ABSTRACT

Title of dissertation:	SPECTRAL FRAME ANALYSIS AND LEARNING THROUGH GRAPH STRUCTURE
	Chae Almon Clark, Doctor of Philosophy, 2016
Dissertation directed by:	Professor Kasso A. Okoudjou Professor Wojciech K. Czaja Department of Mathematics

This dissertation investigates the connection between spectral analysis and frame theory. When considering the spectral properties of a frame, we present a few novel results relating to the spectral decomposition. We first show that scalable frames have the property that the inner product of the scaling coefficients and the eigenvectors must equal the inverse eigenvalues. From this, we prove a similar result when an approximate scaling is obtained.

We then focus on the optimization problems inherent to the scalable frames by first showing that there is an equivalence between scaling a frame and optimization problems with a non-restrictive objective function. Various objective functions are considered, and an analysis of the solution type is presented. For linear objectives, we can encourage sparse scalings, and with barrier objective functions, we force dense solutions. We further consider frames in high dimensions, and derive various solution techniques.

From here, we restrict ourselves to various frame classes, to add more specificity to the results. Using frames generated from distributions allows for the placement of probabilistic bounds on scalability. For discrete distributions (Bernoulli and Rademacher), we bound the probability of encountering an ONB, and for continuous symmetric distributions (Uniform and Gaussian), we show that symmetry is retained in the transformed domain. We also prove several hyperplane-separation results.

With the theory developed, we discuss graph applications of the scalability framework. We make a connection with graph conditioning, and show the infeasibility of the problem in the general case. After a modification, we show that any complete graph can be conditioned.

We then present a modification of standard PCA (robust PCA) developed by Candès, and give some background into Electron Energy-Loss Spectroscopy (EELS). We design a novel scheme for the processing of EELS through robust PCA and least-squares regression, and test this scheme on biological samples.

Finally, we take the idea of robust PCA and apply the technique of kernel PCA to perform robust manifold learning. We derive the problem and present an algorithm for its solution. There is also discussion of the differences with RPCA that make theoretical guarantees difficult.

SPECTRAL FRAME ANALYSIS AND LEARNING THROUGH GRAPH STRUCTURE

by

Chae Almon Clark

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Kasso A. Okoudjou, Chair/Advisor
Professor Wojciech K. Czaja, Co-Chair/Advisor
Professor John J. Benedetto
Professor Radu V. Balan
Professor Larry S. Davis Dean's Representative

© Copyright by
Chae Almon Clark
2016

Dedication

Dedicated to my mother, Melanie Clark, for her infinite patience and unending support.

Acknowledgments

There are far too many people to thank that have accompanied and supported me in my journey to this thesis. No one finishes alone, and no one fails together, so I am in debt to all of my friends and family.

I would like to give a special thanks to my advisors, Prof. Kasso Okoudjou and Prof. Wojciech Czaja. They are the reason I applied to grad school in the first place, and they are the main reason for my successful completion of grad requirements. Thank you!

I could fill a full thesis with all of the friends that have helped me succeed in grad school, but there are a special few that I must mention. Thank you Dr. Matthew (Jokemaster) Beguè, Dr. David (RED ROOM) Darmon, Dr. Stefan (YOLO Swag) Doboszcak, Dr. Matthew (Not before 12) Guay, Dr. Benjamin (No return call) Manning, Dr. James (The volume knob is broken) Murphy, and Dr. Daniel (Shearlets are horrible) Weinberg.

The Norbert Wiener Center has been my home away from home and I would like to thank the Professors, John Benedetto, Radu Balan, Wojciech Czaja, Kasso Okoudjou, and all of my collaborators at the center (Franck, Weilin, Christiana, Mark, Ariel).

Lastly, I would like to give a BIG thank you to Asia Wyatt for her support, love, and understanding. Party Time Barton! Work Time Lakiem! Where is your doubt now Kiara?! Also, Auntie Kim is the best!

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Foreword

The use of mathematics in the physical and social sciences relies heavily on the analysis of sampled/collected data. These datasets may be the result of a data-driven process or direct/indirect measurements, but in modern frameworks, two things are universal; the datasets are extremely large in the number of features, and the datasets contain noise.

The former issue of data dimensionality has developed through the refinement of sensors, allowing for the collection of massive amounts of data, raising issues not only about its efficient analysis, but also of its storage. Attempting to distinguish any correlations in these datasets can become a rush for larger/faster computational resources, but this is not a solution. Datasets are being produced of such increasing sizes, that the brute force approach of more computational resources can't keep up. Other oddities occur when applying mathematical models to data lying in high dimensions. Distinguishing differences in datapoints becomes more difficult, as visualized through the volume of ℓ_2-balls in high dimensions [63], and datasets become arbitrarily separable when considering datasets in high dimensions when compared with the number of examples [87].

The latter issue of noise handling has always been a major component of data analysis, as noise in the sample is generally magnified in the analysis, producing unreliable results. This can take place anywhere in the data analysis pipeline, from data acquisition, transmission, processing, to storage. Many mathematical approaches attempts to mitigate/separate noise by using prior information about the noise present [39], or using the lack of correlation between noise and sampled data [19].

To handle both of these issues, it is known that dealing with a reduced version of the dataset, results in stable results and more accurate analysis. “Reduced” in this context, is a representation of the data with only the important features. Under such a framework, we expect these features to be statistically relevant (to mitigate the risk of noise corruption), and we expect them to be uncorrelated with each other (to mitigate the risk of biasing any results). This naturally leads to the application of the *spectral decomposition* of a dataset, and the work of this thesis.

1.2 Outline

In Chapter 2 we give useful preliminary results for the subsequent chapters. The notions presented are necessary for a more complete background of the subject, but most results used in the respective chapters are presented where needed. This is done mainly to keep the chapters as self-contained as possible, but still give a general background. These complementary sections serve to give the reader historical context, background, and notation. We start with an introduction of the *spectral* *decomposition*, along with the notion of a *well-conditioned* system. *Finite Frame* notions are then introduced, giving a connection to spectral methods. *Scalable Frames* are a major focus in this thesis, and we give a concise introduction of the work, along with the necessary motivation. Finally, *dimension reduction* methods are discussed, to prepare the reader for the notions presented in the later chapters.

1.3 Summary of Results

In Chapter 3, we present three major results relating to the spectral decomposition of finite frames. We first show that scalable frames have the property that the inner product of the scaling coefficients and the eigenvectors of scalable frames must be equal to the reciprocal of the corresponding eigenvalues. From this, we prove a similar result when an approximate scaling is obtained. We conclude Chapter 3 by showing the equivalence of scalability and the 0-eigenvalue problem of the frame transform, and present some figures that explore the space of scalable frames.

Chapter 4 focuses on the optimization problems inherent to Scalable Frames. We first show that there is equivalence between scaling a frame and optimization problems with non-restrictive objective functions. Various objective functions are considered, and an analysis of the solution type is presented. For linear objectives, we can encourage sparse scalings, and with barrier objectives, we can force dense solutions. We further consider frames in high dimensions, and derive various solution techniques.

In Chapter 5, restricting ourselves to various frame classes, we add more specificity to the results obtained. Using frames generated from distributions allows us to place probabilistic bounds on scalability. For discrete distributions, we bound the probability of encountering an ONB, and for continuous symmetric distributions, we show that symmetry is retained in the transformed domain, and we prove several hyperplane-separation results. We end the chapter with some numerical results comparing the scalability of certain frame types.

Chapter 6 discusses graph applications of the scalability framework. We make a connection with graph conditioning, and show the in-feasibility of the problem in general. After we modify the definition of scalability, we shown that all complete graphs can be perfectly conditioned. We conclude with a formulation of the graph scaling problem, and present numerical results on example graphs.

In Chapter 7 we present the modification of standard PCA (robust PCA) presented by Candès [16]. We give some background into *Electron Energy-Loss Spectroscopy* (EELS), and then design a novel scheme for the processing of EELS through robust PCA and least-squares regression. The scheme is tested on artificial and experimental biological samples, and the results are presented.

Chapter 8 takes the idea of RPCA and applies the techniques of kernel PCA to perform manifold learning (for nonlinear manifolds). We then derive the problem and present an algorithm for its solution, discussing the differences with robust PCA simultaneously. We end the chapter with a few select applications showing the expressiveness of the technique on a number of datasets.

Chapter 2: Preliminaries

2.1 Notation

We denote by the superscript A^T of a matrix A, the *transpose*, and we denote the *set compliment* with superscript A^c. Let $\|f\|_2$ denote the standard ℓ_2 norm for a vector f, and denote by $\|A\|_2$, the *operator norm* of a matrix A,

$$\|A\|_2 = \sup_{v \neq \mathbf{0}} \left\{ \frac{\|Av\|_2}{\|v\|_2} \right\}.$$

The *Frobeneous norm* of a matrix, $\|A\|_F$, is defined to be the square-root of the sum of matrix elements of A squared, $\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2}$. We denote the *Hadamard product*, $\odot$, of two matrices, A and B, as the element-wise product, $A \odot B$. We take the *inner product* of two vectors, $\langle u, v \rangle$, to be $u^T v$. The identity matrix is denoted I, and $\mathbb{1}$ denotes a vector of ones. $\mathbf{0}$ denotes a matrix or vector of zeros with necessary size. A set of vectors $A = \{a_k\}_{k=1}^m \subset \mathbb{R}^n$ shall also denote a matrix $A = [a_1, a_2, \dots, a_m]$ when needed (we shall distinguish which we are using). When discussing probabilistic distributions, we shall write i.i.d. to denote that values are drawn from the same distribution, independently, and identically. e_k denotes the kth column of the identity matrix (standard basis vectors). The $\operatorname{sgn}(A)$ of a matrix is a function that acts element-wise on A, and returns a matrix containing the sign of each element. The *cardinality* of a set is the number of elements in the set, and is denoted $|\cdot|$. The *pseudo-inverse* of a matrix, $A^\dagger$, is denoted with a superscript †.

2.2 Spectral Analysis

Much of the background in this section is present in [42, 70].

Theorem 2.2.1. (Spectral Decomposition SVD) Let M be an $n \times m$ matrix with elements on $\mathbb{R}$. Then M can be decomposed into,

$$M = U\Sigma V^T,$$

where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ are both real unitary matrices referred to as the left and right singular matrices respectively. $\Sigma \in \mathbb{R}^{n\times m}$ is a diagonal matrix whose entries are non-negative.

When referring to the vectors of U and V, let $u_i \in \mathbb{R}^n$ denote the ith column of U, and let v_k denote the kth column of V. The diagonal entries of Σ are referred to as singular values, and are denoted σ_i. For symmetric matrices we have an equivalent theorem for spectral decomposition.

Theorem 2.2.2. (Spectral Decomposition EVD) Let $G = M^TM$ be an $m \times m$ symmetric matrix with elements on $\mathbb{R}$. Then G can be decomposed into,

$$G = V\Lambda V^T,$$

where $V \in \mathbb{R}^{m\times m}$ is a real unitary matrix referred to as the eigenvector matrix of G. $\Lambda \in \mathbb{R}^{m\times m}$ is a diagonal matrix whose entries real and is referred to as the eigenvalue matrix.

When referring to the eigenvectors of G, we use v_k to denote the kth column of V, and thus, the kth eigenvector of G. This eigen-decomposition is readily seen by performing the singular value decomposition on the matrix G,

$$\begin{aligned} G &= M^T M \\ &= (U\Sigma V^T)^T(U\Sigma V^T) \\ &= V\Sigma^T U^T U\Sigma V^T \\ &= V\Sigma^T\Sigma V^T. \end{aligned}$$

Let $\Lambda = \Sigma^T\Sigma$,

$$G = V\Lambda V^T.$$

By *spectral decomposition* of a matrix, we mean the eigenvalue decomposition (EVD) when the matrix is symmetric, and the singular value decomposition (SVD) otherwise. By *spectrum*, we mean the set of eigenvalues or singular values in a similar fashion. So when discussing spectral properties, we are referring to the properties of the eigenvalues or singular values respectively.

While we shall generally be agnostic to the spectral decomposition used in numerical experiments, it is worth mentioning a classical technique in determining eigenvalues and eigenvectors. Finding the zeros of the characteristic polynomial $p(\lambda) = \det(G - I\lambda)$ is in general not possible [76], so many techniques attempt to approximate zeros of $p(\lambda)$ (and similarly the eigenvectors). In [67] the von Mises iteration is introduced to numerically solve for the largest eigenvalue and its eigenvector. This algorithm became the classical *power iteration*, used to determine extremal eigen-pairs of large matrices [42, 45, 59, 70, 71]. The scheme progresses by successively applying matrix multiplications to an initial vector. For convenience, let G be symmetric matrix with largest eigenvalue $\lambda_1 = 1$. Squaring the matrix results in,

Algorithm 1 Power Iteration

```
Given symmetric $G$ and approximate eigenvector $v^{(0)}$
for $i = 1, 2, \ldots,$ do
    $v^{(i)} \leftarrow \frac{Gv^{(i-1)}}{\|Gv^{(i-1)}\|_2}$
    $\lambda^{(i)} \leftarrow \langle v^{(i)}, Gv^{(i)} \rangle$
end for
$v^{(i)}$ is an approximate eigenvector and $\lambda^{(i)}$ is an approximate eigenvalue
```

$$G^2 = V\Lambda V^T V\Lambda V^T = V\Lambda\Lambda V^T = V\Lambda^2 V^T.$$

The square of G retains the same eigenvectors, but the eigenvalues are squared, resulting in the top eigenvalue remaining constant, and the smaller eigenvalues decreasing. As we continue to apply this operation, the smaller eigenvalues will approach 0, while the largest eigenvalue $\lambda_1^t = 1$,

$$\lim_{t\to\infty} \lambda_i^t = 0, \quad i = 2, \ldots, n.$$

For $t \gg 1$, this constricts the range of G^t to be the span of the largest eigenvector, as the remaining eigenvectors have eigenvalues that are effectively 0. Applying the matrix G^t to a non-trivial vector results in a vector proportional to the eigenvector corresponding to the largest eigenvalue. After making the resulting vector unit norm, we have the standard form of the eigenvector.

This process finds the largest eigen-pair of the matrix. To find the second eigen-pair, we must first remove the current largest eigen-pair. This process can be made explicit by decomposing G into its rank-1 decomposition,

$$G = V\Lambda V^T = \sum_{i=1}^{n} \lambda_i v_i v_i^T.$$

As we have determined λ_1 and v_1 through the initial application of Algorithm (2.2), we can remove this eigen-pair by subtracting the rank-1 matrix $\lambda_1 v_1 v_1^T$ from G,

$$\widetilde{G} = G - \lambda_1 v_1 v_1^T = \sum_{i=2}^{n} \lambda_i v_i v_i^T.$$

The largest eigenvalue of $\widetilde{G}$ is λ_2, and Algorithm (2.2) will result in the second eigen-pair.

We now present notation for the condition of a matrix. Review the works of [42, 43, 70, 83] for a more comprehensive study. Consider the linear system,

$$Gf = b,$$

with $n \times n$ invertible symmetric matrix G, vector b, and unknown vector f. For various reasons [42, 70], we may be unable to determine the solution, f, exactly, and instead we find an approximate solution $\widetilde{f}$ through various methods [26, 89, 98]. This approximate solution solves the approximate linear system,

$$\widetilde{G}\widetilde{f} = b,$$

where $\widetilde{G}$ is the original G with an error matrix E ($\widetilde{G} = G + E$). Performing a relative error analysis of the two solutions we have,

$$\begin{aligned}
\widetilde{G}\widetilde{f} - Gf &= 0 \\
(G+E)\widetilde{f} - Gf &= 0 \\
G(\widetilde{f} - f) &= -E\widetilde{f} \\
\widetilde{f} - f &= -G^{-1}E\widetilde{f} \\
\|\widetilde{f} - f\|_2 &= \|G^{-1}E\widetilde{f}\|_2 \\
\|\widetilde{f} - f\|_2 &\leq \|G^{-1}\|_2\|E\|_2\|\widetilde{f}\|_2 \\
\frac{\|\widetilde{f} - f\|_2}{\|\widetilde{f}\|_2} &\leq \frac{\|G^{-1}\|_2\|E\|_2\|G\|_2}{\|G\|_2} \\
\frac{\|\widetilde{f} - f\|_2}{\|\widetilde{f}\|_2} &\leq (\|G^{-1}\|_2\|G\|_2)\frac{\|E\|_2}{\|G\|_2}.
\end{aligned}$$

The relative error in the approximate solution is bounded by the error matrix E, but also properties of the matrix G. The matrix norms of G and G^{-1} are the largest and reciprocal smallest eigenvalues respectively,

$$\|G\|_2 = \lambda_1 \quad , \quad \|G^{-1}\|_2 = \frac{1}{\lambda_n}.$$

Definition 2.2.3 (Condition Number)**.** We define the condition number of an $n \times n$ matrix G to be,

$$\kappa(G) = \frac{\lambda_1}{\lambda_n}.$$

If the smallest eigenvalue value is 0, we take the condition number to be ∞. We shall extend this definition of condition number to apply to non-square matrices using the singular values. The condition number of a $n \times m$ matrix, M, is defined to be the square-root ratio of the largest and min(n, m)th eigenvalues of $M^T M$.

Definition 2.2.4 (Ill-Conditioned System)**.** We call a matrix M ill-conditioned if the condition number is much larger that 1,

$$\kappa(M) \gg 1.$$

In finding better approximate solutions, two lines of parallel research focus on more accurate approximation methods (reducing $\|E\|_2$ [42]), or using a better conditioned matrix (reducing $\kappa(M)$ [25]). This thesis focuses on the latter notion of preconditioning a matrix to accentuate favourable spectrum properties. This will take the form of optimal condition number $\kappa(M) = 1$ when discussing frames, and the quick decay of the spectrum when discussing dimension reduction.

2.3 Frame Theory

Introduced by Duffin & Schaeffer in [33], frame theory has become a staple of applied harmonic analysis. Where the Fourier transform takes signals and reveals their frequency information, frames allow for the analysis of other inherent properties of the signal. We give a brief introduction of finite frame theory here, and refer to [5, 6, 7, 8, 22, 27, 30, 38].

Definition 2.3.1 (Frame Definition)**.** Given $2 \leq n \leq m$, a finite frame for $\mathbb{R}^n$ is a set $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ such that there exist positive constants $0 < A \leq B < \infty$ for which

$$A\|f\|_2^2 \leq \sum_{k=1}^m |\langle f, \varphi_k\rangle|^2 \leq B\|f\|_2^2,$$

for all signals $f \in \mathbb{R}^n$. A and B are referred to as the upper and lower frame bounds.

The frame bounds as stated are not unique (e.g. an upper frame bound of $B = 2$ implies an upper frame bound of $B = 3$ is true as well), to account for this, we define A_{opt} and B_{opt} to be the supremum of lower bounds and the infimum of upper bounds respectively. We refer to A_{opt} and B_{opt} as the *optimal lower bound* and *optimal upper bound* respectively.

This definition extends the standard notion of a basis representation, but there are two important differences. First, when representing a signal with a frame expansion, the coefficients produced in the analysis are not unique. In fact, there will generally be an infinite number of expansions for a signal. This can be seen by considering a frame consisting of two orthonormal bases $\Phi = \{\varphi_k\}_{k=1}^n \cup \{\phi_k\}_{k=1}^n$. Any signal f can be represented completely with either basis, but could also be represented with the full frame by taking both individual expansions and scaling by a factor $\frac{1}{\sqrt{2}}$. Moreover, the frame elements are not linearly independent. This can be observed by noticing that the number of frame elements, m, is generally greater than the frame dimension, n, and for non-trivial vectors this necessarily implies that the frame elements are linearly dependent.

We analyze a frame through matrices that encode all relevant information. In particular, a frame denoted $\Phi = \{\varphi_k\}_{k=1}^m$ shall also have its matrix $\Phi = [\varphi_1, \ldots, \varphi_m]$ denoted the same way. This matrix Φ is the *synthesis operator*, as it is used to reconstruct the signal with the frame expansion coefficients. Its transpose-conjugate, Φ^T, is the *analysis operator*, as applying it produces the sequence of coefficients to be analyzed. The product of these two matrices is denoted by $S = \Phi\Phi^T$, and is referred to as the *frame operator*. S is full rank and positive definite, and thus admits a well-defined inverse S^{-1}. This comes from the fact that the frame elements span $\mathbb{R}^n$, and thus Φ has full row rank, implying $\Phi\Phi^T$ has full row rank (and equivalently, full column rank).

A frame is called a *tight frame* if the optimal frame bounds are equal, $A_{\text{opt}} = B_{\text{opt}}$. If $A_{\text{opt}} = B_{\text{opt}} = 1$, the frame is also called *Parseval*, because the Parseval identity holds, in that

$$\|f\|_2^2 = \sum_{k=1}^{m} |\langle f, \varphi_k\rangle|^2 = \|f\|_2^2.$$

With bases, F, the reconstruction of the original signal, f, can be performed by multiplying by the basis inverse, F^{-1}, but with frames this operation must be performed with the synthesis operator of a frame. Moreover, as the inverse of a basis is itself only in the case of an orthonormal basis, and a frame's analysis and synthesis operator are transpose-conjugates of each other, we must define the notion of a *dual frame*.

Definition 2.3.2 (Canonical Dual Frame)**.** To every frame $\Phi = \{\varphi_k\}_{k=1}^m$, and associated frame operator $S = \Phi\Phi^T$, denote by Ψ the canonical dual frame,

$$\Psi = \{\psi_k\}_{k=1}^m = \{S^{-1}\varphi_k\}_{k=1}^m.$$

Further denote by Ψ the matrix of dual frame elements.

We define this frame/frame-dual pair this system to use as a surrogate for basis inversion, as

$$\Psi\Phi^T = S^{-1}\Phi\Phi^T = S^{-1}S = I.$$

This leads to the reconstruction formula,

$$f = \sum_{k=1}^{m} c_k \psi_k = \sum_{k=1}^{m} \langle f, \varphi_k \rangle \psi_k = \Psi \Phi^T f.$$

Furthermore, we have that a Parseval frame is self-dual, $S = \Phi\Phi^T = \Psi\Phi^T = I$.

We shall now connect the notion of frame expansions, with that of spectral analysis and conditioning.

Theorem 2.3.3. [22, Theorem 1.5] Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame for $\mathbb{R}^n$, with the frame operator, $S = \Phi\Phi^T$, having eigenvalues,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n-1} \geq \lambda_n.$$

Then λ_1 corresponds with the optimal upper frame bound $B_{\text{opt}} = \lambda_1$, and λ_n corresponds with the optimal lower frame bound $B_{\text{opt}} = \lambda_1$.

This theorem states that for a unit-norm signal f, the ℓ_2 norm of the frame coefficients $\Phi^T f$ will be bounded above by the largest eigenvalue of the frame operator, and will be bounded below by the lowest eigenvalue of the frame operator. This sheds light on the usefulness of a well-conditioned spectrum.

2.4 Scalable Frames

Given a frame $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$, we have a number of methods available to produce tight (or Parseval) frames. We can apply the square root of the inverse frame operator $S^{-\frac{1}{2}}$ to produce a Parseval frame [22],

$$\begin{aligned}\Psi &= S^{-\frac{1}{2}}\Phi \qquad (2.1)\\ \Psi\Psi^T &= S^{-\frac{1}{2}}\Phi\Phi^T S^{-\frac{1}{2}}\\ \Psi\Psi^T &= S^{-\frac{1}{2}}SS^{-\frac{1}{2}}\\ \Psi\Psi^T &= S^{-\frac{1}{2}}S^{\frac{1}{2}}S^{\frac{1}{2}}S^{-\frac{1}{2}}\\ \Psi\Psi^T &= I.\end{aligned}$$

We can perform a singular value decomposition $\Phi = U\Sigma V^T$, and use the truncated right singular vectors as the Parseval frame,

$$\begin{aligned}\Psi &= I_{n\times m}V^T \qquad (2.2)\\ \Psi\Psi^T &= I_{n\times m}V^T V I_{m\times n}\\ \Psi\Psi^T &= I_{n\times m}I_{m\times n}\\ \Psi\Psi^T &= I.\end{aligned}$$

Here $I_{n\times m}$ is the low rank identity matrix of size $n \times m$. We can also perform a Gram-Schmidt orthogonalization on the rows of Φ to produce a Parseval frame,

$$\Psi_i = \Phi_i - \sum_{j=1}^{i-1} \frac{\langle \Phi_j, \Psi_j \rangle}{\|\Psi_j\|_2^2}, \text{ for } i = 1, 2, \ldots, n, \qquad (2.3)$$

where Ψ_i and Φ_i correspond to the rows of the Parseval and original frame respectively. As the rows are orthonormal, $\Psi\Psi^T$ must be the identity, and as such, the frame Ψ is Parseval.

Another technique forgoes the explicit creation of the dual frame or inverting the frame operator, and instead iteratively solves for the unknown signal from the frame coefficients. These methods generally fall under the classification of *frame algorithms* [22, 44]. We present a classical examples below. Note that the algorithm below multiplies the frame operator to a signal, Sf, which can be decomposed into $\Phi(\Phi^T f)$, where $\Phi^T f$ is a vector of the frame coefficients, $(\Phi^T f)_k = \langle f, \varphi_k \rangle$ for $k = 1, \ldots, m$.

Proposition 2.4.1 (Frame Algorithm)**.** [44, Lemma 1] Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame for $\mathbb{R}^n$, with the frame operator $S = \Phi\Phi^T$, and optimal frame bounds $A_{\text{opt}}, B_{\text{opt}}$. For an unknown signal $f \in \mathbb{R}^n$, define a sequence $\{g_i\}_{i=0}^\infty$ by

$$\begin{aligned} g_0 &= \mathbf{0}, \\ g_i &= g_{i-1} + \frac{2}{A_{\text{opt}} + B_{\text{opt}}} S(f - g_{i-1}). \end{aligned}$$

Then, $\{g_i\}_{i=0}^\infty$ converges to f in $\mathbb{R}^n$, and the rate of convergence is

$$\|f - g_i\|_2 \leq \left(\frac{B_{\text{opt}} - A_{\text{opt}}}{B_{\text{opt}} + A_{\text{opt}}} \right)^i \|f\|_2.$$

The methods discussed above for signal synthesis/analysis (determining Parseval frames or approximating dual frames) can be insufficient when the frame elements must retain certain properties(e.g. preserve the angle between the elements). A common problem in signal processing is the reconstruction of a signal from frame measurements [39],

$$\Phi^T f = c.$$

In real-world applications, this frame may have a large number of elements, and computing a canonical dual frame may be too computationally expensive (as this generally requires $O[n^3]$ operations [10, 48, 84]), or it may be the case that the frame is ill-conditioned, making the inversion of the frame operator unreliable [42]. Further, computing eigenvectors can become unstable producing inaccurate results if the methods in (2.1) and (2.2) are applied [42].

Frames transformed into tight frames by only changing the length of the frame elements was introduced as Scalable Frames in [54, 56] partially as an alternative to the above methods.

Definition 2.4.2 (Scalable Frames)**.** Let $2 \leq n \leq m < \infty$ be given. A frame $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ is scalable if there exist a subset $\Phi_J = \{\varphi_k\}_{k \in J}$ with $J \subseteq \{1, 2, \ldots, m\}$, and positive scalars $\{x_k\}_{k \in J}$ such that the system $\Psi_J = \{x_k \varphi_k\}_{k \in J}$ is a Parseval frame for $\mathbb{R}^n$.

By only changing the length of the frame elements, we retain their relative angles, and applying a scalar x_k is a computationally efficient operation (assuming we have an efficient way to find these coefficients). Considering the scaled analysis operator applied to a signal $f \in \mathbb{R}^n$,

$$\langle x_k \varphi_k, f \rangle,$$

we can write this operator as the product of a diagonal matrix $X = X_{kk} = x_k$ and the original analysis operator Φ^T,

$$X^T \Phi^T f.$$

For the scaled synthesis operator, we have a corresponding relation,

$$\Phi X.$$

We shall denote the set of scalable frames in dimension n and m elements as $\mathcal{S}F(n,m)$.

From the definition of scalable frames, we have that the scaled analysis/synthesis operator is a Parseval frame,

$$(\Phi X)(\Phi X)^T = \Phi X X^T \Phi^T = \Phi X^2 \Phi^T = I. \tag{2.4}$$

Also, assume that the scaling coefficients x_k are non-zero, then the angle between frame elements is preserved,

$$\frac{|\langle x_k \phi_k, x_{k'} \phi_{k'} \rangle|}{\|x_k \phi_k\|_2 \|x_{k'} \phi_{k'}\|_2} = \frac{|x_k x_{k'}| |\langle \phi_k, \phi_{k'} \rangle|}{|x_k x_{k'}| \|\phi_k\|_2 \|\phi_{k'}\|_2} = \frac{|\langle \phi_k, \phi_{k'} \rangle|}{\|\phi_k\|_2 \|\phi_{k'}\|_2}.$$

Furthermore, if we are attempting to solve a signal reconstruction system for unknown f and known frame coefficients c,

$$\Phi^T f = c,$$

and our frame Φ is scalable with diagonal scaling matrix X, then we can solve the analogous system,

$$\Phi X \widetilde{f} = c,$$

where $\widetilde{f}$ is a change of variables $f = X\widetilde{f}$ and the matrix ΦX has optimal condition number ($\kappa(\Phi X) = 1$). This relation is also valid when some of the scaling coefficients are zero. In this case, a coefficient of $x_k = 0$ implies that the corresponding frame coefficient, φ_k, is removed from the frame.

In later work in this area [55], the authors form a system of linear equations that guarantee scalability. We derive this system using the frame operator equation

(2.4). We first define two versions of the frame transform T and F. The first variation is more general and will be useful in later chapters in regards to graphs, and the latter variation is reduced in the transformed dimension.

