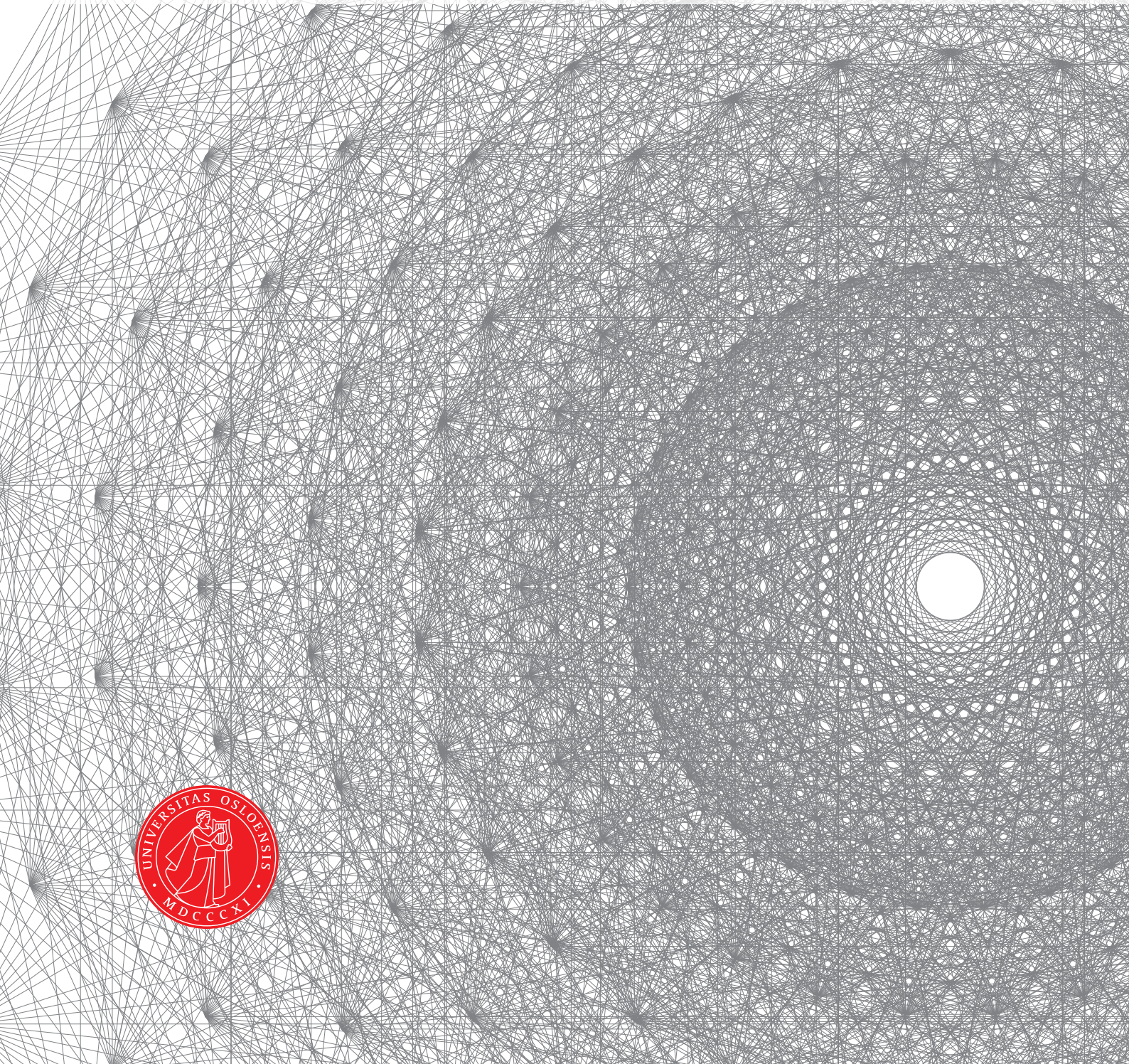
UiO : Department of Mathematics
University of Oslo

On Recent Advances in Compressed Sensing

Teah Kaasa McLean

Master's Thesis, Autumn 2021



This master's thesis is submitted under the master's program *Computational Science*, with program option *Applied Mathematics and Risk Analysis*, 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 has roused great interest in research and many industries over the last few decades. This is because we can recover signals from vastly undersampled measurements, under certain assumptions: *sparsity, incoherence* and *uniform random subsampling*.

However, recent research has shown that the traditional theory yields poor recovery results in many practical cases. This has lead to the development of a new compressed sensing theory, based on local structure in the signals. The new theory defines asymptotic sparsity, asymptotic incoherence and multilevel random subsampling. With these new principles, we see much better recovery results.

In order to apply CS in practice, we need to be able to solve the main optimization problem *basis pursuit* efficiently for large data sets. The spectral projected gradient ℓ_1 (SPGL1) algorithm serves this purpose. It restates the optimization problem as a root finding problem of a single-variable non-linear equation, and utilizes an inexact Newton method to find this root.

The purpose of this text is to give an introduction to the field of compressed sensing, provide the mathematical motivation for the SPGL1 algorithm and highlight some recent advances in compressed sensing.

Acknowledgements

First and foremost, I would like to thank my supervisors Øyvind Ryan and Vegard Antun. Øyvind's door has always been open, and he has been a massive help in figuring out the mathematical details for the SPGL1 algorithm and has given valuable feedback on the drafts of this thesis. Vegard has helped me navigate the recent advances in compressed sensing by suggesting relevant resources and answering any and all of my questions. He has also been a great help with the code for this thesis.

I am so grateful to have met Aasne and Ine in my first year at the University of Oslo. Together we got through all of our mandatory assignments and exams, and we have had many fun conversations during our long lunch breaks. Christian deserves a special thanks for his infectious optimism and for always being willing to discuss details and concepts. I am beyond thankful for the effort he has put into helping me proofread.

Finally, I want to thank Tonje, Stig and my other friends and family for all of their support and words of encouragement. I could not have done it without you.

Contents

List of Acronyms

BP Basis pursuit

BPDN Basis pursuit denoise problem

C-LASSO Constrained LASSO

CS Compressed sensing

DB4 Daubechies 4

DWT Discrete wavelet transform

FWHT Forward Walsh-Hadamard transform

IDWT Inverse discrete wavelet transform

IWHT Inverse Walsh-Hadamard transform

NSP Null space property

PSNR Peak signal-to-noise ratio

QCBP Quadratically constrained basis pursuit

RIC Restricted isometry constant

RICL Restricted isometry constant in levels

RIP Restricted isometry property

RIPL Restricted isometry property in levels

RNSP Robust null space property

SNSP Stable null space property

SPG Spectral projected gradient

SPGL1 Spectral projected gradient ℓ_1

U-LASSO Unconstrained LASSO

WHT Walsh-Hadamard transform

Code

The figures in this thesis have been produced utilizing the Compressive Imaging library [4] by Vegard Antun and the open source MATLAB solver SPGL1 [9] by Michael P. Friedlander and Ewout van den Berg. For interested readers, the test images and the scripts for generating the figures in this text can be found at the author's GitHub page[1].

[1] https://github.com/teahkm/thesis_code

CHAPTER 1

Introduction

Compressed sensing (CS) is motivated by the observation that many natural signals are compressible. For example, if we perform an appropriate change of basis on a natural image, only a small percentage of the coefficients are non-zero and necessary to encode the image. We say that the image is *sparse* in the new basis. We can discard the coefficients that are zero and be left with a compressed image that to the human eye looks identical to the original. However, we are wasting time and resources collecting samples that are to be discarded.

With CS, we wish to collect only the samples that will contain interesting information. Then from these samples, we can recover the original signal. This corresponds to solving the linear system of equations

$$A\mathbf{x} = \mathbf{y}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the signal, $A \in \mathbb{R}^{m \times N}$ is a matrix describing how the measurements are sampled and $\mathbf{y} \in \mathbb{R}^m$ is the subsampled measurements. Because we are subsampling, $m < N$ and the system is underdetermined. This means that there are infinitely many solutions $\mathbf{x}$. We seek to find the sparsest $\mathbf{x}$ that fits the measurements. One can show that this boils down to solving the convex optimization problem *basis pursuit (BP)*

$$\underset{\mathbf{z} \in \mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \text{subject to} \quad \|A\mathbf{z} - \mathbf{y}\|_2 = 0. \tag{1.1}$$

We know from traditional signal processing that there is a lower bound on how many samples are necessary to recover the signal. The Shannon-Nyquist sampling theorem tells us that to reconstruct a signal with highest frequency ω, we only need a sampling rate higher than 2ω. Unfortunately, for high frequency signals, this number can still be very large. Making a few additional reasonable assumptions on the signal, the CS theory provides a much smaller estimate on the number of samples required to recover the signal exactly. If we have an incoherent measurement matrix A and we sample uniformly at random, all vectors $\mathbf{x}$ with at most s non-zero coefficients can be recovered from $\mathbf{y} = A\mathbf{x}$ provided

$$m \geq Cs\mu L,$$

where $C > 0$ is a universal constant, μ is the coherence of the matrix and L is a log-factor.

For CS to work in practice, we need to be able to solve the optimization problem (1.1) efficiently for large data sets. Several algorithms exist for solving

this problem, however most of these require that the measurement matrix be stored explicitly. This is intractable for large data sets due to the limitations on computer memory. The spectral projected gradient ℓ_1 (SPGL1) algorithm [9] allows the matrix to be an operator, thereby handling the memory issue. This algorithm solves BP and two other related problems for large data sets. It relates the optimization problems to finding the root of a single-variable non-linear equation. The root finding is done by applying an inexact Newton method.

Standard CS yields poor reconstruction results in most practical setups. Empirical observations suggest that there is an asymptotic structure in the sparsity of a signal and the coherence of the measurement matrix, and that this structure plays an important role in the recovery of the signal. Therefore, the global notions of sparsity, coherence and sampling in the standard CS theory do not suffice. Instead, new local versions of these principles can be defined to improve recovery results and better describe how CS works in practice. The new principles are asymptotic sparsity, asymptotic incoherence and multilevel random sampling.

This thesis is an exposition of the literature in the field of compressed sensing. We will state and prove some of the important results from the classical CS theory, provide mathematical details for the SPGL1 algorithm, and present and discuss the new local CS theory.

Convention

The two main references in this thesis are [18] for the CS theory and [10] for the SPGL1 algorithm. For most of the thesis, we will follow the notation established in [18]. However, in Chapter 5 we will follow the notation in [10], for example using $\mathbf{b} \in \mathbb{R}^m$ for the measured data instead of $\mathbf{y} \in \mathbb{R}^m$, so as to better align with the source code for the SPGL1 algorithm.

We will be working with the application of recovering images with Hadamard sampling. In this case, our signal $\mathbf{x}$ has real coefficients, and therefore all the theory is stated for the case $\mathbf{x} \in \mathbb{R}^N$. We note, however, that most of the results also hold in the case where $\mathbf{x} \in \mathbb{C}^N$. Proofs for the complex case can be seen in the literature.

Outline

The rest of the thesis is organized in the following manner:

Chapter 2 reviews the most important results from the standard compressed sensing theory.

Chapter 3 gives a brief introduction to wavelets and the Hadamard transform and examines the coherence between them.

Chapter 4 provides thorough mathematical motivation for the SPGL1 algorithm and outlines some parts of the algorithm.

Chapter 5 demonstrates the flaws in the traditional CS theory through numerical experiments and states new theory to support how CS works in practice.

Chapter 6 summarizes the thesis.

Appendix A features some extra derivations.

CHAPTER 2

Traditional Compressed Sensing

In this chapter we introduce the traditional compressed sensing problem, establish some important results and discuss some concerns in the theory.

2.1 Notation

Throughout most of this thesis we use the following notation. We use the notation $[N]$ to mean the set $\{1, 2, \ldots, N\}$ and $\text{card}(S)$ to mean the cardinality of the set S. We use $\overline{S}$ to denote the complement of the set S, i.e., the set $[N] \setminus S$. We write $A \lesssim B$ to mean there exists a universal constant $C > 0$ such that $A \leq CB$, and similarly for $A \gtrsim B$.

For a vector $\mathbf{v} \in \mathbb{R}^N$ and a set $S \subset [N]$, we use the notation $\mathbf{v}_S$ to mean either the vector in $\mathbb{R}^S$ which is the restriction of $\mathbf{v}$ to the indices in S, or the vector in $\mathbb{R}^N$ which coincides with $\mathbf{v}$ on the indices in S and is extended to zero outside S.

For subsampling, we use the notation Ω for a subset of $[N]$ with $\text{card}(\Omega) = m$. We use P_Ω for the projection matrix in $\mathbb{R}^{m \times N}$ that selects which of the m rows to sample.

2.2 Sparse Solutions to Underdetermined Systems

The key assumption in order for compressed sensing to work is that the vectors we wish to recover are *sparse*. In this section we define the notions of sparsity and compressibility and set up the compressed sensing problem.

Definition 2.2.1. [18, Definition 2.1] The *support* of a vector $\mathbf{x} \in \mathbb{R}^N$ is the set of indices for which $\mathbf{x}$ has non-zero entries,

$$\text{supp}(\mathbf{x}) \coloneqq \{j \in [N] : x_j \neq 0\}.$$

A vector $\mathbf{x} \in \mathbb{R}^N$ is called *s-sparse* if it has at most s non-zero entries, or that

$$\|\mathbf{x}\|_0 \coloneqq \text{card}(\text{supp}(\mathbf{x})) \leq s.$$

It is standard to use the norm notation $\|\mathbf{x}\|_0$ to mean the number of non-zero entries in the vector $\mathbf{x}$. However, it is important to note that $\|\mathbf{x}\|_0$ is not a norm nor a quasinorm. This use of the notation comes from the observation

that the limit of the ℓ_p-norm of $\mathbf{x}$ when $p \to 0$ is the number of non-zero entries in $\mathbf{x}$, i.e.,

$$\lim_{p\to 0}\|\mathbf{x}\|_p^p = \lim_{p\to 0}\sum_{j=1}^{N}|x_j|^p = \sum_{j=1}^{N}\mathbf{1}_{\{x_j\neq 0\}} = \operatorname{card}(\{j \in [N] : x_j \neq 0\}),$$

where $\mathbf{1}_{\{x_j\neq 0\}}$ is 1 when $x_j \neq 0$ and 0 when $x_j = 0$.

