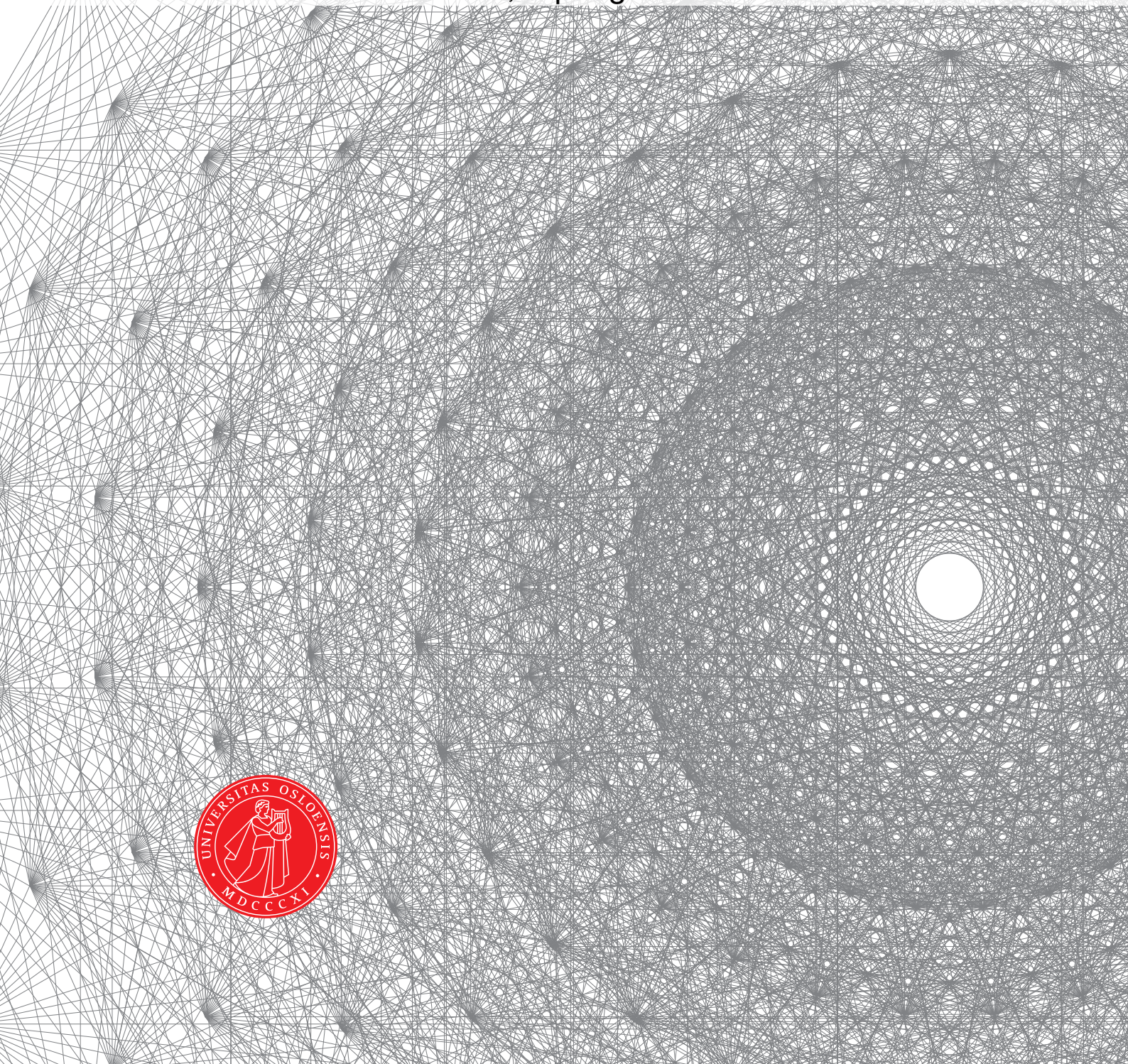
UiO : Department of Mathematics
University of Oslo

Compressed Sensing and the Quadratic Bottleneck Problem:

A Combinatorics Approach

Kevin Sheehan

Master's Thesis, Spring 2018



This master's thesis is submitted under the master's programme *Computational Science and Engineering*, with programme option *Computational Science*, at the Department of Mathematics, University of Oslo. The scope of the thesis is 60 credits.

The front page depicts a section of the root system of the exceptional Lie group E_8, projected into the plane. Lie groups were invented by the Norwegian mathematician Sophus Lie (1842–1899) to express symmetries in differential equations and today they play a central role in various parts of mathematics.

Abstract

Compressed sensing is the study of solving underdetermined systems of linear equations with unique sparse solutions. In addition to this, applications of compressed sensing require that the number of rows of the measurement matrix is minimized.

In general, the entries of a measurement matrix can be complex. A combinatorial measurement matrix narrows this down to just zeros and ones. These measurement matrices may be obtained from other objects such as the incidence matrix of a combinatorial design or the bipartite adjacency matrix of a bipartite graph.

In many applications, a measurement matrix is normally a randomly constructed matrix. This is because it has been shown that with high probability, certain classes of random measurement matrices have on the order of the optimal number of rows required for sparse reconstruction. Finding deterministically constructed classes of measurement matrices whose number of rows scale on the same order as classes of random measurement matrices has been an open problem for at least a decade. This problem is referred to as the *quadratic bottleneck problem.*

The quadratic bottleneck problem is important for several different reasons. A specific randomly generated measurement matrix cannot be checked if it does indeed recover every sparse vector uniquely. An engineer developing a sensor system might find it undesirable or not possible to implement a physical system that can take random measurements. Recovery algorithms may be developed for a specific class of deterministically constructed measurement matrices. Or, it could just be considered a proof of concept that it can be done.

The goal of this thesis is to define the sufficient compressed sensing theory, graph theory, and combinatorial design theory to attempt an answer at the quadratic bottleneck problem with combinatorial measurement matrices. After this, an attempt to define a solution to the quadratic bottleneck problem and describe a valid argument that could be used to demonstrate this is made. Finally, a selection of common compressed sensing recovery algorithms are selected and optimized for these explicit constructions.

Acknowledgements

The completion of this master thesis would not have been possible without the feedback and guidance of my advisor Øyvind Ryan, and Head of the Departement of Mathematics Geir Dahl who pointed me towards compressed sensing. I am grateful for and would like to thank Anders for his research into this field. I would also like to thank my fellow master students for our discussions. A special thanks to my mother Lisa Baker for always being there for me, your continued support means a lot to me. Ida, thank you for being the best partner I could hope for. Lastly I would like to thank my friends, Chance, Dino, Pfluger, Richard, John and Mikael, and my brothers Patrick and Kyle, for your encouragements through this process.

Contents

Part I

The General Approach to Compressed Sensing and the Quadratic Bottleneck Problem

1 The Compressed Sensing Problem and its Variations

1.1 Introduction

Compressed sensing is the study of solving underdetermined linear systems of equations with unique sparse solutions. More precisely, consider a matrix $A \in \mathbb{C}^{m \times N}$ with $m < N$ and a sparse vector $\mathbf{x} \in \mathbb{C}^N$. That is, a vector where most of the entries are zero. Define the vector $\mathbf{y} = A\mathbf{x} \in \mathbb{C}^m$. Then the system of linear equations $A\mathbf{z} = \mathbf{y}$ is underdetermined and consistent. Thus, an infinite number of solutions exist and uniquely recovering $\mathbf{x}$ from the matrix A and the vector $\mathbf{y}$ seems impractical. However, in some situations the sparsity assumption of $\mathbf{x}$ allows us to efficiently obtain $\mathbf{x}$ as the only sparse solution of $A\mathbf{z} = \mathbf{y}$. These situations are the study of compressed sensing.

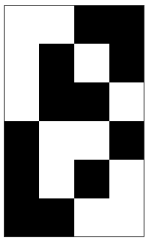
Applications. The main application of compressed sensing is as a signals processing technique. In this setting the measurement matrix A corresponds to a measurement scheme of a signal $\mathbf{x}$, where the signal is sparse in some basis. The measurement vector $\mathbf{y}$ contains the measurements obtained from the measurement scheme. We will only focus on the mathematical problems of compressed sensing. Questions on whether an engineer can design a sensor that achieves a measurement scheme which corresponds to the measurement matrix A are important, but will not be addressed here.

Notation. We first start off with some notation that will be used throughout this thesis. Let $A \in \mathbb{C}^{m \times N}$ be a matrix and $\mathbf{z} \in \mathbb{C}^N$. Whenever we refer to a matrix A as a *measurement matrix*, we assume that $m < N$. The column vectors of A will be denoted by $\mathbf{a}_1, \dots, \mathbf{a}_N$. We denote $[N]$ as the set of integers $\{1, \dots, N\}$. Let $S \subset [N]$. Then denote A_S by the submatrix of A with the columns of A indexed by the set S and denote $\mathbf{z}_S$ by the subvector of $\mathbf{z}$ with the entries of $\mathbf{z}$ indexed by the set S. In some situations, we zero out the other entries rather then delete them. It should be clear from the context which meaning of A_S and $\mathbf{z}_S$ is used. Since in the second part of this thesis we use binary matrices, it helps to have a way to easily visualize a binary matrix. We do this by creating a corresponding black and white image to the binary matrix

in the following way. Consider the matrix

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

If we let the ones correspond to white rectangles and the zeros correspond to black rectangles, then we will denote this matrix with the following image.



If B is a square matrix of order n, then we denote the eigenvalues of B by $\lambda_1, \ldots, \lambda_n$. If B is a normal matrix, then we order the eigenvalues by

$$\lambda_{min} := \lambda_1 \leq \ldots \leq \lambda_n =: \lambda_{max}$$

Format To avoid getting distracted with smaller lemmas that we use in proofs to theorems more relevant to this thesis, we will only cite where to find the proof of the lemma. This is done to keep the thesis moving at a decent pace. Since the focus of this thesis is compressed sensing, we aim to do a full survey of the foundational results in this theory. When we do introductions to graph theory, combinatorial design theory, and numerical linear algebra, we will only introduce the relevant material to this thesis. Summary sections will be used at the end of each part to describe in more detail the references that were used and how they were used, but it will not act as a history of the progression of the theory. In addition to this, when a theorem is used which was obtained from a different author, the citation will be placed in parentheses as well as the where to locate it in their work.

1.2 Sparsity

First, we make the notion of a sparse vector precise by defining a measure for how sparse it is.

Definition 1 (support)**.** The *support* of a vector $\mathbf{x} \in \mathbb{C}^N$, denoted $\text{supp}(\mathbf{x})$, is the set of indices of the nonzero entries of $\mathbf{x}$.

Definition 2 (ℓ_0-norm)**.** The *ℓ_0-norm* of a vector $\mathbf{x} \in \mathbb{C}^N$ is defined to be $\| \mathbf{x} \|_0 := |\text{supp}(\mathbf{x})|$, the number of nonzero entries of $\mathbf{x}$.

The ℓ_0-norm is not actually a norm. To see this, consider an arbitrary scalar a and arbitrary vector $\mathbf{x}$. Then $\| a\mathbf{x} \|_0 = \| \mathbf{x} \|_0 \neq |a| \| \mathbf{x} \|_0$ in general. The reason for the name and notation of the ℓ_0-norm follows from taking the limit of the p-norm as p approaches zero. We have

$$\lim_{p \to 0} \| \mathbf{x} \|_p^p = \lim_{p \to 0} \sum_{j=1}^{N} |x_j|^p = \sum_{x_j \neq 0} \mathbf{1}_j = | \ 1 \ | = \| \mathbf{x} \|_0^0 .$$

Definition 3 (Sparsity)**.** Let $s \in [N]$. Then the vector $\mathbf{x} \in \mathbb{C}^N$ is *s-sparse* if $\| \mathbf{x} \|_0 \leq s$.

Using this measure, we can formally define the compressed sensing problem.

Problem 1 (Compressed Sensing)**.** Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix. For any s-sparse $\mathbf{x} \in \mathbb{C}^N$, does there exist an algorithm such that $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = \mathbf{y}$, where $\mathbf{y} = A\mathbf{x}$ is the measurement vector of $\mathbf{x}$?

1.3 Compressible Vectors

In many applications, the vectors to be recovered are not sparse but are "almost" sparse. These almost sparse vectors are called *compressible vectors* and are characterized by having a small number of entries that have relatively large absolute values compared to the rest of the entries. Thus, compressible vectors can be related to sparse vectors by measuring its minimum distance to a closest sparse vector, where the distance is measured by any p-norm.

Definition 4 (Compressible Vectors)**.** Let $p > 0$, $\mathbf{x} \in \mathbb{C}^N$, and $s \in [N]$. Then the *ℓ_p-error of best s-term approximation*, denoted by $\sigma_s(\mathbf{x})_p$, is defined by

$$\sigma_s(\mathbf{x})_p = \inf\{ \| \mathbf{x} - \mathbf{z} \|_p \mid \mathbf{z} \in \mathbb{C}^N \text{is } s\text{-sparse} \}.$$

By choosing $\mathbf{z} = \mathbf{x}_S$, where S is an index set of the s largest absolute values of $\mathbf{x}$, we obtain the infimum. However, if the s and $s+1$ largest absolute entries of $\mathbf{x}$ are equal, then the vector $\mathbf{z}$ is not unique. This non-uniqueness does not change the value of $\sigma_s(\mathbf{x})_p$. Using this notion of compressible vectors, we can define the slightly more complicated problem of compressed sensing with compressible vectors.

Problem 2 (Compressed Sensing with Compressible Vectors)**.** Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix and let b be a positive real number. For any compressible vector $\mathbf{x} \in \mathbb{C}^N$, does there exist an algorithm such that every s-sparse solution $\mathbf{x}^\sharp$ of $A\mathbf{z} = \mathbf{y}$ satisfies $\left\| \mathbf{x} - \mathbf{x}^\sharp \right\|_1 \leq b$, where $\mathbf{y} = A\mathbf{x}$ is the measurement vector of $\mathbf{x}$?

We quickly observe that a compressible vector can be a sparse vector. To see this, observe that if $\sigma_s(\mathbf{x})_p = 0$, then $\| \mathbf{x} - \mathbf{z} \|_p = 0$ where $\mathbf{z}$ is s-sparse. Therefore $\mathbf{x} = \mathbf{z}$, so $\mathbf{x}$ is s-sparse. This shows that the compressed sensing problem with compressible vectors is the compressed sensing problem with sparse vectors if $\sigma_s(\mathbf{x})_p = 0$.

Since a best s-term approximation to compressible vector $\mathbf{x}$ is not unique, we cannot guarantee that a solution to the compressed sensing problem with

compressible vectors is unique. However, if we can find a small upperbound b on the ℓ_1-error, then this is not to much of a problem.

1.4 Measurement Error

In many applications, exact measurement precision of a sparse or compressible signal is unlikely. When precision is not exact, we have measurement error.

Definition 5 (Measurement Error)**.** Let $\mathbf{x} \in \mathbb{C}^N$ be a compressible vector and let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix. Then the measurement vector $\mathbf{y}$ has a *measurement error* $\mathbf{e} \in \mathbb{C}^m$ if $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ and $\| \mathbf{e} \| \leq \eta$ for some $\eta > 0$.

If we assume we have measurement error when measuring compressible vectors, then we arrive at the following version of the problem.

Problem 3 (Compressed sensing with Compressible Vectors and Measurement Error)**.** Let $A \in C^{m \times N}$ be a measurement matrix and let b be a positive real number. For any compressible vector $\mathbf{x} \in \mathbb{C}^N$, does there exist an algorithm such that every s-sparse solution $\mathbf{x}^\sharp$ of $A\mathbf{z} = \mathbf{y}$ satisfies $\left\| \mathbf{x} - \mathbf{x}^\sharp \right\|_1 \leq b$, where $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ is the measurement vector of $\mathbf{x}$ with measurement error $\mathbf{e}$ controlled by $\| \mathbf{e} \| \leq \eta$?

Because of the ubiquity of measurement error in applications, the compressed sensing problem with compressible vectors and measurement error is the most realistic variation.

As a last note, observe that if $\eta = 0$, then $\| \mathbf{e} \| \leq 0$. This implies $\mathbf{e} = \mathbf{0}$ which gives us $\mathbf{y} = A\mathbf{x} + \mathbf{e} = A\mathbf{x}$. This shows that the compressed sensing problem with measurement error $\eta = 0$ is the compressed sensing problem with compressible vectors. The connections between the versions of the compressed sensing problem provide us with some intuition that each of these problems are handled in a similar way. In fact, we will see in future sections that this is indeed true.

2 ℓ_0-minimization, Basis Pursuit, and Other Approximation Methods

In this chapter we discuss two approaches to solving the compressed sensing problems defined in the first chapter. The first section discusses a naive approach and the second section discusses an approach that is widely applied in compressed sensing.

2.1 ℓ_0-Minimization

A first approach to solving the compressed sensing problem usually begins with considering the ℓ_0-minimization problem.

$$\underset{\mathbf{z}\in\mathbb{C}^N}{\text{minimize}} \| \mathbf{z} \|_0 \text{ s.t. } A\mathbf{z} = \mathbf{y} \tag{P_0}$$

The solution of the combinatorial optimization problem (P_0) is related to the solution of the compressed sensing problem in the following way.

Theorem 1 ([18]). *Let $\boldsymbol{x} \in \mathbb{C}^N$ be an s-sparse vector. Let A be a measurement matrix and $\boldsymbol{y} = A\boldsymbol{x}$ be the measurement vector of $\boldsymbol{x}$. Then $\boldsymbol{x}$ is the unique s-sparse solution of the compressed sensing problem if and only if $\boldsymbol{x}$ is the unique s-sparse solution of (P_0).*

Proof. First, suppose $\mathbf{x}$ is the unique s-sparse solution of the compressed sensing problem. That is, $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = \mathbf{y}$. Let $\mathbf{x}^\sharp$ be a solution of (P_0). Then $A\mathbf{x}^\sharp = \mathbf{y}$ and $\mathbf{x}^\sharp$ is s-sparse since $\| \mathbf{x}^\sharp \|_0 \leq \| \mathbf{x} \|_0$. But by the assumption, the uniqueness of $\mathbf{x}$ as an s-sparse solution implies $\mathbf{x}^\sharp = \mathbf{x}$. Therefore $\mathbf{x}$ is the unique solution to (P_0).

Next, suppose $\mathbf{x}^\sharp$ is the unique solution of (P_0). Then $\mathbf{x}^\sharp$ is the unique s-sparse solution to $A\mathbf{z} = \mathbf{y}$. Therefore $\mathbf{x}^\sharp$ is the unique solution to the compressed sensing problem. □

By theorem 1, to compute the solution of the compressed sensing problem, it suffices to find an efficient algorithm for computing the combinatorial optimization problem (P_0). We start by naively trying to find the sparsest solution by solving every system $A_S\mathbf{z} = \mathbf{y}$ over every subset $S \subset [N]$ with $|S| \leq s$. If we find an S such that this system of linear equations is consistent, then we have obtained the correct solution. However, this is not a practical approach for large N and large s since this requires us to solve $\binom{N}{s}$ systems of linear

equations. This naive approach raises concerns over the complexity of (P_0). Indeed, it is known from [25] that (P_0) is NP-Hard.

There are several known equivalent conditions to the solution of the compressed sensing problems via (P_0) which are summarized in the following theorem. We will occasionally use these later in the thesis.

Theorem 2 ([18], Theorem 2.13)**.** *Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix and $\boldsymbol{x}$ be an s-sparse vector. Let $\boldsymbol{y} = A\boldsymbol{x}$ be the measurement vector of $\boldsymbol{x}$. Then the following are equivalent.*

1. *$\boldsymbol{x}$ is the unique s-sparse solution of $A\boldsymbol{z} = \boldsymbol{y}$.*
2. *The only $2s$-sparse vector in the null space $\mathcal{N}(A)$ of A is the zero vector.*
3. *For any $S \subset [N]$ with $|S| \leq 2s$, the matrix A_S is an injective matrix.*
4. *Every $2s$-column vectors of A is a set of linearly independent columns.*

Proof. $(1 \Rightarrow 2)$. Suppose $\mathbf{x}$ is the unique s-sparse solution to the underdetermined system $A\mathbf{z} = \mathbf{y}$. Let $\mathbf{v} \in \mathcal{N}(A)$ be a $2s$-sparse vector with $\mathbf{v} = \mathbf{x} - \mathbf{z}$ for some s-sparse vector $\mathbf{z}$ with a disjoint support set to $\mathbf{x}$. This implies

$$A\mathbf{v} = A(\mathbf{x} - \mathbf{z}) = A\mathbf{x} - A\mathbf{z} = \mathbf{0}$$

and it follows from the last equality that $A\mathbf{x} = A\mathbf{z}$. Since the supports of $\mathbf{x}$ and $\mathbf{z}$ are disjoint, it follows that $\mathbf{x} = \mathbf{z} = \mathbf{0}$. Thus, $\mathbf{v} = \mathbf{0}$ and the null space of A does not contain any other $2s$-sparse vectors other then the zero vector.

$(2 \Rightarrow 1)$. Suppose the only $2s$-sparse vector in $\mathcal{N}(A)$ is the zero vector. Let $\mathbf{z}$ be an s-sparse vector with $A\mathbf{z} = \mathbf{y} = A\mathbf{x}$. Then $A(\mathbf{z} - \mathbf{x}) = \mathbf{0}$ and $\mathbf{z} - \mathbf{x} \in \mathcal{N}(A)$. Since $\mathbf{z} - \mathbf{x}$ is $2s$-sparse, the assumption implies $\mathbf{z} - \mathbf{x} = \mathbf{0}$. Thus $\mathbf{z} = \mathbf{x}$. Therefore, $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = \mathbf{y}$.

$(2 \Rightarrow 3)$. Suppose that the only $2s$-sparse vector in $\mathcal{N}(A)$ is the zero vector. Consider an arbitrary vector $\mathbf{v}$. Let $S \subset [N]$ with $|S| \leq 2s$ be arbitrary. Then $A\mathbf{v}_S = A_S\mathbf{v}_S = \mathbf{0}$ implies $\mathbf{v}_S = \mathbf{0}$ since $\mathbf{v}_S$ is $2s$-sparse. Thus, $\mathcal{N}(A_S) = \{\mathbf{0}\}$, so A_S is injective.

$(3 \Rightarrow 4)$. Suppose that for any $S \subset [N]$ with $|S| \leq 2s$, the matrix of A_S is injective. Then $A_S\mathbf{v} = A_S\mathbf{v}_S = \mathbf{0}$ if and only if $\mathbf{v}_S = \mathbf{0}$. Thus every $2s$-columns of A are linearly independent.

$(4 \Rightarrow 2)$. Suppose every $2s$-columns of A are linearly independent. Let $S \subset [N]$ be an arbitrary subset with $|S| \leq 2s$. Then $A\mathbf{v}_S = A_S\mathbf{v}_S = \mathbf{0}$ if and only if $\mathbf{v} = \mathbf{0}$. Thus, the only $2s$-sparse vector in $\mathcal{N}(A)$ is the zero vector. □

Even though these equivalent conditions are nice to have for theoretical purposes, they do not provide a tractable approach of solving the compressed sensing problem. This is because each of these equivalent conditions involves doing $\binom{N}{2s}$ checks, which is not practical.

The following definition is very similar to the conditions of the previous theorem. This will allow us to put upperbounds on the sparsity s of all of the equivalent conditions in the previous theorem.

Definition 6 (Spark)**.** Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix. Then the *spark* of A, denoted $\text{spark}(A)$ is defined by

$$\text{spark}(A) := \min\{ \| \mathbf{x} \|_0 \mid A\mathbf{x} = \mathbf{0} \text{ with } \mathbf{x} \neq \mathbf{0} \}.$$

Notice that conditions $2, 3$, and 4 in theorem 2 hold if and only if

$$\text{spark}(A) > 2s.$$

2.2 Basis Pursuit

Since solving the problem (P_0) is NP-hard, we look for a way to relax the problem so we can obtain an approximate solution to (P_0). This can be handled in a variety of ways. Perhaps the most known approximation method is ℓ_1-minimization, which is defined as

$$\underset{\mathbf{z}\in\mathbb{C}^N}{\text{minimize}} \| \mathbf{z} \|_1 \text{ s.t. } A\mathbf{z} = \mathbf{y}. \tag{P_1}$$

This approach is normally referred to as *basis pursuit*. At this point, it is not clear that basis pursuit does indeed provide a good approximation for (P_0). We clear this up in next chapter, where we will prove a necessary and sufficient condition on the measurement matrix. When measurement error is involved, basis pursuit is insufficient. Because of this, we introduce a variation of basis pursuit called *quadratically constrained basis pursuit*, which is defined by

$$\underset{\mathbf{z}\in\mathbb{C}^N}{\text{minimize}} \| \mathbf{z} \|_1 \text{ s.t. } \| A\mathbf{z} - \mathbf{y} \|_2 \leq \eta. \tag{$P_{1,\eta}$}$$

Basis pursuit and quadratically constrained basis pursuit are not algorithms. However, efficient algorithms that solve these problems exist and are classified as *optimization algorithms*. In many cases, linear programming can be used as well as other optimized algorithms such as the *homotopy method* (or the *least-angle regression algorithm* [15]) and the *iteratively reweighted least squares algorithm* [12]. Many of the theorems throughout parts 1 and 2 of this thesis are developed with the intent of answering the compressed sensing problems with algorithms that solve (P_1) and $(P_{1,\eta})$.