Definition 2.4.3 (Frame Transformation)**.** Define the *frame transformation*, $T : \mathbb{R}^n \to \mathbb{R}^{d+1}$, for $d = \frac{(n-1)(n+2)}{2}$, as follows:

$$T(z) := \begin{bmatrix} T_0(z) \\ T_1(z) \\ \vdots \\ T_n(z) \end{bmatrix} \text{ where } T_0(z) := \begin{bmatrix} z_1^2 \\ z_2^2 \\ \vdots \\ z_n^2 \end{bmatrix} \quad T_i(z) := \begin{bmatrix} z_i z_{i+1} \\ z_i z_{i+2} \\ \vdots \\ z_i z_n \end{bmatrix},$$

where $i = 1, \ldots, n$. When applied to a matrix $Z = [z_1, \ldots, z_m]$, the transformation acts on the columns,

$$T(Z) = [T(z_1), \ldots, T(z_m)].$$

Definition 2.4.4 (Reduced Frame Transformation)**.** Define the *reduced frame transformation*, $F : \mathbb{R}^n \to \mathbb{R}^d$, for $d = \frac{(n-1)(n+2)}{2}$, as follows:

$$F(z) := \begin{bmatrix} F_0(z) \\ F_1(z) \\ \vdots \\ F_n(z) \end{bmatrix} \text{ where } F_0(z) := \begin{bmatrix} z_1^2 - z_2^2 \\ z_1^2 - z_3^2 \\ \vdots \\ z_1^2 - z_n^2 \end{bmatrix} \quad F_i(z) := \begin{bmatrix} z_i z_{i+1} \\ z_i z_{i+2} \\ \vdots \\ z_i z_n \end{bmatrix},$$

where $i = 1, \ldots, n$. When applied to a matrix $Z = [z_1, \ldots, z_m]$, the transformation acts on the columns,

$$F(Z) = [F(z_1), \ldots, F(z_m)].$$

Theorem 2.4.5. A frame $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ is scalable if and only if there exists a $u \in \mathbb{R}_+^m$ that solves the linear system,

$$T(\Phi)u = \begin{bmatrix} \mathbb{1} \\ \\ \mathbf{0} \end{bmatrix}.$$

Proof. From Definition 2.4.2, we have that if a frame is scalable with scaling matrix X, and the corresponding Parseval frame is ΦX, the following equation holds,

$$\Phi X^2 \Phi^T = I,$$

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1m} \\ \varphi_{21} & \ddots & & \varphi_{2m} \\ \vdots & & & \vdots \\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{nm} \end{bmatrix} X^2 \begin{bmatrix} \varphi_{11} & \varphi_{21} & \cdots & \varphi_{n1} \\ \varphi_{12} & \ddots & & \varphi_{n2} \\ \vdots & & & \vdots \\ \varphi_{1m} & \varphi_{2m} & \cdots & \varphi_{nm} \end{bmatrix} = I.$$

We can model the diagonal equations in terms of the square of the scaling weights x_k^2,

$$\sum_{k=1}^m \varphi_{ik}^2 x_k^2 = 1 \quad \text{for } i = 1, \ldots, n.$$

The off diagonal equations can be written similarly,

$$\sum_{k=1}^m \varphi_{ik}\varphi_{jk} x_k^2 = 0 \quad \text{for } i = 1, \ldots, n-1 \;\; i < j < n.$$

Perform the change of variables $u_k = x_k^2$. This adds the constraint $u \geq 0$, and we can now write the system linearly (writing the diagonal equations first) with the frame transformation T,

$$T(\Phi)u = \begin{bmatrix} \mathbb{1} \\ \\ \mathbf{0} \end{bmatrix},$$

with $u \in \mathbb{R}_+^m$. We have only rewritten the equations, and so it follows that this argument holds if and only if. ■

Corollary 2.4.6. A frame $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ is scalable if and only if there exists a $u \in \mathbb{R}_+^m \backslash \mathbf{0}$ that solves the linear system,

$$F(\Phi)u = \mathbf{0}.$$

Proof. For this simplification, it suffices to note that since we require the diagonal elements to all be 1, we can subtract the equations from each other,

$$\sum_{k=2}^{m} (\varphi_{i1}^2 x_k^2 - \varphi_{ik}^2 x_k^2) = 0 \quad \text{for } i = 2, \ldots, n,$$

reducing the number of equations by 1. This new system allows for the trivial solution $u = \mathbf{0}$, so we add a non-zero constraint. We have lost the guarantee that the resulting frame is Parseval (it is only required to be tight), but with an appropriate scaling we can construct the corresponding Parseval frame. Perform the change of variables $u_k = x_k^2$. This adds the constraint $u \geq 0$, and we can now write the system linearly (writing the diagonal equations first) with the reduced frame transformation F,

$$F(\Phi)u = \mathbf{0},$$

with $u \in \mathbb{R}_+^m \backslash \mathbf{0}$. ■

2.5 Low-Rank Embeddings and Pre-Image Problems

Principal Component Analysis (PCA) is a standard tool for data analysis and low-rank approximations [51, 63]. Its ubiquitous use is partly due to the intuition associated with the process. Viewing the eigenvalues as variance indicators is a clear and concise explanation of the projection, and when data's true manifold is linear/affine, PCA is optimal in its representation. For notation, and for connections with later notions, we shall denote PCA as an eigen-value/vector problem of a data matrix M,

$$M^T MV = V\Lambda,$$

where the eigen-decomposition of the gram matrix $M^T M$ is,

$$M^T M = V\Lambda V^T.$$

The assumption of linearity on the manifold is generally violated for complex datasets. The standard examples on which PCA fails are swiss-roll, open box, twin peaks, and spiral datasets [63]. Any linear projection of these surfaces will result in a sub-optimal embedding (sub-optimal here, meaning not preserving structure/distances). Nonlinear dimension reduction techniques were designed to alleviate this drawback. The canonical example being Kernel PCA [63, 74]. Instead of analyzing the data directly, kernel methods analyze the relationship between data points. In [9, 46], various non-linear dimension reduction methods (Isomap [86], Laplacian Eigenmaps [4], Locally Linear Embeddings [72], etc.) are shown to fall under the kernel PCA model, and as such, Kernel PCA is the only nonlinear dimension reduction method we shall consider in this thesis.

The kernel PCA problem for a dataset M, with respect to a kernel $K(M) = K_M$, shall be denoted,

$$K_M V = V\Lambda.$$

It's important to note that we have slightly abused notation here, as the eigenvector matrix V is not necessarily the eigenvector matrix presented above (this is only the case when $K_M = M^T M$). This is done to keep with standard notation in the literature, and wherever the eigenvectors are used, context will differentiate the two cases. The embedding Θ of this dataset, shall be denoted,

$$\Theta = \Lambda^{\frac{1}{2}} V^T,$$

where V is the eigenvector matrix of K_M.

The strength of linear methods like PCA, are that they admit a well-defined inverse, and by inverse, we mean a transformation that takes an embedding obtained through a nonlinear dimension reduction method, and produces an appropriate reconstruction in the original space. For kernel methods, and non-linear dimension reduction methods in general, the inverse problem is ill-posed [66]. The act of using similarity between points to embed a dataset, introduces a number of possible points that give the embedding Θ. The local nature in which many kernel methods produce embeddings further exacerbate the situation by removing any guarantees about the global structure of the original dataset.

In cases where the embedding is used to cluster data points, this is not an issue, as the original dataset is known, and the embedding is only used to aid processing. In cases where the embedding is altered, or when only the embedding is known, a dataset in the original space must be determined. A number of papers explore this topic [2, 47, 57, 66], giving various problem formulations and heuristics. Stated abstractly, given an embedding Θ, the *pre-image* problem is to find a dataset $\widehat{M}$, such that $\widehat{\Theta} \approx \Theta$. This is generally phrased as an optimization problem with various constraints and norms [66],

$$\min_{\widehat{M}} : \left\| K_{\widehat{M}} - K_M \right\| .$$

The authors in [66, 73] show that for *Gaussian kernels*, employing a fixed-point iteration can result in an approximate pre-image. In [57], a method using *multidimensional scaling* [53] is presented, and shown to apply to a wide range of kernels. The works of [57, 79] show various optimization methods that employ gradient descent to converge to an approximate inverse.

Chapter 3: Spectral Analysis of Scalable Frames

3.1 Overview

In this chapter we show the connection between frame scalings and spectral analysis by employing spectral decompositions. This shall lay the framework for the analysis performed in chapter 6, when we discuss connections to graph analysis. We begin with exact scalability results relating to the singular value decomposition, and then prove similar results for frames that can approximately be scaled. This leads to an equation for the eigenvalues of the frame operator as an inner-product between eigenvectors and scaling coefficients. We then discuss an alternate formulation of scalability from an eigen-decomposition point of view, and end the chapter with some visualizations of the space of possible scalings.

Much of the motivation for this section can be drawn from spectral graph theory [28]. By analyzing the spectrum of the frame, we can give necessary and sufficient conditions for the space of scalable frames. In the same way that connectivity of a graph can be characterized by the smallest non-zero eigenvalue of its Laplacian, scalable frames can be completely characterized by the eigenvectors corresponding to the zero eigenvalues of the frame transformation, F, applied to the frame.

3.2 Spectral Analysis

Theorem 3.2.1 (Spectral Frame Decomposition)**.** Let Φ be a frame in $\mathbb{R}^n$ with m elements, and assume Φ is scalable with diagonal scaling matrix X. Furthermore, let V be an $m \times m$ matrix of the right singular vectors of Φ, such that the singular value decomposition is

$$\Phi = U\Sigma V^T.$$

Then there exists an $m \times n$ sub-block of V (denoted $\widetilde{V}$) such that

$$\widetilde{V}^T X^2 \widetilde{V} = \Lambda^{-1}.$$

Proof. Assuming Φ is scalable, we have

$$\Phi X^2 \Phi^T = I.$$

Using a singular value decomposition of Φ, we have

$$\begin{aligned}(U\Sigma V^T)X^2(U\Sigma V^T)^T &= I,\\ U\Sigma V^T X^2 V \Sigma^T U^T &= I.\end{aligned}$$

We can simplify this system by performing left and right matrix multiplications of U^T and U respectively.

$$\begin{aligned}U^T U\Sigma V^T X^2 V\Sigma^T U^T U &= U^T I U,\\ \Sigma V^T X^2 V \Sigma^T &= I.\end{aligned}$$

We shall now perform left and right matrix multiplications by Σ^T and Σ respectively. In this case, we obtain block matrices where the upper-left $n \times n$ block is a diagonal matrix of non-zero eigenvalues, Λ, and all other blocks are zero matrices.

$$\Sigma^T \Sigma V^T X^2 V \Sigma^T \Sigma = \Sigma^T I \Sigma,$$

$$\begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} V^T X^2 V \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

We write V in block form as well,

$$\begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix}^T \begin{bmatrix} X_1 & \mathbf{0} \\ \mathbf{0} & X_2 \end{bmatrix}^2 \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix} \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Given this structure, we shall further simplify the system by removing the zero matrices, obtaining the result,

$$\Lambda \widetilde{V}^T X^2 \widetilde{V} \Lambda = \Lambda,$$

$$\widetilde{V}^T X^2 \widetilde{V} = \Lambda^{-1} \Lambda \Lambda^{-1},$$

$$\widetilde{V}^T X^2 \widetilde{V} = \Lambda^{-1}.$$

■

In the general case, we cannot do any further analysis, as the matrix $\widetilde{V}$ will not have orthogonal rows (due to the truncation). If Φ contains an orthogonal set, then the truncated singular vectors will be orthogonal, but little more is known. The trouble lies in requiring that the matrix $\widetilde{V}$ be row orthogonal after applying a diagonal matrix, but this is in general only true when the diagonal matrix is a scale of the identity. We can, though, say some things relating the eigenvalues to the scaling achieved.

Corollary 3.2.2. Let Φ be a frame in $\mathbb{R}^n$ with m elements, and assume Φ is scalable with diagonal scaling matrix X. Furthermore, let V be an $m \times m$ matrix of the right singular vectors of Φ, such that the singular value decomposition is

$$\Phi = U\Sigma V^T.$$

Then the inverse of each eigenvalue of the frame operator $S = \Phi\Phi^T$ can be written as the sum of squares of the right singular vectors $(v_i)_k = v_{ik}$ and the scaling weights $X_{kk} = x_k$,

$$\frac{1}{\lambda_i} = \langle v_i \odot v_i, x \odot x\rangle = \sum_{k=1}^{m} (v_{ik}x_k)^2 \quad \text{for } i = 1, \ldots, n.$$

Proof. Consider the equation

$$\widetilde{V}^T X^2 \widetilde{V} = \Lambda^{-1}.$$

Take the ith diagonal entry of Λ^{-1}. This location is equivalent to the inner product of the ith row of the matrix $\widetilde{V}$ (after X is applied) with itself. This results in the equation,

$$\frac{1}{\lambda_i} = \sum_{k=1}^{m} (v_{ik}x_k)^2.$$

■

3.3 Perturbed Spectral Analysis

It will often occur that a frame will not be exactly scalable. In this case we can still analyze the frame, although the results will not be as robust. We assume a frame is non-scalable, with approximate scaling matrix Y, and prove worst-case bounds for the change in smallest eigenvalue.

Theorem 3.3.1 (Perturbed Spectral Decomposition)**.** Let Φ be a frame in $\mathbb{R}^n$ with m elements, and let V be an $m \times m$ matrix of the right singular vectors of Φ, such that the singular value decomposition is

$$\Phi = U\Sigma V^T.$$

Also, let $\widetilde{V}$ denote an $m \times n$ sub-block of V. Given a non-trivial, non-negative diagonal matrix Y (and thus positive semi-definite), we shall write the general scalability equality as

$$\Phi Y^2 \Phi^T = I + E,$$

with an error matrix, E, bounded by

$$E \preceq \delta \mathbb{1}\mathbb{1}^T,$$

for some $\delta > 0$. Then the following inequality holds,

$$\left\| \widetilde{V}^T Y^2 \widetilde{V} \right\|_2 \leq \frac{1 + \delta n}{\lambda_n}.$$

Proof. Following the process outlined in the proof of Theorem 3.2.1, we shall simplify the scaled frame,

$$\begin{aligned}
\Phi Y^2 \Phi^T &= I + E, \\
(U\Sigma V^T) Y^2 (U\Sigma V^T)^T &= I + E, \\
U\Sigma V^T Y^2 V \Sigma^T U^T &= I + E, \\
U^T U\Sigma V^T Y^2 V \Sigma^T U^T U &= U^T I U + U^T E U, \\
I\Sigma V^T Y^2 V \Sigma^T I &= U^T U + U^T E U, \\
\Sigma V^T Y^2 V \Sigma^T &= I + U^T E U, \\
\Sigma^T \Sigma V^T Y^2 V \Sigma^T \Sigma &= \Sigma^T I \Sigma + \Sigma^T U^T E U \Sigma, \\
\Lambda \widetilde{V}^T Y^2 \widetilde{V} \Lambda &= \Lambda + \Lambda^{1/2} U^T E U \Lambda^{1/2}, \\
\widetilde{V}^T Y^2 \widetilde{V} &= \Lambda^{-1} \Lambda \Lambda^{-1} + \Lambda^{-1/2} U^T E U \Lambda^{-1/2}, \\
\widetilde{V}^T Y^2 \widetilde{V} &= \Lambda^{-1} + \Lambda^{-1/2} U^T E U \Lambda^{-1/2}.
\end{aligned}$$

Taking the norm of both sides of the equation, and applying the bound, we have on the error matrix E,

$$\begin{aligned}
\|\widetilde{V}^T Y^2 \widetilde{V}\|_2 &= \|\Lambda^{-1} + \Lambda^{-1/2} U^T E U \Lambda^{-1/2}\|_2, \\
\|\widetilde{V}^T Y^2 \widetilde{V}\|_2 &\leq \|\Lambda^{-1}\|_2 + \|\Lambda^{-1/2} U^T E U \Lambda^{-1/2}\|_2, \\
\|\widetilde{V}^T Y^2 \widetilde{V}\|_2 &\leq \|\Lambda^{-1}\|_2 + \|\Lambda^{-1/2} U^T (\delta \mathbb{1}\mathbb{1}^T) U \Lambda^{-1/2}\|_2, \\
\|\widetilde{V}^T Y^2 \widetilde{V}\|_2 &\leq \frac{1}{\lambda_n} + \frac{\delta}{\lambda_n} \|U^T (\mathbb{1}\mathbb{1}^T) U\|_2, \\
\|\widetilde{V}^T Y^2 \widetilde{V}\|_2 &\leq \frac{1}{\lambda_n} + \frac{\delta}{\lambda_n} \|\mathbb{1}\mathbb{1}^T\|_2, \\
\|\widetilde{V}^T Y^2 \widetilde{V}\|_2 &\leq \frac{1}{\lambda_n} + \frac{\delta n}{\lambda_n}.
\end{aligned}$$

■

This can also be a useful analysis method when we have a scalable frame, but only have an approximate solution. In the following result, we bound the error in approximate scaling matrices.

Corollary 3.3.2. Let Φ be a frame in $\mathbb{R}^n$ with m elements, and let V be an $m \times m$ matrix of the right singular vectors of Φ, such that the singular value decomposition is

$$\Phi = U \Sigma V^T.$$

Also, let $\widetilde{V}$ denote an $m \times n$ sub-block of V. Given a non-trivial, non-negative diagonal matrix Y, we shall write the general scalability equality as

$$\Phi Y^2 \Phi^T = I + E,$$

with an error matrix, E, bounded by

$$E \preceq \delta \mathbb{1}\mathbb{1}^T,$$

for some $\delta > 0$. If Φ is scalable with scaling matrix X, and the difference between X and Y is denoted $D^2 := X^2 - Y^2$, then the following inequality holds,

$$\left\| \widetilde{V}^T D^2 \widetilde{V} \right\|_2 \leq \frac{\delta n}{\lambda_n}.$$

Proof. Decompose the above inequality into a difference of diagonal matrices,

$$\begin{aligned}
\Phi Y^2 \Phi^T &= I + E, \\
\Phi(X^2 - D^2)\Phi^T &= I + E, \\
\Phi X^2 \Phi^T - \Phi D^2 \Phi^T &= I + E, \\
I - \Phi D^2 \Phi^T &= I + E, \\
\Phi D^2 \Phi^T &= E. \\
U\Sigma V^T D^2 V \Sigma^T U^T &= E
\end{aligned}$$

$$\begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix}^T \begin{bmatrix} D_1 & \mathbf{0} \\ \mathbf{0} & D_2 \end{bmatrix}^2 \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix} \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

$$\widetilde{V}^T D^2 \widetilde{V} = \Lambda^{-1/2} U^T E U \Lambda^{-1/2}.$$

Applying a norm to both sides of the equation, and following the process presented in the proof of Theorem 3.3.1, gives the result,

$$\|\widetilde{V}^T D^2 \widetilde{V}\|_2 \leq \frac{\delta n}{\lambda_n}.$$

■

3.4 Spectral Interpretation of Scalability

We now present a spectral scalability theorem.

Theorem 3.4.1 (Spectral Scalability)**.** A frame Φ is scalable if and only if there exists a $u \in \mathbb{R}^m_+$ that solves the linear system,

$$T(\widetilde{V})u = \begin{bmatrix} \lambda_1^{-1} \\ \vdots \\ \lambda_n^{-1} \\ \mathbf{0} \end{bmatrix},$$

where λ_i are the eigenvalues of the frame operator $S = \Phi\Phi^T$.

Proof. From Theorem 3.2.1, we have that if a frame is scalable, the following equation holds,

$$\Phi X^2 \Phi^T = I,$$

$$\widetilde{V}^T X^2 \widetilde{V} = \Lambda^{-1}.$$

From Corollary 3.2.2, we can model the diagonal equations as,

$$\sum_{k=1}^{m} v_{ik}^2 x_k^2 = \frac{1}{\lambda_i} \quad \text{for } i = 1, \ldots, n.$$

The off diagonal equations can be written similarly,

$$\sum_{k=1}^{m} v_{ik} v_{jk} x_k^2 = 0 \quad \text{for } i = 1, \ldots, n-1 \quad i < j < n.$$

Define $u_k := x_k^2$. This adds the constraint $u \geq 0$, and we can now write the system linearly (writing the diagonal equations first) with the transformation T,

$$T(\widetilde{V})u = \begin{bmatrix} \lambda_1^{-1} \\ \vdots \\ \lambda_n^{-1} \\ \mathbf{0} \end{bmatrix},$$

with $u \in \mathbb{R}^m_+$. As all we have done is rewrite the equation, it is clear that this argument holds if and only if. ■

We can also phrase scalability as some combination of right singular vectors (or similarly, the eigenvectors).

Theorem 3.4.2. A frame Φ is scalable if and only if there exists a $u \in \mathbb{R}^m_+$ that is a linear combination of the right singular vectors of $F(\Phi)$ that correspond to zero singular values.

Proof. For clarity and without loss of generality, we shall show that this problem can be written as zero eigenvalue problem. In this result, the eigenvectors are the right singular vectors. Consider the following linear system and multiply by the transpose of the reduced frame transform $F(\Phi)^T$,

$$F(\Phi)u = \mathbf{0},$$

$$F(\Phi)^T F(\Phi)u = \mathbf{0}.$$

Excluding the trivial solution, any zero eigenvector is a solution to this system, and as there are at least $m - d$ of them, any linear combination of these eigenvectors is a solution to the system. Scalability is achieved if the linear combination results in a non-negative solution u. ■

3.5 Scalability Projections

In the last section of this chapter, we aim to give a graphical representation of the differences in scalable and non-scalable frames. We now present scalability maps where each pixel maps to the residual ($\|F(\Phi)u\|_1$) from a candidate scaling normalized to have unit sum. For frames with $m > 2$ vectors, the computation results in a $N_1 \times N_2 \times \cdots \times N_m$ dimensional array. To display the result in 2D,

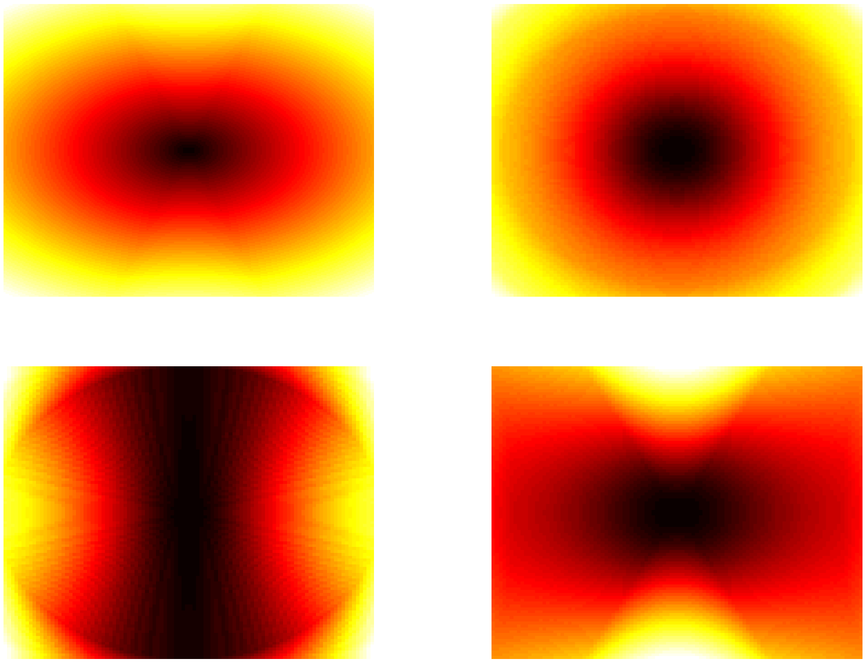

Figure 3.1: The scalability maps of 4 non-scalable frames. The scalability maps are projections onto the xy-plane that attain a value of 0 (Black) when the frame is scalable. In these examples, the center of the map is the trivial solution $u = \mathbf{0}$, and due to these frames being non-scalable, we see that only the radius around the origin attains a small residual.

we take the minimum over the other dimensions, such that a pixel value is the best possible over the scalings.

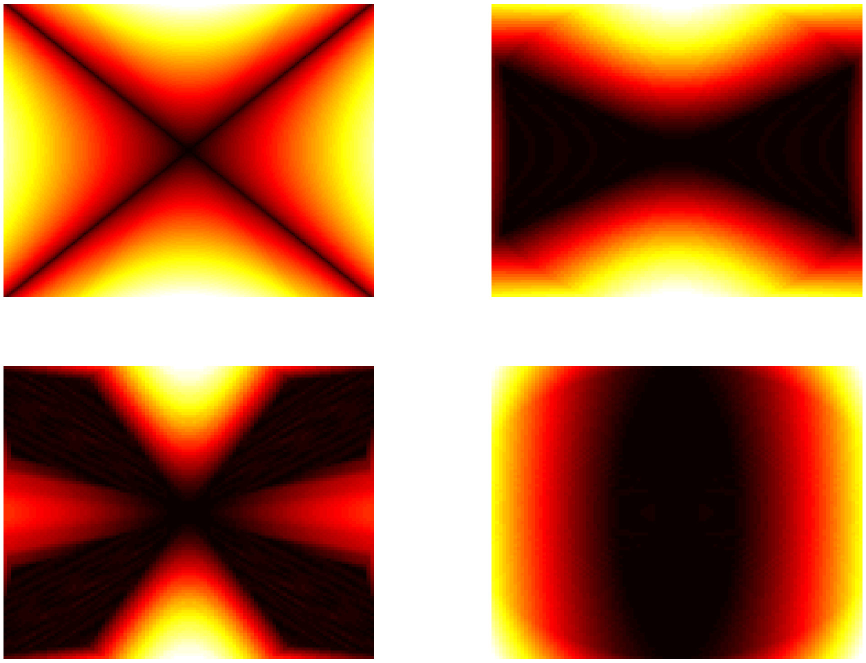

Figure 3.2: The scalability maps of 4 scalable frames. The scalability maps are projections onto the xy-plane that attain a value of 0 (Black) when the frame is scalable. In these examples, the center of the map is the trivial solution $u = \mathbf{0}$, and due to these frames being scalable, we see that there are lines or areas away from the origin that also have residual 0.

Chapter 4: Generating Frame Scalings through Optimization

4.1 Overview

As we have discussed the spectral characterizations of scalable frames, we now turn to the implementation of scaling techniques. We shall show in this chapter that the scalability problem can be solved as a convex optimization problem. Furthermore, the type of solution obtained can be determined through careful selection of the objective function (e.g. sparse or dense scalings). At the end of this chapter, we give a treatment of scalings for frames in high dimensions and high redundancy, along with some numerical experiments to show the effectiveness of these techniques.

Some of the work presented here is also presented in [29], and when introduced, citation of the article is given. Current techniques for scaling generally fall into three distinct categories. In [32], the authors construct scalings by concatenating orthonormal bases with selected vectors. With this construction, the scaling coefficients can be explicitly obtained in certain cases. The author in [69] presents a complete characterization of frames in $\mathbb{R}^2$. Finally, in [21], the authors formulate an optimization problem seeking to generate a tight frame, or in the cases where the frame is not scalable, produce a well-conditioned frame.

Recall that the *primal* and *dual* linear mathematical programming problems are defined respectively, as follows:

$$\text{minimize: } c^T x$$

$$\text{subject to: } Ax = b$$

$$x \geq \mathbf{0}.$$

$$\text{maximize: } b^T y$$

$$\text{subject to: } A^T y \leq c$$

$$y \in \mathbb{R}^n.$$

Theorem 4.1.1 (Strong Duality)**.** If either the primal or dual problem has a finite optimal value, then so does the other. The optimal values coincide, and optimal solutions to both the primal and dual problems exist.

Theorem 4.1.2 (Complimentary Slackness)**.** Let x^* and $y for these frames^*$ be feasible solution vectors to the primal and dual problems respectively. Let A be an n by m matrix, where A_j denotes the jth column and a_i denotes the ith row of A. Then x^* and y^* are optimal solutions to their respective problems if and only if

$$y_i(a_i \cdot x - b_i) = 0 \text{ for all } i = 1, \ldots, n,$$

and

$$x_i(c_j - y^T A_j) = 0 \text{ for all } j = 1, \ldots, m.$$

Theorem 4.1.3. [11, Farkas' Lemma] Let A be a matrix of dimensions m by n and let b be a vector in $\mathbb{R}^m$. Then, exactly one of the following two alternatives holds:

1. There exists some $x \geq \mathbf{0}$ such that $Ax = b$.

2. There exists some vector p such that $p^T A \geq \mathbf{0}$ and $p^T b < 0$.

4.2 Scalability on a Convex Domain

Consider the sets $\mathcal{S}_1$ and $\mathcal{S}_2$ given by

$$\mathcal{S}_1 := \{u \in \mathbb{R}^m \,|\, F(\Phi)u = \mathbf{0}\,,\, u \geq \mathbf{0}\,,\, u \neq \mathbf{0}\},$$

and

$$\mathcal{S}_2 := \{v \in \mathbb{R}^m \,|\, F(\Phi)v = \mathbf{0}\,,\, v \geq \mathbf{0}\,,\, \|v\|_1 = 1\}.$$

$\mathcal{S}_1$ is a subset of the null space of $F(\Phi)$, and each $u \in \mathcal{S}_1$ is associated a scaling matrix X_u, defined as

$$X_u := (X_{ij})_u = \begin{cases} \sqrt{u_i} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{S}_2 \subset \mathcal{S}_1 \cap B_{\ell^1}$ where B_{ℓ^1} is the unit ball under the ℓ^1 norm.