In practice, it is unrealistic to assume that a vector is exactly sparse. Instead, we consider vectors that are nearly sparse, or *compressible.* A vector $\mathbf{x}$ is compressible if its distance to an s-sparse vector decays quickly in s. This distance is measured by the vector's ℓ_p-error of best s-term approximation.

Definition 2.2.2. [18, Definition 2.2] For $p > 0$, the *ℓ_p-error of best s-term approximation* to a vector $\mathbf{x} \in \mathbb{R}^N$ is defined by

$$\sigma_s(\mathbf{x})_p \coloneqq \inf\{\|\mathbf{x} - \mathbf{z}\|_p \colon \mathbf{z} \in \mathbb{R}^N \text{ is } s\text{-sparse}\}.$$

The Main Optimization Problem

We are working with the equation $A\mathbf{x} = \mathbf{y}$, where $\mathbf{x} \in \mathbb{R}^N$ is the sparse signal, $A \in \mathbb{R}^{m\times N}$ with $m < N$ describes the measurement process, and $\mathbf{y} \in \mathbb{R}^m$ is the measured data. Our goal is to solve this equation for $\mathbf{x}$. Since $m < N$, this system is underdetermined, which means there are infinitely many solutions $\mathbf{x}$. We are interested in finding the *sparsest* $\mathbf{x}$ that satisfies this equation, i.e., solving

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_0 \quad \text{subject to} \quad A\mathbf{z} = \mathbf{y}. \tag{P_0}$$

Unfornutately, (P_0) is non-convex and NP-hard in general, see Theorem 2.17 in [18]. However, because $\|\mathbf{z}\|_q^q$ tends to $\|\mathbf{z}\|_0$ as $q > 0$ tends to 0, we can instead consider the following optimization problem:

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_q \quad \text{subject to} \quad A\mathbf{z} = \mathbf{y}. \tag{P_q}$$

For most values of q we are not able to solve (P_q) efficiently, or the solution does not coincide with the solution to (P_0). We will see that the only appropriate value is $q = 1$. First, we show that for $q > 1$ even 1-sparse vectors are not solutions.

Proof. We let $q > 1$ and let $A \in \mathbb{R}^{m\times N}$ with $m < N$. We assume for contradiction that all 1-sparse vectors are minimizers of (P_q). This implies that all standard basis vectors $\mathbf{e}_j$ are minimizers of (P_q), since they are 1-sparse. We note that since $m < N$, the kernel of A is non-trivial, because the columns of A are linearly dependent. Thus, there exists a vector $\mathbf{v} \neq \mathbf{0}$ such that $A\mathbf{v} = \mathbf{0}$. We choose an index j such that $v_j \neq 0$. Then, for any t we can define the function

$$g(t) \coloneqq \|\mathbf{e}_j + t\mathbf{v}\|_q^q = |1 + tv_j|^q + \sum_{k\neq j}|tv_k|^q = |1 + tv_j|^q + |t|^q\sum_{k\neq j}|v_k|^q.$$

We consider two new functions:

$$g_+(t) = (1 + tv_j)^q + t^q\sum_{k\neq j}|v_k|^q,$$

$$g_-(t) = (1 + tv_j)^q + (-t)^q\sum_{k\neq j}|v_k|^q.$$

Suppose that $|t| < 1/v_j$. Then we have $(1 + t\,v_j) > 0$ and consequently $|1 + t\,v_j| = 1 + t\,v_j$. Then, for $t \geq 0$, $g(t)$ corresponds to $g_+(t)$ because $|t| = t$. For $t < 0$, $g(t)$ corresponds to $g_-(t)$ because $|t| = -t$.

We compute the derivatives with respect to t:

$$g'_+(t) = qv_j(1 + tv_j)^{q-1} + qt^{q-1} \sum_{k \neq j} |v_k|^q,$$
$$g'_-(t) = qv_j(1 + tv_j)^{q-1} - q(-t)^{q-1} \sum_{k \neq j} |v_k|^q.$$

For $q > 1$, we have $(q - 1) > 0$, and thus taking the limit as t tends to 0 yields

$$\lim_{t \to 0^+} g'_+(t) = qv_j(1)^{q-1} = qv_j,$$
$$\lim_{t \to 0^-} g'_-(t) = qv_j(1)^{q-1} = qv_j.$$

By this, we have that the derivative of $g(0) = qv_j$. Since $q > 1$ and $v_j \neq 0$, the derivative of $g(t)$ at $t = 0$ is non-zero. But then $g(t)$ cannot have a minimum at $t = 0$. Since $\|\mathbf{e}_j\|_q$ corresponds to $\|\mathbf{e}_j + t\mathbf{v}\|_q^q$ when $t = 0$, the 1-sparse vector $\mathbf{e}_j$ cannot be a minimizer of (P_q), since $\|\mathbf{e}_j\|_q$ is not a minimum. ■

For the values $0 < q < 1$, it can be shown that the problem (P_q) is non-convex and also NP-hard in general. Therefore, the critical value is $q = 1$. For $q = 1$, we get the convex optimization problem referred to as ℓ_1-minimization or *basis pursuit (BP)*,

$$\underset{\mathbf{z} \in \mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \text{subject to} \quad A\mathbf{z} = \mathbf{y}. \tag{BP}$$

The next few sections will discuss conditions that ensure that (BP) solves (P_0).

2.3 Null Space Properties

We now look into conditions on the matrix A that guarantee exact reconstruction of sparse vectors or approximate reconstruction of compressible vectors.

The Null Space Property

A necessary and sufficient condition for exact recovery of sparse vectors via basis pursuit is the *null space property (NSP)*.

Definition 2.3.1. [18, Definition 4.1] A matrix $A \in \mathbb{R}^{m \times N}$ is said to satisfy the *null space property* relative to a set $S \subset [N]$ if

$$\|\mathbf{v}_S\|_1 < \|\mathbf{v}_{\overline{S}}\|_1 \quad \text{for all} \quad \mathbf{v} \in \ker A \setminus \{\mathbf{0}\}. \tag{2.1}$$

If a matrix satisfies the null space property relative to any set $S \subset [N]$ with $\text{card}(S) \leq s$, it is said to satisfy the null space property of order s.

There are two useful reformulations of the NSP. The first is obtained by adding $\|\mathbf{v}_S\|_1$ to both sides of the inequality:

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

$$2\|\mathbf{v}_S\|_1 < \|\mathbf{v}\|_1. \tag{2.2}$$

The second is obtained by choosing S as an index set of s largest absolute entries of $\mathbf{v}$ and adding $\|\mathbf{v}_{\overline{S}}\|_1$ to both sides:

$$\begin{aligned} \|\mathbf{v}_S\|_1 + \|\mathbf{v}_{\overline{S}}\|_1 &< \|\mathbf{v}_{\overline{S}}\|_1 + \|\mathbf{v}_{\overline{S}}\|_1 \\ \|\mathbf{v}\|_1 &< 2\|\mathbf{v}_{\overline{S}}\|_1 \\ \|\mathbf{v}\|_1 &< 2\sigma_s(\mathbf{v})_1, \end{aligned} \tag{2.3}$$

where we have used that $\|\mathbf{v}_{\overline{S}}\|_1 = \sigma_s(\mathbf{v})_1$. To see this, recall that $\sigma_s(\mathbf{v})_1 = \inf_{\|\mathbf{z}\|_0 \le s} \|\mathbf{v} - \mathbf{z}\|_1$. Since $\mathbf{v}_S$ contains the s largest absolute entries of $\mathbf{v}$, we have $\|\mathbf{v}_S\|_0 \le s$, and $\|\mathbf{v}_{\overline{S}}\|_1 = \|\mathbf{v} - \mathbf{v}_S\|_1$ satisfies the infimum.

The following theorem states that the NSP is a necessary and sufficient condition for exact recovery.

Theorem 2.3.2. *[18, Theorem 4.4] Given a matrix $A \in \mathbb{R}^{m \times N}$, every vector $\mathbf{x} \in \mathbb{R}^N$ supported on a set S is the unique solution of* (BP) *with $\mathbf{y} = A\mathbf{x}$ if and only if A satisfies the null space property relative to S.*

Proof. Let S be a fixed index set and assume that every vector $\mathbf{x} \in \mathbb{R}^N$ supported on S is the unique minimizer of $\|\mathbf{z}\|_1$ subject to $A\mathbf{z} = A\mathbf{x}$. Thus, for any vector $\mathbf{v} \in \ker A \setminus \{\mathbf{0}\}$, the vector $\mathbf{v}_S$ is the unique minimizer of $\|\mathbf{z}\|_1$ subject to $A\mathbf{z} = A\mathbf{v}_S$. That is, $\|\mathbf{v}_S\|_1 < \|\mathbf{z}\|_1$ for any $\mathbf{z} \in \mathbb{R}^N$ such that $A\mathbf{z} = A\mathbf{v}_S$. Since $\mathbf{v} = \mathbf{v}_S + \mathbf{v}_{\overline{S}}$ and $\mathbf{v} \in \ker A \setminus \{\mathbf{0}\}$, we have

$$\begin{aligned} A(\mathbf{v}_S + \mathbf{v}_{\overline{S}}) &= 0 \\ A\mathbf{v}_S + A\mathbf{v}_{\overline{S}} &= 0 \\ A\mathbf{v}_S &= -A\mathbf{v}_{\overline{S}} \\ A\mathbf{v}_S &= A(-\mathbf{v}_{\overline{S}}). \end{aligned}$$

Since $\mathbf{v} \neq \mathbf{0}$, we must have $\mathbf{v}_S \neq -\mathbf{v}_{\overline{S}}$. Then $\mathbf{v}_S$ is a unique minimizer of $A\mathbf{v}_S = A(-\mathbf{v}_{\overline{S}})$, i.e., $\|\mathbf{v}_S\|_1 < \|\mathbf{v}_{\overline{S}}\|$ and the null space property relative to S is satisfied.

Conversely, we assume that the null space property relative to S holds. Let $\mathbf{x} \in \mathbb{R}^N$ be supported on S and let $\mathbf{z} \in \mathbb{R}^N$ be such that $\mathbf{z} \neq \mathbf{x}$ and $A\mathbf{z} = A\mathbf{x}$. We consider the vector $\mathbf{v} \coloneqq \mathbf{x} - \mathbf{z} \in \ker A \setminus \{\mathbf{0}\}$. Since $\mathbf{x}$ is supported on S, we have $\mathbf{v}_S = \mathbf{x}_S - \mathbf{z}_S = \mathbf{x} - \mathbf{z}_S$ and $\mathbf{v}_{\overline{S}} = \mathbf{x}_{\overline{S}} - \mathbf{z}_{\overline{S}} = -\mathbf{z}_{\overline{S}}$. Then, using the triangle inequality and the condition for the NSP, we obtain

$$\begin{aligned} \|\mathbf{x}\|_1 \le \|\mathbf{x} - \mathbf{z}_S\|_1 + \|\mathbf{z}_S\|_1 &= \|\mathbf{v}_S\|_1 + \|\mathbf{z}_S\|_1 \\ &< \|\mathbf{v}_{\overline{S}}\|_1 + \|\mathbf{z}_S\|_1 = \|-\mathbf{z}_{\overline{S}}\|_1 + \|\mathbf{z}_S\|_1 = \|\mathbf{z}\|_1. \end{aligned}$$

We have shown that $\|\mathbf{x}\|_1 < \|\mathbf{z}\|_1$, i.e., $\mathbf{x}$ is the unique minimizer of $\|\mathbf{z}\|_1$ subject to $A\mathbf{z} = A\mathbf{x}$. ■

Note that if $\mathbf{x}$ does not need to be unique, then $\|\mathbf{v}_S\|_1 \le \|\mathbf{v}_{\overline{S}}\|_1$ is a necessary and sufficient condition for exact recovery.

If we let S vary, we obtain the following theorem as a consequence of Theorem 2.3.2.

Theorem 2.3.3. *[18, Theorem 4.5] Given a matrix $A \in \mathbb{R}^{m\times N}$, every s-sparse vector $\mathbf{x} \in \mathbb{R}^N$ is the unique solution of* (BP) *with $\mathbf{y} = A\mathbf{x}$ if and only if A satisfies the null space property of order s.*

This theorem tells us that when the NSP of order s holds, the solution to (BP) is the solution to the ℓ_0-minimization problem. Let us assume that every s-sparse vector $\mathbf{x}$ is recovered via ℓ_1-minimization from $\mathbf{y} = A\mathbf{x}$. Let $\mathbf{z}$ be the solution to the ℓ_0-minimization problem with $\mathbf{y} = A\mathbf{x}$. Then $\|\mathbf{z}\|_0 \leq \|\mathbf{x}\|_0$. Since $\mathbf{x}$ is s-sparse, then so is $\mathbf{z}$. But since every s-sparse vector is the unique ℓ_1-minimizer by Theorem 2.3.3, it follows that $\mathbf{x} = \mathbf{z}$.

The Stable Null Space Property

The null space property we have discussed so far assumes the vectors are perfectly sparse. However, this is rarely the case. More often, the vectors are only close to sparse vectors. If we strengthen the null space property, we have that basis pursuit is stable with respect to sparsity defect, i.e., we can recover the vector with an error that is controlled by its distance to s-sparse vectors.