Other Approximation Algorithms Other common families of approximation algorithms for (P_0) can be classified as *combinatorial algorithms*, *greedy algorithms*, and *iterative thresholding algorithms*. A selection of algorithms will be presented in part 3.

3 The Null Space Properties

In the first chapter, we claimed that the primary focus of compressed sensing is situations where the sparsity assumption allows us to efficiently obtain $\mathbf{x}$ as the only s-sparse solution. We make this vague notion clear in this chapter by introducing on a necessary and sufficient condition for basis pursuit and quadratically constrained basis pursuit called the null space property.

3.1 The Null Space Property

Definition 7 (Null Space Property)**.** A measurement matrix $A \in \mathbb{C}^{m\times N}$ satisfies the *null space property of order* s if, for all $S \subset [N]$ with $|S| \leq s$,

$$\| \mathbf{v}_S \|_1 < \| \mathbf{v}_{\bar{S}} \|_1 \text{ for all } \mathbf{v} \in \mathcal{N}(A) \setminus \{0\}.$$

The fact that the null space property of a measurement matrix is a necessary and sufficient condition for a unique solution to the compressed sensing problem is proved in theorem 3.

Theorem 3 ([18], Theorem 4.4)**.** *Let $A \in \mathbb{C}^{m\times N}$ be measurement matrix. Then every s-sparse vector $\boldsymbol{x} \in \mathbb{C}^N$ is the unique solution of*

$$\underset{\boldsymbol{z}\in\mathbb{C}^N}{\text{minimize}} \| \boldsymbol{z} \|_1 \;\; s.t. \;\; A\boldsymbol{z} = \boldsymbol{y}, \tag{P_1}$$

where $\boldsymbol{y} = A\boldsymbol{x}$ is the measurement vector, if and only if A satisfies the null space property of order s.

Proof. Suppose every s-sparse vector $\mathbf{x} \in \mathbb{C}^N$ is the unique solution of (P_1) with the measurement vector $\mathbf{y} = A\mathbf{x}$. Let $\mathbf{v} \in \mathcal{N}(A) \setminus \{\mathbf{0}\}$ be an arbitrary vector and let S be an arbitrary s-subset of $[N]$. Then $\mathbf{v}_S$ is the unique solution of (P_1) with the measurement vector $\mathbf{y} = A\mathbf{v}$. That is,

$$\| \mathbf{v}_S \|_1 < \| \mathbf{z} \|_1 \text{ for all } \mathbf{z} \in \mathbb{C}^N \setminus \{\mathbf{v}_S\} \text{ with } A\mathbf{z} = A\mathbf{v}_S.$$

Since $A(\mathbf{v}_S + \mathbf{v}_{\bar{S}}) = A\mathbf{v} = \mathbf{0}$ implies $A(-\mathbf{v}_{\bar{S}}) = A\mathbf{v}_S$, the assumption gives us $\| \mathbf{v}_S \|_1 \leq \| \mathbf{v}_{\bar{S}} \|_1$. Since $\mathbf{v}_S \neq \mathbf{v}_{\bar{S}}$, we have $\| \mathbf{v}_S \|_1 < \| \mathbf{v}_{\bar{S}} \|_1$. Therefore, the measurement matrix A satisfies the null space property of order s.

Next, suppose A satisfies the null space property of order s. Let $\mathbf{x} \in \mathbb{C}^N$ be an s-sparse vector with $S = \text{supp}(\mathbf{x})$. Let $\mathbf{z} \in \mathbb{C}^N$ be an arbitrary vector that

satisfies $A\mathbf{z} = A\mathbf{x}$ and $\mathbf{z} \neq \mathbf{x}$. Define $\mathbf{v} := \mathbf{x} - \mathbf{z} \in \mathcal{N}(A) \setminus \{0\}$. Then we have

$$\begin{aligned} \| \mathbf{x} \|_1 &= \| \mathbf{x}_S - \mathbf{z}_S + \mathbf{z}_S \|_1 \\ &\leq \| \mathbf{x}_S - \mathbf{z}_S \|_1 + \| \mathbf{z}_S \|_1 \\ &= \| \mathbf{v}_S \|_1 + \| \mathbf{z}_S \|_1 \\ &< \| \mathbf{v}_{\bar{S}} \|_1 + \| \mathbf{z}_S \|_1 \ \text{(NSP)} \\ &= \| \mathbf{x}_{\bar{S}} - \mathbf{z}_{\bar{S}} \|_1 + \| \mathbf{z}_S \|_1 \\ &= \| \mathbf{z}_{\bar{S}} \|_1 + \| \mathbf{z}_S \|_1 \\ &= \| \mathbf{z} \|_1 . \end{aligned}$$

Therefore $\| \mathbf{x} \|_1 < \| \mathbf{z} \|_1$ for any $\mathbf{z} \in \mathbb{C}^N$, so $\mathbf{x}$ is the unique s-sparse solution of (P_1). □

Theorem 4 proves that if the measurement matrix satisfies the null space property, then the solution of the intractable problem (P_0) is approximated by the solution of basis pursuit.

Theorem 4 ([18]). *Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix that satisfies the null space property of order s. If $\boldsymbol{x}$ is s-sparse and $\boldsymbol{y} = A\boldsymbol{x}$ is the measurement vector of $\boldsymbol{x}$, then the solution of (P_1) is the solution of (P_0).*

Proof. Let $\mathbf{x}$ be an s-sparse vector and $\mathbf{y} = A\mathbf{x}$ be its measurement vector. Then the previous theorem implies that $\mathbf{x}$ is the unique s-sparse solution of (P_1) with $\mathbf{y} = A\mathbf{x}$. Let $\mathbf{z}$ be the solution to (P_0). Then $\| \mathbf{z} \|_0 \leq \| \mathbf{x} \|_0$, so this implies $\mathbf{z}$ is s-sparse as well. However, $\mathbf{x}$ is the unique s-sparse solution of (P_1), so $\mathbf{z} = \mathbf{x}$. Therefore the solution of (P_1) is the solution of (P_0). □

By taking the theorems 1 and 4 together, we have proven that if A satisfies the null space property of order s, then an algorithm solves (P_1) solves the compressed sensing problem.

3.2 The Stable Null Space Property

In chapter 1, several versions of the compressed sensing were defined. We hinted at the fact that since the problems were so closely related, they could be handled in similar ways. We now address the compressed sensing problem with compressible vectors by introducing a very similar property to the previous section.

Definition 8 (Stable Null Space Property). A measurement matrix $A \in \mathbb{C}^{m \times N}$ satisfies the *stable null space property of order s* with $0 < \rho < 1$ if, for any $S \subset [N]$ with $|S| \leq s$, we have,

$$\| \mathbf{v}_S \|_1 \leq \rho \| \mathbf{v}_{\bar{S}} \|_1 \ \text{for all } \mathbf{v} \in \mathcal{N}(A) \setminus \{\mathbf{0}\}.$$

Since the proofs for the stable null space property in the next two theorems follow immediately from the proofs in the next section, we will not dwell on this section.

Theorem 5 ([18], Theorem 4.14)**.** *Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix. Then A satisfies the stable null space property of order s with $0 < \rho < 1$ if and only if for all $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{C}^N$ with $A\boldsymbol{z} = A\boldsymbol{x}$, we have*

$$\| \boldsymbol{z} - \boldsymbol{x} \|_1 \leq \frac{1+\rho}{1-\rho}(\| \boldsymbol{z} \|_1 - \| \boldsymbol{x} \|_1 + 2 \| \boldsymbol{x}_{\bar{S}} \|_1)$$

Recall from problem 2 that when the vector to be recovered is only compressible, we want to find a small bound b on the ℓ_1-error of a solution of an algorithm. Theorem 6 provides such bound.

Theorem 6 ([18], Theorem 4.12)**.** *Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix that satisfies the stable null space property of order s with $0 < \rho < 1$. Let $\boldsymbol{x} \in \mathbb{C}^N$. Then any solution $\boldsymbol{x}^\sharp$ of (P_1) approximates $\boldsymbol{x}$ with error bound*

$$\left\| \boldsymbol{x} - \boldsymbol{x}^\sharp \right\|_1 \leq \frac{1+\rho}{1-\rho} 2\sigma_s(\boldsymbol{x})_1.$$

3.3 The Robust Null Space Property

In this final section, we introduce a property for the compressed sensing problem with compressible vectors and measurement error.

Definition 9 (Robust Null Space Property)**.** A measurement matrix $A \in \mathbb{C}^{m\times N}$ satisfies the *robust null space property of order s* with $0 < \rho < 1$, $\tau > 0$, and with respect to the norm $\| \cdot \|$ if, for all subsets $S \subset [N]$ with $|S| \leq s$, we have

$$\| \mathbf{v}_S \|_1 \leq \rho \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \| \text{ for all } \mathbf{v} \in \mathbb{C}^N.$$

The next theorem demonstrates that if we want to demonstrate that a matrix satisfies the robust null space property, then all we need to show is that the inequality holds for the s largest absolute entries of $\mathbf{v}$. This is useful for theoretical purposes, since we only have to show the inequality holds for one case rather then for all cases.

Theorem 7 ([18])**.** *Let $\boldsymbol{v} \in \mathbb{C}^N$ and let $S \subset [N]$ be the index set of the s largest absolute entries of $\boldsymbol{v}$. If the inequality*

$$\| \boldsymbol{v}_S \|_1 \leq \rho \| \boldsymbol{v}_{\bar{S}} \|_1 + \tau \| A\boldsymbol{v} \|$$

holds for some $0 < \rho < 1$ and $\tau > 0$, then A satisfies the robust null space property order s with $0 < \rho < 1$ and $\tau > 0$.

Proof. Suppose $\| \mathbf{v}_S \|_1 \leq \rho \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \|$ holds for some $0 < \rho < 1$ and $\tau > 0$. Since S is the index set of the s largest elements of $\mathbf{v}$, we have $\| \mathbf{v}_{S'} \|_1 \leq \| \mathbf{v}_S \|_1$ and $\| \mathbf{v}_{\bar{S}} \|_1 \leq \| \mathbf{v}_{\bar{S}'} \|_1$ for any other subset $S' \subset [N]$ with $|S'| \leq s$. This gives us

$$\| \mathbf{v}_{S'} \|_1 \leq \| \mathbf{v}_S \|_1 \leq \rho \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \|_1 \leq \rho \| \mathbf{v}_{\bar{S}'} \|_1 + \tau \| A\mathbf{v} \|.$$

Therefore, A satisfies the robust null space property of order s with $0 < \rho < 1$ and $\tau > 0$. □

The next theorem and proof is the most complicated situation we have to handle, but it is the most common in practice. Due to its complicated nature, when we prove results that are not central to thesis in later chapters we will normally only prove the result for the standard basis pursuit. However, theorem 12 will prove that the robust null space property holds for a certain class of matrices which will be defined in part 2.

Theorem 8 ([18], Theorem 4.20)**.** *Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix. Then A satisfies the robust null space property of order s with $0 < \rho < 1$ and $\tau > 0$ if and only if for all $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{C}^N$, we have*

$$\| \boldsymbol{z} - \boldsymbol{x} \|_1 \leq \frac{1+\rho}{1-\rho} \left(\| \boldsymbol{z} \|_1 - \| \boldsymbol{x} \|_1 + 2 \| \boldsymbol{x}_{\bar{S}} \|_1 \right) + \frac{2\tau}{1-\rho} \| A(\boldsymbol{z} - \boldsymbol{x}) \|$$

Proof. Suppose, for all $\mathbf{x}, \mathbf{z} \in \mathbb{C}^N$, we have

$$\| \mathbf{z} - \mathbf{x} \|_1 \leq \frac{1+\rho}{1-\rho} \left(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1 \right) + \frac{2\tau}{1-\rho} \| A(\mathbf{z} - \mathbf{x}) \|$$

Let $\mathbf{v} \in \mathbb{C}^N$ be an arbitrary vector. Then, for any $S \subset [N]$ with $|S| \leq s$, we can apply this inequality with $\mathbf{z} = \mathbf{v}_S$ and $\mathbf{x} = -\mathbf{v}_S$ to obtain

$$\| \mathbf{v} \|_1 \leq \frac{1+\rho}{1-\rho} (\| \mathbf{v}_{\bar{S}} \|_1 - \| \mathbf{v}_S \|_1) + \frac{2\tau}{1-\rho} \| A\mathbf{v} \|$$

$$\Leftrightarrow (1-\rho) \| \mathbf{v} \|_1 \leq (1+\rho)(\| \mathbf{v}_{\bar{S}} \|_1 - \| \mathbf{v}_S \|_1) + 2\tau \| A\mathbf{v} \|$$

$$\Leftrightarrow \qquad \| \mathbf{v}_S \|_1 \leq \rho \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \| .$$

Therefore, A satisfies the robust null space property of order s with $0 < \rho < 1$ and $\tau > 0$.

Next, suppose A satisfies the robust null space property of order s with $0 < \rho < 1$ and $\tau > 0$. Let $\mathbf{x}, \mathbf{z} \in \mathbb{C}^N$ be arbitrary vectors. Consider the inequalities

$$\begin{aligned} \| \mathbf{x} \|_1 &= \| \mathbf{x}_{\bar{S}} \|_1 + \| \mathbf{x}_S \|_1 \\ &= \| \mathbf{x}_{\bar{S}} \|_1 + \| \mathbf{x}_S - \mathbf{z}_S + \mathbf{z}_S \|_1 \\ &\leq \| \mathbf{x}_{\bar{S}} \|_1 + \| \mathbf{x}_S - \mathbf{z}_S \|_1 + \| \mathbf{z}_S \|_1 , \\ &\text{and} \\ \| \mathbf{x}_{\bar{S}} - \mathbf{z}_{\bar{S}} \|_1 &\leq \| \mathbf{x}_{\bar{S}} \|_1 + \| \mathbf{z}_{\bar{S}} \|_1 . \end{aligned}$$

If we add these inequalities together and subtract $\| \mathbf{x} \|_1$ from both sides, we obtain

$$\| \mathbf{x}_{\bar{S}} - \mathbf{z}_{\bar{S}} \|_1 \leq \| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + \| \mathbf{x}_S - \mathbf{z}_S \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1 .$$

Define $\mathbf{v} := \mathbf{z} - \mathbf{x}$. Then

$$\| \mathbf{v}_{\bar{S}} \|_1 \leq \| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + \| \mathbf{v}_S \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1 .$$

By applying this inequality to the robust null space property, we have

$$\begin{aligned}
&\| \mathbf{v}_S \|_1 \leq \rho \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \| \\
&\qquad \leq \rho(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + \| \mathbf{v}_S \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \tau \| A\mathbf{v} \| \\
\Leftrightarrow &(1-\rho) \| \mathbf{v}_S \|_1 \leq \rho(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \tau \| A\mathbf{v} \| \\
\Leftrightarrow &\qquad \| \mathbf{v}_S \|_1 \leq \frac{1}{1-\rho}(\rho(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \tau \| A\mathbf{v} \|).
\end{aligned}$$

By again applying the robust null space property, we have

$$\begin{aligned}
\| \mathbf{v} \|_1 &= \| \mathbf{v}_S \|_1 + \| \mathbf{v}_{\bar{S}} \|_1 \\
&\leq (\rho \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \|) + \| \mathbf{v}_{\bar{S}} \|_1 \\
&= (1+\rho) \| \mathbf{v}_{\bar{S}} \|_1 + \tau \| A\mathbf{v} \| \\
&\leq (1+\rho)(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + \| \mathbf{v}_S \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \tau \| A\mathbf{v} \| \\
&\leq (1+\rho)(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + (\frac{1}{1-\rho}(\rho(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \\
&\qquad \tau \| A\mathbf{v} \|)) + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \tau \| A\mathbf{v} \| \\
&= \left(\frac{1+\rho}{1-\rho}\rho + 1 + \rho\right)(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|) + \left(\frac{1+\rho}{1-\rho} + 1\right)\tau \| A\mathbf{v} \| \\
&= \frac{1+\rho}{1-\rho}(\| \mathbf{z} \|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|) + \frac{2\tau}{1-\rho} \| A\mathbf{v} \| .
\end{aligned}$$

This is the inequality we originally set out to prove. □

Theorem 9 provides a bound b on the error for quadratically constrained basis pursuit, as discussed in problem 3. This error bound is larger than the error bound in theorem 6, which is to be expected because of the added measurement error.

Theorem 9 ([18], Theorem 4.19)**.** *Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix that satisfies the robust null space property of order s with $0 < \rho < 1$ and $\tau > 0$. Let $\boldsymbol{x} \in \mathbb{C}^N$. Then any solution $\boldsymbol{x}^\sharp$ of $(P_{1,\eta})$ with the measurement vector $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{e}$ and $\| e \| \leq \eta$ approximates $\boldsymbol{x}$ with error bound*

$$\left\| \boldsymbol{x} - \boldsymbol{x}^\sharp \right\|_1 \leq \frac{1+\rho}{1-\rho} 2\sigma_s(\boldsymbol{x})_1 + \frac{4\tau}{1-\rho}\eta.$$

Proof. Let $\mathbf{x}^\sharp$ be a solution of $(P_{1,\eta})$ so that $\left\| \mathbf{x}^\sharp \right\|_1 \leq \| \mathbf{x} \|_1$ and $\left\| A\mathbf{x}^\sharp - \mathbf{y} \right\| \leq \eta$. Let S be a subset of $[N]$ with $|S| \leq s$ and $\mathbf{x}_S$ contains the s largest absolute entries of $\mathbf{x}$ so that $\| \mathbf{x}_{\bar{S}} \|_1 = \sigma_s(\mathbf{x})_1$. Using the upperbound from the previous theorem with $\mathbf{z} := \mathbf{x}^\sharp$, we obtain

$$\begin{aligned}
\left\| \mathbf{x}^\sharp - \mathbf{x} \right\|_1 &\leq \frac{1+\rho}{1-\rho}(\left\| \mathbf{x}^\sharp \right\|_1 - \| \mathbf{x} \|_1 + 2 \| \mathbf{x}_{\bar{S}} \|_1) + \frac{2\tau}{p-1} \left\| A\mathbf{x}^\sharp - A\mathbf{x} \right\| \\
&= \frac{1+\rho}{1-\rho}(\left\| \mathbf{x}^\sharp \right\|_1 - \| \mathbf{x} \|_1 + 2\sigma_s(\mathbf{x})_1) + \frac{2\tau}{p-1} \left\| A\mathbf{x}^\sharp - (\mathbf{y} - \mathbf{e}) \right\| \\
&\leq \frac{1+\rho}{1+\rho}(\| \mathbf{x} \|_1 - \| \mathbf{x} \|_1 + 2\sigma_s(\mathbf{x})_1) + \frac{2\tau}{\rho-1}(\left\| A\mathbf{x}^\sharp - \mathbf{y} \right\| + \| \mathbf{e} \|) \\
&\leq \frac{1+\rho}{1-\rho} 2\sigma_s(\mathbf{x})_1 + \frac{4\tau}{\rho-1}\eta.
\end{aligned}$$

□

4 The Coherence, the Restricted Isometry Property, and the Quadratic Bottleneck Problem

In the previous chapter, we showed that if a measurement matrix satisfies the null space property, then the compressed sensing problem is solved by an algorithm that computes the solution to (P_1). However, the null space property is difficult to check in general. In this chapter, we develop measures on the measurement matrix that allow us to avoid checking any of the variations of the null space property directly. We define two of these measures in the first section. In the second section, we show that measurement matrices with certain values for these measures satisfy the null space property. In the third section, we discuss the optimal values measurement matrices can obtain. In the fourth section, the optimal values of the previous section are used to deduce lowerbounds on the number of rows a measurement matrix must have to guarantee sparse recovery from basis pursuit. In the final section, a more complicated measure, called restricted isometry property is defined.

Before starting, we note that all of the measures in this chapter require us to use a measurement matrices with ℓ_2-normalized columns. This is in direct conflict with the measures and the measurement matrices used in part 2 of this thesis, but they are still worth mentioning because of their ubiquity in compressed sensing theory.

4.1 Coherence and ℓ_1-Coherence Function

The coherence and the ℓ_1-coherence function are easy to compute measures for determining the quality of a measurement matrix. However, we will see that these measures are limited. Even with these limitations, the coherence and the ℓ_1-coherence function measures are still worth discussing because they are computable and the restricted isometry constant (defined in section 4.5) of a measurement matrix is a generalization of the coherence.

Definition 10 (Coherence)**.** Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix with ℓ_2-normalized columns. Then the *coherence* $\mu(A)$ of A is defined by,

$$\mu(A) := \max_{1\leq i<j\leq N} |\langle \mathbf{a}_i, \mathbf{a}_j \rangle|.$$

Definition 11 (ℓ_1-Coherence Function)**.** Let $A \in \mathbb{C}^{m\times N}$ be a measurement matrix with ℓ_2-normalized columns. Then the *ℓ_1-coherence function* $\mu_1(A, s)$

of A is defined by,

$$\mu_1(A, s) := \max_{i \in [N]} \max\{\sum_{j \in S} |\langle \mathbf{a}_i , \mathbf{a}_j \rangle| : S \subset [N] \text{ with } |S| = s \text{ and } i \notin S\}.$$

It will be shown in theorem 10 that a measurement matrix with a small ℓ_1-coherence function is a higher quality measurement matrix. Since it follows directly from the definitions of μ and μ_1 that

$$\mu(A) \leq \mu_1(A, s) \leq s\mu(A),$$

we also have that a matrix with small coherence is also a higher quality measurement matrix.

4.2 Coherence Recovery Guarantees

This section demonstrates the connection between the ℓ_1-coherence function and basis pursuit.

Theorem 10 ([18], Theorem 5.15)**.** *Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix with ℓ_2-normalized columns. If A satisfies*

$$\mu_1(A, s) + \mu_1(A, s-1) < 1,$$

then any s-sparse vector $\boldsymbol{x} \in \mathbb{C}^{m \times N}$ is the unique solution of basis pursuit with measurement vector $\boldsymbol{y} = A\boldsymbol{x}$.

Proof. The strategy of this proof is to prove that A satisfies the null space property of order s. Let the vector $\mathbf{v} \in \mathcal{N}(A) \setminus \{\mathbf{0}\}$ and the subset $S \subset [N]$ with $|S| \leq s$ both be arbitrary. Then

$$A\mathbf{v} = \sum_{j \in [N]} v_j \mathbf{a}_j = 0.$$

This implies that for any $i \in S$, we have

$$v_i \mathbf{a}_i = - \sum_{j \in [N] \setminus \{i\}} v_j \mathbf{a}_j.$$

Using this and the assumption that the columns of A are ℓ_2-normalized implies,

$$\begin{aligned} v_i &= v_i \| \mathbf{a}_i \|_2^2 \\ &= \langle v_i \mathbf{a}_i , \mathbf{a}_i \rangle \\ &= - \sum_{j \in [N] \setminus \{i\}} \langle v_j \mathbf{a}_j , \mathbf{a}_i \rangle \\ &= - \sum_{l \in \overline{S}} \langle v_l \mathbf{a}_l , \mathbf{a}_i \rangle - \sum_{j \in S \setminus \{i\}} \langle v_j \mathbf{a}_j , \mathbf{a}_i \rangle. \end{aligned}$$

By taking the absolute value of this equation and the triangle inequality, we obtain

$$|v_i| \leq |\sum_{l\in\overline{S}} \langle\, v_l \mathbf{a}_l \,,\, \mathbf{a}_i \,\rangle| + |\sum_{j\in S\setminus\{i\}} \langle\, v_j \mathbf{a}_j \,,\, \mathbf{a}_i \,\rangle|.$$

Therefore, by summing over all i, we obtain

$$\begin{aligned}
\| \mathbf{v}_S \| &= \sum_{i\in S} |v_i| \\
&\leq \sum_{l\in\overline{S}} \left(|v_l| \sum_{i\in S} |\langle\, \mathbf{a}_l \,,\, \mathbf{a}_i \,\rangle| \right) + \sum_{j\in S} \left(|v_j| \sum_{i\in S\setminus\{j\}} |\langle\, \mathbf{a}_j \,,\, \mathbf{a}_i \,\rangle| \right) \\
&\leq \sum_{l\in\overline{S}} |v_l| \mu_1(A,s) + \sum_{j\in S\setminus\{i\}} |v_j| \mu_1(A,s-1) \\
&= \mu_1(A,s) \,\| \mathbf{v}_{\overline{S}} \|_1 + \mu_1(A,s-1) \,\| \mathbf{v}_S \|_1 \,.
\end{aligned}$$

By subtracting $\| \mathbf{v}_S \|_1 \, \mu_1(A,s-1)$ from both sides of this inequality and using the assumption $\mu_1(A,s) < 1 - \mu_1(A,s-1)$, we have

$$\mu_1(A,s) \,\| \mathbf{v}_S \|_1 < (1 - \mu_1(A,s-1)) \,\| \mathbf{v}_S \|_1 \leq \mu_1(A,s) \,\| \mathbf{v}_{\overline{S}} \|_1 \,.$$

Finally, by dividing both sides of this inequality by $\mu_1(A,s)$, we arrive at

$$\| \mathbf{v}_S \|_1 < \| \mathbf{v}_{\overline{S}} \|_1 \,.$$

Thus, A satisfies the null space property of order s and basis pursuit recovers every s-sparse vector $\mathbf{x}$ from its measurement vector $\mathbf{y} = A\mathbf{x}$. □

This theorem does not address the connection between the ℓ_1-coherence function and quadratically constrained basis pursuit. We do not dwell on this because a stronger version will be proven for a different measure in part 2 of this thesis.