We observe that a frame $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ is scalable if and only if there exists a scaling matrix X_u with $u \in \mathcal{S}_1$. Consequently, one can associate to X_u a scaling matrix X_v with $v \in \mathcal{S}_2$. The normalized set $\mathcal{S}_2$ ensures that the constraints in the optimization problems, to be presented, are convex.

Theorem 4.2.1. [29, Theorem 2.1] Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^m$ be a frame, and let $f : \mathbb{R}^m \to \mathbb{R}$ be a convex function. Then the program

$$\begin{aligned} \text{minimize: } & f(u) \\ \text{subject to: } & F(\Phi)u = \mathbf{0} \\ & \|u\|_1 = 1 \\ & u \geq \mathbf{0} \end{aligned} \tag{4.1}$$

has a solution if and only if the frame Φ is scalable.

Proof. Any feasible solution u^* of (4.1) is contained in the set $\mathcal{S}_2$, which itself is contained in $\mathcal{S}_1$, and thus corresponds to a scaling matrix X_u.

Conversely, any $u \in \mathcal{S}_1$ can be mapped to a $v \in \mathcal{S}_2$ by appropriate scaling factor. This provides an initial feasible solution to (4.1), and as f is convex and the constraints are closed, convex, and bounded, there must exist a minimizer of (4.1). ■

Theorem 4.2.1 is very general in that the convex objective function f can be chosen so that the resulting frame has desirable properties. We now consider certain interesting examples of objective functions f. These examples can be related to the sparsity (or lack thereof) of the desired solution. Using a linear objective function promotes sparsity, while barrier objectives promote dense solutions (large support solutions, u).

4.3 Sparse Solutions

A number of works consider the problem of sparse scalings (minimal scalings) [14, 23, 32], and while much of the work focuses on creating *minimal scalings*, as opposed to finding the minimal scaling subset, we still present related results.

Definition 4.3.1 (Minimal Scalings)**.** Let a frame $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be scalable with scaling coefficients $\{x_k\}$ supported on $J \subseteq \{1, 2, \ldots, m\}$. If there is no proper subset of the frame vectors $\{\varphi_k\}_{k\in J}$ that is itself a scalable frame, then the scaling $\{x_k\}_{k\in J}$ is called J-minimal.

What can be gleamed from this definition is that if the solution found is "sparse enough", we have created a minimal scaling. Our goal is to use optimization to find such a scaling. Essentially, we wish to solve the minimal support problem ($\min : \|u\|_0$), and while this is NP-hard in general, there is a vast array of literature available to solve this problem [3, 18, 26]. We take the approach of solving an ℓ_1 minimization problem.

Much of the work on sparse solutions employs an ℓ_1 objective in some form or another. We present the definition here, because when mixed with the non-negativity constraints on u, we can write the norm as an inner product, and summarily solve problem (4.1) as a linear programming problem.

$$f|_{u\geq 0}(u) = \sum_{k=1}^{m} |u_k| = \sum_{k=1}^{m} u_k = \langle \mathbb{1}, u \rangle.$$

As the ℓ_1 norm "encourages" sparsity, methods like basis pursuit and interior-point methods are readily available to solve this problem [11, 26, 89]. To further encourage sparsity, weighted ℓ_1 norms can be applied [20]. An approach considered in this thesis uses the columns of $F(\Phi)$ itself. Define the objective function of (4.1) to be $f(u) := \langle a, u \rangle$,

$$\begin{aligned} \text{minimize: } & \langle a, u \rangle \\ \text{subject to: } & F(\Phi)u = 0 \\ & \|u\|_1 = 1 \\ & u \geq 0, \end{aligned} \tag{4.2}$$

where a is the multiplicative inverse of some measure of similarity between columns of $F(\Phi)$. Our goal is to take a non-negative combination of the columns, $F(\varphi_k)$, that result in a null vector. In the situation where columns of $F(\Phi)$ are anti-correlated, we obtain a null vector by adding. It is this property we wish to reward by inversely weighting columns in $F(\Phi)$ with high similarity.

4.3.1 Strong Dual Formulation

One of the advantages of linear programs is that they admit a strong dual formulation. To the primal problem (4.2) corresponds the following dual problem:

Proposition 4.3.2. [29, Proposition 2.2] Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame. The program

$$\begin{aligned} \text{maximize:}\, & w \\ \text{subject to:}\, & [F(\Phi)^T \ \mathbb{1}] \begin{bmatrix} v \\ \\ w \end{bmatrix} \leq a \\ & w \in \mathbb{R}\, , \, v \in \mathbb{R}^d \end{aligned} \tag{4.3}$$

is the strong dual of (4.2).

Proof. This result follows exactly from the construction of dual formulations for linear programs. The primal problem can be formulated as follows:

$$\begin{aligned} \text{minimize:}\, & \sum_{k=1}^m a_k u_k \\ \text{subject to:}\, & F(\Phi)u = \mathbf{0} \\ & \sum_{k=1}^m u_k = 1 \\ & u \geq \mathbf{0}. \end{aligned}$$

The strong dual of this problem is:

$$\begin{aligned} &\text{maximize: } w \\ &\text{subject to: } [F(\Phi)^T \ \mathbb{1}] \begin{bmatrix} v \\ \\ w \end{bmatrix} \leq a. \end{aligned}$$

■

Numerical optimization schemes generally consist of a search for an initial feasible solution, and then a search for an optimal solution. In analyzing the linear program formulation (4.2), we notice that we either have an optimal solution or the problem is infeasible, but there is no case when the problem is unbounded (due to the bounding constraint $\|u\|_1 = 1$). Any feasible solution must have norm 1, and thus, cannot be unbounded. The dual problem has the property that it either has an optimal solution, or is unbounded (from duality). Consequently, for any frame, Φ, $w = \min\{a\}$ and $v = 0$ is always a feasible solution to the dual problem. This removes the requirement that an initial solution be found [11].

4.3.2 Sparse Solutions from Combinatorics

We now briefly diverge from standard signal processing techniques to those in operations research and combinatorics to create a heuristic to find sparse solutions. This technique will be employed when discussing graph applications, but we present the optimization result here for thematic reasons.

We have formulated the ℓ_1 minimization problem above, and a wide range of literature is available on the subject of finding sparse solutions to this problem

[19, 20, 26, 89]. Under certain conditions, we can ensure that the minimum solution will be integer. We first present some necessary background from [91, 95].

Definition 4.3.3. (Totally Unimodular Matrices) A matrix A is totally unimodular if every square sub-matrix of A has determinant $\{0\,,\,\pm 1\}$. Furthermore, this implies that

Theorem 4.3.4. The linear program

$$\begin{aligned} \text{minimize: } & \langle c, x\rangle \\ \text{subject to: } & Ax \geq b \\ & x \geq \mathbf{0}, \end{aligned}$$

has an integer optimal solution for all integer vectors b, for which it has a finite optimal value, if and only if A is totally unimodular.

Proof. Let B denote the basis for the optimal solution. By Cramer's rule, we can solve for each variable,

$$x_i = \frac{\det(B|_i)}{\det(B)},$$

where $B|_i$ is the basis with the ith column replaced with the vector b. As B is a basis and totally unimodular, $\det(B) = \pm 1$, and as everything is integer, $\det(B|_i)$ will be integer as well. ■

This result can be generalized for other matrices where there are a finite number of distinct determinant values. Using this, we can characterize scalings for frames with this property (note the work of [91]).

Definition 4.3.5. (Bimodular Matrices) A matrix A is totally bimodular if every square sub-matrix of A has determinant $\{0\,,\,\pm 1\,,\,\pm 2\}$.

Theorem 4.3.6. Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^m$ be a frame. Also let the transformed frame $F(\Phi)$ be totally bimodular. Then the program

$$\begin{aligned} \text{minimize: } & \langle a, u\rangle \\ \text{subject to: } & F(\Phi)u = \mathbf{0} \\ & \|u\|_1 \geq 1 \\ & 0 \leq u \leq 1 \end{aligned} \tag{4.4}$$

has a solution if and only if the frame Φ is scalable. Furthermore, the solution u^* will have elements that are in the set $\{0\,,\,\frac{1}{2}\,,\,1\}$.

Proof. First dealing with the constraints, we can write the ℓ_1 norm as the inner product $\mathbb{1}^T u$, and the constraints $0 \leq u \leq 1$ as matrix inequalities,

$$\begin{bmatrix} -I \\ I \end{bmatrix} u \leq \begin{bmatrix} -\mathbb{1} \\ \mathbf{0} \end{bmatrix}.$$

Our new problem can be expressed as,

$$\text{minimize: } \langle a, u\rangle$$

$$\text{subject to: } \begin{bmatrix} F(\Phi) \\ \mathbb{1}^T \\ -I \\ I \end{bmatrix} [u] \begin{bmatrix} = \\ \geq \\ \geq \\ \geq \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \\ -\mathbb{1} \\ \mathbf{0} \end{bmatrix}$$

It is clear from this formulation that by adjoining the vector of ones and the identities, we have preserved the bimodularity of the matrix.

As with Theorem 4.2.1, if we have an optimal solution to the problem, we have a scaling for the frame. For the converse, we can normalize the scaling to lie in the interval $[0, 1]$.

Now mirroring the proof of Theorem 4.3.6 we shall prove the remainder of the result. Let B denote the basis for the optimal solution. By Cramer's rule, we can solve for each variable,

$$u_k = \frac{\det(B|_k)}{\det(B)},$$

where $B|_k$ is the basis with the kth column replaced with the vector on the right hand side. As B is a basis and totally bimodular, $\det(B) \in \{\pm 1\,, \pm 2\}$, and as everything is integer, $\det(B|_k)$ will be integer as well. If $\det(B|_k)$ is divisible by 2, u_k will be an integer, and if $\det(B|_k)$ is not divisible, u_k will have a factor of $\frac{1}{2}$ (the variable will be exactly $\dfrac{1}{2}$ from the constraint $u \leq 1$). ■

To further encourage sparsity, we can employ a weighted objective function. If we set the normalization constraint to $\|u\|_1 = s$, where s is the desired support of the solution, we ensure that the solution found will sum to s. Given the assumptions in Theorem 4.3.6, if our solution has a $\frac{1}{2}$ present, it must then have a pair. And so our goal turns to stopping these pairs from occurring. Take the objective function,

$$f(u) := \sum_{k=1}^{m} 2^{k-1} u_k.$$

What this objective aims to do is penalize non-sparse solutions by unbalancing the objective function coefficients.

4.4 Dense Solutions

A sparse solution to the linear program produces a frame in which the frame elements corresponding to the zero coefficients are removed. In contrast, one may wish to have a full solution, that is, one may want to retain all of, or most of, the frame vectors. To enforce this property, we introduce two types of barrier objective results.

Proposition 4.4.1. [29, Proposition 2.3] Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame, and define $0 \leq \epsilon \ll 1$. If the problem

$$\begin{aligned} \text{maximize: } & \sum_{k=1}^{m} \ln(u_k + \epsilon) \\ \text{subject to: } & F(\Phi)u = \mathbf{0} \\ & \|u\|_1 = 1 \\ & u \geq \mathbf{0}. \end{aligned} \tag{4.5}$$

has a feasible solution u^* with a finite objective function value, then the frame Φ is scalable, and the scaling matrix X is a diagonal operator where the elements are the square-roots of the feasible solution u^*. Moreover, for $\epsilon = 0$, if a finite solution u^* exists, all elements of u^* are strictly positive.

Proof. Assume u^* is a feasible solution to (4.5) with $0 < \epsilon \ll 1$ and the objective function finite. Then from the proof of Theorem 4.2.1, we have that the frame Φ is scalable. Now assume $\epsilon = 0$. If one of the variables u_k were zero, then the objective function would have a value of $-\infty$. Since we assume the function is finite, this cannot be the case. A negative value for u_k would result in the objective function value being undefined, this also cannot be the case due to the finite objective. Therefore, u_k must be positive for all k. ■

While the barrier method above finds a non-sparse solution if one exists, for $\epsilon = 0$, there is the possibility that the problem is infeasible and the frame is still scalable. The formulation we present below is guaranteed to find a non-sparse solution if one exists, but will always return a solution if the frame is scalable (even if the only solution is sparse).

Proposition 4.4.2. [29, Proposition 2.4] Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame. If the problem

$$\begin{aligned} \text{maximize: } & \min_{k=1,\ldots,m} \{u_k\} \\ \text{subject to: } & F(\Phi)u = \mathbf{0} \\ & \|u\|_1 = 1 \\ & u \geq \mathbf{0}. \end{aligned} \tag{4.6}$$

has a feasible solution u^* with a finite objective function value, then the frame Φ is scalable, and the scaling matrix X is a diagonal operator where the elements are the square-roots of the feasible solution u^*. Moreover, a solution exists with positive elements if and only if the solution produced by solving this problem has positive elements.

Proof. To show this, we shall rewrite this problem as a linear program.

$$\begin{aligned}
\text{maximize:}\, & t \\
\text{subject to:}\, & F(\Phi)u = \mathbf{0} \\
& \sum_{k=1}^{m} u_k = 1 \\
& t \leq u_k \\
& t > 0\,,\; u \geq \mathbf{0}.
\end{aligned}$$

Here, t is an auxiliary variable, taken to be the minimum element of u. This linear program can be solved to optimality. Moreover, as this problem is convex, the optimum achieved is global. If the objective function at optimality has a value of 0, then there can exist no solution with all positive coefficients. ■

4.5 Numerical Comparison of Formulations

We have implemented the formulations above, in the programming language Matlab. This decision is primarily due to flexibility in syntax, its ubiquitous use by the scientific community, and the wide availability of support and toolboxes. In particular, the optimization, linear algebra, and sparse matrix support, make it an ideal choice. Specifically, we have implemented the linear programming formulation [PLP] and its weighted variant [WLP] (4.2) , the strong dual formulation [DLP] (4.3), and the dense formulation (maximin program [MMLP] (4.6)). We omit the nonlinear formulation to avoid any issues with unboundedness of the objective function. We test three aspects of the formulations:

1. Solution Type: Do solutions bias towards certain frame elements?

2. Sparsity: How does the support change with each formulation?

3. Computation Time: Do the methods have similar computational requirements?

We omit discussion of the third item, as this is more implementation dependent, and this is not the focus of the thesis. We shall still mention this as a comparison tool when we discuss frame in high dimensions.

We consider frames in the dimensions $n \in \{2, 5, 10\}$, with the number of frame elements varying with the dimension. We first consider frames with lower redundancy $m \in \{n+1, 2n+1, 3n+1\}$, and then consider high-redundancy frames $m = \{n^2+n+1, 2n^2+n+1, 3n^2+n+1, 4n^2+n+1, 5n^2+n+1\}$. For any tests of the frame properties, we perform 500 trials (a frame of the specified size is generated 500 times).

4.5.1 Solution Type

Solution type matters in the sense that we desire to transform a given frame into a tight frame while retaining properties inherent to the frame. The obtained solutions are for the four different formulations sorted in ascending order, and then averaged across the 500 trials. The resulting solution is shown for frames with the highest redundancy tested $m = 5n^2 + n + 1$. See Figure 4.1

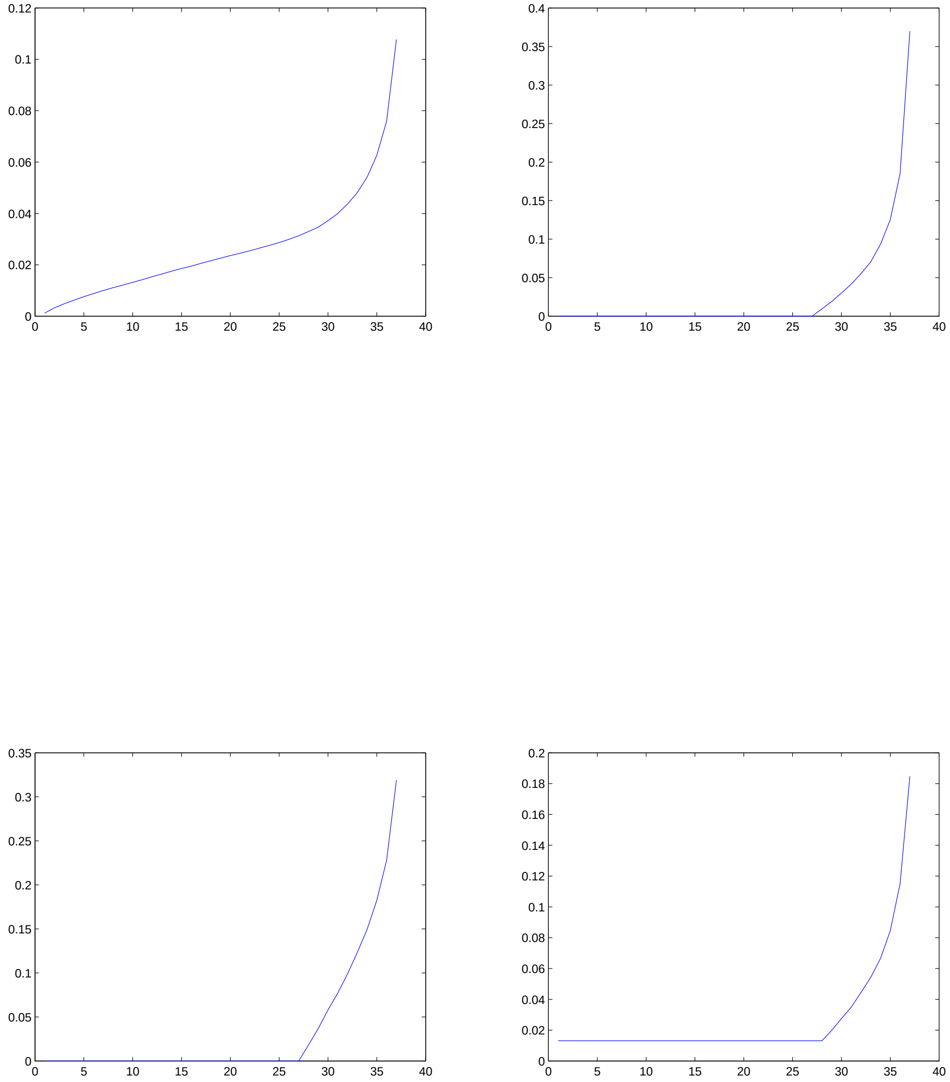

Figure 4.1: The average solutions (sorted) produced over 500 trials by the methods: PLP (Top-Left), WLP (Top-Right), DLP (Bottom-Left), and MMLP (Bottom-Right).

4.5.2 Sparsity

We present this section mainly to show the effectiveness of the weighted linear programming and the dual linear programming formulations. Ultimately, we find that the sparsity level achieved is a percentage of the redundancy, which we then describe from the frames and optimization point of view. We omit sparsity results for the maximin formulation [MMLP], as the results obtained were all non-sparse (save a few outliers).

The results are averaged across 500 trials as before, and we also only consider Gaussian frames. A scaling weight x_k is considered zero if its value is below the threshold 10^{-10}. This is mostly to account for round-off effects that occur when taking square-roots and the like. The results are presented in tables below as a sparsity percentage (1 - average percent of non-zero values), and any fields with "-" did not produce any scalable frames to be tested. All percentages were rounded to the nearest tenth of a percent. See Table 4.1, Table 4.2, and Table 4.3

The results show the consistency in the sparsity achieved across frame dimension, n, and across problem formulation. It's clear from the results that the standard linear programming formulation doesn't produce sparse results, and in the few cases where the weights are sparse, the frame has low redundancy. This consistency in sparsity percentage hints at a fixed number of frame elements that are retained in the sparse scaling, and indeed this is the case, as the number of elements retained is the length of the transformed vectors $(d+1) = \frac{n(n+1)}{2}$. No matter how many frame elements are present, the sparse (seemingly the sparsest) solution obtained will be approximately this constant $d+1$. To explain this, we need look no further than the number of constraints present in the linear programming constraints,

$$\begin{aligned} F(\Phi)u &= 0 \\ \|u\|_1 &= 1 \\ u &\geq 0. \end{aligned}$$

Ignoring the non-negativity constraint $u \geq 0$ for the moment, the linear system will have a solution, generally, when there are as many frame elements as constraints, $d+1=m$. Furthermore, the sparsest solution that we generally expect to achieve will be a solution where the non-zero scaling coefficients will correspond to a basis (of size $d+1$) in the linear system. This hints at the sparse solutions being the sparsest possible solution, but also hints at the requirement that the number of frame elements required to scale a frame should be at least $d+1$.

It should be noted here, that the solution achieved from linear programming can be highly dependent upon the algorithm employed. Interior point methods [11] admit polynomial-time schemes, but as the objective function in not strictly convex, the methods can often lead to non-sparse solutions. The Simplex algorithm on the other hand, visits the extreme points of the polytope, which produces sparse solutions in most cases [11]. Instead of electing to use the “best” method (in terms of inducing sparsity), we choose an average case, and use an interior point scheme.

$n = 2$	Frame Elements							
Formulation	3	5	7	7	11	15	19	23
PLP	00.0	00.0	00.0	00.0	00.0	00.0	00.0	00.0
WLP	00.0	40.0	57.1	57.1	72.7	80.0	84.2	86.9
DLP	00.0	40.0	57.1	57.1	72.7	80.0	84.2	86.9

Table 4.1: The percentage of zero elements in the scaling weights x_k.

$n = 5$	Frame Elements							
Formulation	6	11	16	31	56	81	106	131
PLP	-	-	-	00.1	00.1	00.0	00.0	00.0
WLP	-	-	-	51.6	73.2	81.4	85.8	88.5
DLP	-	-	-	51.6	73.2	81.4	85.8	88.5

Table 4.2: The percentage of zero elements in the scaling weights x_k.

$n = 10$	Frame Elements							
Formulation	11	21	31	111	211	311	411	511
PLP	-	-	-	00.4	00.0	00.0	00.0	00.0
WLP	-	-	-	50.4	73.9	82.3	86.6	89.2
DLP	-	-	-	50.4	73.9	82.3	86.6	89.2

Table 4.3: The percentage of zero elements in the scaling weights x_k.

Dimension	Average Computation Time
$n = 10$	0.0080
$n = 25$	11.9384
$n = 50$	271.4127

Table 4.4: The average computation time, in seconds, required to find a scaling for an $n \times 4n^2$ frame using the PLP formulation. This average is taken over 50 trials.

4.6 Frames in High Dimensions

So far we have kept the frame dimension low for computational reasons. The size of the linear system to be solved increases as the square of the dimension n. A further confounding issue are the number of frame elements required to reach scalability (also proportional to n^2). These properties together make exact scaling computationally infeasible. In this section we employ schemes designed to perform a scaling for large frames approximately. We perform 50 trials, and present the average computation time, and the average condition number achieved (as our metric of how much we have scaled the frame). To give a baseline for the efficiency of the methods presented below, we also present the average time to employ the PLP formulation on moderately sized frames. See Table 4.4.

4.6.1 Matching Pursuit

In general, there will be no true solution to the scalability problem for a small number of frame elements (or low redundancy). So we present a way to find an approximate solution using matching pursuit as the base algorithm.

Given a set of vectors $A = \{A_i\}_{i=1}^m \subset \mathbb{R}^n$ and a vector $b \in \mathbb{R}^n$, if we are asked to find the best approximation (in the ℓ_2 sense) of b by the elements of A, we have the following problem,

$$\min_{x \in \mathbb{R}^n} : \|Ax - b\|_2^2.$$

There is a closed-form solution to this problem, where x is the pseudo-inverse ($A^+ := (AA^T)^{-1}$) applied to the vector b. If we were then asked to only use at most s (with $s << m$) columns of A, we could come up with a number of solutions of a similar sort to the pseudo-inverse. One such solution would be to find the least-squares solution and project onto the positive orthant (we discuss this in a following section). This is equivalent to finding a scalable frame with exactly s non-zero weights and $A := F(\Phi)$ and $b := \mathbf{0}$. Our problem is

$$\min_{u \in \mathbb{R}^n\,,\, u \neq \mathbf{0}} : \|F(\Phi)u - \mathbf{0}\|_2^2.$$

If we relax the requirement that $\|F(\Phi)u - \mathbf{0}\|_2^2 = 0$, there are a number of approximate schemes that we can create. We now present the matching pursuit formulation.

This method starts by finding the two vectors in $F(\Phi)$ whose normalized inner product is most negative (or whose correlation is closest to -1). An inner product near -1 implies that the vectors are parallel to each other, and facing opposite directions. This assures that their sum will be approximately $\mathbf{0}$ for an appropriate positive scaling constant. This constant can be obtained for $F(\varphi_k)$ and $F(\varphi_{k'})$ by writing the optimization problem,

$$a^* = \arg\min_{a\in\mathbb{R}} : \|F(\varphi_k) - aF(\varphi_{k'})\|_2^2.$$

The solution is

$$a^* = -\frac{\langle F(\varphi_k), F(\varphi_{k'})\rangle}{\|F(\varphi_{k'})\|_2^2}.$$

We set $u_k = 1$ and $u_{k'} = a^*$. This is the start of our approximation. After this initialization step, we will have 2 elements in our approximation (u_k and $u_{k'}$). From here, we can add elements to our approximation by choosing columns that are "most" negatively correlated with our current approximation until either s elements have been added to the solution, or there are no more negatively correlated columns in $F(\Phi)$. Our solution $\widehat{u}^*$ produces an approximate tight frame.

The initial step of this greedy search requires the comparison of $\binom{m}{2} := \frac{m!}{2(m-2)!}$ inner products. The ith step in our process requires $m-i$ inner product comparisons, for a total complexity of,

$$\begin{aligned} O\left[\frac{m!}{2(m-2)!} + \sum_{i=1}^{s-2}(m-i)\right] &= O\left[\frac{m(m-1)}{2} + \sum_{i=1}^{s-2} m - \sum_{i=1}^{s-2} i\right] \\ &= O\left[\frac{m(m-1)}{2} + (s-2)m - \frac{(s-2)(s-1)}{2}\right] \\ &= O[m^2]. \end{aligned}$$

We can reduce the number of operations by changing our initialization step to a process that randomly selects p initial columns randomly from $F(\Phi)$ to produce the

Algorithm 2 Scalable Frame Matching Pursuit

```
Given unit norm $F(\Phi)$ and tolerance $\tau$
$u \leftarrow \mathbf{0}$
$d_k \leftarrow F(\varphi_k) - \mu_{F(\varphi_k)}$
$D \leftarrow [d_1, d_2, \dots, d_m]$
$C \leftarrow D^T D$
$(i, j) \leftarrow \arg\min_{ij}\{C_{ij}\}$
if $C_{ij} \geq 0$ then
    return $u$
end if
$u_i \leftarrow 1$
$u_j \leftarrow -\langle F(\varphi_i), F(\varphi_j)\rangle$
$\mathcal{S} \leftarrow \{i, j\}$
$r \leftarrow F(\Phi)u$
while  $r > \tau$  do
    $c \leftarrow \langle r, D_{\mathcal{S}^c}\rangle$
    $i \leftarrow \arg\min_j\{c_j\}$
    if $c_i \geq 0$ then
        return $u$
    end if
    $u_i \leftarrow \langle r, F(\varphi_i)\rangle$
    $r \leftarrow F(\Phi)u$
    $\mathcal{S} \leftarrow \{\mathcal{S}, i\}$
end while
```

initial solution. Our complexity with this reduction, becomes

$$O[pm] << O[m^2].$$

4.6.2 Delayed Column Generation

As a further modification of the standard matching pursuit problem, we can selectively apply the transformation to a frame element $F(\varphi_k)$ as a means of memory management. The operation is relatively cheap in a computational sense, but to store all transformed vectors in memory requires $O[n^2m]$ bytes, and this does not take into consideration any other memory requirements for operations performed on $F(\Phi)$. When employing matching pursuit, there is no requirement that the columns, $F(\varphi_k)$, be stored when not in use. In effect we can delay generating the column until its use is required. For more background on the method applied to linear programs see [11], and for a treatment given for combinatorics, see [24].

Employing this technique though, does require a modification of the selection process. Currently, we take the best column and add it into the basic solution. This would require us to transform all other frame elements for each iteration. And while the frame transform is relatively cheap to implement, a more efficient scheme would only look at a few of these frame elements.

4.6.3 Projected Least-Squares

The second large-scale method we present here, takes the least-squares solution to the system,

$$\begin{bmatrix} F(\Phi) \\ \\ \mathbb{1}^T \end{bmatrix} u = \begin{bmatrix} \mathbf{0} \\ \\ 1 \end{bmatrix},$$

and project onto the non-negative orthant $\mathbb{R}_+$. For situations where speed is the determining factor, finding a fast solution and thresholding the negative values is a standard approach [42]. This method is far less precise, and in cases when the frame is well-conditioned already, this method may increase condition number. It's only advantageous to use this method when the frame considered is large and ill-conditioned.

4.6.4 Computational Results

Here, we consider frames with in high dimension and with large redundancy. We perform a similar analysis to that of the low-dimensional case above. See Table 4.5.

It speaks to set of possible scalings, that for methods that are highly dependent upon the starting conditions, we are able to find "good" approximate solutions. This seems due to two main properties of the scalability problem. First, for frames with frame elements larger than the transformed dimension, $m > d$, the system $F(\Phi)u = \mathbf{0}$ becomes under-determined and the number of solutions (and near-solutions) becomes infinite. Second, the number of minimal (and near-minimal)

	Average Computation Time		
Dimension	PLP	MP	LS
$n = 10$	0.0080	0.0305	0.0030
$n = 25$	11.9384	4.8171	0.1925
$n = 50$	271.4127	83.8872	24.3478
$n = 100$	-	3770.9	1215.7

Table 4.5: The average computation time, in seconds, required to find a scaling for an $n \times 4n^2$ frame using the PLP formulation, the matching pursuit formulation MP, and the least-squares formulation LS. This average is taken over 50 trials. A "-" indicates that the method did not produce a result in the computation time allowed.

scalings appears to be dense for high redundancy frames.