Definition 2.3.4. [18, Definition 4.11] A matrix $A \in \mathbb{R}^{m\times N}$ is said to satisfy the *stable null space property (SNSP)* with constant $0 < \rho < 1$ relative to a set $S \subset [N]$ if

$$\|\mathbf{v}_S\|_1 \leq \rho\|\mathbf{v}_{\overline{S}}\|_1 \quad \text{for all} \quad \mathbf{v} \in \ker A. \tag{2.4}$$

We say that A satisfies the stable null space property of order s if it satisfies the stable null space property with constant $0 < \rho < 1$ relative to any set $S \subset [N]$ with $\text{card}(S) \leq s$.

It is pretty straight forward to show that the NSP implies the SNSP using the formulation (2.2), see page 85 in [18].

The main stability result is found in the following theorem, but will be improved upon in the stronger Theorem 2.3.6.

Theorem 2.3.5. *[18, Theorem 4.12] Suppose that a matrix $A \in \mathbb{R}^{m\times N}$ satisfies the stable null space property of order s with constant $0 < \rho < 1$. Then, for any $\mathbf{x} \in \mathbb{R}^N$, a solution $\mathbf{x}^\#$ of* (BP) *with $\mathbf{y} = A\mathbf{x}$ approximates the vector $\mathbf{x}$ with ℓ_1-error*

$$\|\mathbf{x} - \mathbf{x}^\#\|_1 \leq \frac{2(1+\rho)}{(1-\rho)}\sigma_s(\mathbf{x})_1. \tag{2.5}$$

From this theorem we see that we can recover the vector with an error that is controlled by its distance to an s-sparse vector. We recall that the distance to an s-sparse vector is measured by the ℓ_p error of best s-term approximation $\sigma_s(\mathbf{x})_p$.

The following theorem improves the previous theorem, and says that the distance between an s-sparse vector $\mathbf{x} \in \mathbb{R}^N$ and a vector $\mathbf{z} \in \mathbb{R}^N$ satisfying $A\mathbf{z} = A\mathbf{x}$ is controlled by the difference between their norms if and only if the SNSP holds.

Theorem 2.3.6. *[18, Theorem 4.14] The matrix $A \in \mathbb{R}^{m\times N}$ satisfies the stable null space property with constant $0 < \rho < 1$ relative to S if and only if*

$$\|\mathbf{z}-\mathbf{x}\|_1 \leq \frac{1+\rho}{1-\rho}(\|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1) \tag{2.6}$$

for all vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$ with $A\mathbf{z} = A\mathbf{x}$.

In order to prove this theorem, we need the following lemma:

Lemma 2.3.7. *[18, Lemma 4.15] Given a set $S \subset [N]$ and vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$,*

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

Proof. First we note that

$$\|\mathbf{x}\|_1 = \|\mathbf{x}_{\overline{S}}\|_1 + \|\mathbf{x}_S\|_1 \leq \|\mathbf{x}_{\overline{S}}\|_1 + \|(\mathbf{x}-\mathbf{z})_S\|_1 + \|\mathbf{z}_S\|_1,$$

and

$$\|(\mathbf{x}-\mathbf{z})_{\overline{S}}\|_1 \leq \|\mathbf{x}_{\overline{S}}\|_1 + \|\mathbf{z}_{\overline{S}}\|_1.$$

Taking the sum of these two inequalities yields

$$\begin{aligned}\|\mathbf{x}\|_1 + \|(\mathbf{x}-\mathbf{z})_{\overline{S}}\|_1 &\leq \|\mathbf{x}_{\overline{S}}\|_1 + \|(\mathbf{x}-\mathbf{z})_S\|_1 + \|\mathbf{z}_S\|_1 + \|\mathbf{x}_{\overline{S}}\|_1 + \|\mathbf{z}_{\overline{S}}\|_1 \\ \|(\mathbf{x}-\mathbf{z})_{\overline{S}}\|_1 &\leq \|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + \|(\mathbf{x}-\mathbf{z})_S\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1,\end{aligned}$$

which is the desired inequality. ■

Proof of Theorem 2.3.6. First, we assume that the matrix A satisfies (2.6) for all vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$ such that $A\mathbf{z} = A\mathbf{x}$. Let $\mathbf{v} \in \ker A$. Then, since $A\mathbf{v}_{\overline{S}} = A(-\mathbf{v}_S)$, we can apply (2.6) with $\mathbf{x} = -\mathbf{v}_S$ and $\mathbf{z} = \mathbf{v}_{\overline{S}}$.

$$\begin{aligned}\|\mathbf{v}_{\overline{S}} - (-\mathbf{v}_S)\|_1 &\leq \frac{1+\rho}{1-\rho}(\|\mathbf{v}_{\overline{S}}\|_1 - \|\mathbf{v}_S\|_1 + 2\|(\mathbf{v}_S)_{\overline{S}}\|_1) \\ \|\mathbf{v}\|_1 &\leq \frac{1+\rho}{1-\rho}(\|\mathbf{v}_{\overline{S}}\|_1 - \|\mathbf{v}_S\|_1).\end{aligned}$$

Rewriting $\|\mathbf{v}\|_1$ as $\|\mathbf{v}_S\|_1 + \|\mathbf{v}_{\overline{S}}\|_1$ and multiplying by $(1-\rho)$ on both sides of the inequality yields

$$(1-\rho)(\|\mathbf{v}_S\|_1 + \|\mathbf{v}_{\overline{S}}\|_1) \leq (1+\rho)(\|\mathbf{v}_{\overline{S}}\|_1 - \|\mathbf{v}_S\|_1).$$

Rearranging the terms, we get

$$\begin{aligned}\|\mathbf{v}_S\|_1 + \|\mathbf{v}_{\overline{S}}\|_1 - \rho\|\mathbf{v}_S\|_1 - \rho\|\mathbf{v}_{\overline{S}}\|_1 &\leq \|\mathbf{v}_{\overline{S}}\|_1 - \|\mathbf{v}_S\|_1 + \rho\|\mathbf{v}_{\overline{S}}\|_1 - \rho\|\mathbf{v}_S\|_1 \\ 2\|\mathbf{v}_S\|_1 &\leq 2\rho\|\mathbf{v}_{\overline{S}}\|_1 \\ \|\mathbf{v}_S\|_1 &\leq \rho\|\mathbf{v}_{\overline{S}}\|_1,\end{aligned}$$

which is the requirement for the SNSP.

Conversely, we assume that A satisfies the SNSP with constant $0 < \rho < 1$ relative to S. For $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$ with $A\mathbf{z} = A\mathbf{x}$, since $\mathbf{v} \coloneqq \mathbf{z} - \mathbf{x} \in \ker A$, the SNSP yields

$$\|\mathbf{v}_S\|_1 \leq \rho\|\mathbf{v}_{\overline{S}}\|_1. \tag{2.7}$$

Lemma 2.3.7 gives

$$\|(\mathbf{x}-\mathbf{z})_{\overline{S}}\|_1 = \|\mathbf{v}_{\overline{S}}\|_1 \leq \|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + \|\mathbf{v}_S\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1. \tag{2.8}$$

If we substitute (2.7) into (2.8), we get

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

Rearranging the terms gives

$$\begin{aligned}\|\mathbf{v}_{\overline{S}}\|_1 - \rho\|\mathbf{v}_{\overline{S}}\|_1 &\leq \|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1\\ (1-\rho)\|\mathbf{v}_{\overline{S}}\|_1 &\leq \|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1.\end{aligned}$$

Since $\rho < 1$, we can divide by $(1-\rho)$ without flipping the inequality,

$$\|\mathbf{v}_{\overline{S}}\|_1 \leq \frac{1}{1-\rho}(\|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1). \tag{2.9}$$

Now, using (2.7) once more and (2.9), we derive the desired inequality:

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

■

The Robust Null Space Property

We know that having perfectly sparse vectors is rare. Similarly, it is not realistic to measure a signal $\mathbf{x} \in \mathbb{R}^N$ with infinite precision. This means that we will have some error in our measurement vector $\mathbf{y} \in \mathbb{R}^m$. In this case, $\mathbf{y}$ is an approximation of $A\mathbf{x}$ with

$$\|A\mathbf{x} - \mathbf{y}\| \leq \eta$$

for some $\eta \geq 0$ and for some norm $\|\cdot\|$ on $\mathbb{R}^N$. We now look to solve the convex optimization problem

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \text{subject to} \quad \|A\mathbf{x} - \mathbf{y}\| \leq \eta. \tag{$P_{1,\eta}$}$$

By strengthening the null space property further, we can guarantee that the basis pursuit algorithm for $(P_{1,\eta})$ is robust with respect to measurement error.

Definition 2.3.8. [18, Definition 4.17] A matrix $A \in \mathbb{R}^{m\times N}$ is said to satisfy the *robust null space property (RNSP)* (with respect to $\|\cdot\|$) with constants $0 < \rho < 1$ and $\tau > 0$ relative to a set $S \in [N]$ if

$$\|\mathbf{v}_S\|_1 < \rho\|\mathbf{v}_{\overline{S}}\|_1 + \tau\|A\mathbf{v}\| \quad \text{for all} \quad \mathbf{v} \in \mathbb{R}^N. \tag{2.10}$$

The RNSP definition can vary slightly in the literature. For example, in [2] we have the factor $\rho/\sqrt{s}$ in front of $\|\mathbf{v}_{\overline{S}}\|$. However, this does not change the property, as this factor is still between 0 and 1.

The RNSP implies the SNSP. To see this, we note that the condition for RNSP is similar to the condition for SNSP, except for the addition of the penalty term $\tau\|A\mathbf{v}\|$. The penalty term is due to the vector $\mathbf{v}$ no longer being required to be in the null space of A. If, however, we have $\mathbf{v} \in \ker A$, then $\|A\mathbf{v}\| = 0$ and the RNSP becomes the SNSP.

The following theorem is the main robustness result. It is analogous to Theorem 2.3.5 for stability, and states that under the RNSP, we can recover a vector $\mathbf{x}$ with an error that is controlled by the distance from $\mathbf{x}$ to an s-sparse vector.

Theorem 2.3.9. *[18, p. 86] Suppose that a matrix $A \in \mathbb{R}^{m\times N}$ satisfies the robust null space property of order s with constants $0 < \rho < 1$ and $\tau > 0$. Then, for any $\mathbf{x} \in \mathbb{R}^N$, a solution $\mathbf{x}^\#$ of $(P_{1,\eta})$ with $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ and $\|\mathbf{e}\| \leq \eta$ approximates the vector $\mathbf{x}$ with ℓ_1-error*

$$\|\mathbf{x} - \mathbf{x}^\#\|_1 \leq \frac{2(1+\rho)}{(1-\rho)}\sigma_s(\mathbf{x})_1 + \frac{4\tau}{1-\rho}\eta. \tag{2.11}$$

We leave out the proof for this theorem, and instead prove the following, stronger theorem, which is analogous to Theorem 2.3.6. The second part of the proof in [18] is slightly incorrect. The proof in this text is an improved version by the author and her supervisors.

Theorem 2.3.10. *[18, Theorem 4.20] The matrix $A \in \mathbb{R}^{m\times N}$ satisfies the robust null space property with constants $0 < \rho < 1$ and $\tau > 0$ relative to S if and only if*

$$\|\mathbf{z} - \mathbf{x}\|_1 \leq \frac{1+\rho}{1-\rho}(\|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1) + \frac{2\tau}{1-\rho}\|A(\mathbf{z}-\mathbf{x})\| \tag{2.12}$$

for all vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$.

Proof. First, we assume that A satisfies (2.12) for all vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$. Then, by setting $\mathbf{x} = -\mathbf{v}_S$ and $\mathbf{z} = \mathbf{v}_{\overline{S}}$ for $\mathbf{v} \in \mathbb{R}^N$, we get

$$\|\mathbf{z} - \mathbf{x}\|_1 = \|\mathbf{v}\|_1 \leq \frac{1+\rho}{1-\rho}(\|\mathbf{v}_{\overline{S}}\|_1 - \|\mathbf{v}_S\|_1) + \frac{2\tau}{1-\rho}\|A\mathbf{v}\|,$$

where the term $2\|\mathbf{x}_{\overline{S}}\|$ vanishes. Similar to the proof of Theorem 2.3.6, we can rearrange these terms to get

$$\begin{aligned}(1-\rho)(\|\mathbf{v}_S\|_1 + \|\mathbf{v}_{\overline{S}}\|_1) &\leq (1+\rho)(\|\mathbf{v}_{\overline{S}}\|_1 - \|\mathbf{v}_S\|_1) + 2\tau\|A\mathbf{v}\| \\ \|\mathbf{v}_S\|_1 &\leq \rho\|\mathbf{v}_{\overline{S}}\|_1 + \tau\|A\mathbf{v}\|,\end{aligned}$$

which is the condition for the RNSP with constants $0 < \rho < 1$ and $\tau > 0$ relative to S.

Conversely, we will assume that A satisfies the RNSP with constant $0 < \rho < 1$ and $\tau > 0$ relative to S. We let $\mathbf{v} = \mathbf{z} - \mathbf{x}$ for arbitrary vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$. We combine the condition for the RNSP with Lemma 2.3.7 and get

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

We rearrange this inequality by moving the $\|\mathbf{v}_S\|_1$-terms to the left-hand side and then dividing by $(1-\rho)$ on both sides, and get

$$\|\mathbf{v}_S\|_1 \leq \frac{1}{1-\rho}(\rho(\|\mathbf{z}\|_1 - \|\mathbf{x}\|_1 + 2\|\mathbf{x}_{\overline{S}}\|_1) + \tau\|A\mathbf{v}\|). \tag{2.13}$$

By first using the condition for the RNSP, then Lemma 2.3.7, and then (2.13), we get

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

To get the desired inequality, we distribute the factors in the inequality above and combine like terms. Then, since $(1+\rho) + \frac{(1+\rho)\cdot\rho}{1-\rho} = \frac{(1+\rho)}{(1-\rho)}$, we get

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

■

We can improve the robustness result by replacing the ℓ_1-error estimate with an ℓ_p-error estimate for $p \geq 1$. For this, we need the ℓ_q-robust null space property, defined below.