4.3 Optimal Coherence

Now that we have demonstrated that a matrix with a small μ and μ_1 recovers less sparse vectors, we turn our attention to how small can we make μ and μ_1 and what class of measurement matrices satisfy these values. Again, we do not dwell to much on the details of either of these question because only their existence is of interest to us in the next section.

First we consider the range of values the coherence can have. Observe that $\mu(A) \leq 1$ follows from the Cauchy-Schwartz inequality,

$$|\langle\, \mathbf{a}_i \,,\, \mathbf{a}_j \,\rangle| \leq \| \mathbf{a}_i \|_2 \,\| \mathbf{a}_j \|_2 = 1.$$

Similarly, $\mu(A) = 0$ if and only if $|\langle\, \mathbf{a}_i \,,\, \mathbf{a}_j \,\rangle| = 0$ for all distinct $i, j \in [N]$. This occurs precisely when A is square and unitary. Since we have assumed that measurement matrices are not square, we have $\mu(A) > 0$. Therefore,

$$\mu(A) \in (0, 1],$$

and it follows immediately from the definition of the ℓ_1-coherence function that

$$\mu_1(A, s) \in (0, s].$$

Because the coherence and the ℓ_1-coherence function of a matrix cannot be equal to zero, a sharp lower bound is desirable as well as the class of matrices that satisfy this bound. With that being said, the following classes of measurement matrices are relevant.

Definition 12 (Equiangular Tight Frames)**.** Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix with ℓ_2-normalized columns. Then A is *equiangular* if there exists $c > 0$ such that $|\langle \mathbf{a}_i, \mathbf{a}_j \rangle| = c$ for all distinct $i, j \in [N]$. The matrix A is a *tight frame* if there exists $\lambda > 0$ such that $AA^* = \frac{1}{\lambda} I$. The matrix A is an *equiangular tight frame* if it is both equiangular and a tight frame.

It is known that the coherence of measurement matrix A satisfies the *Welch bound*,

$$\mu(A) \geq \sqrt{\frac{N-m}{m(N-1)}}.$$

Similarly, the *Welch bound* on the ℓ_1-coherence function is

$$\mu_1(A, s) \geq s\sqrt{\frac{N-m}{m(N-1)}}$$

for any $s \in [N-1]$. In theorems 5.7 and 5.8 of [18] it is shown that equality holds in both of these inequalities if and only if A is an equiangular tight frame.

4.4 The Quadratic Bottleneck Problem

Minimizing the Number of Rows of the Measurement Matrix Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix. Recall that in the applications paragraph of chapter 1, we said that a measurement matrix corresponds to a measurement scheme of a sparse signal. We motivate this section by elaborating on this.

Each row of the measurement matrix corresponds to a single measurement of the sparse signal. Therefore, minimizing the number of rows m of the measurement matrix minimizes the number of measurements taken of the signal. Intuitively, we cannot take an arbitrarily small amount of measurements and expect to reconstruct the signal at the same time. From this, we can deduce that there must exist a lower bound on the minimum number of rows a measurement matrix must have in order to recover a sparse vector from its measurement vector.

Minimum Number of Rows Required for Basis Pursuit Now that we have cited that the Welsh bound is the sharp lowerbound for the ℓ_1-coherence function, we can use the Welsh bound to obtain a lowerbound on the minimum number of rows a measurement matrix must have so that any s-sparse vector can be proven to be the unique solution of basis pursuit. Recall that the Welch bound on the ℓ_1-coherence function for any measurement matrix $A \in \mathbb{C}^{m \times N}$ is

$$\mu_1(A, s) \geq s\sqrt{\frac{N-m}{m(N-1)}}.$$

Since measurement matrices satisfy $N \gg m$, we have $\frac{N-m}{n-1} \approx 1$. Thus,

$$\mu_1(A, s) \geq s\sqrt{\frac{N-m}{m(N-1)}} \approx \frac{s}{\sqrt{m}}.$$

Suppose there exists a small constant $c > 0$ such that A satisfies

$$\mu_1(A, s) = c\left(\frac{s}{\sqrt{m}}\right).$$

This implies A is close to satisfying the Welch bound. By theorem 10, every s-sparse vector is the unique solution basis pursuit if

$$1 > \mu_1(A, s) + \mu_1(A, s-1) = c\frac{s}{\sqrt{m}} + c\frac{s-1}{\sqrt{m}} = c\frac{2s-1}{\sqrt{m}}.$$

Reordering this inequality gives us

$$\sqrt{m} \geq Cs,$$

or equivalently

$$m \geq Cs^2, \tag{4.1}$$

for some constant $C > 0$ that depends only on c. This is known as the *square root bottleneck* or *quadratic bottleneck* in compressed sensing. There are many classes of matrices that require the number of rows to scale quadratically with respect to sparsity, such as equiangular tight frames and a few others that we will introduce in part 2. The quadratic bottleneck earns its name because it is an open problem to explicitly construct matrices that can be proven to recover s-sparse vectors that scale smaller then (4.1). A solution to the quadratic problem is not unique, so many explicitly constructed classes of measurement matrices may exist. It is known that there exist measurement matrices such that

$$m \geq Cs\log(N/s) \tag{4.2}$$

holds for some positive constant C. It would be preferable to explicitly construct a measurement matrix with the bound (4.2) rather then the bound (4.1). This is because we can choose the number of rows m to be much smaller as the sparsity s and the number of rows N becomes arbitrarily large. Certain classes of random measurement matrices have been shown with high probability to recover every s-sparse vector with the lowerbound (4.2) on the number of rows. We do not go into detail here about these classes of random matrices because our focus is on explicitly constructed matrices.

With this being said, there are some issues of using random measurement matrices. One of these issues is that we have to randomly generate the matrix every time we use it. This requires some extra computation and it may not be desirable for an engineer to create a physical sensor capable of doing this. The second issue is that we cannot verify if a given random measurement matrix satisfies this bound because it is NP-hard to compute this for reasons discussed in the next section. Even with these shortcomings, the use of random measurement matrices are a central focus of compressed sensing research and are still very useful.

The ℓ_1-Coherence Function is Insufficient We conclude this section with this final remark. If we are limited to only using the ℓ_1-coherence function, then we cannot find a solution to the quadratic bottleneck problem. To see this, consider the following proof by contradiction. Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix with

$$m \leq (2s-1)^2/2 = Cs^2.$$

Since we know that N is much larger then m, we can assume that $N \geq 2m$. Suppose the sparsity satisfies $s < \sqrt{N-1}$ and the measurement matrix A satisfies $1 > \mu_1(A,s)+\mu_1(A,s-1)$. Then, it follows from theorem 10 that every s-sparse vector is the unique solution of basis pursuit from its measurement vector and it follows from the Welch bound that

$$\begin{aligned}
1 &> \mu_1(A,s) + \mu_1(A,s-1) \\
&\geq s\sqrt{\frac{N-m}{m(N-1)}} + (s-1)\sqrt{\frac{N-m}{m(N-1)}} \\
&\geq (2s-1)\sqrt{\frac{2(N-m)}{(2s-1)^2(N-1)}} \\
&\geq \sqrt{\frac{2N-N}{N-1}} \\
&= \sqrt{\frac{N}{N-1}},
\end{aligned}$$

which is a contradiction. Therefore, the condition $1 > \mu_1(A,s)+\mu_1(A,s-1)$ can only be used to show that a measurement matrix must have at least $m \geq Cs^2$ rows so that every s-sparse vector is the unique solution of basis pursuit from its measurement vector.

4.5 The Restricted Isometry Property

The following measure is an alternative to the two measures defined in the previous sections.

Definition 13 (Restricted Isometry Property)**.** Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix with ℓ_2-normalized columns. Then the *s-th restricted isometry constant* of A, denoted δ_s, is defined as the minimimum δ that satisfies

$$(1-\delta) \| \mathbf{x} \|_2^2 \leq \| A\mathbf{x} \|_2^2 \leq (1+\delta) \| \mathbf{x} \|_2^2, \tag{4.3}$$

for all s-sparse vectors $\mathbf{x} \in \mathbb{C}^N$. The equation (4.3) is called the *restricted isometry property.*

The restricted isometry constant can be used to prove much stronger recovery results than the ℓ_1-coherence function. Precisely, a matrix with a small δ_s can be used to prove that the number of rows m of a matrix must have at least

$$m \geq Cs \log(N/s)$$

instead of at least

$$m \geq Cs^2,$$

to uniquely recover all s-sparse vectors from basis pursuit, where $C > 0$ is a small constant that does not depend on s. Thus, δ_s can be used to determine if a matrix with ℓ_2-normalized columns is a solution to the quadratic bottleneck. However, it is known from [2] that computing δ_s of a given matrix is NP-hard in general.

It is natural then to ask why we should even bother defining the restricted isometry property when we have the null space property. This appears to be because that restricted isometry constant is more easily approximated. That is, if we can prove that there exists a small δ such that $\delta_s \leq \delta$, then we have proven δ_s is small. Unfortunately, bounds for explicitly constructed matrices usually have to rely on the coherence of a matrix or Gershgorin's circle theorem. From the discussion at the end of the previous section, this approach to approximating δ_s is insufficient for proving stronger bounds then the quadratic bottleneck.

Another reason the restricted isometry property is used in place of the null space property is that in many of the convergence theorems of compressed sensing recovery algorithms, δ_s is used to prove that the algorithm does indeed converge. In addition to this, δ_s is used to determine the rate of convergence of the algorithm.

It can be verified that δ_s of a matrix A can be computed from

$$\delta_s = \max_{\substack{S \subset [N] \\ |S| \leq s}} \left\| A_S^T A_S - I \right\|_2 . \tag{4.4}$$

Equation (4.4) is still NP-hard to compute in general but it is still a very common equality to use in the analysis of algorithms.

As stated earlier in this chapter, δ_s is a generalization of μ. This is made precise in the next theorem.

Theorem 11 ([18], Prop 6.2)**.** *Let $A \in \mathbb{C}^{m \times N}$ be a measurement matrix with ℓ_2-normalized columns. Then $\delta_2 = \mu$.*

Proof. First observe that for all distinct $i, j \in [N]$, we have

$$A_{\{i,j\}}^* A_{\{i,j\}} - I = \begin{bmatrix} 0 & \langle \mathbf{a}_j , \mathbf{a}_i \rangle \\ \langle \mathbf{a}_i , \mathbf{a}_j \rangle & 0 \end{bmatrix}.$$

It follows immediately that the eigenvalues of this matrix are precisely $\pm \langle \mathbf{a}_i , \mathbf{a}_j \rangle$. Therefore,

$$\left\| A_{\{i,j\}}^* A_{\{i,j\}} - I \right\|_2 = \langle \mathbf{a}_i , \mathbf{a}_j \rangle .$$

Using the previous equation and equation (4.4), we have

$$\delta_2 = \max_{1 \leq i < j \leq N} \left\| A_{\{i,j\}}^* A_{\{i,j\}} - I \right\|_2 = \max_{1 \leq i < j \leq N} \langle \mathbf{a}_i , \mathbf{a}_j \rangle = \mu,$$

which is the equality we set out to prove. □

It follows from theorem 10 and theorem 11 that δ_2 can be used to recover sparse vectors from basis pursuit. In fact, if we can prove $\delta_{2s} < 1/3$, then it is known from theorem 6.9 of [18] that every s-sparse vector is the unique solution of basis pursuit from its measurement vector. We will not prove this here since we will prove a similar result in theorem 13 of part 2. The second part of this thesis builds on the quadratic bottleneck problem and explicitly constructing of measurement matrices that are solutions to this problem.

5 Summary of Reference for Part One

The the results of this section are foundational to the theory and are well known to specialists in compressed sensing. Because of this, part 1 can be considered a survey of the relevant material.

Many of the definitions and results from part 1 were obtained from [18], a textbook of the mathematical theory of compressed sensing. This text was used in an introductory course to the theory and taught by Øyvind Ryan. Since the text is a reference work, the original sources are listed here, if they are available. It should also be noted that [8] was used as a first introduction to compressed sensing.

The quadratic bottleneck problem was introduced to me by the introductory chapter in [18]. However, Terence Tao informally wrote about this problem about a decade ago at [26]. Here, he phrases the quadratic bottleneck problem as a derandomization problem in complexity theory.

It was obtained from [3] that the only solution to the quadratic bottlneck problem that exists is given in [7]. As suggested by the name of [3], most approaches to the quadratic bottleneck problem involves using the restricted isometry property. The discussion for the insufficiency of the ℓ_1-coherence function was originally obtained from chapter 5 of [18], but variations of this has shown up in other places such as [3]. Equation 4.4 was also obtained from chapter 6 in [18].

The spark was obtained from [3], but was first defined in [14]. The terms stable and robust null space property were introduce in the text [18], but similar concepts exist throughout compressed sensing theory. A deeper analysis of the literature of the topics that we only mentioned in passing can be done. In particular, the topics of equiangular tight frames, random measurement matrices, the physical applications of compressed sensing, and the restricted expansion property. Not much time was spent on these topics when writing this thesis other than their existence, which, again were obtained from [18].

Part II

The Combinatorics Approach to Compressed Sensing and the Quadratic Bottleneck Problem

6 Lossless Expander Graphs

Up until this chapter we have dealt with compressed sensing in the most general setting. That is, we have proved results relevant to compressed sensing with arbitrary complex measurement matrices. From this chapter on, we will only be dealing with sparse measurement matrices that have entries 0 and 1. We will first start with some preliminary definitions. Following this, we will define the class of graphs that is relevant to compressed sensing.

6.1 Introduction to Graph Theory

Throughout part 2, we will encounter several kinds of discrete structures. We introduce the first one of these now.

Definition 14 (Graph)**.** A *graph* G is an ordered pair (V, E), where the set V is the set of *vertices* of G and the set E is the set of *edges* of G. Each edge of G is defined by an unordered pair of vertices of G.

There are many classes of graphs, but we will only require a few of them. The first class of graphs is characterized by partitioning its vertices in the following way.

Definition 15 (Bipartite Graph)**.** A graph G is called a *bipartite graph* if its vertex set can be partitioned into two subsets, which we call the *left vertex set* L and the *right vertex set* R, so that every edge has one end in L and the other end in R.

Normally bipartite graphs do not make the distinction between left and right vertex sets. When an application requires one subset of vertices to have distinct properties from the other subset of vertices, it is useful to include the orientation of left and right vertex sets to differentiate between the two. The following classes of graphs will only be used as subgraphs of a bipartite graph.

Definition 16 (Cycle and Path)**.** A *cycle* is a graph whose vertices can be arranged in a cyclic sequence so that if the two vertices are consecutive in the sequence then there exists an edge between the two vertices. A *path* is a graph whose vertices can be arranged in a linear sequence so that if the two vertices are consecutive in the sequence then there exists an edge between the two vertices.

The following two definitions are measures on paths and cycles.

Definition 17 (Length and Distance)**.** Let G be a graph with vertices u and v. Then the *length* of the cycle or path in G is the number of edges it contains. The *distance* between the vertices u and v is the length of the shortest path from u to v.

We will always assume that a graph is a *simple graph*. That is, every graph we consider does not have a cycle of length 1 or 2. We will use the following necessary condition for a cycle to exist as a subgraph of a bipartite graph several times throughout part 2 of this thesis.

Lemma 1 ([6], Ex 1.1.3)**.** *If G is a bipartite graph then, G it does not contain a cycle of odd length.*

Proof. Suppose G is a bipartite graph with left vertex set L and right vertex set R. Suppose there exists a cycle $C := (v_1, \ldots, v_n)$ of odd length. Since G is a bipartite graph, every edge has ends that lie in distinct vertex sets. Without loss of generality, we can assume that L contains all the odd indexed vertices and R contains all the even indexed vertices in the cyclic sequence of C. However, since n is odd, this implies that there exists an edge $e = \{v_1, v_n\}$ with $v_1, v_n \in L$. This contradicts the assumption that G is a bipartite graph. Therefore, G cannot contain a cycle of odd length. □

The following two graph properties will be used in future chapters to describe how connected a graph is.

Definition 18 (Girth and Diameter)**.** Let G be a graph. If G has at least one cycle then, the length of the shortest cycle in G is called the *girth* of G and is denoted by girth(G). The *diameter* of G, denoted diam(G), is the greatest distance between any two vertices of G.

The following lemma will be of use to us in future proofs.

Lemma 2 ([6], Ex 3.1.10)**.** *Let G be a graph. If there exists an integer k such that girth$(G) = 2k$, then diam$(G) \geq k$.*

Proof. Suppose girth$(G) = 2k$, for some integer k. Then there exists a cycle $C = (v_1, \ldots, v_{2k})$ in G of length $2k$. This implies that there exists a path of length k from v_1 to v_{k+1}. Suppose there exists a shorter path from v_1 to v_{k+1}. Then there exists a cycle of length less then $2k$. But this contradicts the assumption that girth$(G) = 2k$. Thus a shorter path from v_1 to v_{k+1} cannot exist and we have shown that diam$(G) \geq k$. □

In the next definition, we introduce a term that characterizes the local structure of a graph.

Definition 19 (Neighborhood)**.** Let G be a graph and let v be a vertex of G. Then the set of all adjacent vertices to v in G is called the set of *neighbors* of v and is denoted by $N(v)$. Similarly, if S is a subset of vertices of G, then the set of vertices adjacent to vertices in S is the *neighborhood* of S and is denoted $N(S)$. The *degree* d of the vertex v in G is defined by $d = |N(v)|$.

If G is a bipartite graph and we are interested in the neighborhood of a subset of left vertices, we will denote the set $N(S)$ by $R(S)$ and refer to it as the set of *right vertices of* S.

Definition 20 (Biregular Graphs)**.** Suppose G be a bipartite graph. If every left vertex has degree d, then G is called a *d-left regular biparite graph*. If every right vertex has degree p, then G is called a *p-right regular bipartite graph*. If G is both a d-left vertex graph and a p-right vertex graph, then G is called a *biregular graph* and is denoted by $G_{p,d}$.

Next, we introduce a specific way of obtaining a subgraph of a graph.

Definition 21 (Induced Graph)**.** Let G be a graph with vertex set V. Let X and $Y := V - X$ be two subsets of vertices of G. Then the *induced subgraph* $G[Y]$ of G is the subgraph of G obtained by deleting the vertices of X and all edges with an end in X.

One of the nice properties of graphs is that they can be easily related to other types of discrete structures, such as matrices.

Definition 22 (Bipartite Adjacency Matrix)**.** Let G be a bipartite graph with N left vertices and m right vertices. Then the *bipartite adjacency matrix* of the bipartite graph G is the matrix $A \in \{0,1\}^{m \times N}$ such that $a_{i,j} = 1$ if the jth left vertex is adjacent to the ith right vertex of G and $a_{i,j} = 0$ otherwise.

The bipartite adjacency matrix of the class of left regular bipartite graphs is what will be of primary interest to us. When the number of left vertices is much larger then the number of right vertices, the number of columns in the bipartite adjacency matrix is much larger then the number of rows. In this context, the bipartite adjacency matrix may be able to be used as measurement matrix.

6.2 Defining Lossless Expander Graphs

Not every graph in the class of left regular bipartite graphs is necessarily useful to us. To differentiate between the useful matrices and the useless matrices, we define the following measure.

Definition 23 (Restricted Expansion Property)**.** Let G be a d-left regular bipartite graph. If for every subset S of left vertices with $|S| \leq s$, the bipartite graph G satisfies the *restricted expansion property*

$$|R(S)| \geq (1 - \theta)d|S|,$$

then G is an *(s, d, θ)-lossless expander*. The minimum $\theta \geq 0$ for which the expansion property holds for all subsets of left vertices S with $|S| \leq s$ is called the *s^{th} restricted expansion constant θ_s*.

The restricted expansion constant is the measure that we referred to several times in chapter 4. An (s, d, θ)-lossless expander graph will sometimes be referred to as just a *lossless expander graph* when the parameters aren't necessary for the discussion. It should be noted that when an author talks about an expander graph, they are usually referring to a general graph where each small set of vertices have a lot of neighbors [1]. Since we only work with bipartite graphs here, this shouldn't cause any ambiguity.

By requiring θ_s to be "small", we can prove that the bipartite adjacency matrix A satisfies the robust null space property. Unfortunately it is NP-Hard

to compute θ_s in general by the same reasoning for which the combinatorial optimization problem (P_0) is NP-hard. A good approach is to define a more restrictive family of lossless expander graphs with additional properties. We can then use these additional properties to approximate the restricted expansion constant θ_s by a hopefully not much larger θ so that

$$|R(S)| \geq (1 - \theta_s)d|S| \geq (1 - \theta)d|S|$$

holds. This inequality demonstrates that θ_s is used to provide a sharp bound on the minimum number of right vertices any subset of less then s left vertices can have, whereas the approximation θ just provides a not necessarily sharp bound.

As a final note, observe that since we require θ_s to hold for all $|S| \leq s$, it follows directly from the definition of the restrict expansion property that

$$0 = \theta_1 \leq \ldots \leq \theta_{s-1} \leq \theta_s \leq \theta_{s+1} \leq \ldots \theta_N.$$

This implies that if we find an approximation θ for θ_s, then we have found an approximation for all restricted expansion constants of order less then s.

7 Lossless Expander Graphs Satisfy the Robust Null Space Property

In chapter 3, we proved that a matrix that satisfies the robust null space property solves the most general compressed sensing problem, where vector to be recovered is compressible and measurement error exists. In the last chapter, we introduced the bipartite adjacency matrices of lossless expander graphs. We will now show that if the restricted expansion constant is small enough, then these matrices satisfy the robust null space property. Following this, we will discuss some strategies for approximating the restricted expansion constant.

7.1 Lossless Expander Graphs Satisfy the Robust Null Space Property

To prove that the bipartite adjacency matrix of a lossless expander graph with small restricted expansion constant satisfies the robust null space property, we need to use lemmas 6.10, 13.12, and 13.13 in [18]. The statements of these lemmas are complicated but their application is simple in that they will all be called in one line of the proof. We will make it clear when we use them.

Theorem 12 ([18], Theorem 13.11)**.** *Let $A \in \{0,1\}^{m\times N}$ be the bipartite adjacency matrix of an (s,d,θ)-lossless expander graph G with $\theta_{2s} < 1/6$. Then A satisfies the robust null space property of order s with respect to the ℓ_1-norm and the parameters*

$$\rho = \frac{2\theta_{2s}}{1-4\theta_{2s}} \qquad \tau = \frac{1}{(1-4\theta_{2s})d}.$$

That is, for any subset S of $[N]$ with $|S| \leq s$ any vector $\boldsymbol{v} \in \mathbb{C}^N$, we have

$$\| \boldsymbol{v}_S \|_1 \leq \frac{2\theta_{2s}}{1-4\theta_{2s}} \| \boldsymbol{v}_{\bar{S}} \|_1 + \frac{1}{(1-4\theta_{2s})d} \| A\boldsymbol{v} \|_1 .$$

Proof. Let $\mathbf{v} \in \mathbb{C}^N$ be arbitrary. Partition the index set $[N]$ of $\mathbf{v}$ into the subsets $S_0, S_1, \ldots$, where the subset S_0 is the s largest absolute elements of $\mathbf{v}$, the subset S_1 is the next s largest absolute elements of $\mathbf{v}$, continuing in this manner for all the remaining subsets. Since S_0 is the s largest elements of $\mathbf{v}$, if

$$\| \mathbf{v}_{S_0} \|_1 \leq \frac{2\theta_{2s}}{1-4\theta_{2s}} \left\| \mathbf{v}_{\bar{S}_0} \right\|_1 + \frac{1}{(1-4\theta_{2s})d} \| A\mathbf{v} \|_1$$

holds, then it follows from theorem 7 that the robust null space property holds. To keep the notation simple, we label each of the left vertices of G with the column indices $[N]$ of A. Similarly, we label the right nodes of G with the row indices $[m]$ of A. This essentially allows us to treat the subsets $S_0, S_1, \ldots$ as both subsets of left nodes and subsets of column indices.