Chapter 5: Frames Drawn from Distributions

5.1 Overview

The aim of this chapter is to characterize the space of scalable frames generated from discrete and continuous distributions. The four distributions considered are Bernoulli, Rademacher, Uniform, and Gaussian, and we bound the probability of a frame being scalable. For frames sampled from the two discrete distributions, we prove bounds related to the expected time until an orthonormal basis is formed from the sampling, and for the Gaussian and Uniform distributions, we prove a result using the symmetry of the distribution and Farkas' Lemma that shows the chance of a non-scalable frame decreases as a power law of the frame elements. We present some intuition as to how the continuous distribution results can be extended to the class of symmetric distributions, and close the chapter with some numerical results.

We first define the distributions we use, and give any relevant statistics. For more information on probability theory, the interested reader is encouraged to review the works of [85, 94].

Definition 5.1.1 (Bernoulli Frames)**.** Denote the Bernoulli distribution as a function supported on the set $\{0,1\}$ with equal probability. The corresponding probability mass function is,

$$p(X) = \begin{cases} 1/2 & X = 0 \\ 1/2 & X = 1 \end{cases}. \tag{5.1}$$

A *Bernoulli frame* Φ is defined to be a frame where each entry is chosen i.i.d. from the Bernoulli distribution.

Definition 5.1.2 (Rademacher Frames)**.** Denote the Rademacher distribution as a function supported on the set $\{-1,1\}$ with equal probability. The corresponding probability mass function is,

$$p(X) = \begin{cases} 1/2 & X = 1 \\ 1/2 & X = -1 \end{cases}. \tag{5.2}$$

A *Rademacher frame* Φ is defined to be a frame where each entry is chosen i.i.d. from the Rademacher distribution.

Definition 5.1.3 (Uniform Frames)**.** Denote the Uniform distribution as a function supported on the interval $[-1,1]$ with constant probability. The corresponding density function is,

$$p(X) = 1/2 \text{ for } X \in [-1,1]. \tag{5.3}$$

A *Uniform frame* Φ is defined to be a frame where each entry is sampled i.i.d. from the Uniform distribution.

Definition 5.1.4 (Gaussian Frames)**.** A *Gaussian frame* Φ is defined to be a frame where each entry is sampled i.i.d. from the Standard Normal distribution with 0 mean and unit standard deviation.

5.2 Bernoulli Frames

For frames drawn from a Bernoulli distribution, we can characterize the entire space of scalable frames. An interpretation of the scalability criterion relies on positive and negative values in the frame transform $F(\cdot)$. This is never satisfied when the frame entries are all positive, or similarly, all negative, but in the case where frame entries are chosen from non-negative/non-positive values (values of 0 are allowed), the frame is scalable only if a subset of the frame elements φ_k form an orthogonal basis (i.e. scalings of the standard basis vectors e_i).

Lemma 5.2.1. Let $\Phi = \{\varphi_k\}_{k=1}^m$ be a Bernoulli frame where each entry is drawn from the Bernoulli distribution. Then Φ is scalable if and only if it contains an n-dimensional identity sub-matrix.

Proof. If Φ contains an identity sub-matrix, it is apparent that Φ is scalable, as the identity is an ONB.

To show the converse, we consider what happens to Bernoulli vectors under the frame transformation $F_i(\cdot)$. $F_i(\varphi_k) = (\varphi_k(i)\varphi_k(j)) \geq 0$ for all $1 \leq i < j \leq n$. For a frame to be scalable, we require that a non-negative sum of these values equal 0 ($\sum_{k=1}^m u_k F_i(\varphi_k) = \mathbf{0}$). Excluding the trivial case of $u = 0$, this is only possible if $F_i(\varphi_k) = 0$ for all $1 \leq i < j \leq n$ for some subset of $\{1, \ldots, m\}$. The product pairs $\varphi_k(i)\varphi_k(j)$ are 0 only if $\varphi_k = e_{(\cdot)}$.

To show that the full standard basis is required it is sufficient to notice that without the standard basis, any proposed scaling $\{x_k\varphi_k\}$ would not span all of

$\mathbb{R}^n$. ■

For other frames with non-negative/non-positive entries, we can extend this to a requirement that the frame elements have disjoint support and a single non-zero element.

Theorem 5.2.2 (Non-negative Frames)**.** Let $\Phi = \{\varphi_k\}_{k=1}^m$ be a frame with only non-negative or only non-positive entries. Then Φ is scalable if and only if it contains an n-dimensional identity sub-matrix (after normalizing each column of Φ to be unit norm).

Proof. By an argument equivalent to Lemma 5.2.1, it suffices to shown that we have the same requirement by a change of scaling variables $\widetilde{x}_k = \frac{1}{\|\varphi_k\|_2} x_k$. ■

We now derive the probabilities of a Bernoulli frame containing the identity sub-matrix. We consider the cases when the frame is sampled from the set of all possible Bernoulli vectors of size n, both with and without replacement.

Lemma 5.2.3. Let Ψ be a random subset of m elements sampled i.i.d.. Also let this random subset be sampled with replacement from the set of all possible Bernoulli vectors in $\mathbb{R}^n \backslash \{\mathbf{0}\}$ with $2 \leq n < m < \infty$. The probability that Ψ contains an orthonormal basis $\{e_i\}_{i=1}^n$ can be expressed with the following equality,

$$P\left(\{e_1, e_2, \ldots, e_n\} \subset \Psi\right) = 1 - \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{2^n - 1}\right)^i \left(\frac{2^n - 1 - (n-i)}{2^n - 1}\right)^{m-i}. \quad (5.4)$$

Proof. We start by considering the complement of our probability,

$$\begin{aligned}
P\left(\{e_i\}_{i=1}^n \subset \Psi\right) = 1-&P\left(\{e_i\}_{i=1}^n \not\subset \Psi\right), \\
P\left(\{e_i\}_{i=1}^n \subset \Psi\right) = 1-&\sum_{i=1}^{n} P([\{e_i\} \subset \Psi] \cap [\{e_q\}_{q\neq i} \not\subset \Psi]) - \dots \\
&\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} P([\{e_i, e_j\} \subset \Psi] \cap [\{e_q\}_{q\neq i,j} \not\subset \Psi]) - \dots \\
&\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{k=j+1}^{n} P([\{e_i, e_j, e_k\} \subset \Psi] \cap [\{e_q\}_{q\neq i,j,k} \not\subset \Psi]) - \dots \\
&\vdots \\
&\sum_{i=1}^{n} P([\{e_q\}_{q\neq i} \subset \Psi] \cap [\{e_i\} \not\subset \Psi]).
\end{aligned} \tag{5.5}$$

Notice that for any set of indices Ω, $P\left([\{e_i\}_{i\in\Omega} \subset \Psi] \cap [\{e_j\}_{j\in\Omega^c} \not\subset \Psi]\right) = a$, where a only depends on the size of the set Ω (Ω^c denotes the compliment). Hence, we have,

$$\begin{aligned}
P\left(\{e_i\}_{i=1}^n \subset \Psi\right) = 1-&\binom{n}{0} P\left(\{e_i\}_{i=1}^n \not\subset \Psi\right) - \dots \\
&\binom{n}{1} P\left([\{e_1\} \subset \Psi] \cap [\{e_i\}_{i\neq 1} \not\subset \Psi]\right) - \dots \\
&\binom{n}{2} P\left([\{e_1, e_2\} \subset \Psi] \cap [\{e_i\}_{i\neq 1,2} \not\subset \Psi]\right) - \dots \\
&\vdots \\
&\binom{n}{n-1} P\left([\{e_i\}_{i\neq n} \subset \Psi] \cap [\{e_n\} \not\subset \Psi]\right).
\end{aligned} \tag{5.6}$$

The probability of choosing any particular element in the set of Bernoulli vectors is $\frac{1}{2^n-1}$, and since our choices are independent, we can use the multiplicative rule to expand the probabilities in the following manner,

$$\begin{aligned}
P\left(\{e_i\}_{i=1}^{n} \not\subset \Psi\right) &= \left(\frac{2^n - 1 - (n)}{2^n - 1}\right)^{m}, \\
P\left([\{e_1\} \subset \Psi] \cap [\{e_i\}_{i\neq 1} \not\subset \Psi]\right) &= \left(\frac{1}{2^n - 1}\right)^{1} \left(\frac{2^n - 1 - (n-1)}{2^n - 1}\right)^{m-1}, \\
P\left([\{e_1, e_2\} \subset \Psi] \cap [\{e_i\}_{i\neq 1,2} \not\subset \Psi]\right) &= \left(\frac{1}{2^n - 1}\right)^{2} \left(\frac{2^n - 1 - (n-2)}{2^n - 1}\right)^{m-2}, \\
&\vdots \\
P\left(\{e_i\}_{i\neq n} \subset \Psi \,\cap\, \{e_n\} \not\subset \Psi\right) &= \left(\frac{1}{2^n - 1}\right)^{n-1} \left(\frac{2^n - 1 - (1)}{2^n - 1}\right)^{m-n+1}.
\end{aligned} \tag{5.7}$$

Joining the probabilities and binomial coefficients $\binom{n}{i}$, and taking the summation, we have,

$$P\left(\{e_i\}_{i=1}^{n} \subset \Psi\right) = 1 - \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{2^n - 1}\right)^{i} \left(\frac{2^n - 1 - (n-i)}{2^n - 1}\right)^{m-i}. \tag{5.8}$$

■

Theorem 5.2.4 (Bernoulli Frame Scalability)**.** Let Ψ be a random subset of m elements sampled i.i.d.. Also let this random subset be sampled with replacement from the set of all possible Bernoulli frames in $\mathbb{R}^n \backslash \{\mathbf{0}\}$ with $2 \leq n < m < \infty$. The probability that Ψ is scalable can be bounded below with the following inequality,

$$P\left(\Psi \in \mathcal{S}F(n,m)\right) \geq 1 - \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{2^n - 1}\right)^{i} \left(\frac{2^n - 1 - (n-i)}{2^n - 1}\right)^{m-i}. \tag{5.9}$$

Proof. From Lemma 5.2.1, we have that Bernoulli frames are scalable if and only if they contain an orthonormal basis (i.e. they contain the identity). From Lemma 5.2.3, we know the probability of the identity matrix being present in a Bernoulli matrix. Given that the set of Bernoulli frames is contained in the set of Bernoulli matrices, and the set of Bernoulli matrices with identity sub-matrices is contained in the set of scalable Bernoulli frames. Let Ψ_1 and Ψ_2 be random sets from the collection of Bernoulli frames and Bernoulli vectors respectively, then we have the following result,

$$\begin{aligned} P\left(\Psi_1 \in \mathcal{S}F(n,m)\right) &= P\left(\{e_i\}_{i=1}^n \subset \Psi_1\right) \\ &\geq P\left(\{e_i\}_{i=1}^n \subset \Psi_2\right), \qquad (5.10) \\ &\geq 1 - \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{2^n - 1}\right)^i \left(\frac{2^n - 1 - (n-i)}{2^n - 1}\right)^{m-i}. \end{aligned}$$

■

Lemma 5.2.5. Let Ψ be a random subset of m elements sampled i.i.d.. Also let this random subset be sampled without replacement from the set of all possible Bernoulli vectors in $\mathbb{R}^n \backslash \{\mathbf{0}\}$ with $2 \leq n < m \leq 2^n - 1$. The probability that Ψ contains an orthonormal basis $\{e_i\}_{i=1}^n$ can be expressed with the following equality,

$$P\left(\{e_i\}_{i=1}^n \subset \Psi\right) = 1 - \sum_{i=0}^{n-1} \binom{n}{i} \frac{\binom{2^n-1-n}{m-i}}{\binom{2^n-1}{m-i}} \frac{1}{\binom{2^n-1-(m-i)}{i}}. \qquad (5.11)$$

Proof. We start by considering the complement of our probability,

$$\begin{aligned} P\left(\{e_i\}_{i=1}^n \subset \Psi\right) &= 1 - P\left(\{e_i\}_{i=1}^n \not\subset \Psi\right), \\ P\left(\{e_i\}_{i=1}^n \subset \Psi\right) &= 1 - \sum_{i=1}^{n} P([\{e_i\} \subset \Psi] \cap [\{e_q\}_{q\neq i} \not\subset \Psi]) - \ldots \\ & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P([\{e_i, e_j\} \subset \Psi] \cap [\{e_q\}_{q\neq i,j} \not\subset \Psi]) - \ldots \\ & \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} P([\{e_i, e_j, e_k\} \subset \Psi] \cap [\{e_q\}_{q\neq i,j,k} \not\subset \Psi]) - \ldots \\ & \vdots \\ & \sum_{i=1}^{n} P([\{e_q\}_{q\neq i} \subset \Psi] \cap [\{e_i\} \not\subset \Psi]). \end{aligned} \qquad (5.12)$$

Notice that for any set of indices Ω, $P\left([\{e_i\}_{i\in\Omega} \subset \Psi] \cap [\{e_j\}_{j\in\Omega^c} \not\subset \Psi]\right) = a$, where a only depends on the size of the set Ω (Ω^c denotes the compliment). Hence, we have,

$$\begin{aligned} P\left(\{e_i\}_{i=1}^n \subset \Psi\right) = 1 - &\binom{n}{0} P\left(\{e_i\}_{i=1}^n \not\subset \Psi\right) - \dots \\ &\binom{n}{1} P\left([\{e_1\} \subset \Psi] \cap [\{e_i\}_{i\neq 1} \not\subset \Psi]\right) - \dots \\ &\binom{n}{2} P\left([\{e_1, e_2\} \subset \Psi] \cap [\{e_i\}_{i\neq 1,2} \not\subset \Psi]\right) - \dots \\ &\vdots \\ &\binom{n}{n-1} P\left([\{e_i\}_{i\neq n} \subset \Psi] \cap [\{e_n\} \not\subset \Psi]\right). \end{aligned} \tag{5.13}$$

Considering the components in the difference separately, we can compute the probabilities exactly by considering the probability of selecting the corresponding number of basis elements, and corresponding number of non-basis elements,

$$\begin{aligned} P\left(\{e_i\}_{i=1}^n \not\subset \Psi\right) &= \frac{\binom{2^n-1-(n)}{m}}{\binom{2^n-1}{m}} \frac{1}{\binom{2^n-1-(m)}{0}}. \\ P\left([\{e_1\} \subset \Psi] \cap [\{e_i\}_{i\neq 1} \not\subset \Psi]\right) &= \frac{\binom{2^n-1-(n)}{m-1}}{\binom{2^n-1}{m-1}} \frac{1}{\binom{2^n-1-(m-1)}{1}}. \\ P\left([\{e_1, e_2\} \subset \Psi] \cap [\{e_i\}_{i\neq 1,2} \not\subset \Psi]\right) &= \frac{\binom{2^n-1-(n)}{m-2}}{\binom{2^n-1}{m-2}} \frac{1}{\binom{2^n-1-(m-2)}{2}}. \\ &\vdots \\ P\left([\{e_i\}_{i\neq n} \subset \Psi] \cap [\{e_n\} \not\subset \Psi]\right) &= \frac{\binom{2^n-1-(n)}{m-(n-1)}}{\binom{2^n-1}{m-(n-1)}} \frac{1}{\binom{2^n-1-(m-(n-1))}{n-1}}. \end{aligned} \tag{5.14}$$

Joining the probabilities and binomial coefficients $\binom{n}{i}$ and taking the summation, we have,

$$P\left(\{e_i\}_{i=1}^n \subset \Psi\right) = 1 - \sum_{i=0}^{n-1} \binom{n}{i} \frac{\binom{2^n-1-n}{m-i}}{\binom{2^n-1}{m-i}} \frac{1}{\binom{2^n-1-(m-i)}{i}}. \tag{5.15}$$

■

Theorem 5.2.6. Let Ψ be a random subset of m elements sampled i.i.d.. Also let this random subset be sampled without replacement from the set of all possible Bernoulli vectors in $\mathbb{R}^n \backslash \{\mathbf{0}\}$ with $2 \leq n < m \leq 2^n - 1$ such that the set forms a frame for $\mathbb{R}^n$. The probability that Ψ is scalable can be bounded below by the following inequality,

$$P\left(\Psi \in \mathcal{S}F(n,m)\right) \geq 1 - \sum_{i=0}^{n-1} \binom{n}{i} \frac{\binom{2^n-1-n}{m-i}}{\binom{2^n-1}{m-i}} \frac{1}{\binom{2^n-1-(m-i)}{i}}. \tag{5.16}$$

Proof. From Lemma 5.2.1, we have that Bernoulli frames are scalable if and only if they contain an orthonormal basis (i.e. they contain the identity). From Lemma 5.2.5, we know the probability of the identity matrix being present in a Bernoulli matrix. Given that the set of Bernoulli frames is contained in the set of Bernoulli matrices, and the set of Bernoulli matrices with identity sub-matrices is contained in the set of scalable Bernoulli frames. Let Ψ_1 and Ψ_2 be random sets from the collection of Bernoulli frames and Bernoulli vectors respectively, then we have the following result,

$$\begin{aligned} P\left(\Psi_1 \in \mathcal{S}F(n,m)\right) &= P\left(\{e_i\}_{i=1}^n \subset \Psi_1\right) \\ &\geq P\left(\{e_i\}_{i=1}^n \subset \Psi_2\right), \qquad (5.17) \\ &\geq 1 - \sum_{i=0}^{n-1} \binom{n}{i} \frac{\binom{2^n-1-n}{m-i}}{\binom{2^n-1}{m-i}} \frac{1}{\binom{2^n-1-(m-i)}{i}}. \end{aligned}$$

■

5.3 Rademacher Frames

If instead of drawing from the set $\mathcal{S} = \{0, 1\}$, we draw from the unimodular set $\mathcal{S} = \{-1, 1\}$, we have Rademacher frames. The results here are more positive, as frame transformation simplifies the system to be solved, and the possible solutions are more plentiful. The downside of attempting to analyze these frames by the probability of forming an orthogonal basis is the difficulty in forming the basis itself. This relates heavily to the problem of constructing *Hadamard matrices* (orthogonal ±1 bases) for general dimensions [1]. To avoid this issue, we restrict ourselves to frames of size 2^q that allow for the Syllvester construction [1] of Hadamard matrices. With this, we have the result that Rademacher frames for $\mathbb{R}^2$ are always scalable.

Lemma 5.3.1. Let $\varphi \in \mathbb{R}^n$ be a Rademacher vector. Then the frame transformation $F_0(\varphi)$ is the $(n-1)$ zero vector.

Proof.

$$\begin{aligned} F_0(\varphi) &= \varphi(1)^2 - \varphi(i)^2 \quad i = 2, 3, \dots n, \\ &= |\varphi(1)|^2 - |\varphi(i)|^2 \quad i = 2, 3, \dots n, \\ &= 1^2 - 1^2 \quad i = 2, 3, \dots n, \\ &= 0 \quad i = 2, 3, \dots n. \end{aligned} \tag{5.18}$$

■

Furthermore, due to this lemma, what we require for a Rademacher frame to be scalable, is that the weighted sum of the cross terms (which are also Rademacher) equal 0.

Proposition 5.3.2. Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^2$ be a Rademacher frame containing $m \geq 3$ elements. Furthermore, let the frame elements φ_k be drawn from the set of all possible Rademacher vectors with or without replacement. Then the frame is scalable.

Proof. Considering the space of vectors on the unit hypercube for dimension $n = 2$, we have the following set,

$$S = \{s_1 = [-1, -1]^T, s_2 = [-1, 1]^T, s_3 = [1, -1]^T, s_4 = [1, 1]^T\}. \tag{5.19}$$

To show that a frame is scalable, it is sufficient to show that the frame contains an ONB. Note that in dimension $n = 2$, the only spanning sets are ONB $\{\{s_1, s_2\}, \{s_1, s_3\}, \{s_4, s_2\}, \{s_4, s_3\}\}$, implying that all Rademacher frames are scalable in dimension $n = 2$. ■

Lemma 5.3.3. Let Ψ be a random subset of m elements sampled i.i.d.. Also let this random subset be sampled with replacement from the set of all possible Rademacher vectors in $\mathbb{R}^n \backslash \{\mathbf{0}\}$ with $2 \leq n < m < \infty$ and $n = 2^q$ for $q \in \mathbb{Z}_+$. The probability that Ψ contains an orthonormal basis $\{e_i\}_{i=1}^n$ can be expressed with the following inequality,

$$P\left(\Psi \in SC(n, m)\right) \geq 1 - \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{2^n}\right)^i \left(\frac{2^n - (n-i)}{2^n - 1}\right)^{m-i}.$$

Proof. For this proof, it will suffice to make the connection to Lemma 5.2.3 and Theorem 5.2.4. Recall that if a frame contains an ONB it is scalable. For Rademacher matrices with columns of size $n = 2^q$, there always exists an ONB (or Hadamard matrix). Using this basis in place of the identity in Lemma 5.2.3 and Theorem 5.2.4, and also noticing that there are 2^n elements in the set of unimodular vectors (instead of $2^n - 1$), we have our result. ■

We shall perform a similar analysis when the elements are sampled without replacement.

Lemma 5.3.4. Let Ψ be a random subset of m elements sampled i.i.d.. Also let this random subset be sampled without replacement from the set of all possible Rademacher vectors in $\mathbb{R}^n \backslash \{\mathbf{0}\}$ with $2 \leq n < m \leq 2^q$ and $n = 2^q$ for $q \in \mathbb{Z}_+$. The probability that Ψ contains an orthonormal basis $\{e_i\}_{i=1}^n$ can be expressed with the following inequality,

$$P\left(\Psi \in SC(n,m)\right) \geq 1 - \sum_{i=0}^{n-1} \binom{n}{i} \frac{\binom{2^n-n}{m-i}}{\binom{2^n}{m-i}} \frac{1}{\binom{2^n-(m-i)}{i}}.$$

Proof. For this proof, it will suffice to make the connection to Lemma 5.2.5 and Theorem 5.2.6. Recall that if a frame contains an ONB it is scalable. For Rademacher matrices with columns of size $n = 2^q$, there always exists an ONB (or Hadamard matrix). Using this basis in place of the identity in Lemma 5.2.5 and Theorem 5.2.6, and also noticing that there are 2^n elements in the set of unimodular vectors (instead of $2^n - 1$), we have our result. ■

In the case of Rademacher frames, while we have similar results for Bernoulli, these bounds are actually less tight. In the analysis of Bernoulli frames, we are implicitly using the property that there exists only one ONB, and this property is not necessarily true for Rademacher frames of any dimension,

$$\left\{ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \right\}. \tag{5.20}$$

5.4 Symmetric Distributions

For the analysis of continuous distributions, we need an alternative tool, as the probability of encountering an ONB is zero. What we shall use instead is Farkas'

Lemma and separating hyperplanes to bound the probability of encountering a frame that is not scalable, and from this, bound the probability that a frame is scalable. These results hold for general symmetric distributions, but we focus on Gaussian and Uniform frames. We conclude the section with a Theorem of Alternatives for frames drawn from Gaussian, Uniform, and Rademacher distributions. There are a number of technical lemmas that we first require, but the theme shall proceed as follows:

1. Alternatives Theorem for scalability
2. Frames drawn from symmetric distributions are symmetric in frame transform domain
3. Alternatives results for Gaussian, Uniform, and Rademacher frames
4. Independence in the frame transform domain
5. Bounds for frames drawn from symmetric distributions

Lemma 5.4.1 (Farkas Lemma with Scalability)**.** Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame, and let $F(\Phi)$ be the associated frame transform of dimensions $d \times m$ where $d = \frac{(n-1)(n+2)}{2}$. Then, exactly one of the following two alternatives hold:

(a) There exists some $u \geq \mathbf{0}$ such that $\begin{bmatrix} F(\Phi) \\ \mathbb{1}^T \end{bmatrix} u = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$.

(b) There exists some vector $w \in \mathbb{R}^{d+1}$ such that $w^T \begin{bmatrix} F(\Phi) \\ \mathbb{1}^T \end{bmatrix} \geq \mathbf{0}^T$ and $w^T \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} < 0$.

Moreover, if alternative (a) is true, then the frame Φ is scalable.

Proof. The first statement follows directly from the proof of Farkas' Lemma given in (4.1.3), and scalability follows directly from the equivalence of Corollary (2.4.6). ∎

Lemma 5.4.2. Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a frame, and let $F(\Phi)$ be the associated frame transform of dimensions $d \times m$ where $d = \frac{(n-1)(n+2)}{2}$. Then if Φ is not scalable, there exists a vector $w \in \mathbb{R}^{d+1}$ with $w^T \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} < 0$, and furthermore, $w_{d+1} < 0$.

Proof. This follows from the fact that the only non-zero term in $\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$ is the last element. And for the product $(1 \cdot w_{d+1})$ to be negative, w_{d+1} must be negative itself. ∎

Theorem 5.4.3 (Hyperplane Separation for Scalable Frames)**.** Let $\Phi = \{\varphi_k\}_{k=1}^m$ be frame for $\mathbb{R}^n$. Then, exactly one of the following two alternatives hold:

(a) $\Phi \in \mathcal{S}F(n, m)$.

(b) There exists a halfspace, defined by a normal vector $w \in \mathbb{R}^{d+1}$, that contains all of the transformed frame vectors,

$$\{\langle F(\varphi_k), \widetilde{w} \rangle \geq -w_{d+1}\}_{k=1}^m. \tag{5.21}$$

Proof. Combining Lemma 5.4.1 and Lemma 5.4.2, we have the alternatives result. We can normalize the length of the vectors $\{F(\varphi_k)\}_{k=1}^m$ to lie on the unit hypersphere. If the transformed frame vectors have a maximum angle of $< \pi$ radians, no weighted non-negative sum of $\{F(\varphi_k)\}_{k=1}^m$ will equal $\mathbf{0}$. Furthermore, there exists a halfspace that contains $\{F(\varphi_k)\}_{k=1}^m$. ■

Corollary 5.4.4 (Hyperplane Separation for Rademacher Frames)**.** Let $\Phi = \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$ be a Rademacher frame. Then, exactly one of the following two alternatives hold:

(a) $\Phi \in \mathcal{S}F(n,m)$.

(b) There exists a halfspace, defined by a normal vector $w \in \mathbb{R}^{d+1}$ with $\{w_i = 0\}_{i=1}^{n-1}$, that contains all of the transformed frame vectors,

$$\{\langle F(\varphi_k), \widetilde{w}\rangle \geq -w_{d+1}\}_{k=1}^m. \tag{5.22}$$

Proof. For the added result $\{w_i = 0\}_{i=1}^{n-1}$, we employ Lemma 5.3.1, which states that,

$$F_0(\varphi) = \mathbf{0}. \tag{5.23}$$

■

We shall now present some probabilistic results for symmetric distributions. These results stem from a quasi-independence for the transformed frame elements $F(\Phi)$.

Lemma 5.4.5 (Frame Transform Symmetry)**.** Let $\Phi = \{\varphi_k\}_{k=1}^m$ be a frame in $\mathbb{R}^n$, and let φ_k be a frame element drawn i.i.d. from a symmetric distribution $\Theta^n \subseteq \mathbb{R}^n$. Then the entries of the frame transform, $F(\varphi_k)$, have entries drawn from a symmetric distribution (the expected value is 0).

Proof. To prove the result, we shall handle this in the two cases. For the difference of square terms, we can write the random variable interpretation as,

$$F_{0,j}(\varphi) = X_1^2 - X_j^2\,, \text{ where } X_1, X_j \sim \Theta. \tag{5.24}$$

Now consider the expected value,

$$E[F_{0,j}(\varphi)] = E[X_1^2 - X_j^2] = E[X_1^2] - E[X_j^2] = \mu_{X_1^2} - \mu_{X_j^2} = 0. \tag{5.25}$$

For the cross products, we can write the random variable interpretation as,

$$F_{i,j}(\varphi) = X_i X_j\,, \text{ where } X_i, X_j \sim \Theta. \tag{5.26}$$

Now consider the expected value (using independence),

$$E[F_{i,j}(\varphi)] = E[X_i] \cdot E[X_j] = 0. \tag{5.27}$$

■

Lemma 5.4.6 (Frame Transform Independence)**.** Let $\Phi = \{\varphi_k\}_{k=1}^m$ be a from in $\mathbb{R}^n$, and let φk be a frame element drawn i.i.d. from a symmetric distribution $\Theta^n \subseteq \mathbb{R}^n$. Then the transformed frame vector entries $\{F_i(\varphi_k)\}_{k=1}^m$ and $F_{0,1}(\varphi)$ are pairwise independent in that,

$$P(F_{i,j}(\varphi_k) > 0 \,|\, F_{i',j'}(\varphi_k) > 0) = P(F_{i,j}(\varphi_k) > 0). \tag{5.28}$$

The first subscript, i, denotes the transformation $F_i(\cdot)$, and the second index, j, denotes the entry in that vector.

Proof. In general, we choose $F_{0,1}(\cdot)$ for convenience. We do not have independence for all of the entries of $F_0(\varphi)$ (as we show in the proof), but we are able to use a single entry, and we choose the first, $F_{0,1}(\cdot)$.