Definition 2.3.11. [18, Definition 4.21] Given $q \geq 1$, the matrix $A \in \mathbb{R}^{m\times N}$ is said to satisfy the *ℓ_q-robust null space property* of order s (with respect to $\|\cdot\|$) with constants $0 < \rho < 1$ and $\tau > 0$ if, for any set $S \subset [N]$ with $\mathrm{card}(S) \leq s$,

$$\|\mathbf{v}_S\|_p \leq \frac{\rho}{s^{1-1/p}}\|\mathbf{v}_{\overline{S}}\|_1 + \tau s^{1/p-1/q}\|A\mathbf{v}\| \quad \text{for all} \quad \mathbf{v} \in \mathbb{R}^N.$$

The second main result establishes the robustness of the quadratically constrained basis pursuit algorithm, i.e., eq. $(P_{1,\eta})$ with the two-norm.

Theorem 2.3.12. *[18, Theorem 4.22] Suppose that $A \in \mathbb{R}^{m\times N}$ satsifies the ℓ_2-robust null space property of order s with constants $0 < \rho < 1$ and $\tau > 0$. Then, for any $\mathbf{x} \in \mathbb{R}^N$, a solution $\mathbf{x}^\#$ of $(P_{1,\eta})$ with $\|\cdot\| = \|\cdot\|_2$, $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ and $\|\mathbf{e}\|_2 \leq \eta$ approximates the vector $\mathbf{x}$ with ℓ_p-error*

$$\|\mathbf{x} - \mathbf{x}^\#\|_p \leq \frac{C}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + Ds^{1/p-1/2}\eta, \quad 1 \leq p \leq 2,$$

for some constants $C, D > 0$ depending only on ρ and τ.

Proof. We first remark that ℓ_2-RNSP implies ℓ_1-RNSP and ℓ_p-RNSP for $p \leq 2$ in the forms

$$\|\mathbf{v}_S\|_1 \leq \rho\|\mathbf{v}_{\overline{S}}\|_1 + \tau s^{1-1/2}\|A\mathbf{v}\|, \tag{2.14}$$

$$\|\mathbf{v}_S\|_p \leq \frac{\rho}{s^{1-1/p}}\|\mathbf{v}_{\overline{S}}\|_1 + \tau s^{1/p-1/2}\|A\mathbf{v}\|, \tag{2.15}$$

for all $\mathbf{v} \in \mathbb{R}^N$ and $S \subset [N]$ with $\mathrm{card}(S) \leq s$. To see this, we deduce the inequality $\|\mathbf{v}_S\|_p \leq s^{1/p-1/2}\|\mathbf{v}_S\|_2$ from (A.3) in Appendix A in [18]. Then we get (2.15) by applying the condition for ℓ_2-RNSP to this inequality,

$$\begin{aligned}\|\mathbf{v}_S\|_p &\leq s^{1/p-1/2}\|\mathbf{v}_S\|_2 \\ &\leq s^{1/p-1/2}\left(\frac{\rho}{s^{1-1/2}}\|\mathbf{v}_{\overline{S}}\|_1 + \tau\|A\mathbf{v}\|\right) \\ &= \frac{\rho}{s^{1-1/p}}\|\mathbf{v}_{\overline{S}}\|_1 + \tau s^{1/p-1/2}\|A\mathbf{v}\|.\end{aligned}$$

Inserting $p = 1$ gives (2.14).

In view of (2.14), applying Theorem 2.3.10 with $\mathbf{z} = \mathbf{x}^\#$ and S chosen as the index set of the s largest (in absolute value) entries of $\mathbf{x}$, we get

$$\|\mathbf{x}^\# - \mathbf{x}\|_1 \leq \frac{1+\rho}{1-\rho}(\|\mathbf{x}^\#\|_1 - \|\mathbf{x}\|_1 + 2\sigma_s(\mathbf{x})_1) + \frac{2\tau}{1-\rho}s^{1-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\|. \tag{2.16}$$

Next we choose S as the index set of the s largest (in absolute value) entries of $(\mathbf{x}^\# - \mathbf{x})$, and separate $(\mathbf{x}^\# - \mathbf{x})$ into $(\mathbf{x}^\# - \mathbf{x})_S$ and $(\mathbf{x}^\# - \mathbf{x})_{\overline{S}}$. Then, by using the triangle inequality and Theorem 2.5 in [18], we get

$$\begin{aligned}\|\mathbf{x}^\# - \mathbf{x}\|_p &\leq \|(\mathbf{x}^\# - \mathbf{x})_{\overline{S}}\|_p + \|(\mathbf{x}^\# - \mathbf{x})_S\|_p \\ &= \sigma_s(\mathbf{x}^\# - \mathbf{x})_p + \|(\mathbf{x}^\# - \mathbf{x})_S\|_p \\ &\leq \frac{1}{s^{1-1/p}}\|\mathbf{x}^\# - \mathbf{x}\|_1 + \|(\mathbf{x}^\# - \mathbf{x})_S\|_p.\end{aligned}$$

Applying (2.15) to this and using $\|(\mathbf{x}^\# - \mathbf{x})_{\overline{S}}\|_1 \leq \|\mathbf{x}^\# - \mathbf{x}\|_1$ results in

$$\begin{aligned}\|\mathbf{x}^\# - \mathbf{x}\|_p &\leq \frac{1}{s^{1-1/p}}\|\mathbf{x}^\# - \mathbf{x}\|_1 + \frac{\rho}{s^{1-1/p}}\|(\mathbf{x}^\# - \mathbf{x})_{\overline{S}}\|_1 + \tau s^{1/p-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\| \\ &\leq \frac{1}{s^{1-1/p}}\|\mathbf{x}^\# - \mathbf{x}\|_1 + \frac{\rho}{s^{1-1/p}}\|\mathbf{x}^\# - \mathbf{x}\|_1 + \tau s^{1/p-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\| \\ &\leq \frac{1+\rho}{s^{1-1/p}}\|\mathbf{x}^\# - \mathbf{x}\|_1 + \tau s^{1/p-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\|.\end{aligned}$$

Finally, substituting (2.16) into this yields

$$\begin{aligned}\|\mathbf{x}^\# - \mathbf{x}\|_p &\leq \frac{1+\rho}{s^{1-1/p}}\left(\frac{1+\rho}{1-\rho}(\|\mathbf{x}^\#\|_1 - \|\mathbf{x}\|_1 + 2\sigma_s(\mathbf{x})_1) + \frac{2\tau}{1-\rho}s^{1-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\|\right) \\ &\quad + \tau s^{1/p-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\| \\ &= \frac{(1+\rho)^2}{(1-\rho)s^{1-1/p}}(\|\mathbf{x}^\#\|_1 - \|\mathbf{x}\|_1 - 2\sigma_s(\mathbf{x})_1) \\ &\quad + \frac{(1+\rho)\cdot 2\tau s^{1-1/2}}{(1-\rho)s^{1-1/p}}\|A(\mathbf{x}^\# - \mathbf{x})\| + \tau s^{1/p-1/2}\|A(\mathbf{x}^\# - \mathbf{x})\| \\ &= \frac{(1+\rho)^2}{(1-\rho)s^{1-1/p}}(\|\mathbf{x}^\#\|_1 - \|\mathbf{x}\|_1 - 2\sigma_s(\mathbf{x})_1) \\ &\quad + \left(\frac{1+\rho}{s^{1-1/p}}\cdot\frac{2\tau s^{1-1/2}}{1-\rho} + \tau s^{1/p-1/2}\right)\|A(\mathbf{x}^\# - \mathbf{x})\| \\ &= \frac{(1+\rho)^2}{(1-\rho)s^{1-1/p}}(\|\mathbf{x}^\#\|_1 - \|\mathbf{x}\|_1 - 2\sigma_s(\mathbf{x})_1) + \frac{(3+\rho)\tau}{(1-\rho)}s^{1/p-1/2}\eta.\end{aligned}$$

In the final equation we have used that $\|A(\mathbf{x}^\# - \mathbf{x})\| \leq \eta$ and that

$$\begin{aligned}&\frac{1+\rho}{s^{1-1/p}} \cdot \frac{2\tau s^{1-1/2}}{1-\rho} + \tau s^{1/p-1/2} \\ &= \frac{(1+\rho)2\tau s^{1/p-1/2} + (1-\rho)\tau s^{1/p-1/2}}{(1-\rho)} \\ &= \frac{(3+\rho)\tau}{(1-\rho)} s^{1/p-1/2}.\end{aligned}$$

Thus, with $C \coloneqq (1+\rho)^2/(1-\rho)$ and $D \coloneqq (3+\rho)\tau/(1-\rho)$, we have reached the desired result. ■

2.4 Coherence

Although the null space properties give nice guarantees on the recovery of sparse vectors via basis pursuit, they can be difficult to verify. The *coherence* of a matrix is a much simpler measure of the suitability of the measurement matrix. There are different notions of the coherence in the literature. We will define two of the notions here. The first notion, which we will continue to call the coherence, will be used throughout the remainder of the thesis. It is defined in Definition 2.4.1. The second notion will only be used in this section. We will refer to it as the *column coherence* to avoid confusion, and it is defined in Definition 2.4.2. The rest of this section will establish some results for the column coherence. Results based on the first notion of coherence can be found in the literature.

Generally speaking, regardless of the specific definition of the coherence, we say that the smaller the coherence is, the better the recovery is.

In the following, we work with the assumption that the columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$ of the matrix A are always ℓ_2-normalized, i.e., $\|\mathbf{a}_i\|_2 = 1$ for all $i \in [N]$. We start by giving the formal definitions of the coherence and the column coherence.

Definition 2.4.1. [20, Definition 2.1] Let $U \in \mathbb{R}^{N\times N}$ be an isometry with elements $u_{i,j}$. The coherence $\mu = \mu(U)$ of the matrix U is defined as

$$\mu \coloneqq \max_{1\leq i,j\leq N} |u_{i,j}|^2.$$

Definition 2.4.2. [18, Definition 5.1] Let $A \in \mathbb{R}^{m\times N}$ be a matrix with ℓ_2-normalized columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$. The *column coherence* $\mu_c = \mu_c(A)$ of the matrix A is defined as

$$\mu_c \coloneqq \max_{1\leq i\neq j\leq N} |\langle \mathbf{a}_i, \mathbf{a}_j\rangle|. \tag{2.17}$$

A generalization of the column coherence is the so-called ℓ_1-column coherence function, defined below.

Definition 2.4.3. [18, Definition 5.2] Let $A \in \mathbb{R}^{m\times N}$ be a matrix with ℓ_2-normalized columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$. The *$\ell_1$-column coherence function* μ_1 of the matrix A is defined for $s \in [N-1]$ by

$$\mu_1(s) \coloneqq \max_{i\in[N]} \max \left\{ \sum_{j\in S} |\langle \mathbf{a}_i, \mathbf{a}_j\rangle|, S \subset [N], \operatorname{card}(S) = s, i \notin S \right\}. \tag{2.18}$$

For $s = 1$, the ℓ_1-column coherence function corresponds to the usual column coherence.

Bounds on the Column Coherence

As mentioned briefly, small coherence generally means that the matrix is well-conditioned for recovery. Therefore, it is of interest to know the bounds on the coherence. We quickly note that for the coherence in Definition 2.4.1, we have $1/N \leq \mu(U) \leq 1$. For the column coherence in Definition 2.4.2, we can easily see by the Cauchy-Schwartz inequality and the assumption that the columns are ℓ_2-normalized that the column coherence μ_c is bounded above by 1,

$$\mu_c = \max_{1 \leq i \neq j \leq N} |\langle \mathbf{a}_i, \mathbf{a}_j \rangle| \leq \|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2 = 1.$$

Since the column coherence is the greatest absolute value of the inner product between columns, it cannot be less than 0. For a rectangular matrix $A \in \mathbb{R}^{m \times N}$ with $m \geq N$, $\mu_c = 0$ if and only if the columns of A form an orthonormal system. This means that for a square matrix, we have $\mu_c = 0$ if and only if A is a unitary matrix. If we have $m < N$, which is the case in compressed sensing, we cannot have $\mu_c = 0$ because some of the columns must be linearly dependent. We will look into the limitations on how small the column coherence can be in this case, but first we need to define some terms.

Definition 2.4.4. [18, Definition 5.5] A system of ℓ_2-normalized vectors $(\mathbf{a}_1, \ldots, \mathbf{a}_N)$ in $\mathbb{R}^m$ is called *equiangular* if there is a constant $c \geq 0$ such that

$$|\langle \mathbf{a}_i, \mathbf{a}_j \rangle| = c \quad \text{for all } i, j \in [N], i \neq j.$$

Definition 2.4.5. [18, Definition 5.6] A system of vectors $(\mathbf{a}_1, \ldots, \mathbf{a}_N)$ in $\mathbb{R}^m$ is called a *tight frame* if there exists a constant $\lambda > 0$ such that one of the following equivalent conditions holds:

a) $\|\mathbf{x}\|_2^2 = \lambda \sum_{j=1}^N |\langle \mathbf{x}, \mathbf{a}_j \rangle|^2 \quad$ for all $\mathbf{x} \in \mathbb{R}^m$,

b) $\mathbf{x} = \lambda \sum_{j=1}^N \langle \mathbf{x}, \mathbf{a}_j \rangle \mathbf{a}_j \quad$ for all $\mathbf{x} \in \mathbb{R}^m$,

c) $AA^* = \frac{1}{\lambda} I_m$, where A is the matrix with columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and I_m is the $(m \times m)$ identity matrix.