Next, define the function $\ell : [m] \to [N]$ such that

$$\ell(i) := \arg\max_{j \in [N]} \{|v_j| \, : \, \{j, i\} \text{ is an edge of } G\}.$$

For each subset of indices S of $[N]$, define the subset of edges $E(S)$ by the set of edges with left vertex in S. Then consider the following equalities, which follow from the bijective relationships between the degree of a left node, the edges incident to it, and its adjacent right neighbors.

$$\begin{aligned} d \, \| \, \mathbf{v}_{S_0} \, \|_1 = & \; d \sum_{j \in S_0} | \, 1 \, | \\ = & \sum_{\{j,i\} \in E(S_0)} | \, 1 \, |, \\ = & \sum_{i \in R(S_0)} \sum_{\substack{j \in S_0 \\ \{j,i\} \in E}} | \, 1 \, | \\ = & \sum_{i \in R(S_0)} | \, 1 \, | \quad + \quad \sum_{i \in R(S_0)} \sum_{\substack{j \in S_0 \setminus \{\ell(i)\} \\ \{j,i\} \in E}} | \, 1 \, |. \end{aligned} \tag{7.1}$$

Similarly, for all $i \in R(S_0)$, we have the following equalities which are consequences of the bijective relationships between the lossless expander graph and its bipartite adjacency matrix.

$$\begin{aligned} (A\mathbf{v})_i = & \sum_{j \in [N]} a_{i,j} v_j \\ = & \sum_{\substack{j \in [N] \\ \{j,i\} \in E}} v_j \\ = & \sum_{k \geq 0} \sum_{\substack{j \in S_k \\ \{j,i\} \in E}} v_j \\ = & \sum_{\substack{j \in S_0 \\ \{i,j\} \in E}} v_j \quad + \quad \sum_{k \geq 1} \sum_{\substack{j \in S_k \\ \{j,i\} \in E}} v_j \\ = & \; v_{\ell(i)} \quad + \sum_{\substack{j \in S_0 \setminus \{\ell(i)\} \\ \{j,i\} \in E}} v_j \quad + \quad \sum_{k \geq 1} \sum_{\substack{j \in S_k \\ \{i,j\} \in E}} v_j. \end{aligned} \tag{7.2}$$

Isolating the term $v_{\ell(i)}$ in (7.2) and taking absolute value gives us the following

inequality.

$$\begin{aligned} |\ 1\ | &= |\ -\sum_{\substack{j\in S_0\setminus\{\ell(i)\}\\ \{j,i\}\in E}} v_j - \sum_{k\geq 1}\sum_{\substack{j\in S_k\\ \{i,j\}\in E}} v_j + (A\mathbf{v})_i| \\ &\leq \sum_{\substack{j\in S_0\setminus\{\ell(i)\}\\ \{j,i\}\in E}} |\ 1\ | + \sum_{k\geq 1}\sum_{\substack{j\in S_k\\ \{i,j\}\in E}} |\ 1\ | + |\ 1\ |. \end{aligned} \tag{7.3}$$

Substituting the inequality (7.3) into the equality (7.1) and by applying lemmas 6.10, 13.12, and 13.13 in [18] discussed at the start of the chapter in the fourth line gives us the following inequality.

$$\begin{aligned} d\,\|\,\mathbf{v}_{S_0}\,\|_1 &= \sum_{i\in R(S_0)} |\ 1\ | \quad + \quad \sum_{i\in R(S_0)}\ \sum_{\substack{j\in S_0\setminus\{\ell(i)\}\\ \{j,i\}\in E}} |\ 1\ | \\ &\leq \sum_{i\in R(S_0)} \left[\sum_{\substack{j\in S_0\setminus\{\ell(i)\}\\ \{j,i\}\in E}} |\ 1\ | + \sum_{k\geq 1}\sum_{\substack{j\in S_k\\ \{i,j\}\in E}} |\ 1\ | + |\ 1\ | \right] \\ &\qquad + \sum_{i\in R(S_0)}\ \sum_{\substack{j\in S_0\setminus\{\ell(i)\}\\ \{j,i\}\in E}} |\ 1\ | \\ &= 2\sum_{i\in R(S_0)}\ \sum_{\substack{j\in S_0\setminus\{\ell(i)\}\\ \{j,i\}\in E}} |\ 1\ | + \sum_{k\geq 1}\sum_{i\in R(S_0)}\ \sum_{\substack{j\in S_k\\ \{i,j\}\in E}} |\ 1\ | \\ &\qquad + \sum_{i\in R(S_0)} |\ 1\ | \\ &\leq 2\,(\theta_s d\,\|\,\mathbf{v}_{S_0}\,\|_1) \ + \ (2\theta_{2s} d\,\|\,\mathbf{v}\,\|_1) \ + \ \|\,A\mathbf{v}\,\|_1 \\ &= 2\theta_s d\,\|\,\mathbf{v}_{S_0}\,\|_1 \ + \ (2\theta_{2s} d\,\|\,\mathbf{v}_{S_0}\,\|_1 \ + \ 2\theta_{2s} d\,\big\|\,\mathbf{v}_{\bar{S_0}}\,\big\|_1) \ + \ \|\,A\mathbf{v}\,\|_1 \\ &\leq 4\theta_{2s} d\,\|\,\mathbf{v}_{S_0}\,\|_1 \ + \ 2\theta_{2s} d\,\big\|\,\mathbf{v}_{\bar{S_0}}\,\big\|_1 \ + \ \|\,A\mathbf{v}\,\|_1\,. \end{aligned}$$

By subtracting the term $4\theta_{2s} d\,\|\,\mathbf{v}_{S_0}\,\|_1$ from both sides of this inequality, we obtain

$$(1-4\theta_{2s})d\,\|\,\mathbf{v}_{S_0}\,\|_1 \leq 2\theta_{2s} d\,\big\|\,\mathbf{v}_{\bar{S_0}}\,\big\|_1 + \|\,A\mathbf{v}\,\|_1\,.$$

Dividing both sides of the inequality by $(1-4\theta_{2s})d$ gives us

$$\|\,\mathbf{v}_{S_0}\,\|_1 \leq \frac{2\theta_{2s}}{1-4\theta_{2s}}\,\big\|\,\mathbf{v}_{\bar{S_0}}\,\big\|_1 + \frac{1}{(1-4\theta_{2s})d}\,\|\,A\mathbf{v}\,\|_1\,.$$

Therefore, the robust null space property of order s holds and the proof is complete. □

In the previous theorem, θ_{2s} is chosen to be strictly less then $1/6$ so that the parameter ρ can only take values in the interval $(0, 1)$. More precisely, if $\theta_{2s} < 1/6$, then

$$\rho = \frac{2\theta_{2s}}{1 - 4\theta_{2s}} < \frac{2(1/6)}{1 - 4(1/6)} = \frac{(1/3)}{(1/3)} = 1.$$

Since θ_{2s} only makes sense for strictly positive numbers, this implies $\rho > 0$. This implies that $0 < \rho < 1$, which is the values ρ can take in the definition of the robust null space property. We now prove an upperbound on the ℓ_1-error of a solution of $(P_{1,\eta})$.

Theorem 13 ([18], Theorem 13.10)**.** *Let $A \in \{0,1\}^{m \times N}$ be the bipartite adjacency matrix of an (s, d, θ)-lossless expander G with $\theta_{2s} < 1/6$ and let $\boldsymbol{x} \in \mathbb{C}^N$. Then any solution $\boldsymbol{x}^\sharp$ of $(P_{1,\eta})$ with the measurement vector $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{e}$ and measurement error controlled by $\| \boldsymbol{e} \| \leq \eta$ approximates $\boldsymbol{x}$ with error*

$$\left\| \boldsymbol{x} - \boldsymbol{x}^\sharp \right\|_1 \leq \frac{1 - 2\theta_{2s}}{1 - 6\theta_{2s}} 2\sigma_s(\boldsymbol{x})_1 + \frac{4}{(1 - 6\theta_{2s})d}\eta$$

Proof. Since $\theta_{2s} < 1/6$, the theorem 12 implies that A satisfies the robust null space property of order s with parameters

$$\rho = \frac{2\theta_{2s}}{1 - 4\theta_{2s}} \quad \text{and} \quad \tau = \frac{1}{(1 - 4\theta_{2s})d}.$$

By theorem 9, we have the following inequality

$$\begin{aligned}
\left\| \mathbf{x} - \mathbf{x}^\sharp \right\|_1 &\leq \frac{1 + \rho}{1 - \rho} 2\sigma_s(\mathbf{x})_1 + \frac{4\tau}{1 - \rho}\eta \\
&= \frac{1 + \left(\frac{2\theta_{2s}}{1 - 4\theta_{2s}}\right)}{1 - \left(\frac{2\theta_{2s}}{1 - 4\theta_{2s}}\right)} 2\sigma_s(\mathbf{x})_1 + \frac{4\left(\frac{1}{(1 - 4\theta_{2s})d}\right)}{1 - \left(\frac{2\theta_{2s}}{1 - 4\theta_{2s}}\right)}\eta \\
&= \frac{1 - 2\theta_{2s}}{1 - 6\theta_{2s}} 2\sigma_s(\mathbf{x})_1 + \frac{4}{(1 - 6\theta_{2s})d}\eta.
\end{aligned}$$

This is the upperbound on the error of the solution of $(P_{1,\eta})$ that we set out to prove. □

Unfortunately, if θ_{2s} is close to $1/6$ in the previous theorem, then the denominators of the of the error bound is close to 0. This shows that the value of the error bound becomes arbitrarily larger as θ_{2s} approaches $1/6$.

7.2 Approximating θ_s Using the Bipartite Adjacency Matrix

A Note on the Restricted Expansion Constant The previous section demonstrates why the restricted expansion constant θ_{2s} of a lossless expander

graph is a useful measure in compressed sensing. Now we address some concerns of how we go about finding an approximation θ for θ_{2s}. For any d-left regular bipartite graph, consider a modified version of the restricted expansion property

$$|R(S)| \geq (1-\theta)ds \text{ for all } S \subset [N] \text{ with } |S| = s.$$

Then, for any $S' \subsetneq S$, we have $|S'| = s' < s$, which implies

$$|R(S)| \geq (1-\theta)ds > (1-\theta)ds' \text{ and } |R(S)| \geq |R(S')|.$$

This shows the two inequalities are incomparable. Therefore, we have

$$|R(S)| \geq (1-\theta)ds \text{ for all } s \subset [N] \text{ with } |S| = s$$

does *not* in general imply

$$|R(S)| \geq (1-\theta)ds \text{ for all } S \subset [N] \text{ with } |S| \leq s.$$

This is a subtle and important point for when we make claims about approximating the restricted expansion constant of a lossless expander.

Finding θ from the Bipartite Adjacency Matrix Let A be the bipartite adjacency matrix of a d-left regular graph and let S be an arbitrary subset of s columns of A. Consider the submatrix A_S. Then $|R(S)|$ is equal to the number of nonzero rows of A_S. Thus, to find θ_s it is sufficient to find a subset S_{min} of s columns of A so that $A_{S_{min}}$ contains the minimal number of nonzero rows that a submatrix of s columns of A can have. Hence, an approximation θ says that there must be at least $(1-\theta)ds$ nonzero rows in A_S.

7.3 Graph Properties of Good Lossless Expander Graphs

Since we require a lossless expander graph to have restricted expansion constant $\theta_{2s} < 1/6$, it helps to consider what graph properties yield a small θ_{2s}. In this section, we see that the definition of lossless expanders with small θ_{2s} requires the graph to be both sparse and highly connected. These properties have a tendency to disagree with each other. Usually increasing the sparsity of the graph decreases the connectivity and vice versa. Thus constructing large lossless expander graphs is known to be a difficult task.

Sparsity of the Graph We first give a definition of the density of a left regular bipartite graph.

Definition 24 (Density of a Graph)**.** Let G be a d-left regular bipartite graph with m right vertices. Then the *density* of G is defined by

$$\rho(G) := \frac{d}{m}.$$

In the next theorem, we show that if the graph is sparse, then the restricted expansion constant must be small.

Theorem 14. *Let G be an (s, d, θ)-lossless expander with m right nodes. If the density of the graph $\rho(G)$ is small then θ is small. More precisely, for all $S \subset [N]$ with $|S| \leq s$, we have*

$$\frac{1}{(1-\theta)|S|} \geq \rho(G).$$

Proof. Since any of subset S of left vertices satisfies $|R(S)| \leq m$, it follows from the restricted expansion property that

$$m \geq |R(S)| \geq (1-\theta)d|S|.$$

This is if and only if

$$\frac{1}{(1-\theta)|S|} \geq \rho(G)$$

If the graph is sparse, then $\rho(G)$ is small and this forces θ to be small. □

This proves that sparse d-left regular graphs are better quality lossless expander graphs.

Connectivity of the Graph The fact that highly connected d-left regular graphs make higher quality measurement matrices follows directly from the definition of the restricted expansion property and some intuition. Observe that for all $S \subset [N]$ with $|S| \leq s$, if θ is small then

$$|R(S)| \geq (1-\theta)d|S|$$

requires every subset S of s left vertices to have many neighbors. This occurs when the graph is highly connected. The property that every small subset of left vertices has a large set of neighbors is referred to as *expansion* which is the motivation for the name lossless "expander" graphs.

8 Combinatorial Design Theory as a Source of Measurement Matrices

In this chapter, we take a very general approach to the quadratic bottleneck problem. We do not attempt to explicitly construct any measurement matrices that can be a solution to the quadratic bottleneck problem here. Rather, we layout a selection of large classes of discrete objects such as t-designs, Steiner systems, finite incidence structures, configurations, and generalized polygons. The incidence matrices of these classes of objects can be used as a bipartite adjacency matrix of a left regular bipartite graph.

8.1 Introduction to Combinatorial Design Theory

Combinatorial design theory is the study of the intersection properties of systems of sets. Since in compressed sensing we are working with matrices and not systems of sets, the applications of combinatorial design theory in compressed sensing is not immediately clear. We address this ambiguity first by introducing one of the most general classes of combinatorial designs and then define a corresponding matrix.

Definition 25 (t-designs)**.** A t-(v, k, λ) *design*, or t-*design*, is an ordered pair $(V, \mathcal{B})$, where the v elements of V are called the *points* and the b elements of $\mathcal{B}$ are called the *blocks*. The blocks are k-subsets of points. Any t-subset of points is contained in exactly λ blocks.

Definition 26 (Incidence Matrix)**.** The *incidence matrix* of a t-(v, k, λ) design is the $v \times b$ binary matrix A defined by $a_{i,j} = 1$ if the ith point is incident to the jth block and $a_{i,j} = 0$ otherwise.

All of the combinatorial designs defined throughout this chapter have a corresponding incidence matrix. It follows from the previous definitions that the incidence matrix of a t-(v, k, λ) design can be viewed as the bipartite adjacency matrix of a k-left regular bipartite graph with v left vertices, b right vertices, and has the property that every set of t right vertices is contained in the neighborhood of λ individual left vertices. When an incidence matrix of a combinatorial design is used in this way, we have the following definition.

Definition 27 (Levi Graph)**.** Let A be the incidence matrix of a combinatorial design. Then the bipartite matrix G associated with the bipartite adjacency matrix A is called the *Levi graph* of the combinatorial design.

We can take a moment to consider an example. Define the set of points $V = \{0, \ldots, 6\}$ and the set $\mathcal{B}$ of blocks $\{0,1,2\}$, $\{0,3,4\}$, $\{0,5,6\}$, $\{1,3,5\}$, $\{1,4,6\}$, $\{2,3,6\}$, $\{2,4,5\}$. Then $(V, \mathcal{B})$ is a 2-$(7,3,1)$ design. The incidence matrix of this design is

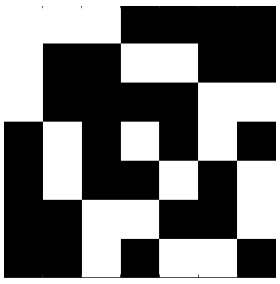


Figure 8.1: Incidence Matrix of the Fano Plane

This block design known as the *Fano plane* and is good for building intuition because it is one of the simplest t-designs. Constructing a t-(v,k,λ) design with large parameters is not trivial. For small parameters, databases of t-(v,k,λ) designs exist, many of which are in the earlier chapters of [11]. However, many of these are not easy to describe, analyze, and don't offer much in the way of generalizing to larger matrices. For t-designs to be useful for our applications, we will require these parameters to be large, so it helps to begin by narrowing down what parameters cannot be used for t-designs. We first present a necessary condition for the existence of t-(v,k,λ) designs in the following theorem.

Theorem 15 ([11], Theorem I 4.8)**.** *Let (V,B) be a t-(v,k,λ) design and let S be an s-subset of points with $s \leq t$. Define λ_s to be the number of blocks that contain S as a subset. Then*

$$\lambda_s = \lambda \frac{\binom{v-s}{k-s}}{\binom{k-s}{t-s}}.$$

Proof. Observe that the number of t-subsets of points that contains the s-subset S is $\binom{v-s}{t-s}$. Since every t-subset of points is contained in exactly λ blocks, there exists at most $\lambda\binom{v-s}{t-s}$ blocks that contain S. Let B be an arbitrary block that contains S as a subset. Since B is a k-subset of points and $t \leq k$, there exists $\binom{k-s}{t-s}$ t-subsets of B which contains the subset S. Since B was arbitrary, we arrive at

$$\lambda_s = \lambda \frac{\binom{v-s}{t-s}}{\binom{k-s}{t-s}} \tag{8.1}$$

blocks that contain S as a subset. □

Note that the number of blocks in a t-design is precisely $b = \lambda_0$ and the number of blocks a single point is contained is $r = \lambda_1$. The necessary condition for t-designs that (8.1) must hold for all $s \leq t$ is sometimes called the *divisibility condition*. The reasoning for the divisibility condition being a necessary condition is as follows. If there exists an $s \leq t$ such that $\lambda\binom{v-s}{t-s}/\binom{k-s}{t-s}$ is not

an integer, then λ_s is not an integer. This implies that a t-design with these parameters can not exist since the number of blocks that contain an s-set S must be an integer. With this in mind, we introduce the following definition.

Definition 28 (Admissible Designs)**.** Let (λ, t, v, k) be an ordered quadruple of positive integers. Then the parameters (λ, t, v, k) are *admissible*, denoted t-(v, k, λ), if $\lambda\binom{v-s}{t-s}/\binom{k-s}{t-s}$ is an integer for every $s \leq t$.

Finding a t-(v, k, λ) design for an admissible t-(v, k, λ) is not trivial. A brute force approach of filling in 1's in a matrix in a way that builds a t-(v, k, λ) design takes exponential time to compute, as stated in remark VII 6.5 of [11]. We introduce another definition for parameters of a t-design that have been constructed.

Definition 29 (Realizable Designs)**.** An admissible t-(v, k, λ) is *realizable* if a t-(v, k, λ) design exists.

Since it is hard to construct t-designs in general, many infinite families of t-designs arise from other combinatorial, algebraic, or geometric objects and are usually defined by an incidence relation. Otherwise, to find if an admissible t-(v, k, λ) is realizable, heuristics such as backtracking, hill climbing, and simulated annealing can be used as well as other methods listed in table VII 6.7 of [11].

The class of t-designs are among the most general class of combinatorial designs. Since lossless expander graphs are sparse, restricting our attention to the following subclass of t-designs is useful.

Definition 30 (Steiner Systems)**.** A t-$(v, k, 1)$ design is called a *Steiner system* and is denoted by $S(t, k, v)$.

In our original example at the start of this chapter, the Fano plane is also an $S(2, 3, 7)$ Steiner system.

8.2 The Incidence matrix of a Steiner System as a Measurement Matrix

Since the Fano plane is a square matrix, it would not make a good measurement matrix. However, if we consider a different set parameters for a Steiner system, we can obtain the rectangular matrices we require for compressed sensing. Consider a $S(2, 5, 125)$ Steiner system. These parameters are admissible since

$$\lambda_0 = \frac{\binom{125}{2}}{\binom{125}{2}} = 775 \quad \text{and} \quad \lambda_1 = \frac{\binom{125}{1}}{\binom{125}{1}} = 25$$

are both integers. Since $b = \lambda_0$, the incidence matrix of this Steiner system is a rectangular 125×775 matrix. With this example in mind, we look further into the details of the incidence matrix of Steiner systems as measurement matrices by computing the coherence and the ℓ_1-coherence function.

Theorem 16. *Let A be the incidence matrix of a $S(t, k, v)$ Steiner system. Then the coherence of A is*

$$\mu(A) = \frac{t-1}{k} \tag{8.2}$$

and the ℓ_1-coherence function of A is

$$\mu_1(A, s) = \frac{(t-1)s}{k} = \mu s \tag{8.3}$$

for all $s \leq \binom{k}{t-1}(\lambda_{t-1} - 1)$.

Proof. Since A does not have ℓ_2-normalized columns, we use

$$k \cdot \mu(A) = \max_{1 \leq i < j \leq b} \langle \mathbf{a}_i , \mathbf{a}_j \rangle$$

for the coherence and

$$k \cdot \mu_1(A, s) = \max_{i \in [b]} \max\{\sum_{j \in S} |\langle \mathbf{a}_i , \mathbf{a}_j \rangle| : S \subset [b] \text{ with } |S| = s \text{ and } i \notin S\}.$$

for the ℓ_1-coherence function of a binary matrix A with k ones in each column.

First observe that the blocks of a Steiner system correspond to the columns of its incidence matrix in the following way,

$$|B_i \cap B_j| = \langle \mathbf{a}_i , \mathbf{a}_j \rangle \text{ for all } B_i, B_j \in \mathcal{B}.$$

Since every t-subset of points is contained in exactly one block, we have $|B_i \cap B_j| < t$ for all distinct $B_i, B_j \in \mathcal{B}$. Let T be an arbitrary $(t-1)$-subset of points. For any pair of distinct points t_1 and t_2 not contained in T, define the t-subsets of points $T_1 = T \cup \{t_1\}$ and $T_2 = T \cup \{t_2\}$. Since every t-subset of points is contained in exactly one block, there exists blocks B_1 and B_2 such that $T_1 \subset B_1$ and $T_2 \subset B_2$. Thus $T \subset B_1 \cap B_2$ which implies

$$t - 1 = |T| \leq |B_1 \cap B_2| < t.$$

Therefore, we have

$$k \cdot \mu(A) = \max_{1 \leq i < j \leq b} \langle \mathbf{a}_i , \mathbf{a}_j \rangle = \max_{1 \leq i < j \leq b} |B_i \cap B_j| = t - 1.$$

Dividing by k, we have

$$\mu(A) = \frac{t-1}{k},$$

which completes the proof for the first part of the theorem.

Next, observe the second part of the theorem is proved if we show that for any block B, there exists $\binom{k}{t-1}(\lambda_{t-1} - 1)$ other blocks that intersect B with cardinality $t-1$. Let T be an arbitrary $(t-1)$-subset of B. Since there are λ_{t-1} blocks that contain T, there are $\lambda_{t-1} - 1$ blocks that intersect B with intersection T. Since there are $\binom{k}{t-1}$ ways to choose the $(t-1)$-subset T of B, there are $\binom{k}{t-1}(\lambda_{t-1} - 1)$ other blocks that intersect B with cardinality $t-1$. Thus

$$k \cdot \mu_1(A, s) = (t-1)s$$

for all $s \leq \binom{k}{t-1}(\lambda_{t-1} - 1)$. □

Using theorem 10 and theorem 16, if the measurement matrix is the incidence matrix of an $S(t,k,v)$ Steiner system then every s-sparse vector is the unique s-sparse solution of basis pursuit if s satisfies

$$1 > \mu_1(A,s) + \mu_1(A,s-1) = \frac{(t-1)s + (t-1)(s-1)}{k} = \frac{(t-1)(2s-1)}{k}.$$

To see how well this works, we can use the example $S(2,5,125)$ Steiner system above. By substituting in the values for t and k, this implies every s-sparse vector with $s < 3$ is the unique solution of basis pursuit. This result is useless since s is so small, but it does provide a little more motivation as to why we want to avoid depending on the ℓ_1-coherence function and find solutions to the quadratic bottleneck problem.

To improve on this, we need to use an approximation θ for the restricted expansion constant θ_{2s} of the Levi graph of an $S(t,v,k)$ Steiner system. To obtain an approximation that yields better results than the quadratic bottleneck, we must know more about how every subset of $2s$ columns behave. This is very difficult to analyze in practice because it requires a very convenient incidence relation. Although obtaining an incidence relation that makes analysis possible, it is not clear that such a class of Steiner systems exist.

The intuition of using Steiner systems is that they minimize the compression ratio m/N of the class of measurement matrices with coherence $\mu(A) = (t-1)/k$ and A is the bipartite adjacency matrix of a k-left regular bipartite graph. The approach of minimizing the compression ratio at first seems to be the best approach to take. However, the compression ratio does not take into consideration the sparsity of the vector which is a very important factor. Thus, it is not required that the compression ratio m/N of a measurement matrix needs to be minimized to be a solution to the quadratic bottleneck problem and obtain the optimal lowerbound on the number of rows required

$$m \geq Cs \log(N/s)$$

for a small constant C that does not depend on s. Because of this, Steiner systems can be considered rigid in their construction without much benefit in return.

8.3 Finite Incidence Structures and Configurations

In this section, we will turn our attention to classes of combinatorial designs that are similar to the t-designs and Steiner systems defined in the first section of this chapter. However, these combinatorial designs are more flexible in their definition. The classes of combinatorial designs defined in this section are also common in finite geometry.

Definition 31 (Finite Incidence Structures)**.** A *finite incidence structure* is an ordered triple (P,L,I), where the elements of the finite set P are called the *points* of the incidence structure, the elements of the finite set L are called the *lines* of the incidence structure, and I is the *incidence relation* between the points and the lines.

Since the incidence matrix of a t-design can be an incidence matrix of a finite incidence structure, the class of incidence matrices of finite incidence structures

are more general then the class of incidence matrices of t-designs. Thus, the definition of the incidence matrix of a finite incidence structure without any added constraints is far too general to use as a possible measurement matrix. The next definition narrows things down significantly.