We separate this proof into several cases. Define the following random variables,

$$
\begin{aligned}
x_i &:= X_1^2 - X_i^2 = \varphi(1)^2 - \varphi(i)^2, \\
y_{ij} &:= X_i X_j = \varphi(i)\varphi(j).
\end{aligned}
\tag{5.29}
$$

This simplifies to proving independence in six cases:

(a) $P\left(x_i > 0 \,|\, x_{i'} > 0\right) \neq P\left(x_i > 0\right)$

(b) $P\left(x_i > 0 \,|\, y_{1i} > 0\right) = P\left(x_i > 0\right) = 1/2$

(c) $P\left(x_i > 0 \,|\, y_{ij} > 0\right) = P\left(x_i > 0\right) = 1/2$

(d) $P\left(y_{ij} > 0 \,|\, y_{ij'} > 0\right) = P\left(y_{ij} > 0\right) = 1/2$

(e) $P\left(y_{ij} > 0 \,|\, y_{i'j'} > 0\right) = P\left(y_{ij} > 0\right) = 1/2$

(f) $P\left(x_i > 0 \,|\, y_{i'j} > 0\right) = P\left(x_i > 0\right) = 1/2$

Case (a):

Consider the sample space with $X_1^2 > X_i^2$ equivalent to $|X_1| > |X_i|$. Order the variables in decreasing order $(X_1, X_i, X_{i'}) \to X_1 > X_i > X_{i'}$, and the sample space is,

$$
\mathcal{S} = \{(X_1, X_i, X_{i'}), (X_1, X_{i'}, X_i), (X_i, X_1, X_{i'}), (X_i, X_{i'}, X_1), (X_{i'}, X_1, X_i), (X_{i'}, X_i, X_1)\}.
\tag{5.30}
$$

Computing the conditional probability, we have,

$$
P\left(x_i > 0 \,|\, x_{i'} > 0\right) = \frac{P\left(x_i > 0 \,\cap\, x_{i'} > 0\right)}{P\left(x_{i'} > 0\right)} = \frac{1}{3}.
\tag{5.31}
$$

Case (b):

We need the probability that $x_i > 0$ (or the probability that $|X_1| > |X_i|$) to be independent of the sign of X_1 and X_i, but this is trivially true, as we lose any information about the sign on X_1 and X_i by taking the absolute value.

Case (c):

We need the probability that $x_i > 0$ (or the probability that $|X_1| > |X_i|$) to be independent of the sign of X_i and X_j, but this is also trivially true, as we lose any information about the sign on X_i by taking the absolute value, and X_j is independent of $|X_1|$ and $|X_i|$.

Case (d):

$$\begin{aligned} P\left(y_{ij} > 0 \,|\, y_{ij'} > 0\right) &= \frac{P\left(y_{ij} > 0 \cap y_{ij'} > 0\right)}{P\left(y_{ij'} > 0\right)} \\ &= \frac{P\left(y_{ij} > 0 \cap y_{ij'} > 0\right)}{\frac{1}{2}}. \end{aligned} \tag{5.32}$$

Write the sample space denoting the sign outcomes ± for the variables $(X_i, X_j, X_{j'})$,

$$\mathcal{S}_{(i,j,j')} = \{(+++), (++-), (+-+), (-++), (+--), (-+-), (--+), (---)\}. \tag{5.33}$$

The only two cases, of the eight total, that result in both products being positive is $(+++)$ and $(---)$.

$$\begin{aligned} P\left(y_{ij} > 0 \,|\, y_{ij'} > 0\right) &= \frac{P\left(y_{ij} > 0 \cap y_{ij'} > 0\right)}{\frac{1}{2}} \\ &= \frac{\frac{2}{8}}{\frac{1}{2}} \\ &= \frac{1}{2} \\ &= P\left(y_{ij} > 0\right). \end{aligned} \tag{5.34}$$

Case (e) and (f):

For these cases it is sufficient to notice that the events have no variables in common, and so are independent. ■

Theorem 5.4.7 (Probability Bound on Scalability)**.** Let $\Psi = \{\varphi_k\}_{k=1}^m$ be frame where the entries are drawn from the symmetric distribution $\Theta \subseteq \mathbb{R}$. Also let the probability that there exists a vector $\widetilde{w}$ such that,

$$\{\langle F(\varphi_k), \widetilde{w}\rangle \geq 0\}_{k=1}^m, \tag{5.35}$$

be denoted with a function of the dimension a_n. Then the probability that Ψ is scalable can be bounded below with the inequality,

$$P(\Psi \in \mathcal{S}F(n,m)) = 1 - P(\Psi \notin \mathcal{S}F(n,m)) \geq 1 - a_n^m. \tag{5.36}$$

Proof. We prove this result by deriving the lower bound of $P(\Psi \notin \mathcal{S}F(n,m))$. Using Theorem 5.4.3, we have

$$P(\Psi \notin \mathcal{S}F(n,m)) = P(\widetilde{w}^T F(\Psi) \geq \mathbf{0}). \tag{5.37}$$

The transformed frame elements $F(\varphi_k)$ are independent of each other, so we can simplify the probability,

$$P(\Psi \notin \mathcal{S}F(n,m)) = P(\widetilde{w}^T F(\Psi) \geq -w_{d+1}\mathbb{1}^T) \leq [P(\langle \widetilde{w}, \varphi_k\rangle \geq \mathbf{0})]^m. \tag{5.38}$$

The probability only depends on a function of n, that we shall denote a_n,

$$[P(\langle \widetilde{w}, \varphi_k\rangle \geq \mathbf{0})]^m = a_n^m. \tag{5.39}$$

Finally, taking the complement of this upper bound, we have the lower bound for scalability. ■

With this result, we see that the probability that a frame is scalable increases to 1, as m increases. What remains vague, is the function a_n. This is partially due to the lack of independence as shown in Lemma 5.4.6.

5.5 Numerical Results

In this section we present some numerical computations of the distribution of scalable frames when drawn from the above densities. For the results presented, we consider Bernoulli, Rademacher, Uniform, and Gaussian frames. We consider frames in the dimensions $n \in \{2, 3, 4, 5, 6, 7, 8, 9\}$, with the number of frame elements varying with the dimension. We first consider frames with lower redundancy $m \in \{n+1, 2n+1, 3n+1\}$, and then consider high-redundancy frames $m = \{n^2+n+1, 2n^2+n+1, 3n^2+n+1, 4n^2+n+1, 5n^2+n+1\}$. For any tests of the frame properties, we perform 500 trials (a frame of the specified size is generated 500 times). See Figures 5.1 and 5.2.

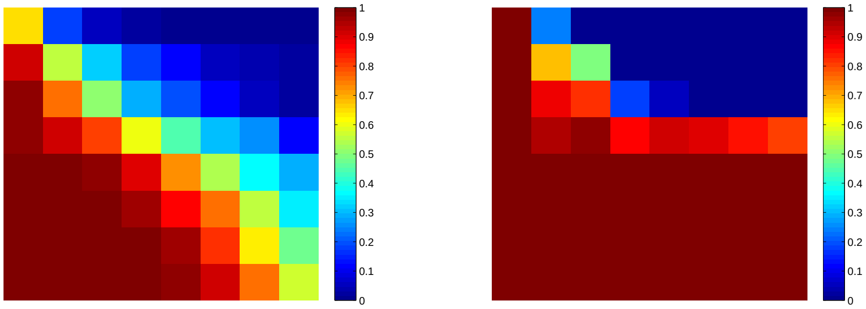

Figure 5.1: The probability maps of Bernoulli (Left) and Rademacher (Right) frames. Over 500 trials, the percentage of scalable frames tabulated. The x-axis varies the frame dimension n, and the y-axis varies the number of frame elements m.

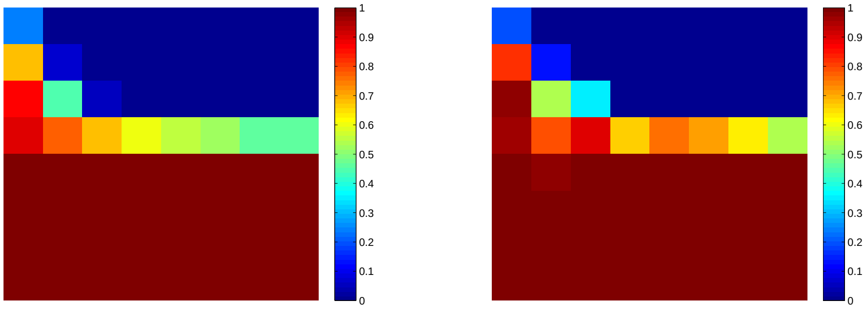

Figure 5.2: The probability maps of Gaussian (Left) and Uniform (Right) frames. Over 500 trials, the percentage of scalable frames tabulated. The x-axis varies the frame dimension n, and the y-axis varies the number of frame elements m.

Chapter 6: Learning Graph Structure

In this chapter we present some select applications of the scalable frames model. We give a connection to graph partitioning, scaling, and sparsification. The topic of scalable frames, attempts to find a set of weights that control the spectrum of the frame. This is accomplished by giving small weights to frame elements with lots of redundancy, and larger weights to singular frame elements. In the previous topics, we were intentionally vague about what our frame elements represented, but in this section we focus specifically on the scaling of graphs. If we view an edge representation of a graph as a frame element, we can then apply the scaling techniques described above. It turns out that there is a natural interpretation of the scaling weights as importance weightings on the structure of the graph. Depending on the input graph, the scaling attempts to produce complete sub-graphs.

The motivation for this scheme comes from the need to process large graphs efficiently. This has ties with many graph analysis techniques (i.e. min/max-cut [41], community detection [40, 58], clustering [81, 92], etc. [41, 80, 81]), and in general, these graphs are extremely large, and extremely sparse. In real-life examples, though, the graph can be more complex (and provides more detail) than is actually required in an analysis. The underlying connection in all of these applications, is the desired partitioning of a given network into meaningful constituent parts. This chapter shall make the case that a spectral approach can efficiently process large graphs without explicitly computing the spectrum.

6.1 Graph Background and Notation

A little notation and background is presented here, but much of this material is taken from [28, 92]. Denote a *graph* by $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{\nu_1, \nu_2, \ldots, \nu_n\}$ is a set of vertices on the graph. $\mathcal{E} := \{e_1, e_2, \ldots, e_m\}$ is the ordered set of edge pairs that denotes a connection between two nodes. A weight $0 \leq \omega_{ij} \leq 1$ denotes the similarity between two nodes (ν_i, ν_j), and $\omega_{ij} = 0$ if the nodes are not connected. We shall consider only undirected graphs in this work, implying $\omega_{ij} = \omega_{ji}$. The matrix of these weights is referred to as the adjacency matrix W, and we denote the degree of a node, ν_i, as

$$d_i := \sum_{j=1}^{n} \omega_{ij}. \tag{6.1}$$

The *degree matrix* D is then a diagonal matrix with entries $D_{ii} = d_i$ for $i = 1, \ldots, n$.

We now define the *Laplacian matrix* on the graph as

$$L_{\mathcal{G}} := L = D - W. \tag{6.2}$$

An alternate formulation of the Laplacian uses the *incidence matrix* $B = [b_1, b_2, \ldots, b_m]$, defined as an $n \times m$ matrix where every column in B represents an edge (ν_i, ν_j) in

$\mathcal{E}$. For a column in the incidence matrix, b_k, we have

$$b_k(i) := \begin{cases} \sqrt{\omega_{ij}} & : (\nu_i, \nu_j) \in \mathcal{E}\,,\ i < j \\ -\sqrt{\omega_{ij}} & : (\nu_i, \nu_j) \in \mathcal{E}\,,\ i > j \\ 0 & : \text{else} \end{cases} . \tag{6.3}$$

The Laplacian can now be defined as

$$L := BB^T. \tag{6.4}$$

Finally, we define a common graph type we will consider, namely *complete graphs*, where all nodes are connected to each other node $(\nu_i, \nu_j) \in \mathcal{E}$ for all $i \neq j$.

6.2 Graph Conditioning

We now turn out attention to defining what is meant by graph conditioning (or graph scaling). Considering our definition of frame conditioning (weights that result in a frame with constant spectra), we shall define graph conditioning as determining weights that result in a graph with constant spectra. With this, we must make clear which graph representation we are using. Instead of using the adjacency or Laplacian, we shall take the incidence matrix, as it is the most frame-like. The incidence matrix multiplied by its transpose results in the Laplacian matrix as the frame operator analog, and analysing the spectral properties of L is usually a goal in spectral methods (as we generally desire with frame methods). It should be noted that the incidence matrix of a graph is not full rank, and thus, is not scalable (in the usual sense). Scalability also generally requires order n^2 frame elements, and a graph has at most $\frac{n(n-1)}{2}$ edges (the complete graph) excluding self-edges. To get around this issue, we turn to the definition of the *graph condition number.* The graph condition number is defined as the ratio of the largest and smallest non-zero eigenvalues of the Laplacian,

$$\kappa(L_{\mathcal{G}}) := \frac{\lambda_1}{\lambda_r} \text{ for } \lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_n = 0. \tag{6.5}$$

Where eigenvalues of 0 can lead to numerically unstable solutions for linear systems, eigenvalues of 0 in the graph setting disconnect the graph, where this can be a desired property in the case of partitioning and clustering. This function is simply the condition number of the Laplacian (ignoring the zero eigenvalues), so scaling in this context, leads to the interpretation that well-conditioned graphs as sets of complete sub-graphs. With this in mind, to condition (or scale) a graph, means to scale edges to create these complete sub-graphs. This encourages the use of the incidence matrix B as the frame Φ, which in turn leads to the Laplacian L as the frame operator S. We can now formulate the graph conditioning problem for the incidence matrix.

Definition 6.2.1. Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ be a graph with incidence matrix B, and Laplacian matrix $L = BB^T$. Then B is scalable if there exists a non-negative, non-zero diagonal matrix X, such that the graph condition number $\kappa(\widetilde{\mathcal{L}})$ of the scaled Laplacian $\widetilde{L} = BX^2B^T$ is equal to 1,

$$\widetilde{\lambda}_1 = \cdots = \widetilde{\lambda}_r > \widetilde{\lambda}_{r+1} = \ldots \lambda_n = 0. \tag{6.6}$$

With the connection made, it may seem tempting to phrase the problem as we have for standard frames, but this is not possible for graphs. The standard equation we wish to solve, finds an X, such that

$$\Phi X^2 \Phi^T = I \rightarrow BX^2B^T = I, \tag{6.7}$$

but in the case of graphs, there will never exist a scaling matrix X that satisfies this equation. A simple argument for this, is to notice that a frame must be full rank (and this is clear by noting that we scale to make all eigenvalues of the frame operator/Laplacian equal 1), and graphs by design are low rank. The scalability problem is ill-posed here because we have not considered fully the expanded definition of the condition number of a graph.

Theorem 6.2.2. Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ be a complete graph with incidence matrix B, and Laplacian matrix $L = BB^T$. Then there exists no scaling X such that,

$$BX^2B^T = I. \tag{6.8}$$

Proof. Scaling an incidence matrix places weights on the edges of the graph, and requiring the scaling to result in an identity Laplacian, is equivalent to requiring the scaled graph to have only self-edges, and this is impossible. ■

There is, though, another way to condition our graph without resulting in the identity matrix. If our ultimate goal is to perfectly condition a graph, then we desire our scaling to result in a *degree-balanced complete graph* (complete graph with all nodes equal degree). This changes the desired Laplacian from the identity to $L = nI - \mathbb{1}\mathbb{1}^T$, and this problem can be solved. We now redefine graph scalability.

Definition 6.2.3 (Graph Conditioning)**.** Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ be a complete graph with incidence matrix B, and Laplacian $L = BB^T$. Then B is scalable if there exists a non-negative, non-zero diagonal matrix X, such that the graph condition number $\kappa(\widetilde{\mathcal{L}})$ of the scaled Laplacian $\widetilde{L} = BX^2B^T$ is equal to 1. In other words, the following system of equations hold,

$$\widetilde{L} = BX^2B^T = nI - \mathbb{1}\mathbb{1}^T. \tag{6.9}$$

To make this more concrete, we present a few small examples to motivate our solution.

6.3 Motivating Examples

Take the complete graph $\mathcal{G}_1$ with n nodes and $m = (n-1)^2$ edges. Furthermore, let n be even. From [28] we have that all of the non-zero eigenvalues are n, resulting in a graph that has optimal condition number,

$$\kappa(L_{\mathcal{G}_1}) = \frac{n}{n} = 1. \tag{6.10}$$

This is not the only way to optimize the graph condition number (as we have defined above). If we randomly partition the nodes into groups $\mathcal{A}$ and $\mathcal{B}$ where $|\mathcal{A}| = \frac{n}{2}$ and $|\mathcal{B}| = \frac{n}{2}$, remove all edges that cross from one set to the other, there are now two disconnected complete graphs, and the condition number is still optimal,

$$\kappa(L_{\mathcal{G}_1}) = \frac{n/2}{n/2} = 1. \tag{6.11}$$

We can perform a partitioning scheme similar to this until we have disconnected pairs of connected nodes, but the condition number will still be optimal. This leads to the conclusion that we desire some notion of "fullness" in this conditioning. In other words, we will require that the degree of each node must be at least some value chosen a priori. Relating this to min/max-cut problems, in [92] the author considers the problem of partitioning a graph in a balanced way, which leads to a ratio cut that attempts to balance the total degree of the partitions generated. In the same way, we are attempting to condition a graph, while keeping balance when disconnections occur.

Now consider a graph $\mathcal{G}_2$ that consists of two distinct complete subgraphs of equal size connected by a single edge. We can optimally condition this by removing the connecting edge, but what should happen to the displaced weight is another issue. The naive approach would be to simply "throw away" the excess weight, but this seems unsatisfactory, as the degree of two nodes has been decreased and may not satisfy our degree requirements. To account for this we shall further impose the constraint that the degree of the nodes remains constant. So in the example considered here, after the connecting edge is removed, all of the remaining edges would need to be increased by a, where $\sum_{j=1}^{m} aw_{ij} = d_i$ is the number of edges.

6.4 Problem Formulation

As discussed in Chapter 3, we can cast the problem of scaling a frame as finding a non-negative solution to the system formed by the frame transform T,

$$\begin{aligned} T(B)u &= \begin{bmatrix} \mathbb{1} \\ \mathbf{0} \end{bmatrix}, \\ \mathbb{1}^T u &\geq 1, \\ u &\geq \mathbf{0}. \end{aligned} \tag{6.12}$$

We have to make an adjustment to the right-hand-side to account for our updated definition of scalability,

$$\begin{aligned} T(B)u &= \begin{bmatrix} d_{11} \\ d_{22} \\ \vdots \\ d_{nn} \\ -\mathbb{1} \end{bmatrix}, \\ \mathbb{1}^T u &\geq 1, \\ u &\geq \mathbf{0}, \end{aligned} \tag{6.13}$$

with $d_{ii} = n - 1$. We can also form an equivalent version using the reduced frame transform F,

$$\begin{aligned} F(B)u &= \begin{bmatrix} \mathbf{0} \\ \\ -\mathbb{1} \end{bmatrix}, \\ \mathbb{1}^T u &\geq 1, \\ u &\geq \mathbf{0}, \end{aligned} \tag{6.14}$$

where we have the difference of the degree terms resulting in 0.

Proposition 6.4.1. Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ be a complete graph with incidence matrix B, and Laplacian matrix $L = BB^T$. Then B is scalable with scaling weights $x_k = \dfrac{1}{\sqrt{w_{ij}}}$, such that,

$$\widetilde{L}_{\mathcal{G}} = BX^2B^T = nI - \mathbb{1}\mathbb{1}^T, \tag{6.15}$$

and $\kappa(\widetilde{L}_{\mathcal{G}}) = 1$.

Proof. The graph is complete, and if we scale all of the edges to have weight 1, the degree of each node will be equal to $n-1$, and the resulting graph will be complete. As the complete graph has Laplacian eigenvalues $\lambda_i = n$ for $i = 1, \dots, n-1$, the graph has condition number 1. ■

As every complete graph is trivially scalable, we move on to more complicated graphs, namely, non-complete graphs. Keeping in mind that complete graphs are optimally conditioned (in the graph setting), we can simplify (6.14). Noticing that the off-diagonal elements $F_i(\cdot)$ are unnecessary, we can remove the constraints from the system (as the off-diagonal elements are set by the scaling weights given for the diagonal equations $F_0(\cdot)$),

$$\begin{aligned} F_0(B)u &= \mathbf{0}, \\ \mathbb{1}^T u &\geq 1, \\ u &\geq \mathbf{0}. \end{aligned} \tag{6.16}$$

Using the techniques developed in Chapter 4 we shall phrase this as a linear optimization problem,

$$\begin{aligned} \text{minimize: } & \langle a, u \rangle \\ \text{subject to: } & F_0(B)u = \mathbf{0}, \\ & \mathbb{1}^T u \geq 1, \\ & u \geq \mathbf{0}. \end{aligned} \tag{6.17}$$

For graphs that are not complete, there is, in general, no scaling that results in a perfectly conditioned graph, but our goal isn't necessarily to condition the graph, but to learn the importance of edges, and the formulation above allows us to do that. Consider again the second graph in the Section 6.3. We have two complete graphs connected by a single edge between them. We shall assume the graph is unweighted, and now present the results from the conditioning (Figure 6.1 and Table 6.4). The edge corresponding to scaling coefficient with the smallest value will be removed from the graph, and the spectrum will be considered.

Consider another example, where we take the complete graph with n nodes, and remove $n-2$ edges from the first vertex. We present the results in Table 6.4, but it should be noted that the scaling achieved, removed the edges ($x_k = 0$) connected to the second node. The original graph and resulting graph are shown in Figure 6.2.

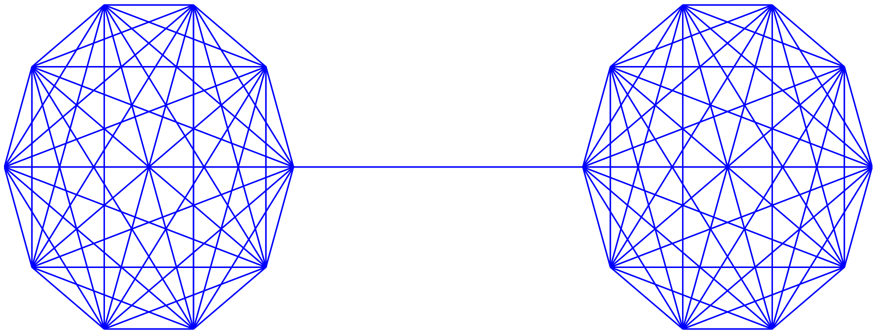

Figure 6.1: The Two-Complete Graphs with 10 nodes each.

Two-Complete Graphs Spectrum				
	original	g1	g2	g3
$\sigma_{\max}$	3.4396	3.4396	3.3977	3.1623
$\sigma_{\min}$	0.4112	0.4112	0.4089	3.1623
$\kappa(L_{\mathcal{G}})$	8.3648	8.3648	8.3094	1.0000
x_k	-	0.0101	0.0112	0.0090

The singular values of the incidence matrix after the removal of edges. The original column contains the max/min singular values of the starting graph. The “g1” and “g2” columns contain the max/min singular values after an edge from one of the complete graphs is removed. The “g3” column contains the max/min singular values after the edge between the two graphs is removed. The x_k row contains the unique scaling weights.

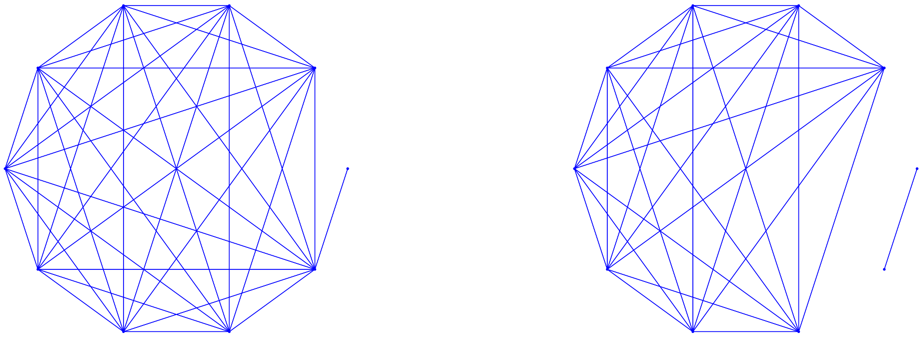

Figure 6.2: The outlier node complete graph (Left), and the scaled graph (Right), resulting in two complete graphs.

Outlier Complete Graph Spectrum

	original	g1	g2	g3
$\sigma_{\max}$	3.1623	3.0000	3.1623	3.0927
$\sigma_{\min}$	1.0000	3.0000	1.0000	0.9909
$\kappa(L_{\mathcal{G}})$	10.000	1.0000	10.000	9.7412
x_k	-	0.2000	0.0286	0.0000

The singular values of the incidence matrix after the removal of edges. The original column contains the max/min singular values of the starting graph. The "g1," "g2," and "g3," columns contain the max/min singular values after an edge from one of the complete graphs is removed. It should be noted that while removing the edge from the outlier results in a better condition number, it also results in a single node graph. The x_k row contains the unique scaling weights.

Chapter 7: Learning Linear Structure

7.1 Overview

In this chapter we turn our focus to learning the linear structure of datasets. We do this by employing a robust version of PCA, and then applying this method to the field of Electron Energy-loss Spectroscopy (EELS). We give a brief introduction and motivation of robust PCA methods, and then give an overview of the electron imaging technique. After the notation and definitions are introduced, we present a novel scheme designed take advantage of the linear structure of the images produced by the electron imaging. Finally, we show some experimental results on artificial and imaged biological samples.

7.2 Robust PCA Background

A notable pitfall of employing PCA as a learning technique, is the requirement that the underlying manifold be linear in nature, but a far more subtle, and equally important, difficulty is the failure of PCA in the presence of noise (as noted in [51]). In the spectral sense, we can see this sensitivity with the following example. Take the dataset $\Phi = \{\varphi_k\}_{k=1}^{100} \subset \mathbb{R}^{100}$ where $\varphi_k = k\mathbb{1}$. The matrix representation is

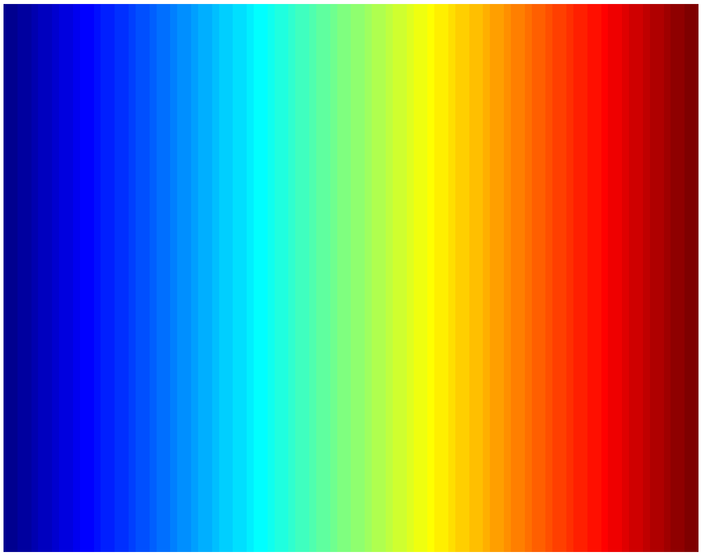

Figure 7.1: The original rank-1 dataset Φ.

visualized in Figure 7.1. Φ truly lies on a linear manifold, as the dataset is rank 1 (every element φ_k is generated by scales of $\mathbb{1}$). If we add noise with a uniform random pattern by biasing 15% of the elements by 50, we obtain a noise pattern, and a corrupted dataset, which is presented in Figure 7.2. Denote the noise pattern and the corrupted dataset by $E = \{\epsilon_k\}_{k=1}^{100} \subset \mathbb{R}^{100}$ and $\widetilde{\Phi} = \{\varphi_k + \epsilon_k\}_{k=1}^{100} \subset \mathbb{R}^{100}$, respectively. Performing a spectral analysis of this corrupted datset, we see that the spectrum decays by several orders-of-magnitude after the first singular value. The case can be made that the "true" dimensionality is 1, but if we take the top rank 1 component, a sub-optimal reconstruction is obtained (Figure 7.2). If we look at the spectrum of the individual components (true dataset Φ and the corruption mask E) we see that the spectrum and eigenvector pattern are uncorrelated (Figure 7.3), and this is an important fact.

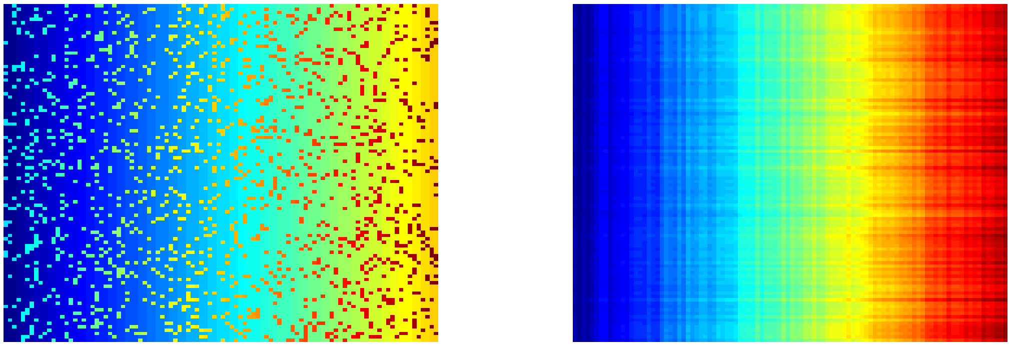

Figure 7.2: The corrupted dataset $\widetilde{\Phi}$ with the unknown noise component E biasing 15% of the elements in Φ (Left), and the reconstructed dataset $\widehat{\Phi}$, taking the largest singular value (Right).

Presented in [16, 17], there is a mathematically rigorous way to separate these components, and it uses the lack of correlation between the respective spectra. Introduced as Robust Principal Component Analysis (RPCA), we can separate uncorrelated spectra exactly. The result requires some machinery before its full statement.

