf{x}}_r + \widehat{\mathbf{x}}_r^\top (\mathbf{A} - \mathbf{A}_r) \widehat{\mathbf{x}}_r \\
&\overset{(\alpha)}{\geq} (1-\epsilon) \cdot \mathbf{x}_r{}^\top \mathbf{A}_r \mathbf{x}_r + \widehat{\mathbf{x}}_r^\top (\mathbf{A} - \mathbf{A}_r) \widehat{\mathbf{x}}_r \\
&\overset{(\beta)}{\geq} (1-\epsilon) \cdot \mathbf{x}_r{}^\top \mathbf{A}_r \mathbf{x}_r \\
&= (1-\epsilon) \cdot \mathrm{OPT}_r \\
&= (1-\epsilon) \cdot \rho_r \cdot \mathrm{OPT}, \qquad \text{(A.7)}
\end{aligned}$$

where inequality (a) follows from Lemma A.5.22, and (β) from the fact that $\mathbf{A} - \mathbf{A}_r$ is a PSD matrix. Note that by Lemma A.2.19, $\rho_r = \mathrm{OPT}_r/\mathrm{OPT}$

satisfies

$$\rho_r \;\geq\; \max\left\{\frac{s}{2n},\; \frac{1}{1+2\frac{n}{s}\lambda_{r+1}/\lambda_1}\right\}.$$

The complexity of Alg. 2.3 is established in Lemma A.5.22, which completes the proof. □

A.5.1 Analysis of Algorithm 2.3

In this subsection, we examine Alg. 2.3 in detail and gradually build towards establishing Lemma A.5.22.

Given an $n \times n$ PSD matrix $\mathbf{A}$ and an accuracy parameter r, Alg. 2.3 first computes the r leading eigenvectors of $\mathbf{A}$ to obtain the rank-r approximation $\mathbf{A}_r$. Let $\mathbf{V}$ be an $n \times r$ square root of $\mathbf{A}_r$. That is, $\mathbf{A}_r = \mathbf{V}\mathbf{V}^\top$. In subsection 2.4.2, we showed that the rank-r nonnegative sparse PCA problem on $\mathbf{A}_r$ can be written as

$$\max_{\mathbf{x}\in\mathbb{S}_s^n} \mathbf{x}^\top \mathbf{A}_r \mathbf{x} = \max_{\mathbf{c}\in\mathbb{S}^r,\; \mathbf{x}\in\mathbb{S}_s^n} \max \Big((\mathbf{V}\mathbf{c})^\top \mathbf{x}\Big)^2. \tag{A.8}$$

For a fixed $\mathbf{c}$, the optimal $\mathbf{x}$ can be easily determined as described in Section 2.4.1. In principle, scanning all vectors $\mathbf{c}$ on the surface of the r-dimensional unit sphere $\mathbb{S}^r$ would suffice to detect the nonnegative, s-sparse principal component $\mathbf{x}_r$.

Alg. 2.3 approximately solves the double maximization in (A.8) considering a finite set of $m = O(\epsilon^{-r} \cdot \log n)$ points $\mathbf{c}_1, \ldots, \mathbf{c}_m$ drawn randomly and independently, uniformly distributed over $\mathbb{S}^r$. Each random point $\mathbf{c}_i$ corresponds to an n-dimensional vector $\mathbf{a}_i = \mathbf{V}\mathbf{c}_i$ in the range of $\mathbf{A}_r$, for which

Alg. 2.3 solves the rank-1 nonnegative sparse PCA problem

$$\max_{\mathbf{x}\in\mathbb{S}_s^n}\Big((\mathbf{V}\mathbf{c}_i)^\top\mathbf{x}\Big)^2 = \max_{\mathbf{x}\in\mathbb{S}_s^n}\Big(\mathbf{a}_i{}^\top\mathbf{x}\Big)^2.$$

The rank-1 problem is solved in time $O(n)$ as described in Section 2.4.1, and yields a candidate solution $\mathbf{x}$. Alg. 2.3 outputs the candidate that maximizes $\|\mathbf{V}^\top\mathbf{x}\|^2 = \mathbf{x}^\top\mathbf{A}_r\mathbf{x}$.

In the following, we argue that the $m = O(\epsilon^{-r}\cdot\log n)$ random samples suffice to establish the approximation guarantees of Lemma A.5.22, and in turn Theorem 2.2.

Randomized Construction of an ϵ-net An ϵ-net on the unit sphere $\mathbb{S}^r$ is a set $\mathcal{N}_\epsilon^r$ of points on $\mathbb{S}^r$ such that for any point on $\mathbb{S}^r$ there exists a point in $\mathcal{N}_\epsilon^r$ within euclidean distance ϵ. More formally,

Def. A.5.3. *An ϵ-net of $\mathbb{S}^r$ is a set $\mathcal{N}_\epsilon^r \subset \mathbb{S}^r$ such that*

$$\forall\mathbf{c}\in\mathbb{S}^r,\ \ \exists\ \widehat{\mathbf{c}}\in\mathcal{N}_\epsilon^r : \|\mathbf{c}-\widehat{\mathbf{c}}\|_2 \le \epsilon.$$

Consider a $(\epsilon/2)$-net $\mathcal{N}_{\epsilon/2}^r$ on $\mathbb{S}^r$ for some given constant $0 < \epsilon \le 1$. The following lemma states that if we solve the maximization in (A.8) over the points $\mathbf{c}$ in the finite set of points $\mathcal{N}_{\epsilon/2}^r$ instead of the entire sphere $\mathbb{S}^r$, we obtain a solution that is within a factor $(1-\epsilon)$ from OPT_r.

Lemma A.5.23. *Let $\mathcal{N}_{\epsilon/2}^r$ be a $\epsilon/2$-net of $\mathbb{S}^r$. Then,*

$$(1-\epsilon)\cdot\mathrm{OPT}_r \ \le\ \max_{\mathbf{c}\in\mathcal{N}_{\epsilon/2}^r}\max_{\mathbf{x}\in\mathbb{S}_s^n}\Big(\mathbf{c}^\top\mathbf{V}^\top\mathbf{x}\Big)^2 \ \le\ \mathrm{OPT}_r.$$

Proof. The upper bound follows from the fact that $\mathcal{N}^r_{\epsilon/2} \subseteq \mathbb{S}^r$. For the lower bound, let $(\mathbf{x}_r, \mathbf{c}_r)$ denote the optimal solution of (A.8), *i.e.*,

$$\mathrm{OPT}_r = (\mathbf{c}_r \mathbf{V}^\top \mathbf{x}_r)^2.$$

By definition, the $\epsilon/2$-net $\mathcal{N}^r_{\epsilon/2}$ contains a vector $\widehat{\mathbf{c}}_r$ such that $\mathbf{c}_r = \widehat{\mathbf{c}}_r + \mathbf{r}$ for some $\mathbf{r} \in \mathbb{R}^r$ with $\|\mathbf{r}\| \leq \epsilon/2$. Then,

$$\begin{aligned} \mathrm{OPT}_r^{1/2} &= |\mathbf{c}_r^\top \mathbf{V}^\top \mathbf{x}_r| = |(\widehat{\mathbf{c}}_r + \mathbf{r})^\top \mathbf{V}^\top \mathbf{x}_r| \\ &\overset{(\alpha)}{\leq} |\widehat{\mathbf{c}}_r^\top \mathbf{V}^\top \mathbf{x}_r| + \frac{\epsilon}{2} \cdot \|\mathbf{V}^\top \mathbf{x}_r\| \\ &= |\widehat{\mathbf{c}}_r^\top \mathbf{V}^\top \mathbf{x}_r| + \frac{\epsilon}{2} \cdot \sqrt{\mathrm{OPT}_r}, \end{aligned} \tag{A.9}$$

where (α) is due to the triangle inequality, the Cauchy-Schwartz inequality, and the fact that $\|\mathbf{r}\| \leq \epsilon/2$. From (A.9), it follows that

$$\left(\widehat{\mathbf{c}}_r^\top \mathbf{V}^\top \mathbf{x}_r\right)^2 \geq (1 - \epsilon/2)^2 \cdot \mathrm{OPT}_r \geq (1 - \epsilon) \cdot \mathrm{OPT}_r.$$

Noting that

$$\max_{\mathbf{c} \in \mathcal{N}^r_{\epsilon/2}} \max_{\mathbf{x} \in \mathbb{S}^n_s} \left(\mathbf{c}^\top \mathbf{V}^\top \mathbf{x}\right)^2 \;\geq\; \left(\widehat{\mathbf{c}}_r^\top \mathbf{V}^\top \mathbf{x}_r\right)^2$$

completes the proof. □

There are many constructions for ϵ-nets on the sphere, both deterministic and randomized [124, 30, 58]. In the following we review a simple randomized construction, initially studied by Wyner [151] in the asymptotic $r \to \infty$ regime. First, note the following existential result.

Lemma A.5.24 ([140]). *For any $0 < \epsilon \leq 1$, there exists an ϵ-net $\mathcal{N}_\epsilon^r$ of the unit sphere $\mathbb{S}^r$ with cardinality at most $m_{\epsilon,r} \leq (1 + 2/\epsilon)^r$.*

Consider a set of sphere-caps of radius $\widehat{\epsilon}$ centered at the points of the $\widehat{\epsilon}$-net $\mathcal{N}_{\widehat{\epsilon}}^r$. The caps cover the entire sphere surface. It can be easily shown, based on a simple triangle inequality, that an arbitrary collection of points comprising at least one point from each cap, forms a $(2\widehat{\epsilon})$-net. Further, using standard balls and bins arguments, we conclude that randomly and independently drawing $O(m_{\widehat{\epsilon},r} \cdot \log\big(n \cdot m_{\widehat{\epsilon},r}\big))$ points uniformly distributed over $\mathbb{S}^r$ suffice for at least one random point to lie in each sphere cap with probability at least $1 - 1/n$. That is, $O(\widehat{\epsilon}^{-r} \cdot \log n)$ points suffice to form a $(2\widehat{\epsilon})$-net. Hence, for $\widehat{\epsilon} = \epsilon/4$, we will obtain an $\epsilon/2$-net.

Lemma A.5.25. *Randomly and independently drawing $O(\epsilon^{-r} \cdot \log n)$ points uniformly distributed on $\mathbb{S}^r$ suffices to form an $\epsilon/2$-net of $\mathbb{S}^r$, with probability at least $1 - 1/n$.*

Proof of Lemma A.5.22 By Lemma A.5.25, the $O(\epsilon^{-r} \cdot \log n)$ points drawn randomly and independently uniformly over $\mathbb{S}^r$, form an $\epsilon/2$-net $\mathcal{N}_{\epsilon/2}^r$, with probability at least $1 - 1/n$. Alg. 2.3 solves the double maximization problem in (A.8) over the points in $\mathcal{N}_{\epsilon/2}^r$, and outputs a nonnegative, s-sparse, unit-norm vector $\widehat{\mathbf{x}}_r$. By Lemma A.5.23, $\widehat{\mathbf{x}}_r$ is within factor of $(1 - \epsilon)$ from OPT_r, which proves the desired approximation guarantee.

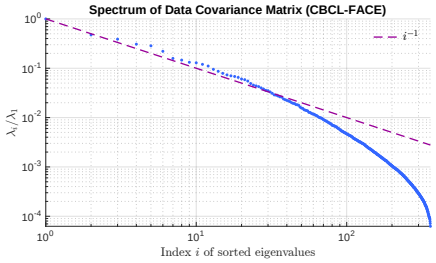
Alg. 2.3 examines $O(\epsilon^{-r} \log n)$ points in the range of the $n \times r$ matrix $\mathbf{V}$.

Each sample yields a candidate solution computed in $O(n)$. The total computational complexity is $O(\epsilon^{-r} \cdot n \log n)$, plus the time required to compute the r columns of $\mathbf{V}$, *i.e.*, the r leading eigenvectors of $\mathbf{A}$, which completes the proof. ■

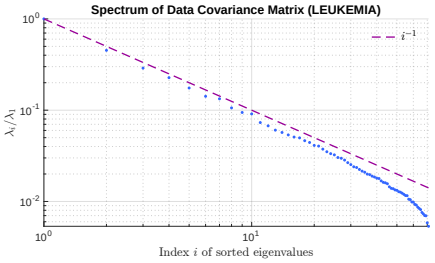
A.6 Examples of Spectral Decay in Real Data

The approximation guarantees of our algorithm are contingent on the spectrum of the data covariance matrix: the sharper the eigenvalue decay, the tighter the approximation. Here, we provide empirical evidence that real datasets often exhibit a steep decline in the spectrum of their empirical covariance matrix.

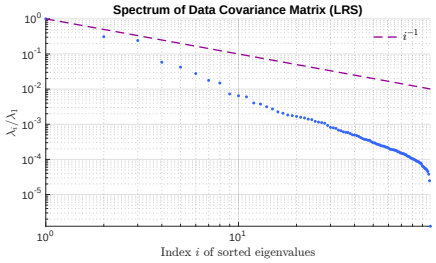
Fig. A.3 depicts the leading eigenvalues of the empirical covariance matrix of various datasets, normalized by the maximum eigenvalue, λ_1. In all depicted cases, the eigenvalues can be upper bounded by a power law decay function, *i.e.*, $\lambda_i \leq c \cdot \lambda_1 \cdot i^{\alpha}$, for some constant c and $\alpha \geq 1$.



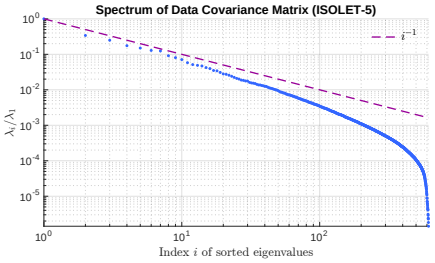
(a) CBCL Face Dataset



(b) Leukemia Dataset



(c) Low Res. Spectr. Dataset



(d) Isolet Dataset

Figure A.3: Spectrum of the empirical covariance matrix of various real datasets. The eigenvalues exhibit approximately power law decay. (Datasets A.3(b), A.3(c) and A.3(d) are available at [22]).

Appendix B

Appendix for Chapter 3

B.1 On the sub-optimality of deflation – An example

We provide a simple example demonstrating the sub-optimality of deflation based approaches for computing multiple sparse components with disjoint supports. Consider the real 4×4 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \epsilon \\ 0 & \delta & 0 & 0 \\ 0 & 0 & \delta & 0 \\ \epsilon & 0 & 0 & 1 \end{bmatrix},$$

with $\epsilon, \delta > 0$ such that $\epsilon + \delta < 1$. Note that $\mathbf{A}$ is PSD; $\mathbf{A} = \mathbf{B}^\top \mathbf{B}$ for

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & \epsilon \\ 0 & \sqrt{\delta} & 0 & 0 \\ 0 & 0 & \sqrt{\delta} & 0 \\ 0 & 0 & 0 & \sqrt{1-\epsilon^2} \end{bmatrix}.$$

We seek two 2-sparse components with disjoint supports, *i.e.*, the solution to

$$\max_{\mathbf{X} \in \mathcal{X}} \sum_{j=1}^{2} \mathbf{x}_j^\top \mathbf{A} \mathbf{x}_j, \tag{B.1}$$

where

$$\mathcal{X} \triangleq \Big\{ \mathbf{X} \in \mathbb{R}^{4 \times 2} : \|\mathbf{x}_i\|_2 \leq 1, \|\mathbf{x}_i\|_0 \leq 2 \; \forall \, i \in \{1,2\}, \mathrm{supp}(\mathbf{x}_1) \cap \mathrm{supp}(\mathbf{x}_2) = \emptyset \Big\}.$$

Iterative extraction with deflation Following an iterative, greedy procedure with a deflation step, we compute one component at the time. The first component is

$$\mathbf{x}_1 = \underset{\|\mathbf{x}\|_0=2, \|\mathbf{x}\|_2=1}{\operatorname{argmax}} \mathbf{x}^\top \mathbf{A}\mathbf{x}. \tag{B.2}$$

Recall that for any unit norm vector $\mathbf{x}$ with support $I = \operatorname{supp}(\mathbf{x})$,

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} \leq \lambda_{\max}(\mathbf{A}_{I,I}), \tag{B.3}$$

where $\mathbf{A}_{I,I}$ denotes the principal submatrix of $\mathbf{A}$ formed by the rows and columns indexed by I. Equality can be achieved in (B.3) for $\mathbf{x}$ equal to the leading eigenvector of $\mathbf{A}_{I,I}$. Hence, it suffices to determine the optimal support for $\mathbf{x}_1$. Due to the small size of the example, it is easy to determine that the set $I_1 = \{1, 4\}$ maximizes the objective in (B.3) over all sets of two indices, achieving value

$$\mathbf{x}_1^\top \mathbf{A}\mathbf{x}_1 = \lambda_{\max}\left(\begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}\right) = 1 + \epsilon. \tag{B.4}$$

Since subsequent components must have disjoint supports, it follows that the support of the second 2-sparse component $\mathbf{x}_2$ is $I_2 = \{2, 3\}$, and $\mathbf{x}_2$ achieves value

$$\mathbf{x}_2^\top \mathbf{A}\mathbf{x}_2 = \lambda_{\max}\left(\begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix}\right) = \delta. \tag{B.5}$$

In total, the objective value in (B.1) achieved by the greedy computation with a deflation step is

$$\sum_{j=1}^{2} \mathbf{x}_j^\top \mathbf{A}\mathbf{x}_j = 1 + \epsilon + \delta. \tag{B.6}$$

Sub-optimality of deflation Consider an alternative pair of 2-sparse components $\mathbf{x}'_1$ and $\mathbf{x}'_2$ with support sets $I'_1 = \{1, 2\}$ and $I'_2 = \{3, 4\}$, respectively. Based on the above, such a pair achieves objective value in (B.1) equal to

$$\lambda_{\max}\left(\begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}\right) + \lambda_{\max}\left(\begin{bmatrix} \delta & 0 \\ 0 & 1 \end{bmatrix}\right) = 1 + 1 = 2,$$

which clearly outperforms the objective value in (B.6) (under the assumption $\epsilon + \delta < 1$), demonstrating the sub-optimality of the $\mathbf{x}_1$, $\mathbf{x}_2$ pair computed by the deflation-based approach. In fact, for small ϵ, δ the objective value in the second case is larger than the former by almost a factor of two.

B.2 Construction of Bipartite Graph

The following algorithm formally outlines the steps for generating the bipartite graph $G = \left(\{U_j\}_{j=1}^k, V, E\right)$ given a *weight* $d \times k$ matrix $\mathbf{W}$.

Algorithm B.21 Generate Bipartite Graph

input $\mathbf{W}$ – $d \times k$ real matrix

output G – Weighted bipartite graph $G = \left(\{U_j\}_{j=1}^k, V, E\right)$. See Fig. 3.1.

for $j = 1, \ldots, k$ **do**
  $U_j \leftarrow \left\{u_1^{(j)}, \ldots, u_s^{(j)}\right\}$
end for
$U \leftarrow \cup_{j=1}^k U_j$ $\{|U| = k \cdot s\}$
$V \leftarrow \{1, \ldots, d\}$
$E \leftarrow U \times V$
for $i = 1, \ldots, d$ **do**
  for $j = 1, \ldots, k$ **do**
    for each $u \in U_j$ **do**
      $w(u, v_i) \leftarrow W_{ij}^2$
    end for
  end for
end for

B.3 Proofs

B.3.1 Guarantees of Algorithm 3.5

Lemma 3.3.1. *For any real $d \times k$ matrix $\mathbf{W}$, and Algorithm 3.5 outputs*

$$\widetilde{\mathbf{X}} = \underset{\mathbf{X} \in \mathcal{X}_k}{\operatorname{argmax}} \sum_{j=1}^{k} \left\langle \mathbf{X}^j, \mathbf{W}^j \right\rangle^2 \tag{B.7}$$

in time $O(d \cdot (s \cdot k)^2)$.

Proof. Consider a matrix $\mathbf{X} \in \mathcal{X}_k$ and let I_j, $j = 1 \ldots, k$ denote the support sets of its columns. By the constraints in $\mathcal{X}_k$, those sets are disjoint, *i.e.*, $I_{j_1} \cap I_{j_2} = \emptyset \; \forall j_1, j_2 \in \{1, \ldots, k\}, j_1 \neq j_2$, and

$$\sum_{j=1}^{k} \left\langle \mathbf{X}^j, \, \mathbf{W}^j \right\rangle^2 = \sum_{j=1}^{k} \left(\sum_{i \in I_j} X_{ij} \cdot W_{ij} \right)^2 \leq \sum_{j=1}^{k} \left(\sum_{i \in I_j} W_{ij}^2 \right). \tag{B.8}$$

The last inequality is due to Cauchy-Schwarz and the fact that $\|\mathbf{X}^j\|_2 \leq 1$, $\forall \, j \in \{1, \ldots, k\}$. In fact, if the supports sets I_j, $j = 1, \ldots, k$ were known, the upper bound in (B.8) would be achieved by setting $\mathbf{X}^j_{I_j} = \mathbf{W}^j_{I_j} / \|\mathbf{W}^j_{I_j}\|_2$, *i.e.*, setting the nonzero subvector of the jth column of $\mathbf{X}$ colinear to the corresponding subvector of the jth column of $\mathbf{W}$. Hence, the key step towards computing the optimal solution $\widetilde{\mathbf{X}}$ is to determine the support sets I_j, $j = 1, \ldots, k$ of its columns.

Consider the set of binary matrices $\mathcal{Z} \triangleq \mathcal{Z}^{(1)} \cap \mathcal{Z}^{(2)}$ where

$$\mathcal{Z}^{(1)} \triangleq \left\{ \mathbf{Z} \in \{0,1\}^{d \times k} : \|\mathbf{Z}^j\|_0 \leq s, \; \forall \, j \in [k] \right\},$$
$$\mathcal{Z}^{(2)} \triangleq \left\{ \mathbf{Z} \in \{0,1\}^{d \times k} : \operatorname{supp}(\mathbf{Z}^i) \cap \operatorname{supp}(\mathbf{Z}^j) = \emptyset, \; \forall \, i, j \in [k], i \neq j \right\}.$$

The set represents all possible supports for the members of $\mathcal{X}_k$. Taking into account the previous discussion, the maximization in (B.7) can be written with respect to $\mathbf{Z} \in \mathcal{Z}$:

$$\max_{\mathbf{X} \in \mathcal{X}_k} \sum_{j=1}^{k} \left\langle \mathbf{X}^j, \mathbf{W}^j \right\rangle^2 = \max_{\mathbf{Z} \in \mathcal{Z}} \sum_{j=1}^{k} \sum_{i=1}^{d} Z_{ij} W_{ij}^2. \tag{B.9}$$

Let $\widetilde{\mathbf{Z}} \in \mathcal{Z}$ denote the optimal solution, which corresponds to the (support) indicator of $\widetilde{\mathbf{X}}$. Next, we show that computing $\widetilde{\mathbf{Z}}$ boils down to solving a maximum weight matching problem on the bipartite graph generated by Algorithm B.21. Recall that given $\mathbf{W} \in \mathbb{R}^{d \times k}$, Algorithm B.21 generates a complete weighted bipartite graph $G = (U, V, E)$ where

- V is a set of d vertices $v_1, \ldots, v_d$, corresponding to the d variables, *i.e.*, the d rows of $\widehat{\mathbf{X}}$.
- U is a set of $k \cdot s$ vertices, conceptually partitioned into k disjoint subsets $U_1, \ldots, U_k$, each of cardinality s. The jth subset, U_j, is associated with the support $\mathcal{I}_j$; the s vertices $u_\alpha^{(j)}$, $\alpha = 1, \ldots, s$ in U_j serve as placeholders for the variables/indices in $\mathcal{I}_j$.
- Finally, the edge set is $E = U \times V$. The edge weights are determined by the $d \times k$ matrix $\mathbf{W}$ in (3.6). In particular, the weight of edge $(u_\alpha^{(j)}, v_i)$ is equal to W_{ij}^2. Note that all vertices in U_j are effectively identical; they all share a common neighborhood and edge weights.

It is straightforward to verify that any $\mathbf{Z} \in \mathcal{Z}$ corresponds to a perfect matching in G and vice versa; $Z_{ij} = 1$ if and only if vertex $v_i \in V$ is matched with

a vertex in U_j (all vertices in U_j are equivalent with respect to their neighborhood). Further, for a given $\mathbf{Z} \in \mathcal{Z}$ the objective value in (B.9) is equal to the weight of the corresponding matching in G. More formally, For a given perfect matching $\mathcal{M} \subset E$, the corresponding indicator matrix $\mathbf{Z} \in \mathcal{Z}$ (and equivalently the support of its columns) is determined by setting

$$I_j \leftarrow \{i \in [d] : (u, v_i) \in \mathcal{M}, u \in U_j\}, \quad j = 1, \ldots, k. \tag{B.10}$$

The weight of the matching $\mathcal{M}$ is

$$\sum_{(u,v)\in\mathcal{M}} w(u,v) = \sum_{j=1}^{k} \sum_{\substack{(u,v_i)\in\mathcal{M}: \\ u\in U_j}} w(u,v_i) = \sum_{j=1}^{k} \sum_{i\in I_j} W_{ij}^2 = \sum_{j=1}^{k} \sum_{i=1}^{d} Z_{ij} \cdot W_{ij}^2, \tag{B.11}$$

which is equal to the objective function in (B.9). Conversely, any given indicator matrix $\mathbf{Z} \in \mathcal{Z}$ corresponds to a perfect matching $\mathcal{M} \subset E$. In particular, letting $I_j \triangleq \text{supp}(\mathbf{Z}^j)$, and for an arbitrary ordering $\sigma_j : [s] \to I_j$ of the elements of I_j, $\mathcal{M} \leftarrow \left\{(u_\alpha^{(j)}, v_{\sigma_j(\alpha)}), \alpha = 1, \ldots, s,\ j = 1, \ldots, k\right\}$ is a perfect matching in G. The weight of the matching $\mathcal{M}$ is equal to the objective value in (B.9) for that $\mathbf{Z}$:

$$\sum_{j=1}^{k} \sum_{i=1}^{d} Z_{ij} \cdot W_{ij}^2 = \sum_{j=1}^{k} \sum_{i\in I_j} W_{ij}^2 = \sum_{j=1}^{k} \sum_{\alpha=1}^{s} W_{I_j(\alpha),j}^2 = \sum_{(u,v)\in\mathcal{M}} w(u,v). \tag{B.12}$$

It follows that to determine $\widetilde{\mathbf{Z}}$ that maximizes (B.9) with respect to $\mathbf{Z} \in \mathcal{Z}$, it suffices to compute a maximum weight perfect matching in G. Then $\widetilde{\mathbf{Z}}$ is obtained as described in (B.10). Finally, the values of the non-zero entries of $\widetilde{\mathbf{X}}$ are determined as described in the beginning of the proof (lines 4-7 of Algorithm 3.5), guaranteeing the optimality of $\widetilde{\mathbf{X}}$ for the maximization in (B.7).

The weighted bipartite graph G is generated in $O(d \cdot (s \cdot k))$. The running time of Algorithm 3.5 is dominated by the computation of the maximum weight matching of G. For the case of unbalanced bipartite graph with $|U| = s \cdot k < d = |V|$ the Hungarian algorithm can be modified [122] to compute the maximum weight bipartite matching in time $O(|E||U| + |U|^2 \log |U|) = O(d \cdot (s \cdot k)^2)$. This completes the proof. □

B.3.2 Guarantees of Algorithm 3.4 – Proof of Theorem 3.3

We first prove a more general version of Theorem 3.3 for arbitrary constraint sets. Combining that with the guarantees of Algorithm 3.5, we prove the Theorem 3.3.

Lemma B.3.26. *For any real $d \times d$ rank-r PSD matrix $\bar{\mathbf{A}}$ and arbitrary set $\mathcal{X} \subset \mathbb{R}^{d \times k}$, let $\overline{\mathbf{X}}_\star \triangleq \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \operatorname{Tr}\left(\mathbf{X}^\top \bar{\mathbf{A}} \mathbf{X}\right)$. Assuming that there exists an operator $P_{\mathcal{X}} : \mathbb{R}^{d \times k} \to \mathcal{X}$ such that $P_{\mathcal{X}}(\mathbf{W}) = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \langle \mathbf{x}_j, \mathbf{w}_j \rangle^2$, then Algorithm 3.4 outputs $\overline{\mathbf{X}} \in \mathcal{X}$ such that*

$$\operatorname{Tr}\left(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\right) \geq (1 - \epsilon) \cdot \operatorname{Tr}\left(\overline{\mathbf{X}}_\star^\top \bar{\mathbf{A}} \overline{\mathbf{X}}_\star\right),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\left(\left(\frac{4}{\epsilon}\right)^{r \cdot k} \cdot (T_{\mathcal{X}} + kd)\right)$, where $T_{\mathcal{X}}$ is the time required to compute $P_{\mathcal{X}}(\cdot)$ and $\mathrm{T}_{\mathsf{SVD}}(r)$ the time required to compute the truncated SVD of $\bar{\mathbf{A}}$.