Definition 32 (Configurations)**.** A *configuration* (v_r, b_k) is an incidence structure (P, L, I) with $v = |P|$ and $b = |L|$ such that each of the lines is incident to k points, each point is incident to k lines, and any pair of points are incident to at most one line.

The incidence matrix of a $S(2, k, v)$ Steiner system is a (v_r, b_k) configuration, where $b = \lambda_0$ and $r = \lambda_1$ follows from theorem 15. Since we have more flexibility in the choice of b and r in the definition of a configuration than in Steiner systems, the class of incidence matrices of configurations are a generalization of the class of incidence matrices of $S(2, k, v)$ Steiner systems. The fact that $t = 2$ in this Steiner system is a useful restriction since this would require its Levi graph to be more sparse then if $t > 2$.

In the next theorem, we prove that the Levi graph of a configuration does not have small girth.

Theorem 17 ([11], Remark VI 7.9)**.** *The matrix A is the incidence matrix of a (v_r, b_k) configuration if and only if A is the bipartite adjacency matrix of the biregular graph $G_{r,k}$ with b left vertices, v right vertices, and*

$$girth(G_{r,k}) \geq 6.$$

Proof. Suppose A is the incidence matrix of a (v_r, b_k) configuration. Let G be the Levi graph of A. Then G has v right vertices and b left vertices since A is a $v \times b$ matrix. Since each column of A has k ones and each row of A has r ones, this implies that the Levi graph G is a biregular graph and we can use the notation $G_{r,k}$. Since every pair of points is incident to at most 1 line in the configuration, there cannot exist a submatrix of A of the form

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

In terms of the Levi graph $G_{r,k}$, this implies that a cycle of length 4 cannot exist. Lemma 1 and the assumption that G is a simple graph implies that cycles of odd length and cycles of length 2 cannot exist as a subgraph of G. Therefore,

$$\text{girth}(G_{r,k}) \geq 6.$$

This proves the first direction of the theorem. The proof of the reverse direction of the theorem can be obtained in a similar manner. □

A necessary condition for the parameters of configurations, similar to the divisibility condition of t-designs, exists. We present this in the next theorem

Theorem 18 ([11], Remark VI 7.10)**.** *If a (v_r, b_k) configuration exists, then the parameters must satisfy*

$$vr = bk.$$

Proof. Let $G_{r,k}$ be the Levi graph of the configuration (v_r, b_k). Since every edge must have one end in the left vertex set and one end in the right vertex set, the sum of the degrees of the left vertex set must equal the sum of the degrees of the right vertex set. This gives us the conclusion

$$vr = bk.$$

□

Notice that the necessary condition of configurations in the previous theorem is much more flexible then the divisibility condition of t-designs. Intuitively, this means that it is much easier to build configurations than it is to build t-designs.

Finally, we note that configurations are similar to Steiner systems with $t = 2$ in that every pair of points is incident to at most one line (every block). This and theorem 16 implies that the coherence of the incidence matrix of a (v_r, b_k) configuration satisfies $\mu = 1/k$.

8.4 Finite Generalized Polygons

In the previous section, we claimed that the incidence matrix of a configuration is sparse relative to the other classes of combinatorial designs and is relatively flexible since it can be viewed as a less restrictive Steiner system. However, the definition of a configuration does not require its Levi graph to have any connectivity properties. Since a good lossless expander graph is not only sparse, but also highly connected, we address this problem in this section.

The definitions introduced in this section are done so in a way that is convenient for the rest of this thesis. More technical definitions exist for when the reader is working within the context of finite geometry.

Definition 33 (Finite Generalized Polygon)**.** A *Finite generalized n-gon of order* (s, t) is an incidence structure (P, L, I) with the following properties. Every point is incident to exactly $t + 1$ lines and every pair of distinct points are incident to at most one line. Every line is incident to exactly $s + 1$ points and ever pair of distinct lines are incident to at most one point. Lastly, the Levi graph of the generalized n-gon has girth $2n$ and diameter n.

When the number n is not relevant to the discussion, we will refer to the finite generalized n-gon as a *finite generalized polygon*

Unfortunately, not every value of n implies that a finite generalized n-gon exists. It is known from [16] that we can only work with the values $n = 3, 4, 6, 8$. Thus, the intuitive names are assigned to each n-gon. That is, we refer to a generalized n-gon as a *generalized triangle*, *generalized quadrangle*, *generalized hexagon*, and *generalized octagon* for $n = 3, 4, 6$, and 8, respectively.

The incidence matrices of generalized triangles and of generalized quadrangles are of primary interest in the next chapter. However, the incidence matrices of generalized hexagons and generalized octagons are likely useful and a good source of future research.

The next definition defines a subclass of generalized triangles. We will use the incidence matrix of this subclass of generalized triangles in the next chapter.

Definition 34 (Finite Projective Planes)**.** Let q be a prime number. A *finite projective plane* of *order* q is a generalized triangle of order (q, q).

A finite projective plane of order q is also a $(\, q^2 + q + 1_{q+1}\, ,\, q^2 + q + 1_{q+1})$ configuration with the property that every two lines are also incident with exactly one point. Similarly, its incidence matrix is also the incidence matrix of a $S(2, q + 1, q^2 + q + 1)$ Steiner system. The Fano plane is an example of a finite projective plane of order 2.

Large Girth is a Sufficient Condition Consider the Levi graph G of a finite generalized n-gon. Since G has girth $2n$, we claim this makes G sparse and highly connected. We make this claim by showing that the local structure of G is sparse and highly connected and then we conjecture that this must effect the way larger portions of the graph must behave. We take this approach because it is not obvious on how to do this deterministically. However, in the next chapter we will present a few ways one might be able to approximate how this in a more rigorous way for similarly defined graphs.

First, we consider the connectivity properties of G by considering its expansion properties. Consider a subset S of $n - 1$ left vertices. Then a cycle cannot exist in the induced subgraph $G[S \cup R(S)]$, since this would require n left vertices. Intuitively, this implies that the right vertices of S cannot be too bunched together. That is, $|R(S)|$ must be large. Even though this is for a relatively small subset of left vertices, this does have an impact on how the size of $|R(S)|$ scales as the subset of s left vertices increases. For larger subsets of left vertices, it is conjectured that the expansion of G is dependent on the number of cycles that can exist in $G[S \cup R(S)]$.

Next, we consider the sparsity of G. Consider a cycle C of length $2n$. Since $\text{girth}(G) = 2n$, no more edges can exist with both ends in this cycle in the graph G. Since there are n^2 edges that can exist in a bipartite graph with n left vertices and n right vertices, we have shown

$$\rho(C) = \frac{2n}{n^2} = \frac{2}{n}.$$

Thus,

$$\rho(G) \leq \frac{2}{n}.$$

This bound is not great because this does not prove that the graph is very sparse. However, when we consider that the fact that there are many cycles of length $2n$ in G, this implies that we end up deleting at most $2/n$ edges for each cycle.

Notice that it is not shown here to be necessary here for a left regular graph to have large girth. It could be possible to build a lossless expander graph with small restricted expansion constant where the girth of the graph is not a factor.

Finite Generalized Polygons are Still not Desirable. There are some negative qualities of using the Levi graph of a finite generalized n-gon as a lossless expander graph in compressed sensing. The first of which is that they are very rigid in their construction, so they are poor at adapting to different situations with different sparsity or different vector length. The second being that many of the constructions of finite generalized n-gons that exist do not have $m \ll N$. We will handle this problem in the next chapter.

9 Explicit Constructions of Two Infinite Families of Biregular Graphs

Last chapter we took a very general approach to the quadratic bottleneck problem and laid out many classes of similar discrete structures whose Levi graph may be used as a lossless expander graph. In this chapter, we commit to one of these approaches. The approach used in this chapter has been chosen for two reasons. The first of which is because of its nice algorithmic properties which will be discussed in the third part of this thesis. The second reason is because we can define two very similarly defined classes of matrices that, after some numerical testing, we can conjecture have very different reconstruction guarantees.

In the first section, we define these two classes of biregular graphs as well as present some immediate facts of their immediate properties. In the second section, we define some of the structure of these graphs. Then, we demonstrate a necessary condition on the sparsity of a vector for sparse recovery to be possible. In the third section, we prove the girth of the constructions are not small and prove that these graphs are the Levi graphs of configurations. In the final section, we show the constructions relationship to generalized polygons.

9.1 The Two Constructions

For both of the constructions, we will need to use the following notation. If p is a prime number, then define the set $P := \{0, \dots, p-1\}$. Similarly, if d is a natural number, then define the set $D := \{0, \dots, d-1\}$.

First Construction ([23] and [22], Def 1)**.** Let p be a prime number and d be an integer such that $2 \leq d \leq p$. Define the biregular graph $G^{[1]}_{p,d}$ in the following way. Label the left vertices by the ordered pairs of $P \times P$ and label the right vertices by the ordered pairs of $D \times P$. A left vertex (x, y) is adjacent to a right vertex (a, b) if and only if

$$y = ax + b \pmod{p}.$$

When the parameters d and p are not relevant to the discussion, we abbreviate the notation by $G^{[1]}$.

Notice that $G^{[1]}_{p,d}$ is indeed a biregular graph. To see this, let (x, y) be an arbitrary left vertex. Then for any $a \in D$, there exists a unique $b \in P$ such that $b = y - ax \pmod{p}$. Since the cardinality of D is d, this implies that

every left vertex has degree d. A similar argument can be made for the right vertices.

Second Construction ([23] and [22], Def 2)**.** Let p be a prime number and d be an integer such that $2 \leq d \leq p$. Define the biregular graph $G^{[2]}_{p,d}$ in the following way. Label the left vertices by the ordered triples $P \times P \times P$ and label the right vertices by the ordered triples $D \times P \times P$. A left vertex (x, y, z) is adjacent to a right vertex (a, b, c) if and only if

$$y = ax + b \pmod{p} \quad \text{and} \quad z = ay + c \pmod{p}.$$

When the parameters d and p are not relevant to the discussion, we abbreviate the notation by $G^{[2]}$.

Similarly, we can observe that $G^{[2]}_{p,d}$ is indeed a biregular graph. Let (x, y, z) be an arbitrary left vertex. Then for any $a \in D$, there exists a unique $b \in P$ such that

$$b = y - ax \pmod{p}.$$

Solving for y in the previous equation implies that there exists a unique c such that

$$c = ay - z \pmod{p}.$$

Since the cardinality of D is d, this implies every left vertex has degree d. The same argument holds for the degrees of the right vertices.

The proofs for the $G^{[1]}$ and $G^{[2]}$ usually depend on the linear equations in their definitions. Since the two constructions are so similar in their definition, many of their proofs are identical. In this case, we will not be redundant and list the same argument twice.

Notice that the parameters $2 \leq d \leq p$ in both the $G^{[1]}_{p,d}$ and $G^{[2]}_{p,d}$ determine the number of rows m and the number of columns N of its corresponding bipartite adjacency matrix. In $G^{[1]}_{p,d}$, we have $m = dp$ since there are dp right nodes and $N = p^2$ since there are p^2 left nodes. In $G^{[2]}_{p,d}$, we have $m = dp^2$ and $N = p^3$ which follows from the same reasoning. When we talk about the bipartite adjacency matrices of $G^{[1]}_{p,d}$ and $G^{[2]}_{p,d}$, we will not mention this fact every time.

Notice that it follows from the dimensions of the adjacency matrices that for any given parameters $2 \leq d \leq p$, the number of vertices in $G^{[2]}_{p,d}$ is much larger than the number of vertices in $G^{[1]}_{p,d}$. This shows that even though the adjacency conditions of these two constructions are very similar, these two graphs have very different properties. However, one of their similarities is that both constructions have the same left and right degrees. Intuitively, this implies $G^{[2]}_{p,d}$ is much sparser then $G^{[1]}_{p,d}$. Indeed, we have

$$\rho(G^{[1]}_{p,d}) = \frac{d}{dp} = \frac{1}{p} \quad \text{and} \quad \rho(G^{[2]}_{p,d}) = \frac{d}{dp^2} = \frac{1}{p^2}.$$

9.2 Necessary Condition for Sparse Recovery

Figures 9.1 and 9.2 are examples of the bipartite adjacency matrices of the first and second construction with small parameters p and d. These figures are

included because the adjacency conditions alone do not provide much intuition of the behavior of the graph.

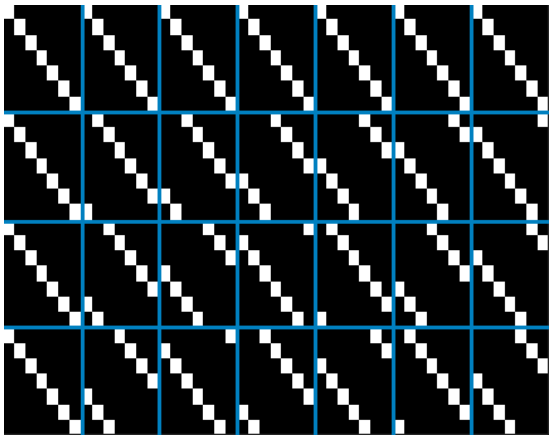


Figure 9.1: Bipartite Adjacency Matrix of $G^{[1]}_{7,4}$

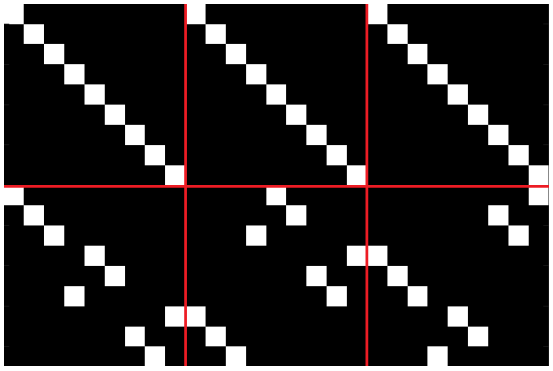


Figure 9.2: Bipartite Adjacency Matrix of $G^{[2]}_{3,2}$

The lines partitioning the matrices demonstrate some of the structure of the graphs and their bipartite adjacency matrices. The next definition makes these partitions clear.

Definition 35 (Blocks)**.** Let p be a prime number and d be an integer such that $2 \leq d \leq p$. Let G denote the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$. For any $x \in P$, the *left vertex block* L_x of G is the set of left vertices that have x as its first coordinate. If A is the bipartite adjacency matrix of G then the set of column vectors of A corresponding to the left vertices in the left vertex block L_x is called the *column block* corresponding to L_x. Likewise, for any $a \in D$, the *right vertex block* R_a of G is the set of all right vertices that have a as its first coordinate. The set of row vectors corresponding to the right vertices in the right vertex block R_a is called the *row block* corresponding to R_a.

In the figures, the partitioned column vectors denote the column blocks and the partitioned row vectors denote the row blocks. With these definitions, we can obtain a first result that is relevant to compressed sensing.

Theorem 19. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. If A is the bipartite adjacency matrix of $G^{[1]}_{p,d}$ then*

$$spark(A) \leq 2p.$$

Similarly, if A is the bipartite adjacency matrix of $G^{[2]}_{p,d}$ then

$$spark(A) \leq 2p^2.$$

Proof. Let A be the bipartite adjacency matrix of $G^{[1]}_{p,d}$. Denote the column block corresponding to the left vertex block L_x by A_x, for any $x \in P$. Note that there are p left vertices in L_x since

$$\{x\} \times P$$

has p elements. Denote $\mathbf{1}_n$ by the column vector of ones of length n and denote $\mathbf{0}_n$ by the column vector of zeros of length n. It can be verified that

$$A_x \mathbf{1}_p = \mathbf{1}_m.$$

Next, define the vector

$$\mathbf{z} := [\mathbf{1}_p; -\mathbf{1}_p; \mathbf{0}_p; \ldots; \mathbf{0}_p].$$

Then

$$A\mathbf{z} = \mathbf{0}.$$

Since $\| \mathbf{z} \|_0 = 2p$, we have

$$\text{spark}(A) \leq 2p.$$

This proves the first part of the theorem. A similar argument for the bipartite adjacency matrix A of $G^{[2]}_{p,d}$ can be made with the only change being that every column block has the length p^2 instead of p. This implies that

$$\text{spark}(A) \leq 2p^2,$$

which proves the second part of the theorem. □

Using theorem 2 and theorem 19, we obtain a necessary condition on recovery of s-sparse vectors.

Theorem 20. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Let $\boldsymbol{x}$ be an s-sparse vector. If A is the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ (the graph $G^{[2]}_{p,d}$) and $\boldsymbol{x}$ is the unique s-sparse solution of $A\boldsymbol{z} = \boldsymbol{y}$ where $\boldsymbol{y} = A\boldsymbol{x}$, then the sparsity must satisfy $s < p$ ($s < p^2$).*

Proof. Suppose A is the bipartite adjacency matrix of $G^{[1]}_{p,d}$ and $\mathbf{x}$ is the unique s-sparse solution $A\mathbf{z} = \mathbf{y}$ with $\mathbf{y} = A\mathbf{x}$. Then theorem 2 and theorem 19 implies

$$2s < \text{spark}(A) \leq 2p.$$

This implies $s < p$, which completes the first part of the proof. The second part of the proof follows from the same argument, so the proof is complete. □

This does *not* imply we obtain unique recovery for all $s < p$ and $s < p^2$, respectively. That is, we cannot guarantee unique recovery for every s that satisfies these bounds. Notice that the necessary condition for the second construction is much larger then the necessary condition of the first construction.

9.3 Girth of $G^{[1]}$ and $G^{[2]}$

Next, motivated by the discussion at the end of the previous chapter, we evaluate the girth of the two constructions.

Theorem 21 ([23] and [22])**.** *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Then girth$(G^{[1]}_{p,d}) = 6$ and girth$(G^{[2]}_{p,d}) = 8$*

Proof. First consider the graph $G^{[1]}_{p,d}$. Since $G^{[1]}_{p,d}$ is a simple bipartite graph, only cycles of even length greater then 2 can exist. Suppose there exists a cycle of 4. Then there exists distinct right vertices (a, b) and (a', b') both adjacent to two distinct left vertices (x, y) and (x', y'). By the definition of $G^{[1]}_{p,d}$, we have the following equations

$$y = ax + b \qquad y = a'x + b' \qquad y' = ax' + b \qquad y' = a'x' + b' \pmod{p}.$$

By solving these equations for x and x', we obtain

$$x = (a - a')^{-1}(b' - b) = x' \pmod{p}.$$

Substituting $x = x'$ into the original equations give us

$$y = ax + b = ax' + b = y' \pmod{p}.$$

Therefore, every pair of right vertices have a unique left vertex in common. This implies a cycle of length 4 can't exist in $G^{[1]}_{p,d}$. Next, consider the left vertices $(0, 0)$, $(3, 0)$, and $(1, 1)$ and the right vertices $(0, 0)$, $(2, 4)$, and $(1, 0)$. Substituting these vertices into the adjacency condition of G demonstrates that this is a cycle of length 6. Therefore,

$$\text{girth}(G^{[1]}_{p,d}) = 6.$$

Next, consider the graph $G^{[2]}_{p,d}$. Since the adjacency condition of $G^{[2]}_{p,d}$ is the adjacency condition of $G^{[2]}_{p,d}$ with an additional linear equation to be satisfied, this shows that

$$\text{girth}(G^{[2]}_{p,d}) \geq 6.$$

For the rest of this proof, if two vertices v_1 and v_2 are adjacent, then we will denote this by $v_1 \sim v_2$. Let (x, y, z) be an arbitrary left vertex and (a, b, c) be an arbitrary right vertex such that $(x, y, z) \nsim (a, b, c)$. Suppose that there exists a path of length 3 in between (x, y, z) and (a, b, c). If we can show that this path must be the unique path of length 3 in between (x, y, z) and (a, b, c), then we have shown

$$\text{girth}(G^{[2]}_{p,d}) \geq 8.$$

This is because a cycle of length 6 would require two paths of length 3 in between (x, y, z) and (a, b, c). Since we have assumed that there exists a path

of length 3, we can assume that there exists a right vertex (e, f, g) and a left vertex (u, v, w) such that

$$(x, y, z) \sim (e, f, g) \sim (u, v, w) \sim (a, b, c).$$

By the definition of $G^{[2]}_{p,d}$, we have

$$y = dx + e \text{ and } z = dy + f \tag{9.1}$$
$$v = du + e \text{ and } w = dv + e \tag{9.2}$$
$$v = au + b \text{ and } w = av + c. \tag{9.3}$$

We count all of the paths of length 3 that can exist between (x, y, z) and (a, b, c). The variables x, y, z, a, b, and c must remain fixed and the variables e, f, g, u, v, and w can take on any set of values that are consistent with the above equations.

Let e and u vary. Then it follows form equation (9.1) and equation (9.2) that d is a bijective function of e and v is a bijective function of u. We can denote these functions by $d = \psi(e)$ and $v = \varphi(u)$. Then it follows from equation (9.2) that

$$\varphi(u) = v = du + e = \psi(e)u + e$$

Since the functions φ and ψ are bijective, there exists only one choice of e and u that satisfies this equation. By applying this procedure to the pairs f, v and g, w of variables, we show that there is only one choice of e, f, g, u, v, and w that satisfies the equations (9.1), (9.2), and (9.3). Therefore, the path of length 3 in between (x, y, z) and (a, b, c) is unique and we have shown that

$$\text{girth}(G^{[2]}_{p,d}) \geq 8.$$

By substituting the left vertices $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 1)$ and $(1, 1, 0)$ and the right vertices $(0, 0, 0)$, $(1, 1, 1)$, $(0, 1, 1)$, and $(-1, 0, 0)$, where -1 is the additive inverse of 1 in mod p, into the adjacency condition of $G^{[2]}_{p,d}$, we see that these vertices form a cycle of length 8. Thus,

$$\text{girth}(G^{[2]}_{p,d}) = 8$$

and the proof is complete. □

Theorem 21 and theorem 17 gives us the following corollary.

Corollary 1. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. If A is the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$, then A is the incidence matrix of a (m_p, N_d) configuration.*

The next theorem follows directly from the previous corollary.

Theorem 22. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Let A be the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$. If we ℓ_2-normalize the columns of A by multiplying every entry of A by $1/\sqrt{d}$, then the coherence satisfies*

$$\mu(A) = \max_{1 \leq i < j \leq N} \left\langle \frac{\boldsymbol{a}_i}{\sqrt{d}}, \frac{\boldsymbol{a}_j}{\sqrt{d}} \right\rangle = \frac{1}{d}.$$

Theorem 22 will be used in later sections to prove an approximation for the restricted expansion constant of the bipartite adjacency matrix of both the graph $G^{[1]}_{p,d}$ and the graph $G^{[2]}_{p,d}$.

9.4 $G^{[1]}$ and $G^{[2]}$ Relationship to Generalized Polygons

We now ask the question if either of these constructions are finite generalized polygons. In both of these situations, this turns out not to be the case. We prove this in the next theorem.

Theorem 23. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Then the graphs $G^{[1]}_{p,d}$ and $G^{[2]}_{p,d}$ are not Levi graphs of finite generalized polygons.*

Proof. Let $G := G^{[1]}_{p,d}$. Since $\text{girth}(G) = 6$, it only makes sense to check if G is the Levi graph of a finite generalized triangle. We proceed with a proof by contradiction. Suppose G is the Levi graph of a finite generalized triangle. Recall that the Levi graph of a finite generalized triangle has diameter 3. This implies that every pair of vertices in G are vertices in a cycle of length 6. Since $d \leq p$, there exists a pair of vertices that are not in a cycle of length 6. This is a contradiction. Therefore, G is not a finite generalize triangle. Proving $G^{[2]}_{p,d}$ is not a generalized quadrangle is identical to the proof for the first part of the theorem. This completes the proof. □

This theorem actually turns out to be a positive result. Recall the advantages and the disadvantages of using finite generalized polygons at the end of the previous chapter. We claimed that a graph with large girth is a sufficient condition for a good lossless expander graph. Since the girth of $G^{[1]}$ is equal to the girth of the Levi graph of a generalized triangle, we deduce that the reconstruction properties of $G^{[1]}$ scale about as well as that of the Levi graph of a finite generalized triangle. Similarly, since the girth of $G^{[2]}$ is equal to the girth of the Levi graph of a generalized quadrangle, we deduce that the reconstruction properties of $G^{[2]}$ scale about as well as that of the Levi graph of a finite generalized quadrangle. Moreover, we conjecture that the reconstruction properties of $G^{[2]}$ is better then the reconstruction properties of $G^{[1]}$ since

$$\text{girth}(G^{[2]}) > \text{girth}(G^{[1]})$$

The disadvantages of using finite generalized polygons is that they are rigid in their construction. However, $G^{[1]}$ and $G^{[2]}$ are very flexible. This is because we can change the number of rows the bipartite adjacency matrix and keep the number of columns fixed by just changing the value of the left degree d. Also recall the disadvantage that generalized n-gons only exist for $n = 3, 4, 6$, and 8. It is possible that classes of graphs similar to $G^{[1]}$ and $G^{[2]}$ exist and have larger girth. This shows that $G^{[1]}$ and $G^{[2]}$ avoids the disadvantages of generalized polygons while preserving the benefits.