Definition 7.2.1 (Incoherence Condition η)**.** Let $\Phi = \{\varphi_k\}_{k=1}^n$ be represented as an $n \times n$ matrix of rank r, resulting in the SVD components $U = [u_1, \ldots, u_r]$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$, and $V = [v_1, \ldots, v_r]$. Φ is said to satisfy *Incoherence Property* η, if the following inequalities hold:

$$\max_k \|U^T e_k\|_2^2 \leq \frac{r\eta}{n},$$

$$\max_k \|V^T e_k\|_2^2 \leq \frac{r\eta}{n},$$

$$\|UV^T\|_\infty^2 \leq \frac{\sqrt{\eta r}}{n}.$$

The following result is Theorem 1.1 in [17].

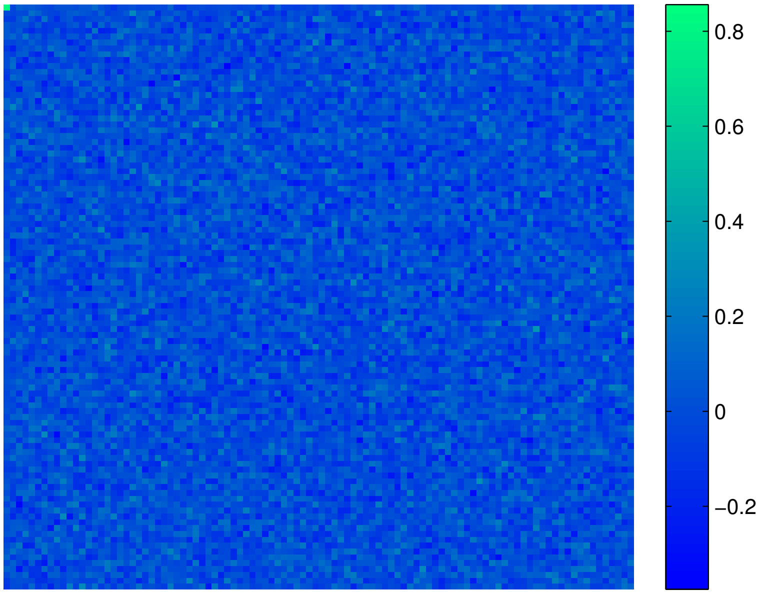

Figure 7.3: The correlation matrix of the eigenvectors of Φ and E. The vectors are all uncorrelated except for the two leading vectors. This is due to the biasing from the increasing elements in Φ and the non-negative elements of E. This is the reason for the inability of the leading component to accurately reconstruct the rank 1 component. The average correlation is -.0012.

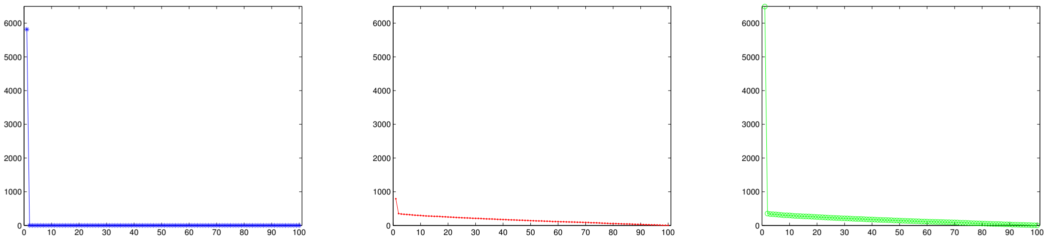

Figure 7.4: The singular value decay of the rank-1 dataset (Left). And the singular value decay of the noise mask (Middle). The singular value decay of the noise-corrupted dataset (Right). These plots are all presented on the same scale.

Theorem 7.2.2. Suppose $\Phi = \{\varphi_k\}_{k=1}^n$ is represented as an $n \times n$ matrix and satisfies the incoherence condition with parameter η, and let the support of a *noise matrix*, E, have uniformly distributed support of cardinality $|E| = s$. Take $\gamma = \frac{1}{\sqrt{n}}$, $s \leq \rho_1 n^2$, and

$$\text{rank}(\Phi) \leq \frac{\rho_2 n}{\eta(\ln(n))^2},$$

with ρ_1 and ρ_2 being positive numerical constants. Then, there is a numerical constant a such that, with probability at least $1 - an^{-10}$, the corrupted dataset $\widetilde{\Phi}$ can be decomposed into the true low-rank and noise components, Φ and E, respectively, by solving,

$$\begin{aligned} &\text{minimize: } \|\Phi\|_* + \gamma\|E\|_1, \\ &\text{subject to: } \Phi + E = \widetilde{\Phi}. \end{aligned} \tag{7.1}$$

Theorem 7.2.2 is related to many of the notions from Compressive Sensing [3, 18, 19]. The optimization problem being solved is the convex relaxation of the NP-hard problem,

$$\begin{aligned} &\text{minimize: } \text{rank}(\Phi) + \gamma\|E\|_0, \\ &\text{subject to: } \Phi + E = \widetilde{\Phi}. \end{aligned}$$

Here, the ℓ_0 "norm" ($\|E\|_0$) is the number of non-zero values in E. The rank(Φ) term represents the effective number of principal components in the corrupted dataset $\widetilde{\Phi}$. The incoherence property ensures that the spectral properties of Φ don't overlap with the those of E, and in this case, we can solve an NP-hard problem with high probability.

Several methods have been proposed to solve the separation problem of (7.1) [64, 96]. The class of algorithms we consider here is described in [64] and utilizes

Augmented Lagrange Multipliers (ALM). For more information on general ALM formulation see [34, 65, 98]. ALM is employed here mainly for its guarantee of convergence to an optimal solution under mild conditions, and we state the algorithm below for convenience.

Definition 7.2.3 (Spatial Shrinkage Operator)**.** We denote by $\mathcal{S}_\gamma$ the *Spatial Shrinkage Operator*, which performs a soft thresholding on a given $n \times m$ matrix by subtracting positive constant, γ, from each element and thresholding all negative values to 0:

$$\mathcal{S}_\gamma[A] = \max\{A - \gamma \mathbb{1}\mathbb{1}^T \, , \, \mathbf{0}\},$$

where we use the entry-wise max function.

Definition 7.2.4 (Spectral Shrinkage Operator)**.** We denote by $\widehat{\mathcal{S}}_\mu$ the *Spectral Shrinkage Operator*, which performs a soft thresholding on a given $n \times m$ matrix by subtracting positive constant, μ, from each singular value and thresholding all negative values to 0:

$$\widehat{\mathcal{S}}_\mu[A] = U \cdot \max\{\Sigma - \mu I \, , \, \mathbf{0}\} \cdot V^T,$$

where $A = U\Sigma V^T$.

Presented as Algorithm 4 in [64], we now present this scheme as Algorithm (7.2).

The authors in [16, 17] are not the first to consider extensions of PCA. Due to PCA's widely known sensitivity to outliers, a number of techniques were developed under the umbrella of robust PCA. This generally refers to the ability of a technique

Algorithm 3 ALM for Robust Principal Component Analysis

```
Require: $\Phi\,,\,\gamma\,,\,\mu_0\,,\,\rho$
J = $\max\{\|\widehat{\Phi}\|_2\,,\,\gamma^{-1}\|\widehat{\Phi}\|_\infty\}$
$Y_0 = \dfrac{\text{sgn}(\widehat{\Phi})}{J}$
$i \leftarrow 0$
while not converged do
    $\Phi^0_{i+1} \leftarrow \Phi^*_i$
    $E^0_{i+1} \leftarrow E^*_i$
    $j \leftarrow 0$
    while not converged do
        $\Phi^{j+1}_{i+1} \leftarrow \widehat{\mathcal{S}}_{\mu_i^{-1}}[\widehat{\Phi} - E^j_{i+1} - \mu_i^{-1} Y_i]$
        $E \leftarrow \mathcal{S}_{\gamma\mu_i^{-1}}[\widehat{\Phi} - \Phi^j_{i+1} - \mu_i^{-1} Y_i]$
        $j \leftarrow j + 1$
    end while
    $Y_{i+1} \leftarrow Y_i + \mu_i^{-1}(\widehat{\Phi} - \Phi^j_{i+1} - E^j_{i+1})$
    $\mu_{i+1} \leftarrow \rho\mu_i$
    $i \leftarrow i + 1$
end while
```

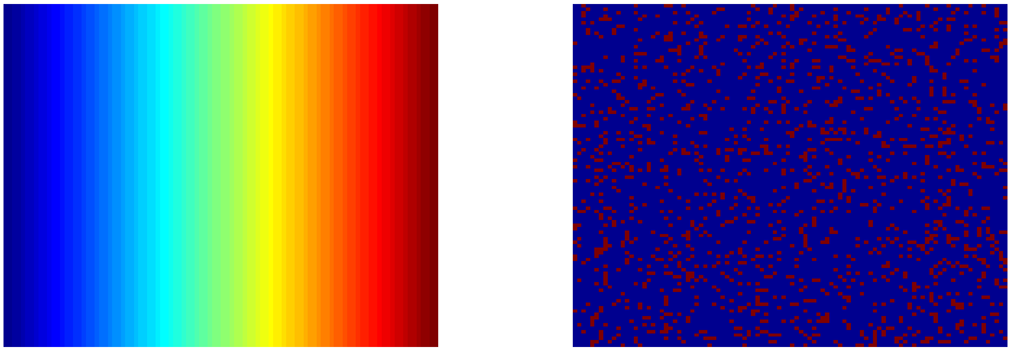

Figure 7.5: The reconstructed rank-1 dataset (Left), and the extracted noise mask (Right).

to find true principal components in the presence of outliers/noise. The two main approaches to robust PCA attempt to either find a better correlation between points, or use an iterative/projective approach to converge to a solution that separates the outliers. Robust correlation estimators are employed in [31, 97], and projective approaches are employed in [52, 96]. There have also been combination approaches that balance effectiveness and speed [49]. The approaches developed in [16] are iterative methods that attempt to determine the principal components and separate them from noise. Revisiting the corrupted rank-1 dataset, we can employ Algorithm 7.2 to exactly separate the rank-1 and noise component (Figure 7.5).

7.3 Electron Energy-Loss Spectroscopy

Moving on to real-world data, we consider the removal of noise and artifacts from biological samples. In this section we introduce the imaging technique of Electron Energy-Loss Spectroscopy (EELS), where a biological sample placed in a microscope and imaged with electrons. High energy electrons are passed through the sample, and the energy remaining in the electrons, after leaving the sample, are measured. The number of electrons measured at a given energy level are binned to form a histogram of counts at those energies. As the energies are on the order of 10^9eV, it is common to analyze the measurements in terms of energy lost while passing through the sample. This histogram, collected for each pixel, is called an electron energy-loss spectra, and this imaging process produces a spectrum image. This naming convention overlaps with our definition of spectrum (singular/eigenvalues), but we will explicitly distinguish between the two uses when necessary, and all other cases will be clear from context.

With the spectral image, we can analyze the biological sample's properties. For the purposes of this thesis, we will restrict our analysis to that of attempting to determine the elemental makeup. See Figure 7.6 for an example of spectra and the spectral image. Electrons lose a variable amount of energy depending on the element that they intersect (depending on the number of electrons/protons/etc.) The specific response pattern of these elements is referred to as reference spectra (Figure 7.7).

The general techniques for EELS processing follows the standards of many image processing techniques involving channel dimensions not necessarily in the visible spectrum [77, 89]. As many of the electrons fail to interact with the elements in the sample, most of the electrons lose little-to-no energy, resulting in a large "zero-loss" response. This background peak is several orders-of-magnitude larger than any signal response from the elements, so many EELS processing techniques first separate

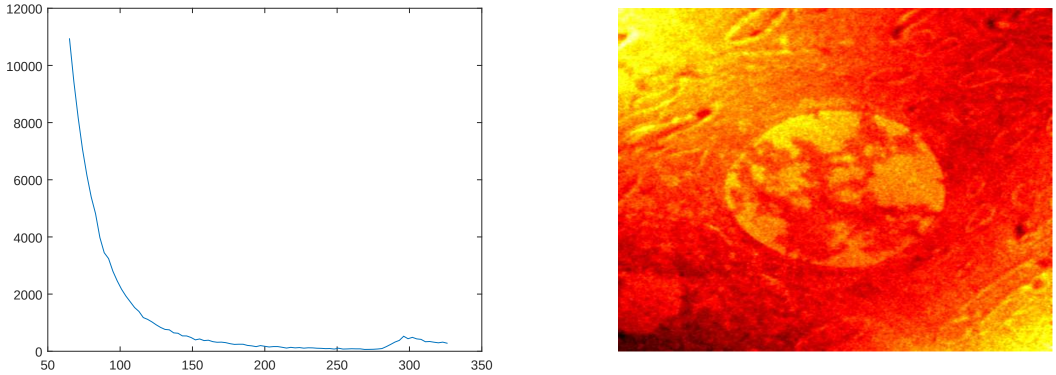

Figure 7.6: A collected EEL spectra from a biological sample (Left), and a spectral image from the sample biological sample. The number of electrons measured is the y-axis for the amount of energy lost (x-axis).

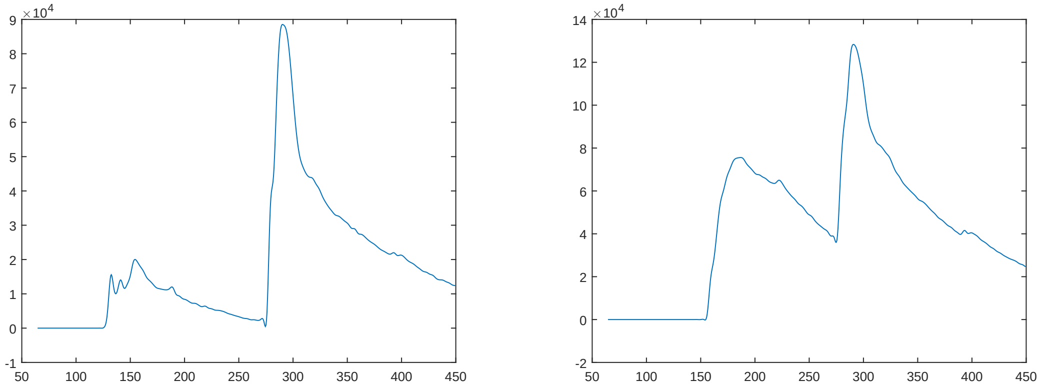

Figure 7.7: A response spectra for phosphorus (Left) and sulfur (Right). Note that, also shown here is a carbon response that is present in organic samples, is the second peak on the far right of the images.

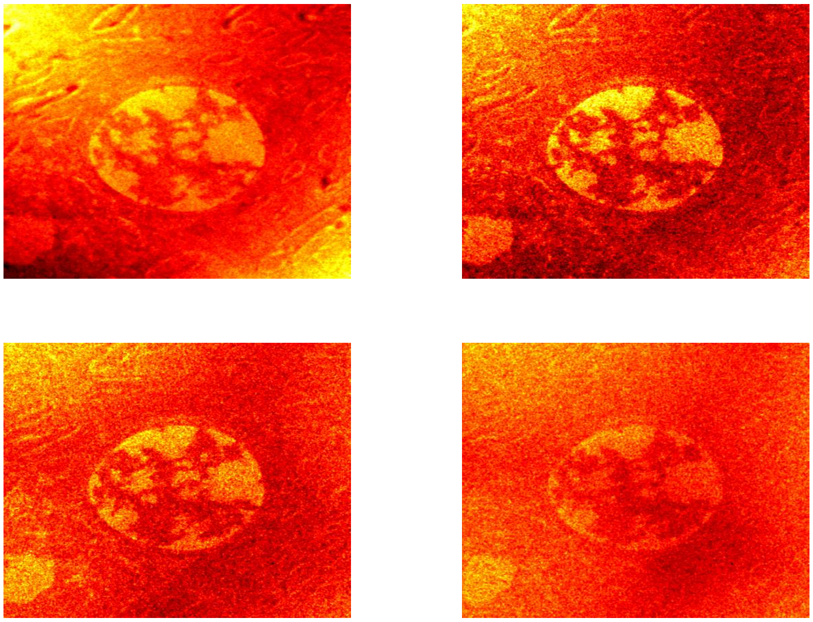

Figure 7.8: Image slices taken at various energy-loss levels (92eV, 152eV, 212eV, and 272eV), arranged from top-left to bottom-right.

the background from the sample as a first step. Also, when measurements of the electron energies are made, there are a number of confounding issues that add noise to the sample. Electrons that come into contact with other electrons can change trajectory, and the results from those electrons are registered incorrectly. This is in addition to anomalies added by the microscope itself (e.g. lens calibration issues, vacuum sealing, environmental factors, etc.). Figure 7.8 shows some corrupted slices from an EEL image.

We now give a brief summary of the techniques and language used to process these energy-loss spectra. For more general information on the processing of EELS, see [36, 37, 50, 90]. Much of the work presented here is available in greater detail in [35, 36, 37, 50, 50, 60, 61, 62, 90, 93].

Due to the nature of imaging trace elements in samples, the spectra will be dominated by the background. This complicates processing efforts, as the integrated value of a core-edge may be several orders-of-magnitude smaller than the background. To account for this, many EELS processing techniques remove the background loss before attempting to determine trace elements [37]. The *linear least-squares* technique uses the well known *inverse power-law* to predict background spectra [35, 36, 37],

$$b[i] = a_0 x[i]^{-a_1}. \tag{7.2}$$

where $x[\cdot]$ is the energy-loss, and $b[\cdot]$ is the electron count at a specific energy-loss (for parameters a_0 and a_1). The behavior of background counts can be modeled by finding the appropriate parameters, a_0 and a_1, for the function. This is known to well approximate the background locally, and we can subsequently remove the background by this fitting. The trace elements are then obtained by subtracting the background.

In processing schemes that depend on modeling the entire energy-loss spectrum, the zero-loss and low-loss region of the spectrum can skew trace element results due to its large magnitude, relative to core-loss edges. When comparing spectra, small differences in the zero-loss are weighted more than differences during higher energy-losses (lower electron counts). Using knowledge that the core-edge losses fluctuate rapidly, using the difference (approximately, the local slope) of the spectra, results in low values along the zeros-loss, and distinctive values for the element edge-losses. When assuming the inverse power-law model from (7.2), the difference spectra for the background will be approximately constant, allowing for its subtraction from the sample [50, 93]. Where presented in the thesis, we shall use the forward, backward, and centered difference of a spectra $\varphi[\cdot]$ where applicable,

$$\begin{aligned}\text{Forward Difference: } d_i &= \frac{\varphi[i+1]-\varphi[i]}{x[i+1]-x[i]};\\ \text{Centered Difference: } d_i &= \frac{\varphi[i+1]-\varphi[i-1]}{2(x[i+1]-x[i])};\\ \text{Backward Difference: } d_i &= \frac{\varphi[i]-\varphi[i-1]}{x[i]-x[i-1]}.\end{aligned}$$

A modification of linear least-squares, attempts to determine multiple trace elements simultaneously [60, 61, 62]. *Multiple least-squares* was presented as a tool useful when detecting overlapping core edges. The flexibility in the method allows for the use of linear least-squares to fit the background (and then fit the element core edges), or fit the background and core edges together. This also allows for reference background spectra to be used in the least squares fit. As the method utilizes least squares curve-fitting, quantitative methods can be employed, allowing for elemental counts.

Statistical analysis techniques allow for the study of global relationships in the sample. For an image represented as an $n \times n \times c$ data cube, there are $m = n^2$ pixels, and to each pixel, there is an associated spectra of length c. We now define the matrix of spectra as,

$$\Phi = [\varphi_1, \varphi_2, \ldots, \varphi_m],$$

a collection of spectra obtained from the imaging (spectra are obtained from each pixel of the image). Statistical techniques analyze the sample by considering the mean and variance of the spectra [12, 13]. The specific example we give here is PCA

[88]. This method performs a principal component decomposition where each of the spectra are separated into components that explain the variance in the sample. Furthermore, the components lie on subspaces that are perpendicular to each other (essentially meaning that there is no mixing between components). Principal component separation is meant to separate the major aspects of the sample, thereby separating the background and any trace elements. The separation can be written as:

$$\Phi^T \Phi = \sum_{i=1}^{r} \lambda_i v_i v_i^T.$$

Here, v_i represents the ith component of the sample (eigenvector), and λ_i represents the ith scaling coefficient (eigenvalue). In this regard, PCA is flexible, as it allows for separation of spectral maps, and/or the separation of the average core-loss edges. This can be employed to find, eigen-spectra (the spectra that capture the variance of the sample), or eigen-maps (a mapping that gives the components of the image that contain the most variance).

Analyzing the EELS spectra through statistical methods [88], produces results that consider the global relationships of the data. This approach is susceptible to noise and artifacts present in the sample. To increase the accuracy of these methods, a sub-sample of the spectra can be used to approximate the background and reference spectra. An experimentalist selects a subset of pixels from the spectrum-image and performs PCA. If attempting to separate the background spectra, sub-blocks are chosen where only the background (or no element trace) is present in the sample. As PCA is sensitive to outliers [51], we desire to have a strong signal response from only the background. By selecting pixels that only have the background spectrum present, there is no influence from the trace elements, and a clear background spectrum is obtained taking the top rank 1 component of the sample. This helps to mitigate the effect of spectrum mixing.

7.4 Removing EELS Artifacts with RPCA

In this section we propose a framework for obtaining high-accuracy EELS spectral maps given biological samples. This is true even in the presence of imaging artifacts and noise. This is accomplished by pre-processing the samples using robust PCA to separate the noise and true signal, and then using a combination of standard techniques to produce the spectral maps. By "high-accuracy," we mean that the residuals associated with the fittings are small. The effectiveness of this technique becomes more apparent when overlapping core-edges are present. Finally, we compare our results with standard techniques and show the gain in quality when employing the ensemble. Note that this work was in collaboration with Dr. Ricahd Leapman and Dr. Maria Aronova at the National Institutes of Health. The experimental samples are presented courtesy of the Dr. Leapman, and the National Biomedical Imaging and Bioengineering Laboratory.

By taking the global structure of the sample spectra into consideration, the processing methods become more robust to outliers, i.e., less sensitive to noise corruption. Global techniques have the effect of finding the central tendency of the data, and we shall exploit this structure to produce high-accuracy spectral maps.

We accomplish this goal by employing RPCA. In the imaging of biological samples, the artifacts that occur can be thought of as sparse errors, first, as they corrupt a small percentage of the sample pixels, and second, as the spectrum of the noise/artifacts are not necessarily correlated to the biological sample itself. This is the case with various imperfections, like lens mis-calibrations and measurement fluctuations. Under a model like standard PCA, these anomolies become outliers, that mix the sample and anomaly spectrums. Robust PCA does not suffer from this issue.

To form this method, we shall use the combined strength of the techniques discussed above. We still wish to perform quantitative analysis, so it's essential that after processing, specific elemental counts are still possible. An explanation of the method is given below, detailing the steps taken and their effect.

1 Pre-processing:

After a sample is imaged, each spectra is stacked into a matrix Φ. To ensure that the sample is unbiased, the mean count for each energy loss is computed and subtracted from the sample. The sample is then normalized by dividing the largest count (with the mean subtracted).

2 De-noise with Robust PCA:

To remove any artifacts and speckle noise from the sample, robust PCA is performed to separate the noise component, E, from the sample. The signal component, Φ, obtained, is now used in further processing.

3 User Selects Background Area:

To isolate the background in the sample, an area (or set of areas) is chosen that contains no trace elements. This is performed by an experimenter, and makes use of their experience with similar samples.

4 Use PCA to find Background Spectra:

With the background areas selected, PCA is performed. Taking the largest eigen-spectra, results in the background reference to be used.

5 Subtract Background:

With this background reference, a least-squares fit is performed for each spectra, and the background is removed from the sample.

6 Processing:

To allow for quantitative analysis, the divided normalizing constant must be multiplied, and the subtracted mean must be re-added. This provides a sample that can be interpreted as electron counts of the remaining trace elements.

7 Perform Multiple Least Squares:

With the background removed, the reference spectra for the trace elements present are used in a multiple least-squares fit to account for any overlapping core-edges. This produces the corresponding component maps that can be used for quantitative analysis.

7.4.1 Phantom EELS Processing

To show the effectiveness of using a combined, or ensemble, approach, we first apply the method to an artificial sample with noise added. The spectral-image slices at various eV are shown in Figure 7.10. The sample is generated over the range [90 - 270 eV] (with a step size of 3 eV). For generality, we shall denote the number of energy losses with c (The given sample has $c = 61$.), and the sample contains a background following the inverse power law,

$$b[i] = a_0 x[i]^{-a_1},$$

with $a_0 = 10^5$ and $a_1 = 3.25$ (that we assume is unknown when processing the sample). Phosphorus and sulfur are present in the phantom sample, with phosphorus in the large organelle, and sulfur in the smaller islets. Where they are present, phosphorus and sulfur core-edge intensities are 2 orders-of-magnitude smaller than the background signal. Figure 7.9 gives an image representation of the phosphorus/sulfur locations.

We employ the Poisson distribution,

$$p(\lambda, k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

to add shot noise to the data. The parameters λ and k are the mean and occurrence count, respectively. As working with large counts can be inaccurate due to round-off errors, we will approximate the noise with a Gaussian distribution,

$$p(\mu, \sigma, t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\dfrac{(t-\mu)^2}{2\sigma^2}}.$$

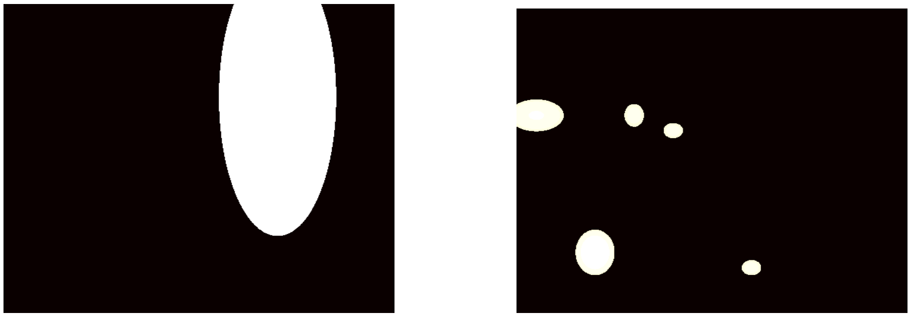

Figure 7.9: The true phosphorus (Left) and sulfur (Right) locations in the artificial sample.

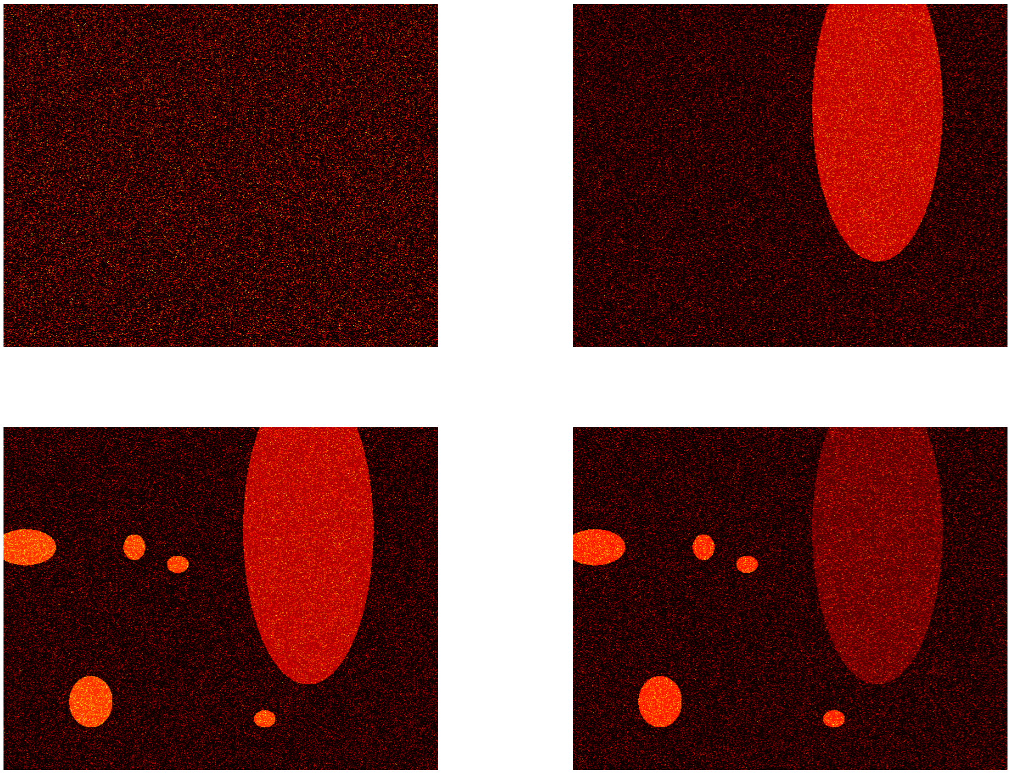

Figure 7.10: Slices from the noisy artificial sample.