Proof. Let $\bar{\mathbf{A}} = \overline{\mathbf{U}} \overline{\mathbf{\Lambda}} \overline{\mathbf{U}}^\top$ denote the truncated eigenvalue decomposition of $\bar{\mathbf{A}}$; $\overline{\mathbf{\Lambda}}$ is a diagonal $r \times r$ whose ith diagonal entry Λ_{ii} is equal to the ith largest eigenvalue of $\bar{\mathbf{A}}$, while the columns of $\overline{\mathbf{U}}$ contain the corresponding eigenvec-

tors. By the Cauchy-Schwartz inequality, for any $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbf{x}^\top \bar{\mathbf{A}} \mathbf{x} = \left\| \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \mathbf{x} \right\|_2^2 \geq \left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \mathbf{x},\, \mathbf{c} \right\rangle^2, \quad \forall\, \mathbf{c} \in \mathbb{R}^r : \|\mathbf{c}\|_2 = 1. \tag{B.13}$$

In fact, equality in (B.13) is achieved for $\mathbf{c}$ colinear to $\overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}} \mathbf{x}$, and hence,

$$\mathbf{x}^\top \bar{\mathbf{A}} \mathbf{x} = \max_{\mathbf{c} \in \mathbb{S}_2^{r-1}} \left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \mathbf{x},\, \mathbf{c} \right\rangle^2. \tag{B.14}$$

In turn,

$$\textsc{Tr}\left(\mathbf{X}^\top \bar{\mathbf{A}} \mathbf{X}\right) = \sum_{j=1}^k \mathbf{X}^{j\top} \bar{\mathbf{A}} \mathbf{X}^j = \max_{\mathbf{C}: \mathbf{C}^j \in \mathbb{S}_2^{r-1} \forall j} \sum_{j=1}^k \left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \mathbf{X}^j,\, \mathbf{C}^j \right\rangle^2. \tag{B.15}$$

Recall that $\overline{\mathbf{X}}_\star$ is the optimal solution of the trace maximization on $\bar{\mathbf{A}}$, *i.e.*,

$$\overline{\mathbf{X}}_\star \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}} \textsc{Tr}\left(\mathbf{X}^\top \bar{\mathbf{A}} \mathbf{X}\right).$$

Let $\overline{\mathbf{C}}_\star$ be the maximizing value of $\mathbf{C}$ in (B.15) for $\mathbf{X} = \overline{\mathbf{X}}_\star$, *i.e.*, $\overline{\mathbf{C}}_\star$ is an $r \times k$ matrix with unit-norm columns such that for all $j \in \{1, \ldots, k\}$,

$$\overline{\mathbf{X}}_\star^{j\top} \bar{\mathbf{A}} \overline{\mathbf{X}}_\star^j = \left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \overline{\mathbf{C}}_\star^j \right\rangle^2. \tag{B.16}$$

Algorithm 3.4 iterates over the points ($r \times k$ matrices) $\mathbf{C}$ in $\mathcal{N}_{\epsilon/2}^{\otimes k}\left(\mathbb{S}_2^{r-1}\right)$, the kth cartesian power of a finite $\epsilon/2$-net of the r-dimensional l_2-unit sphere. At each such point $\mathbf{C}$, it computes a candidate

$$\widetilde{\mathbf{X}} = \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}} \sum_{j=1}^k \left\langle \mathbf{X}^j,\, \mathbf{U} \mathbf{\Lambda}^{1/2} \mathbf{C}^j \right\rangle^2$$

via Algorithm 3.5 (See Lemma B.3.1 for the guarantees of Algorithm 3.5). By construction, the set $\mathcal{N}_{\epsilon/2}^{\otimes k}\left(\mathbb{S}_2^{r-1}\right)$ contains a $\mathbf{C}_\sharp$ such that

$$\left\| \mathbf{C}_\sharp - \overline{\mathbf{C}}_\star \right\|_{\infty,2} = \max_{j \in \{1,\ldots,k\}} \left\| \mathbf{C}_\sharp^j - \overline{\mathbf{C}}_\star^j \right\|_2 \leq \epsilon/2. \tag{B.17}$$

Based on the above, for all $j \in \{1, \ldots, k\}$,

$$\begin{aligned}
\left(\overline{\mathbf{X}}_\star^{j\top} \bar{\mathbf{A}} \overline{\mathbf{X}}_\star^j\right)^{1/2} &= \left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \overline{\mathbf{C}}_\star^j \right\rangle\right| \\
&= \left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle + \left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \left(\overline{\mathbf{C}}_\star^j - \mathbf{C}_\sharp^j\right) \right\rangle\right| \\
&\leq \left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle\right| + \left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \left(\overline{\mathbf{C}}_\star^j - \mathbf{C}_\sharp^j\right) \right\rangle\right| \\
&\leq \left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle\right| + \left\|\overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j\right\| \cdot \left\|\overline{\mathbf{C}}_\star^j - \mathbf{C}_\sharp^j\right\| \\
&\leq \left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle\right| + (\epsilon/2) \cdot \left(\overline{\mathbf{X}}_\star^{j\top} \bar{\mathbf{A}} \overline{\mathbf{X}}_\star^j\right)^{1/2}. \qquad \text{(B.18)}
\end{aligned}$$

The first step follows by the definition of $\overline{\mathbf{C}}_\star$, the second by the linearity of the inner product, the third by the triangle inequality, the fourth by Cauchy-Schwarz inequality and the last by (B.17). Rearranging the terms in (B.18),

$$\left|\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle\right| \geq \left(1 - \tfrac{\epsilon}{2}\right) \cdot \left(\overline{\mathbf{X}}_\star^{j\top} \bar{\mathbf{A}} \overline{\mathbf{X}}_\star^j\right)^{1/2} \geq 0,$$

and in turn,

$$\left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle^2 \geq \left(1 - \frac{\epsilon}{2}\right)^2 \cdot \overline{\mathbf{X}}_\star^{j\top} \bar{\mathbf{A}} \overline{\mathbf{X}}_\star^j \geq (1 - \epsilon) \cdot \overline{\mathbf{X}}_\star^{j\top} \bar{\mathbf{A}} \overline{\mathbf{X}}_\star^j \qquad \text{(B.19)}$$

Summing the terms in (B.19) over all $j \in \{1, \ldots, k\}$,

$$\sum_{j=1}^k \left\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\star^j,\, \mathbf{C}_\sharp^j \right\rangle^2 \geq (1 - \epsilon) \cdot \textsc{Tr}\left(\overline{\mathbf{X}}_\star^\top \bar{\mathbf{A}} \overline{\mathbf{X}}_\star\right). \qquad \text{(B.20)}$$

Let $\mathbf{X}_\sharp \in \mathcal{X}$ be the candidate solution produced by the algorithm at $\mathbf{C}_\sharp$, *i.e.*,

$$\mathbf{X}_\sharp \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}} \sum_{j=1}^k \left\langle \mathbf{x}_j,\, \overline{\mathbf{U}} \overline{\mathbf{\Lambda}}^{1/2} \mathbf{C}_\sharp^j \right\rangle^2. \qquad \text{(B.21)}$$

Then,

$$\begin{aligned}
\textsc{Tr}\Big(\mathbf{X}_\sharp^\top \bar{\mathbf{A}} \mathbf{X}_\sharp\Big) &\overset{(\alpha)}{=} \max_{\mathbf{C}:\mathbf{C}^j \in \mathbb{S}_2^{r-1} \forall j} \sum_{j=1}^{k} \Big\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\sharp^j,\, \mathbf{C}^j \Big\rangle^2 \\
&\overset{(\beta)}{\geq} \sum_{j=1}^{k} \Big\langle \overline{\mathbf{\Lambda}}^{1/2} \overline{\mathbf{U}}^\top \overline{\mathbf{X}}_\sharp^j,\, \mathbf{C}_\sharp^j \Big\rangle^2 \\
&\overset{(\gamma)}{\geq} \sum_{j=1}^{k} \Big\langle \overline{\mathbf{X}}_\star^j,\, \overline{\mathbf{U}} \overline{\mathbf{\Lambda}}^{1/2} \mathbf{C}_\sharp^j \Big\rangle^2 \\
&\overset{(\delta)}{\geq} (1-\epsilon) \cdot \textsc{Tr}\Big(\overline{\mathbf{X}}_\star^\top \bar{\mathbf{A}} \overline{\mathbf{X}}_\star\Big), \qquad \text{(B.22)}
\end{aligned}$$

where (α) follows from the observation in (B.15), (β) from the sub-optimality of $\mathbf{C}_\sharp$, (γ) by the definition of $\mathbf{X}_\sharp$ in (B.21), while (δ) follows from (B.20). According to (B.22), at least one of the candidate solutions produced by Algorithm 3.4, namely $\mathbf{X}_\sharp$, achieves an objective value within a multiplicative factor $(1-\epsilon)$ from the optimal, implying the guarantees of the lemma.

Finally, the running time of Algorithm 3.4 follows immediately from the cost per iteration and the cardinality of the $\epsilon/2$-net on the unit-sphere. Note that matrix multiplications can exploit the singular value decomposition which is performed once. □

Theorem 3.3. *For any real $d \times d$ rank-r PSD matrix $\bar{\mathbf{A}}$, desired number of components k, number s of nonzero entries per component, and accuracy parameter $\epsilon \in (0,1)$, Algorithm 3.4 outputs $\overline{\mathbf{X}} \in \mathcal{X}_k$ such that*

$$\textsc{Tr}\Big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\Big) \geq (1-\epsilon) \cdot \textsc{Tr}\Big(\mathbf{X}_\star^\top \bar{\mathbf{A}} \mathbf{X}_\star\Big),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\Big((4/\epsilon)^{r \cdot k} \cdot d \cdot (s \cdot k)^2\Big)$. Here, $\mathbf{X}_\star \in \mathcal{X}_k$ is the point that

maximizes $\textsc{Tr}\left(\mathbf{X}^\top \bar{\mathbf{A}} \mathbf{X}\right)$ *over all* $\mathbf{X} \in \mathcal{X}_k$*, and* $\mathrm{T}_{\mathsf{SVD}}(r)$ *denotes the time required to compute the rank-*r *truncated SVD of* $\bar{\mathbf{A}}$*.*

Proof. Recall that $\mathcal{X}_k$ is the set of $d \times k$ matrices $\mathbf{X}$ whose columns have unit length and pairwise disjoint supports. Algorithm 3.5, given any $\mathbf{W} \in \mathbb{R}^{d \times k}$, computes $\mathbf{X} \in \mathcal{X}_k$ that optimally solves the constrained maximization in line 5 in time $O(d \cdot (s \cdot k)^2)$ (See Lemma B.3.1 for the guarantee of Algorithm 3.5). The desired result then follows by Lemma B.3.26 for the constrained set $\mathcal{X}_k$. □

B.3.3 Guarantees of Algorithm 3.6 – Proof of Theorem 3.4

We prove Theorem 3.4 with the approximation guarantees of Algorithm 3.6.

Lemma B.3.27. *For any* $d \times d$ *PSD matrices* $\mathbf{A}$ *and* $\bar{\mathbf{A}}$*, and any set* $\mathcal{X} \subseteq \mathbb{R}^{d \times k}$ *let*

$$\mathbf{X}_\star \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}} \textsc{Tr}\left(\mathbf{X}^\top \mathbf{A} \mathbf{X}\right), \quad \textit{and} \quad \overline{\mathbf{X}}_\star \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}} \textsc{Tr}\left(\mathbf{X}^\top \bar{\mathbf{A}} \mathbf{X}\right).$$

Then, for any $\overline{\mathbf{X}} \in \mathcal{X}$ *such that* $\textsc{Tr}\left(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\right) \geq \gamma \cdot \textsc{Tr}\left(\overline{\mathbf{X}}_\star^\top \bar{\mathbf{A}} \overline{\mathbf{X}}_\star\right)$ *for some* $0 < \gamma < 1$*,*

$$\textsc{Tr}\left(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\right) \geq \gamma \cdot \textsc{Tr}\left(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\right) - 2 \cdot \|\mathbf{A} - \bar{\mathbf{A}}\|_2 \cdot \max_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\|_{\mathrm{F}}^2.$$

Proof. By the optimality of $\overline{\mathbf{X}}_\star$ for $\bar{\mathbf{A}}$,

$$\textsc{Tr}\left(\overline{\mathbf{X}}_\star^\top \bar{\mathbf{A}} \overline{\mathbf{X}}_\star\right) \geq \textsc{Tr}\left(\mathbf{X}_\star^\top \bar{\mathbf{A}} \mathbf{X}_\star\right).$$

In turn, for any $\overline{\mathbf{X}} \in \mathcal{X}$ such that $\textsc{Tr}\big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\big) \geq \gamma \cdot \textsc{Tr}\big(\overline{\mathbf{X}}_\star^\top \bar{\mathbf{A}} \overline{\mathbf{X}}_\star\big)$ for some $0 < \gamma < 1$,

$$\textsc{Tr}\big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\big) \geq \gamma \cdot \textsc{Tr}\big(\mathbf{X}_\star^\top \bar{\mathbf{A}} \mathbf{X}_\star\big). \tag{B.23}$$

Let $\mathbf{E} \triangleq \mathbf{A} - \bar{\mathbf{A}}$. By the linearity of the trace,

$$\begin{aligned} \textsc{Tr}\big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\big) &= \textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\big) - \textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{E} \overline{\mathbf{X}}\big) \\ &\leq \textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\big) + \big|\textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{E} \overline{\mathbf{X}}\big)\big|. \end{aligned} \tag{B.24}$$

By Lemma B.4.34,

$$\big|\textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{E} \overline{\mathbf{X}}\big)\big| \leq \|\overline{\mathbf{X}}\|_{\mathrm{F}} \cdot \|\overline{\mathbf{X}}\|_{\mathrm{F}} \cdot \|\mathbf{E}\|_2 \leq \|\mathbf{E}\|_2 \cdot \max_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\|_{\mathrm{F}}^2 \triangleq R. \tag{B.25}$$

Continuing from (B.24),

$$\textsc{Tr}\big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\big) \leq \textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\big) + R. \tag{B.26}$$

Similarly,

$$\begin{aligned} \textsc{Tr}\big(\mathbf{X}_\star^\top \bar{\mathbf{A}} \mathbf{X}_\star\big) &= \textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\big) - \textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{E} \mathbf{X}_\star\big) \\ &\geq \textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\big) - \big|\textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{E} \mathbf{X}_\star\big)\big| \\ &\geq \textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\big) - R. \end{aligned} \tag{B.27}$$

Combining the above, we have

$$\begin{aligned} \textsc{Tr}\big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\big) &\geq \textsc{Tr}\big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\big) - R \\ &\geq \gamma \cdot \textsc{Tr}\big(\mathbf{X}_\star^\top \bar{\mathbf{A}} \mathbf{X}_\star\big) - R \\ &\geq \gamma \cdot \Big(\textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\big) - R\Big) - R \\ &= \gamma \cdot \textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\big) - (1 + \gamma) \cdot R \\ &\geq \gamma \cdot \textsc{Tr}\big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\big) - 2 \cdot R, \end{aligned}$$

where the first inequality follows from (B.26) the second from (B.23), the third from (B.27), and the last from the fact that $R \geq 0$ and $0 < \gamma \leq 1$. This concludes the proof. □

Remark B.3.2. *If in Lemma B.3.27 the PSD matrices* $\mathbf{A}$ *and* $\bar{\mathbf{A}} \in \mathbb{R}^{d \times d}$ *are such that* $\mathbf{A} - \bar{\mathbf{A}}$ *is also PSD, then the following tighter bound holds:*

$$\textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\Big) \geq \gamma \cdot \textsc{Tr}\Big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\Big) - \sum_{i=1}^{k} \lambda_i\Big(\mathbf{A} - \bar{\mathbf{A}}\Big).$$

Proof. This follows from the fact that if $\mathbf{E} \triangleq \mathbf{A} - \bar{\mathbf{A}}$ is PSD, then

$$\textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{E} \overline{\mathbf{X}}\Big) = \sum_{j=1}^{d} \mathbf{x}_j^\top \mathbf{E} \mathbf{x}_j \geq 0,$$

and the bound in (B.24) can be improved to

$$\textsc{Tr}\Big(\overline{\mathbf{X}}^\top \bar{\mathbf{A}} \overline{\mathbf{X}}\Big) = \textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\Big) - \textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{E} \overline{\mathbf{X}}\Big) \leq \textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{A} \overline{\mathbf{X}}\Big).$$

Further, by Lemma B.4.35, the bound in (B.25) can be improved to

$$\textsc{Tr}\Big(\overline{\mathbf{X}}^\top \mathbf{E} \overline{\mathbf{X}}\Big) \leq \sum_{i=1}^{k} \lambda_i(\mathbf{E}) \triangleq R.$$

The rest of the proof follows as is. □

Theorem 3.4. *For any* $n \times d$ *input data matrix* $\mathbf{S}$*, with corresponding empirical covariance matrix* $\mathbf{A} = {}^1\!/_n \cdot \mathbf{S}^\top \mathbf{S}$*, any desired number of components* k*, and accuracy parameters* $\epsilon \in (0, 1)$ *and* r*, Algorithm 3.6 outputs* $\mathbf{X}_{(r)} \in \mathcal{X}_k$ *such that*

$$\textsc{Tr}\Big(\mathbf{X}_{(r)}^\top \mathbf{A} \mathbf{X}_{(r)}\Big) \;\geq\; (1 - \epsilon) \cdot \textsc{Tr}\Big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{X}_\star\Big) - 2 \cdot k \cdot \|\mathbf{A} - \bar{\mathbf{A}}\|_2,$$

in time $\mathrm{T}_{\textsc{sketch}}(r) + \mathrm{T}_{\textsc{svd}}(r) + O\Big((4/\epsilon)^{r \cdot k} \cdot d \cdot (s \cdot k)^2\Big)$*. Here,* $\mathbf{X}_\star$ *is a feasible point that maximizes* $\textsc{Tr}\Big(\mathbf{X}^\top \mathbf{A} \mathbf{X}\Big)$ *over all* $\mathbf{X} \in \mathcal{X}_k$*.*

Proof. The theorem follows from Lemma B.3.27 and the approximation guarantees of Algorithm 3.4. □

B.3.4 Proof of Theorem 3.5

First, we restate and prove the following lemma by [6].

Lemma B.3.28. *Let $\mathbf{A} \in \mathbb{R}^{d\times d}$ be an positive semidefinite matrix with entries in $[-1,1]$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$ such that $\mathbf{A} = \mathbf{V}\mathbf{V}^\top$. Further, let $\mathbf{R} \in \mathbb{R}^{d\times r}$ be a random matrix with entries drawn independently according to a Gaussian distribution $\mathcal{N}(0, 1/r)$, and define*

$$\bar{\mathbf{A}} = \mathbf{V}\mathbf{R}\mathbf{R}^\top\mathbf{V}^\top.$$

Then, for $r = O(\epsilon^{-2}\log d)$, it holds that

$$\left|A_{ij} - \bar{A}_{ij}\right| \leq \epsilon$$

for all i, j with probability at least $1 - poly(1/d)$.

Proof. The proof relies on the Johnson-Lindenstrauss (JL) Lemma [47], according to which for any two unit norm vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\mathbf{R}$ generated as described in the lemma,

$$\Pr\left(\left|\mathbf{x}^\top\mathbf{R}\mathbf{R}^\top\mathbf{y} - \mathbf{x}^\top\mathbf{y}\right| \geq \epsilon\right) \leq 2\cdot e^{-(\epsilon^2-\epsilon^3)\cdot r/4}.$$

By the definition of $\mathbf{V}$, we have

$$A_{ij} = \langle \mathbf{V}_{i:}, \mathbf{V}_{j:}\rangle, \quad \forall i,j \in [d]\times[d]. \tag{B.28}$$

Recall that by assumption $|A_{ij}| \leq 1$, $\forall i, j \in [d] \times [d]$. In particular, the bound holds for the diagonal entries of $\mathbf{A}$ which in turn implies that $\|\mathbf{V}_{i:}\|_2^2 \leq 1$, for all $i = 1, \ldots, d$. Setting $r = O(\epsilon^{-2} \log d)$ and using the JL lemma and a union bound over all $O(d^2)$ vector pairs $\mathbf{V}_{i:}$, $\mathbf{V}_{j:}$ we obtain the desired result. $\square$

Next, we provide the proof of Theorem 3.5 for the simple case of $k = 1$; the proof easily generalizes to the multi-component case $k > 1$. According to Lemma B.3.28, choosing $d = O((\delta/6)^{-2} \log n) = O(\delta^{-2} \log n)$ suffices for all entries of $\bar{\mathbf{A}}$ constructed as described in the lemma to satisfiy

$$\left|A_{ij} - \bar{A}_{ij}\right| \leq \frac{\delta}{6}$$

with probability at least $1 - 1/d$. In turn, for any s-sparse, unit ℓ_2-norm vector $\mathbf{x} \in \mathbb{R}^d$, we have

$$\begin{aligned}
\left|\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{x}^\top \bar{\mathbf{A}}\mathbf{x}\right| &= \left|\sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j \left(A_{ij} - \bar{A}_{ij}\right)\right| \\
&\leq \frac{\delta}{6} \cdot \left(\sum_{i=1}^{n} |x_i|\right) \cdot \left(\sum_{j=1}^{n} |x_j|\right) \\
&\leq \frac{\delta}{6} \cdot \|\mathbf{x}\|_1^2 \\
&\leq \frac{\delta}{6} \cdot \left(\sqrt{s} \cdot \|\mathbf{x}\|_2\right)^2 = \frac{\delta}{6} \cdot s, \qquad \text{(B.29)}
\end{aligned}$$

where the lst inequality follows from the fact that $\mathbf{x}$ is s-sparse has unit ℓ_2-norm.

We run Algorithm 3.4 (for $k = 1$) with input argument the rank-r matrix $\bar{\mathbf{A}}$, desired sparsity s and accuracy parameter $\epsilon = \delta/6$. Algorithm 3.4

outputs an s-sparse unit ℓ_2-norm vector $\widehat{\mathbf{x}}$ which according to Theorem 3.3 satisfies

$$(1-\delta/6)\cdot \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d \leq \widehat{\mathbf{x}}^\top \bar{\mathbf{A}}\widehat{\mathbf{x}} \leq \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d, \tag{B.30}$$

where $\mathbf{x}_d$ is the true s-sparse principal component of $\bar{\mathbf{A}}$. In turn,

$$\left|\widehat{\mathbf{x}}^\top \bar{\mathbf{A}}\widehat{\mathbf{x}} - \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d\right| \leq \frac{\delta}{6}\cdot \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d \leq \frac{\delta}{6}\cdot\left(1+\frac{\delta}{6}\right)\cdot s \leq \frac{\delta}{3}\cdot s, \tag{B.31}$$

where the second inequality follows from the fact that the entries of $\bar{\mathbf{A}}$ lie in the interval $[-1-\delta/6,\, 1+\delta/6]$, and $\widehat{\mathbf{x}}$ is an s-sparse vector with unit ℓ_2-norm.

Of course, we are interested in how the quadratic value achieved by $\widehat{\mathbf{x}}$ compares to the optimal on the original matrix $\mathbf{A}$ rather than the rank-r approximation $\bar{\mathbf{A}}$. Let $\mathbf{x}_\star$ denote the s-sparse principal component of $\mathbf{A}$ and define $\mathrm{OPT} \triangleq \mathbf{x}_\star^\top \mathbf{A}\mathbf{x}_\star$. Then,

$$\begin{aligned}\left|\mathrm{OPT} - \widehat{\mathbf{x}}^\top \mathbf{A}\widehat{\mathbf{x}}\right| &= \left|\mathrm{OPT} - \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d + \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d - \widehat{\mathbf{x}}^\top \mathbf{A}\widehat{\mathbf{x}}\right| \\ &\leq \underbrace{\left|\mathrm{OPT} - \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d\right|}_{D_1} + \underbrace{\left|\mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d - \widehat{\mathbf{x}}^\top \mathbf{A}\widehat{\mathbf{x}}\right|}_{D_2}.\end{aligned} \tag{B.32}$$

For the first quanity on the RHS of (B.32), we have

$$\begin{aligned}D_1 &= \left|\mathrm{OPT} - \mathbf{x}_d^\top \mathbf{A}\mathbf{x}_d \;+\; \mathbf{x}_d^\top \mathbf{A}\mathbf{x}_d - \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d\right| \\ &\leq \left|\mathrm{OPT} - \mathbf{x}_d^\top \mathbf{A}\mathbf{x}_d\right| + \left|\mathbf{x}_d^\top \mathbf{A}\mathbf{x}_d - \mathbf{x}_d^\top \bar{\mathbf{A}}\mathbf{x}_d\right| \\ &\leq \mathrm{OPT} - \mathbf{x}_d^\top \mathbf{A}\mathbf{x}_d + \frac{\delta}{6}\cdot s,\end{aligned} \tag{B.33}$$

where the last inequality follows from the fact that $\mathrm{OPT} \geq \mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d$ and inequality (B.29). Continuing from (B.33),

$$\begin{aligned} D_1 &\leq \mathrm{OPT} - \mathbf{x}_d^\top \mathbf{A} \mathbf{x}_d + \frac{\delta}{6} \cdot s + \underbrace{\mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d - \mathbf{x}_\star^\top \bar{\mathbf{A}} \mathbf{x}_\star}_{\geq 0} \\ &= \mathbf{x}_\star^\top \mathbf{A} \mathbf{x}_\star - \mathbf{x}_\star^\top \bar{\mathbf{A}} \mathbf{x}_\star + \mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d - \mathbf{x}_d^\top \mathbf{A} \mathbf{x}_d + \frac{\delta}{6} \cdot s \\ &\leq \left| \mathbf{x}_\star^\top \mathbf{A} \mathbf{x}_\star - \mathbf{x}_\star^\top \bar{\mathbf{A}} \mathbf{x}_\star \right| + \left| \mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d - \mathbf{x}_d^\top \mathbf{A} \mathbf{x}_d \right| + \frac{\delta}{6} \cdot s \\ &\leq \frac{\delta}{2} \cdot s, \end{aligned} \tag{B.34}$$

where the last inequality follows by applying (B.29) twice. Similarly, for the second quanity on the RHS of (B.32), we have

$$\begin{aligned} D_2 &= \left| \mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d - \widehat{\mathbf{x}}^\top \bar{\mathbf{A}} \widehat{\mathbf{x}} + \widehat{\mathbf{x}}^\top \bar{\mathbf{A}} \widehat{\mathbf{x}} - \widehat{\mathbf{x}}^\top \mathbf{A} \widehat{\mathbf{x}} \right| \\ &\leq \left| \mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d - \widehat{\mathbf{x}}^\top \bar{\mathbf{A}} \widehat{\mathbf{x}} \right| + \left| \widehat{\mathbf{x}}^\top \bar{\mathbf{A}} \widehat{\mathbf{x}} - \widehat{\mathbf{x}}^\top \mathbf{A} \widehat{\mathbf{x}} \right| \\ &\leq \left| \mathbf{x}_d^\top \bar{\mathbf{A}} \mathbf{x}_d - \widehat{\mathbf{x}}^\top \bar{\mathbf{A}} \widehat{\mathbf{x}} \right| + \frac{\delta}{6} \cdot s \qquad \text{(B.35)} \\ &\leq \frac{2\delta}{6} \cdot s + \frac{\delta}{6} \cdot s \qquad \text{(B.36)} \\ &\leq \frac{\delta}{2} \cdot s. \qquad \text{(B.37)} \end{aligned}$$

where inequality (B.35) follows from (B.29) since $\widehat{\mathbf{x}}$ is an s-sparse unit ℓ_2-norm vector, and inequality (B.36) from (B.31). Continuing from (B.32), and combining (B.34) with (B.37), we find

$$\left| \mathrm{OPT} - \widehat{\mathbf{x}}^\top \mathbf{A} \widehat{\mathbf{x}} \right| \leq \frac{\delta}{2} \cdot s + \frac{\delta}{2} \cdot s = \delta \cdot s,$$

which is the desired result. This completes the proof of Theorem 3.5.

B.4 Auxiliary Lemmas

Lemma B.4.29. *For any real $d \times n$ matrix $\mathbf{M}$, and any $r, k \leq \min\{d, n\}$,*

$$\sum_{i=r+1}^{r+k} \sigma_i(\mathbf{M}) \leq \frac{k}{\sqrt{r+k}} \cdot \|\mathbf{M}\|_{\mathrm{F}},$$

where $\sigma_i(\mathbf{M})$ is the ith largest singular value of $\mathbf{M}$.