Here we will show that the bipartite adjacency matrix of $G^{[1]}$ is the submatrix of the incidence matrix of a finite projective plane. We demonstrate this in the next theorem by constructing the incidence matrix of a finite projective plane from the bipartite adjacency matrix of $G^{[1]}$.

Theorem 24. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Then the bipartite adjacency matrix A of the graph $G^{[1]}_{p,d}$ is the submatrix of the incidence matrix of a finite projective plane of order p.*

Proof. Let B be the bipartite adjacency matrix of $G_{p,p}$. If I is the identity matrix of order p and $\mathbf{1}$ is the column vector of p ones, then define the matrix

$$C := I \otimes \mathbf{1},$$

where $\otimes$ is the Kronecker product. Let E be the square matrix of order $p+1$ defined by

$$e_{i,j} = \begin{cases} 1, & \text{if } i = p+1 \text{ or } j = p+1 \\ 0, & \text{otherwise.} \end{cases}$$

Then the matrix

$$D := \begin{bmatrix} B & C \\ C^T & E \end{bmatrix}$$

is the incidence matrix of an finite projective plane of order p. Since A is the first dp rows of B and B is a submatrix of D, it follows that A is a submatrix of D. Therefore, the bipartite adjacency matrix A of the graph $G^{[1]}_{p,d}$ is the submatrix of the incidence matrix of a finite projective plane of order p. □

We will provide an example of theorem 24. Let $p = 5$ and $d = 3$. Then the following is a figure of the incidence matrix D of a finite projective plane of order 5. Then the bipartite adjacency matrix A of $G^{[1]}_{5,3}$ is the top left submatrix outlined in red and the submatrices B, C, and E are outlined in green in the following figure.

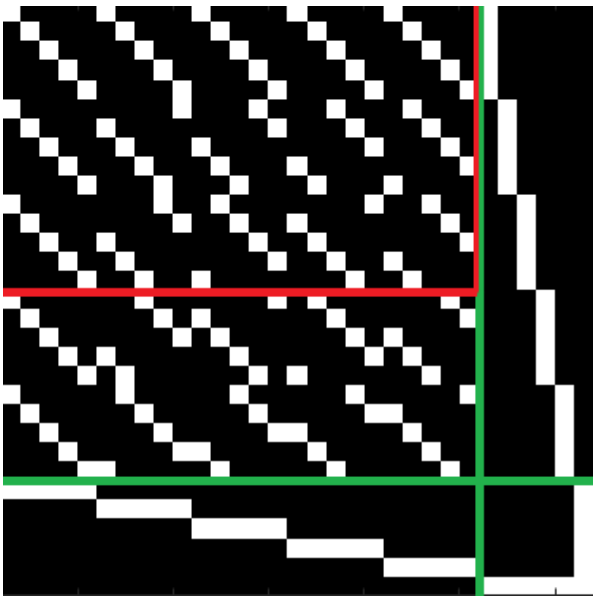


Figure 9.3: Bipartite Adjacency Matrix of $G^{[1]}_{5,3}$

The bipartite adjacency matrix of $G^{[2]}$ is likely to be the incidence matrix of a generalized quadrangle. However, a proof like the one for $G^{[1]}$ in theorem 24 has not yet been obtained. Nevertheless, we make the following conjecture.

Conjecture 1. *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Then the bipartite adjacency matrix A of the graph $G^{[2]}_{p,d}$ is the submatrix of the incidence matrix of a generalized quadrangle.*

10 Computing θ_s of $G^{[1]}$ and $G^{[2]}$

In this chapter, we discuss the restricted expansion constants of the constructions and then derive sufficient conditions on the number of columns required. We first do this by using the coherence. Following this, we consider using the adjacency conditions to compute the restricted expansion constant. How we would proceed with an approach like this is not clear, if it is even computable at all. Because of this, we follow up on the conjectures made in the previous chapters that the girth of the graph is sufficient for computing the restricted expansion constant, at least for small order. We then use these results to provide a strategy for computing larger order restricted expansion constants. This analysis has not been carried out yet so at this point it is just an approach for future research. Finally, we provide some empirical results. These empirical results are not proof, but they do turn out to be interesting.

10.1 Initial Approximation Provided by the Coherence

We start by finding an approximation θ for θ_{2s} by using the coherence of both of the constructions.

Theorem 25 ([18], Ex 13.2)**.** *Let p be a prime number and d be an integer with $2 \leq d \leq p$. Let A be the adjacency matrix of the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$. For any sparsity s, the restricted expansion constant θ_{2s} satisfies*

$$\theta_{2s} \leq \frac{2s-1}{2}\mu(A) = \frac{2s-1}{2d},$$

where $\mu(A)$ is the coherence of the ℓ-normalized columns of A.

Proof. Recall from theorem 22, that $\mu(A) = 1/d$. In terms of the bipartite matrix, this implies that every pair of left vertices have at most one right vertex in common. This gives us,

$$|R(v_1, v_2)| \geq 2d - 1 \text{ for any distinct } v_1, v_2 \in L.$$

By generalizing this to every pair of a subset of $2s$ left vertices, we obtain

$$|R(S)| \geq 2sd - \binom{2s}{2} \text{ for any } S \subset L \text{ with } |S| \leq 2s.$$

Using this bound, we solve for an approximation θ of θ_{2s} by solving for θ in the equation

$$(1-\theta)2sd = 2sd - \binom{2s}{2}.$$

This is precisely

$$\theta_{2s} \leq \theta = \frac{2s-1}{2d} = \frac{2s-1}{2}\mu(A),$$

which proves the theorem. □

Suppose we want to uniquely recover any sparse vector of a given length. Then, of course, we cannot manipulate the length of the vector or its sparsity. So the number of columns of the measurement matrix must remain fixed. However, if we have some knowledge of the sparsity of the vector, then it would be convenient to be able to easily change the number of rows of the measurement matrix. The bipartite adjacency matrices of $G^{[1]}$ and $G^{[2]}$ satisfy this property. In other words, suppose the number of columns $N = p^2$ and the sparsity of the vector s is fixed. Then p is fixed and the number of rows $m = dp$ is a function of the left degree d. We formalize this process in the next theorem.

Theorem 26. *Let p be a prime number. Suppose $\boldsymbol{x}$ is an arbitrary s-sparse vector of length $N = p^2$. Then $\boldsymbol{x}$ is the unique solution of basis pursuit with measurement vector $\boldsymbol{y} = A\boldsymbol{x}$ if the measurement matrix A is the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$ with the left degree d satisfying*

$$6s - 3 < d \leq p.$$

Proof. By the definition of the $G^{[1]}_{p,d}$, we have $d \leq p$. Recall from theorem 25 that

$$\theta_{2s} \leq \frac{2s-1}{2d}.$$

Also recall from theorem 13 that every s-sparse vector is the unique solution of basis pursuit if $\theta_{2s} < 1/6$. Thus, for $\mathbf{x}$ to be unique solution of basis pursuit, we must choose the degree d so that

$$\frac{2s-1}{2d} < \frac{1}{6}$$

holds. This is if and only if

$$6s - 3 < d$$

which proves the theorem. □

Since we always want to minimize the number of rows m, if we are limited to the bound in the previous theorem then it is always the best choice to pick the left degree

$$d = (6s-3) + 1 = 6s - 2.$$

The flexibility of the bipartite adjacency matrices of $G^{[1]}$ and $G^{[2]}$ can be compared to the incidence matrix of a finite generalized polygon, which, as we have stated several times before, is very rigid in its construction.

The next theorem confirms the fact that if we are limited to using the coherence of the bipartite adjacency matrix A of $G^{[1]}$ or $G^{[2]}$, then we cannot prove that A is a solution to the quadratic bottleneck problem. This result is not exactly interesting since it comes after discussion in chapter 4. What is interesting is that $G^{[2]}$ appears to scale terribly if we just look at the coherence.

Theorem 27. *Let p be a prime number, s be a positive integer, and d be an integer such that $d > 6s - 3$. If A is the bipartite adjacency matrix of $G_{p,d}^{[1]}$, then every s-sparse vector is recovered from basis pursuit with the measurement vector $\boldsymbol{y} = A\boldsymbol{x}$ if*

$$m \geq Cs^2,$$

for some constant $C > 0$. Similarly, If A is the bipartite adjacency matrix of an $G_{p,d}^{[2]}$, then every s-sparse vector is recovered from basis pursuit with the measurement vector $\boldsymbol{y} = A\boldsymbol{x}$ if

$$m \geq Cs^3,$$

for some constant $C > 0$.

Proof. Since we have assumed the left degree d satisfies $d > 6s - 3$, theorem 26 implies that every s-sparse vector is recovered from basis pursuit with the measurement vector $\mathbf{y} = A\mathbf{x}$. Since $G_{p,d}^{[1]}$ has $m = dp$ rows, m must satisfy

$$m = dp \geq d^2 \geq (6s - 2)^2 \geq Cs^2,$$

for some constant $C > 0$. Since $G_{p,d}^{[2]}$ has $m = dp^2$ rows, m must satisfy

$$m = dp^2 \geq d^3 \geq (6s - 2)^3 \geq Cs^3,$$

for some constant $C > 0$. This completes the proof. □

10.2 Solving Systems of Linear Equations to Find θ_s

Consider the graph $G_{p,d}^{[1]}$. That is, the left vertex (x, y) is adjacent to the right vertex (a, b) if and only if

$$y = ax + b \pmod{p}.$$

Technically, we can obtain the smallest set of right vertices of any set of s left vertices by solving a system of these equations. However, this approach has *many* problems. First of all, we can only have integer solutions. Next, these solutions must all be distinct. Third, we would need to compute a system of $\binom{s}{2}$ equations (one for each pair of left nodes to see if they have a right node in common). Fourth, since both the first coordinates a and x are unknown, it appears we may need matrix completion as well, also with distinct integer solutions. These problems only get worse when we consider this approach for the graph $G_{p,d}^{[1]}$.

Even though the simplicity of the adjacency condition seems like it could provide a way to compute θ_s, this seems like a misguided approach with a little bit of thought. By considering the difficulties of this problem, this shows the difficulties of computing the restricted expansion constant, not just in general, but also for a specific family of graphs. Because of these difficulties, we will not pursue this approach any further.

10.3 Explicitly Computing θ_s for Small s of $G^{[1]}$ and $G^{[2]}$

The girth of the $G^{[1]}$ and $G^{[2]}$ is sufficient for computing the restricted expansion constant of small order. We first consider $G^{[1]}$.

Theorem 28. *Let A be the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ with the parameters $2 \leq d \leq p$. Then the restricted expansion constant of order 3 is*

$$\theta_3 = \frac{1}{d}$$

Proof. Recall that, for any subset S of left vertices, the value of $|R(S)|$ is determined by the number of nonzero rows of A_S. Since the girth of $G^{[1]}_{p,d}$ is 6, this implies that the densest submatrix of 3 columns of A is the bipartite adjacency matrix of a cycle of length 6. Let the matrix

$$D := \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

be such a matrix. We can use this submatrix to find θ_3. Let S be a subset of columns of A so that a matrix similar to D is a submatrix of A_s. Then there is exactly $3 + 3(d-2) = 3d - 3$ nonzero rows of A_S. Thus

$$|R(S)| \geq 3d - 3 \text{ for all } S \subset [N] \text{ with } |S| = 3.$$

Recall that proving

$$|R(S)| \geq (1-\theta)d|S| \text{ for all } S \subset [N] \text{ with } |S| = s$$

does not imply the restricted expansion property

$$|R(S)| \geq (1-\theta)d|S| \text{ for all } S \subset [N] \text{ with } |S| \leq s$$

holds. Thus, at this point we can only conjecture that θ_3 satisfies

$$(1-\theta_3)3d = 3d - 3.$$

This is if and only if

$$\theta_3 = \frac{1}{d}.$$

However, since $\theta_1 = 0$ holds and it follows from theorem 25 that

$$\theta_2 = \frac{1}{2d}$$

holds, we have

$$0 < \frac{1}{2d} < \frac{1}{d}.$$

This implies that the conjectured value for θ_3 satisfies

$$\theta_1 < \theta_2 < \theta_3.$$

Therefore, we have shown

$$|R(S)| \geq \left(1 - \frac{1}{d}\right) d|S| \text{ for all } S \subset [N] \text{ with } |S| \leq 3.$$

This completes the proof. □

The conclusion of this theorem is not interesting. To see why this is, it follows from theorem 25 that θ_3 satisfies

$$\theta_3 \leq \frac{3-1}{2d} = \frac{1}{d}.$$

Thus the only new information we receive from the previous theorem is that equality holds. The more important result of this theorem is the proof technique of using the girth of the graph to compute the restricted expansion constant. In the next theorem, we consider this proof technique for $G^{[2]}$ and the results prove to be much more interesting.

Theorem 29. *Let A be the bipartite adjacency matrix of the graph $G^{[2]}_{p,d}$ with the parameters $2 \leq d \leq p$. Then the restricted expansion constant of order 3 is*

$$\theta_3 = \frac{2}{3d}$$

and the restricted expansion constant of order 4 is

$$\theta_4 = \frac{1}{d}.$$

Proof. Since the girth of $G^{[2]}_{p,d}$ is 8, this implies that the densest submatrix of 4 columns of A is the bipartite adjacency matrix of a cycle of length 8 such as

$$D := \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

We use this matrix to first find θ_3 and then find θ_4. Let S be a subset of 3 columns of A so that any 3 columns of the matrix D is a submatrix of A_s. Then there is exactly $4 + 3(d-2) = 3d - 2$ nonzero rows of A_S. This implies

$$|R(S)| \geq 3d - 2 \text{ for all } S \subset [N] \text{ with } |S| = 3.$$

Thus, we conjecture that θ_3 satisfies

$$(1 - \theta_3)3d = 3d - 2.$$

This is if and only if

$$\theta_3 = \frac{2}{3d}.$$

We now move on to conjecturing the value for θ_4. Let S be a subset of 4 columns of A so that any matrix similar to D is a submatrix of A_S. Then there is exactly $4 + 4(d-2) = 4d - 4$ nonzero rows of A_S. Thus,

$$|R(S)| \geq 4d - 4 \text{ for all } S \subset [N] \text{ with } |S| = 4.$$

We conjecture that θ_4 satisfies

$$(1 - \theta_4)4d = 4d - 4.$$

This is if and only if

$$\theta_4 = \frac{1}{d}.$$

We now prove that both of the conjectured values of the θ_s do indeed hold for all subsets smaller then s. Since

$$0 < \frac{1}{2d} < \frac{2}{3d} < \frac{1}{d}$$

holds, we have

$$\theta_1 < \theta_2 < \theta_3 < \theta_4.$$

Therefore, the restricted expansion constants hold for these values. □

Notice that for even very small sparsities, we see a difference between the bipartite adjacency matrices of $G^{[1]}$ and the bipartite adjacency matrices of $G^{[2]}$. What is significant about $G^{[2]}$ is that it appears that it could scales better then the approximation of θ_s provided by the coherence in theorem 25. Precisely, we have

$$\theta_3 = \frac{2}{3d} < \frac{1}{d} = \frac{3-1}{2d}$$

and

$$\theta_4 = \frac{1}{d} < \frac{3}{2d} = \frac{4-1}{2d}.$$

Counting Cycles Let G denote either the graph $G^{[1]}$ or the graph $G^{[2]}$ and let $g = \text{girth}(G)$. Consider any subset S of s left vertices of G. What is the maximum number of cycles of length g that can exist in the induced subgraph $G[S \cup R(S)]$ with out having any cycles of length less then g? Once we have this, we have an approximation on the bound of $|R(S)|$. The reason for this is because the number of right vertices is reduced as the more cycles exist. Thus, obtaining the maximum number of cycles gives us a lowerbound on $|R(S)|$.

Fortunately, this question looks like it is answerable. Tools from extremal graph theory may prove very useful in this situation.

10.4 Empirical Results of θ_s for Large s of $G^{[1]}$ and $G^{[2]}$

This section does not provide proof of the actual values of θ_s, nor an approximation θ. The purpose of this section is to show how the bipartite adjacency matrix A of $G^{[1]}$ and $G^{[2]}$ work in practice.

How the Procedure is Chosen Recall again that for any subset S of left vertices, the value of $|R(S)|$ can be obtained from A by counting the number of nonzero rows of A_S. More precisely, we have

$$|R(S)| = \| A_S \mathbf{1} \|_0 \text{ for all } S \subset [N].$$

This provides us with a relatively cheap way of computing the number of right vertices of a given subset of left vertices. Also recall that the bipartite adjacency matrix satisfies the robust null space property of order $\frac{s}{2}$ if $\theta_s < 1/6$.

We cannot check every possible subset of s columns of A because that would take $\binom{N}{s}$ computations of $\| A_S \mathbf{1} \|_0$. However, we can take a very small set T of s-subsets of $[N]$ at random and claim that if there exists an $S \in T$ with

$$\| A_S \mathbf{1} \|_0 \not> (1 - 1/6)\, sd,$$

then

$$1/6 \not> \theta_s > \ldots \theta_N.$$

Thus, A cannot satisfy the robust null space property of order greater than or equal to s. In this case, we must increment the left degree d and try again. Otherwise, if we show that

$$\| A_S \mathbf{1} \|_0 > (1 - 1/6)\, sd \text{ for every } S \in T,$$

then it is still possible that A does not satisfy the robust null space property. With this in mind, it is still worth pursuing this empirical approach. We can still find out if the bipartite adjacency matrix of $G^{[1]}$ or the bipartite adjacency matrix of $G^{[2]}$ is not a solution of the quadratic bottleneck problem. We do this by considering the following.

Suppose we take a random subset T of s-subsets of $[N]$ and we have

$$\| A_S \mathbf{1} \|_0 > (1 - 1/6)\, sd \text{ for every } S \in T.$$

With the concerns from the previous paragraph in mind, suppose that for the choice of parameters $2 \leq d \leq p$, the matrix does actually satisfy the robust null space property of order s. Then, we can use these parameters to calculate the constants C in the quadratic bottleneck

$$m \geq Cs^2$$

and the optimal bound

$$m \geq Cs \log (N/s)$$

By taking these values over an interval of sparsities s, then we can obtain empirical results on how the number of rows required scales as s increases. There are two main cases that can occur:

1. If the constant for the optimal bound becomes arbitrarily large as the sparsity s increases, then it is likely that the bipartite adjacency matrix of this graph does not satisfy the optimal bound and the quadratic bottleneck still holds.

2. If the constant for the quadratic bottleneck bound approaches 0 as s increases and the constant for the optimal bound does not increase to rapidly, then it is likely that the bipartite adjacency matrix of this graph is an actual solution of the quadratic bottleneck problem.

Note that if we are convinced that the second case holds for the bipartite adjacency matrix of either $G^{[1]}$ or $G^{[2]}$, then this does not constitute proof and we are not claiming this here. We would still need to carry out the analysis described in the previous sections of this chapter.

The Procedure We will use the discussion from the first part of this section as a guide and chose concrete parameters to test on. First we choose the length of the sparse vector to be recovered by selecting prime numbers for both of the constructions. Note that if we use the same prime p in $G^{[1]}_{p,d}$ and $G^{[2]}_{p,d}$, then the bipartite adjacency matrix of $G^{[2]}_{p,d}$ will be much larger then $G^{[1]}_{p,d}$. This approach does not make much sense so instead we take the vector length N to be approximately 50000. Then we choose the prime of the first construction to be $p_1 = 223$ and the prime of the second construction to be $p_2 = 37$ since

$$223^2 = 49729 \approx 50000 \approx 50653 = 37^3.$$

We start with the degree $d = 2$. For every $s = 10, 15, \ldots$, we take the set T to be 10000 random s-subsets of the columns of both of the bipartite adjacency matrices. Notice that since

$$\frac{10000}{\binom{N}{s}} \approx \frac{10000}{\binom{50000}{s}} \approx 0,$$

the number of subsets we have chosen is very small in comparison to all the possible subsets. If there exists an $S \in T$ with

$$\| A_S \mathbf{1} \|_0 \not> (1 - 1/6.2)\, sd,$$

then

$$1/6 \not> \theta_s > \ldots \theta_N$$

so the robust null space property of order s is not satisfied. In this case we increment d by one, then iterate this process again. Otherwise, if

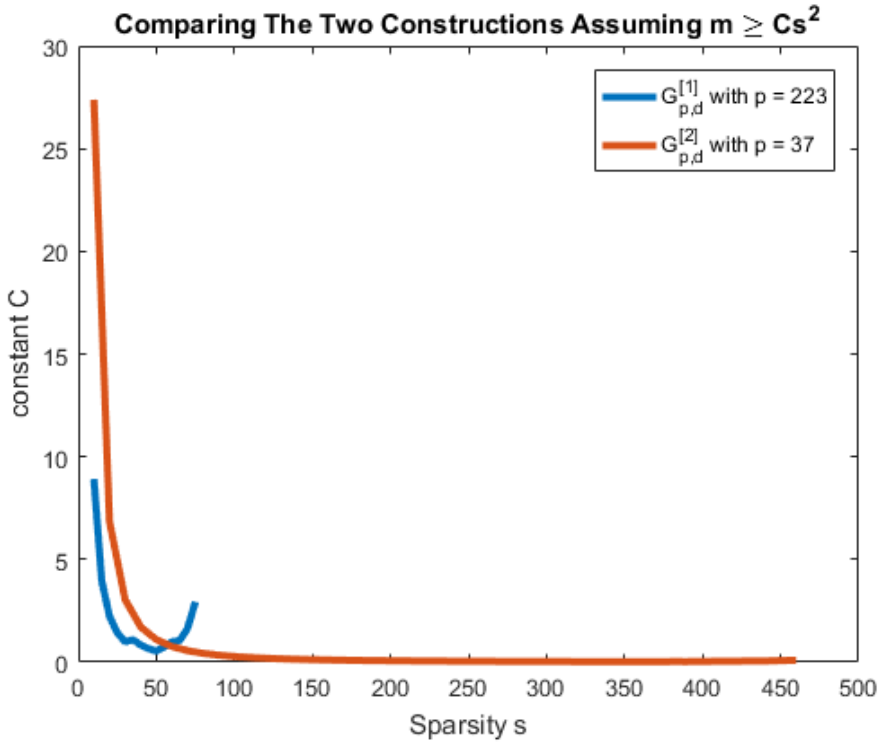
$$\| A_S \mathbf{1} \|_0 > (1 - 1/6.2)\, sd \text{ for every } S \in T,$$

then we store values s and d. Once we obtain $d = p$, we stop this process and then compute the constants in the quadratic bottleneck and the optimal bound. Notice that we chose the denominator to be 6.2 instead of 6. This provides us with some extra certainty that the robust null space property does indeed hold. The results are plotted in the figures on page 59.

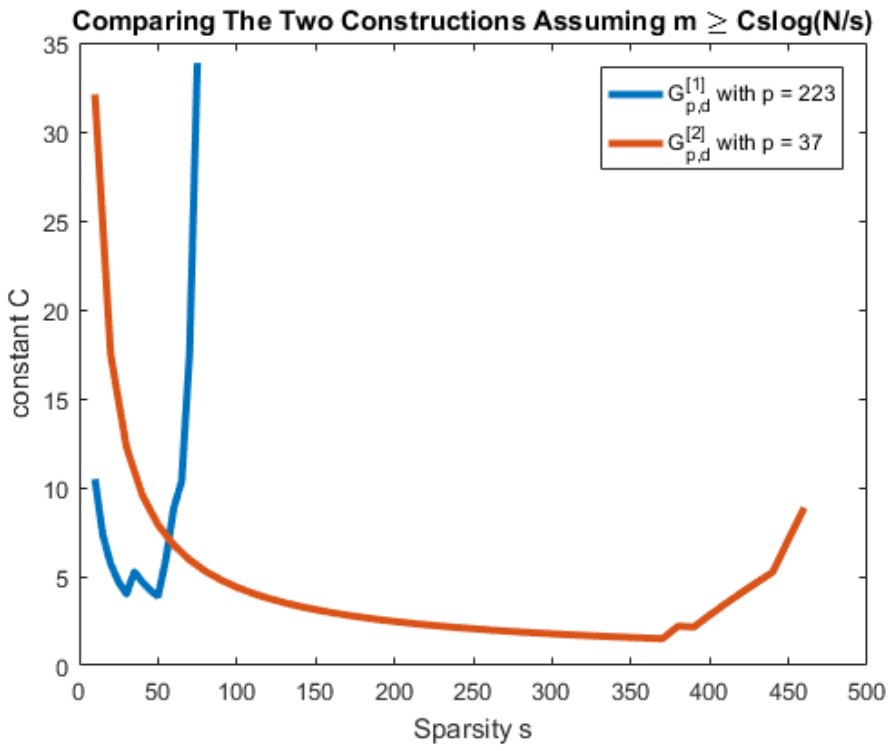
Results There are several observations we can make from the plots. First, the bipartite adjacency matrix of $G^{[1]}$ appears to not be a solution of the quadratic bottleneck problem. We make this claim because the constant C in the optimal bound blows up to infinity as s becomes larger. This is an unfortunate result.

However, the bipartite adjacency matrix of $G^{[2]}$ is shown to likely be a solution to the quadratic bottleneck problem. Clearly, the constant C for the quadratic bottleneck approaches 0 as s becomes larger. We also see that the constant C for the optimal bound does increase a bit as s increases, but not at a rate where we would conjecture that this bound does not hold.