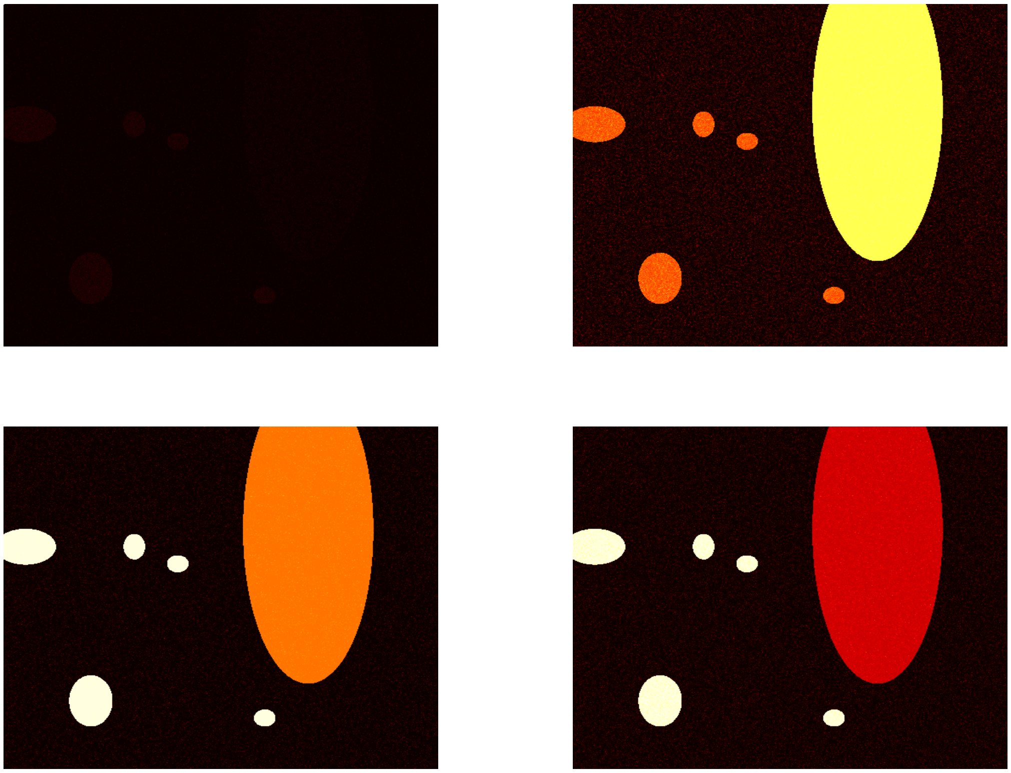

Figure 7.11: Slices from the artificial sample after removing the noise with RPCA.

In this approximation, the mean $\mu = \lambda$, and the standard deviation $\sigma = \sqrt{\lambda}$. The efficacy of this approximation is shown with the *Central Limit Theorem* [85, 94]. For each energy loss over the number of pixels, m, the estimated mean $\widehat{\mu}$ is computed by the maximum likelihood estimator,

$$\widehat{\mu}(i) = \frac{1}{m} \sum_{k=1}^{m} \Phi(i,k),$$

where $\Phi(i,k)$ is the energy-loss count for the ith energy level and kth pixel.

After applying the ensemble method to the phantom, we present, in Figure 7.12, visual maps comparing results between standard multiple least-squares and PCA with our method. In Table 7.1, we present the χ^2 results for the phantom sample, with noise and without. We also give the signal-to-noise ratios obtained, in Table 7.2. To generate the χ^2 statistic, we first compute the sample standard deviations for each energy loss s_i,

$$s_i = \sqrt{\frac{1}{m-1} \sum_{k=1}^{m} (\Phi(i,k) - \widehat{\mu}(i))^2}.$$

The χ^2 is then generated with,

$$\chi^2 = \sum_{i=1}^{n} \sum_{k=1}^{m} \frac{(\Phi(i,k) - f(i,k))^2}{s_i},$$

where $f(i,k)$ is the generated fit from PCA, multiple least-squares, and the ensemble method. The peak signal-to-noise ratio is,

$$\text{psnr} = 20 \log_{10}(\max(\Phi)) - 10 \log_{10}(\text{mse}),$$

where mse is the mean-squared-error between the sample spectra and the predicted spectra.

χ^2 Table			
	Phosphorus	Sulfur	Residual
PCA	6.7E9	2.8E9	1.8E9
MLS	2.7E9	1.8E8	1.9E9
Ensemble	**1.1E9**	**1.3E8**	**1.0E9**

Table 7.1: A table of chi-square values.

PSNR Table			
	Phosphorus	Sulfur	Residual
PCA	-6.94	-4.49	21.75
MLS	8.38	**21.57**	**21.62**
Ensemble	**10.55**	21.23	21.47

Table 7.2: A table of peak signal-to-noise ratios.

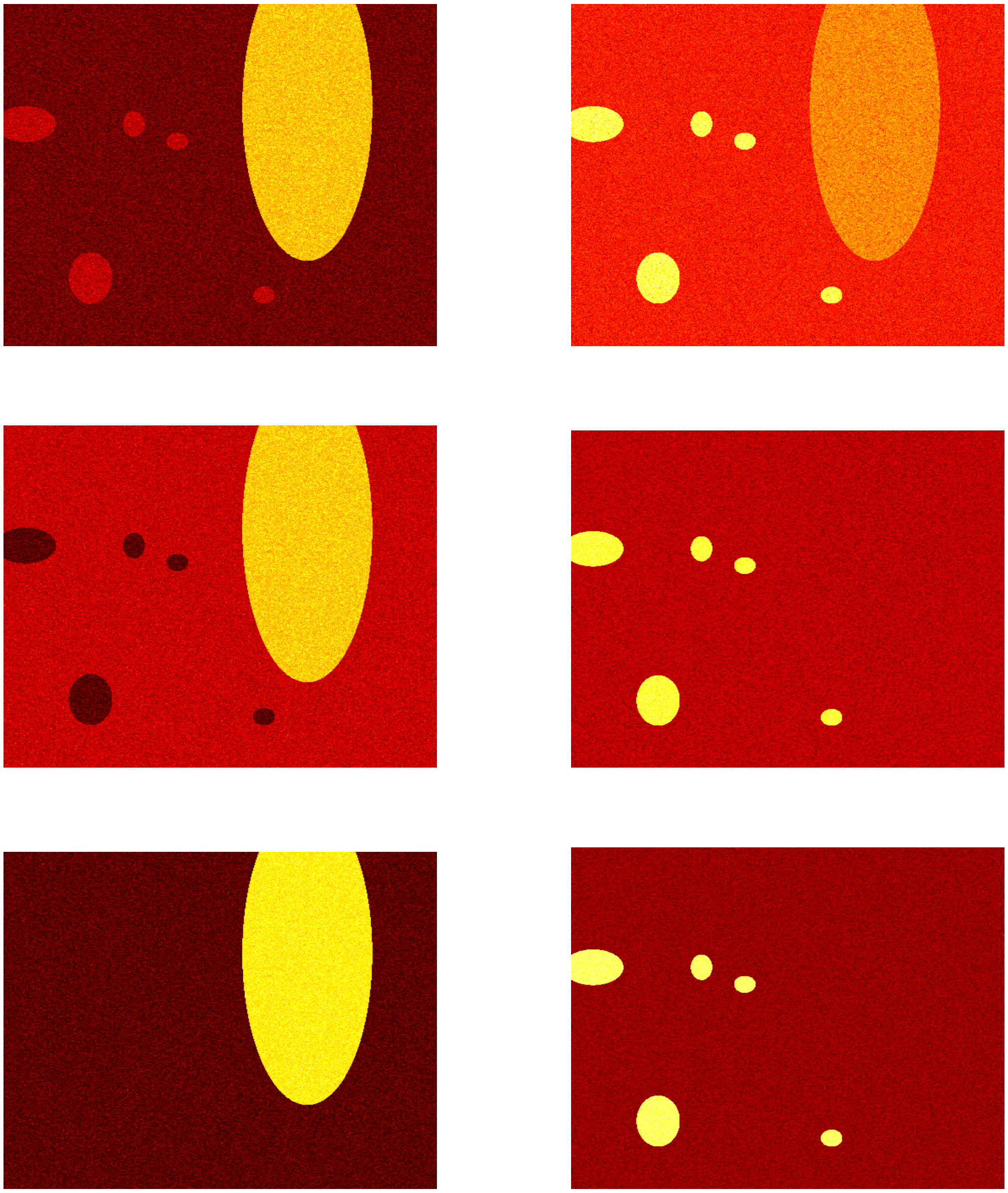

Figure 7.12: The results after applying the three schemes discussed, PCA (Top row), MLS (Middle row), and Ensemble (Bottom row).

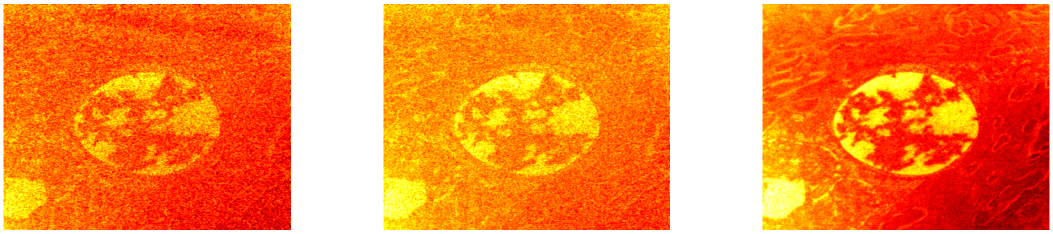

Figure 7.13: The phosphorus maps produced by the PCA (Left), MLS (Middle), and the Ensemble (Right).

7.4.2 Experimental EELS Samples

We now perform a similar analysis on an experiential sample. Unfortunately, as we don't have ground truth, our results are more qualitative in nature. We compare with standard techniques and show RPCA's effectiveness at removing noise. See Figure 7.13.

Chapter 8: Learning Nonlinear Structure

8.1 Robust Manifold Learning

The natural extension of PCA is to compare similarities between nonlinear transformations of the dataset in the form of kernels (KPCA). In this same vein, we may wish to add a notion of robustness to KPCA by employing an error regularizing term. This motivates our introduction of the Robust Manifold Learning (RML) problem,

$$\begin{aligned} &\text{minimize: } \operatorname{rank}(K(\Phi)) + \gamma\|E\|_0 \\ &\text{subject to: } \Phi + E = \widetilde{\Phi}, \end{aligned} \tag{8.1}$$

where $K(\Phi)$ is the *kernel matrix* (or similarity matrix) with $K_{i,j} = f(\varphi_i, \varphi_j)$ for a kernel function f. Here, the kernel is normalized to have $\lambda_1 = 1$. As with many formulations, we shall study the convex relaxation of this problem,

$$\begin{aligned} &\text{minimize: } \|K(\Phi)\|_* + \gamma\|E\|_1 \\ &\text{subject to: } \Phi + E = \widetilde{\Phi}. \end{aligned} \tag{8.2}$$

Much of the intuition behind this approach can be gleamed from an understanding of robust PCA and its variations. Our discussion of convergence in the following section was encouraged partly from the results obtained in [68] for non-convex RPCA. The work by Candès on the convergence of iterative schemes to minimize the nuclear norm, inspired the use of iterative methods under this framework [15], and the unification of many of the standard nonlinear dimensionality reduction tools under the kernel PCA umbrella in [9, 46] was the basis for its use here (as well as its embedding results for clustering and learning [63, 78]). In the realm of related work, there is the paper by Shahid et. al. [75]. The authors start with the RPCA assumption of a low rank and sparse decomposition, and add a regularizer term that enforces smoothness on the similarity graph of the the dataset,

$$\min_{\widetilde{\Phi}=\Phi+E} : \|\Phi\|_* + \gamma\|E\|_1 + \tau \cdot \operatorname{trace}(\Phi C \Phi^T),$$

with $\Phi C \Phi^T$ as the similarity. Taking the trace of this term imposes a penalty for non-smooth results, as it encourages a slow variation in the spectrum. The difference in the proposed method is that we don't require the low-rank assumption in the original space. In fact, we place no assumptions on the dataset, except that there exists a true low-rank embedding.

For convenience, we shall generally define notation where used, but for brevity, we define a few terms that are used throughout the chapter. We take the dataset $\Phi = [\varphi_1, \varphi_2, \ldots, \varphi_m]$ to be an $n \times m$ matrix where each column corresponds to a datapoint. By embedding of a kernel, we write,

$$\Theta = [\theta_1, \theta_2, \ldots, \theta_m] = \Lambda^{\frac{1}{2}} V^T,$$

where $K(\Phi) = \Theta^T \Theta = V \Lambda V^T$. By *neighbor set* $\Omega(\varphi_k)$ and $\Omega(\theta_k)$ we refer to the indices of the closest (in ℓ_2 norm) datapoints in Φ and Θ respectively.

8.2 Inverse Mapping

While this RML formulation is similar in form to that of (7.1), there are major differences that must be addressed. We shall, in general, restrict the kernel function to those acting on the inner-product between elements $f(\varphi_i, \varphi_j) = f(\langle \varphi_i, \varphi_j \rangle)$. We shall also restrict ourselves to kernel matrices $K(\cdot)$ that are positive semi-definite (or positive definite), but even with these simplifying assumptions, the class of kernel functions is still too large for the same analytic techniques applied to problems with linear structure. If we plan to use this technique in practical applications, we require a method to minimize the rank of the kernel matrix, and this necessitates a spectral decomposition. This brings us to the major issue with nonlinear methods; there is in general no well-defined inverse for an embedding obtained from KPCA. As mentioned in the preliminaries, many have studied the inverse problem [2, 47, 57, 66] with many techniques making the assumption that a reasonable inverse uses an interpolation technique to approximate the points in the original space [47, 66].

The algorithm employed to solve the RML problem is modeled from the Augmented Lagrange multiplier algorithms presented for RPCA (Algorithm 7.2). The major additions to this approach are the inclusion of the nonlinear embedding and its inversion. A kernel K is formed for the dataset Φ (generally the heat kernel $e^{-\frac{\|\varphi_i - \varphi_j\|_2^2}{\sigma^2}}$), and an embedding is formed and thresholded for the kernel. Once the threshold has been applied, an inverse operation is performed as follows,

$$\varphi_k = \sum_{i \in \Omega(\theta_k)} \frac{a}{\|\theta_k - \theta_i\|_2^2} \varphi_i. \tag{8.3}$$

Here a is a normalization constant chosen so that the sum of ℓ_2 distances is 1,

$$\sum_{i=1}^{m} \sum_{j \in \Omega(\theta_i)} \|\theta_i - \theta_j\|_2^2 = \frac{1}{a}.$$

Before presenting the algorithm, we define the embedding and inverse operators.

Definition 8.2.1 (Embedding Operator)**.** Define $\mathcal{E}$ as the embedding operator such that,

$$\Theta = \mathcal{E}[K(\Phi)],$$

where $K(\Phi)$ is the kernel matrix on the dataset Φ.

Definition 8.2.2 (Inverse Operator)**.** Define $\mathcal{E}^{-1}$ as the embedding operator such that,

$$\Phi = \mathcal{E}^{-1}[\Theta],$$

where Θ is an embedding formed from a kernel matrix. The inverse is performed using the interpolation formula above (8.3).

Note that in Algorithm 8.2, we use the shrink operators defined in Definition 7.2.3 and 7.2.4.

8.3 Convergence Discussion

Let $\Phi^i, E^i \in \mathbb{R}^{n \times m}$ be the manifold approximation and sparse term, respectively, from the ith iteration of the algorithm. Let Φ^* be the ideal manifold, and E^* be the ideal sparse error term. Assume E^* is sparse with s non-zero terms. This

Algorithm 4 ALM for Robust Manifold Learning

```
Require: $\Phi$ , $\gamma$ , $\mu_0$ , $\rho$
$J = \max\{\|\widetilde{\Phi}\|_2 , \gamma^{-1}\|\widetilde{\Phi}\|_\infty\}$
$Y_0 = \frac{\mathrm{sgn}(\widehat{\Phi})}{J}$
$i \leftarrow 0$
while not converged do
    $\Phi_{i+1}^0 \leftarrow \Phi_i^*$
    $E_{i+1}^0 \leftarrow E_i^*$
    $j \leftarrow 0$
    while not converged do
        $K_{i+1}^{j+1} \leftarrow K(\widetilde{\Phi} - E_{i+1}^j - \mu_i^{-1} Y_i)$
        $\Theta_{i+1}^{j+1} \leftarrow \mathcal{E}[K_{i+1}^{j+1}]$
        $\Theta_{i+1}^{j+1} \leftarrow \widehat{\mathcal{S}}_\mu[\Theta_{i+1}^{j+1}]$
        $\Phi_{i+1}^{j+1} \leftarrow \mathcal{E}^{-1}[\Theta_{i+1}^{j+1}]$
        $E \leftarrow \mathcal{S}_{\gamma\mu_i^{-1}}[\widetilde{\Phi} - \Phi_{i+1}^j - \mu_i^{-1} Y_i]$
        $j \leftarrow j + 1$
    end while
    $Y_{i+1} \leftarrow Y_i + \mu_i^{-1}(\widetilde{\Phi} - \Phi_{i+1}^j - E_{i+1}^j)$
    $\mu_{i+1} \leftarrow \rho\mu_i$
    $i \leftarrow i + 1$
end while
```

means at most $\frac{s}{n}$ points contain errors, with $s \ll n$. Moreover, these s points are randomly distributed across the manifold. Let $\Omega = \{j : \exists i \text{ such that } E^*_{j,i} \neq 0\}$, such that $|\Omega| \leq s$.

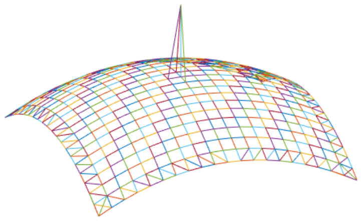

Figure 8.1: Example of manifold assumption with a corrupted point

We define $K(\Phi^i) = K(\Phi^*) + K^\epsilon$, where K^ϵ is the error in the kernel caused by corruption of the s points. We need three assumptions to guarantee that the low frequency eigenvectors on the corrupted kernel $K(\Phi^i)$ will remain sufficiently smooth and faithful to the unperturbed eigenvectors of $K(\Phi^*)$.

First, the manifold must be densely sampled compared to the number of corruptions, as in Figure 8.1. This is because there exists an all zeros sub matrix of K^ϵ of size $\left(m - \frac{s}{m}\right) \times \left(m - \frac{s}{m}\right)$. As s grows, so does $\|K^\epsilon\|$, which in turn corrupts the eigen-decomposition.

The second assumption is that the eigenvectors are sufficiently spread across the manifold. This is interpreted as the non-linear version of the Incoherence Condition η (Definition 7.2.1) from [17]. Namely, we assume the low frequency eigenvectors of Φ^*, which we denote $V = [v_1, ..., v_r]$, satisfy

$$\max_i \|V^\intercal e_i\|^2 < \frac{\eta r}{n},$$

where r is the dimension of the manifold. This guarantees that none of the low frequency eigenvectors are concentrated on a small section of the manifold. If this assumption is violated, and eigenvectors v_i were concentrated on some small number of points $c \ll m$, then one or two corrupted points on that section of the manifold would have a large effect on v_i.

The third assumption is that the eigengap between λ_r^t and λ_{r+1}^t is sufficiently large for some diffusion time t. Given the assumption that the data lies on an r-dimensional manifold, and given the freedom of choice in diffusion time, this assumption can be easily satisfied.

The interplay of these assumptions can be more easily observed given the eigenspace perturbation Theorem 4.11 of Stewart [82]. By assuming that the errors in K^ϵ are sufficiently randomly distributed and the low frequency eigenvectors are sufficiently spread out, the norm of the error generated will be sufficiently small with respect to the eigengap. Thus, the eigenvectors of the the perturbed kernel $K(\Phi^i)$ are sufficiently close to the original eigenvectors V of the kernel $K(\Phi^*)$.

8.4 Circle Embeddings

We now present some results obtained using the RML approach. We start with a clear example of when standard RPCA fails by adding sparse noise to an embedding of a circle.

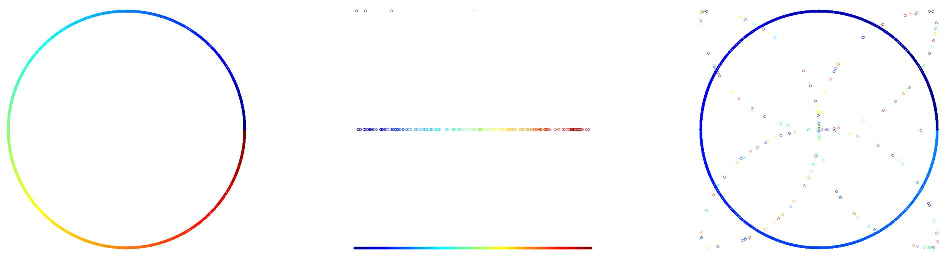

Figure 8.2: This figure shows the original embedding (Left), noise signal (Middle), and combined signal (Right).

We sample sine and cosine functions $m = 1000$ times using $n = 4$ frequencies for each, resulting in a dataset, Φ, of size $2n$-by-m where each column is a point in Φ. Sparse noise, E, is then added to the dataset by randomly selecting 80 indices and biasing the location. This is a relatively small amount of corruption, but due to the true manifold being non-linear, PCA and RPCA will fail to converge to the true solution. We also compare our results to those of standard KPCA, which results in a solution that still contains the sparse errors. To further show the effectiveness of our technique, we also present the iterates of the embedding produced during the algorithm. As we sample trigonometric functions here, the optimal 2D embedding is that of a circle. See Figures 8.3 and 8.4.

The results show that with this example, RML converges to the solution rather quickly (after 9 iterations). The results in Figure 8.4 show the stark difference in results. Particularly, RPCA converges to a completely different manifold, displaying one of the pitfalls with assuming linear structure. The "closest" linear manifold is completely different from the true manifold.

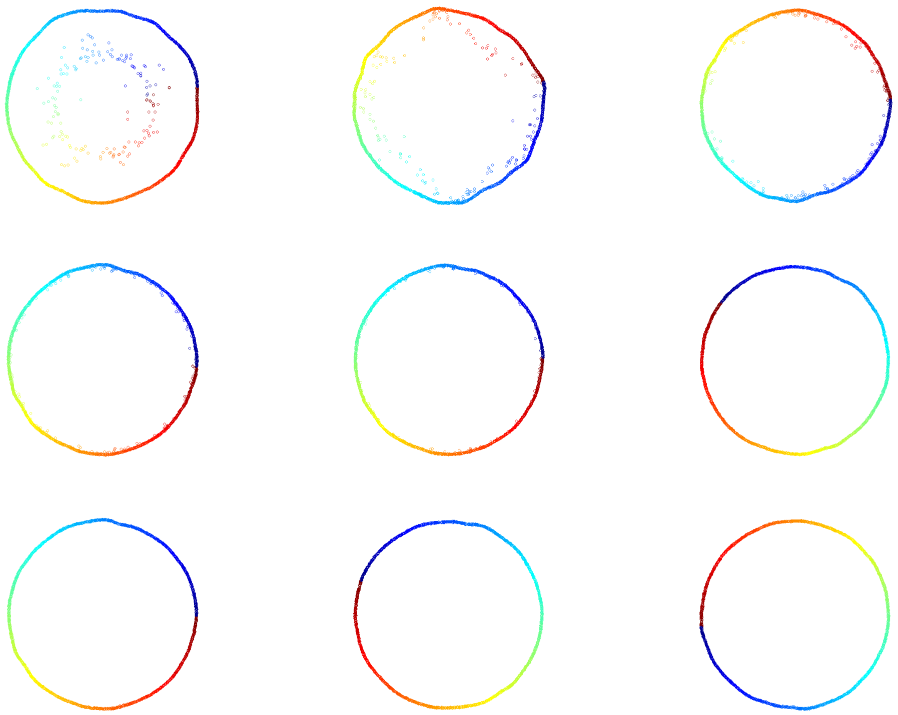

Figure 8.3: This figure shows the low-rank results for iterations 1 through 9 using RML.

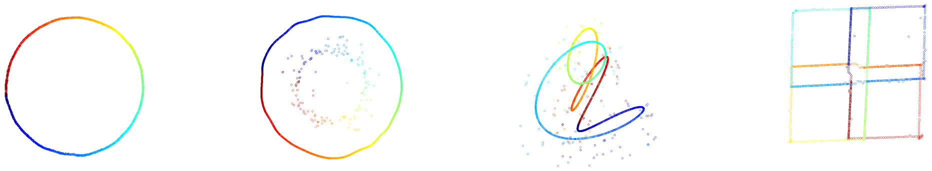

Figure 8.4: This figure shows the embeddings obtained after the various techniques are employed. From left to right, we present the final results after RML, kernel PCA, standard PCA, and standard robust PCA.

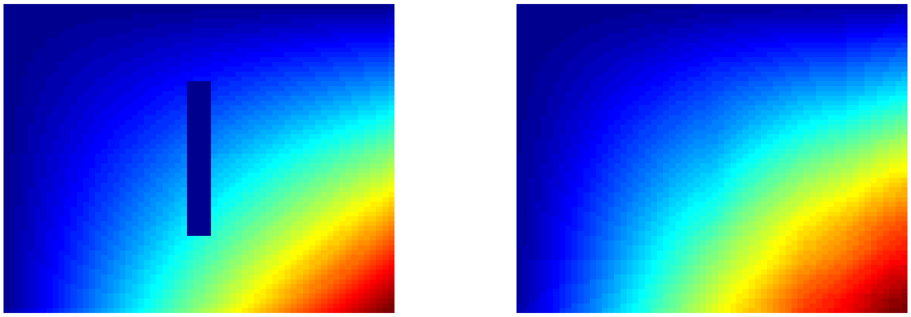

Figure 8.5: The image shown on the left is the original, and the image on the right is the result after employing the RML algorithm.

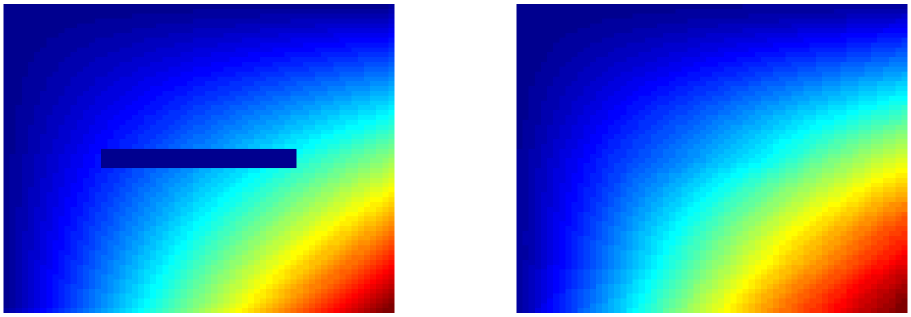

Figure 8.6: The image shown on the left is the original, and the image on the right is the result after employing the RML algorithm.

8.5 Inpainting

We employ the RML algorithm to inpaint images with missing sections. The results are shown in Figures 8.5, 8.6, 8.7, and 8.8.

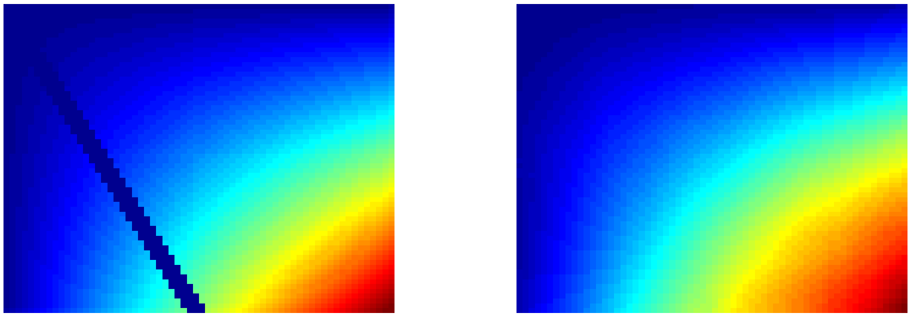

Figure 8.7: The image shown on the left is the original, and the image on the right is the result after employing the RML algorithm.

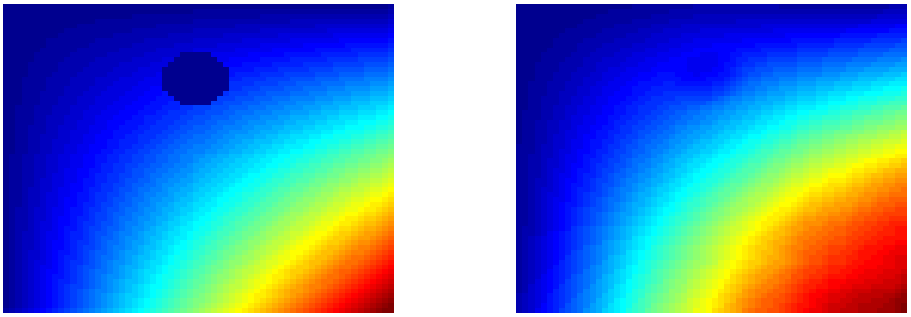

Figure 8.8: The image shown on the left is the original, and the image on the right is the result after employing the RML algorithm.

8.6 Towards Nonlinear Inverse Mappings

For the inverse mapping described in Equation (8.3) we take the standard convention of interpolating the updated datapoint using the embedded points. This is unsatisfactory, as it provides little guarantees as to the solution produced. We would expect this to be the case, as we place no restrictions on the dataset or kernel employed. What we aim to do in this section, is present an initial path to the rigorous study of manifold learning techniques that require a nonlinear mapping. While the results presented here are not complete, they do illuminate some of the issues inherent to inverse/pre-image problem.

The key insight, is that we perturb the spectrum of the kernel matrix by an adjustable parameter μ. If this parameter is sufficiently small, we might expect to be able to bound the accuracy of an inversion method. We start this analysis by defining the *spectrum-parametrized kernel matrix*, then bound the error of small perturbations of the embedding set (with some assumptions). We then show that, after assuming some restrictions on the inverse mapping, the error between the optimal kernel and the current iteration approaches 0 as the iterations progress.