Proof. By the Cauchy-Schwartz inequality,

$$\sum_{i=r+1}^{r+k} \sigma_i(\mathbf{M}) = \sum_{i=r+1}^{r+k} |\sigma_i(\mathbf{M})| \leq \left(\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}) \right)^{1/2} \cdot \|\mathbf{1}_k\|_2 = \sqrt{k} \cdot \left(\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}) \right)^{1/2}.$$

Note that $\sigma_{r+1}(\mathbf{M}), \ldots, \sigma_{r+k}(\mathbf{M})$ are the k smallest among the $r+k$ largest singular values. Hence,

$$\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}) \leq \frac{k}{r+k} \sum_{i=1}^{r+k} \sigma_i^2(\mathbf{M}) \leq \frac{k}{r+k} \sum_{i=1}^{\min\{d,n\}} \sigma_i^2(\mathbf{M}) = \frac{k}{r+k} \|\mathbf{M}\|_{\mathrm{F}}^2.$$

Combining the two inequalities, the desired result follows. □

Corollary 2. *For any real $d \times n$ matrix $\mathbf{M}$ and $k \leq \min\{d, n\}$, $\sigma_k(\mathbf{M}) \leq k^{-1/2} \cdot \|\mathbf{M}\|_{\mathrm{F}}$.*

Proof. It follows immediately from Lemma B.4.29. □

Lemma B.4.30. *Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be $2n$ real numbers and let p and q be two numbers such that $1/p + 1/q = 1$ and $p > 1$. We have*

$$\left| \sum_{i=1}^{n} a_i b_i \right| \leq \left(\sum_{i=1}^{n} |a_i|^p \right)^{1/p} \cdot \left(\sum_{i=1}^{n} |b_i|^q \right)^{1/q}.$$

Lemma B.4.31. *For any two real matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *of appropriate dimensions,*

$$\|\mathbf{AB}\|_{\mathrm{F}} \leq \min\{\|\mathbf{A}\|_2\|\mathbf{B}\|_{\mathrm{F}},\ \|\mathbf{A}\|_{\mathrm{F}}\|\mathbf{B}\|_2\}.$$

Proof. Let $\mathbf{b}_i$ denote the ith column of $\mathbf{B}$. Then,

$$\|\mathbf{AB}\|_{\mathrm{F}}^2 = \sum_i \|\mathbf{Ab}_i\|_2^2 \leq \sum_i \|\mathbf{A}\|_2^2\|\mathbf{b}_i\|_2^2 = \|\mathbf{A}\|_2^2 \sum_i \|\mathbf{b}_i\|_2^2 = \|\mathbf{A}\|_2^2\|\mathbf{B}\|_{\mathrm{F}}^2.$$

Similarly, using the previous inequality,

$$\|\mathbf{AB}\|_{\mathrm{F}}^2 = \|\mathbf{B}^\top\mathbf{A}^\top\|_{\mathrm{F}}^2 \leq \|\mathbf{B}^\top\|_2^2\|\mathbf{A}^\top\|_{\mathrm{F}}^2 = \|\mathbf{B}\|_2^2\|\mathbf{A}\|_{\mathrm{F}}^2.$$

Combining the two upper bounds, the desired result follows. □

Lemma B.4.32. *For any* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times k}$,

$$|\langle \mathbf{A}, \mathbf{B}\rangle| \triangleq \left|\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{B}\right)\right| \leq \|\mathbf{A}\|_{\mathrm{F}}\|\mathbf{B}\|_{\mathrm{F}}.$$

Proof. The inequality follows from Lemma B.4.30 for $p = q = 2$, treating $\mathbf{A}$ and $\mathbf{B}$ as vectors. □

Lemma B.4.33. *For any real* $m \times n$ *matrix* $\mathbf{A}$*, and any* $k \leq \min\{m,\ n\}$,

$$\max_{\substack{\mathbf{Y}\in\mathbb{R}^{n\times k}\\ \mathbf{Y}^\top\mathbf{Y}=\mathbf{I}_k}} \|\mathbf{AY}\|_{\mathrm{F}} = \left(\sum_{i=1}^{k}\sigma_i^2(\mathbf{A})\right)^{1/2}.$$

The maximum is attained by $\mathbf{Y}$ *coinciding with the* k *leading right singular vectors of* $\mathbf{A}$.

Proof. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the singular value decomposition of $\mathbf{A}$; $\mathbf{U}$ and $\mathbf{V}$ are $m \times m$ and $n \times n$ unitary matrices respectively, while Σ is a diagonal matrix with $\Sigma_{jj} = \sigma_j$, the jth largest singular value of $\mathbf{A}$, $j = 1, \ldots, d$, where $d \triangleq \min\{m, n\}$. Due to the invariance of the Frobenius norm under unitary multiplication,

$$\|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}^2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Y}\|_{\mathrm{F}}^2 = \|\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Y}\|_{\mathrm{F}}^2. \tag{B.38}$$

Continuing from (B.38),

$$\begin{aligned}\|\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Y}\|_{\mathrm{F}}^2 = \mathrm{Tr}\Big(\mathbf{Y}^\top\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top\mathbf{Y}\Big) &= \sum_{i=1}^{k}\mathbf{y}_i^\top\left(\sum_{j=1}^{d}\sigma_j^2\cdot\mathbf{v}_j\mathbf{v}_j^\top\right)\mathbf{y}_i \\ &= \sum_{j=1}^{d}\sigma_j^2\cdot\sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2.\end{aligned}$$

Let $z_j \triangleq \sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2$, $j = 1, \ldots, d$. Note that each individual z_j satisfies

$$0 \le z_j \triangleq \sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2 \le \|\mathbf{v}_j\|^2 = 1,$$

where the last inequality follows from the fact that the columns of $\mathbf{Y}$ are orthonormal. Further,

$$\sum_{j=1}^{d} z_j = \sum_{j=1}^{d}\sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2 = \sum_{i=1}^{k}\sum_{j=1}^{d}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2 = \sum_{i=1}^{k}\|\mathbf{y}_i\|^2 = k.$$

Combining the above, we conclude that

$$\|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}^2 = \sum_{j=1}^{d}\sigma_j^2\cdot z_j \le \sigma_1^2 + \ldots + \sigma_k^2. \tag{B.39}$$

Finally, it is straightforward to verify that if $\mathbf{y}_i = \mathbf{v}_i$, $i = 1, \ldots, k$, then (B.39) holds with equality. □

Lemma B.4.34. *For any real $d \times n$ matrix $\mathbf{A}$, and pair of $d \times k$ matrix $\mathbf{X}$ and $n \times k$ matrix $\mathbf{Y}$ such that $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_k$ and $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k$ with $k \leq \min\{d,\ n\}$, the following holds:*

$$\left|\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{Y}\right)\right| \leq \sqrt{k} \cdot \left(\sum_{i=1}^{k} \sigma_i^2(\mathbf{A})\right)^{1/2}.$$

Proof. By Lemma B.4.32,

$$|\langle \mathbf{X},\ \mathbf{A}\mathbf{Y}\rangle| = \left|\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{Y}\right)\right| \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}} = \sqrt{k} \cdot \|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}.$$

where the last inequality follows from the fact that $\|\mathbf{X}\|_{\mathrm{F}}^2 = \textsc{Tr}\left(\mathbf{X}^\top \mathbf{X}\right) = \textsc{Tr}(\mathbf{I}_k) = k$. Combining with a bound on $\|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}$ as in Lemma B.4.33, completes the proof. □

Lemma B.4.35. *For any real $d \times d$ PSD matrix $\mathbf{A}$, and $k \times d$ matrix $\mathbf{X}$ with $k \leq d$ orthonormal columns,*

$$\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{X}\right) \leq \sum_{i=1}^{k} \lambda_i(\mathbf{A})$$

where $\lambda_i(\mathbf{A})$ is the ith largest eigenvalue of $\mathbf{A}$. Equality is achieved for $\mathbf{X}$ coinciding with the k leading eigenvectors of $\mathbf{A}$.

Proof. Let $\mathbf{A} = \mathbf{V}\mathbf{V}^\top$ be a factorization of the PSD matrix $\mathbf{A}$. Then, $\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{X}\right) = \textsc{Tr}\left(\mathbf{X}^\top \mathbf{V}\mathbf{V}^\top \mathbf{X}\right) = \|\mathbf{V}^\top \mathbf{X}\|_{\mathrm{F}}^2$. The desired result follows by Lemma B.4.33 and the fact that $\lambda_i(\mathbf{A}) = \sigma_i^2(\mathbf{V})$, $i = 1, \ldots, d$. □

Appendix C

Appendix for Chapter 4

C.1 Proofs

C.1.1 On the connection of ONMF with NNPCA

Lemma C.1.36. *Let* $\mathcal{E}_\star \triangleq \min_{\mathbf{W}\geq\mathbf{0}\mathbf{H}\geq\mathbf{0},\mathbf{W}^\top\mathbf{W}=\mathbf{I}_k} \|\mathbf{M}-\mathbf{W}\mathbf{H}^\top]\|_{\mathrm{F}}^2$ *be the optimal ONMF approximation error for a given* $m\times n$ *real nonnegative matrix* $\mathbf{M}$ *and target dimension* k*, as defined in* (4.1)*. Then,*

$$\mathcal{E}_\star = \|\mathbf{M}\|_{\mathrm{F}}^2 - \max_{\substack{\mathbf{W}\geq\mathbf{0}_{m\times k}\\ \mathbf{W}^\top\mathbf{W}=\mathbf{I}_k}} \|\mathbf{M}^\top\mathbf{W}\|_{\mathrm{F}}^2, \tag{C.1}$$

If $\mathbf{W}_\star$ *is a solution of the maximization in* (C.1)*, then the pair* $\mathbf{W}_\star, \mathbf{H}_\star \triangleq \mathbf{M}^\top\mathbf{W}_\star$ *is a feasible solution to the ONMF problem in* (4.1)*, achieving the minimum error* $\mathcal{E}_\star$*, i.e.,* $\|\mathbf{M}-\mathbf{W}_\star\mathbf{W}_\star^\top\mathbf{M}\|_{\mathrm{F}}^2 = \mathcal{E}_\star$*.*

Proof. Recall that by assumption, $\mathbf{W}$ is a $m\times k$ nonnegative matrix with orthonormal columns. The subsequent analysis holds even in the case where $\mathbf{W}$ is allowed to contain all-zero columns, as such columns do not contribute to the objective function and can be ignored effectively reducing the dimension k of the factorization.

Given a real nonnegative $m\times k$ matrix $\mathbf{W}$, the real $n\times k$ matrix $\mathbf{H}$ that minimizes the Frobenius error $\|\mathbf{M}-\mathbf{W}\mathbf{H}^\top\|_{\mathrm{F}}^2$ over all real $n\times k$ matrices

(ignoring temporarily the fact what we seek a nonnegative $\mathbf{H}$), is given by $\mathbf{H}^\top = \mathbf{W}^\dagger\mathbf{M}$, where $\mathbf{W}^\dagger$ denotes the pseudo-inverse of $\mathbf{W}$. Here, however, the columns of $\mathbf{W}$ are orthonormal and hence $\mathbf{W}^\dagger = \mathbf{W}^\top$. Moreover, since $\mathbf{M}$ is nonnegative, $\mathbf{H}^\top = \mathbf{W}_\star{}^\dagger\mathbf{M} = \mathbf{W}_\star^\top\mathbf{M}$ automatically satisfies the additional nonnegativity constraint. Therefore, the ONMF problem (defined in (4.1)) reduces to a minimization in a single variable:

$$\mathcal{E}_\star = \min_{\substack{\mathbf{W}\geq\mathbf{0}_{m\times k}\\ \mathbf{W}^\top\mathbf{W}=\mathbf{I}_k}} \|\mathbf{M} - \mathbf{W}\mathbf{W}^\top\mathbf{M}\|_{\mathrm{F}}^2. \tag{C.2}$$

Expanding the objective in (C.2),

$$\begin{aligned}\|\mathbf{M} - \mathbf{W}\mathbf{W}^\top\mathbf{M}\|_{\mathrm{F}}^2 &= \|\mathbf{M}\|_{\mathrm{F}}^2 - 2\cdot\mathrm{Tr}\Big(\mathbf{W}^\top\mathbf{M}\mathbf{M}^\top\mathbf{W}\Big) + \mathrm{Tr}\Big(\mathbf{M}^\top\mathbf{W}\mathbf{W}^\top\mathbf{M}\Big)\\ &= \|\mathbf{M}\|_{\mathrm{F}}^2 - \mathrm{Tr}\Big(\mathbf{W}^\top\mathbf{M}\mathbf{M}^\top\mathbf{W}\Big)\\ &= \|\mathbf{M}\|_{\mathrm{F}}^2 - \|\mathbf{M}^\top\mathbf{W}\|_{\mathrm{F}}^2, \end{aligned}\tag{C.3}$$

where the first step follows from the cyclic property of the trace and the fact that $\mathbf{W}^\top\mathbf{W} = \mathbf{I}_k$. This concludes the proof. □

C.1.2 Proof of Lemma 4.3.2

Lemma 4.3.2. *For any real $m \times n$ matrix $\overline{\mathbf{M}}$ with rank r, desired number of components k, and accuracy parameter $\epsilon \in (0,1)$, Algorithm 4.7 outputs $\overline{\mathbf{W}} \in \mathcal{W}_k$ such that*

$$\|\overline{\mathbf{M}}^\top\overline{\mathbf{W}}\|_{\mathrm{F}}^2 \;\geq\; (1-\epsilon)\cdot\|\overline{\mathbf{M}}^\top\overline{\mathbf{W}}_\star\|_{\mathrm{F}}^2,$$

where $\overline{\mathbf{W}}_\star$ is the optimal solution defined in (4.3), in time $\mathrm{T}_{\mathsf{SVD}} + O\Big(\big(\frac{2}{\epsilon}\big)^{r\cdot k}\cdot k\cdot m\Big)$.

Proof. Let $\overline{\mathbf{M}} = \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\overline{\mathbf{V}}^\top$ denote the truncated eigenvalue decomposition of $\overline{\mathbf{M}}$; $\overline{\mathbf{\Sigma}}$ is a diagonal $r \times r$ matrix with Σ_{ii} being equal to the ith largest singular value of $\overline{\mathbf{M}}$. For any $\mathbf{w} \in \mathbb{R}^m$,

$$\begin{aligned}\left\|\overline{\mathbf{M}}^\top \mathbf{w}\right\|_2^2 &= \left\|\overline{\mathbf{V}}\overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{w}\right\|_2^2 \\ &= \left\|\overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{w}\right\|_2^2 \geq \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{w},\, \mathbf{c}\right\rangle^2, \quad \forall\, \mathbf{c} \in \mathbb{R}^r : \|\mathbf{c}\|_2 = 1, \end{aligned} \tag{C.4}$$

where the first equality follows from the fact that the columns of $\overline{\mathbf{V}}$ are orthonormal and span the entire row space of $\overline{\mathbf{M}}$, and the inequality is due to Cauchy-Schwartz. In fact, equality is achieved for $\mathbf{c}$ colinear to $\overline{\mathbf{\Sigma}}\overline{\mathbf{U}}\mathbf{w}$, appropriately scaled to unit-length, and hence,

$$\left\|\overline{\mathbf{M}}^\top \mathbf{w}\right\|_2^2 = \max_{\mathbf{c} \in \mathbb{S}^{r-1}} \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{w},\, \mathbf{c}\right\rangle^2. \tag{C.5}$$

In turn,

$$\left\|\overline{\mathbf{M}}^\top \mathbf{W}\right\|_{\mathrm{F}}^2 = \sum_{j=1}^{k} \left\|\overline{\mathbf{M}}^\top \mathbf{w}_j\right\|_2^2 = \sum_{j=1}^{k} \max_{\mathbf{c}_j \in \mathbb{S}^{r-1}} \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \mathbf{w}_j,\, \mathbf{c}_j\right\rangle^2. \tag{C.6}$$

Recall that $\overline{\mathbf{W}}_\star$ by definition maximizes the left hand side of (C.6) over all $\mathbf{W} \in \mathcal{W}_k$. Let $\widetilde{\mathbf{c}}_{\star 1}, \ldots, \widetilde{\mathbf{c}}_{\star k} \in \mathbb{S}^{r-1}$ be the set of k vectors achieving equality in (C.6) for $\mathbf{W} = \overline{\mathbf{W}}_\star$, and let $\widetilde{\mathbf{C}}_\star \in \mathbb{R}^{r \times k}$ be the matrix formed by stacking the k vectors. Algorithm 4.7 iterates over a set $\mathcal{N}_{\epsilon/2}^{\otimes k}(\mathbb{S}^{r-1})$ of points ($r \times k$ matrices) $\mathbf{C}$. Recall that $\mathcal{N}_{\epsilon/2}^{\otimes k}(\mathbb{S}^{r-1})$ is the kth cartesian power of an $\epsilon/2$-net of the r-dimensional ℓ_2-unit sphere. By construction, the set contains a matrix $\mathbf{C}_\sharp$ such that

$$\|\mathbf{C}_\sharp - \widetilde{\mathbf{C}}_{\star j}\|_{\infty,2} \leq \epsilon/2. \tag{C.7}$$

Then, for all $j \in \{1, \ldots, k\}$,

$$\begin{aligned}
\left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}_{\star j}\right\|_2 &= \left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \widetilde{\mathbf{c}}_{\star j}\right\rangle\right| \\
&= \left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle + \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, (\widetilde{\mathbf{c}}_{\star j} - \mathbf{c}_{\sharp j})\right\rangle\right| \\
&\leq \left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle\right| + \left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, (\widetilde{\mathbf{c}}_{\star j} - \mathbf{c}_{\sharp j})\right\rangle\right| \\
&\leq \left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle\right| + \left\|\overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j}\right\|_2 \cdot \|\widetilde{\mathbf{c}}_{\star j} - \mathbf{c}_{\sharp j}\|_2 \\
&\leq \left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle\right| + (\epsilon/2) \cdot \left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}_{\star j}\right\|_2. \qquad \text{(C.8)}
\end{aligned}$$

The first step follows by the definition of $\widetilde{\mathbf{C}}_\star$, the second by the linearity of the inner product, the third by the triangle inequality, the fourth by Cauchy-Schwarz inequality and the last by the fact that $\|\widetilde{\mathbf{c}}_{\star j} - \mathbf{c}_{\sharp j}\| \leq \epsilon/2$, $\forall i \in \{1, \ldots, k\}$ (by (C.7)). Rearranging the terms in (C.8),

$$\left|\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle\right| \geq \left(1 - \tfrac{\epsilon}{2}\right) \cdot \left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}_{\star j}\right\|_2 \geq 0,$$

which in turn implies (by taking the square on both sides) that

$$\left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle^2 \geq \left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}_{\star j}\right\|_2^2 \geq (1 - \epsilon) \cdot \left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}_{\star j}\right\|_2^2 \qquad \text{(C.9)}$$

Summing the terms in (C.9) over all $j \in \{1, \ldots, k\}$,

$$\sum_{j=1}^{k} \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\star j},\, \mathbf{c}_{\sharp j}\right\rangle^2 \geq (1 - \epsilon) \cdot \left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}_{\star}\right\|_{\mathrm{F}}^2. \qquad \text{(C.10)}$$

Let $\mathbf{W}_\sharp \in \mathcal{W}_k$ be the candidate solution produced by the algorithm at $\mathbf{C}_\sharp$, *i.e.*,

$$\mathbf{W}_\sharp \triangleq \operatorname*{argmax}_{\mathbf{W} \in \mathcal{W}_k} \sum_{j=1}^{k} \left\langle \mathbf{w}_j,\, \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\mathbf{c}_{\sharp j}\right\rangle^2 \qquad \text{(C.11)}$$

Then,

$$\begin{aligned}\left\|\overline{\mathbf{M}}^\top \mathbf{W}_\sharp\right\| &\overset{(\alpha)}{=} \sum_{j=1}^{k} \max_{\mathbf{c}_j \in \mathbb{S}^{r-1}} \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\sharp j},\, \mathbf{c}_j \right\rangle^2 \\ &\overset{(\beta)}{\geq} \sum_{j=1}^{k} \left\langle \overline{\mathbf{\Sigma}}\overline{\mathbf{U}}^\top \overline{\mathbf{W}}_{\sharp j},\, \mathbf{c}_{\sharp j} \right\rangle^2 \\ &\overset{(\gamma)}{\geq} \sum_{j=1}^{k} \left\langle \overline{\mathbf{W}}_{\star j},\, \overline{\mathbf{U}}\overline{\mathbf{\Sigma}}\mathbf{c}_{\sharp j} \right\rangle^2 \\ &\overset{(\delta)}{\geq} (1-\epsilon) \cdot \left\|\overline{\mathbf{M}}\overline{\mathbf{W}}_\star\right\|_{\mathrm{F}}^2, \end{aligned} \tag{C.12}$$

where (α) follows from the observation in (C.6), (β) from the suboptimality of $\mathbf{C}_\sharp$, (γ) from the fact that $\mathbf{W}_\sharp$ maximizes the sum by its definition in (C.11), while (δ) follows from (C.10). According to (C.12), at least one of the candidate solutions produced by Algorithm 4.7, namely $\mathbf{W}_\sharp$, achieves an objective value within a multiplicative factor $(1-\epsilon)$ from the optimal, implying the guarantees of the lemma.

Finally, the running time of Algorithm 4.7 follows immediately from the cost per iteration and the cardinality of the $\epsilon/2$-net on the unit-sphere. Note that matrix multiplications can exploit the available singular value decomposition which is performed once. □

C.1.3 Proof of Theorem 4.6

We first prove some auxiliary lemmata. The proof of the Theorem is given in the end of this section.

Lemma C.1.37. *For any real $m \times n$ matrices $\mathbf{M}$ and $\overline{\mathbf{M}}$, let*

$$\mathbf{W}_\star \triangleq \operatorname*{argmax}_{\mathbf{W} \in \mathcal{W}_k} \left\| \mathbf{M}^\top \mathbf{W} \right\|_{\mathrm{F}}^2 \quad \text{and} \quad \overline{\mathbf{W}}_\star \triangleq \operatorname*{argmax}_{\mathbf{W} \in \mathcal{W}_k} \left\| \overline{\mathbf{M}}^\top \mathbf{W} \right\|_{\mathrm{F}}^2, \tag{C.13}$$

respectively. Then, for any $\overline{\mathbf{W}} \in \mathcal{W}_k$ such that $\left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}} \right\|_{\mathrm{F}}^2 \geq \gamma \cdot \left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}}_\star \right\|_{\mathrm{F}}^2$ for some $0 < \gamma < 1$,

$$\left\| \mathbf{M}^\top \overline{\mathbf{W}} \right\|_{\mathrm{F}}^2 \geq \gamma \cdot \left\| \mathbf{M}^\top \mathbf{W}_\star \right\|_{\mathrm{F}}^2 - 2 \cdot k \cdot \|\mathbf{M} - \overline{\mathbf{M}}\|_2^2.$$

Proof. By the optimality of $\overline{\mathbf{W}}_\star$ for $\overline{\mathbf{M}}$,

$$\left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}}_\star \right\|_{\mathrm{F}}^2 \geq \left\| \overline{\mathbf{M}}^\top \mathbf{W}_\star \right\|_{\mathrm{F}}^2.$$

In turn, for any $\overline{\mathbf{W}} \in \mathcal{W}_k$ satisfying the assumptions of the lemma,

$$\left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}} \right\|_{\mathrm{F}}^2 \geq \gamma \cdot \left\| \overline{\mathbf{M}}^\top \mathbf{W}_\star \right\|_{\mathrm{F}}^2. \tag{C.14}$$

Let $\mathbf{A} \triangleq \mathbf{M}\mathbf{M}^\top$, $\widetilde{\mathbf{A}} \triangleq \overline{\mathbf{M}}\overline{\mathbf{M}}^\top$, and $\mathbf{E} \triangleq \mathbf{A} - \widetilde{\mathbf{A}}$. By the linearity of the trace,

$$\begin{aligned} \left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}} \right\|_{\mathrm{F}}^2 &= \mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{A} \overline{\mathbf{W}} \right) - \mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{E} \overline{\mathbf{W}} \right) \\ &\leq \mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{A} \overline{\mathbf{W}} \right) + \left| \mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{E} \overline{\mathbf{W}} \right) \right|. \end{aligned} \tag{C.15}$$

By Lemma C.2.44,

$$\left| \mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{E} \overline{\mathbf{W}} \right) \right| \leq \|\overline{\mathbf{W}}\|_{\mathrm{F}}^2 \cdot \|\mathbf{E}\|_2 \leq k \cdot \|\mathbf{E}\|_2 \triangleq R, \tag{C.16}$$

where the last inequality follows from the fact that $\|\mathbf{W}\|_{\mathrm{F}}^2 \leq k$ for any $\mathbf{W} \in \mathcal{W}_k$. Continuing from (C.15),

$$\left\| \overline{\mathbf{M}}^\top \overline{\mathbf{W}} \right\|_{\mathrm{F}}^2 \leq \mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{A} \overline{\mathbf{W}} \right) + R. \tag{C.17}$$

Similarly,

$$\begin{aligned}\left\|\overline{\mathbf{M}}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 &= \mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{A}\mathbf{W}_\star\right) - \mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{E}\mathbf{W}_\star\right) \\ &\geq \mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{A}\mathbf{W}_\star\right) - \left|\mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{E}\mathbf{W}_\star\right)\right| \\ &\geq \mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{A}\mathbf{W}_\star\right) - R. \end{aligned} \tag{C.18}$$

Combining the above, we have

$$\begin{aligned}\mathrm{Tr}\left(\overline{\mathbf{W}}^\top \mathbf{A}\overline{\mathbf{W}}\right) &\geq \left\|\overline{\mathbf{M}}^\top \overline{\mathbf{W}}\right\|_{\mathrm{F}}^2 - R \\ &\geq \gamma \cdot \left\|\overline{\mathbf{M}}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 - R \\ &\geq \gamma \cdot \left(\mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{A}\mathbf{W}_\star\right) - R\right) - R \\ &= \gamma \cdot \mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{A}\mathbf{W}_\star\right) - (1+\gamma) \cdot R \\ &\geq \gamma \cdot \mathrm{Tr}\left(\mathbf{W}_\star^\top \mathbf{A}\mathbf{W}_\star\right) - 2 \cdot R, \end{aligned}$$

where the first inequality follows from (C.17) the second from (C.14), the third from (C.18), and the last from the fact that $R \geq 0$ and $0 < \gamma \leq 1$. This concludes the proof. □

Remark C.1.3. *If in Lemma C.1.37* $\overline{\mathbf{M}}$ *is such that* $\mathbf{M}\mathbf{M}^\top - \overline{\mathbf{M}}\overline{\mathbf{M}}^\top$ *is PSD, then*

$$\left\|\mathbf{M}^\top \overline{\mathbf{W}}\right\|_{\mathrm{F}}^2 \geq \gamma \cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 - \sum_{i=1}^{k} \lambda_i\left(\mathbf{M}\mathbf{M}^\top - \overline{\mathbf{M}}\overline{\mathbf{M}}^\top\right).$$

Proof. This follows from the fact that if $\mathbf{E} \triangleq \mathbf{A} - \widetilde{\mathbf{A}}$ is PSD, then

$$\mathrm{Tr}\left(\widetilde{\mathbf{X}}^\top \mathbf{E}\widetilde{\mathbf{X}}\right) = \sum_{j}^{m} \mathbf{x}_j^\top \mathbf{E}\mathbf{x}_j \geq 0,$$

and the bound in (C.15) can be improved to

$$\begin{aligned}\left\|\overline{\mathbf{M}}^{\top}\overline{\mathbf{W}}\right\|_{\mathrm{F}}^{2} &= \mathrm{Tr}\left(\widetilde{\mathbf{X}}^{\top}\mathbf{A}\widetilde{\mathbf{X}}\right) - \mathrm{Tr}\left(\widetilde{\mathbf{X}}^{\top}\mathbf{E}\widetilde{\mathbf{X}}\right) \\ &\leq \mathrm{Tr}\left(\widetilde{\mathbf{X}}^{\top}\mathbf{A}\widetilde{\mathbf{X}}\right).\end{aligned}$$

Further, by Lemma C.2.44 (Corollary 5) the bound in (C.16) becomes

$$\mathrm{Tr}\left(\overline{\mathbf{W}}^{\top}\mathbf{E}\overline{\mathbf{W}}\right) \leq \|\overline{\mathbf{W}}\|_{\mathrm{F}}^{2} \cdot \|\mathbf{E}\|_{2} \leq \sum_{i=1}^{k} \lambda_i(\mathbf{E}).$$

The rest of the proof follows. □

Theorem 4.6. *For any real $m \times n$ (not necessarily nonnegative) matrix $\mathbf{M}$ and desired number of components k, let $\mathbf{W}_{\star} \triangleq \mathrm{argmax}_{\mathbf{W} \in \mathcal{W}_k} \left\|\mathbf{M}^{\top}\mathbf{W}\right\|_{\mathrm{F}}^{2}$. Let $\overline{\mathbf{M}}$ be the best rank-r approximation of $\mathbf{M}$. Algorithm 4.7 with input $\overline{\mathbf{M}}$ and accuracy parameters ϵ and r, outputs $\overline{\mathbf{W}} \in \mathcal{W}_k$ such that*

$$\left\|\mathbf{M}^{\top}\overline{\mathbf{W}}\right\|_{\mathrm{F}}^{2} \geq (1-\epsilon) \cdot \left\|\mathbf{M}^{\top}\mathbf{W}_{\star}\right\|_{\mathrm{F}}^{2} - k \cdot \|\mathbf{M} - \overline{\mathbf{M}}\|_{2}^{2}$$

in time $\mathrm{T}_{\mathsf{SVD}} + O\left((1/\epsilon)^{r \cdot k} \cdot k \cdot m\right)$.