The proof of the equivalence of the conditions is left out. A system of ℓ_2-normalized columns is called an equiangular tight frame if it is both equiangular and a tight frame. The following theorem states the lower bound on the column coherence, which is known as the *Welch bound*.

Theorem 2.4.6. *[18, Theorem 5.7] The column coherence of a matrix $A \in \mathbb{R}^{m \times N}$ with ℓ_2-normalized columns satisfies*

$$\mu_c \geq \sqrt{\frac{N - m}{m(N - 1)}}. \tag{2.19}$$

Equality holds if and only if the columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$ of the matrix A form an equiangular tight frame.

We can extend the Welch bound to the ℓ_1-column coherence function for small values of its argument s.

Theorem 2.4.7. *[18, Theorem 5.8] The ℓ_1-column coherence of a matrix $A \in \mathbb{R}^{m\times N}$ with ℓ_2-normalized columns satisfies*

$$\mu_1(s) \geq s\sqrt{\frac{N-m}{m(N-1)}} \quad \text{whenever} \quad s < \sqrt{N-1}. \tag{2.20}$$

Equality holds if and only if the columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$ of the matrix A form an equiangular tight frame.

Since we in compressed sensing are interested in both having a small column coherence and $(m \times N)$-matrices with m much smaller than N, it is impossible to meet the Welch bound. To have small column coherence, the number of measurements m will scale quadratically with the sparsity s, which we will see later. Therefore, m could potentially become quite large.

The next theorem shows that the number of vectors N in an equiangular tight frame, i.e., the number of columns in A required to meet the Welch bound, cannot be arbitrarily large.

Theorem 2.4.8. *[18, Theorem 5.10] The cardinality N of an equiangular system $(\mathbf{a}_1, \ldots, \mathbf{a}_N)$ of ℓ_2-normalized vectors in $\mathbb{R}^m$ satisfies*

$$N \leq \frac{m(m+1)}{2}.$$

If equality is achieved, then the system $(\mathbf{a}_1, \ldots, \mathbf{a}_N)$ is also a tight frame.

If we were working in the complex space, N would be less than or equal to m^2 [18, p. 117].

Analysis of Basis Pursuit

We will now show that a small column coherence guarantees the success of basis pursuit.

Theorem 2.4.9. *[18, Theorem 5.15] Let $A \in \mathbb{R}^{m\times N}$ be a matrix with ℓ_2-normalized columns. If*

$$\mu_1(s) + \mu_1(s-1) < 1, \tag{2.21}$$

then every s-sparse vector $\mathbf{x} \in \mathbb{R}^N$ is exactly recovered from the measurement vector $\mathbf{y} = A\mathbf{x}$ via basis pursuit.

Proof. By Theorem 2.3.2, it is necessary and sufficient to prove that A satisfies the null space property of order s, i.e., that

$$\|\mathbf{v}_S\|_1 < \|\mathbf{v}_{\overline{S}}\|_1 \tag{2.22}$$

for any non-zero vector $\mathbf{v} \in \ker A$ and any index set $S \subset [N]$ with $\operatorname{card}(S) = s$. Let $\mathbf{a}_1, \ldots, \mathbf{a}_N$ be the columns of A. The condition $\mathbf{v} \in \ker A$, i.e., $A\mathbf{v} = \mathbf{0}$ can be written as $\sum_{j=1}^{N} v_j \mathbf{a}_j = \mathbf{0}$.

Taking the inner product of this sum with some $\mathbf{a}_i$, $i \in S$ gives

$$\langle \sum_{j=1}^{N} v_j \mathbf{a}_j, \mathbf{a}_i \rangle = \sum_{j=1}^{N} v_j \langle \mathbf{a}_j, \mathbf{a}_i \rangle = 0.$$

We isolate the term v_i,

$$v_i \langle \mathbf{a}_i, \mathbf{a}_i \rangle + \sum_{j=1, j\neq i}^{N} v_j \langle \mathbf{a}_j, \mathbf{a}_i \rangle = 0$$

$$v_i = - \sum_{j=1, j\neq i}^{N} v_j \langle \mathbf{a}_j, \mathbf{a}_i \rangle.$$

In the last equation have used that the columns are ℓ_2-normalized, i.e., $\langle \mathbf{a}_i, \mathbf{a}_i \rangle = 1$. We split the sum in the last equation into indices belonging to S and $\overline{S}$, and get

$$v_i = - \sum_{\ell \in \overline{S}} v_\ell \langle \mathbf{a}_\ell, \mathbf{a}_i \rangle - \sum_{j \in S, j \neq i} v_j \langle \mathbf{a}_j, \mathbf{a}_i \rangle.$$

By taking the absolute value on both sides and using the triangle inequality, we get

$$|v_i| \leq \sum_{\ell \in \overline{S}} |v_\ell| |\langle \mathbf{a}_\ell, \mathbf{a}_i \rangle| + \sum_{j \in S, j \neq i} |v_j| |\langle \mathbf{a}_j, \mathbf{a}_i \rangle|.$$

We take the sum over all $i \in S$ on both sides and interchange the summations,

$$\begin{aligned} \|\mathbf{v}_S\|_1 = \sum_{i \in S} |v_i| &\leq \sum_{\ell \in \overline{S}} |v_\ell| \sum_{i \in S} |\langle \mathbf{a}_\ell, \mathbf{a}_i \rangle| + \sum_{j \in S} |v_j| \sum_{i \in S, i \neq j} |\langle \mathbf{a}_j, \mathbf{a}_i \rangle| \\ &\leq \sum_{\ell \in \overline{S}} |v_\ell| \mu_1(s) + \sum_{j \in S} |v_j| \mu_1(s-1) \\ &= \mu_1(s) \|\mathbf{v}_{\overline{S}}\|_1 + \mu_1(s-1) \|\mathbf{v}_S\|_1. \end{aligned}$$

We combine the $\|\mathbf{v}_S\|_1$-terms on one side of the inequality, and get

$$(1 - \mu_1(s-1)) \|\mathbf{v}_S\|_1 \leq \mu_1(s) \|\mathbf{v}_{\overline{S}}\|_1. \tag{2.23}$$

Because of (2.21), we have $1 - \mu_1(s-1) > \mu_1(s)$. But then (2.23) cannot hold unless $\|\mathbf{v}_S\|_1 < \|\mathbf{v}_{\overline{S}}\|_1$, i.e., the null space property holds. ■

The Quadratic Bottleneck

If we choose a matrix $A \in \mathbb{R}^{m \times N}$ with small column coherence $\mu_c \leq d\sqrt{m}$ for some constant $d > 0$, we need

$$m \geq C s^2$$

measurements to satisfy (2.21), i.e., to ensure exact recovery of s-sparse vectors via basis pursuit.

It is straightforward to see that $\mu_c \leq \mu_1(s) \leq s\,\mu_c$ for $1 \leq s \leq N-1$. We refer to Appendix A.2 for a derivation. Using this, we see that the condition (2.21) is satisfied when $(2s-1)\mu_c < 1$,

$$\mu_1(s) + \mu_1(s-1) \leq s\mu_c + (s-1)\mu_c = (2s-1)\mu_c < 1.$$

Then, inserting $\mu_c = d/\sqrt{m}$, we get

$$\begin{aligned}(2s-1)\frac{d}{\sqrt{m}} &< 1\\ (2s-1)^2 d^2 &< m\\ s^2(4d^2) - 2sd^2 + d^2 &< m,\end{aligned}$$

where $s^2(4d^2)$ is the dominating term. The left-hand side is thus essentially on the form Cs^2.

Unfortunately, an estimate of the number of required measurements m where the sparsity s enters quadratically is often way too large, and thus the column coherence is not commonly used in practice.

2.5 The Restricted Isometry Property

The measures we have discussed so far for determining suitability of the matrix A are not typically used in practice. The null space properties are difficult to verify and the column coherence gives a rather pessimistic estimate on the required number of measurements. The *restricted isometry property (RIP)* is more commonly used, because large classes of random matrices are known to satisfy this property with high probability. We later show that the RIP implies the RNSP, which further implies that the RIP with high probability yields uniform recovery.

Definition 2.5.1. [2, Definition 5.15] Let $1 \leq s \leq N$. The s-th *restricted isometry constant* (RIC) δ_s of $A \in \mathbb{R}^{m\times N}$ is the smallest $\delta \geq 0$ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \tag{2.24}$$

for all s-sparse vectors $\mathbf{x} \in \mathbb{R}^N$. If $0 < \delta_s < 1$, then A is said to have the *restricted isometry property (RIP)* of order s.

In essence, if δ_s is small, then A is well-suited. In the next theorem we see that there exists a bound on the $2s$-th RIC δ_{2s} such that the matrix A satisfies the RNSP.

Theorem 2.5.2. *[18, Theorem 6.13] If the $2s$-th RIC of $A \in \mathbb{R}^{m\times N}$ obeys*

$$\delta_{2s} < \frac{4}{\sqrt{41}}, \tag{2.25}$$

then the matrix A satisfies the ℓ_2-robust null space property of order s with constants $0 < \rho < 1$ and $\tau > 0$ depending only on δ_{2s}.

The proof of this theorem is rather extensive and we will therefore only include a sketch of the proof here. For the full proof, interested readers are referred to [18, pp. 144–147].

Sketch of proof for Theorem 2.5.2. The goal is to find expressions for ρ and τ depending only on δ_{2s} such that we have

$$\|\mathbf{v}_S\|_2 \leq \frac{\rho}{\sqrt{s}}\|\mathbf{v}_{\overline{S}}\|_1 + \tau\|A\mathbf{v}\|_2 \quad \text{for all } \mathbf{v} \in \mathbb{R}^N \tag{2.26}$$

and $0 < \rho < 1$ and $\tau > 0$.

Given a $\mathbf{v} \in \mathbb{R}^N$, it is sufficient to consider the set $S \coloneqq S_0$ of s largest absolute entries of $\mathbf{v}$. We partition the complement of this set as $\overline{S_0} = \bigcup_{i \geq 1} S_i$, where S_1 is the index set of the s largest absolute entries of $\mathbf{v}$ in $\overline{S_0}$, and S_2 is the index set of the s largest absolute entries of $\mathbf{v}$ in $\overline{S_0 \cup S_1}$ and so on.

Since $\mathbf{v}_{S_0}$ is s-sparse, we have

$$\|A\mathbf{v}_{S_0}\|_2^2 = (1+t)\|\mathbf{v}_{S_0}\|_2^2 \quad \text{with } |t| \leq \delta_s.$$

We will manipulate this expression to get an inequality on the form (2.26). We observe that

$$\begin{aligned} \|A\mathbf{v}_{S_0}\|_2^2 &= \langle A\mathbf{v}_{S_0}, A\mathbf{v}_{S_0} \rangle \\ &= \langle A\mathbf{v}_{S_0}, A(\mathbf{v} - \sum_{k \geq 1} \mathbf{v}_{S_k}) \rangle. \end{aligned} \tag{2.27}$$

Next we establish a bound on $|\langle A\mathbf{v}_{S_0}, A\mathbf{v}_{S_k} \rangle|$ for any $k \geq 1$:

$$|\langle A\mathbf{v}_{S_0}, A\mathbf{v}_{S_k} \rangle| \leq \sqrt{\delta_{2s}^2 - t^2} \|\mathbf{v}_{S_0}\|_2 \|\mathbf{v}_{S_k}\|_2.$$

We will use the bound above and properties of inner products to manipulate (2.27) to get

$$\|A\mathbf{v}_{S_0}\|_2^2 \leq \|\mathbf{v}_{S_0}\| \big(\sqrt{1+t} \|A\mathbf{v}\|_2 + \sqrt{\delta_{2s}^2 - t^2} \sum_{k \geq 1} \|\mathbf{v}_{S_k}\|_2 \big). \tag{2.28}$$

By using the square root lifting inequality (Lemma 6.14 in [18]), we can bound the sum $\sum_{k \geq 1} \|\mathbf{v}_{S_k}\|_2$ by an expression of $\|\mathbf{v}_{S_0}\|_1$ and $\|\mathbf{v}_{\overline{S_0}}\|_1$:

$$\sum_{k \geq 1} \|\mathbf{v}_{S_k}\|_2 \leq \frac{1}{\sqrt{s}} \|\mathbf{v}_{\overline{S_0}}\|_1 + \frac{1}{4} \|\mathbf{v}_{S_0}\|_1.$$

Substituting this into (2.28) and replacing $\|A\mathbf{v}_{S_0}\|_2^2$ with $(1+t)\|\mathbf{v}_S\|_2^2$, we get

$$(1+t)\|\mathbf{v}_{S_0}\|_2^2 \leq \|\mathbf{v}_{S_0}\|_2 \left(\sqrt{1+t} \|A\mathbf{v}\|_2 + \frac{\sqrt{\delta_{2s}^2 - t^2}}{\sqrt{s}} \|\mathbf{v}_{\overline{S_0}}\|_1 + \frac{\sqrt{\delta_{2s}^2 - t^2}}{4} \|\mathbf{v}_{S_0}\|_2 \right),$$

which we can solve for $\|\mathbf{v}_{S_0}\|_2$. Then we get an expression on the form (2.26), with

$$\rho \coloneqq \frac{\delta_{2s}}{\sqrt{1 - \delta_{2s}^2} - \delta_{2s}/4} \quad \text{and} \quad \tau \coloneqq \frac{\sqrt{1+\delta_{2s}}}{\sqrt{1 - \delta_{2s}^2} - \delta_{2s}/4}.$$

We have $0 < \rho < 1$ and $\tau > 0$ when $\delta_{2s} < 4/\sqrt{41}$. ■

The next theorem tells us that with a certain number of samples m, the matrix A with an appropriate scaling will satisfy the RIP with high probability.