We also observe that the bipartite adjacency matrix of $G^{[2]}$ is capable of recovering *much* denser vectors than the bipartite adjacency matrix of $G^{[1]}$. However, if the vector is very sparse, then the bipartite adjacency matrix of $G^{[1]}$ is a better choice since the constant C is smaller. Since we are mostly concerned with denser vectors, it appears the bipartite adjacency matrix of a graph with larger girth is the better choice in practice.



(a) Quadratic Bottleneck



(b) Optimal

11 Summary of Reference for Part Two

The definitions and theorems from chapter 6 were obtained from the graph theory textbook [6]. This text covers many of the topics within graph theory and is a good introduction to the theory. Similarly, the theorems and definitions from chapter 8 were primarily obtained from the reference source for combinatorial design theory [11]. This text is not a great introduction to combinatorial design theory but it does have a very large collection of relevant material for researchers who are familiar with the topics. Øyvind Ryan assisted me with the proof of theorem 15.

A survey of expander graphs, including some information on lossless expander graphs, can be found in [20]. More information can be found in chapter 21 of the textbook [1]. This textbook also contains interesting complexity topics such as derandomization, which was mentioned in the previous summary of references.

The note on the restricted expansion constant in section 7.2 was motivated by exercise 13.1 of [18]. Similarly, theorem 25 is an instantiation of exercise 13.2 of [18]. The fact that lossless expander graphs satisfy the robust null space property first appeared in [4], but was later improved apon in [18].

Applying generalized polygons to compressed sensing is taken into the consideration in [19]. In this paper, they compared their results to random lossless expander graphs. Unfortunately, most of the literature that exists for lossless expander graphs in the context of compressed sensing is for random constructions. However, explicit constructions have been published, for example, in [13].

The two explicit constructions were motivated by the paper [22], but originated in [23]. This is a paper written for the error correction coding community. In the paper, they do not introduce a left degree d and always take the full square $p^2 \times p^2$ matrix. The existence of an 8-cycle we used in the proof of 21 was also an example 8 cycle from [22]. The definition of the row and column blocks were also introduced from [22], however, we extended the definition to left vertex blocks and right vertex blocks for our application.

Part III

Compressed Sensing Recovery Algorithms Optimized for $G^{[1]}$ and $G^{[2]}$

12 Linear Algebra Algorithms Optimized for $G^{[1]}$ and $G^{[2]}$

In Part 3 we will, we will see that the bipartite adjacency matrix of $G^{[1]}$ and the bipartite adjacency matrix of $G^{[2]}$ have useful algorithmic properties. Specifically, we never need to create the matrix in memory to multiply with it and the multiplication itself requires few arithmetic operations. In this chapter, we optimize common linear algebra algorithms, such as matrix-vector multiplication and the conjugate gradient method for these matrices. The linear algebra algorithms presented in this chapter are the building blocks of the compressed sensing recovery algorithms discussed in the next section. Matlab code has been created in addition to this thesis which can be obtained for the reader if requested.

12.1 Matrix-Vector Multiplication Algorithms

Due to the simplicity of $G^{[1]}$, we will only explicitly state how we handle matrix-vector multiplication algorithms with its bipartite adjacency matrix. Since $G^{[2]}$ is defined by applying an additional linear equation to the linear equation in $G^{[1]}$, it should be clear how the matrix-vector multiplication algorithms are generalized to the bipartite adjacency matrix of $G^{[2]}$. We will list the pseudocode for the algorithms stated in this chapter in Appendix A.

Throughout this section, we will assume the following notation. Let A be the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ with parameters $2 \leq d \leq p$. Let $\mathbf{v} \in \mathbb{C}^m$ and $\mathbf{z} \in \mathbb{C}^N$ be arbitrary vectors. In this section, we define optimized matrix-vector multiplication algorithms to compute $A\mathbf{z}$, $A^T\mathbf{v}$, and $A^TA\mathbf{z}$. If $\mathbf{z}$ or $\mathbf{v}$ is sparse, then the algorithms we define here exploit the both the sparsity of the vector and the sparsity of the matrix. Obtaining these algorithms are of fundamental importance in compressed sensing because most of the recovery algorithms iterate matrix-vector multiplication.

Computing $\mathbf{v} = A\mathbf{z}$.

Let $j \in [N]$ and $i \in [m]$ be arbitrary indices. Let (x, y) of be the left vertex of $G^{[1]}_{p,d}$ corresponding with the index j and let (a, b) be the right vertex of $G^{[1]}_{p,d}$ corresponding to the index i. Observe that it follows from the definition of

$G_{p,d}^{[1]}$ that the matrix A is defined by

$$a_{i,j} = \begin{cases} 1, & \text{if } y = ax + b \pmod{p} \\ 0, & \text{if } y \neq ax + b \pmod{p}. \end{cases}$$

This implies the vector $\mathbf{v} = A\mathbf{z}$ is defined by adding together specific entries of $\mathbf{z}$. If $\mathbf{z}$ is an s-sparse vector with the support set S, then we compute the vector $\mathbf{v}$ by implementing the following.

Initialize: The vector $\mathbf{v} = \mathbf{0}$.

Iterate: For each index $j \in S$,

1. **Compute** the corresponding left vertex (x, y) to j.

   Iterate: For each $(a, b) \in R((x, y))$

   a) **Compute** the corresponding index (i) to (a, b)

   b) **Update** the vector $\mathbf{v}(i) = \mathbf{v}(i) + \mathbf{z}(j)$

Since $\mathbf{z}$ is s-sparse, we have $|S| = s$. Since $G_{p,d}^{[1]}$ is a d-left regular bipartite graph, we have $|R((x, y))| = d$, for all left vertices (x, y). This implies that the number of arithmetic operations required to compute $\mathbf{v} = A\mathbf{z}$ is $\mathcal{O}(sd)$. Notice that the input vector $\mathbf{z}$ does not need to be sparse. If the vector $\mathbf{z}$ is dense, then this algorithm requires $\mathcal{O}(Nd)$ arithmetic operations which can be pretty slow and be a problem if applied iteratively. However, for most applications, the number of arithmetic operations on the order of sd is a useful property to have. The pseudocode for this algorithm can be found the appendix as algorithm 1.

Computing $\mathbf{z} = A^T\mathbf{v}$.

To compute the vector $\mathbf{z} = A^T\mathbf{v}$, we essentially apply the same algorithm. The main difference between this algorithm and the previous algorithm is that the complexity is slightly different. Since $G_{p,d}^{[1]}$ is a p-right regular bipartite graph, we have $|N((a, b))| = p$, for all right vertices (a, b). If $\mathbf{v}$ is s-sparse then the number of arithmetic operations required to compute $\mathbf{z} = A^T\mathbf{v}$ is $\mathcal{O}(sp)$. The pseudocode for this algorithm is found the appendix as algorithm 2.

Computing $\mathbf{w} = A^TA\mathbf{z}$

The matrix-vector algorithm for $\mathbf{w} = A^TA\mathbf{z}$ is not necessary to define, since it can be obtained by composing the previous two algorithms. Nevertheless, observe that

$$(A^TA)_{j_1,j_2} = \langle \mathbf{a}_{j_1}, \mathbf{a}_{j_2} \rangle \text{ for any } j_1, j_2 \in [N].$$

For any indices j_1 and j_2 with corresponding left vertices (x_1, y_1) and (x_2, y_2), respectively, we have

$$\langle \mathbf{a}_{j_1}, \mathbf{a}_{j_2} \rangle = \begin{cases} d, & \text{if } j_1 = j_2 \\ 1, & \text{if there exists right vertex } (a, b) \text{ such that} \\ & (a, b) \text{ is adjacent to both } (x_2, y_2) \text{ and } (x_1, y_1) \\ 0, & \text{otherwise.} \end{cases}$$

It follows from the adjacency condition of $G^{[1]}$ that a right vertex (a, b) is adjacent to both (x_1, y_1) and (x_2, y_2) if

$$y_1 = ax_1 + b \pmod{p} \quad \text{and} \quad y_2 = ax_2 + b \pmod{p}$$

hold. Putting these two equations together implies

$$\langle \mathbf{a}_{j_1}, \mathbf{a}_{j_2} \rangle = 1 \quad \text{if and only if} \quad ax_1 - y_1 = ax_2 - y_2 \pmod{p}$$

The algorithm to compute the vector $\mathbf{w} = A^T A\mathbf{z}$ follows by iterating through all values of $j_1, j_2 \in \text{supp}(\mathbf{z})$. The number of arithmetic operations to run $\mathbf{w} = A^T A\mathbf{z}$ is $\mathcal{O}(s^2 d)$, where s is the sparsity of $\mathbf{z}$. Notice that the number of arithmetic operations required is marginally better than if we were to compose the previous two algorithms rather than take this approach. Algorithm 3 in the appendix provides pseudocode for this algorithm.

Bijective Functions Between the Indices of the Matrix and the Vertices of $G^{[1]}$

To compute the previous algorithms, we must define a bijective relationship between the indices of the columns of A and the left vertices of $G^{[1]}_{p,d}$, as well as a bijective function between the row indices of A and the right vertices of $G^{[1]}_{p,d}$. It is important to minimize the number of arithmetic operations required to compute both directions of these bijective relationship because they will be computed many times over the course of a matrix-vector multiplication. Since the algorithms presented in this section are applied many times in compressed sensing recovery algorithms, it is even more important to minimize the number of required arithmetic operations.

Since the vertices of $G^{[1]}_{p,d}$ are defined by ordered pairs, we correspond each index $j \in [N]$ with the left vertex

$$(x, y) = \left(\left\lfloor \frac{j-1}{p} \right\rfloor, \; j - 1 \right) \pmod{p}.$$

To obtain the index that corresponds with the left vertex (x, y), we compute

$$j = xp + y + 1.$$

Similarly, we correspond each index $i \in [m]$ with the right vertex

$$(a, b) = \left(\left\lfloor \frac{i-1}{p} \right\rfloor, \; i - 1 \right) \pmod{p}.$$

To obtain the index that corresponds with the right vertex (a, b), we compute

$$i = ap + b + 1.$$

12.2 Least Squares Minimization Problem

For the next two sections we will assume the following. Let $\mathbf{x}$ be an arbitrary s'-sparse vector. Let A be the bipartite adjacency matrix of the graph $G^{[1]}_{p,d}$ or the

bipartite adjacency matrix of the graph $G_{p,d}^{[2]}$ with the parameters $2 \leq d \leq p$. Let $\mathbf{y} = A\mathbf{x}$ be the measurement vector of $\mathbf{x}$. Many of the recovery algorithms we will be discussing in the next chapter will require us to find a solution to

$$\mathbf{x}^\sharp = \underset{\mathbf{z}\in\mathbb{C}^N}{\arg\min}\{\| \mathbf{y} - A\mathbf{z} \|_2 \mid \text{supp}(\mathbf{z}) \subset S\}, \tag{12.1}$$

for some $S \subset [N]$ with $|S| = s \leq s'$. Equation (12.1) is the *least squares minimization problem* over the support set S. If we have $\theta_{2s} < 1/6$, then the robust null space property of order s is satisfied. Then theorem 2 implies every set of $2s$-columns of A are linearly independent. It follows from theorems 8.4 and 8.5 of [24] that the least squares solution $\mathbf{x}^\sharp$ in (12.1) is unique and we can obtain the least squares solution from the *normal equations*,

$$A_S^T A_S \mathbf{z}_S = A_S^T \mathbf{y}_S. \tag{12.2}$$

One can solve the normal equations with a QR-decomposition of A_S or solving

$$\mathbf{z}_S = (A_S^T A_S)^{-1} A_S \mathbf{y}_S.$$

However, both of these approaches are not desirable to use in this situation for a variety of reasons. The latter option requires us to compute the inverse of $A_S^T A_S$, which is computationally expensive. The QR-decomposition requires us to store the orthogonal matrix Q and the upper triangular matrix R. Similarly, the latter option requires us to store the matrix $(A_S^T A_S)^{-1}$. This implies that in both cases we must now be concerned with memory constraints. Neither of these approaches exploit the matrix-vector algorithms we defined in the previous section. The next section describes an alternative method that avoids these problems.

12.3 Conjugate Gradient Method

Since we have matrix-vector multiplication algorithms for A and A^T, we can solve the normal equations in (12.2) efficiently. Rather then use a direct method, such as one of the two presented above or Gaussian elimination, we will be applying an iterative method called the *conjugate gradient method*. Much can be said about the conjugate gradient method, but we will only mention here what is necessary for the rest of the thesis.

The conjugate gradient method iteratively solves sparse positive definite systems of linear equations. Since every set of $2s$ columns of A are linearly independent, it follows from lemma 3.8 of [24] the matrix $A_S^T A_S$ is positive definite. This implies we can apply the conjugate gradient method to solve

$$B_S \mathbf{z}_S = \mathbf{b}_S,$$

where $B_S := A_S^T A_S$ and $\mathbf{b}_S := A_S^T \mathbf{y}_S$. To define the conjugate gradient method, we need the following definitions.

Definition 36 (A-norm and A-inner product)**.** Let $A \in \mathbb{R}^{n\times n}$ be any positive definite matrix. Then, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the *$A$-norm* of $\mathbf{x}$ is defined by

$$\| \mathbf{x} \|_A := \sqrt{\mathbf{x}^T A \mathbf{x}}$$

and the *A-inner product* of $\mathbf{x}$ and $\mathbf{y}$ is defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle_A := \mathbf{x}^T A \mathbf{y}.$$

Computation

To solve for the solution $\mathbf{x}^\sharp$ of $B_S\mathbf{z}_S = \mathbf{b}_S$, we implement the following.

Initialize :

1. **Choose** a small tolerance $\epsilon > 0$.
2. **Choose** a best initial guess $\mathbf{x}^{(0)}$ of $\mathbf{x}$.
3. **Compute** the residual

$$\mathbf{r}^{(0)} = B_S\mathbf{x}_S^{(0)} - \mathbf{b}_S.$$

Iterate: For each step $n = 1, 2 \ldots$

1. **Choose** a *step direction* $\mathbf{p}^{(n)}$ by

$$\mathbf{p}^{(n)} := \mathbf{r}^{(n)} - \sum_{i=0}^{n-1} \frac{\langle\, \mathbf{r}^{(n)}, \mathbf{p}^{(i)} \,\rangle_{B_S}}{\langle\, \mathbf{p}^{(i)}, \mathbf{p}^{(i)} \,\rangle_{B_S}} \mathbf{p}^{(i)}.$$

2. **Choose** a *step length* $\alpha^{(n)}$ by

$$\alpha^{(n)} := \frac{\left\| \mathbf{r}^{(n)} \right\|_{B_S}^2}{\left\| \mathbf{p}^{(n)} \right\|_{B_S}^2}.$$

3. **Update** the n-th approximation $\mathbf{x}^{(n+1)}$

$$\mathbf{x}_S^{(n+1)} := \mathbf{x}_S^{(n)} + \alpha^{(n)}\mathbf{p}^{(n)}.$$

4. **Update** the residual

$$\mathbf{r}^{(n+1)} := \mathbf{r}^{(n)} - \alpha^{(n)} B_S \mathbf{p}^{(n)}.$$

Halt: If $\left\| \mathbf{r}^{(n+1)} \right\| < \epsilon$

The slowest part of the conjugate gradient method is iteratively applying algorithm 3 with the matrix B_S. Notice that applying algorithm 3 to $\mathbf{w} = B_S\mathbf{z}$ with $T = \mathrm{supp}(\mathbf{z})$ may be significantly less expensive than the original $\mathcal{O}(s^2 d)$. This is because we have to do arithmetic on $t = |T \cap S|$ indices. This shows that $\mathbf{w} = B_S\mathbf{z}$ requires $\mathcal{O}(t^2 d)$ arithmetic operations.

If we apply the algorithms introduced in the first section of this chapter, then at no point in this algorithm do we need to create or store a matrix. When we call the conjugate gradient method in this context, we will denote this function by

$$\mathbf{x}^\sharp = \mathrm{CG}(\mathbf{x}^{(0)}, \mathbf{b}, p, d, S).$$

Convergence

The following theorem shows that the conjugate gradient demonstrates the rate of convergence of the conjugate gradient method.

Theorem 30 ([24], Theorem 12.16)**.** *Let A be any positive definite matrix. Then the n-th step of the conjugate gradient method satisfies*

$$\frac{\left\| \boldsymbol{x} - \boldsymbol{x}^{(n)} \right\|_A}{\left\| \boldsymbol{x} - \boldsymbol{x}^{(0)} \right\|_A} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n,$$

where $\kappa = \lambda_{max}/\lambda_{min}$ is the spectral condition number of A.

Since the matrix B_S is a positive definite matrix, the inequality in theorem 30 holds for every step of the conjugate gradient method. Theorem 30 demonstrates a couple of properties of the algorithm. First, since

$$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} < 1, \tag{12.3}$$

we have

$$\lim_{n \to \infty} 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n = 0. \tag{12.4}$$

Therefore, the conjugate gradient method always converges. Observe that if λ_{min} relatively close in value to λ_{max}, then the inequality (12.3) is close to 1. Thus, equation (12.4) approaches 0 at a slower rate and it follows that the conjugate gradient method takes more iterations to complete. Unfortunately, due to the nature of the construction of B_S, it is not clear how to find a good approximation of κ. This implies we can not say much about the exact rate of convergence.

Next, suppose we have a bad initial estimate $\mathbf{x}^{(0)}$ for $\mathbf{x}$. Then $\left\| \mathbf{x} - \mathbf{x}^{(0)} \right\|_{B_S}$ is large and it follows from the inequality in 30 that the error of the n-th estimate $\mathbf{x}^{(n)}$ of $\mathbf{x}$ may be large. This problem we will actually be avoidable and this will be made clear when we introduce compressed sensing recovery algorithms.

13 Compressed Sensing Recovery Algorithms with $G^{[1]}$ and $G^{[2]}$

In this next chapter, we take a step back from the quadratic bottleneck problem and we discuss various compressed sensing recovery algorithms when the measurement matrix is the bipartite adjacency matrix of $G^{[1]}$ or $G^{[2]}$. In many instances, the way we present the algorithms are easily generalized to general measurement matrices, but this is not always the case.

13.1 Lossless Expander Graphs Relationship to the Scaled Restricted Isometry Property in ℓ_1

Before we discuss the common compressed sensing recovery algorithms, we define a variation to the restricted isometry property that may be useful for proving theoretical results about convergence. In Chapter 4 we introduced the restricted isometry property for matrices with ℓ_2-normalized columns. This does not translate well to the bipartite adjacency matrix of a lossless expander graph. If we were to just relax the definition to allow for binary matrices, then the paper [10] shows that the number of rows m cannot obtain the optimal bound

$$m \geq Cs \log (N/s).$$

This shows that the restricted isometry property is an insufficient measure for us. Since many proofs in compressed sensing depend on the restricted isometry property, this is an unfortunate property for binary matrices to have. The variation of the restricted isometry property that we will use is called the *scaled restricted isometry property* in ℓ_1. The main difference between these two properties is that instead of ℓ_2-norm, we use the ℓ_1-norm and instead of the restricted isometry constant δ_s, we use the restricted expansion constant θ_s. It should be noted that it seems the scaled restricted isometry property in ℓ_1 is far less common in the literature.

Before continuing we first need to introduce some notation and a lemma whose proof can be found in lemma 13.4 of [18].

Lemma 3 ([18], Lemma 13.4)**.** *Let G be an (s, d, θ)-lossless expander graph. Let S be any set of left vertices of G with $|S| \leq s$. Denote the set of right vertices of S with i left vertices in S by $R_i(S)$. Then*

$$|R_1(S)| \geq (1 - 2\theta_s)d|S|.$$

With this, we are ready to prove the main result of this section.

Theorem 31 ([18], Ex 13.5)**.** *Let $A \in \{0,1\}^{m\times N}$ be the bipartite adjacency matrix of an $(s,d,\theta)-$lossless expander graph. Then A satisfies the scaled restricted isometry property in ℓ_1*

$$d(1-2\theta)\,\|\,\boldsymbol{x}\,\|_1 \leq \|\,A\boldsymbol{x}\,\|_1 \leq d\,\|\,\boldsymbol{x}\,\|_1$$

for all s-sparse $\boldsymbol{x}\in\mathbb{C}^N$.

Proof. Let $\mathbf{x}\in\mathbb{C}^N$ be an arbitrary s-sparse vector. We start with the upper bound. The operator 1-norm is subordinate to the vector 1-norm, so we have

$$\|\,A\mathbf{x}\,\|_1 \leq \|\,A\,\|_1\,\|\,\mathbf{x}\,\|_1\,.$$

Since the operator 1-norm of B is equal to the maximum column sum of A and A has d ones in each column, we have $\|\,A\,\|_1 = d$. This gives us,

$$\|\,A\mathbf{x}\,\|_1 \leq d\,\|\,\mathbf{x}\,\|_1$$

and this proves the upper bound.

We now prove the lower bound. Let S be an arbitrary subset of left vertices with $|S|\leq s$. Let $\neg R(S)$ denote the set of right vertices with no adjacent left vertex in S. Then we can write $\|\,A\mathbf{x}\,\|_1$ as the following sum,

$$\|\,A\mathbf{x}\,\|_1 = \left\|\,(A\mathbf{x})_{\neg R(S)}\,\right\|_1 + \left\|\,(A\mathbf{z})_{R_1(S)}\,\right\|_1 + \left\|\,(A\mathbf{x})_{R_{\geq 2}(S)}\,\right\|_1. \tag{13.1}$$

In general, if s is large enough then there exists a set of left vertices S such that $R(S)$ is equal to all of the right vertices of G. In this case $\neg R(S)$ is equal to the empty set, so that $\left\|\,(A\mathbf{x})_{\neg R(S)}\,\right\|_1 = 0$. Thus, this is the lower bound and for an arbitrary S we have

$$\left\|\,(A\mathbf{x})_{\neg R(S)}\,\right\|_1 \geq 0.$$

Similarly, if s is small enough, then there exists a set of left vertices S such that all of the right vertices in $R(S)$ have a unique left vertex in S. In this case $R_{\geq 2}(S)$ is equal to the empty set and $\left\|\,(A\mathbf{x})_{R_{\geq 2}(S)}\,\right\|_1 = 0$. Thus, this is the lower bound and for an arbitrary S we have

$$\left\|\,(A\mathbf{x})_{R_{\geq 2}(S)}\,\right\|_1 \geq 0.$$

Recalling from lemma 3 that $|R_1(S)| \geq (1-2\theta_s)ds$ and $\theta_s \leq \theta$, we have

$$\left\|\,(A\mathbf{x})_{R_1(S)}\,\right\|_1 \geq (1-2\theta)d\,\|\,\mathbf{x}_S\,\|_1\,.$$

By applying all of these inequalities to the equation (13.1), we obtain

$$\begin{aligned}\|\,A\mathbf{x}\,\|_1 &= \left\|\,(A\mathbf{x})_{\neg R(S)}\,\right\|_1 + \left\|\,(A\mathbf{z})_{R_1(S)}\,\right\|_1 + \left\|\,(A\mathbf{x})_{R_{\geq 2}(S)}\,\right\|_1 \\ &\geq d(1-2\theta)\,\|\,\mathbf{x}_S\,\|_1 \\ &= d(1-2\theta)\,\|\,\mathbf{x}\,\|_1\,.\end{aligned}$$

This proves the lowerbound and it follows that the bipartite adjacency matrix of a lossless expander graph satisfies the scaled restricted isometry property in ℓ_1. □

The next corollary further demonstrates why we require θ_{2s} to be small.

Corollary 2. *Let $A \in \{0,1\}^{m \times N}$ be the bipartite adjacency matrix of an (s,d,θ)-lossless expander with $\theta_{2s} < 1/2$. Then every s-sparse vector has a unique measurement vector.*

Proof. Let $\mathbf{x}, \mathbf{x}' \in \{0,1\}^N$ be distinct s-sparse vectors and let their measurement vectors be $\mathbf{y} = A\mathbf{x}$ and $\mathbf{y}' = A\mathbf{x}'$, respectively. Then it follows from theorem 31 and the assumption that $\theta_{2s} < d/2$ that the following inequality holds

$$0 < d(1 - 2\theta_{2s}) \| \mathbf{x} - \mathbf{x}' \|_1 \leq \| A(\mathbf{x} - \mathbf{x}') \|_1 .$$

This implies that we have

$$\mathbf{y} - \mathbf{y}' = A\mathbf{x} - A\mathbf{x}' \neq \mathbf{0},$$

which is if and only if $\mathbf{y} \neq \mathbf{y}'$. Therefore, every s-sparse vector has a unique measurement vector. □

13.2 Unique Neighborhood Algorithm

Combinatorial Recovery Algorithms This first class of recovery algorithms require a specific type of measurement matrix. Specifically, combinatorial recovery algorithms require a measurement matrix A with entries 0 and 1. Thus, matrix-vector multiplication implies that every entry of the measurement vector $\mathbf{y} = A\mathbf{x}$ of a sparse vector $\mathbf{x}$ is a sum of the entries in the support of $\mathbf{x}$. If the measurement matrix is also sparse, then many of the values of the entries of $\mathbf{y}$ are not just the sum of entries in the support of $\mathbf{x}$, but are the actual values of the entries in the support of $\mathbf{x}$. When we use the bipartite adjacency matrix of a left regular graph, there tends to be repeated values of $\mathbf{x}$ in $\mathbf{y}$. This is because there are multiple ones in each column of the measurement matrix. The combinatorial approach uses these repeated values in $\mathbf{y}$ to find the locations of that value in $\mathbf{x}$.