Proof. Let $\overline{\mathbf{W}}$ be the output of Algorithm 4.7 with input the best rank-r approximation of of $\mathbf{M}$, $\overline{\mathbf{M}}$. By the guarantees of Algorithm 4.7, (Lemma 4.3.2), the output $\overline{\mathbf{W}} \in \mathcal{W}_k$ of Algorithm 4.7 is such that

$$\|\overline{\mathbf{M}}^{\top}\overline{\mathbf{W}}\|_{\mathrm{F}}^{2} \geq (1-\epsilon) \cdot \|\overline{\mathbf{M}}^{\top}\overline{\mathbf{W}}_{\star}\|_{\mathrm{F}}^{2},$$

where $\overline{\mathbf{W}}_{\star} \triangleq \mathrm{argmax}_{\mathbf{W} \in \mathcal{W}_k} \|\overline{\mathbf{M}}^{\top}\mathbf{W}\|_{\mathrm{F}}^{2}$. In turn, by Lemma C.1.37 (and in particular taking into account the remark C.1.3 whose conditions are satisfied

since $\mathbf{M}\mathbf{M} - \overline{\mathbf{M}}\overline{\mathbf{M}}^\top$ is PSD), we have

$$\begin{aligned}\left\|\mathbf{M}^\top\overline{\mathbf{W}}\right\|_{\mathrm{F}}^2 &\geq (1-\epsilon)\cdot\left\|\mathbf{M}^\top\mathbf{W}_\star\right\|_{\mathrm{F}}^2 - \sum_{i=1}^{k}\lambda_i\left(\mathbf{M}\mathbf{M}^\top - \overline{\mathbf{M}}\overline{\mathbf{M}}^\top\right)\\ &= (1-\epsilon)\cdot\left\|\mathbf{M}^\top\mathbf{W}_\star\right\|_{\mathrm{F}}^2 - \sum_{i=r+1}^{r+k}\sigma_i^2(\mathbf{M}).\end{aligned} \tag{C.19}$$

The desired result readily follows. □

C.1.4 Proof of Theorem 4.7

Lemma C.1.38. *For any $m \times n$ real nonnegative matrix $\mathbf{M}$, target dimension k, and accuracy parameters $r \in [n]$ and $\epsilon > 0$, Algorithm 4.8 outputs an ONMF pair $\overline{\mathbf{W}}, \overline{\mathbf{H}}$, such that*

$$\|\mathbf{M} - \overline{\mathbf{W}}\overline{\mathbf{H}}^\top\|_{\mathrm{F}}^2 \leq \mathcal{E}_\star + \epsilon\cdot\sum_{i=1}^{k}\sigma_i^2(\mathbf{M}) + \sum_{i=r+1}^{r+k}\sigma_i^2(\mathbf{M}),$$

in time $\mathrm{T}_{\mathsf{SVD}} + O\left((2/\epsilon)^{r\cdot k}\cdot k\cdot m\right)$.

Proof. Recall that given a real nonnegative $m \times k$ matrix $\mathbf{W} \in \mathcal{W}_k$, $\mathbf{H}^\top = \mathbf{W}_\star^\top\mathbf{M}$ minimizes the Frobenius error $\|\mathbf{M} - \mathbf{W}\mathbf{H}^\top\|_{\mathrm{F}}^2$ over the set of real nonnegative $n \times k$ matrices (Proof of Lemma C.1.36). In turn, for any $\mathbf{W} \in \mathcal{W}_k$, and $\mathbf{H}$ selected as above,

$$\|\mathbf{M} - \mathbf{W}\mathbf{H}^\top\|_{\mathrm{F}}^2 = \|\mathbf{M}\|_{\mathrm{F}}^2 - \|\mathbf{M}^\top\mathbf{W}\|_{\mathrm{F}}^2. \tag{C.20}$$

Let $\overline{\mathbf{W}}$ be the output of Algorithm 4.7, for input matrix $\mathbf{M}_r$ the best rank-r approximation of of $\mathbf{M}$. That is, in the sequel of this proof, $\overline{\mathbf{M}} = \mathbf{M}_r$. By the guarantees of Algorithm 4.7, (Lemma 4.3.2), the output $\overline{\mathbf{W}} \in \mathcal{W}_k$ of

Algorithm 4.7 is such that

$$\|\mathbf{M}_r^\top \overline{\mathbf{W}}\|_{\mathrm{F}}^2 \geq (1-\epsilon)\cdot \|\mathbf{M}_r^\top \overline{\mathbf{W}}_\star\|_{\mathrm{F}}^2,$$

where $\overline{\mathbf{W}}_\star \triangleq \mathrm{argmax}_{\mathbf{W}\in\mathcal{W}_k} \|\mathbf{M}_r^\top \mathbf{W}\|_{\mathrm{F}}^2$. In turn, by Lemma C.1.37 (and in particular taking into account the remark C.1.3 whose conditions are satisfied since $\mathbf{M}\mathbf{M} - \mathbf{M}_r\mathbf{M}_r^\top$ is PSD), we have

$$\begin{aligned}\left\|\mathbf{M}^\top \overline{\mathbf{W}}\right\|_{\mathrm{F}}^2 &\geq (1-\epsilon)\cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 - \sum_{i=1}^{k} \lambda_i\left(\mathbf{M}\mathbf{M}^\top - \mathbf{M}_r\mathbf{M}_r^\top\right)\\ &= (1-\epsilon)\cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 - \sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}). \qquad \text{(C.21)}\end{aligned}$$

Given the output $\overline{\mathbf{W}}$ of Algorithm 4.7, Algorithm 4.8 outputs the pair $\overline{\mathbf{W}}, \overline{\mathbf{H}}^\top \triangleq \overline{\mathbf{W}}^\top \mathbf{M}$. By (C.20), for this choice of $\overline{\mathbf{H}}$, taking into account (C.21),

$$\begin{aligned}\|\mathbf{M} - \overline{\mathbf{W}}\overline{\mathbf{H}}^\top\|_{\mathrm{F}}^2 &= \|\mathbf{M}\|_{\mathrm{F}}^2 - \|\mathbf{M}^\top \overline{\mathbf{W}}\|_{\mathrm{F}}^2\\ &\leq \|\mathbf{M}\|_{\mathrm{F}}^2 - (1-\epsilon)\cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 + \sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M})\\ &= \|\mathbf{M}\|_{\mathrm{F}}^2 - \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 + \epsilon\cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 + \sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M})\\ &= \|\mathbf{M} - \mathbf{W}_\star\mathbf{H}_\star^\top\|_{\mathrm{F}}^2 + \epsilon\cdot \left\|\mathbf{M}^\top \mathbf{W}_\star\right\|_{\mathrm{F}}^2 + \sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M})\\ &= \|\mathbf{M} - \mathbf{W}_\star\mathbf{H}_\star^\top\|_{\mathrm{F}}^2 + \epsilon\cdot \sum_{i=1}^{k} \sigma_i^2(\mathbf{M}) + \sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}),\end{aligned}$$

where the last inequality follows by Lemma C.2.44. This competes the proof.

□

Theorem 4.7. *For any $m \times n$ real nonnegative matrix $\mathbf{M}$, target dimension k, and desired accuracy $0 < \epsilon < 1$, Algorithm 4.8 with parameters ϵ and $r = \lceil k/\epsilon \rceil$*

outputs an ONMF pair $\mathbf{W}, \mathbf{H}$*, such that*

$$\|\mathbf{M} - \mathbf{W}\mathbf{H}^\top\|_{\mathrm{F}}^2 \;\leq\; \mathcal{E}_\star + \varepsilon \cdot \|\mathbf{M}\|_{\mathrm{F}}^2,$$

in time $\mathrm{T}_{\mathsf{SVD}} + (1/\epsilon)^{(k^2/\epsilon)} \cdot (k \cdot m)$.

Proof. By Lemma C.1.38, Algorithm 4.8 with parameters r and ϵ, outputs an ONMF pair $\overline{\mathbf{W}}, \overline{\mathbf{H}}$, such that

$$\|\mathbf{M} - \overline{\mathbf{W}}\overline{\mathbf{H}}^\top\|_{\mathrm{F}}^2 \leq \mathcal{E}_\star + \epsilon \cdot \sum_{i=1}^{k} \sigma_i^2(\mathbf{M}) + \sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}). \tag{C.22}$$

Noting that for $k < r$ the k squared singular values $\sigma_i^2(\mathbf{M})$, $i = r+1, \ldots, r+k$ are the smallest among the r squared singular values $\sigma_i^2(\mathbf{M})$, $i = k+1, \ldots, r+k$, the last term in the right-hand side can be upper bounded as follows:

$$\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}) \leq \frac{k}{r} \cdot \sum_{i=k+1}^{r+k} \sigma_i^2(\mathbf{M}). \tag{C.23}$$

For $r = \lceil k/\epsilon \rceil$, and combining (C.23) and (C.22), we have

$$\begin{aligned}
\|\mathbf{M} - \overline{\mathbf{W}}\overline{\mathbf{H}}^\top\|_{\mathrm{F}}^2 &\leq \mathcal{E}_\star + \epsilon \cdot \sum_{i=1}^{k} \sigma_i^2(\mathbf{M}) + \epsilon \cdot \sum_{i=k+1}^{r+k} \sigma_i^2(\mathbf{M}) \\
&= \mathcal{E}_\star + \epsilon \cdot \sum_{i=1}^{r+k} \sigma_i^2(\mathbf{M}) \\
&\leq \mathcal{E}_\star + \epsilon \cdot \|\mathbf{M}\|_{\mathrm{F}}^2,
\end{aligned}$$

which is the desired guarantee. The time complexity readily follows from the steps of Algorithm 4.8 and that of Algorithm 4.7, which concludes the proof. □

C.1.5 Correctness of Algorithm 4.9

Lemma C.1.39. *For any $m \times k$ matrix $\mathbf{A}$, Algorithm 4.9 outputs the $m \times k$ nonnegative matrix*

$$\widehat{\mathbf{W}} = \operatorname*{argmax}_{\mathbf{W} \in \mathcal{W}_k} \sum_{j=1}^{k} \langle \mathbf{w}_j, \mathbf{a}_j \rangle^2,$$

in time $O(2^k \cdot k \cdot m)$.

Proof. Let $\mathcal{I}_j \subseteq [m]$, $j = 1 \ldots, k$ denote the supports of the k columns of optimal solution $\widehat{\mathbf{W}}$. The orthogonality requirements (in conjunction with nonnegativity) imply that the supports $\mathcal{I}_j$, $j = 1, \ldots, k$ are disjoint. Further, it is straightforward to verify that due to the nonnegativity constrains in $\mathcal{W}_k$, the support of the jth column, $\mathcal{I}_j$, must contain only indices corresponding to nonnegative or nonpositive entries of $\mathbf{a}_j$, but not a combination of both. Algorithm 4.9 considers all 2^k sign combinations for the support sets, *e.g.*, $\mathcal{I}_1$ containing positive entries, $\mathcal{I}_2$ negative, etc., by equivalently solving the maximization on all 2^k matrices $\hat{\mathbf{A}} = \mathbf{A} \cdot \operatorname{diag}(\mathbf{s})$, $b \in \{\pm 1\}^k$ and returning the solution that performs best on the original input $\mathbf{A}$. Therefore, without loss of generality, in the sequel we assume that all support sets correspond to nonnegative entries of $\mathbf{A}$.

If an oracle reveals the supports $\mathcal{I}_j$, $j = 1, \ldots, k$, the exact value of $\widehat{\mathbf{X}}$ can be readily determined, according to the Cauchy-Schwarz inequality: the jth column, $\widehat{\mathbf{x}}_j$, is supported only on $\mathcal{I}_j$, and its nonzero sub-vector is set to $(\widehat{\mathbf{x}}_j)_{\mathcal{I}_j} = [\mathbf{a}_j]_{\mathcal{I}_j} / \|[\mathbf{a}_j]_{\mathcal{I}_j}\|$, which maximizes the inner product with the

corresponding sub-vector of $\mathbf{a}_j$. In turn, the objective function attains value equal to

$$\sum_{j=1}^{k}(\widehat{\mathbf{x}}_j^\top \mathbf{a}_j)^2 = \sum_{j=1}^{k}\sum_{i\in\mathcal{I}_j} A_{ij}^2, \tag{C.24}$$

where the first equality stems from the fact that $[\mathbf{a}_j]_{\mathcal{I}_j} \geq \mathbf{0}$. It suffices to show that Alg. 4.9 correctly determines the collection of support sets $\mathcal{I}_j$, $j = 1, \ldots, k$.

Alg. 4.9 constructs the collection of support sets $\mathcal{I}_j$, $j = 1, \ldots, k$, according to the following rule:

$$i \in \mathcal{I}_j \Leftrightarrow A_{ij} > \max\{0,\ A_{iw}\},\ \forall w \in [k]\backslash\{j\}, \tag{C.25}$$

i.e., index $i \in [m]$ is assigned to the support of the jth column if and only if A_{ij} is positive and the largest entry in the ith row of $\mathbf{A}$. Note that any procedure to construct supports that satisfy the requirements described in the beginning of this proof would assign each index $i \in [m]$ to at most one of the sets $\mathcal{I}_j$, $j \in \{1, \ldots, k\}$, while it would need to ensure that $i \in \mathcal{I}_j$ if and only if $A_{ij} > 0$. The rule in (C.25) additionally requires that index $i \in [m]$ is assigned to $\mathcal{I}_j$ if and only if A_{ij} is the *largest* (positive) entry in the ith row of $\mathbf{A}$.

Assume, for the sake of contradiction, that there exists a set of optimal supports $\mathcal{I}_j$, $j = 1, \ldots, k$ which does not adhere to the rule in (C.25), *i.e.*, there exist $u \in [k]$ and $q \in [m]$, such $q \in \mathcal{I}_u$, while $0 < A_{qu} < A_{qv}$ for some $v \in [k], v \neq u$. Consider a collection of supports $\widetilde{\mathcal{I}}_j$, $j = 1, \ldots, k$, with

$$\widetilde{\mathcal{I}}_j = \mathcal{I}_j,\ \forall j \in [k]\backslash\{u, v\}, \quad \widetilde{\mathcal{I}}_u = \mathcal{I}_u\backslash\{q\} \quad \text{and} \quad \widetilde{\mathcal{I}}_v = \mathcal{I}_v \cup \{q\}. \tag{C.26}$$

Note that the collection of supports in (C.26) satisfies the desired constraints. Further, the objective value achieved for the new supports (according to (C.24)) is equal to

$$\sum_{j=1}^{k}\sum_{i\in\widetilde{\mathcal{I}}_j} A_{ij}^2 = \sum_{j=1}^{k}\sum_{i\in\mathcal{I}_j} A_{ij}^2 - A_{qu}^2 + A_{qv}^2 > \sum_{j=1}^{k}\sum_{i\in\mathcal{I}_j} A_{ij}^2,$$

contradicting the optimality of the collection $\mathcal{I}_j$, $j = 1, \ldots, k$. We conclude that the collection of optimal support sets must satisfy (C.25).

The construction of the support sets according to (C.25) requires determining the largest entry of each of the m rows of $\mathbf{A}$, which can be done in $O(km)$. Once the supports are determined, each of the k columns of $\widehat{\mathbf{X}}$ is constructed in $O(m)$. Taking into account that the above procedure is repeated 2^k times for each of the sign patters, the desired result follows. □

C.2 Auxiliary Lemmas

Lemma C.2.40. *For any real $m \times n$ matrix $\mathbf{M}$, and any $r, k \leq \min\{m, n\}$,*

$$\sum_{i=r+1}^{r+k} \sigma_i(\mathbf{M}) \leq \frac{k}{\sqrt{r+k}} \cdot \|\mathbf{M}\|_{\mathrm{F}},$$

where $\sigma_i(\mathbf{M})$ is the ith largest singular value of $\mathbf{M}$.

Proof. By the Cauchy-Schwartz inequality,

$$\sum_{i=r+1}^{r+k} \sigma_i(\mathbf{M}) = \sum_{i=r+1}^{r+k} |\sigma_i(\mathbf{M})| \leq \left(\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M})\right)^{1/2} \cdot \|\mathbf{1}_k\|_2 = \sqrt{k} \cdot \left(\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M})\right)^{1/2}.$$

Note that $\sigma_{r+1}(\mathbf{M}), \ldots, \sigma_{r+k}(\mathbf{M})$ are the k smallest among the $r+k$ largest singular values. Hence,

$$\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{M}) \leq \frac{k}{r+k} \sum_{i=1}^{r+k} \sigma_i^2(\mathbf{M}) \leq \frac{k}{r+k} \sum_{i=1}^{\min\{m,n\}} \sigma_i^2(\mathbf{M}) = \frac{k}{r+k} \|\mathbf{M}\|_{\mathrm{F}}^2.$$

Combining the two inequalities, the desired result follows. □

Corollary 3. *For any real $m \times n$ matrix $\mathbf{M}$ and $k \leq \min\{m, n\}$, $\sigma_k(\mathbf{M}) \leq k^{-1/2} \cdot \|\mathbf{M}\|_{\mathrm{F}}$.*

Proof. It follows immediately from Lemma C.2.40. □

Lemma C.2.41. *Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be $2n$ real numbers and let p and q be two numbers such that $1/p + 1/q = 1$ and $p > 1$. We have*

$$\left| \sum_{i=1}^{n} a_i b_i \right| \leq \left(\sum_{i=1}^{n} |a_i|^p \right)^{1/p} \cdot \left(\sum_{i=1}^{n} |b_i|^q \right)^{1/q}.$$

Lemma C.2.42. *For any $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times k}$,*

$$|\langle \mathbf{A}, \mathbf{B} \rangle| \triangleq \left| \mathrm{TR}\left(\mathbf{A}^\top \mathbf{B} \right) \right| \leq \|\mathbf{A}\|_{\mathrm{F}} \|\mathbf{B}\|_{\mathrm{F}}.$$

Proof. The inequality follows from Lemma C.2.41 for $p = q = 2$, treating $\mathbf{A}$ and $\mathbf{B}$ as vectors. □

Lemma C.2.43. *For any two real matrices $\mathbf{A}$ and $\mathbf{B}$ of appropriate dimensions, $\|\mathbf{AB}\|_{\mathrm{F}} \leq \min\{\|\mathbf{A}\|_2 \|\mathbf{B}\|_{\mathrm{F}}, \ \|\mathbf{A}\|_{\mathrm{F}} \|\mathbf{B}\|_2\}$.*

Proof. Let $\mathbf{b}_i$ denote the ith column of $\mathbf{B}$. Then,

$$\|\mathbf{AB}\|_{\mathrm{F}}^2 = \sum_i \|\mathbf{A}\mathbf{b}_i\|_2^2 \leq \sum_i \|\mathbf{A}\|_2^2 \|\mathbf{b}_i\|_2^2 = \|\mathbf{A}\|_2^2 \sum_i \|\mathbf{b}_i\|_2^2 = \|\mathbf{A}\|_2^2 \|\mathbf{B}\|_{\mathrm{F}}^2.$$

Similarly, using the previous inequality,

$$\|\mathbf{AB}\|_{\mathrm{F}}^2 = \|\mathbf{B}^\top \mathbf{A}^\top\|_{\mathrm{F}}^2 \leq \|\mathbf{B}^\top\|_2^2 \|\mathbf{A}^\top\|_{\mathrm{F}}^2 = \|\mathbf{B}\|_2^2 \|\mathbf{A}\|_{\mathrm{F}}^2.$$

Combining the two upper bounds, the desired result follows. □

Corollary 4. *Let $\mathbf{M}_r$ denote the best rank-r approximation of $\mathbf{M}$, obtained by the truncated singular value decomposition of $\mathbf{M}$. Then, for any $d > r$, $\sigma_d \leq \|\mathbf{M} - \mathbf{M}_r\|_{\mathrm{F}}/\sqrt{d-r}$.*

Proof. By definition

$$\|\mathbf{M} - \mathbf{M}_r\|_{\mathrm{F}}^2 = \sum_{i=r+1}^{n} \sigma_i^2 \geq \sum_{i=r+1}^{d} \sigma_i^2 \geq (d-r) \cdot \sigma_d^2,$$

from which the desired result follows. □

Lemma C.2.44. *For any real $m \times n$ matrix $\mathbf{A}$, and any $k \leq \min\{m,\, n\}$,*

$$\max_{\substack{\mathbf{Y} \in \mathbb{R}^{n \times k} \\ \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k}} \|\mathbf{AY}\|_{\mathrm{F}} = \left(\sum_{i=1}^{k} \sigma_i^2(\mathbf{A}) \right)^{1/2}.$$

achieved for $\mathbf{Y}$ coinciding with the k leading right singular vectors of $\mathbf{A}$.

Proof. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the singular value decomposition of $\mathbf{A}$; $\mathbf{U}$ and $\mathbf{V}$ are $m \times m$ and $n \times n$ unitary matrices respectively, while Σ is a diagonal matrix with $\Sigma_{jj} = \sigma_j$, the jth largest singular value of $\mathbf{A}$, $j = 1, \ldots, d$, where $d \triangleq \min\{m, n\}$. Due to the invariance of the Frobenius norm under unitary multiplication,

$$\|\mathbf{AY}\|_{\mathrm{F}}^2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \mathbf{Y}\|_{\mathrm{F}}^2 = \|\mathbf{\Sigma}\mathbf{V}^\top \mathbf{Y}\|_{\mathrm{F}}^2. \tag{C.27}$$

Continuing from (C.27),

$$\begin{aligned}\|\mathbf{\Sigma V}^\top \mathbf{Y}\|_{\mathrm{F}}^2 &= \mathrm{Tr}\Big(\mathbf{Y}^\top \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top \mathbf{Y}\Big) \\ &= \sum_{i=1}^{k} \mathbf{y}_i^\top \left(\sum_{j=1}^{d} \sigma_j^2 \cdot \mathbf{v}_j \mathbf{v}_j^\top\right) \mathbf{y}_i = \sum_{j=1}^{d} \sigma_j^2 \cdot \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2.\end{aligned}$$

Let $z_j \triangleq \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2$, $j = 1, \ldots, d$. Note that each individual z_j satisfies

$$0 \leq z_j \triangleq \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2 \leq \|\mathbf{v}_j\|^2 = 1,$$

where the last inequality follows from the fact that the columns of $\mathbf{Y}$ are orthonormal. Further,

$$\sum_{j=1}^{d} z_j = \sum_{j=1}^{d} \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2 = \sum_{i=1}^{k} \sum_{j=1}^{d} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2 = \sum_{i=1}^{k} \|\mathbf{y}_i\|^2 = k.$$

Combining the above, we conclude that

$$\|\mathbf{AY}\|_{\mathrm{F}}^2 = \sum_{j=1}^{d} \sigma_j^2 \cdot z_j \leq \sigma_1^2 + \ldots + \sigma_k^2. \tag{C.28}$$

Finally, it is straightforward to verify that if $\mathbf{y}_i = \mathbf{v}_i$, $i = 1, \ldots, k$, then (C.28) holds with equality. □

Corollary 5. *For any real $m \times m$ PSD matrix $\mathbf{A}$, and $k \times m$ matrix $\mathbf{X}$ with $k \leq m$ orthonormal columns,*

$$\mathrm{Tr}\Big(\mathbf{X}^\top \mathbf{A}\mathbf{X}\Big) = \sum_{i=1}^{k} \lambda_i(\mathbf{A})$$

where $\lambda_i(\mathbf{A})$ is the ith largest eigenvalue of $\mathbf{A}$. Equality is achieved for $\mathbf{X}$ coinciding with the k leading eigenvectors of $\mathbf{A}$.

Proof. Let $\mathbf{A} = \mathbf{V}\mathbf{V}^\top$ be a factorization of the PSD matrix $\mathbf{A}$. Then, $\textsc{Tr}\Big(\mathbf{X}^\top\mathbf{A}\mathbf{X}\Big) = \textsc{Tr}\Big(\mathbf{X}^\top\mathbf{V}\mathbf{V}^\top\mathbf{X}\Big) = \|\mathbf{V}^\top\mathbf{X}\|_{\mathrm{F}}^2$. The desired result follows by Lemma C.2.44 and the fact that $\lambda_i(\mathbf{A}) = \sigma_i^2(\mathbf{V})$, $i = 1, \ldots, m$. □

C.3 Net of the ℓ_2-unit sphere

In this section, we provide a simple probabilistic construction of an ϵ-net of the ℓ_2-unit sphere $\mathbb{S}_2^{d-1}$, *i.e.*, the set of points $\mathbf{x}$ such that $\|\mathbf{x}\|_2 = 1$.

Lemma C.3.45 ([140], Lemma 5.2)**.** *For any $\epsilon > 0$, there exists an ϵ-net $\mathcal{N}_\epsilon$ of the unit Euclidean sphere $\mathbb{S}_2^{d-1}$ equipped with the Euclidean metric, such that*

$$m_\epsilon \triangleq |\mathcal{N}_\epsilon| \leq (1 + 2/\epsilon)^d.$$

Proof. Let $\mathcal{N}_\epsilon$ be a maximal ϵ-separated subset of $\mathbb{S}_2^{d-1}$. In other words, $d(x, y) \geq \epsilon$ for all $x, y \in \mathcal{N}_\epsilon$, $x \neq y$, and no set containing $\mathcal{N}_\epsilon$ has this property.

Such a set can be constructed iteratively: select an arbitrary point on the sphere and at each subsequent step select a point that is at distance at least ϵ from all previously selected points. By the compactness of the sphere, the iterative construction process will terminate after a finite number of steps, and the resulting set will satisfy the above properties.

The maximality property, implies that $\mathcal{N}_\epsilon$ is an ϵ-net of $\mathbb{S}_2^{d-1}$. If this was not the case, then there would exist an $x \in \mathbb{S}_2^{d-1}$ such that $d(x, y) > \epsilon$, $\forall y \in \mathcal{N}_\epsilon$. The $\mathcal{N}_\epsilon \cup \{x\}$ would an ϵ-separated set, that contains $\mathcal{N}_\epsilon$, contradicting the maximality of the latter.

By the separation property, we infer that the balls of radius $\epsilon/2$ centered at the points of $\mathcal{N}_\epsilon$ are disjoint. This follows from the triangle inequality. Further, all such balls lie in the ball $(1+\epsilon/2)\mathbb{B}_2^d$, where $\mathbb{B}_2^d$ denotes the unit Euclidean ball centered at the origin. Comparing the volumes, we have

$$\mathrm{Vol}\Big(\tfrac{\epsilon}{2}\mathbb{B}_2^d\Big) \cdot |\mathcal{N}_\epsilon| \leq \mathrm{Vol}\Big((1+\tfrac{\epsilon}{2})\mathbb{B}_2^d\Big).$$

Taking into account that $\mathrm{Vol}\Big(r \cdot \mathbb{B}_2^d\Big) = r^d \cdot \mathrm{Vol}\Big(\mathbb{B}_2^d\Big)$,

$$|\mathcal{N}_\epsilon| \leq \Big(1+\tfrac{\epsilon}{2}\Big)^d / \Big(\tfrac{\epsilon}{2}\Big)^d = \Big(1+\tfrac{2}{\epsilon}\Big)^d,$$

which is the desired result. □

Lemma C.3.45 regards the unit Euclidean sphere. However, the sequence of arguments used in the proof essentially hold for the case of the unit ball $\mathbb{B}_2^d$, *i.e.*, there exists an ϵ-net of $\mathbb{B}_2^d$, with at most $(1+2/\epsilon)^d$ points.

Constructing an ϵ-net of the unit sphere. There are many constructions for ϵ-nets on the sphere, both deterministic and randomized. In the following we review a simple randomized construction, initially studied by Wyner [151] in the asymptotic $d \to \infty$ regime.

By Lemma C.3.45, there exists an ϵ-net $\mathcal{N}_\epsilon$ of $\mathbb{S}_2^{d-1}$ Consider the balls of radii ϵ centered at the points of $\mathcal{N}_\epsilon$. The balls cover all points of $\mathbb{S}_2^{d-1}$; if there existed a point x on $\mathbb{S}_2^{d-1}$ not included in any ball, it would imply that this point is at distance at least ϵ from all points of $\mathcal{N}_\epsilon$ contradicting the fact that $\mathcal{N}_\epsilon$ is a ϵ-net.

The intersection of each of the previous balls with $\mathbb{S}_2^{d-1}$ is a spherical cap, and hence, according to the above, the collection of sperical caps covers $\mathbb{S}_2^{d-1}$. (Note that the spherical caps, as well as the balls, overlap.)

Consider a set $\mathcal{Q}$, containing at least one point from each spherical cap. Then, $\mathcal{Q}$ is a 2ϵ-net of $\mathbb{S}_2^{d-1}$. To verify that, note the following. Consider a point $x \in \mathbb{S}_2^{d-1}$. By construction, $\mathcal{N}_\epsilon$ contains a point y such that $d(x, y) \leq \epsilon$. Consider the spherical cap centered at y. By definition, $\mathcal{Q}$ contains a point $\widetilde{y}$ in that spherical cap, and hence $d(y, \widetilde{y}) \leq \epsilon$. By triangle inequality, it follows that $d(x, \widetilde{y}) \leq 2\epsilon$. Since the point x is arbitrary, we conclude that $\mathcal{Q}$ is a 2ϵ-net.

We draw points randomly and independently, uniformly distributed on $\mathbb{S}_2^{d-1}$. This can be accomplished, for instance, by randomly and independently generating vectors in $\mathbb{R}^d$ distributed according to the a the multivariate normal distribution $N(\mathbf{0}, \mathbf{I})$ and normalizing their length to 1. A randomly selected point lies in a specific spherical cap with probability $p \geq 1/m_\epsilon$. By a standard probability arguments (Coupon collector's problem), $O(m_\epsilon \ln (m_\epsilon/\delta))$ points uniformly distributed over $\mathbb{S}_2^{d-1}$ suffice for at least one random point to lie in each sphere cap with probability at least $1 - \delta$. Substituting the value of m_ϵ from Lemma C.3.45, we find that $O\Big(d\epsilon^{-d} \cdot \ln \frac{1}{\epsilon\delta}\Big)$ suffice to form a 2ϵ-net, with probability at least $1 - \delta$. Note that δ can be chosen to scale with the dimension n of the problem.