Theorem 2.5.3. *[1, Theorem 2.5] Let $U \in \mathbb{R}^{N \times N}$ be an isometry, $\epsilon > 0$ and $\delta < 1$. Let $t_1, \ldots, t_m$ be chosen uniformly and independently from the set $\{1, \ldots, N\}$ and set $\Omega = \{t_1, \ldots, t_m\}$. If*

$$m \gtrsim \delta^{-2} \cdot s \cdot \mu(U) \cdot (\log(2m) \log(2N) \log^2(2s) + \log(1/\epsilon)),$$

then with probability at least $1 - \epsilon$, the matrix $A = \frac{1}{\sqrt{p}} P_\Omega U \in \mathbb{R}^{m \times N}$ with $p = \frac{m}{N}$ satisfies the RIP of order s with $\delta_s \leq \delta$.

We are now ready for our main result, which gives an estimate on how many samples are necessary to have a high probability of uniform recovery.

Theorem 2.5.4. *Let $U \in \mathbb{R}^{N\times N}$ be an isometry, $A = \frac{1}{\sqrt{p}} P_\Omega U \in \mathbb{R}^{m\times N}$ with $p = \frac{m}{N}$ and $\epsilon > 0$. Suppose that*

$$m \gtrsim s \cdot \mu(U) \cdot (\log(2m)\log(2N)\log^2(2s) + \log(1/\epsilon)). \tag{2.29}$$

Then with probability at least $1-\epsilon$, for any $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^m$ with $\|A\mathbf{x} - \mathbf{y}\|_2 \leq \eta$, a solution $\mathbf{x}^\# \in \mathbb{R}^N$ of

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \textit{subject to} \quad \|A\mathbf{z} - \mathbf{y}\|_2 \leq \eta$$

approximates the vector $\mathbf{x}$ with errors

$$\|\mathbf{x} - \mathbf{x}^\#\|_1 \leq C\sigma_s(\mathbf{x})_1 + D\sqrt{s}\eta, \tag{2.30}$$

$$\|\mathbf{x} - \mathbf{x}^\#\|_2 \leq \frac{C}{\sqrt{s}}\sigma_s(\mathbf{x})_1 + D\eta, \tag{2.31}$$

where $C, D > 0$ are universal constants.

Proof. Assume that (2.29) holds. Then by Theorem 2.5.3, the matrix $A = \frac{1}{\sqrt{p}} P_\Omega U \in \mathbb{R}^{m\times N}$ with $p = \frac{m}{N}$ satisfies the RIP of order $2s$ with δ_{2s} less than some constant $\delta < 1$. We can assume $\delta_{2s} < 4/\sqrt{41}$. This implies $\delta_{2s}^{-2} > 41/16$. Inserting this into the estimate in Theorem 2.5.3, we get

$$m \gtrsim \frac{41}{16} \cdot 2s \cdot \mu(U) \cdot (\log(2m)\log(2N)\log^2(2s) + \log(1/\epsilon)), \tag{2.32}$$

which is on the form (2.29). By Theorem 2.5.2, A satisfies the ℓ_2-RNSP with constants $0 < \rho < 1$ and $\tau > 0$ depending only on δ_{2s}. Since $\delta_{2s} < 4/\sqrt{41}$ is fixed, the variables ρ and τ are also fixed. From Theorem 2.3.12, it follows that

$$\|\mathbf{x} - \mathbf{x}^\#\|_p \leq \frac{C}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + Ds^{1/p-1/2}\eta \tag{2.33}$$

for constants $C, D > 0$ depending only on ρ and τ. Since ρ and τ are fixed, C and D are universal. Setting $p = 1$ in (2.33) gives us (2.30) and inserting $p = 2$ gives (2.31). ■

2.6 Other Optimization Problems

So far, all of our theory and analysis has been based on basis pursuit. For CS to be applicable, we must be able to solve basis pursuit efficiently. In Chapter 5 we will go into detail about the SPGL1 algorithm which does this. The algorithm takes advantage of a few other formulations of the ℓ_1-minimization as well. In this section we therefore introduce three alternative formulations and establish a connection between the three.

The different formulations have varying names throughout the literature. First, we have the problem which is called *quadratically constrained basis pursuit*

(QCBP) in [18] and the basis pursuit denoise problem (BPDN) in [10]. In this text we will refer to it as QCBP. The problem is formulated as

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \text{subject to} \quad \|A\mathbf{z}-\mathbf{y}\|_2 \leq \eta, \tag{QCBP}$$

for some $\eta \geq 0$. It takes into account that the measurement vector $\mathbf{y}$ might not be exactly equal to $A\mathbf{x}$.

Next, we have what we will call the *unconstrained LASSO*,

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \lambda\|\mathbf{z}\|_1 + \|A\mathbf{z}-\mathbf{y}\|_2^2, \tag{U-LASSO}$$

for some $\lambda \geq 0$. This problem is referred to as basis pursuit denoising in [18] and the quadratically penalized least-squares problem in [10].

The final related version is the *constrained LASSO*,

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|A\mathbf{z}-\mathbf{y}\|_2 \quad \text{subject to} \quad \|\mathbf{z}\|_1 \leq \tau, \tag{C-LASSO}$$

for some $\tau \geq 0$. This problem is called LASSO in both [18] and [10], but we will refer to it as the constrained LASSO problem here to differentiate from the unconstrained LASSO.

The links between these minimization problems are given in the following proposition.

Proposition 2.6.1. *[18, Proposition 3.2]*

a) If $\mathbf{x}$ *is a minimizer of* (U-LASSO) *with* $\lambda > 0$*, then there exists* $\eta = \eta_{\mathbf{x}} \geq 0$ *such that* $\mathbf{x}$ *is a minimizer of* (QCBP)*.*

b) If $\mathbf{x}$ *is a unique minimizer of* (QCBP) *with* $\eta \geq 0$*, then there exists* $\tau = \tau_{\mathbf{x}} \geq 0$ *such that* $\mathbf{x}$ *is a unique minimizer of* (C-LASSO)*.*

c) If $\mathbf{x}$ *is a minimizer of* (C-LASSO) *with* $\tau > 0$*, then there exists* $\lambda = \lambda_{\mathbf{x}} \geq 0$ *such that* $\mathbf{x}$ *is a minimizer of* (U-LASSO)*.*

Proof.

a) Assume $\mathbf{x}$ is a minimizer of (U-LASSO) with $\lambda > 0$. We set $\eta \coloneqq \|A\mathbf{x}-\mathbf{y}\|_2$. Consider $\mathbf{z} \in \mathbb{R}^N$ such that $\|A\mathbf{z}-\mathbf{y}\|_2 \leq \eta$, i.e., $\|A\mathbf{z}-\mathbf{y}\|_2 \leq \|A\mathbf{x}-\mathbf{y}\|_2$. Then, since $\mathbf{x}$ is a minimizer of (U-LASSO),

$$\lambda\|\mathbf{x}\|_1 + \|A\mathbf{x}-\mathbf{y}\|_2^2 \leq \lambda\|\mathbf{z}\|_1 + \|A\mathbf{z}-\mathbf{y}\|_2^2 \leq \lambda\|\mathbf{z}\|_1 + \|A\mathbf{x}-\mathbf{y}\|_2^2.$$

If we simplify this, we get that $\|\mathbf{x}\|_1 \leq \|\mathbf{z}\|_1$. Since $\mathbf{z}$ satisfies the constraint of (QCBP), $\mathbf{x}$ is a minimizer of (QCBP) with $\eta \coloneqq \|A\mathbf{x}-\mathbf{y}\|_2$.

b) Assume $\mathbf{x}$ is a unique minimizer of (QCBP) with $\eta \geq 0$, and set $\tau \coloneqq \|\mathbf{x}\|_1$. We consider $\mathbf{z} \in \mathbb{R}^N$, $\mathbf{z} \neq \mathbf{x}$, such that $\|\mathbf{z}\|_1 \leq \tau$, i.e., $\|\mathbf{z}\|_1 \leq \|\mathbf{x}\|_1$. Since $\mathbf{x}$ is a unique minimizer of (QCBP), $\mathbf{z}$ cannot satisfy the constraint of (QCBP), otherwise $\mathbf{z}$ would be the minimizer. Thus,

$$\|A\mathbf{z}-\mathbf{y}\|_2 \geq \eta \geq \|A\mathbf{x}-\mathbf{y}\|_2.$$

Since $\|A\mathbf{x}-\mathbf{y}\|_2 \leq \|A\mathbf{z}-\mathbf{y}\|_2$ and $\mathbf{z}$ satisfies the constraint for (C-LASSO) and is not equal to $\mathbf{x}$, we have that $\mathbf{x}$ is a unique minimizer of (C-LASSO).

c) To prove this part, we will use convex analysis. We start by noting that the (C-LASSO) problem is equivalent to

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|A\mathbf{z}-\mathbf{y}\|_2^2 \quad \text{subject to} \quad \|\mathbf{z}\|_1 \leq \tau. \tag{2.34}$$

The Lagrangian function for (2.34) is given by

$$L(\mathbf{z},\xi) = \|A\mathbf{z}-\mathbf{y}\|_2^2 + \xi(\|\mathbf{z}\|_1 - \tau),$$

where ξ is the Lagrangian multiplier. We assume $\mathbf{x}$ is a minimizer of (C-LASSO), and thus also of (2.34). Then by Theorem 4.2.1, strong duality holds for (C-LASSO). Strong duality implies that the primal-dual optimal point $(\mathbf{x},\xi^\#)$ is a saddle-point, i.e., that $L(\mathbf{x},\xi^\#) \leq L(\mathbf{z},\xi^\#)$ for all $\mathbf{z}\in\mathbb{R}^N$ (see [18, p. 562]). Hence, $\mathbf{x}$ is also a minimizer for the Lagrange function $L(\mathbf{z},\xi^\#) = \|A\mathbf{z}-\mathbf{y}\|_2^2 + \xi^\#\|\mathbf{z}\|_1 - \tau\xi^\#$. The constant term $-\tau\xi^\#$ does not affect the minimizer. Thus $\mathbf{x}$ is a minimizer for

$$\|A\mathbf{z}-\mathbf{y}\|_2^2 + \xi^\#\|\mathbf{z}\|_1,$$

which is (U-LASSO) with $\lambda = \xi^\#$.

■

2.7 Setting up Compressed Sensing

We have been working with the equation $A\mathbf{x}=\mathbf{y}$, where $\mathbf{x}\in\mathbb{R}^N$ is the sparse signal, $A\in\mathbb{R}^{m\times N}$ with $m<N$ describes the measurement process, and $\mathbf{y}\in\mathbb{R}^m$ is the measured data.

In reality, signals are not actually sparse in the domain in which they naturally occur. Let us consider that our signal is an image. If the image were sparse in the domain it naturally occurs, then most of the coefficients would be zero, and we would perceive the image as mostly black. Fortunately, most natural signals *are* sparse after an appropriate change of basis. Our sparse $\mathbf{x}$ can then be seen as $\mathbf{x}=\Psi\mathbf{w}$, where $\mathbf{w}$ is the original, non-sparse signal, and Ψ is the sparsifying transform. It turns out that *wavelets* have a sparsifying effect on most natural signals. They are therefore commonly used when working with image compression. We will use the discrete wavelet transform (DWT) for Ψ in this text.

The matrix A can be decomposed into several parts. Let $A = P_\Omega\Phi\Psi^{-1}$. The matrix P_Ω selects which m rows to sample from $\Phi\Psi^{-1}$. The matrix Φ is the model for the sampling pattern. In Chapter 5 we will discuss how we should design P_Ω and Φ using multilevel sampling. We wish to choose Φ and Ψ in such a way that the resulting matrix A has low coherence and thereby guarantees the success of basis pursuit. We want Φ to be incoherent with the DWT. It can be seen that the Paley and sequency ordered Hadamard transform is incoherent with the Daubechies wavelets [5, Theorem 6.3]. Therefore, we have chosen to use the Hadamard transform as Φ in this text.

Chosing the Hadamard transform has other benefits as well, besides being incoherent with wavelets. The matrix for the Hadamard transform consists of only values +1 and -1, so computing a matrix-vector multiplication can be done

using only subtraction and addition, and thus reduces the cost from $O(N^2)$ to $O(N \log N)$. The Hadamard matrix can also be computed using operators, so that there is no need to store the explicit matrix. This is useful when N is very large.

The next chapter will give a short introduction to some basic concepts of wavelets and the Hadamard transform.

CHAPTER 3

Wavelets and the Walsh-Hadamard Transform

In Section 2.7 we mentioned that we can use the discrete wavelet transform (DWT) and the Walsh-Hadamard transform (WHT) to fulfill the need for an incoherent matrix A. Both of these transforms perform a change of coordinates. The DWT is a change from a full resolution representation of the signal to a representation in lower resolution. The WHT decomposes a signal into basis functions called the Walsh functions and is useful in modeling the binary sampling we need in compressed sensing. Next we shall introduce some basic concepts for each of these transforms.

3.1 Wavelets

Wavelet theory is based on representing a signal with a basis of functions, *wavelets*, that is intentionally chosen to have specific properties. There are many different choices of bases. We adapt our choice so that we can better approximate differentiable signals or provide sparse representations of the signals. In this thesis we will only consider *orthonormal* wavelets, due to our focus on imaging applications and subsampled unitary linear systems. Such wavelets include the Haar wavelet (Daubechies 1), higher order Daubechies wavelets [6, 15, 16] or symlets [21, Section 7.2.3]. To make this introduction brief, the theory in this section is restricted to the simplest wavelet, the Haar wavelet.