The fact that combinatorial recovery algorithms require finding the same values in a measurement vector many times is a large setback in applications of compressed sensing. This is because we cannot properly deal with measurement error. Because of this, we will only present one variation of this class of algorithm. Though, it should be noted though that there are many algorithms that exist in this class.

Unique Neighborhood algorithm

Let A be the bipartite adjacency of an (s,d,θ)-lossless expander graph G. Let $\mathbf{x}$ be an arbitrary s-sparse vector with measurement vector $\mathbf{y} = A\mathbf{x}$. Recall from lemma 3 that, for any subset S of left vertices, we have

$$|R_1(S)| \geq (1 - 2\theta)d|S|.$$

This inequality implies that for any left vertex v, we have

$$|R_1(v)| \geq (1 - 2\theta)d.$$

When put into the terms of the measurement vector $\mathbf{y}$, this implies that for every $j \in \text{supp}(\mathbf{x})$, there exists at least $(1-2\theta)d$ entries in $\mathbf{y}$ with the value $\mathbf{x}(j)$. Thus, if we find $(1-2\theta)d$ entries in $\mathbf{y}$ with the same value, then we can obtain the location of that value in $\mathbf{x}$. Using this line of reasoning, we can recover the s-sparse vector $\mathbf{x}$ by implementing the following.

Computation

Initialize :

1. **Choose** a small tolerance $\epsilon > 0$.
2. **Choose** a best initial guess $\mathbf{x}^{(0)}$ of $\mathbf{x}$.
3. **Compute** the residual

$$\mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{y}.$$

Iterate: For each step $n = 0, 1 \ldots$

1. **Find** a set of indices R_{val} of $(1-\theta)d$ entries of $\mathbf{r}^{(n)}$ with the same value val.
2. **Find** the unique index column index $j \in [N]$ of A that satisfies the property that for all $i \in R_{val}$, we have $a_{i,j} = 1$.
3. **Update** the n-th approximation $\mathbf{x}^{(n+1)}$ of $\mathbf{x}$ by

$$\mathbf{x}^{(n+1)}(j) = \mathbf{x}^{(n)}(j) + val$$

4. **Update** the residual

$$\mathbf{r}^{(n+1)} = \mathbf{y} - A\mathbf{x}^{(n+1)}.$$

Halt: If $\left\| \mathbf{r}^{(n+1)} \right\| < \epsilon$

This algorithm is optimal for random binary measurement matrices, since no knowledge of the measurement matrix is known beforehand. The slowest step of each iteration is finding the column of the matrix that matches the set of indices of R_{val} since we have to check $\mathcal{O}(N)$ set intersections. However, if A is the bipartite adjacency matrix of the graph $G^{[1]}$ or the graph $G^{[2]}$, then we can optimize this procedure. We will only consider how this can be optimized for $G^{[1]}$ and the procedure for $G^{[2]}$ can be easily generalized.

Let $i_1, i_2 \in R_{val}$ be distinct indices with corresponding right vertices (a_1, b_1) and (a_2, b_2), respectively. Then, there exists at most one left vertex that is adjacent to both of these right vertices. Let (x, y) be such a right vertex. Then, we have

$$y = a_1 x + b_1 \quad \text{and} \quad y = a_2 x + b_2 \pmod{p}$$

which is if and only if

$$x = (a_1 - a_2)^{-1}(b_2 - b_1) \pmod{p}$$

and

$$y = a_1(a_1 - a_2)^{-1}(b_2 - b_1) + b_1 \pmod{p}.$$

Finally, we find the column index $j \in [N]$ corresponding to (x, y) and update the approximation

$$\mathbf{x}^{(n+1)}(j) = \mathbf{x}^{(n)}(j) + val.$$

This procedure requires a constant number of arithmetic operations rather then the linear in N required in the non-optimal version.

13.3 Orthogonal Matching Pursuit and Compressive Sampling Matching Pursuit

Orthogonal Matching Pursuit

The orthogonal matching pursuit recovery algorithm is the first of the two greedy algorithms presented. The reason why orthogonal matching pursuit is considered a greedy method is because optimization occurs on each iteration and it is on a local level rather than in a basis pursuit algorithm which does global optimization. More precisely, orthogonal matching pursuit adds another index to the support set on each iteration in a way that obtains the maximum distance between $\mathbf{x}$ and its n-th approximation $\mathbf{x}^{(n)}$. It then updates the approximation $\mathbf{x}^{(n+1)}$ so that the this maximized distance is decreased.

Computation Let $2 \leq d \leq p$ be the parameters of the bipartite adjacency matrix A of the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$. Let $\mathbf{x}$ be an arbitrary s-sparse vector. Then to recover the vector $\mathbf{x}$ from its measurement vector $\mathbf{y} = A\mathbf{x}$ with *orthogonal matching pursuit*, we implement the following.

Initialize :

1. **Choose** a small tolerance $\epsilon > 0$.
2. **Choose** a best initial guess $\mathbf{x}^{(0)}$ of $\mathbf{x}$.
3. **Compute** the residual
$$\mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{y}.$$

Iterate: For each step $n = 1, 2 \ldots$

1. **Choose** the index
$$j^{(n+1)} = \arg\max_{j \in [N]} \{ |(A^T(\mathbf{y} - A\mathbf{x}^{(n)}))_j| \}.$$
2. **Update** the support set
$$S^{(n+1)} = S^{(n)} \cup \{j^{(n+1)}\}.$$
3. **Update** the n-th approximation $\mathbf{x}^{(n+1)}$ of $\mathbf{x}$ by solving the least squares minimization problem over the support set $S^{(n+1)}$ defined by
$$\mathbf{x}^{(n+1)} = \arg\min_{\mathbf{z} \in \mathbb{C}^N} \{ \| \mathbf{y} - A\mathbf{z} \|_2 \mid \mathrm{supp}(\mathbf{z}) \subset S^{(n+1)} \}.$$
This is done by solving the conjugate gradient method on the normal equations
$$\mathbf{x}^{(n+1)} = \mathrm{CG}(\mathbf{x}^{(n)}, \mathbf{b}, p, d, S^{(n+1)}),$$
where $\mathbf{b} = A^T\mathbf{y}$.

4. **Update** the residual

$$\mathbf{r}^{(n+1)} = \mathbf{y} - A\mathbf{x}^{(n+1)}.$$

Halt: If $\left\| \mathbf{r}^{(n+1)} \right\| < \epsilon$

The slowest part of the orthogonal matching pursuit algorithm is the computation of the conjugate gradient method. Also notice that since we only add one index on each iteration, orthogonal matching pursuit requires at least s iterations to complete. Thus, we must apply the conjugate gradient method at least s times, which is less then desirable for large s. The upside of this is that for the smaller values of s, the conjugate gradient method is applied on a very small support set, so this requires little arithmetic operations.

Analysis We now go back to the original discussion about why orthogonal matching pursuit is considered a greedy algorithm. First, observe that requiring the restricted expansion constant θ_s to be small implies

$$A_S^T A_S \approx dI \text{ for all } S \subset [N] \text{ with } |S| \leq s, 0$$

where I is the identity matrix of order s. Consider the vector

$$A^T(\mathbf{y} - A\mathbf{x}^{(n)}) = A^T A(\mathbf{x} - \mathbf{x}^n) \approx dI(\mathbf{x} - \mathbf{x}^n) = d(\mathbf{x} - \mathbf{x}^n),$$

which is the vector the index $j^{(n+1)}$ is maximized over. Thus, our choice of $j^{(n+1)}$ is chosen to maximize the distance between $\mathbf{x}^{(n)}$ and $\mathbf{x}$. We then solve the least squares problem over this support set to update the approximation in a way that reduces this maximum distance.

As a final note, we recall from theorem 30 that the error bound of the n-th iteration of the conjugate gradient method may be large if we have a bad initial approximation vector. As seen from orthogonal matching pursuit, we use the previous iterations approximation on each iteration of orthogonal matching pursuit as an initial vector in the conjugate gradient method. Since each iteration does not change much from step to step, this shows that this choice of initialization should be a good choice.

Compressive Sampling Matching Pursuit

Before we describe the compressive sampling matching pursuit algorithm, we need to introduce the following nonlinear operators.

Definition 37 (Hard Thresholding Operator)**.** Let $\mathbf{z} \in \mathbb{C}^N$ be a vector. Then define the set $L_s(\mathbf{z})$ to be the index set of the s largest absolute entries of $\mathbf{z}$ and the *hard thresholding operator* $H_s(\mathbf{z})$ to be the s largest absolute entries of $\mathbf{z}$ with the rest of the entries of $\mathbf{z}$ being zeroed out.

The fact that these two operators are nonlinear can make the analysis of algorithms less straight forward.

Compressive sampling matching pursuit again does local optimization. The main difference from orthogonal matching pursuit is that instead of choosing a

single index on each iteration, we choose an index set of size $2s$ of the maximum absolute entries of the vector

$$A^T(\mathbf{y} - A\mathbf{x}^{(n)}).$$

This slows down each iteration significantly, but the benefit to this algorithm is that it could take far less iterations to complete in comparison to orthogonal matching pursuit if the sparsity s is large.

Computation Let $2 \leq d \leq p$ be the parameters of the bipartite adjacency matrix A of the graph $G^{[1]}_{p,d}$ or the graph $G^{[2]}_{p,d}$. Let $\mathbf{x}$ be an arbitrary s-sparse vector. Then to recover the vector $\mathbf{x}$ from its measurement vector $\mathbf{y} = A\mathbf{x}$ with *compressive sampling matching pursuit*, we implement the following.

Initialize :

1. **Choose** a small tolerance $\epsilon > 0$.
2. **Choose** a best initial guess $\mathbf{x}^{(0)}$ of $\mathbf{x}$.
3. **Compute** the residual

$$\mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{y}.$$

Iterate: For each step $n = 1, 2 \ldots$

1. **Choose** the index set of size greater then $2s$ by

$$S^{(n+1)} = \text{supp}(\mathbf{x}^{(n)}) \cup L_{2s}(A^T(\mathbf{y} - A\mathbf{x}^{(n)})).$$

2. **Compute** the least squares solution $\mathbf{u}^{(n+1)}$ from the least squares minimization problem over the support set $S^{(n+1)}$ defined by

$$\mathbf{u}^{(n+1)} = \underset{\mathbf{z} \in \mathbb{C}^N}{\arg\min}\{\| \mathbf{y} - A\mathbf{z} \|_2 \mid \text{supp}(\mathbf{z}) \subset S^{(n+1)}\}.$$

   This is done by solving the conjugate gradient method on the normal equations

$$\mathbf{u}^{(n+1)} = \text{CG}(\mathbf{u}^{(n)}, \mathbf{b}, p, d, S^{(n+1)}),$$

   where $\mathbf{b} = A^T\mathbf{y}$.

3. **Compute** the s largest absolute entries of $\mathbf{u}^{n+1}$ by

$$\mathbf{x}^{n+1} = H_s(\mathbf{u}^{n+1}).$$

4. **Update** the residual

$$\mathbf{r}^{(n+1)} = \mathbf{y} - A\mathbf{x}^{(n+1)}.$$

Halt: If $\left\| \mathbf{r}^{(n+1)} \right\| < \epsilon$

13.4 Iterative Hard Thresholding and Hard Thresholding Pursuit

In this section we define two recovery algorithms that are based on iterative methods and the hard thresholding operator of H_s.

Derivation of the Algorithms Let A be the bipartite adjacency matrix of the graph $G^{[1]}$ or the graph $G^{[2]}$. Let $\mathbf{x}$ be an arbitrary s-sparse vector with the measurement vector $\mathbf{y} = A\mathbf{x}$. Recall from the least squares minimization section that we want to solve the normal equations

$$A^T A\mathbf{z} = A\mathbf{y},$$

If we take the splitting matrix to be the identity matrix I, then we have the following system of linear equations,

$$\begin{aligned}\mathbf{x} &= (I - A^T A)\mathbf{x} + A^T\mathbf{y} \\ &= \mathbf{x} + A^T(\mathbf{y} - A\mathbf{x}).\end{aligned}$$

This implies we can use the fixed point iteration method

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + A^T(\mathbf{y} - A\mathbf{x}^{(n)}).$$

By applying the hard thresholding operator H_s to the right hand side of this equation gives us

$$\mathbf{x}^{(n+1)} = H_s(\mathbf{x}^{(n)} + A^T(\mathbf{y} - A\mathbf{x}^{(n)})),$$

which ensures that $\mathbf{x}^{n+1}$ is s-sparse. By iterating this for all $n = 1, 2, \ldots$, we obtain the *iterative hard thresholding* algorithm. If instead we were to take the index set of the s largest elements on the right hand side of the equation, then we obtain the index set

$$S^{(n+1)} = L_s(\mathbf{x}^{(n)} + A^T(\mathbf{y} - A\mathbf{x}^{(n)})).$$

Then the least squares minimization problem over the support set $S^{(n+1)}$

$$\mathbf{x}^{(n+1)} = \{\| \mathbf{y} - A\mathbf{z} \|_2 \mid \operatorname{supp}(\mathbf{z}) \subset S^{(n+1)}\}$$

obtained by solving the conjugate gradient method on the normal equations to obtain the n-th approximation of $\mathbf{x}$

$$\mathbf{x}^{(n+1)} = \mathrm{CG}(\mathbf{x}^{(n)}, \mathbf{b}, p, d, S^{(n+1)})$$

gives us the *hard thresholding pursuit* algorithm.

Analysis It is known from [17] that the iterative hard thresholding and hard thresholding pursuit algorithms converge if a measurement matrix B satisfies

$$\| B \|_2^2 < 1.$$

Since $G^{[1]}$ and $G^{[2]}$ are the Levi graphs of configurations, the convergence properties of the iterative hard thresholding algorithm and the hard thresholding pursuit algorithm with the measurement matrix A follows from the next theorem.

Theorem 32. *Let A be the incidence matrix of a (v_r, b_k) configuration. Then*

$$\| A \|_2^2 = rk.$$

Proof. First, recall theorems 6.5 and 7.18 of [24] that

$$\| A \|_2^2 = \sigma_{max}^2 = \lambda_{max},$$

where σ_{max} is the largest eigenvalue of A and λ_{max} is the largest eigenvalue of $A^T A$. Then observe that

$$\frac{1}{rk} A^T A$$

is a stochastic matrix. Since the largest eigenvalue of a stochastic matrix is 1 with the eigenvector $\mathbf{1}$, we have

$$\frac{1}{rk} A^T A \mathbf{1} = \mathbf{1}.$$

Multiplying both sides of this equation by rk gives us

$$A^T A \mathbf{1} = rk\mathbf{1}.$$

Therefore,

$$\left\| A^T A \right\|_2^2 = \lambda_{max} = rk.$$

□

Since the bipartite adjacency matrix A of the graph $G_{p,d}^{[1]}$ or the graph $G_{p,d}^{[2]}$ is a (m_p, N_d) configurations, where m and N are determined by the graph chosen, we have

$$\| A \|_2^2 = pd > 1.$$

Thus, iterative hard thresholding and hard thresholding pursuit do not converge in general with the measurement matrix A. However, in [17] an augmented iterative hard thresholding and hard thresholding pursuit is presented by introducing a factor μ. The *iterative hard thresholding-μ* algorithm is defined by iterating

$$\mathbf{x}^{(n+1)} = H_s(\mathbf{x}^{(n)} + \mu A^T(\mathbf{y} - A\mathbf{x}^{(n)}))$$

and the *hard thresholding pursuit-μ* algorithm is defined by iterating

$$S^{(n+1)} = L_s(\mathbf{x}^{(n)} + \mu A^T(\mathbf{y} - A\mathbf{x}^{(n)})).$$

It is know from [17] that these augmented versions of the iterative algorithms converge if

$$\mu \| A \|_2^2 < 1.$$

The paper [17] presented a way of updating this factor μ_n for each iteration $n = 1, 2, \ldots$ is defined, so that we can guarantee convergence. This requires several applications of matrix-vector multiplication, which is not desirable even with matrix-vector multiplication requiring only $\mathcal{O}(sd)$ arithmetic operations. The previous proof provides us with everything we need to avoid having to use this μ_n. We state this result in the following corollary.

Corollary 3. *Let A be the bipartite adjacency matrix of the graph $G_{p,d}^{[1]}$ or the graph $G_{p,d}^{[2]}$ with the parameters $2 \leq d \leq p$. Let $\epsilon > 0$ be arbitrarily small. If we choose*

$$\mu = \frac{1}{pd + \epsilon},$$

then the iterative hard thresholding-μ algorithm and the hard thresholding pursuit-μ converge.

We end this section with a quick remark on the complexity of iterative hard thresholding. Since we do not need to apply the conjugate gradient method, this speeds up the algorithm significantly in comparison to the other non-combinatorial algorithms.

Remarks on Convergence and the Compressed Sensing Problems

Throughout this section, let A be the bipartite adjacency matrix of a "good" lossless expander graph. We define this ambiguously since the exact properties of A are not necessary for the discussion. This allows us to move between different contexts freely.

Recall the equation (4.4) for the restricted isometry constant. Unfortunately, an analogous equation is not known for the restricted expansion constant. As stated in chapter 4, many of the known proofs for the convergence properties of recovery algorithms make use of equation (4.4). Since we also cannot rely on other more common tools that come with the 2-norm of a vector, it is clear that we have to take a much different approach to convergence proofs when the measurement matrix is the bipartite adjacency matrix of a lossless expander graph.

In this thesis, we have spent most of our time on proving results for basis pursuit and quadratically constrained basis pursuit. Both approaches solve their respective versions of the compressed sensing problem when the measurement matrix used is A. It is also shown in [21] that the unique neighborhood algorithm converges to the unique sparse solution as well when A is the measurement matrix. Thus, the unique neighborhood algorithm solves the basic version of the compressed sensing problem as well.

We have shown that the iterative hard thresholing-μ algorithm and the hard thresholding pursuit-μ algorithm converge, but we still have not shown when it converges uniquely to the sparse vector we are looking to recover. Because of this, we have not proven when these algorithms solve the compressed sensing problem when the measurement matrix is A. For orthogonal matching pursuit and compressive sampling matching pursuit, we are essentially in the same position when the measurement matrix is A. It is known from proposition 3.5 of [18] that a sufficient condition for unique sparse vector recovery does exist for the orthogonal matching pursuit algorithm. If we were to show that A can satisfy this property, then we have demonstrated that orthogonal matching pursuit solves the compressed sensing problem with measurement matrix A.

14 Summary of References for Part Three

The numerical linear algebra theory used in chapter 12 was derived primarily from the lecture notes written by Tom Lyche. These lecture notes were used in a course on numerical linear algebra taught by Nils Henrik Risebro. The material on least squares minimization and the conjugate gradient method was primarily derived from these lecture notes. The chapter on classical iterative methods was used in conjunction with the iterative hard thresholding algorithm and the hard thresholding pursuit algorithm.

The selection of compressed sensing recovery algorithms presented here, with the exception of the unique neighborhood algorithm, came from [17] and chapter 3 of [18]. The μ augmented iterative algorithms were also obtained from [17]. The unique neighborhood algorithm and its analysis was obtained from [21].

Corollary 2 is applied in the exact same way to the restricted isometry property in many places throughout compressed sensing. It was originally obtained from chapter 6 in [18].

A Pseudocode of Matrix-Vector Multiplication Algorithms

Algorithm 1 Matrix-Vector Multiplication of $G^{[1]}$: $\mathbf{y} = A\mathbf{z}$

Input: The parameters $2 \leq d \leq p$ of the bipartite adjacency matrix A of $G^{[1]}_{p,d}$ and a vector $\mathbf{z} \in \mathbb{C}^N$ with $N = p^2$.
Output: $\mathbf{v} = A\mathbf{z} \in \mathbb{C}^m$ with $m = dp$.

```
S = supp(z)
v = 0
for j = S do
    x = ⌊(j−1)/p⌋ (mod p)
    y = j − 1 (mod p)
    for a = 0 : d − 1 do
        b = y − ax (mod p)
        i = ap + b + 1
        v(i) = v(i) + z(j)
    end for
end for
```

Algorithm 2 Transposed Matrix-Vector Multiplication of $G^{[1]}$: $\mathbf{z} = A^T\mathbf{v}$

Input: The parameters $2 \leq d \leq p$ of the bipartite adjacency matrix A of $G^{[1]}_{p,d}$ and a vector $\mathbf{v} \in \mathbb{C}^m$ with $m = dp$.
Output: $\mathbf{z} = A^T\mathbf{v} \in \mathbb{C}^N$ with $N = p^2$.

```
S = supp(v)
z = 0
for i = S do
    a = ⌊(i−1)/p⌋ (mod p)
    b = i − 1 (mod p)
    for x = 0 : p − 1 do
        y = ax + b (mod p)
        j = xp + y + 1
        z(j) = z(j) + v(i)
    end for
end for
```

Algorithm 3 Transposed Matrix-Matrix-Vector Multiplication: $\mathbf{w} = A^T A\mathbf{z}$

Input: The parameters $2 \leq d \leq p$ of the bipartite adjacency matrix A of $G_{p,d}^{[1]}$ and a vector $\mathbf{z} \in \mathbb{C}^N$ with $N = p^2$.
Output: $\mathbf{w} = A^T A\mathbf{z} \in \mathbb{C}^N$ with $N = p^2$.

```
S = supp(z)
w = 0
for j_1 = S do
  x_1 = ⌊(j_1 − 1)/p⌋ (mod p)
  y_1 = j_1 − 1 (mod p)
  for j_2 = S \ {j_1} do
    x_2 = ⌊(j_2 − 1)/p⌋ (mod p)
    y_2 = j_2 − 1 (mod p)
    for x = 0 : p − 1 do
      if a x_1 − y_1 = a x_2 − y_2 (mod p) then
        w(j_1), w(j_2) = z(j_1) + z(j_2)
        w(j_2) = z(j_2) + z(j_1)
      end if
    end for
  end for
end for
for j = S do
  w(j) = d z(j)
end for
```

Bibliography

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[2] Afonso S Bandeira, Edgar Dobriban, Dustin G Mixon, and William F Sawin. Certifying the restricted isometry property is hard. *IEEE transactions on information theory*, 59(6):3448–3450, 2013.

[3] Afonso S Bandeira, Matthew Fickus, Dustin G Mixon, and Percy Wong. The road to deterministic matrices with the restricted isometry property. *Journal of Fourier Analysis and Applications*, 19(6):1123–1149, 2013.

[4] Radu Berinde, Anna C Gilbert, Piotr Indyk, Howard Karloff, and Martin J Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 798–805. IEEE, 2008.

[5] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[6] A. Bondy and U.S.R. Murty. *Graph Theory*. Graduate Texts in Mathematics. Springer London, 2007.

[7] Jean Bourgain, Stephen Dilworth, Kevin Ford, Sergei Konyagin, Denka Kutzarova, et al. Explicit constructions of rip matrices and related problems. *Duke Mathematical Journal*, 159(1):145–185, 2011.

[8] Kurt Bryan and Tanya Leise. Making do with less: An introduction to compressed sensing. *Siam Review*, 55(3):547–566, 2013.

[9] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[10] Venkat Chandar. A negative result concerning explicit matrices with the restricted isometry property. *preprint*, 2008.

[11] Charles J Colbourn. *Handbook of Combinatorial Designs*. CRC press, 2010.

[12] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on pure and applied mathematics*, 63(1):1–38, 2010.

[13] Ronald A DeVore. Deterministic constructions of compressed sensing matrices. *Journal of complexity*, 23(4-6):918–925, 2007.

[14] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[15] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[16] Walter Feit and Graham Higman. The nonexistence of certain generalized polygons. *Journal of Algebra*, 32(2):114–131, 1964.

[17] Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[18] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*, volume 1. Birkhäuser Basel, 2013.

[19] Kanke Gao, Stella N Batalama, Dimitris A Pados, and Bruce W Suter. Compressive sampling with generalized polygons. *IEEE Transactions on Signal Processing*, 59(10):4759–4766, 2011.

[20] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[21] Sina Jafarpour, Weiyu Xu, Babak Hassibi, and Robert Calderbank. Efficient and robust compressed sensing using optimized expander graphs. *IEEE Transactions on Information Theory*, 55(9):4299–4308, 2009.

[22] Jon-Lark Kim, Uri N Peled, Irina Perepelitsa, Vera Pless, and Shmuel Friedland. Explicit construction of families of ldpc codes with no 4-cycles. *IEEE transactions on information theory*, 50(10):2378–2388, 2004.

[23] Felix Lazebnik and Vasiliy A Ustimenko. Explicit construction of graphs with an arbitrary large girth and of large size. *Discrete Applied Mathematics*, 60(1-3):275–284, 1995.

[24] Tom Lyche. *Lecture Notes for Mat-inf 4130*. University of Oslo, 2017.

[25] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[26] Terence Tao. Open question: deterministic uup matrices, 2007. `https://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices/`, Last accessed on 2018-05-314.