Lemma C.3.46. *A set of $O\Big(d(\epsilon/2)^{-d} \cdot \ln \frac{2}{\epsilon\cdot\delta}\Big)$ randomly and independently drawn points uniformly distributed on $\mathbb{S}_2^{d-1}$ suffices to construct an ϵ-net of $\mathbb{S}_2^{d-1}$*

Component 1	Component 2	Component 3	Component 4	Component 5
american	coach	add	ago	billion
attack	game	cup	called	business
campaign	games	food	com	companies
country	guy	hour	family	company
government	hit	large	help	cost
group	left	makes	high	deal
leader	night	minutes	home	industry
official	play	oil	look	market
political	player	pepper	need	million
president	point	serving	part	money
zzz_al_gore	run	small	problem	number
zzz_bush	season	sugar	program	percent
zzz_george_bush	team	tablespoon	right	plan
zzz_u_s	win	teaspoon	school	stock
zzz_united_states	won	water	show	system

Table C.1: ONMF with $r = 5$ orthogonal components ($102 \cdot 10^3$–dimensional vectors) on the words-by-document matrix of the NY Times bag-of-words dataset [22]. The table depicts the words corresponding to the 15 largest entries of each component. The 5 retrieved components are extremely sparse: 90% of their mass is concentrated in 134, 35, 65, 269 and 59 entries, respectively.

with probability at least $1 - \delta$.

C.4 Additional Experimental Results

Large-scale text analysis: clustering words. We evaluate the performance of ONMFS as a clustering algorithm on the NY Times bag-of-words dataset [22]. The dataset is represented by a 102K $\times$ 300K words-by-articles matrix. Given an approximate ONMF of that matrix, the 102K $\times$ k nonnegative, orthogonal factor $\mathbf{W}$ induces an assignment of words to r clusters, which in this case can be interpreted as *topics*. That is, each column of $\mathbf{W}$ suggests

a topic, defined by the words corresponding to its nonzero entries.

We run ONMFS with target dimension $k = 5$ topics, and accuracy parameter $r = 5$, while we configure it to stop if no progress is observed after $T = 300$ consecutive candidate solutions. Table C.1 lists the words corresponding to the 15 largest entries of each orthogonal component (column of $\mathbf{W}$). Arguably, each component can be interpreted as a distinct topic, illustrating the potential of ONMF in text analysis. Further, we note that although the components of $\mathbf{W}$ are not explicitly restricted to be sparse, they tend to be: 90% of the ℓ_2 mass of each component is concentrated in approximately 100-200 entries (words) out of the roughly 102K present in the dataset.

Appendix D

Appendix for Chapter 5

D.1 Proof of NP-Hardness

GraphPathSPCA	
Input:	$\mathbf{A}$ – $n \times n$ PSD matrix
	G – DAG on n vertices (with auxiliary source and sink vertices S and T)
	ρ – threshold $\rho \in \mathbb{R}_+$.
Question:	Is there $\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{x}\|_2 = 1$ and $\text{supp}(\mathbf{x}) \in \mathcal{P}(G)$[1], such that $\mathbf{x}^\top \mathbf{A}\mathbf{x} \geq \rho$?

Let GraphPathSPCA denote the decision version of the constrained quadratic maximization problem (5.4). We show that GraphPathSPCA is NP-Complete via a reduction from the kCLIQUE, *i.e.*, the problem of determining whether a given arbitrary undirected graph G contains a clique of size k (where the parameter k is part of the input). We perform the reduction in multiple steps. First we show that a seemingly special case of kCLIQUE, referred to as kPkCLIQUE is also NP-Complete. In kPkCLIQUE, ones seeks to determine whether a k-partite undirected graph G contains a k-clique. We show

[1] $\mathcal{P}(G)$ denotes the collection of S-T paths in G Since G is a DAG, an S-T path corresponds uniquely to a set of vertices in G, excluding S and T.

that this problem is NP-Complete by a reduction from the general kCLIQUE problem itself. Subsequently, we describe a reduction from kPkCLIQUE to MultiChoicePCA, a variant of PCA in which variables are subdivided into disjoint groups and the solution must assign a nonzero value to at most one variable from each group. The final step is to show that MultiChoicePCA is only a special case of GraphPathSPCA, which implies the hardness result.

D.1.1 Hardness of kPkCLIQUE

kCLIQUE	
Input: $G = (V, E)$	– Undirected graph
k	– Integer in $\{1, \ldots, \lvert V \rvert\}$
Question: Does G contain a k-clique?	

kCLIQUE is a well known NP-Complete problem. We consider a special case referred to as the k-Partite k-Clique problem, or kPkCLIQUE.

Def. D.1.4. *A k-partite graph $G = (V_1, \ldots, V_s, E)$ is a graph whose vertices can be partitioned into k disjoint sets $V_1, \ldots, V_s$, so that no two vertices within the same set are adjacent.*

In other words, each of the k vertex subsets $V_1, \ldots, V_s$ in the k-partite graph G forms an independent set. The absence of edges within each set V_i, $i = 1, \ldots, k$, implies that any clique in G can contain at most one vertex from each set, and the clique number $\omega(G)$ is upper bounded by k.

kPkCLIQUE is the problem of deciding whether a given undirected

k-partite graph G *with known vertex partition* $V_1, \ldots, V_s$, has a k-clique, *i.e.*, a clique comprising a vertex from each of the sets V_i, $i = 1, \ldots, k$.

κPκCLIQUE

Input: $G = (V_1, \ldots, V_s, E)$– Undirected k-partite graph along with the vertex partition $V_1, \ldots, V_s$.

Question: Does G contain a k-clique?

We show that κPκCLIQUE is NP-Complete by a reduction from κCLIQUE. First, note that κPκCLIQUE is in NP: given a set S of k vertices it can be verified in polynomial time whether these vertices form a k-clique. For the reduction, given an input $[G = (V, E), k]$ for κCLIQUE, consider the undirected k-partite graph $\widehat{G} = (\widehat{V}, \widehat{E})$ with vertex set

$$\widehat{V} \triangleq V \times [k] = \{(v, i) : \ v \in V, i \in [k]\},$$

partitioned into k sets

$$\widehat{V}_i \triangleq \left\{(v, i) \in \widehat{V} : v \in V\right\}, \quad i = 1, \ldots, k.$$

An edge between (v, i) and (u, j) exists and only if

$$v \neq u \ \wedge \ i \neq j \ \wedge \ (v, u) \in E$$

which renders $\widehat{G}$ k-partite with partition $\{V_i\}_{i=1}^{s}$.

$\widehat{G}$ contains a k-clique if and only if G contains a k-clique. If G contains a k-clique among vertices $v_1, \ldots, v_s \in V$, then $(v_1, 1), \ldots, (v_s, k) \in \widehat{V}$ form a k-clique in $\widehat{G}$. Conversely, if $\widehat{G}$ contains a k-clique, the latter must

contain exactly one vertex from each group $\widehat{V}_i$ and hence must be of the form $(v_1, 1), \ldots, (v_s, k)$ for some $v_1, \ldots, v_s \in V$ implying that the latter form a k-clique in G. Finally, note that $\widehat{G}$ is constructed in time polynomial in k and the size of G. We conclude that kPkCLIQUE is NP-Complete.

D.1.2 Hardness of MultiChoicePCA

MultiChoicePCA

Input: $\mathbf{A}$ – $n \times n$ PSD matrix
$P_1, \ldots, P_s$ – a partition of the n variables into k disjoint sets
ρ – threshold $\rho \in \mathbb{R}_+$.
Question: Is there $\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{x}\|_2 = 1$ and $|\text{supp}(\mathbf{x}) \cap P_i| \leq 1 \ \forall i \in [k]$, such that $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq \rho$?

MultiChoicePCA is a constrained version of the vanilla PCA problem, *i.e.*, the problem maximizing the Rayleigh quotient on an $n \times n$ PSD matrix. In MultiChoicePCA, the n variables are subdivided into k classes and the solution of the quadratic maximization can contain at most one nonzero variable from each class. We show that MultiChoicePCA is NP-Complete via a reduction from kPkCLIQUE.

Consider an undirected k-partite graph $G = (V, E)$ on n vertices, with known vertex partition $V_1, \ldots, V_s$. Assume an arbitrary labeling $1, \ldots, n$ of the vertices, and let $\mathbf{A}$ denote the adjacency matrix of G. For any $S \subset V$, let $\mathbf{A}_S$ denote the principal submatrix of $\mathbf{A}$ corresponding to S, *i.e.*, the $|S| \times |S|$ adjacency of the subgraph induced by S.

Lemma D.1.47. *Let* $\mathbf{A}$ *be the adjacency matrix of a graph* G *on* k *vertices. If* G *is the complete graph (an* k*-clique), then* $\lambda_1(\mathbf{A}) = k-1$ *with corresponding eigenvector* $1/\sqrt{s} \cdot \mathbf{1}$*. Otherwise,* $\lambda_1(\mathbf{A}) < k-1$*.*

Proof. Consider an arbitrary labeling $1, \ldots, k$ of the k vertices in G. For any graph G, $\lambda_1(\mathbf{A}) \leq \max_{i \in [k]} d(i)$, where $d(i)$ denotes the degree of vertex i. To verify that, let $\mathbf{u} \neq \mathbf{0}$ be the eigenvector of $\mathbf{A}$ corresponding to the largest eigenvalue. By the Perron-Frobenius theorem $\mathbf{u} \geq \mathbf{0}$. Without loss of generality, let u_j be the largest entry of $\mathbf{u}$. Then,

$$\lambda_1(\mathbf{A}) = \frac{[\mathbf{A}\mathbf{u}]_j}{u_j} = \sum_{i \in \Gamma(j)} \frac{u_i}{u_j} \leq d(j) \leq k-1, \tag{D.1}$$

where $\Gamma(i)$ denotes the neighborhood of vertex i.

If G is the complete graph on k vertices, the adjacency matrix is $\mathbf{A} = \mathbf{1}\mathbf{1}^\top - \mathbf{I}_s$. In that case, $k^{-1/2} \cdot \mathbf{1}$ is an eigenvector of $\mathbf{A}$ achieving the former upper bound. We conclude that if G is the complete graph, then $\lambda_1(\mathbf{A}) = k-1$ with corresponding eigenvector $\mathbf{u} = k^{-1/2} \cdot \mathbf{1}$.

It remains to show that if G is *not* the complete graph, then $\lambda_1(\mathbf{A}) < k-1$. Equivalently, we show that if $\lambda_1(\mathbf{A}) = k-1$, then G *is* the complete graph.

Without loss of generality, we can assume that G is a connected graph. If G is not connected, then $\mathbf{A}$ can be brought (with suitable row/column permutation) into a block diagonal form, where each block corresponds to a connected component. The eigenvalues of $\mathbf{A}$ are obtained by putting together

the eigenvalues of the individual blocks. By (D.1), the largest eigenvalue of a block is upper bounded by $l-1$, where l is the dimension of the block, while the existence of multiple components implies that $l < k$.

Assume that $\lambda_1(\mathbf{A}) = k-1$ and let $\mathbf{u}$ be the corresponding eigenvector. As before, let j be the index of the largest entry in $\mathbf{u}$. By (D.1), we conclude that $d(j) = k-1$ and $u_j = u_i,\ \forall i \in \Gamma(j)$. But the fact that $d(j) = k-1$ implies that $\Gamma(j) = [k]$ and in turn that all entries of $\mathbf{u}$ are equal. Hence, repeating the argument of (D.1) on i (instead of j),

$$k-1 = \lambda_1(\mathbf{A}) = \frac{[\mathbf{Au}]_i}{u_i} \leq d(i) \leq k-1 \quad \forall i \in [k].$$

We conclude that G is the complete graph. □

Consider the constrained quadratic maximization

$$\mathrm{OPT}_s \triangleq \max_{\mathbf{x} \in \mathcal{X}_s} \mathbf{x}^\top \mathbf{A} \mathbf{x}, \tag{D.2}$$

where

$$\mathcal{X}_s \triangleq \{\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ \|\mathbf{x}\|_0 \leq s\}.$$

Let $\mathbf{x}_\star \in \mathcal{X}_s$ denote the solution of (D.2) and $S_\star = \mathrm{supp}(\mathbf{x}_\star)$. The objective value attained at $\mathbf{x}_\star$ is $\mathrm{OPT}_s = \lambda_1(\mathbf{A}_{S_\star})$ By Lemma D.1.47, $\lambda_1(\mathbf{A}_S) \leq k-1$, $\forall S \subset V$, $|S| = k$, with equality achieved if and only if S forms a k-clique. We conclude that *G contains a k-clique if and only if* $\mathrm{OPT}_s = k-1$.

Observe that the aforementioned criterion based on the value of (D.2) holds for arbitrary graphs. Here, G is k-partite. Hence, any k-clique in G (if

one exists) will contain exactly one vertex from each of the sets $V_1, \ldots, V_s$; if $S_\star$ forms a k-clique, then $|S_\star \cap V_i| = 1$, $\forall i \in [k]$. Hence, we can explicitly enforce the constrain $|\text{supp}(\mathbf{x}) \cap V_i| \leq 1$, $\forall i \in [k]$ in (D.2) without affecting the decision criterion. Let

$$\text{OPT}'_s \triangleq \max_{\mathbf{x} \in \mathcal{X}'_s} \mathbf{x}^\top \mathbf{A} \mathbf{x}, \tag{D.3}$$

where

$$\mathcal{X}'_s \triangleq \{\mathbf{x} : \|\mathbf{x}\|_2 = 1,\ |\text{supp}(\mathbf{x}) \cap V_i| \leq 1 \, \forall i \in [k]\},$$

It follows that *G has a k-clique if and only if* $\text{OPT}'_s = k - 1$.

Problem (D.3) closely resembles MultiChoicePCA, but does not satisfy the restriction that the input argument $\mathbf{A}$ must be a PSD matrix. In fact, $\mathbf{A}$ is the adjacency of a graph and will not be PSD. However, we can equivalently solve the quadratic maximization on $\mathbf{A}' = \mathbf{A} + |\lambda_n(\mathbf{A})| \cdot \mathbf{I}$, where $\lambda_n(\mathbf{A})$ is the smallest eigenvalue of $\mathbf{A}$. The new matrix $\mathbf{A}'$ is PSD and can be obtained from $\mathbf{A}$ in polynomial time. Further,

$$\mathbf{x}^\top \mathbf{A}' \mathbf{x} = \mathbf{x}^\top \mathbf{A} \mathbf{x} + |\lambda_n(\mathbf{A})|, \quad \forall \mathbf{x} \in \mathbb{R}^n : \ \|\mathbf{x}\|_2 = 1.$$

Summarizing the above, given an undirected k-partite graph G on n vertices with known vertex partition $V_1, \ldots, V_s$ and adjacency matrix $\mathbf{A}$ as input to kPkCLIQUE, we can decide whether a k-clique exists in G by solving MultiChoicePCA on the PSD matrix $\mathbf{A}' = \mathbf{A} + |\lambda_n(\mathbf{A})| \cdot \mathbf{I}$, variable partition induced by the vertex partition, and threshold $\rho = k - 1 + |\lambda_n(\mathbf{A})|$. We conclude that MultiChoicePCA is NP-Complete.

D.1.3 Hardness of GraphPathSPCA

We conclude this section showing that GraphPathSPCA is NP-Complete based on a straightforward reduction from MultiChoicePCA.

Consider a MultiChoicePCA input $[\mathbf{A}, \{P_1, \ldots, P_s\}, \rho]$. We construct a graph G on n vertices corresponding to the n variables of MultiChoicePCA (dimension of $\mathbf{A}$) partitioned into k disjoint sets $P_1, \ldots, P_s$. G contains a directed edge from each vertex of P_i to all vertices of P_{i+1}, $i = 1, \ldots, k-1$. Finally, we conceptually augment G with two auxiliary vertices S (source) and T (terminal) and directed edges from S to all vertices in P_1 and directed edges from all vertices of P_s to T.

Any set Q of k variables containing exactly one variable from each set P_i, $i = 1, \ldots, k$, corresponds to an S-T path in G. Conversely, any S-T path in G corresponds to a set of variables with a unique representative from each set P_i, $i = 1, \ldots, k$. Therefore, the constraints in the quadratic maximizations associated with MultiChoicePCA and GraphPathSPCA are operationally identical in this case: the two problems share a common set of feasible solutions and yield the same objective value for each such feasible vector. For any threshold ρ, MultiChoicePCA outputs *yes* if and only if GraphPathSPCA outputs *yes* for the corresponding input. We conclude that MultiChoicePCA can be polynomially reduced to GraphPathSPCA. GraphPathSPCA is clearly in NP, and hence it is NP-Complete.

D.2 Proof of Local Packing Lemma

Towards the proof of Lemma 5.3.4, we develop a modified version of the Varshamov-Gilbert Lemma adapted to our specific model: the set of characteristic vectors of the S-T paths of a (p, s, d)-layer graph G.

Let $\delta_H(\mathbf{x}, \mathbf{y})$ denote the Hamming distance between two points $\mathbf{x}, \mathbf{y} \in \{0,1\}^p$:

$$\delta_H(\mathbf{x}, \mathbf{y}) \triangleq |\{i : x_i \neq y_i\}|.$$

Lemma D.2.48. *Consider a (p, s, d)-layer graph G on p vertices and the collection $\mathcal{P}(G)$ of S-T paths in G. Let*

$$\Omega \triangleq \{\mathbf{x} \in \{0,1\}^p : \ \mathrm{supp}(\mathbf{x}) \in \mathcal{P}(G)\},$$

i.e., the set of characteristic vectors of all S-T paths in G. For every $\xi \in (0,1)$, there exists a set, $\Omega_\xi \subset \Omega$ such that

$$\delta_H(\mathbf{x}, \mathbf{y}) > 2(1-\xi) \cdot s, \quad \forall \mathbf{x}, \mathbf{y} \in \Omega_\xi, \mathbf{x} \neq \mathbf{y}, \tag{D.4}$$

and

$$\log|\Omega_\xi| \geq \log \tfrac{p-2}{s} + (\xi \cdot s - 1) \cdot \log d - s \cdot H(\xi), \tag{D.5}$$

where $H(\cdot)$ is the binary entropy function.

Proof. Consider a labeling $1, \ldots, p$ of the p vertices in G, such that variable ω_i is associated with vertex i. Each point $\boldsymbol{\omega} \in \Omega$ is the characteristic vector of a set in $\mathcal{P}(G)$; nonzero entries of $\boldsymbol{\omega}$ correspond to vertices along an S-T path

in G. With a slight abuse of notation, we refer to $\boldsymbol{\omega}$ as a path in G. Due to the structure of the (p, s, d)-layer graph G, all points in Ω have exactly $k+2$ nonzero entries, *i.e.*,

$$\delta_H(\boldsymbol{\omega}, \mathbf{0}) = k+2, \qquad \forall \boldsymbol{\omega} \in \Omega.$$

Each vertex in $\boldsymbol{\omega}$ lies in a distinct layer of G. In turn, for any pair of points $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega$,

$$\delta_H(\boldsymbol{\omega}, \boldsymbol{\omega}') = 2 \cdot (s - |\{i : \ \omega_i = \omega'_i = 1\}| - 2). \tag{D.6}$$

Note that the Hamming distance between the two points is a linear function of the number of their common nonzero entries, while it can take only even values with a maximum value of $2k$.

Without loss of generality, let S and T corresponding to vertices 1 and p, respectively. Then, the above imply that $\omega_1 = \omega_p = 1, \quad \forall \boldsymbol{\omega} \in \Omega$.

Consider a fixed point $\widehat{\boldsymbol{\omega}} \in \Omega$, and let $\mathcal{B}(\widehat{\boldsymbol{\omega}}, r)$ denote the Hamming ball of radius r centered at $\widehat{\boldsymbol{\omega}}$, *i.e.*,

$$\mathcal{B}(\widehat{\boldsymbol{\omega}}, r) \ \triangleq \ \{\boldsymbol{\omega} \in \{0,1\}^p : \ \delta_H(\widehat{\boldsymbol{\omega}}, \boldsymbol{\omega}) \leq r\}.$$

The intersection $\mathcal{B}(\widehat{\boldsymbol{\omega}}, r) \cap \Omega$ corresponds to S-T paths in G that have at least $(k - r/2)$ additional vertices in common with $\widehat{\boldsymbol{\omega}}$ besides vertices 1 and p that are common to all paths in Ω:

$$\begin{aligned}\mathcal{B}(\widehat{\boldsymbol{\omega}}, r) \cap \Omega &= \{\boldsymbol{\omega} \in \Omega : \delta_H(\widehat{\boldsymbol{\omega}}, \boldsymbol{\omega}) \leq r\} \\ &= \Big\{\boldsymbol{\omega} \in \Omega : |\{i : \ \widehat{\omega}_i = \omega_i = 1\}| \geq s - \tfrac{r}{2} + 2\Big\},\end{aligned}$$

where the last equality is due to (D.6). In fact, due to the structure of G, the set $\mathcal{B}(\widehat{\boldsymbol{\omega}}, r) \cap \Omega$ corresponds to the S-T paths that *meet* $\widehat{\boldsymbol{\omega}}$ in at least $k - r/2$ intermediate layers. Taking into account that $|\Gamma_{\text{in}}(v)| = |\Gamma_{\text{out}}(v)| = d$, for all vertices v in $V(G)$ (except those in the first and last layer),

$$|\mathcal{B}(\widehat{\boldsymbol{\omega}}, r) \cap \Omega| \leq \binom{s}{k - \frac{r}{2}} \cdot d^{k-\left(k-\frac{r}{2}\right)} = \binom{s}{k - \frac{r}{2}} \cdot d^{\frac{r}{2}}.$$

Now, consider a *maximal* set $\Omega_\xi \subset \Omega$ satisfying (D.4), *i.e.*, a set that cannot be augmented by any other point in Ω. The union of balls $\mathcal{B}(\boldsymbol{\omega}, 2(1-\xi)\cdot(s-1))$ over all $\boldsymbol{\omega} \in \Omega_\xi$ covers Ω. To verify that, note that if there exists $\boldsymbol{\omega}' \in \Omega \backslash \Omega_\xi$ such that $\delta_H(\boldsymbol{\omega}, \boldsymbol{\omega}') > 2(1-\xi)\cdot(s-1)$, $\forall \boldsymbol{\omega} \in \Omega_\xi$, then $\Omega_\xi \cup \{\boldsymbol{\omega}'\}$ satisfies (D.4) contradicting the maximality of Ω_ξ. Based on the above,

$$\begin{aligned} |\Omega| &\leq \sum_{\boldsymbol{\omega} \in \Omega_\xi} |\mathcal{B}(\boldsymbol{\omega}, 2(1-\xi)\cdot s) \cap \Omega| \leq \sum_{\mathbf{x} \in \Omega_\xi} \tbinom{s}{k-(1-\xi)k} \cdot d^{(1-\xi)\cdot s} \\ &\leq \sum_{\mathbf{x} \in \Omega_\xi} \tbinom{s}{\xi s} \cdot d^{(1-\xi)\cdot s} \leq |\Omega_\xi| \cdot 2^{s \cdot H(\xi)} \cdot d^{(1-\xi)\cdot s}. \end{aligned}$$

Taking into account that

$$|\Omega| = |\mathcal{P}(G)| = \frac{p-2}{s} \cdot d^{s-1},$$

we conclude that

$$\frac{p-2}{s} \cdot d^{s-1} \leq |\Omega_\xi| \cdot 2^{s \cdot H(\xi)} \cdot d^{(1-\xi)\cdot s},$$

from which the desired result follows. □

Lemma 5.3.4. (Local Packing) *Consider a (p, s, d)-layer graph G on p vertices with $k \geq 4$ and $\log d \geq 4 \cdot H(3/4)$. For any $\epsilon \in (0, 1]$, there exists a set $\mathcal{X}_\epsilon \subset \mathcal{X}(G)$ such that*

$$\epsilon/\sqrt{2} \;<\; \|\mathbf{x}_i - \mathbf{x}_j\|_2 \;\leq\; \sqrt{2} \cdot \epsilon,$$

for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_\epsilon$, $\mathbf{x}_i \neq \mathbf{x}_j$, and

$$\log |\mathcal{X}_\epsilon| \;\geq\; \log \frac{p-2}{s} + \frac{1}{4} \cdot s \log d.$$

Proof. Without loss of generality, consider a labeling $1, \ldots, p$ of the p vertices in G, such that S and T correspond to vertices 1 and p, respectively. Let

$$\Omega \triangleq \{\mathbf{x} \in \{0,1\}^p : \; \text{supp}(\mathbf{x}) \in \mathcal{P}(G)\},$$

where $\mathcal{P}(G)$ is the collection of S-T paths in G. By Lemma D.2.48, and for $\xi = 3/4$, there exists a set $\Omega_\xi \subseteq \Omega$ such that

$$\delta_H(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) > \frac{1}{2} \cdot s, \tag{D.7}$$

$\forall \boldsymbol{\omega}_i, \boldsymbol{\omega}_j \in \Omega_\xi$, $\boldsymbol{\omega}_i \neq \boldsymbol{\omega}_j$, and,

$$\begin{aligned}
\log|\Omega_\xi| &\geq \log \tfrac{p-2}{s} + \left(\tfrac{3}{4} \cdot s - 1\right) \log d - s \cdot H\left(\tfrac{3}{4}\right) \\
&\geq \log \tfrac{p-2}{s} + \tfrac{2}{4} \cdot s \cdot \log d - s \cdot H\left(\tfrac{3}{4}\right) \\
&\geq \log \tfrac{p-2}{s} + \tfrac{1}{4} \cdot s \cdot \log d
\end{aligned} \tag{D.8}$$

where the second and third inequalites hold under the assumptions of the lemma; $k \geq 4$ and $\log d \geq 4 \cdot H(3/4)$.

Consider the bijective mapping $\psi : \Omega_\xi \to \mathbb{R}^p$ defined as

$$\psi(\boldsymbol{\omega}) = \left[\sqrt{\tfrac{(1-\epsilon^2)}{2}} \cdot \omega_1,\ \tfrac{\epsilon}{\sqrt{s}} \cdot \boldsymbol{\omega}_{2:p-1},\ \sqrt{\tfrac{(1-\epsilon^2)}{2}} \cdot \omega_p\right].$$

We show that the set

$$\mathcal{X}_\epsilon \triangleq \{\psi(\boldsymbol{\omega}) :\ \boldsymbol{\omega} \in \Omega_\xi\}.$$

has the desired properties. First, to verify that $\mathcal{X}_\epsilon$ is a subset of $\mathcal{X}(G)$, note that $\forall \boldsymbol{\omega} \in \Omega_\xi \subset \Omega$,

$$\text{supp}(\psi(\boldsymbol{\omega})) = \text{supp}(\boldsymbol{\omega}) \in \mathcal{P}(G), \tag{D.9}$$

and

$$\|\psi(\boldsymbol{\omega})\|_2^2 = 2 \cdot \frac{(1-\epsilon^2)}{2} + \frac{\epsilon^2}{s} \cdot \sum_{i=2}^{p-1} \omega_i = 1.$$

Second, for all pairs of points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_\epsilon$,

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \delta_H(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) \cdot \frac{\epsilon^2}{s} \leq 2 \cdot k \cdot \frac{\epsilon^2}{s} = 2 \cdot \epsilon^2.$$

The inequality follows from the fact that $\delta_H(\boldsymbol{\omega}, \mathbf{0}) = k + 2\ \omega_1 = 1$ and $\omega_p = 1$, $\forall \boldsymbol{\omega} \in \Omega_\xi$, and in turn

$$\delta_H(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) \leq 2 \cdot s.$$

Similarly, for all pairs $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_\epsilon$, $\mathbf{x}_i \neq \mathbf{x}_j$,

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \delta_H(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) \cdot \frac{\epsilon^2}{s} \geq \frac{1}{2} \cdot k \cdot \frac{\epsilon^2}{s} = \frac{\epsilon^2}{2},$$

where the inequality is due to (D.7). Finally, the lower bound on the cardinality of $\mathcal{X}_\epsilon$ follows immediately from (D.8) and the fact that $|\mathcal{X}_\epsilon| = |\Omega_\xi|$, which completes the proof. □

D.3 Auxiliary Lemmas

Lemma D.3.49. *Let $X_1, \ldots, X_n$ be n b-subgaussian real random variables for some $b \geq 0$, i.e., $\mathbb{E}[\exp(\lambda X_i)] \leq e^{b^2\lambda^2/2}$ for every $\lambda \in \mathbb{R}$. (Note that Gaussian $\mathcal{N}(0, \sigma^2)$ random variables are subgaussian with parameter $b = \sigma$). Then,*

$$\mathbb{E}\left[\max_{i\in[n]} X_i\right] \leq \sqrt{2b^2 \log n}. \tag{D.10}$$

Proof. Define the random variable $Z = \max_{i\in[n]} X_i$. By the convexity of the exponential function and Jensen's inequality,

$$\exp(\mathbb{E}[\lambda Z]]) \leq \mathbb{E}[\exp(\lambda Z)], \quad \forall \lambda \in \mathbb{R}. \tag{D.11}$$

Moreover,

$$\mathbb{E}[\exp(\lambda Z)] = \mathbb{E}\left[\exp\left(\lambda \max_{i\in[n]} X_i\right)\right] \leq \sum_{i=1}^{n} \mathbb{E}[\exp(\lambda X_i)] \leq n \exp\left(\lambda^2 b^2/2\right),$$

where the first inequality is due to the fact that the exponential function is positive, and the second follows from the assumptions on the distribution of X_i, $i = 1, \ldots, n$. Combining the two, we find

$$\mathbb{E}[Z] \leq \frac{\log n}{\lambda} + \lambda b^2/2, \quad \forall \lambda \in \mathbb{R}.$$

Setting $\lambda = \sqrt{2 \log n / b^2}$, we obtain

$$\mathbb{E}[Z] \leq \frac{\sqrt{b^2 \log n}}{\sqrt{2}} + \frac{\sqrt{b^2 \log n}}{\sqrt{2}} = \sqrt{2b^2 \log n},$$

which is the desired result. □

Appendix E

Appendix for Chapter 6

E.1 Proof of NP-Hardness

We provide a proof for the NP-hardness of the constrained (and specifically sparse) CCA problem via a reduction from sparse PCA. Recall that sparse PCA is the following optimization problem:

$$\max_{\substack{\mathbf{x}:\|\mathbf{x}\|_0=k\\ \|\mathbf{x}\|_2=1}} \mathbf{x}^\top \mathbf{A}\mathbf{x}, \tag{E.1}$$

where k is a given parameter and $\mathbf{A}$ a given $n \times n$ positive semidefinite (PSD) matrix.