Wavelets are a so-called *multiresolution model*, because they make it easy to switch been different levels of detail. We can project the signal onto nested *resolution spaces*, where each consecutive resolution space can include finer details than the previous one. In other words, the resolution spaces allow us to find arbitrarily good approximations to continuous functions. However, again for simplicity, we will restrict the discussion to only one level.

Definition 3.1.1. [25, Definition 1, p. 143] Let $N \in \mathbb{N}$. Then the space V_0 of functions defined on $[0, N)$ that are constant on each subinterval $[n, n+1)$ for $n = 0, \ldots, N$ is called the *resolution space*.

The following lemma tells us the basis for the space V_0.

Lemma 3.1.2. *[25, Lemma 2, pp. 143–144] Let $\phi(t)$ be defined by*

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

The space V_0 has dimension N, and the N functions $\phi(t-0), \phi(t-1), \ldots, \phi(t-(N-1))$ form an orthonormal basis for V_0, with respect to the inner product

$$\langle f, g \rangle = \int_0^N f(t)g(t)\,dt. \tag{3.2}$$

Thus, any $f \in V_0$ can be represented as

$$f = \sum_{n=0}^{N-1} c_n \phi(t-n) \tag{3.3}$$

for suitable coefficients $c_0, c_1, \ldots, c_{N-1}$.

The proof can be seen in [25, p. 144]. The previously mentioned resolution spaces are defined below.

Definition 3.1.3. [25, Definition 3, p. 145] The *refined resolution space* V_m is defined as the space of functions defined on the interval $[0, N)$ that are constant on each subinterval $[n/2^m, (n+1)/2^m)$ for $n = 0, \ldots, 2^m N - 1$.

As m increases, we can represent finer and finer details.

We find a basis for the spaces V_m similar to the basis for V_0.

Lemma 3.1.4. *[25, Lemma 4, p. 145] The dimension of V_m is $2^m N$, and the functions*

$$\phi_{m,n}(t) = 2^{m/2}\phi(2^m t - n), \quad \text{for } n = 0, \ldots, 2^m N - 1 \tag{3.4}$$

form an orthonormal basis for V_m. We will denote this basis by $\boldsymbol{\phi}_m$. Thus, any function $f \in V_m$ can be represented uniquely as

$$f = \sum_{n=0}^{2^m N - 1} c_{m,n}\phi_{m,n}(t) \tag{3.5}$$

for appropriate coefficients $c_{m,n}$.

Again the proof has been left out, but it can be seen in [25, pp. 145–146].

As mentioned, the resolution spaces are nested, i.e., $V_0 \subset V_1 \subset \cdots \subset V_m \subset \cdots$. We provide an explanation for $V_0 \subset V_1$ here. All the basis vectors $\phi_{0,n}$ for V_0 have the value 1 on the subintervals $[n, n+1)$ and 0 elsewhere. These subintervals can be split into two equal halves, $[2n/2, (2n+1)/2)$ and $[(2n+1)/2, (2n+2)/2)$. The basis vectors $\phi_{1,2n}$ for V_1 have the value $\sqrt{2}$ on $[2n/2, (2n+1)/2)$ and 0 elsewhere. The basis vectors $\phi_{1,2n+1}$ have the value $\sqrt{2}$ on $[(2n+1)/2, (2n+2)/2)$ and 0 elsewhere. Therefore, we can write $\phi_{0,n}$ as a linear combination of the basis vectors for V_1,

$$\phi_{0,n} = \frac{1}{\sqrt{2}}\phi_{1,2n} + \frac{1}{\sqrt{2}}\phi_{1,2n+1}.$$

This means that all the basis vectors of $\boldsymbol{\phi}_0$ are in V_1, and thus $V_0 \subset V_1$. This idea can easily be generalized. For a formal proof of this, see [25, p. 148].

If we project V_m onto V_{m-1}, we get a low-resolution approximation. The details that are left out when we replace V_m with this approximation, also called the error, is contained in the corresponding *detail space*.

Definition 3.1.5. [25, Definition 8, p. 148] The orthogonal complement of V_m projected onto V_{m-1} is denoted W_{m-1}. All the spaces W_k are called *detail spaces*.

For a $g_m \in V_m$, we can write $g_m = g_{m-1} + e_{m-1}$, where $g_{m-1} \in V_{m-1}$ and $e_{m-1} \in W_{m-1}$. Since V_{m-1} and W_{m-1} are mutually orthogonal spaces, they are linearly independent. For linearly independent spaces U and V, we can form a new space by taking the *direct sum* of these spaces, denoted by $U \oplus V$. The new vector space consists of vectors on the form $\mathbf{u} + \mathbf{v}$, where $\mathbf{u} \in U$ and $\mathbf{v} \in V$. Thus, we can write $V_m = V_{m-1} \oplus W_{m+1}$. For wavelets with one level, i.e., $m = 1$, we write $V_1 = V_0 \oplus W_0$.

The following definition gives the basis functions for the detail spaces.

Definition 3.1.6. [25, Definition 9, p. 149] We define

$$\psi(t) = \frac{1}{\sqrt{2}}\phi_{1,0}(t) - \frac{1}{\sqrt{2}}\phi_{1,1}(t) = \phi(2t) - \phi(2t-1) \tag{3.6}$$

and

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n) \quad \text{for } n = 0, 1, \ldots, 2^m N - 1. \tag{3.7}$$

It can be shown that $\psi_{m,n}$ for $n = 0, 1, \ldots, 2^m N - 1$ are orthonormal for any m and are thus a basis for the details spaces, see [25, pp. 149–151]. We will denote this basis as $\boldsymbol{\psi}_m$. The function $\psi(t)$ for the Haar wavelet is defined as

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1/2; \\ -1, & \text{if } 1/2 \leq t < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

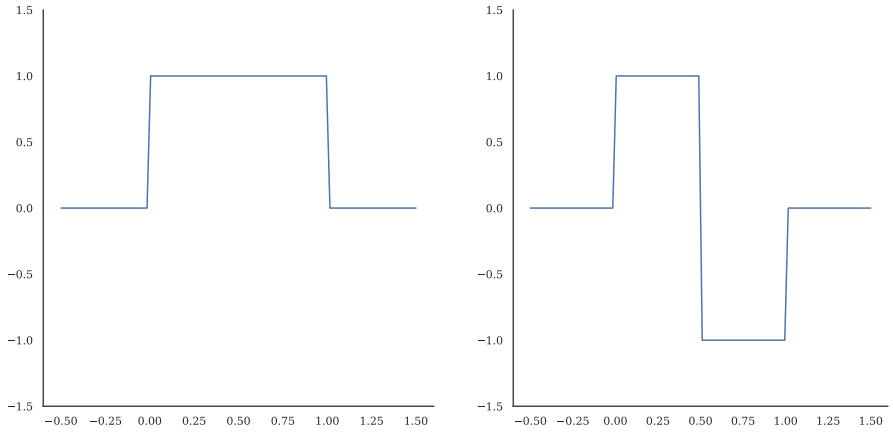
Figure 3.1 shows the functions ϕ and ψ for the Haar wavelet.

The function in (3.8) is an example of a *mother wavelet*. We can choose to use other functions with similar properties and denote them by ψ. This would give a different mother wavelet. The higher order Daubechies wavelets are other popular mother wavelets.

Discrete Wavelet Transform

We now have the necessary terminology to define the discrete wavelet transform (DWT).

Definition 3.1.7. [25, Definition 13, p. 152] The *discrete wavelet transform* is defined as the change of coordinates from $\boldsymbol{\phi}_1$ to $\boldsymbol{\phi}_0, \boldsymbol{\psi}_0$. More generally, the m-level DWT is defined as the change of coordinates from $\boldsymbol{\phi}_m$ to $(\boldsymbol{\phi}_0, \boldsymbol{\psi}_0, \ldots, \boldsymbol{\psi}_{m-1})$. The m-level *inverse discrete wavelet transform (IDWT)* is the change of coordinates in the opposite direction.



$\phi(t)$ $\psi(t)$

Figure 3.1: The functions ϕ and ψ for the Haar wavelet.

For the rest of this section, we are going to use the following notation. Let $\mathcal{B}$ and $\mathcal{C}$ be bases for a vector space V. Then the matrix $P_{\mathcal{C}\leftarrow\mathcal{B}}$ is the change-of-coordinates matrix from $\mathcal{B}$ to $\mathcal{C}$ such that

$$[\mathbf{x}]_{\mathcal{C}} = P_{\mathcal{C}\leftarrow\mathcal{B}}[\mathbf{x}]_{\mathcal{B}},$$

where $[\mathbf{x}]_{\mathcal{B}}$ is the vector $\mathbf{x}$ in $\mathcal{B}$-coordinates and $[\mathbf{x}]_{\mathcal{C}}$ is $\mathbf{x}$ in $\mathcal{C}$-coordinates.

If we let

$$\mathcal{C}_m = \{\phi_{m-1,0}, \psi_{m-1,0}, \phi_{m-1,1}, \psi_{m-1,1}, \ldots, \phi_{m-1,2^{m-1}N-1}, \psi_{m-1,2^{m-1}N-1}\}$$

be the basis for V_m with the basis vectors from $\boldsymbol{\phi}_{m-1}$ and $\boldsymbol{\psi}_{m-1}$ put in alternating order, then we can make the following definition.

Definition 3.1.8. [25, Definition 11, p. 151] The matrices

$$H = P_{\mathcal{C}_m\leftarrow\boldsymbol{\phi}_m} \quad \text{and} \quad G = P_{\boldsymbol{\phi}_m\leftarrow\mathcal{C}_m}$$

are called *kernel transformations*.

Using the kernel transformations, we get the expressions

$$\text{DWT} = P_{(\boldsymbol{\phi}_0,\boldsymbol{\psi}_0)\leftarrow\boldsymbol{\phi}_1} = P_{(\boldsymbol{\phi}_0,\boldsymbol{\psi}_0)\leftarrow\mathcal{C}_1}P_{\mathcal{C}_1\leftarrow\boldsymbol{\phi}_1} = P_{(\boldsymbol{\phi}_0,\boldsymbol{\psi}_0)\leftarrow\mathcal{C}_1}H$$

and

$$\text{IDWT} = GP_{\mathcal{C}_1\leftarrow(\boldsymbol{\phi}_0,\boldsymbol{\psi}_0)}$$

for the 1-level change of coordinates between $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_0, \boldsymbol{\psi}_0$.

It follows from the definitions of ϕ and ψ that the matrix H for the DWT in this case is the matrix where

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

is repeated along the main diagonal N times. The matrix G for the IDWT is equal to H. For example, for $m = 1$ and $N = 4$ the matrices become

$$H = G = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

3.2 The Walsh-Hadamard Transform

The *Walsh-Hadamard transform (WHT)*, sometimes called just the Walsh transform or Hadamard transform, is a nonsinusoidal, orthogonal transform. Like how the Fourier transform decomposes a signal into a set of sine and cosine functions, the WHT decomposes a signal into another set of basis functions— the *Walsh functions.*

Walsh Functions

Walsh functions [8] are an ordered, orthonormal set of rectangular waveforms, which take only two amplitude values, +1 and -1.

To get the values +1 and -1, the definitions of the different Walsh functions include an exponent which utilizes the binary representation of the input variables. We include a short recap of the binary representation here. For $n \in \mathbb{N}$ and $x \in [0, 1)$, let n_i be the i-th bit in the binary representation of n, with n_1 being the least significant bit, so that

$$n = n_1 2^0 + n 2_2^1 + \cdots + n_i 2^{i-1} + \cdots, \quad n_i \in \{0, 1\},$$

and let x_i be the i-th bit in the binary representation of x with x_1 being the most significant fractional bit, so that

$$x = x_1 2^{-1} + x_2 2^{-2} + \cdots + x_i 2^{-i} + \cdots, \quad x_i \in \{0, 1\}.$$

The Walsh functions are defined on a set time interval T and take the time period x as input. The second input variable n describes the ordering of the functions. The ordering is based on the number of sign changes, or zero crossings, in the function. If a function has the value +1 for half the time interval T, and then switches to the value -1 for the rest of T, the function would have one sign change. An intuitive ordering of the functions would be to arrange them in ascending order, with the function with the lowest number of sign changes being first. This is typically called the *sequency* ordering. We denote Walsh functions with this ordering as

$$\mathrm{WAL}(n, x).$$

The function $\mathrm{WAL}(0, x)$ would then be the sequency ordered Walsh function with zero sign changes over the time period x. Later in the section we describe two other orderings, the *ordinary* and *Paley* orderings, in connection with the Hadamard matrices.

Forward and Inverse Walsh-Hadamard Transform

For a 1D signal, the forward Walsh-Hadamard transform (FWHT) and the inverse Walsh-Hadamard transform (IWHT) are defined by

$$\text{FWHT:} \quad y_n = \frac{1}{N} \sum_{i=1}^{N} x_i \cdot \text{WAL}(n, i) \quad \text{for } n = 1, 2, \ldots, N \tag{3.9}$$

$$\text{IWHT:} \quad x_i = \sum_{n=1}^{N} y_n \cdot \text{WAL}(n, i) \quad \text{for } i = 1, 2, \ldots, N \tag{3.10}$$

where $\mathbf{x} = (x_1, \ldots, x_N)$ is the original signal and $\mathbf{y} = (y_1, \ldots, y_N)$ is the result of the FWHT and N is the signal length. The signal length has to be on the form 2^R for some integer $R > 0$. If $N \neq 2^R$, we can pad the signal with zeros. Note that the formulae (3.9) and (3.10) are for the sequency ordered FWHT and IWHT. By replacing $\text{WAL}(n, i)$ with another Walsh function, we would get a differently ordered Walsh-Hadamard transform.