Definition 8.6.1. The point sets Φ and Θ are ordered sets of m points in $\mathbb{R}^n$ and $\mathbb{R}^{n'}$ respectively,

$$\Phi := \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n,$$

$$\Theta := \{\theta_k\}_{k=1}^m \subset \mathbb{R}^{n'}.$$

Moreover, we shall assume that the elements in Φ and Θ are unit norm,

$$\|\varphi_k\|_2 = \|\theta_k\|_2 = 1 \text{ for } k = 1, \dots, n.$$

Definition 8.6.2. The neighborhood of a points φ shall be denoted $\Omega(\varphi)$, where

$$\Omega(\varphi) := (k \,|\, \varphi_k \text{ is a neighbor of } \varphi)\,.$$

Definition 8.6.3. The convex hull of a set of points Φ shall be denoted conv(Φ), where

$$\operatorname{conv}(\Phi) := \left\{ \sum_{k=1}^{|\Phi|} \alpha_k \varphi_k \;\middle|\; \sum_{k=1}^{|\Phi|} \alpha_j = 1 \ , \ \alpha_j \geq 0 \right\}.$$

Definition 8.6.4. The euclidean distance matrix of a set Φ, is defined to be D_Φ, where

$$D_\Phi := D_\Phi(i,j) = \|\varphi_i - \varphi_j\|_2^2.$$

Definition 8.6.5 (Spectrum-Parametrized Kernel Matrix)**.** We define a kernel K, as a function that acts on a dataset, $\Phi := \{\varphi_k\}_{k=1}^m \subset \mathbb{R}^n$, and maps it to a matrix $K(\Phi)$, by a continuously differentiable function. Furthermore, we define its spectrum-parametrization by $K(\Phi, \mu)$, that maps a dataset in the following manner,

$$K(\Phi, t) := K(\Phi) - \frac{1}{\mu} I.$$

The first lemma simply connects the parameter μ to a shift in the spectrum.

Lemma 8.6.6. Given a dataset Φ, its kernel $K(\Phi)$, and the parametrized kernel $K(\Phi, t)$, let $\dfrac{1}{\mu} < \lambda_r$, where λ_r is the smallest non-zero eigenvector of $K(\Phi)$ Then the following equality holds,

$$\|K(\Phi) - K(\Phi, t)\|_2 = \frac{1}{\mu}.$$

Proof.

$$\begin{aligned}
\|K(\Phi) - K(\Phi,\mu)\|_2 &= \|K(\Phi,0) - K(\Phi,\mu)\|_2 \\
&= \|K(\Phi,0) - K(\Phi,0) - K(\mathbf{0},\,\mu)\|_2 \\
&= \| - K(\mathbf{0},t)\|_2 \\
&= \left\|\frac{1}{\mu} I\right\|_2 \\
&= \frac{1}{\mu}\|I\|_2 \\
&= \frac{1}{\mu}.
\end{aligned}$$

■

Lemma 8.6.7. Let $\|\epsilon_j\|_2 \ll 1$ and let $\Theta = \{\theta_i\}_{i=1}^n \subseteq \mathbb{R}^m$. Assume the elements in Θ are unit norm, and define ε to be the maximum value of the perturbation matrix $E = E(i,j) = \epsilon_{ij}$,

$$\varepsilon := \max_{ij}\{\epsilon_{ij}\}.$$

Perturb Θ by E, such that

$$\widetilde{\Theta} := \{\tilde{\theta}_i\}_{i=1}^n = \{\theta_i + \epsilon_i\}_{i=1}^n.$$

Assume that any element in Θ is contained in the convex hull of its neighbor set in Θ and $\widetilde{\Theta}$. Let $\tilde{\alpha} \in \mathbb{R}^{|\Omega(\tilde{\theta}_i)|}$ denote the convex coefficients for the perturbed set $\widetilde{\Theta}$. Then the error (in ℓ_2 norm) of the reconstruction of θ_i as a convex combination of its perturbed coefficients is bounded by ε,

$$\left\|\theta_i - \sum_{j\in\Omega(\theta_i)} \tilde{\alpha}_j \theta_j\right\|_2^2 < O(\varepsilon^2).$$

Proof. Consider the two representation of θ_i. Write θ_i as its convex combinations and simplify the expression,

$$\begin{aligned} 0 =& \|\theta_i - \theta_i\|_2^2 \\ =& \left\| \sum_{j \in \Omega(\theta_i)} \alpha_j \theta_j - \sum_{j \in \Omega(\theta_i)} \tilde{\alpha}_j \tilde{\theta}_j \right\|_2^2 \\ =& \left\| \sum_{j \in \Omega(\theta_i)} \alpha_j \theta_j - \sum_{j \in \Omega(\theta_i)} \tilde{\alpha}_j (\theta_j + \epsilon_j) \right\|_2^2 \\ =& \left\| \sum_{j \in \Omega(\theta_i)} (\alpha_j - \tilde{\alpha}_j) \theta_j - \sum_{j \in \Omega(\theta_i)} \tilde{\alpha} \epsilon_j \right\|_2^2 \end{aligned}$$

Using a variation of the triangle inequality, we have,

$$0 = \left\| \sum_{j \in \Omega(\theta_i)} (\alpha_j - \tilde{\alpha}_j) \theta_j - \sum_{j \in \Omega(\theta_i)} \tilde{\alpha} \epsilon_j \right\|_2^2 \geq \left| \left\| \sum_{j \in \Omega(\theta_i)} (\alpha_j - \tilde{\alpha}_j) \theta_j \right\|_2^2 - \left\| \sum_{j \in \Omega(\theta_i)} \tilde{\alpha} \epsilon_j \right\|_2^2 \right| \geq 0.$$

This gives the equality,

$$0 = \left\| \sum_{j \in \Omega(\theta_i)} (\alpha_j - \tilde{\alpha}_j) \theta_j \right\|_2^2 - \left\| \sum_{j \in \Omega(\theta_i)} \tilde{\alpha} \epsilon_j \right\|_2^2 .$$

Using matrix-vector notation, we have,

$$0 = \|\Theta_{\Omega(\theta_i)}(\alpha - \tilde{\alpha})\|_2^2 - \|E_{\Omega(\theta_i)} \tilde{\alpha}\|_2^2.$$

Here $\Theta_{\Omega(\theta_i)}$ is a matrix of the neighbors of θ_i, and $E_{\Omega(\theta_i)}$ is a matrix of the corresponding perturbations. After rearranging the equality, we have

$$\|\Theta_{\Omega(\theta_i)}(\alpha - \tilde{\alpha})\|_2^2 = \|E_{\Omega(\theta_i)} \tilde{\alpha}\|_2^2.$$

Applying Cauchy-Schwartz, we have

$$\|\Theta_{\Omega(\theta_i)}(\alpha - \tilde{\alpha})\|_2^2 \leq \|E_{\Omega(\theta_i)}\|_2^2 \cdot \|\tilde{\alpha}\|_2^2.$$

As $\tilde{\alpha}$ are convex combination coefficients,

$$\|\tilde{\alpha}\|_2^2 \leq 1.$$

Employing ε, we have

$$\left\| \sum_{j \in \Omega(\theta_i)} (\alpha_j - \tilde{\alpha}_j)\theta_j \right\|_2^2 = \|\Theta_{\Omega(\theta_i)}(\alpha - \tilde{\alpha})\|_2^2 \leq \|\varepsilon \mathbb{1}_m \mathbb{1}_{|\Omega(\theta_i)|}^T\|_2^2 = m|\Omega(\theta_i)|\varepsilon^2 \sim O(\varepsilon^2).$$

■

Lemma 8.6.8. Consider the two sets Φ and Θ. Let the elements in both sets be unit norm. Assume each element in both sets is contained in the convex hull of its respective neighbor set. Further, assume that their neighbor sets are equal,

$$\Omega(\varphi_i) = \Omega(\theta_i),$$

and that the distance matrices are equal,

$$D_\Phi = D_\Theta.$$

Then there exists a vector $\alpha \in \mathbb{R}^{|\Omega(\varphi_i)|}$, such that

$$\varphi_i = \sum_{j \in \Omega(\varphi_i)} \alpha_j \varphi_j,$$

and

$$\theta_i = \sum_{j \in \Omega(\theta_i)} \alpha_j \theta_j,$$

with

$$\sum_{j \in \Omega(\varphi_i)} \alpha_j = 1\,,\ \alpha \geq 0.$$

Proof. From the fact that the elements are unit norm and the pairwise distances are equal between points, we have

$$\langle \varphi_i, \varphi_j \rangle = \langle \theta_i, \theta_j \rangle.$$

Write φ_i and θ_i as their respective convex combinations,

$$\varphi_i = \sum_{j \in \Omega(\varphi_i)} b_j \varphi_j,$$

and

$$\theta_i = \sum_{j \in \Omega(\varphi_i)} c_j \theta_j,$$

with

$$\sum_{j \in \Omega(\varphi_i)} b_j = \sum_{j \in \Omega(\theta_i)} c_j = 1 \ , \ b_j, c_j \geq 0.$$

From the assumption about the equal neighbor sets,

$$|\Omega(\varphi_i)| = |\Omega(\theta_i)|.$$

Take the inner product of the sums and φ_i and θ_i respectively,

$$\langle \varphi_i, \varphi_i \rangle = \left\langle \sum_{j \in \Omega(\varphi_i)} b_j \varphi_j, \varphi_i \right\rangle = 1,$$

and

$$\langle \theta_i, \theta_i \rangle = \left\langle \sum_{j \in \Omega(\theta_i)} c_j \theta_j, \theta_i \right\rangle = 1.$$

Using the linearity of the inner product,

$$\sum_{j \in \Omega(\varphi_i)} b_j \langle \varphi_j, \varphi_i \rangle = \sum_{j \in \Omega(\theta_i)} c_j \langle \theta_j, \theta_i \rangle.$$

From the equality, with respect to inner product, we have

$$b_j = c_j.$$

Letting $\alpha_j = b_j = c_j$ concludes the proof. ■

The next two lemmas are in the spirit of the results presented in [68].

Lemma 8.6.9. Let Φ^* and E^* be the optimal solution to the RML problem presented in (8.1) using the heat kernel. Also let $\Phi^{(k)}$ and $E^{(k)}$ be the kth iterate of Algorithm (8.2). Lastly, let $\Delta E^{(k)} := E^* - E^{(k)}$. Then,

$$\|K_{\Phi^{(k+1)}} - K_{\Phi^*}\|_F^2 \leq \|K_{\Phi^*}\|_F^2 \cdot \sum_i \sum_j [\Delta\varphi]^2 .$$

Proof.

$$\begin{aligned}
\|K_\Phi - K_{\Phi^*}\|_F^2 &= \sum_i \sum_j \left[\exp\left(-\frac{\|\varphi_i - \varphi_j\|_2^2}{\sigma^2}\right) - \exp\left(-\frac{\|\varphi_i^* - \varphi_j^*\|_2^2}{\sigma^2}\right)\right]^2 \\
&= \sum_i \sum_j \left[\exp\left(-\frac{\|\varphi_i^* - \varphi_j^* + \varepsilon_i - \varepsilon_j\|_2^2}{\sigma^2}\right) - \exp\left(-\frac{\|\varphi_i^* - \varphi_j^*\|_2^2}{\sigma^2}\right)\right]^2 \\
&= \sum_i \sum_j \left[\exp\left(-\frac{\|\varphi_i^* - \varphi_j^*\|_2^2}{\sigma^2}\right)\right]^2 \left[\exp\left(-\frac{\|\varepsilon_i - \varepsilon_j\|_2^2 + 2\langle \varepsilon_i - \varepsilon_j, \varphi_i^* - \varphi_j^*\rangle}{\sigma^2}\right) - 1\right]^2 .
\end{aligned}$$

For notational convenience, and clarity, we shall denote the second exponent by $\Delta\varphi$,

$$\Delta\varphi := \frac{\|\varepsilon_i - \varepsilon_j\|_2^2 + 2\langle \varepsilon_i - \varepsilon_j, \varphi_i^* - \varphi_j^*\rangle}{\sigma^2}.$$

$$\|K_\Phi - K_{\Phi^*}\|_F^2 = \sum_i \sum_j \left[\exp\left(-\frac{\|\varphi_i^* - \varphi_j^*\|_2^2}{\sigma^2}\right)\right]^2 \cdot \left[\exp\left(-\Delta\varphi\right) - 1\right]^2 .$$

Using a first-order Taylor approximation, we have,

$$\exp(-\Delta\varphi) = 1 - \frac{|\Delta\varphi|}{1!}.$$

Plugging this in to the equality,

$$\begin{aligned}
\|K_\Phi - K_{\Phi^*}\|_F^2 &\approx \sum_i \sum_j \left[\exp\left(-\frac{\|\varphi_i^* - \varphi_j^*\|_2^2}{\sigma^2}\right)\right]^2 \cdot [\Delta\varphi]^2 \\
&\leq \sum_i \sum_j \left[\exp\left(-\frac{\|\varphi_i^* - \varphi_j^*\|_2^2}{\sigma^2}\right)\right]^2 \cdot \sum_i \sum_j [\Delta\varphi]^2 \\
&\leq \|K_{\Phi^*}\|_F^2 \cdot \sum_i \sum_j [\Delta\varphi]^2 .
\end{aligned}$$

This is sufficient to show the bound approaches 0 due to the following:

1. If points are far apart, $\|\varphi_i^* - \varphi_j^*\|_2^2 >> 0$, then the optimal kernel K_{Φ^*} tends towards 0.

2. If points are close together, $\|\varepsilon_i^* - \varepsilon_j^*\|_2^2 << 1$, and the difference term $\Delta\varphi$ tends to 0.

■

Lemma 8.6.10. Let Φ^* and E^* be symmetric and the optimal solution to the RML problem presented in (8.1) using the heat kernel. Also let $\Phi^{(k)}$ and $E^{(k)}$ be the kth iterate of Algorithm (8.2). Further assume that the inverse mapping is Lipschitz continuous. Lastly, let $\Delta E^{(k)} := E^* - E^{(k)}$. Then,

$$\|\Phi^{(k+1)} - \Phi^*\|_\infty^2 \leq \frac{r|\lambda_q^*|}{2^{k-1}\sigma^2(|\lambda_q^*| - \epsilon_1 - \epsilon_2)}.$$

Proof. Note that we have the following relationships. For notational convenience, we shall generally omit denoting the kth iterate unless necessary for clarity.

- $K(\Phi) = K_\Phi = V\Lambda V^T$ (eigen-decomposition of kernel matrix).
- $\Theta = \Theta_\Phi = \Lambda^{\frac{1}{2}} V^T$ (embedding of the low-rank component using the specified kernel).

Bound the ℓ_∞ norm with the Frobenious norm,

$$\|\Phi^{(k+1)} - \Phi^*\|_\infty^2 \leq \|\Phi^{(k+1)} - \Phi^*\|_F^2.$$

Express the low-rank components as function of their embeddings,

$$\|\Phi - \Phi^*\|_F^2 = \|\Theta^{-1}(\Theta_\Phi) - \Theta^{-1}(\Theta_{\Phi^*})\|_F^2.$$

Since the rank of Θ_Φ is smaller than the original dimension, Θ^{-1} is a perfect embedding (in that all distances are preserved) and we can bound the norm by,

$$\|\Theta^{-1}(\Theta_\Phi) - \Theta^{-1}(\Theta_{\Phi^*})\|_F^2 \leq C_1 \|\Theta_\Phi - \Theta_{\Phi^*}\|_F^2.$$

Expressing the embeddings as their eigen-decompositions and then bound the norm using the boundedness of the spectrum,

$$\begin{aligned} C_1\|\Theta_\Phi - \Theta_{\Phi^*}\|_F^2 = C_1\|\Lambda^{\frac{1}{2}}V^T - (\Lambda^{\frac{1}{2}}V^T)^*\|_F^2 &\leq C_1\|\Lambda^{\frac{1}{2}}V^T - (\Lambda^{\frac{1}{2}}V^T)^*\|_F^2 \\ &\leq C_1\|\Lambda^{\frac{1}{2}}(V_*^T - \Lambda^{\frac{\dagger}{2}}\Lambda_*^{\frac{1}{2}}V_*^T)\|_F^2 \\ &\leq C_1 r\|V_*^T - \Lambda^{\frac{\dagger}{2}}\Lambda_*^{\frac{1}{2}}V_*^T + (V^T - V_*^T)\|_F^2. \end{aligned}$$

Let $D = \Lambda^{\frac{\dagger}{2}}\Lambda_*^{\frac{1}{2}}$.

$$\begin{aligned} C_1\|\Theta_\Phi - \Theta_{\Phi^*}\|_\infty^2 &\leq C_1 r\|(I-D)V_*^T + (V^T - V_*^T)\|_F^2 \\ &\leq C_1 r\|(I-D)V_*^T\|_F^2 + C_1 r\|(V^T - V_*^T)\|_F^2. \end{aligned}$$

We shall now bound the two terms separately.

$$\begin{aligned} \|(I-D)V_*^T\|_F^2 &\leq \|(I-D)\|_F^2\|V_*^T\|_F^2 \\ &\leq \sqrt{r}\|(I-D)\|_F^2 \end{aligned}$$

From Lemma 8.6.9 we have that the difference in spectrum of the kth iterate and the optimal approach 0 as k increases. So from this, the ratio of the spectra converge to 1 (or I when written as a matrix). Bounding the second norm requires some perturbation theory from [82].

$$\begin{aligned}\|(V^T - V_*^T)\|_F^2 \leq \cdots &\leq \frac{\|K_{\Phi^*}\|_F^2|\lambda_q^*|}{2^{k-1}\sigma^2(|\lambda_q^*| - \epsilon_1 - \epsilon_2)} \\ &\leq \frac{r|\lambda_q^*|}{2^{k-1}\sigma^2(|\lambda_q^*| - \epsilon_1 - \epsilon_2)}.\end{aligned}$$

■

Bibliography

[1] Agaian, S., Sarukhanyan, H., Egiazarian, K., and Astola, J. (2011). Hadamard transforms. SPIE.

[2] Arias, P., Randall, G., and Sapiro, G. (2007). Connecting the out-of-sample and pre-image problems in kernel methods. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

[3] Baraniuk, R. (2007). Compressive sensing. *IEEE signal processing magazine*, 24(4).

[4] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396.

[5] Benedetto, J. (1992). Irregular sampling and frames. *wavelets: A Tutorial in Theory and Applications*, 2:445–507.

[6] Benedetto, J. and Fickus, M. (2003). Finite normalized tight frames. *Advances in Computational Mathematics*, 18(2-4):357–385.

[7] Benedetto, J. and Wu, H. (2000). Nonuniform sampling and spiral mri reconstruction. In *International Symposium on Optical Science and Technology*, pages 130–141. International Society for Optics and Photonics.

[8] Benedetto, J., Yilmaz, Ö., and Powell, A. (2004). Sigma-delta quantization and finite frames. In *ICASSP (3)*, pages 937–940. Citeseer.

[9] Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J., Vincent, P., and Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, 16(10):2197–2219.

[10] Bernstein, D. (2009). *Matrix mathematics: theory, facts, and formulas.* Princeton University Press.

[11] Bertsimas, D. and Tsitsiklis, J. (1997). *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA.

[12] Bonnet, N., Brun, N., and Colliex, C. (1999). Extracting information from sequences of spatially resolved eels spectra using multivariate statistical analysis. *Ultramicroscopy*, 77(3):97–112.

[13] Bosman, M., Watanabe, M., Alexander, D., and Keast, V. (2006). Mapping chemical and bonding information using multivariate analysis of electron energy-loss spectrum images. *Ultramicroscopy*, 106(11):1024–1032.

[14] Cahill, J. and Chen, X. (2013). A note on scalable frames. In *10th international conference on Sampling Theory and Applications (SampTA 2013)*, pages 93–96, Bremen, Germany.

[15] Cai, J., Candès, E., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

[16] Candès, E., Li, X., Ma, Y., and Wright, J. (2010). Robust principal component analysis?: Recovering low-rank matrices from sparse errors. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE*, pages 201–204. IEEE.

[17] Candès, E., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11.

[18] Candès, E. and Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969.

[19] Candès, E., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509.

[20] Candès, E., Wakin, M., and Boyd, S. (2008). Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905.

[21] Casazza, P. and Chen, X. (2015). Frame scalings: a condition number approach. *arXiv preprint arXiv:1510.01653*.

[22] Casazza, P., Kutyniok, G., and Philipp, F. (2013). Introduction to finite frame theory. In *Finite Frames*, pages 1–53. Springer.

[23] Chan, A., Domagalski, R., Kim, Y., Narayan, S., Suh, H., and Zhang, X. (2015). Minimal scalings and structural properties of scalable frames. *arXiv preprint arXiv:1508.02266*.

[24] Chen, D., Batson, R., and Dang, Y. (2010). *Applied integer programming: modeling and solution*. John Wiley & Sons.

[25] Chen, K. (2005). *Matrix preconditioning techniques and applications*, volume 19. Cambridge University Press.

[26] Chen, S., Donoho, D., and Saunders, M. (2001). Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159.

[27] Christensen, O. (2013). *An introduction to frames and Riesz bases.* Springer Science & Business Media.

[28] Chung, F. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.

[29] Clark, C. and Okoudjou, K. (2015). On optimal frame conditioners. In *Sampling Theory and Applications (SampTA)*, pages 148–152. IEEE.

[30] Daubechies, I., Grossmann, A., and Meyer, Y. (1986). Painless nonorthogonal expansions. *Journal of Mathematical Physics*, 27(5):1271–1283.

[31] De la Torre, F. and Black, M. (2001). Robust principal component analysis for computer vision. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 362–369. IEEE.

[32] Domagalski, R., Kim, Y., and Narayan, S. (2015). On minimal scalings of scalable frames. In *Sampling Theory and Applications (SampTA), 2015 International Conference on*, pages 91–95. IEEE.

[33] Duffin, R. and Schaeffer, A. (1952). A class of nonharmonic fourier series. *Transactions of the American Mathematical Society*, 72(2):341–366.

[34] Eckstein, J. (2012). Augmented lagrangian and alternating direction methods for convex optimization: a tutorial and some illustrative computational results. Technical report, Rutgers University.

[35] Egerton, R. (1975). Inelastic scattering of 80 kev electrons in amorphous carbon. *Philosophical Magazine*, 31(1):199–215.

[36] Egerton, R. (1979). K-shell ionization cross-sections for use in microanalysis. *Ultramicroscopy*, 4(2):169–179.

[37] Egerton, R. and Malac, M. (2002). Improved background-fitting algorithms for ionization edges in electron energy-loss spectra. *Ultramicroscopy*, 92(2):47–56.

[38] Ehler, M. and Okoudjou, K. (2012). Minimization of the probabilistic p-frame potential. *Journal of Statistical Planning and Inference*, 142(3):645–659.

[39] Elad, M. (2010). *Sparse and redundant representations: from theory to applications in signal and image processing.* Springer-Verlag New York.

[40] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

[41] Garey, M. and Johnson, D. (1979). Computers and intractability: a guide to the theory of np-completeness. *San Francisco, LA: Freeman.*

[42] Golub, G. and Van Loan, C. (2012). *Matrix computations*, volume 3. JHU Press.

[43] Gottlieb, D., Orszag, S., and MA, C. H. I. (1977). *Numerical analysis of spectral methods*, volume 2. SIAM.

[44] Gröchenig, K. (1993). Acceleration of the frame algorithm. *Signal Processing, IEEE Transactions on*, 41(12):3331–3340.

[45] Gubernatis, J. and Booth, T. (2008). Multiple extremal eigenpairs by the power method. *Journal of Computational Physics*, 227(19):8508–8522.

[46] Ham, J., Lee, D., Mika, S., and Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47. ACM.

[47] Honeine, P. and Richard, C. (2009). Solving the pre-image problem in kernel machines: A direct method. In *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on*, pages 1–6. IEEE.

[48] Horn, R. and Johnson, C. (2012). *Matrix analysis*. Cambridge university press.

[49] Hubert, M., Rousseeuw, P., and Vanden Branden, K. (2005). Robpca: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79.

[50] Hunt, J. and Williams, D. (1991). Electron energy-loss spectrum-imaging. *Ultramicroscopy*, 38(1):47–73.

[51] Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.

[52] Karhunen, J. and Joutsensalo, J. (1995). Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549–562.

[53] Kruskal, J. and Wish, M. (1978). *Multidimensional scaling*, volume 11. Sage.

[54] Kutyniok, G., Okoudjou, K., and Philipp, F. Scalable Frames and Convex Geometry. *Contemp. Math. 626 (2014), 19-32.*

[55] Kutyniok, G., Okoudjou, K., and Philipp, F. (2013a). Preconditioning of frames. In *SPIE Optical Engineering+ Applications*, pages 88580G–88580G. International Society for Optics and Photonics.

[56] Kutyniok, G., Okoudjou, K., Philipp, F., and Tuley, E. (2013b). Scalable frames. *Linear Algebra and its Applications*, 438(5):2225 – 2238.

[57] Kwok, J. and Tsang, I. (2003). The pre-image problem in kernel methods. In *ICML*, pages 408–415.

[58] Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110.

[59] Lanczos, C. (1950). *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.* United States Governm. Press Office Los Angeles, CA.

[60] Leapman, R., Hunt, J., Buchanan, R., and Andrews, S. (1993). Measurement of low calcium concentrations in cryosectioned cells by parallel-eels mapping. *Ultramicroscopy*, 49(1):225–234.

[61] Leapman, R. and Newbury, D. (1993). Trace elemental analysis at nanometer spatial resolution by parallel-detection electron energy loss spectroscopy. *Analytical chemistry*, 65(18):2409–2414.

[62] Leapman, R. and Swyt, C. (1988). Separation of overlapping core edges in electron energy loss spectra by multiple-least-squares fitting. *Ultramicroscopy*, 26(4):393–403.

[63] Lee, J. and Verleysen, M. (2007). *Nonlinear dimensionality reduction.* Springer Science & Business Media.

[64] Lin, Z., Chen, M., and Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055.*

[65] Lin, Z., Liu, R., and Su, Z. (2011). Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in neural information processing systems*, pages 612–620.

[66] Mika, S., Schölkopf, B., Smola, A., Müller, K., Scholz, M., and Rätsch, G. (1998). Kernel pca and de-noising in feature spaces. In *NIPS*, volume 4, page 7. Citeseer.

[67] Mises, R. and Pollaczek-Geiringer, H. (1929). Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164.

[68] Netrapalli, P., Niranjan, U., Sanghavi, S., Anandkumar, A., and Jain, P. (2014). Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pages 1107–1115.

[69] Okoudjou, K. (2016 (to appear)). Preconditioning techniques in frame theory and probabilistic frames. In *Finite Frame Theory: A Complete Introduction to Overcompleteness.* AMS.

[70] O'Leary, D. (2009). *Scientific computing with case studies.* SIAM.

[71] Panju, M. (2011). Iterative methods for computing eigenvalues and eigenvectors. *The Waterloo Mathematics Review*, 1(1):9–18.

[72] Saul, L. and Roweis, S. (2000). An introduction to locally linear embedding. Technical report.

[73] Schölkopf, B., Knirsch, P., Smola, A., and Burges, C. (1998). Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. In *Mustererkennung 1998*, pages 125–132. Springer.

[74] Schölkopf, B., Smola, A., and Müller, K. (1997). Kernel principal component analysis. In *Artificial Neural NetworksICANN'97*, pages 583–588. Springer.

[75] Shahid, N., Kalofolias, V., Bresson, X., Bronstein, M., and Vandergheynst, P. (2015). Robust principal component analysis on graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2812–2820.

[76] Skopenkov, A. (2015). A short elementary proof of the ruffini-abel theorem. *arXiv preprint arXiv:1508.03317*.

[77] Smith, R. (2012). Introduction to hyperspectral imaging, microimages inc. *Mentor, OH*.

[78] Smola, A. and Schölkopf, B. (1998). *Learning with kernels*. Citeseer.

[79] Snyder, J., Mika, S., Burke, K., and Müller, K. (2013). Kernels, pre-images and optimization. In *Empirical Inference*, pages 245–259. Springer.

[80] Spielman, D. and Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926.

[81] Spielman, D. and Teng, S. (2004). Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90. ACM.

[82] Stewart, G. (1973). Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM review*, 15(4):727–764.

[83] Stewart, G. (1993). Afternotes on numerical analysis. *University of Maryland at College Park*.

[84] Strang, G. and Press, W. (1993). *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA.

[85] Stroock, D. (2010). *Probability theory: an analytic view*. Cambridge university press.

[86] Tenenbaum, J., De Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

[87] Theodoridis, S. and Koutroumbas, K. (2003). Pattern recognition.

[88] Trebbia, P. and Bonnet, N. (1990). Eels elemental mapping with unconventional methods i. theoretical basis: Image analysis with multivariate statistics and entropy concepts. *Ultramicroscopy*, 34(3):165–178.

[89] Tropp, J. and Gilbert, A. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666.

[90] Verbeeck, J. and Van Aert, S. (2004). Model based quantification of eels spectra. *Ultramicroscopy*, 101(2):207–224.

[91] Veselov, S. and Chirkov, A. (2009). Integer program with bimodular matrix. *Discrete Optimization*, 6(2):220–222.

[92] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

[93] Wang, Y., Ho, R., Shao, Z., and Somlyo, A. (1992). Optimization of quantitative electron energy loss spectroscopy in the low loss region: phosphorus l-edge. *Ultramicroscopy*, 41(1):11–31.

[94] Wasserman, L. (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

[95] Wolsey, L. (1998). *Integer programming*, volume 42. Wiley New York.

[96] Wright, J., Ganesh, A., Rao, S., Peng, Y., and Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088.

[97] Xu, L. and Yuille, A. (1995). Robust principal component analysis by self-organizing rules based on statistical physics approach. *Neural Networks, IEEE Transactions on*, 6(1):131–143.

[98] Yang, J. and Zhang, Y. (2011). Alternating direction algorithms for *ell*_1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1):250–278.