We show that the sparse PCA problem (E.1) reduces to the sparse CCA problem (6.2) and in particular the maximization

$$\max_{\substack{\mathbf{x}:\|\mathbf{x}\|_0=k,\|\mathbf{x}\|_2=1\\ \mathbf{y}:\|\mathbf{y}\|_0=k,\|\mathbf{y}\|_2=1}} \mathbf{x}^\top \mathbf{A}\mathbf{y}. \tag{E.2}$$

The only difference between (E.1) and (E.2) is that in the latter the optimal values for the two variables $\mathbf{x}$ and $\mathbf{y}$ may be different. If we add the constraint $\mathbf{x} = \mathbf{y}$ in (E.2), then then two maximizations are identical. We show that this is not necessary: since $\mathbf{A}$ is PSD, the optimal solution of (E.2) will inherently satisfy $\mathbf{x} = \mathbf{y}$, and in turn the two maximizations are equivalent.

Let $\mathbf{U}, \mathbf{\Lambda}$ be the eigenvalue decomposition of $\mathbf{A}$: the $n \times n$ matrix $\mathbf{U}$ contains the eigenvectors, while the $n \times n$ diagonal $\mathbf{\Lambda}$ contains the eigenvalues $\lambda_1, \ldots, \lambda_n \geq 0$ in decreasing order. Let $(\mathbf{x}_\star, \mathbf{y}_\star)$ be the optimal solution of (E.2). Further, let $\overline{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}_\star$, and $\overline{\mathbf{y}} = \mathbf{U}^\top \mathbf{y}_\star$. Then,

$$\mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star = \mathbf{x}_\star^\top \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top \mathbf{y}_\star = \overline{\mathbf{x}}^\top \mathbf{\Lambda} \overline{\mathbf{y}} = \sum_{i=1}^n \overline{x}_i \overline{y}_i \lambda_i. \tag{E.3}$$

Theorem E.14 (Weighted Cauchy-Schwarz inequality; [46], Theorem 10.1)**.** *Let $a_i, b_i \in \mathbb{R}$ be real numbers and let $m_i \in \mathbb{R}^+$, $i = 1, 2, \ldots, n$. Then,*

$$\left(\sum_{i=1}^n a_i b_i m_i \right)^2 \leq \left(\sum_{i=1}^n a_i^2 m_i \right) \left(\sum_{i=1}^n b_i^2 m_i \right).$$

Equality occurs if and only if $\frac{a_1}{b_2} = \ldots = \frac{a_n}{b_n}$.

By Theorem E.14,

$$\left(\mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star \right)^2 \leq \left(\sum_{i=1}^n \overline{x}_i^2 \lambda_i \right) \left(\sum_{i=1}^n \overline{y}_i^2 \lambda_i \right),$$

with equality if and only if there exists a constant $c \in \mathbb{R}$ such that $\overline{\mathbf{x}} = c \cdot \overline{\mathbf{y}}$, and taking into account that both $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are unit norm vectors, it follows that $\overline{\mathbf{x}} = \overline{\mathbf{y}}$. Finally, since $\mathbf{U}$ is a full rank matrix (in fact orthonormal basis of $\mathbb{R}^n$), it follows that $\mathbf{x}_\star = \mathbf{y}_\star$ must hold.

E.2 Proofs

For the remainder of this section, we define

$$(\mathbf{x}_\star, \mathbf{y}_\star) \triangleq \operatorname*{argmax}_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}} \mathbf{x}^\top \mathbf{A} \mathbf{y},$$

i.e., $\mathbf{x}_\star, \mathbf{y}_\star$ is a feasible pair that maximizes –not necessarily uniquely– the objective. Further, we assume that there exists procedures to compute the exact soluton to $\mathsf{P}_{\mathcal{X}}(\cdot)$ and $\mathsf{P}_{\mathcal{Y}}(\cdot)$ in (6.5) and (6.6), running in time $\mathrm{T}_{\mathcal{X}}$ and $\mathrm{T}_{\mathcal{Y}}$, respectively. The following results can be easily adapted for the case where these procedures yield approximate solutions.

Lemma E.2.50. *For any real $m \times n$ matrix $\mathbf{A}$ with rank$(\mathbf{A}) = r \leq \max\{m, n\}$ and $\epsilon \in (0, 1)$, Algorithm 6.12 with input $\mathbf{A}$, r, and $\mathrm{T} = \widetilde{O}\left(2^{r \cdot \log_2(2/\epsilon)}\right)$ outputs $\mathbf{x}_\sharp \in \mathcal{X}$ and $\mathbf{y}_\sharp \in \mathcal{Y}$ such that*

$$\mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp \geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \epsilon \cdot \sigma_1(\mathbf{A}),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{X}} + \mathrm{T}_{\mathcal{Y}} + r \cdot \max\{m, n\}))$.

Proof. In the sequel, $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$ are used to denote the r-truncated singular value decomposition of $\mathbf{A}$. Note that the lemma assumes that the accuracy parameter r is equal to the rank of the input matrix $\mathbf{A}$ and hence $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.

Recall that $\mathbf{x}_\star, \mathbf{y}_\star$ is a pair —not necessarily unique— that maximizes the objective $\mathbf{x}^\top \mathbf{A} \mathbf{y}$ over all feasible solutions. Define $\mathbf{c}_\star \triangleq \mathbf{V}^\top \mathbf{y}_\star$. Note that $\mathbf{c}_\star$ is a vector in $\mathbf{R}^{r \times 1}$ with $\|\mathbf{c}_\star\|_2 \leq 1$ since the r columns of $\mathbf{V}$ are orthonormal and $\|\mathbf{y}_\star\|_2 = 1$. Finally, let $\overline{\mathbf{c}}_\star \triangleq \mathbf{c}_\star / \|\mathbf{c}_\star\|_2$. Note that $\|\mathbf{c}_\star\|_2 > 0$ since by assumption $\mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star > 0$.

Algorithm 6.12 operates in an iterative fashion. In each iteration, it independently considers a point $\mathbf{c}$ selected randomly and uniformly from the r-dimensional ℓ_2-unit sphere $\mathbb{S}_2^{r-1}$ and generates a candidate solution pair at

each point. For $\mathrm{T} = \widetilde{O}\left(2^{r\cdot\log_2(2/\epsilon)}\right)$, the collection of randomly sampled poins forms an $\epsilon/2$-net for $\mathbb{S}_2^{r-1}$. By definition, the $\epsilon/2$-net contains a point $\widetilde{\mathbf{c}} \in \mathbb{R}^{r\times 1}$, such that

$$\|\widetilde{\mathbf{c}} - \overline{\mathbf{c}}_\star\|_2 \leq \epsilon/2. \tag{E.4}$$

Let $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ be the candidate solution pair computed at $\widetilde{\mathbf{c}}$ by the two step maximization procedure, *i.e.*, let

$$\widetilde{\mathbf{x}} \triangleq \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\, \mathbf{x}^\top \mathbf{U}\boldsymbol{\Sigma}\widetilde{\mathbf{c}} \tag{E.5}$$

and

$$\widetilde{\mathbf{y}} \triangleq \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, \widetilde{\mathbf{x}}^\top \mathbf{A}\mathbf{y}. \tag{E.6}$$

By the definition of $\mathbf{c}_\star$, and letting $\rho \triangleq \|\mathbf{c}_\star\|_2$

$$\begin{aligned}
\mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star &= \mathbf{x}_\star^\top \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_\star \\
&= \rho \cdot \mathbf{x}_\star^\top \mathbf{U}\boldsymbol{\Sigma}\overline{\mathbf{c}}_\star \\
&= \rho \cdot \mathbf{x}_\star^\top \mathbf{U}\boldsymbol{\Sigma}\widetilde{\mathbf{c}} + \rho \cdot \mathbf{x}_\star^\top \mathbf{U}\boldsymbol{\Sigma}(\overline{\mathbf{c}}_\star - \widetilde{\mathbf{c}}) \\
&\leq \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\boldsymbol{\Sigma}\widetilde{\mathbf{c}} + \rho \cdot \mathbf{x}_\star^\top \mathbf{U}\boldsymbol{\Sigma}(\overline{\mathbf{c}}_\star - \widetilde{\mathbf{c}}) \\
&\leq \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\boldsymbol{\Sigma}\widetilde{\mathbf{c}} + \tfrac{\epsilon}{2} \cdot \sigma_1(\mathbf{A}).
\end{aligned} \tag{E.7}$$

The first inequality follows from the fact that $\widetilde{\mathbf{x}}$ by definition maximizes the first term over all $\mathbf{x} \in \mathcal{X}$. The last inequality is due to Lemma E.3.55 and the fact that $\|\mathbf{x}_\star\|_2 = 1$ and $\rho \leq 1$. We further upper bound the right hand side

of (E.7) as follows:

$$\begin{aligned}
\rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}} &= \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\overline{\mathbf{c}}_\star + \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}(\widetilde{\mathbf{c}} - \overline{\mathbf{c}}_\star) \\
&= \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\mathbf{c}_\star + \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}(\widetilde{\mathbf{c}} - \overline{\mathbf{c}}_\star) \\
&= \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \mathbf{y}_\star + \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}(\widetilde{\mathbf{c}} - \overline{\mathbf{c}}_\star) \\
&\leq \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \widetilde{\mathbf{y}} + \rho \cdot \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}(\widetilde{\mathbf{c}} - \overline{\mathbf{c}}_\star) \qquad \text{(E.8)} \\
&\leq \widetilde{\mathbf{x}}^\top \mathbf{A}\widetilde{\mathbf{y}} + \tfrac{\epsilon}{2} \cdot \sigma_1(\mathbf{A}). \qquad \text{(E.9)}
\end{aligned}$$

Inequality (E.8) follows by the fact that $\widetilde{\mathbf{y}}$ by definition (E.6) maximizes the bilinear term $\mathbf{x}^\top \mathbf{A}\mathbf{y}$ over all $\mathbf{y} \in \mathcal{Y}$ when $\mathbf{x} = \widetilde{\mathbf{x}}$. The last inequality is once again due to Lemma E.3.55 and the fact that $\|\mathbf{x}_\star\|_2 = 1$ and $\rho \leq 1$. Combining (E.7) and (E.9), we obtain

$$\widetilde{\mathbf{x}}^\top \mathbf{A}\widetilde{\mathbf{y}} \geq \mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star - \epsilon \cdot \sigma_1(\mathbf{A}).$$

Algorithm 6.12 computes multiple candidate solution pairs and outputs the one that maximizes the objective. Therefore, the output pair $(\mathbf{x}_\sharp, \mathbf{y}_\sharp)$ must achieve a value as least as high as that achieved by $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$, which implies the desired guarantee.

The running time of Algorithm 6.12 follows straightforwardly by inspection. The algorithm first computes the truncated singular value decomposition of inner dimension r in time denoted by $\mathrm{T}_{\mathtt{svd}}(r)$. Subsequently, it performs T iterations. The cost of each iteration is determined by the cost of the matrix-vector multiplications and the running times $\mathrm{T}_{\mathcal{X}}$ and $\mathrm{T}_{\mathcal{Y}}$ of the operators $\mathsf{P}_{\mathcal{X}}(\cdot)$ and $\mathsf{P}_{\mathcal{Y}}(\cdot)$. Note that matrix multiplications can exploit the available

singular value decomposition of $\mathbf{A}$ and are performed in time $r \cdot \max\{m, n\}$. Substituting the value of T, completes the proof. □

Theorem 6.10. *For any real $m \times n$ matrix $\mathbf{A}$, $\epsilon \in (0, 1)$, and $r \leq \max\{m, n\}$, Algorithm 6.12 with input $\mathbf{A}$, r, and $\mathrm{T} = \widetilde{O}\left(2^{r \cdot \log_2(2/\epsilon)}\right)$ outputs $\mathbf{x}_\sharp \in \mathcal{X}$ and $\mathbf{y}_\sharp \in \mathcal{Y}$ such that*

$$\mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp \geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \epsilon \cdot \sigma_1(\mathbf{A}) - 2 \cdot \sigma_{r+1}(\mathbf{A}),$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{X}} + \mathrm{T}_{\mathcal{Y}} + r \cdot \max\{m, n\}))$.

Proof. Recall that Algorithm 6.12 with input an $m \times n$ matrix $\mathbf{A}$ and accuracy parameter r, first computes a rank-r truncated singular value decomposition $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ and operates on that principal subspace of $\mathbf{A}$. Let $\mathbf{B}$ be the $m \times n$ best rank-r approximation of $\mathbf{A}$ under the spectral norm. Then $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. One can easily verify that running Algorithm 6.12 with input $\mathbf{A}$ and accuracy parameter r, is equivalent to applying the algorithm on $\mathbf{B}$ with the same parameters.

By Lemma E.2.50, Algorithm 6.12 outputs $\mathbf{x}_\sharp, \mathbf{y}_\sharp$ such that

$$\mathbf{x}_\sharp^\top \mathbf{B} \mathbf{y}_\sharp \geq \widehat{\mathbf{x}}_\star^\top \mathbf{B} \widehat{\mathbf{y}}_\star - \epsilon \cdot \sigma_1(\mathbf{B}), \tag{E.10}$$

where

$$(\widehat{\mathbf{x}}_\star, \widehat{\mathbf{y}}_\star) \triangleq \operatorname*{argmax}_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}} \mathbf{x}^\top \mathbf{B} \mathbf{y}$$

is a pair that optimally solves the maximization on the rank-r matrix $\mathbf{B}$. By the optimality of the pair $\widehat{\mathbf{x}}_\star, \widehat{\mathbf{y}}_\star$ for the rank-r problem, it follows that

$$\widehat{\mathbf{x}}_\star^\top \mathbf{B} \widehat{\mathbf{y}}_\star \geq \mathbf{x}_\star^\top \mathbf{B} \mathbf{y}_\star. \tag{E.11}$$

Recall that $\mathbf{x}_\star$, $\mathbf{y}_\star$ is the pair that optimally solves the maximization on the original input matrix $\mathbf{A}$. Further,

$$\begin{aligned} \mathbf{x}_\star^\top \mathbf{B} \mathbf{y}_\star &= \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \mathbf{x}_\star^\top (\mathbf{A} - \mathbf{B}) \mathbf{y}_\star \\ &\geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \left| \mathbf{x}_\star^\top (\mathbf{A} - \mathbf{B}) \mathbf{y}_\star \right| \\ &\geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \sigma_{r+1}(\mathbf{A}). \end{aligned} \tag{E.12}$$

Combining (E.12) with (E.10) and (E.11),

$$\mathbf{x}_\sharp^\top \mathbf{B} \mathbf{y}_\sharp \geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - \sigma_{r+1}(\mathbf{A}) - \epsilon \cdot \sigma_1(\mathbf{B}).$$

Finally,

$$\begin{aligned} \mathbf{x}_\sharp^\top \mathbf{B} \mathbf{y}_\sharp &= \mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp - \mathbf{x}_\sharp^\top (\mathbf{A} - \mathbf{B}) \mathbf{y}_\sharp \\ &\leq \mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp + \left| \mathbf{x}_\sharp^\top (\mathbf{A} - \mathbf{B}) \mathbf{y}_\sharp \right|. \\ &\leq \mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp + \sigma_{r+1}(\mathbf{A}). \end{aligned} \tag{E.13}$$

Combining with the previous inequality, we obtain

$$\mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp \geq \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star - 2 \cdot \sigma_{r+1}(\mathbf{A}) - \epsilon \cdot \sigma_1(\mathbf{B}).$$

Noting that $\sigma_1(\mathbf{B}) = \sigma_1(\mathbf{A})$ completes the proof of the approximation guarantee. The running time of the algorithm is established in Lemma E.2.50. □

Lemma E.2.51. *For any real $m \times n$ matrix $\mathbf{A}$ with rank$(\mathbf{A}) = r \leq \max\{m, n\}$ and $\epsilon \in (0, 1)$, if $\mathcal{Y} = \{\mathbf{y} : \|\mathbf{y}\|_2 = 1\}$, then Algorithm 6.12 with input $\mathbf{A}$, r, and $\mathrm{T} = \widetilde{O}\left(2^{r \cdot \log_2(2/\epsilon)}\right)$ outputs $\mathbf{x}_\sharp \in \mathcal{X}$ and $\mathbf{y}_\sharp \in \mathcal{Y}$ such that*

$$\mathbf{x}_\sharp^\top \mathbf{A} \mathbf{y}_\sharp \geq (1 - \epsilon) \cdot \mathbf{x}_\star^\top \mathbf{A} \mathbf{y}_\star$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{X}} + \mathrm{T}_{\mathcal{Y}} + r \cdot \max\{m, n\}))$.

Proof. The lemma focuses on the special case where the feasible region for the variable $\mathbf{y}$ coincides with the set of vectors with unit ℓ_2 norm, *i.e.*,

$$\mathcal{Y} = \{\mathbf{y} : \|\mathbf{y}\|_2 = 1\}. \tag{E.14}$$

The feasible region $\mathcal{X}$ for $\mathbf{x}$ is arbitrary, assuming once again that there exists an efficient operator $\mathsf{P}_{\mathcal{X}}(\cdot)$.

By the Cauchy-Schwarz inequality, for any $\mathbf{x}_0 \in \mathbb{R}^{m \times 1}$,

$$\mathbf{x}_0^\top \mathbf{A} \mathbf{y} \leq \|\mathbf{x}_0^\top \mathbf{A}\|_2, \quad \forall \mathbf{y} \in \mathcal{Y}. \tag{E.15}$$

In fact, equality is achieved when $\mathbf{y}$ is aligned with $\mathbf{A}^\top \mathbf{x}_0$, *i.e.*, for $\mathbf{y} = \mathbf{A}^\top \mathbf{x}_0 / \|\mathbf{A}^\top \mathbf{x}_0\|_2 \in \mathcal{Y}$. In turn, for any $\mathbf{x}_0 \in \mathbb{R}^{m \times 1}$,

$$\begin{aligned} \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}_0^\top \mathbf{A} \mathbf{y} &= \mathbf{x}_0^\top \mathbf{A} \mathbf{A}^\top \mathbf{x}_0 / \|\mathbf{A}^\top \mathbf{x}_0\|_2 \\ &= \|\mathbf{A}^\top \mathbf{x}_0\|_2 \\ &= \|\mathbf{V} \mathbf{\Sigma} \mathbf{U}^\top \mathbf{x}_0\|_2 \\ &= \|\mathbf{\Sigma} \mathbf{U}^\top \mathbf{x}_0\|_2, \end{aligned} \tag{E.16}$$

where the last equality follows from the fact that the r columns of $\mathbf{V}$ are orthonormal.

We now proceed in a fashion very similar to that in the proof of Lemma E.2.50. Recall that $\mathbf{x}_\star, \mathbf{y}_\star$ is a pair that maximizes –not necessarily uniquely– the objective $\mathbf{x}^\top \mathbf{A}\mathbf{y}$ over all feasible solutions, and define $\mathbf{c}_\star \triangleq \mathbf{V}^\top \mathbf{y}_\star$. Note that here,

$$\begin{aligned}\mathbf{c}_\star \triangleq \mathbf{V}^\top \mathbf{y}_\star &= \mathbf{V}^\top \mathbf{A}^\top \mathbf{x}_\star / \|\mathbf{A}^\top \mathbf{x}_\star\|_2 \\ &= \mathbf{\Sigma}\mathbf{U}^\top \mathbf{x}_\star / \|\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{x}_\star\|_2 \\ &= \mathbf{\Sigma}\mathbf{U}^\top \mathbf{x}_\star / \|\mathbf{\Sigma}\mathbf{U}^\top \mathbf{x}_\star\|_2 \end{aligned} \tag{E.17}$$

and hence, $\|\mathbf{c}_\star\|_2 = 1$. Following similar reasoning as in the proof of Lemma E.2.50, Algorithm 6.12 considers a point $\widetilde{\mathbf{c}} \in \mathbb{R}^{r \times 1}$, such that

$$\|\widetilde{\mathbf{c}} - \overline{\mathbf{c}}_\star\|_2 \leq \epsilon.$$

Let $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ be the candidate solution pair computed at $\widetilde{\mathbf{c}}$ by the two step maximization procedure. We have,

$$\begin{aligned}\mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star &= \mathbf{x}_\star^\top \mathbf{U}\mathbf{\Sigma}\mathbf{c}_\star \\ &= \mathbf{x}_\star^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}} + \mathbf{x}_\star^\top \mathbf{U}\mathbf{\Sigma}(\mathbf{c}_\star - \widetilde{\mathbf{c}}) \\ &\leq \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}} + \|\mathbf{x}_\star^\top \mathbf{U}\mathbf{\Sigma}\|_2 \|\mathbf{c}_\star - \widetilde{\mathbf{c}}\|_2 \\ &\leq \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}} + \epsilon \cdot \|\mathbf{x}_\star^\top \mathbf{U}\mathbf{\Sigma}\|_2. \end{aligned} \tag{E.18}$$

where the first inequality follows from the fact that $\widetilde{\mathbf{x}}$ by definition maximizes the first term at $\widetilde{\mathbf{c}}$ over all $\mathbf{x} \in \mathcal{X}$ and the Cauchy-Schwarz inequality. The key

difference from the proof of Lemma E.2.50, is that the term $\|\mathbf{x}_\star^\top \mathbf{U}\mathbf{\Sigma}\|_2$ in the right-hand side coincides with the optimal objective value $\mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star$ as follows from (E.16). For comparison, note that in the proof of Lemma E.2.50 it was loosely upper bounded by $\sigma_1(\mathbf{A})$. Continuing from (E.18),

$$(1-\epsilon)\cdot \mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star \leq \widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}}. \tag{E.19}$$

But, once again by the Cauchy-Schwarz inequality,

$$\begin{aligned}\widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}} &\leq \|\widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\|_2 \|\widetilde{\mathbf{c}}\|_2 \\ &= \|\widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\|_2 \end{aligned} \tag{E.20}$$

and by (E.16),

$$\widetilde{\mathbf{x}}^\top \mathbf{U}\mathbf{\Sigma}\widetilde{\mathbf{c}} = \max_{\mathbf{y}\in\mathcal{Y}} \widetilde{\mathbf{x}}^\top \mathbf{A}\mathbf{y} = \widetilde{\mathbf{x}}^\top \mathbf{A}\widetilde{\mathbf{y}}. \tag{E.21}$$

Combining (E.21) with (E.19),

$$\widetilde{\mathbf{x}}^\top \mathbf{A}\widetilde{\mathbf{y}} \geq (1-\epsilon)\cdot \mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star \tag{E.22}$$

Recalling that Algorithm 6.12 outputs the candidate pair that maximizes the objective among all computed feasible points implies the desired result. □

Theorem 6.11. *For any real $m \times n$ matrix $\mathbf{A}$ and $\epsilon \in (0,1)$, if $\mathcal{Y} = \{\mathbf{y} : \|\mathbf{y}\|_2 = 1\}$, then Algorithm 6.12 with input $\mathbf{A}$, r, and $\mathrm{T} = \widetilde{O}\left(2^{r\cdot \log_2(2/\epsilon)}\right)$ outputs $\mathbf{x}_\sharp \in \mathcal{X}$ and $\mathbf{y}_\sharp \in \mathcal{Y}$ such that*

$$\mathbf{x}_\sharp^\top \mathbf{A}\mathbf{y}_\sharp \geq (1-\epsilon)\cdot \mathbf{x}_\star^\top \mathbf{A}\mathbf{y}_\star - 2\cdot \sigma_{r+1}(\mathbf{A})$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O(\mathrm{T}\cdot(\mathrm{T}_{\mathcal{X}} + \mathrm{T}_{\mathcal{Y}} + r\cdot\max\{m,n\}))$.

Proof. The Theorem follows from Lemma E.2.51. The proof is similar to that of Theorem 6.10. The main difference lies in substituting (E.10) with

$$\mathbf{x}_\sharp^\top \mathbf{B}\mathbf{y}_\sharp \geq (1-\epsilon)\cdot \mathbf{x}_{(B)\star}^\top \mathbf{B}\mathbf{y}_{(B)\star}. \tag{E.23}$$

The remainder of the proof easily follows. □

E.3 Auxiliary Lemmas

Lemma E.3.52. *Let $a_1,\ldots,a_n$ and $b_1,\ldots,b_n$ be $2n$ real numbers and let p and q be two numbers such that $1/p+1/q=1$ and $p>1$. We have*

$$\left|\sum_{i=1}^n a_i b_i\right| \leq \left(\sum_{i=1}^n |a_i|^p\right)^{1/p}\cdot\left(\sum_{i=1}^n |b_i|^q\right)^{1/q}.$$

Lemma E.3.53. *For any $\mathbf{A},\mathbf{B}\in\mathbb{R}^{n\times k}$,*

$$|\langle \mathbf{A},\mathbf{B}\rangle| \triangleq \left|\mathrm{TR}\left(\mathbf{A}^\top\mathbf{B}\right)\right| \leq \|\mathbf{A}\|_{\mathrm{F}}\|\mathbf{B}\|_{\mathrm{F}}.$$

Proof. Treating $\mathbf{A}$ and $\mathbf{B}$ as vectors, the lemma follows immediately from Lemma E.3.52 for $p=q=2$. □

Lemma E.3.54. *For any two real matrices $\mathbf{A}$ and $\mathbf{B}$ of appropriate dimensions, $\|\mathbf{AB}\|_{\mathrm{F}} \leq \min\{\|\mathbf{A}\|_2\|\mathbf{B}\|_{\mathrm{F}},\ \|\mathbf{A}\|_{\mathrm{F}}\|\mathbf{B}\|_2\}$.*

Proof. Let $\mathbf{b}_i$ denote the ith column of $\mathbf{B}$. Then,

$$\begin{aligned}\|\mathbf{AB}\|_{\mathrm{F}}^2 &= \sum_i \|\mathbf{A}\mathbf{b}_i\|_2^2 \leq \sum_i \|\mathbf{A}\|_2^2\|\mathbf{b}_i\|_2^2 \\ &= \|\mathbf{A}\|_2^2\sum_i \|\mathbf{b}_i\|_2^2 = \|\mathbf{A}\|_2^2\|\mathbf{B}\|_{\mathrm{F}}^2.\end{aligned}$$

Similarly, using the previous inequality,

$$\|\mathbf{AB}\|_{\mathrm{F}}^2 = \|\mathbf{B}^\top \mathbf{A}^\top\|_{\mathrm{F}}^2 \leq \|\mathbf{B}^\top\|_2^2 \|\mathbf{A}^\top\|_{\mathrm{F}}^2 = \|\mathbf{B}\|_2^2 \|\mathbf{A}\|_{\mathrm{F}}^2.$$

The desired result follows combining the two upper bounds. □

Lemma E.3.55. *For any real $m \times k$ matrix $\mathbf{X}$, $m \times n$ matrix $\mathbf{A}$, and $n \times k$ matrix $\mathbf{Y}$, $\left|\mathrm{Tr}\left(\mathbf{X}^\top \mathbf{AY}\right)\right| \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{A}\|_2 \cdot \|\mathbf{Y}\|_{\mathrm{F}}$.*

Proof. We have

$$\left|\mathrm{Tr}\left(\mathbf{X}^\top \mathbf{AY}\right)\right| \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{AY}\|_{\mathrm{F}} \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{A}\|_2 \cdot \|\mathbf{Y}\|_{\mathrm{F}},$$

with the first inequality following from Lemma E.3.53 on $|\langle \mathbf{X}, \mathbf{AY} \rangle|$ and the second from Lemma E.3.54. □

Lemma E.3.56. *For any real $m \times n$ matrix $\mathbf{A}$, and pair of $m \times k$ matrix $\mathbf{X}$ and $n \times k$ matrix $\mathbf{Y}$ such that $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_k$ and $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k$ with $k \leq \min\{m, n\}$, the following holds:*

$$\left|\mathrm{Tr}\left(\mathbf{X}^\top \mathbf{AY}\right)\right| \leq \sqrt{k} \cdot \left(\sum_{i=1}^{k} \sigma_i^2(\mathbf{A})\right)^{1/2}.$$

Proof. By Lemma E.3.53,

$$|\langle \mathbf{X}, \mathbf{AY} \rangle| = \left|\mathrm{Tr}\left(\mathbf{X}^\top \mathbf{AY}\right)\right| \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{AY}\|_{\mathrm{F}} = \sqrt{k} \cdot \|\mathbf{AY}\|_{\mathrm{F}}.$$

where the last inequality follows from the fact that $\|\mathbf{X}\|_{\mathrm{F}}^2 = \mathrm{Tr}\left(\mathbf{X}^\top \mathbf{X}\right) = \mathrm{Tr}(\mathbf{I}_k) = k$. Further, for any $\mathbf{Y}$ such that $\mathbf{Y}^T \mathbf{Y} = \mathbf{I}_k$,

$$\|\mathbf{AY}\|_{\mathrm{F}}^2 \leq \max_{\substack{\widehat{\mathbf{Y}} \in \mathbb{R}^{n \times k} \\ \widehat{\mathbf{Y}}^\top \widehat{\mathbf{Y}} = \mathbf{I}_k}} \|\mathbf{A}\widehat{\mathbf{Y}}\|_{\mathrm{F}}^2 = \sum_{i=1}^{k} \sigma_i^2(\mathbf{A}). \tag{E.24}$$

Combining the two inequalities, the result follows. □

Lemma E.3.57. *For any real $m \times n$ matrix $\mathbf{A}$, and any $k \leq \min\{m,\, n\}$,*

$$\max_{\substack{\mathbf{Y}\in\mathbb{R}^{n\times k}\\ \mathbf{Y}^\top\mathbf{Y}=\mathbf{I}_k}} \|\mathbf{AY}\|_{\mathrm{F}} = \left(\sum_{i=1}^{k}\sigma_i^2(\mathbf{A})\right)^{1/2}.$$

The above equality is realized when the k columns of $\mathbf{Y}$ coincide with the k leading right singular vectors of $\mathbf{A}$.

Proof. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the singular value decomposition of $\mathbf{A}$; $\mathbf{U}$ and $\mathbf{V}$ are $m \times m$ and $n \times n$ unitary matrices respectively, while Σ is a diagonal matrix with $\Sigma_{jj} = \sigma_j$, the jth largest singular value of $\mathbf{A}$, $j = 1, \ldots, d$, where $d \triangleq \min\{m, n\}$. Due to the invariance of the Frobenius norm under unitary multiplication,

$$\|\mathbf{AY}\|_{\mathrm{F}}^2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Y}\|_{\mathrm{F}}^2 = \|\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Y}\|_{\mathrm{F}}^2. \tag{E.25}$$

Continuing from (E.25),

$$\begin{aligned}\|\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Y}\|_{\mathrm{F}}^2 &= \mathrm{Tr}\Big(\mathbf{Y}^\top\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top\mathbf{Y}\Big)\\ &= \sum_{i=1}^{k}\mathbf{y}_i^\top\left(\sum_{j=1}^{d}\sigma_j^2\cdot\mathbf{v}_j\mathbf{v}_j^\top\right)\mathbf{y}_i\\ &= \sum_{j=1}^{d}\sigma_j^2\cdot\sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2.\end{aligned}$$

Let $z_j \triangleq \sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2$, $j = 1, \ldots, d$. Note that each individual z_j satisfies

$$0 \leq z_j \triangleq \sum_{i=1}^{k}\Big(\mathbf{v}_j^\top\mathbf{y}_i\Big)^2 \leq \|\mathbf{v}_j\|^2 = 1,$$

where the last inequality follows from the fact that the columns of $\mathbf{Y}$ are orthonormal. Further,

$$\sum_{j=1}^{d} z_j = \sum_{j=1}^{d} \sum_{i=1}^{k} \left(\mathbf{v}_j^\top \mathbf{y}_i\right)^2 = \sum_{i=1}^{k} \sum_{j=1}^{d} \left(\mathbf{v}_j^\top \mathbf{y}_i\right)^2 = \sum_{i=1}^{k} \|\mathbf{y}_i\|^2 = k.$$

Combining the above, we conclude that

$$\|\mathbf{AY}\|_{\mathrm{F}}^2 = \sum_{j=1}^{d} \sigma_j^2 \cdot z_j \leq \sigma_1^2 + \ldots + \sigma_k^2. \tag{E.26}$$

Finally, it is straightforward to verify that if $\mathbf{y}_i = \mathbf{v}_i$, $i = 1, \ldots, k$, then (E.26) holds with equality. □

Appendix F

Appendix for Chapter 7

F.1 Proofs

Lemma 7.4.11. *For any real $m \times n$, rank-r matrix $\mathbf{A}_r$ and arbitrary norm-bounded sets $\mathcal{X} \subset \mathbb{R}^{m\times k}$ and $\mathcal{Y} \subset \mathbb{R}^{n\times k}$, let*

$$(\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}) \triangleq \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}} \mathrm{Tr}\Big(\mathbf{X}^\top \mathbf{A}_r \mathbf{Y}\Big).$$

If there exist operators $\mathrm{P}_\mathcal{X} : \mathbb{R}^{m\times k} \to \mathcal{X}$ such that

$$\mathrm{P}_\mathcal{X}(\mathbf{L}) = \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X}} \mathrm{Tr}\Big(\mathbf{X}^\top \mathbf{L}\Big)$$

and similarly, $\mathrm{P}_\mathcal{Y} : \mathbb{R}^{n\times k} \to \mathcal{Y}$ such that

$$\mathrm{P}_\mathcal{Y}(\mathbf{R}) = \operatorname*{argmax}_{\mathbf{Y}\in\mathcal{Y}} \mathrm{Tr}\Big(\mathbf{R}^\top \mathbf{Y}\Big)$$

with running times $\mathrm{T}_\mathcal{X}$ and $\mathrm{T}_\mathcal{Y}$, respectively, then Algorithm 7.14 outputs $\mathbf{X}^{(r)} \in \mathcal{X}$ and $\mathbf{Y}^{(r)} \in \mathcal{Y}$ such that

$$\mathrm{Tr}\Big(\mathbf{X}^{(r)\top} \mathbf{A}_r \mathbf{Y}^{(r)}\Big) \geq \mathrm{Tr}\Big(\mathbf{X}_\star^{(r)\top} \mathbf{A}_r \mathbf{Y}_\star^{(r)}\Big) - 2\epsilon\sqrt{k} \cdot \|\mathbf{A}_r\|_2 \cdot \mu_\mathcal{X} \cdot \mu_\mathcal{Y},$$

in time $O\Big((2\sqrt{r}/\epsilon)^{r\cdot k} \cdot (\mathrm{T}_\mathcal{X} + \mathrm{T}_\mathcal{Y} + (m+n)r)\Big) + \mathrm{T}_{\mathsf{SVD}}(r)$. Here, $\mu_\mathcal{X} \triangleq \max_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\|_\mathrm{F}$ and $\mu_\mathcal{Y} \triangleq \max_{\mathbf{Y}\in\mathcal{Y}} \|\mathbf{Y}\|_\mathrm{F}$.

Proof. In the sequel, $\widetilde{\mathbf{U}}$, $\widetilde{\mathbf{\Sigma}}$ and $\widetilde{\mathbf{V}}$ are used to denote the r-truncated singular value decomposition of $\mathbf{A}_r$.

Without loss of generality, we assume that $\mu_{\mathcal{X}} = \mu_{\mathcal{Y}} = 1$ since the variables in $\mathcal{X}$ and $\mathcal{Y}$ can be normalized by $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$, respectively, while simultaneously scaling the singular values of $\mathbf{A}_r$ by a factor of $\mu_{\mathcal{X}} \cdot \mu_{\mathcal{Y}}$. Then, $\|\mathbf{Y}\|_{\infty,2} \leq 1$, $\forall\, \mathbf{Y} \in \mathcal{Y}$, where $\|\mathbf{Y}\|_{\infty,2}$ denotes the maximum of the ℓ_2-norm of the columns of $\mathbf{Y}$.

Let $\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}$ be the optimal pair on $\mathbf{A}_r$, *i.e.*,

$$(\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}) \triangleq \underset{\mathbf{X}\in\mathcal{X}, \mathbf{Y}\in\mathcal{Y}}{\operatorname{argmax}} \operatorname{TR}\Big(\mathbf{X}^\top \mathbf{A}_r \mathbf{Y}\Big)$$

and define the $r \times k$ matrix $\widetilde{\mathbf{C}}_\star \triangleq \widetilde{\mathbf{V}}^\top \mathbf{Y}_\star^{(r)}$. Note that

$$\|\widetilde{\mathbf{C}}_\star\|_{\infty,2} = \|\widetilde{\mathbf{V}}^\top \mathbf{Y}_\star^{(r)}\|_{\infty,2} = \max_{1\leq i\leq k} \|\widetilde{\mathbf{V}}^\top [\mathbf{Y}_\star^{(r)}]_{:,i}\|_2 \leq 1, \tag{F.1}$$

with the last inequality following from the facts that $\|\mathbf{Y}\|_{\infty,2} \leq 1\ \forall\ \mathbf{Y} \in \mathcal{Y}$ and the columns of $\widetilde{\mathbf{V}}$ are orthonormal. Alg. 7.14 iterates over the points in $(\mathbb{B}_2^{r-1})^{\otimes k}$. The latter is used to describe the set of $r \times k$ matrices whose columns have ℓ_2 norm at most equal to 1. At each point, the algorithm computes a candidate solution. By (F.1), the ϵ-net contains an $r \times k$ matrix $\mathbf{C}_\sharp$ such that

$$\|\mathbf{C}_\sharp - \widetilde{\mathbf{C}}_\star\|_{\infty,2} \leq \epsilon.$$

Let $\mathbf{X}_\sharp, \mathbf{Y}_\sharp$ be the candidate pair computed at $\mathbf{C}_\sharp$ by the two step maximization, *i.e.*,

$$\mathbf{X}_\sharp \triangleq \underset{\mathbf{X}\in\mathcal{X}}{\operatorname{argmax}} \operatorname{TR}\Big(\mathbf{X}^\top \widetilde{\mathbf{U}} \widetilde{\mathbf{\Sigma}} \mathbf{C}_\sharp\Big)$$

and

$$\mathbf{Y}_\sharp \triangleq \operatorname*{argmax}_{\mathbf{Y}\in\mathcal{Y}} \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \mathbf{A}_r \mathbf{Y}\Big). \tag{F.2}$$

We show that the objective values achieved by the candidate pair $\mathbf{X}_\sharp, \mathbf{Y}_\sharp$ satisfies the inequality of the lemma implying the desired result.

By the definition of $\widetilde{\mathbf{C}}_\star$ and the linearity of the trace,

$$\begin{aligned}
\textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top} \mathbf{A}_r \mathbf{Y}_\star^{(r)}\Big) &= \textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top} \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{C}}_\star\Big) \\
&= \textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top} \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\mathbf{C}_\sharp\Big) + \textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top} \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\widetilde{\mathbf{C}}_\star - \mathbf{C}_\sharp\Big)\Big) \\
&\leq \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\mathbf{C}_\sharp\Big) + \textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top} \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\widetilde{\mathbf{C}}_\star - \mathbf{C}_\sharp\Big)\Big).
\end{aligned} \tag{F.3}$$

The inequality follows from the fact that (by definition (F.2)) $\mathbf{X}_\sharp$ maximizes the first term over all $\mathbf{X} \in \mathcal{X}$. We compute an upper bound on the right hand side of (F.3). Define

$$\widehat{\mathbf{Y}} \triangleq \operatorname*{argmin}_{\mathbf{Y}\in\mathcal{Y}} \|\widetilde{V}^\top \mathbf{Y} - \mathbf{C}_\sharp\|_{\infty,2}.$$

(We note that $\widehat{\mathbf{Y}}$ is used for the analysis and is never explicitly calculated.) Further, define the $r \times k$ matrix $\widehat{\mathbf{C}} \triangleq \widetilde{\mathbf{V}}^\top \widehat{\mathbf{Y}}$. By the linearity of the trace operator

$$\begin{aligned}
\textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\mathbf{C}_\sharp\Big) &= \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widehat{\mathbf{C}}\Big) + \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\mathbf{C}_\sharp - \widehat{\mathbf{C}}\Big)\Big) \\
&= \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^\top \widehat{\mathbf{Y}}\Big) + \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\mathbf{C}_\sharp - \widehat{\mathbf{C}}\Big)\Big) \\
&\leq \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^\top \mathbf{Y}_\sharp\Big) + \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\mathbf{C}_\sharp - \widehat{\mathbf{C}}\Big)\Big) \\
&= \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \mathbf{A}_r \mathbf{Y}_\sharp\Big) + \textsc{Tr}\Big(\mathbf{X}_\sharp^\top \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\mathbf{C}_\sharp - \widehat{\mathbf{C}}\Big)\Big).
\end{aligned} \tag{F.4}$$

The inequality follows from the fact that (by definition (F.2)) $\mathbf{Y}_\sharp$ maximizes the first term over all $\mathbf{Y} \in \mathcal{Y}$. Combining (F.4) and (F.3), and rearranging the terms, we obtain

$$\begin{aligned}
&\mathrm{TR}\Big(\mathbf{X}_\star^{(r)\top}\mathbf{A}_r\mathbf{Y}_\star^{(r)}\Big) - \mathrm{TR}\Big(\mathbf{X}_\sharp^\top\mathbf{A}_r\mathbf{Y}_\sharp\Big) \\
&\quad\le \mathrm{TR}\Big(\mathbf{X}_\star^{(r)\top}\widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\widetilde{\mathbf{C}}_\star - \mathbf{C}_\sharp\Big)\Big) + \mathrm{TR}\Big(\mathbf{X}_\sharp^\top\widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\mathbf{C}_\sharp - \widehat{\mathbf{C}}\Big)\Big). \qquad \text{(F.5)}
\end{aligned}$$

By Lemma F.2.62,

$$\begin{aligned}
\Big|\mathrm{TR}\Big(\mathbf{X}_\star^{(r)\top}\widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\widetilde{\mathbf{C}}_\star - \mathbf{C}_\sharp\Big)\Big)\Big| &\le \|\mathbf{X}_\star^{(r)\top}\widetilde{\mathbf{U}}\|_{\mathrm{F}} \cdot \|\widetilde{\mathbf{\Sigma}}\|_2 \cdot \|\widetilde{\mathbf{C}}_\star - \mathbf{C}_\sharp\|_{\mathrm{F}} \\
&\le \|\mathbf{X}_\star^{(r)}\|_{\mathrm{F}} \cdot \sigma_1(\mathbf{A}_r) \cdot \sqrt{k} \cdot \epsilon \\
&\le \max_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\|_{\mathrm{F}} \cdot \sigma_1(\mathbf{A}_r) \cdot \sqrt{k} \cdot \epsilon \\
&\le \sigma_1(\mathbf{A}_r) \cdot \sqrt{k} \cdot \epsilon. \qquad \text{(F.6)}
\end{aligned}$$

Similarly,

$$\begin{aligned}
\Big|\mathrm{TR}\Big(\mathbf{X}_\sharp^\top\widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\Big(\mathbf{C}_\sharp - \widehat{\mathbf{C}}\Big)\Big)\Big| &\le \|\mathbf{X}_\sharp\widetilde{\mathbf{U}}\|_{\mathrm{F}} \cdot \|\widetilde{\mathbf{\Sigma}}\|_2 \cdot \|\mathbf{C}_\sharp - \widehat{\mathbf{C}}\|_{\mathrm{F}} \\
&\le \max_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\|_{\mathrm{F}} \cdot \sigma_1(\mathbf{A}_r) \cdot \sqrt{k} \cdot \epsilon \\
&\le \sigma_1(\mathbf{A}_r) \cdot \sqrt{k} \cdot \epsilon. \qquad \text{(F.7)}
\end{aligned}$$

The second inequality follows from the fact that by the definition of $\widehat{\mathbf{C}}$,

$$\begin{aligned}
\|\widehat{\mathbf{C}} - \mathbf{C}_\sharp\|_{\infty,2} = \|\widetilde{\mathbf{V}}^\top\widehat{\mathbf{Y}} - \mathbf{C}_\sharp\|_{\infty,2} &\le \|\widetilde{\mathbf{V}}^\top\mathbf{Y}_\star^{(r)} - \mathbf{C}_\sharp\|_{\infty,2} \\
= \|\widetilde{\mathbf{C}}_\star - \mathbf{C}_\sharp\|_{\infty,2} &\le \epsilon,
\end{aligned}$$

which implies that

$$\|\widehat{\mathbf{C}} - \mathbf{C}_\sharp\|_{\mathrm{F}} \le \sqrt{k} \cdot \epsilon.$$

Continuing from (F.5) under (F.6) and (F.7),

$$\operatorname{Tr}\Big(\mathbf{X}_\sharp^\top \mathbf{A}_r \mathbf{Y}_\sharp\Big) \geq \operatorname{Tr}\Big(\mathbf{X}_\star^{(r)\top} \mathbf{A}_r \mathbf{Y}_\star^{(r)}\Big) - 2 \cdot \epsilon \cdot \sqrt{k} \cdot \sigma_1(\mathbf{A}_r).$$

Recalling that the singular values of $\mathbf{A}_r$ have been scaled by a factor of $\mu_{\mathcal{X}} \cdot \mu_{\mathcal{Y}}$ yields the desired result.

The runtime of Alg. 7.14 follows from the cost per iteration and the cardinality of the ϵ-net. Matrix multiplications can exploit the truncated singular value decomposition of $\mathbf{A}_r$ which is performed only once. □

Lemma F.1.58. *For any $\mathbf{A}, \mathbf{A}_r \in \mathbb{R}^{m \times n}$, and norm-boudned sets $\mathcal{X} \subseteq \mathbb{R}^{m \times k}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n \times k}$, let*

$$(\mathbf{X}_\star, \mathbf{Y}_\star) \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}, \mathbf{Y} \in \mathcal{Y}} \operatorname{Tr}\Big(\mathbf{X}^\top \mathbf{A} \mathbf{Y}\Big),$$

and

$$(\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}) \triangleq \operatorname*{argmax}_{\mathbf{X} \in \mathcal{X}, \mathbf{Y} \in \mathcal{Y}} \operatorname{Tr}\Big(\mathbf{X}^\top \mathbf{A}_r \mathbf{Y}\Big).$$

For any $(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}) \in \mathcal{X} \times \mathcal{Y}$ such that

$$\operatorname{Tr}\Big(\mathbf{X}^{(r)\top} \mathbf{A}_r \mathbf{Y}^{(r)}\Big) \geq \gamma \cdot \operatorname{Tr}\Big(\mathbf{X}_\star^{(r)\top} \mathbf{A}_r \mathbf{Y}_\star^{(r)}\Big) - C$$

for some $0 < \gamma \leq 1$, we have

$$\begin{aligned} \operatorname{Tr}\Big(\mathbf{X}^{(r)\top} \mathbf{A} \mathbf{Y}^{(r)}\Big) \geq \gamma \cdot \operatorname{Tr}\Big(\mathbf{X}_\star^\top \mathbf{A} \mathbf{Y}_\star\Big) - C \\ - 2 \cdot \|\mathbf{A} - \mathbf{A}_r\|_2 \cdot \mu_{\mathcal{X}} \cdot \mu_{\mathcal{Y}}. \end{aligned}$$

where $\mu_{\mathcal{X}} \triangleq \max_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\|_{\mathrm{F}}$ and $\mu_{\mathcal{Y}} \triangleq \max_{\mathbf{Y} \in \mathcal{Y}} \|\mathbf{Y}\|_{\mathrm{F}}$.

Proof. By the optimality of $\mathbf{X}_\star^{(r)}, \mathbf{Y}_\star^{(r)}$ for $\mathbf{A}_r$, we have

$$\textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top}\mathbf{A}_r\mathbf{Y}_\star^{(r)}\Big) \geq \textsc{Tr}\Big(\mathbf{X}_\star^\top\mathbf{A}_r\mathbf{Y}_\star\Big).$$

In turn, for any $(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}) \in \mathcal{X} \times \mathcal{Y}$ such that

$$\textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}_r\mathbf{Y}^{(r)}\Big) \geq \gamma \cdot \textsc{Tr}\Big(\mathbf{X}_\star^{(r)\top}\mathbf{A}_r\mathbf{Y}_\star^{(r)}\Big) - C$$

for some $0 < \gamma < 1$ (if such pairs exist), we have

$$\textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}_r\mathbf{Y}^{(r)}\Big) \geq \gamma \cdot \textsc{Tr}\Big(\mathbf{X}_\star^\top\mathbf{A}_r\mathbf{Y}_\star\Big) - C. \tag{F.8}$$

By the linearity of the trace operator,

$$\begin{aligned}\textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}_r\mathbf{Y}^{(r)}\Big) &= \textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}\mathbf{Y}^{(r)}\Big) - \textsc{Tr}\Big(\mathbf{X}^{(r)\top}(\mathbf{A}-\mathbf{A}_r)\mathbf{Y}^{(r)}\Big) \\ &\leq \textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}\mathbf{Y}^{(r)}\Big) + \Big|\textsc{Tr}\Big(\mathbf{X}^{(r)\top}(\mathbf{A}-\mathbf{A}_r)\mathbf{Y}^{(r)}\Big)\Big|. \end{aligned} \tag{F.9}$$

By Lemma F.2.62,

$$\begin{aligned}\Big|\textsc{Tr}\Big(\mathbf{X}^{(r)\top}(\mathbf{A}-\mathbf{A}_r)\mathbf{Y}^{(r)}\Big)\Big| &\leq \|\mathbf{X}^{(r)}\|_{\mathrm{F}} \cdot \|\mathbf{Y}^{(r)}\|_{\mathrm{F}} \cdot \|\mathbf{A}-\mathbf{A}_r\|_2 \\ &\leq \|\mathbf{A}-\mathbf{A}_r\|_2 \cdot \max_{\mathbf{X}\in\mathcal{X}}\|\mathbf{X}\|_{\mathrm{F}} \cdot \max_{\mathbf{Y}\in\mathcal{Y}}\|\mathbf{Y}\|_{\mathrm{F}}. \end{aligned} \tag{F.10}$$

Let R denote the RHS of (F.10). Continuing from (F.9),

$$\textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}_r\mathbf{Y}^{(r)}\Big) \leq \textsc{Tr}\Big(\mathbf{X}^{(r)\top}\mathbf{A}\mathbf{Y}^{(r)}\Big) + R. \tag{F.11}$$

Similarly,

$$\begin{aligned}\textsc{Tr}\Big(\mathbf{X}_\star^\top\mathbf{A}_r\mathbf{Y}_\star\Big) &= \textsc{Tr}\Big(\mathbf{X}_\star^\top\mathbf{A}\mathbf{Y}_\star\Big) - \textsc{Tr}\Big(\mathbf{X}_\star^\top(\mathbf{A}-\mathbf{A}_r)\mathbf{Y}_\star\Big) \\ &\geq \textsc{Tr}\Big(\mathbf{X}_\star^\top\mathbf{A}\mathbf{Y}_\star\Big) - \Big|\textsc{Tr}\Big(\mathbf{X}_\star^\top(\mathbf{A}-\mathbf{A}_r)\mathbf{Y}_\star\Big)\Big| \\ &\geq \textsc{Tr}\Big(\mathbf{X}_\star^\top\mathbf{A}\mathbf{Y}_\star\Big) - R. \end{aligned} \tag{F.12}$$

Combining the above, we have

$$\begin{aligned}
\operatorname{Tr}\left(\mathbf{X}^{(r)\top}\mathbf{A}\mathbf{Y}^{(r)}\right) &\geq \operatorname{Tr}\left(\mathbf{X}^{(r)\top}\mathbf{A}_r\mathbf{Y}^{(r)}\right) - R \\
&\geq \gamma \cdot \operatorname{Tr}\left(\mathbf{X}_\star^\top \mathbf{A}_r \mathbf{Y}_\star\right) - R - C \\
&\geq \gamma \cdot \left(\operatorname{Tr}\left(\mathbf{X}_\star^\top \mathbf{A}\mathbf{Y}_\star\right) - R\right) - R - C \\
&= \gamma \cdot \operatorname{Tr}\left(\mathbf{X}_\star^\top \mathbf{A}\mathbf{Y}_\star\right) - (1+\gamma)\cdot R - C \\
&\geq \gamma \cdot \operatorname{Tr}\left(\mathbf{X}_\star^\top \mathbf{A}\mathbf{Y}_\star\right) - 2\cdot R - C,
\end{aligned}$$

where the first inequality follows from (F.11) the second from (F.8), the third from (F.12), and the last from the fact that $R \geq 0$. This concludes the proof. □

Lemma 7.4.12. *For any* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, let*

$$(\mathbf{X}_\star, \mathbf{Y}_\star) \triangleq \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X},\mathbf{Y}\in\mathcal{Y}} \operatorname{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{Y}\right),$$

where $\mathcal{X} \subseteq \mathbb{R}^{m\times k}$ *and* $\mathcal{Y} \subseteq \mathbb{R}^{n\times k}$ *are sets satisfying the conditions of Lemma 7.4.11. Let* $\mathbf{A}_r$ *be a rank-r approximation of* $\mathbf{A}$*, and* $\mathbf{X}^{(r)} \in \mathcal{X}$*,* $\mathbf{Y}^{(r)} \in \mathcal{Y}$ *be the output of Alg. 7.14 with input* $\mathbf{A}_r$ *and accuracy* ϵ*. Then,*

$$\operatorname{Tr}\left(\mathbf{X}_\star^\top \mathbf{A}\mathbf{Y}_\star\right) - \operatorname{Tr}\left(\mathbf{X}^{(r)\top}\mathbf{A}\mathbf{Y}^{(r)}\right) \leq 2\cdot\left(\epsilon\sqrt{k}\cdot\|\mathbf{A}_r\|_2 + \|\mathbf{A}-\mathbf{A}_r\|_2\right)\cdot\mu_\mathcal{X}\cdot\mu_\mathcal{Y},$$

where $\mu_\mathcal{X} \triangleq \max_{\mathbf{X}\in\mathcal{X}}\|\mathbf{X}\|_\mathrm{F}$ *and* $\mu_\mathcal{Y} \triangleq \max_{\mathbf{Y}\in\mathcal{Y}}\|\mathbf{Y}\|_\mathrm{F}$*.*

Proof. The proof follows the approximation guarantees of Alg. 7.14 in Lemma 7.4.11 and Lemma F.1.58. □

In the sequel, we use $\|\mathbf{X}\|_{\infty,1}$ to denote the maximum of the ℓ_1 norm of the rows of $\mathbf{X}$. When $\mathbf{X} \in \{0,1\}^{d\times k}$, the constraint $\|\mathbf{X}\|_{\infty,1} = 1$ effectively implies that each row of $\mathbf{X}$ has exactly one nonzero entry.

Lemma 7.5.13. *Let* $\mathcal{X} \triangleq \left\{\mathbf{X} \in \{0,1\}^{d\times k} : \|\mathbf{X}\|_{\infty,1} = 1\right\}$. *For any* $d \times k$ *real matrix* $\mathbf{L}$, *Algorithm 7.15 outputs*

$$\mathbf{X}^{(r)} = \operatorname*{argmax}_{\mathbf{X}\in\mathcal{X}} \textsc{Tr}\left(\mathbf{X}^\top \mathbf{L}\right),$$

in time $O(k \cdot d)$

Proof. By construction, each row of $\mathbf{X}$ has exactly one nonzero entry. Let $j_i \in [k]$ denote the index of the nonzero entry in the ith row of $\mathbf{X}$. For any $\mathbf{X} \in \mathcal{X}$,

$$\begin{aligned}\textsc{Tr}\left(\mathbf{X}^\top \mathbf{L}\right) &= \sum_{j=1}^{k} \mathbf{x}_j^\top \mathbf{l}_j = \sum_{j=1}^{k} \sum_{i\in\operatorname{supp}(\mathbf{x}_j)} 1 \cdot L_{ij} \\ &= \sum_{i=1}^{d} L_{ij_i} \le \sum_{i=1}^{d} \max_{j\in[k]} L_{ij}. \end{aligned} \tag{F.13}$$

Algorithm 7.15 achieves equality in (F.13) due to the choice of j_i in line 3. Finally, the running time follows immediately from the $O(k)$ time required to determine the maximum entry of each of the d rows of $\mathbf{L}$. □

F.2 Auxiliary Lemmas

Lemma F.2.59. *Let* $a_1, \ldots, a_n$ *and* $b_1, \ldots, b_n$ *be* $2n$ *real numbers and let* p *and* q *be two numbers such that* $1/p + 1/q = 1$ *and* $p > 1$. *We have*

$$\left|\sum_{i=1}^{n} a_i b_i\right| \le \left(\sum_{i=1}^{n} |a_i|^p\right)^{1/p} \cdot \left(\sum_{i=1}^{n} |b_i|^q\right)^{1/q}.$$

Lemma F.2.60. *For any* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times k}$,

$$|\langle \mathbf{A}, \mathbf{B}\rangle| \triangleq \left|\mathrm{Tr}\left(\mathbf{A}^\top \mathbf{B}\right)\right| \leq \|\mathbf{A}\|_{\mathrm{F}} \|\mathbf{B}\|_{\mathrm{F}}.$$

Proof. Treating $\mathbf{A}$ and $\mathbf{B}$ as vectors, the lemma follows immediately from Lemma F.2.59 for $p = q = 2$. □

Lemma F.2.61. *For any two real matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *of appropriate dimensions,*

$$\|\mathbf{AB}\|_{\mathrm{F}} \leq \min\{\|\mathbf{A}\|_2 \|\mathbf{B}\|_{\mathrm{F}},\ \|\mathbf{A}\|_{\mathrm{F}} \|\mathbf{B}\|_2\}.$$

Proof. Let $\mathbf{b}_i$ denote the ith column of $\mathbf{B}$. Then,

$$\begin{aligned}\|\mathbf{AB}\|_{\mathrm{F}}^2 &= \sum_i \|\mathbf{A}\mathbf{b}_i\|_2^2 \leq \sum_i \|\mathbf{A}\|_2^2 \|\mathbf{b}_i\|_2^2 \\ &= \|\mathbf{A}\|_2^2 \sum_i \|\mathbf{b}_i\|_2^2 = \|\mathbf{A}\|_2^2 \|\mathbf{B}\|_{\mathrm{F}}^2.\end{aligned}$$

Similarly, using the previous inequality,

$$\|\mathbf{AB}\|_{\mathrm{F}}^2 = \|\mathbf{B}^\top \mathbf{A}^\top\|_{\mathrm{F}}^2 \leq \|\mathbf{B}^\top\|_2^2 \|\mathbf{A}^\top\|_{\mathrm{F}}^2 = \|\mathbf{B}\|_2^2 \|\mathbf{A}\|_{\mathrm{F}}^2.$$

The desired result follows combining the two upper bounds. □

Lemma F.2.62. *For any real* $m \times k$ *matrix* $\mathbf{X}$, $m \times n$ *matrix* $\mathbf{A}$, *and* $n \times k$ *matrix* $\mathbf{Y}$, $\left|\mathrm{Tr}\left(\mathbf{X}^\top \mathbf{AY}\right)\right| \leq \|\mathbf{A}\|_2 \cdot \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{Y}\|_{\mathrm{F}}$.

Proof. We have

$$\left|\mathrm{Tr}\left(\mathbf{X}^\top \mathbf{AY}\right)\right| \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{AY}\|_{\mathrm{F}} \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{A}\|_2 \cdot \|\mathbf{Y}\|_{\mathrm{F}},$$

with the first inequality following from Lemma F.2.60 on $|\langle \mathbf{X}, \mathbf{AY}\rangle|$ and the second from Lemma F.2.61. □

Lemma F.2.63. *For any real $m \times n$ matrix $\mathbf{A}$, and pair of $m \times k$ matrix $\mathbf{X}$ and $n \times k$ matrix $\mathbf{Y}$ such that $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_k$ and $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k$ with $k \leq \min\{m,\ n\}$, the following holds:*

$$\left|\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{Y}\right)\right| \leq \sqrt{k} \cdot \left(\sum_{i=1}^{k} \sigma_i^2(\mathbf{A})\right)^{1/2}.$$

Proof. By Lemma F.2.60,

$$|\langle \mathbf{X},\ \mathbf{A}\mathbf{Y}\rangle| = \left|\textsc{Tr}\left(\mathbf{X}^\top \mathbf{A}\mathbf{Y}\right)\right| \leq \|\mathbf{X}\|_{\mathrm{F}} \cdot \|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}} = \sqrt{k} \cdot \|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}.$$

where the last inequality follows from the fact that $\|\mathbf{X}\|_{\mathrm{F}}^2 = \textsc{Tr}\left(\mathbf{X}^\top \mathbf{X}\right) = \textsc{Tr}(\mathbf{I}_k) = k$. Further, for any $\mathbf{Y}$ such that $\mathbf{Y}^T \mathbf{Y} = \mathbf{I}_k$,

$$\|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}^2 \leq \max_{\substack{\widehat{\mathbf{Y}} \in \mathbb{R}^{n \times k} \\ \widehat{\mathbf{Y}}^\top \widehat{\mathbf{Y}} = \mathbf{I}_k}} \|\mathbf{A}\widehat{\mathbf{Y}}\|_{\mathrm{F}}^2 = \sum_{i=1}^{k} \sigma_i^2(\mathbf{A}). \tag{F.14}$$

Combining the two inequalities, the result follows. □

Lemma F.2.64. *For any real $m \times n$ matrix $\mathbf{A}$, and any $k \leq \min\{m,\ n\}$,*

$$\max_{\substack{\mathbf{Y} \in \mathbb{R}^{n \times k} \\ \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k}} \|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}} = \left(\sum_{i=1}^{k} \sigma_i^2(\mathbf{A})\right)^{1/2}.$$

The above equality is realized when the k columns of $\mathbf{Y}$ coincide with the k leading right singular vectors of $\mathbf{A}$.

Proof. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the singular value decomposition of $\mathbf{A}$, with $\Sigma_{jj} = \sigma_j$ being the jth largest singular value of $\mathbf{A}$, $j = 1, \ldots, d$, where $d \triangleq \min\{m, n\}$. Due to the invariance of the Frobenius norm under unitary multiplication,

$$\|\mathbf{A}\mathbf{Y}\|_{\mathrm{F}}^2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \mathbf{Y}\|_{\mathrm{F}}^2 = \|\mathbf{\Sigma}\mathbf{V}^\top \mathbf{Y}\|_{\mathrm{F}}^2. \tag{F.15}$$

Continuing from (F.15),

$$\begin{aligned}\|\mathbf{\Sigma V}^\top \mathbf{Y}\|_{\mathrm{F}}^2 &= \mathrm{Tr}\Big(\mathbf{Y}^\top \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top \mathbf{Y}\Big) \\ &= \sum_{i=1}^{k} \mathbf{y}_i^\top \left(\sum_{j=1}^{d} \sigma_j^2 \cdot \mathbf{v}_j \mathbf{v}_j^\top\right) \mathbf{y}_i \\ &= \sum_{j=1}^{d} \sigma_j^2 \cdot \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2.\end{aligned}$$

Let $z_j \triangleq \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2$, $j = 1, \ldots, d$. Note that each individual z_j satisfies

$$0 \le z_j \triangleq \sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2 \le \|\mathbf{v}_j\|^2 = 1,$$

where the last inequality follows from the fact that the columns of $\mathbf{Y}$ are orthonormal. Further,

$$\sum_{j=1}^{d} z_j = \sum_{j=1}^{d}\sum_{i=1}^{k} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2 = \sum_{i=1}^{k}\sum_{j=1}^{d} \Big(\mathbf{v}_j^\top \mathbf{y}_i\Big)^2 \qquad = \sum_{i=1}^{k} \|\mathbf{y}_i\|^2 = k.$$

Combining the above, we conclude that

$$\|\mathbf{AY}\|_{\mathrm{F}}^2 = \sum_{j=1}^{d} \sigma_j^2 \cdot z_j \le \sigma_1^2 + \ldots + \sigma_k^2. \tag{F.16}$$

Finally, it is straightforward to verify that if $\mathbf{y}_i = \mathbf{v}_i$, $i = 1, \ldots, k$, then (F.16) holds with equality. □

Lemma F.2.65. *For any real $m \times n$ matrix $\mathbf{A}$, let $\sigma_i(\mathbf{A})$ be the ith largest singular value. For any $r, k \le \min\{m, n\}$,*

$$\sum_{i=r+1}^{r+k} \sigma_i(\mathbf{A}) \le \frac{k}{\sqrt{r+k}} \|\mathbf{A}\|_{\mathrm{F}}.$$

Proof. By the Cauchy-Schwartz inequality,

$$\sum_{i=r+1}^{r+k} \sigma_i(\mathbf{A}) = \sum_{i=r+1}^{r+k} |\sigma_i(\mathbf{A})| \leq \left(\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{A}) \right)^{1/2} \|\mathbf{1}_k\|_2 \quad = \sqrt{k} \cdot \left(\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{A}) \right)^{1/2}.$$

Note that $\sigma_{r+1}(\mathbf{A}), \ldots, \sigma_{r+k}(\mathbf{A})$ are the k smallest among the $r+k$ largest singular values. Hence,

$$\sum_{i=r+1}^{r+k} \sigma_i^2(\mathbf{A}) \leq \frac{k}{r+k} \sum_{i=1}^{r+k} \sigma_i^2(\mathbf{A}) \leq \frac{k}{r+k} \sum_{i=1}^{l} \sigma_i^2(\mathbf{A}) \qquad = \frac{k}{r+k} \|\mathbf{A}\|_{\mathrm{F}}^2.$$

Combining the two inequalities, the desired result follows. □

Corollary 6. *For any real $m \times n$ matrix $\mathbf{A}$, the rth largest singular value $\sigma_{\mathrm{r}}(\mathbf{A})$ satisfies $\sigma_{\mathrm{r}}(\mathbf{A}) \leq \|\mathbf{A}\|_{\mathrm{F}}/\sqrt{r}$.*

Proof. It follows immediately from Lemma F.2.65. □

Bibliography

[1] KookJin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2237–2246, 2015.

[2] Nir Ailon, Noa Avigdor-Elgrabli, Edo Liberty, and Anke Van Zuylen. Improved approximation algorithms for bipartite correlation clustering. *SIAM Journal on Computing*, 41(5):1110–1121, 2012.

[3] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

[4] Nir Ailon and Edo Liberty. Correlation clustering revisited: The "true" cost of error minimization problems. In *Automata, Languages and Programming*, pages 24–36. Springer, 2009.

[5] Genevera I. Allen and Mirjana Maletić-Savatić. Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics*, 2011.

[6] Noga Alon, Troy Lee, Adi Shraibman, and Santosh Vempala. The approximate rank of a matrix and its algorithmic applications: approxi-

mate rank. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 675–684. ACM, 2013.

[7] Arash Amini and Martin Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. *The Annals of Statistics*, pages 2877–2921, 2009.

[8] Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 2454–2458. IEEE, 2008.

[9] Noga Amit. The bicluster graph editing problem. Master's thesis, Tel Aviv University, 2004.

[10] Scott A Armstrong, Jane E Staunton, Lewis B Silverman, Rob Pieters, Monique L den Boer, Mark D Minden, Stephen E Sallan, Eric S Lander, Todd R Golub, and Stanley J Korsmeyer. Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature genetics*, 30(1):41–47, 2001.

[11] Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. *arXiv preprint arXiv:1212.4777*, 2012.

[12] Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the 44th symposium on Theory of Computing*, pages 145–162, 2012.

[13] Megasthenis Asteris, Anastasios Kyrillidis, Alex Dimakis, Han-Gyol Yi, and Bharath Chandrasekaran. Stay On Path: PCA Along Graph Paths. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1728–1736, 2015.

[14] Megasthenis Asteris, Anastasios Kyrillidis, Oluwasanmi Koyejo, and Russell Poldrack. A Simple and Provable Algorithm for Sparse Diagonal CCA. In Kilian Q. Weinberger Maria Florina Balcan, editor, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48. JMLR Workshop and Conference Proceedings, 2016.

[15] Megasthenis Asteris, Anastasios Kyrillidis, Dimitris Papailiopoulos, and Alexandros Dimakis. Bipartite Correlation Clustering: Maximizing Agreements. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 121–129, 2016.

[16] Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis. Nonnegative sparse pca with provable guarantees. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1728–1736. JMLR Workshop and Conference Proceedings, 2014.

[17] Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros G Dimakis. Orthogonal nmf through subspace exploration. In *Advances in Neural Information Processing Systems*, pages 343–351, 2015.

[18] Megasthenis Asteris, Dimitris Papailiopoulos, and Georgios Karystinos. The sparse principal component of a constant-rank matrix. *Information Theory, IEEE Transactions on*, 60(4):2281–2290, April 2014.

[19] Megasthenis Asteris, Dimitris Papailiopoulos, Anastasios Kyrillidis, and Alexandros G Dimakis. Sparse PCA via Bipartite Matchings. In *Advances in Neural Information Processing Systems*, pages 766–774, 2015.

[20] Megasthenis Asteris, Dimitris S Papailiopoulos, and George N Karystinos. Sparse principal component of a rank-deficient matrix. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 673–677. IEEE, 2011.

[21] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

[22] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[23] Liviu Badea and Doina Tilivea. Sparse factorizations of gene expression guided by binding data. In *Pacific Symposium on Biocomputing*, 2005.

[24] Luca Baldassarre, Nirav Bhan, Volkan Cevher, Anastasios Kyrillidis, and Siddhartha Satpathi. Group-sparse model selection: Hardness and relaxations. *arXiv preprint arXiv:1303.3207*, 2013.

[25] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[26] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *Information Theory, IEEE Transactions on*, 56(4):1982–2001, 2010.

[27] Marc G Berman, John Jonides, and Derek Evan Nee. Studying mind and brain with fmri. *Social cognitive and affective neuroscience*, 1(2):158–161, 2006.

[28] Victor Bittorf, Benjamin Recht, Christopher Re, and Joel A Tropp. Factoring nonnegative matrices with linear programs. *Advances in Neural Information Processing Systems*, 25:1223–1231, 2012.

[29] Francesco Bonchi, David Garcia-Soriano, and Edo Liberty. Correlation clustering: from theory to practice. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data mining*, pages 1972–1972. ACM, 2014.

[30] Károly Böröczky Jr and Gergely Wintsche. Covering the sphere by equal spherical balls. In *Discrete and Computational Geometry*, pages 235–251. Springer, 2003.

[31] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Sparse features for pca-like linear regression. In *Advances in Neural Information Processing Systems*, pages 2285–2293, 2011.

[32] Gershon Buchsbaum and Orin Bloch. Color categories revealed by non-negative matrix factorization of munsell color spectra. *Vision research*, 42(5):559–563, 2002.

[33] Liming Cai, Michael Fellows, David Juedes, and Frances Rosamond. On efficient polynomial-time approximation schemes for problems on planar structures. *Journal of Computer and System Sciences*, 2003.

[34] Bin Cao, Dou Shen, Jian-Tao Sun, Xuanhui Wang, Qiang Yang, and Zheng Chen. Detect and track latent factors with online nonnegative matrix factorization. In *IJCAI*, volume 7, pages 2689–2694, 2007.

[35] Marco Cesati and Luca Trevisan. On the efficiency of polynomial time approximation schemes. *Information Processing Letters*, 64(4):165–171, 1997.

[36] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 524–533. IEEE, 2003.

[37] Gang Chen, Fei Wang, and Changshui Zhang. Collaborative filtering

using orthogonal nonnegative matrix tri-factorization. *Information Processing & Management*, 45(3):368–379, 2009.

[38] Jun Chen, Frederic D Bushman, James D Lewis, Gary D Wu, and Hongzhe Li. Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*, 14(2):244–258, 2013.

[39] Xi Chen, Han Liu, and Jaime G Carbonell. Structured sparse canonical correlation analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 199–207, 2012.

[40] Koei Chin, Sandy DeVries, Jane Fridlyand, Paul T Spellman, Ritu Roydasgupta, Wen-Lin Kuo, Anna Lapuk, Richard M Neve, Zuwei Qian, Tom Ryder, et al. Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer cell*, 10(6):529–541, 2006.

[41] Seungjin Choi. Algorithms for orthogonal nonnegative matrix factorization. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1828–1832. IEEE, 2008.

[42] Delin Chu, Li-Zhi Liao, Michael K Ng, and Xiaowei Zhang. Sparse canonical correlation analysis: new formulation and algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):3050–3065, 2013.

[43] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[44] William W Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data mining*, pages 475–480. ACM, 2002.

[45] Thomas Cormen, Clifford Stein, Ronald Rivest, and Charles Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[46] Zdravko Cvetkovski. Generalizations of the cauchy–schwarz inequality, chebishev's inequality and the mean inequalities. In *Inequalities*, pages 107–116. Springer Berlin Heidelberg, 2012.

[47] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[48] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *J. Mach. Learn. Res.*, 9:1269–1294, Jun 2008.

[49] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *The Journal of Machine Learning Research*, 9:1269–1294, 2008.

[50] Alexandre d'Aspremont, Francis R. Bach, and Laurent El Ghaoui. Full regularization path for sparse principal component analysis. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 177–184, New York, NY, USA, 2007. ACM.

[51] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I Jordan, and Gert RG Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.

[52] Filip Deleus and Marc M Van Hulle. Functional connectivity analysis of fmri data based on regularized multiset canonical correlation analysis. *Journal of Neuroscience methods*, 197(1):143–157, 2011.

[53] Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2):172–187, 2006.

[54] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[55] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[56] Lei Du, Jingwen Yan, Sungeun Kim, Shannon L Risacher, Heng Huang, Mark Inlow, Jason H Moore, Andrew J Saykin, and Li Shen. A novel structure-aware sparse learning algorithm for brain imaging genetics. In *Medical Image Computing and Computer-Assisted Intervention*, pages 329–336. Springer, 2014.

[57] Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

[58] Ilya Dumer. Covering spheres with spheres. *Discrete & Computational Geometry*, 38(4):665–679, 2007.

[59] Xiaoli Zhang Fern and Carla E Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the 21st International Conference on Machine Learning*, page 36. ACM, 2004.

[60] NA Gillis and S Vavasis. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[61] Nicolas Gillis. Nonnegative matrix factorization complexity, algorithms and applications. *Ph.D Dissertation, Universite Catholique de Louvain*, 2011.

[62] Nicolas Gillis and Stephen A Vavasis. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *arXiv preprint arXiv:1208.1237*, 2012.

[63] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *Proceedings of the 17th annual ACM-SIAM Symposium on Discrete algorithms*, pages 1167–1176. Society for Industrial and Applied Mathematics, 2006.

[64] Michael D Greicius, Kaustubh Supekar, Vinod Menon, and Robert F Dougherty. Resting-state functional connectivity reflects structural connectivity in the default mode network. *Cerebral cortex*, 19(1):72–78, 2009.

[65] University of Minnesota GroupLens Lab. Movielens datasets. http://grouplens.org/datasets/movielens/, 2015. Accessed: 2015-10-03.

[66] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[67] David R Hardoon and John Shawe-Taylor. Technical report, university college london (ucl). 2007.

[68] David R Hardoon and John Shawe-Taylor. Sparse canonical correlation analysis. *Machine Learning*, 83(3):331–353, 2011.

[69] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. A nearly-linear time framework for graph-structured sparsity. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 928–937, 2015.

[70] Harold Hotelling. Relations between two sets of variates. *Biometrika*, pages 321–377, 1936.

[71] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[72] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

[73] K Huang, ND Sidiropoulos, and A Swamiy. Nmf revisited: New uniqueness results and algorithms. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 4524–4528. IEEE, 2013.

[74] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 366–373, 2010.

[75] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 366–373, 2010.

[76] Ruoyi Jiang, Hongliang Fei, and Jun Huan. Anomaly localization for network data streams with graph joint sparse pca. In *Proceedings of the 17th ACM SIGKDD*, pages 886–894. ACM, 2011.

[77] Iain Johnstone and Arthur Yu Lu. Sparse principal components analysis. *Unpublished manuscript*, 2004.

[78] Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Ann. Statist*, 29(2):295–327, 2001.

[79] 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), 2009.

[80] I.T. Jolliffe. Rotation of principal components: choice of normalization constraints. *Journal of Applied Statistics*, 22(1):29–35, 1995.

[81] I.T. Jolliffe, N.T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[82] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11:517–553, 2010.

[83] H.F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

[84] Marek Karpinski and Warren Schudy. Linear time approximation schemes for the gale-berlekamp game and related minimization problems. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pages 313–322. ACM, 2009.

[85] G.N. Karystinos and A.P. Liavas. Efficient computation of the binary vector that maximizes a rank-deficient quadratic form. *Information Theory, IEEE Transactions on*, 56(7):3581–3593, 2010.

[86] Hans Kellerer, Renata Mansini, Ulrich Pferschy, and Maria Grazia Speranza. An efficient fully polynomial approximation scheme for the subset-sum problem. *Journal of Computer and System Sciences*, 66(2):349 – 370, 2003.

[87] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[88] Takumi Kobayashi. S3cca: Smoothly structured sparse cca for partial pattern matching. In *Pattern Recognition (ICPR), 22nd International Conference on*, pages 1981–1986. IEEE, 2014.

[89] Da Kuang, Haesun Park, and Chris HQ Ding. Symmetric nonnegative

matrix factorization for graph clustering. In *SDM*, volume 12, pages 106–117. SIAM, 2012.

[90] Abhishek Kumar, Vikas Sindhwani, and Prabhanjan Kambadur. Fast conical hull algorithms for near-separable non-negative matrix factorization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 231–239, 2013.

[91] Anastasios Kyrillidis, Luca Baldassarre, Marwa El Halabi, Quoc Tran-Dinh, and Volkan Cevher. Structured sparsity: Discrete and convex approaches. In *Compressed Sensing and its Applications*, pages 341–387. Springer, 2015.

[92] Anastasios Kyrillidis and Volkan Cevher. Combinatorial selection and least absolute shrinkage via the clash algorithm. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2216–2220. IEEE, 2012.

[93] Anastasios Kyrillidis, Gilles Puy, and Volkan Cevher. Hard thresholding with norm constraints. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, number EPFL-CONF-183061, pages 3645–3648. Ieee, 2012.

[94] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[95] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2000.

[96] Hualiang Li, Tülay Adal, Wei Wang, Darren Emge, and Andrzej Cichocki. Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 48(1-2):83–97, 2007.

[97] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 362–371. IEEE, 2006.

[98] Xin Li, William KW Cheung, Jiming Liu, and Zhili Wu. A novel orthogonal nmf-based belief compression for pomdps. In *Proceedings of the 24th international conference on Machine learning*, pages 537–544. ACM, 2007.

[99] Yi-Ou Li, Tom Eichele, Vince D Calhoun, and Tulay Adali. Group study of simulated driving fmri data by multiset canonical correlation analysis. *Journal of signal processing systems*, 68(1):31–48, 2012.

[100] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[101] Dongdong Lin, Vince D Calhoun, and Yu-Ping Wang. Correspondence between fmri and snp data by group sparse canonical correlation analysis. *Medical image analysis*, 18(6):891–902, 2014.

[102] Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

[103] Lester Mackey. Deflation methods for sparse pca. In *Advances in Neural Information Processing Systems 21*, NIPS '08, pages 1–8, Vancouver, Canada, Dec 2008.

[104] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[105] Malik Magdon-Ismail. Np-hardness and inapproximability of sparse PCA. *CoRR*, abs/1502.05675, 2015.

[106] Malik Magdon-Ismail and Christos Boutsidis. Optimal sparse linear auto-encoders and sparse pca. *arXiv preprint arXiv:1502.06626*, 2015.

[107] Julien Mairal and Bin Yu. Path coding penalties for directed acyclic graphs. In *Proceedings of the 4th NIPS Workshop on Optimization for Machine Learning (OPT'11)*. Citeseer, 2011.

[108] A Majumdar. Image compression by sparse pca coding in curvelet domain. *Signal, image and video processing*, 3(1):27–34, 2009.

[109] Shahar Mendelson. Empirical processes with a bounded ψ 1 diameter. *Geometric and Functional Analysis*, 20(4):988–1027, 2010.

[110] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. *Advances in neural information processing systems*, 18:915, 2006.

[111] Michael Morley, Cliona M Molony, Teresa M Weber, James L Devlin, Kathryn G Ewens, Richard S Spielman, and Vivian G Cheung. Genetic analysis of genome-wide variation in human gene expression. *Nature*, 430(7001):743–747, 2004.

[112] Katta G. Murty and Santosh N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987.

[113] D. S. Papailiopoulos, A. G. Dimakis, and S. Korokythakis. Sparse pca through low-rank approximations. In *Proceedings of the 30th International Conference on Machine Learning*, ICML '13, pages 767–774. ACM, 2013.

[114] Dimitris S Papailiopoulos, Alexandros G Dimakis, and Stavros Korokythakis. Sparse pca through low-rank approximations. *Proceedings of the 28th International Conference on Machine learning*, pages 747–755, 2013.

[115] Elena Parkhomenko, David Tritchler, and Joseph Beyene. Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–34, 2009.

[116] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[117] Russell A Poldrack. Can cognitive processes be inferred from neuroimaging data? *Trends in cognitive sciences*, 10(2):59–63, 2006.

[118] Jonathan R Pollack, Therese Sørlie, Charles M Perou, Christian A Rees, Stefanie S Jeffrey, Per E Lonning, Robert Tibshirani, David Botstein, Anne-Lise Børresen-Dale, and Patrick O Brown. Microarray analysis reveals a major direct role of dna copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences*, 99(20):12963–12968, 2002.

[119] Filippo Pompili, Nicolas Gillis, P-A Absil, and François Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *arXiv preprint arXiv:1201.0901*, 2012.

[120] Filippo Pompili, Nicolas Gillis, Pierre-Antoine Absil, and François Glineur. Onp-mf: An orthogonal nonnegative matrix factorization algorithm with application to clustering. In *ESANN 2013*, 2013.

[121] Michael I Posner, Steven E Petersen, Peter T Fox, and Marcus E Raichle. Localization of cognitive operations in the human brain. *Science*, 240(4859):1627–1631, 1988.

[122] Lyle Ramshaw and Robert E Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1*, 2012.

[123] Emile Richard, Guillaume R Obozinski, and Jean-Philippe Vert. Tight convex relaxations for sparse matrix factorization. In *Advances in Neural Information Processing Systems*, pages 3284–3292, 2014.

[124] CA Rogers. A note on coverings. *Mathematika*, 4(01):1–6, 1957.

[125] Indrayana Rustandi, Marcel Adam Just, and Tom Mitchell. Integrating multiple-study multiple-subject fmri datasets using canonical correlation analysis. In *Proceedings of the MICCAI 2009 Workshop: Statistical modeling and detection issues in intra-and inter-subject functional MRI data analysis*, 2009.

[126] Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[127] Christian D. Sigg and Joachim M. Buhmann. Expectation-maximization for sparse and non-negative pca. In *Proceedings of the 25th International*

Conference on Machine Learning, ICML '08, pages 960–967, New York, NY, USA, 2008. ACM.

[128] Stephen M Smith, Thomas E Nichols, Diego Vidaurre, Anderson M Winkler, Timothy EJ Behrens, Matthew F Glasser, Kamil Ugurbil, Deanna M Barch, David C Van Essen, and Karla L Miller. A positive-negative mode of population covariation links brain connectivity, demographics and behavior. *Nature neuroscience*, 18(11):1565–1567, 2015.

[129] Barbara E Stranger, Matthew S Forrest, Mark Dunning, Catherine E Ingle, Claude Beazley, Natalie Thorne, Richard Redon, Christine P Bird, Anna de Grassi, Charles Lee, et al. Relative impact of nucleotide and copy number variation on gene expression phenotypes. *Science*, 315(5813):848–853, 2007.

[130] Kah-Kay Sung. *Learning and example selection for object and pattern recognition*. PhD thesis, PhD thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996.

[131] Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the 15th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 526–527. Society for Industrial and Applied Mathematics, 2004.

[132] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos Papadopoulos, and Yannis Manolopoulos. Nearest-biclusters collaborative filtering

with constant values. In *Advances in web mining and web usage analysis*, pages 36–55. Springer, 2007.

[133] Bruce Thompson. *Canonical correlation analysis: Uses and interpretation*. Number 47. Sage, 1984.

[134] Robert M Thorndike. *Correlational procedures for research*. Wiley, 1976.

[135] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, pages 104–117, 2003.

[136] David A Torres, Douglas Turnbull, Luke Barrington, and Gert RG Lanckriet. Identifying words that are musically meaningful. In *ISMIR*, volume 7, pages 405–410, 2007.

[137] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, WU-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[138] Anke Van Zuylen and David P Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34(3):594–620, 2009.

[139] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.

[140] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[141] Michail Vlachos, Francesco Fusco, Charalambos Mavroforakis, Anastasios Kyrillidis, and Vassilios G Vassiliadis. Improving co-cluster quality with application to product recommendations. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 679–688. ACM, 2014.

[142] Vincent Vu and Jing Lei. Minimax rates of estimation for sparse pca in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pages 1278–1286, 2012.

[143] Vincent Q Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. In *NIPS*, pages 2670–2678, 2013.

[144] Sandra Waaijenborg, Philip C Verselewel de Witt Hamer, and Aeilko H Zwinderman. Quantifying the association between gene expressions and dna-markers by penalized canonical correlation analysis. *Statistical Applications in Genetics and Molecular Biology*, 7(1), 2008.

[145] Zhaoran Wang, Fang Han, and Han Liu. Sparse principal component analysis for high dimensional multivariate time series. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 48–56, 2013.

[146] Zhaoran Wang, Huanran Lu, and Han Liu. Nonconvex statistical optimization: minimax-optimal sparse pca in polynomial time. *arXiv preprint arXiv:1408.5352*, 2014.

[147] Ami Wiesel, Mark Kliger, and Alfred O Hero III. A greedy approach to sparse canonical correlation analysis. *arXiv preprint arXiv:0801.2748*, 2008.

[148] Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, page kxp008, 2009.

[149] Daniela M Witten and Robert J Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1):1–27, 2009.

[150] Bianca C Wittmann, Björn H Schott, Sebastian Guderian, Julietta U Frey, Hans-Jochen Heinze, and Emrah Düzel. Reward-related fmri activation of dopaminergic midbrain is associated with enhanced hippocampus-dependent long-term memory formation. *Neuron*, 45(3):459–467, 2005.

[151] Aaron D Wyner. Random packings and coverings of the unit n-sphere. *Bell System Technical Journal*, 46(9):2111–2118, 1967.

[152] Zhirong Yang, Tele Hao, Onur Dikmen, Xi Chen, and Erkki Oja. Clustering by nonnegative matrix factorization using graph random walk. In

Advances in Neural Information Processing Systems, pages 1088–1096, 2012.

[153] Zhirong Yang and Erkki Oja. Linear and nonlinear projective nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 21(5):734–749, 2010.

[154] Bin Yu. Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

[155] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *The Journal of Machine Learning Research*, 14(1):899–925, 2013.

[156] Zhijian Yuan and Erkki Oja. Projective nonnegative matrix factorization for image compression and feature extraction. In *Image Analysis*, pages 333–342. Springer, 2005.

[157] Ron Zass and Amnon Shashua. Nonnegative sparse pca. In *Advances in Neural Information Processing Systems 19*, pages 1561–1568, Cambridge, MA, 2007. MIT Press.

[158] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, pages 25–32. ACM, 2001.

[159] Y. Zhang, A. d'Aspremont, and L.E. Ghaoui. Sparse pca: Convex relaxations, algorithms and applications. *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 915–940, 2012.

[160] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[161] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Vita

Megasthenis Asteris is a graduate student at the Department of Electrical and Computer Engineering of the University of Texas at Austin, pursuing his Ph.D. under the supervision of Prof. Alexandros G. Dimakis. Before that he received the M.Sc. in Electrical Engineering from the University of Southern California in 2012 and the Diploma in Electronic and Computer Engineering from the Technical University of Crete, Greece, in 2010.

Permanent address: megas@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.