The formulae can be interpreted as a change of coordinates between the Walsh basis and the standard basis for $\mathbb{R}^N$. Therefore, we can also write the FWHT and IWHT on matrix form,

$$\text{FWHT:} \quad \mathbf{y} = \frac{1}{N} H_N \mathbf{x} \tag{3.11}$$

$$\text{IWHT:} \quad \mathbf{x} = H_N^{-1} \mathbf{y} = H_N \mathbf{y} \tag{3.12}$$

where H_N is the matrix for the change of basis, known as the *Hadamard matrix*.

The Hadamard Matrix

The Hadamard matrix is an $(N \times N)$ orthogonal matrix with elements that take only the values $+1$ and -1. Matrices with these criteria are only defined for $N = 2^R$ for some integer $R > 0$.

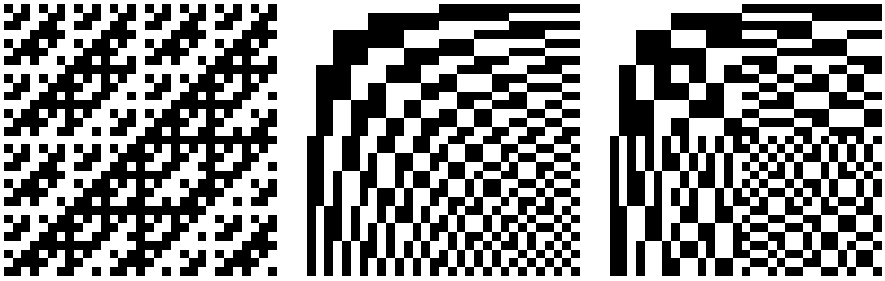
There is a fixed set of rows that will satisfy these criteria. We get different types of Hadamard matrices depending on how we order these rows. There are at least three different orderings, which are known as the *ordinary*, *sequency* and *Paley* orderings, the same orderings as for the Walsh functions. Figure 3.2 shows these orderings.

Ordinary Hadamard Matrix

The *ordinary* ordering gets its name from being the most commonly used when only discussing one ordering. It is also known as the *Hadamard* ordering or the natural ordering.

For $N = 2$, the Hadamard matrix with this ordering is

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$



Ordinary Sequency Paley

Figure 3.2: The ordinary, sequency and Paley ordered Hadamard matrices for $N = 32$. The +1's are represented by white and the -1's by black.

The (4×4) ordinary Hadamard matrix can be generated from H_2, in the following way:

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{3.13}$$

We can continue to generate bigger and bigger Hadamard matrices in this fashion, using the recursive definition

$$H_N = H_2 \otimes H_{N/2} = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \tag{3.14}$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.15}$$

where $\otimes$ is the Kronecker product. This simple definition is the reason the ordinary Hadamard matrix is most commonly used when only using one ordering.

If we label the rows in (3.13) with their number of sign changes, the first row would be labeled 0, the second row labeled 3, the third row labeled 1, and the fourth row labeled 2. The order of the rows in (3.13) is $\{0, 3, 1, 2\}$.

Sequency Ordered Hadamard Matrix

If we change the order of the rows in (3.13) so that they are in ascending order with respect to the number of sign changes, we get the *sequency* ordered Hadamard matrix. These matrices are sometimes called *Walsh matrices*. For $N = 4$, the order would then be $\{0, 1, 2, 3\}$ and the matrix is

$$S_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

The name S_N here was chosen to make the distinction between the different orderings of the Hadamard matrices more clear. The matrix S_N can be used

for H_N in the formulae (3.11) abd (3.12) to get the sequency ordered FWHT and IWHT.

The sequency ordered Hadamard matrices can be generated using Walsh functions.

Definition 3.2.1. [2, Definition 2.6] Let n be a positive integer and $x \in [0, 1)$. The Walsh function for sequency ordered Hadamard transform is

$$\mathrm{WAL}(n, x) \coloneqq (-1)^{\sum_{i=1}^{\infty}(n_i+n_{i+1})x_i}.$$

The value at index (i, j) in the matrix S_N is then given by $\mathrm{WAL}((i-1), (j-1)/2^R)$ for $i, j = 1, 2, \ldots, N$.

Paley Hadamard Matrix

The final of the three orderings is the *Paley* ordering, which is also called the *dyadic* ordering or the gray code ordering. Not surprisingly, this ordering is based on the gray code representation [17, 19] of the number of sign changes for each row. We find the gray code representation for each row: The numbers 0, 1, 2, 3 are 0, 1, 11, 10 in gray code, respectively. If we arrange these in ascending order, we get 0, 1, 10, 11. This corresponds to rearranging the rows in (3.13) in the order $\{0, 1, 3, 2\}$. Doing this, we get

$$D_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Again, the name D_N was chosen for clarity. The matrix D_N can be used in (3.11) and (3.12) to get the Paley ordered FWHT and IWHT.

The Paley ordered Hadamard matrices can be generated by using another Walsh function.

Definition 3.2.2. [2, Definition 2.5] Let n be a positive integer and $x \in [0, 1)$. The Walsh function for the Paley ordered Hadamard transform is

$$\mathrm{PAL}(n, x) \coloneqq (-1)^{\sum_{i=1}^{\infty} n_i x_i}.$$

Similar to the sequency ordered matrix, the value at index (i, j) in the matrix D_N is given by $PAL((i-1), (j-1)/2^R)$ for $i, j = 1, 2, \ldots, N$.

Figure 3.3 shows the first 16 Walsh functions with the Paley and sequency orderings.

3.3 Coherence Between Wavelets and the Hadamard Matrix

The reason for using the combination of the Hadamard and wavelet transforms is that we get a matrix A with small coherence. Recall that we are using Definition 2.4.1 for the coherence. We can find the coherence between these transforms by looking at the values in the matrix $A = |\Phi\Psi^{-1}|$, where Φ is the Hadamard matrix and Ψ is the wavelet matrix. Figure 3.4 shows the magnitudes

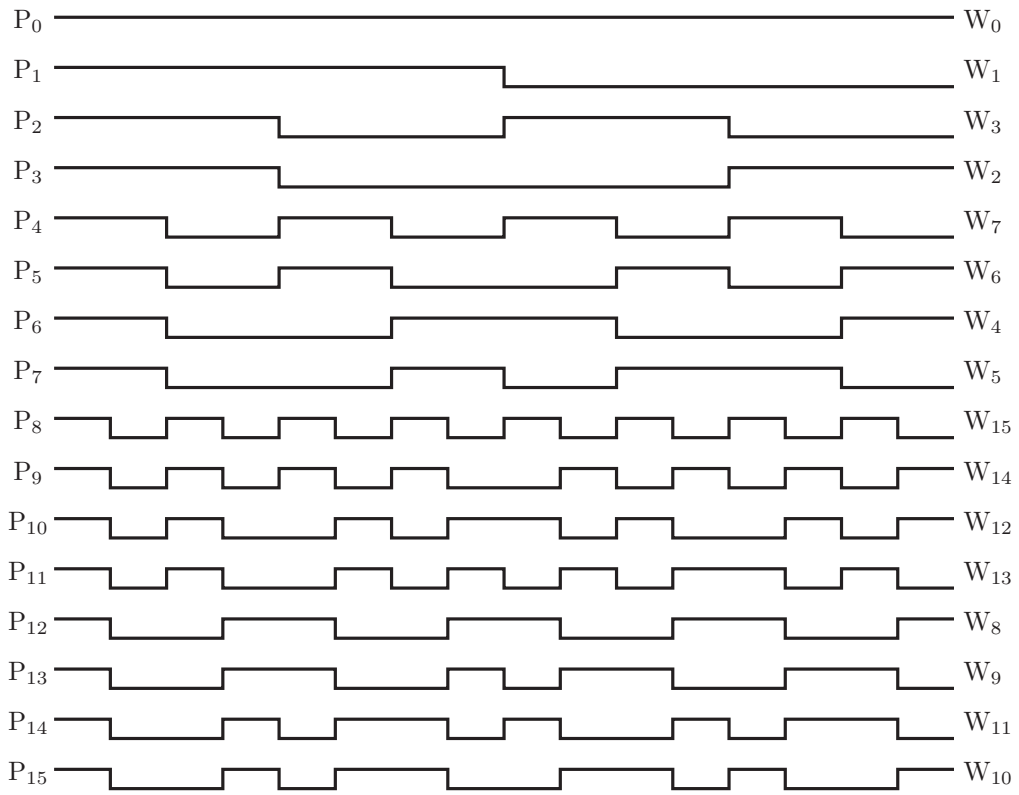


Figure 3.3: The first 16 Walsh functions with the Paley $\mathrm{P}_n \coloneqq \mathrm{PAL}(n, T)$ and sequency $\mathrm{W}_n \coloneqq \mathrm{WAL}(n, T)$ orderings.

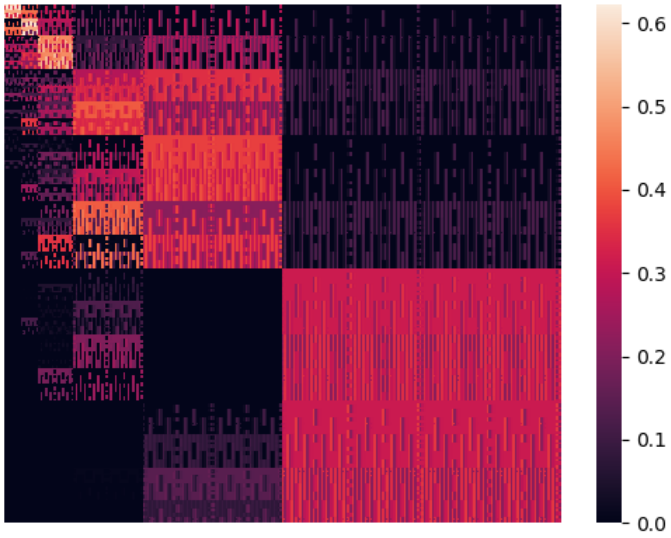
of the entries in this matrix for the Hadamard transform with the Paley ordering and the Haar and Daubechies 4 (DB4) wavelets.

We observe that both of these matrices have a clear diagonal tendency. For the Haar wavelet, we get a perfectly diagonal matrix, with the largest magnitudes in the top left corner. The magnitudes decrease asymptotically away from this corner. With the DB4 wavelet, there is a bit more noise, but it still follows the same pattern with the largest magnitudes in the top left corner.

In Chapter 5 we will discuss how to take advantage of this structure by sampling in levels. We should sample areas with large coherence more fully than areas with small coherence. From the matrices in Figure 3.4 we can infer that in the first sampling level, corresponding to the top left corners of the matrices, we should sample fully. For each consecutive level, as the coherence gets smaller, we can sample less and less. Because these matrices have large areas that are incoherent, most of our levels will have very few samples.

Figure 3.5 shows the success of a reconstruction via basis pursuit using Hadamard sampling and the Haar wavelet.

Haar



DB4

Figure 3.4: The matrix $|\Phi\Psi^{-1}|$, where Φ is the Hadamard matrix with the Paley ordering and Ψ is a wavelet matrix.

Original

Haar

DB4

Figure 3.5: Reconstruction of a 512×512 image using 20% Hadamard sampling and the Haar and DB4 wavelets.

CHAPTER 4

The SPGL1 Algorithm

The traditional CS theory relies on being able to efficiently solve the basis pursuit problem

$$\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}. \tag{BP}$$

The spectral projected gradient ℓ_1 (SPGL1) algorithm by Ewout van den Berg and Michael P. Friedlander solves this problem and two other related problems, the quadratically constrained basis pursuit (QCBP) and the constrained LASSO (C-LASSO).

The formulation in (BP) assumes perfect, non-noisy data. In practice, data is noisy, and we would like to relax the constraint in (BP) by incorporating an estimate of the noise level. Doing this obtains the QCBP problem

$$\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|A\mathbf{x} - \mathbf{b}\|_2 \leq \sigma, \tag{BP$_\sigma$}$$

where the parameter $\sigma \geq 0$ represents the noise level. The SPGL1 algorithm can solve (BP$_\sigma$) for any σ. We note that if $\sigma = 0$, the QCBP problem solves BP.

Many applications of compressed sensing involve large data sets and the matrix A may only be available as an operator, e.g., wavelets. One of the benefits of the SPGL1 algorithm is that it scales well to large problems and works for cases where the matrix is only available as an operator.

The idea behind the algorithm is to recast (BP$_\sigma$) as a problem of finding the root of a non-linear equation depending on a single variable τ, using a Newton-based root finding method. For each iteration of the algorithm, we use an approximation of τ to form a subproblem, the C-LASSO, which we solve using the spectral projected gradient (SPG). In this section we will present the necessary theory for this algorithm, fill in the details in the proofs from [10] and give pseudocode for key parts of the algorithm.

4.1 Approach

Our goal is to solve (BP$_\sigma$) for any $\sigma \geq 0$. As mentioned, we will do this by recasting (BP$_\sigma$) as a problem of finding the root of a non-linear equation $\phi(\tau) = \sigma$, depending on a single variable τ. For each iteration of the algorithm, we will use an estimate of τ to form the convex optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|A\mathbf{x} - \mathbf{b}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq \tau. \tag{LS$_\tau$}$$

Solving (LS_τ) will give us information about the derivative of ϕ, which we will need for the Newton based root finding method.

In Section 4.3 we will explain the relationship between the equation $\phi(\tau) = \sigma$ and (BP_σ). We will also show the differentiability of ϕ and find an expression for the derivative.

Since we are using an estimate of the variable τ, a solution of (LS_τ) only gives us an approximation to $\phi(\tau)$ and $\phi'(\tau)$. The usual convergence analysis for Newton's method does not apply in this case, and we therefore provide rate-of-convergence results for our case in Section 4.4.

Finally, the complexity of the algorithm depends on how efficiently we solve the subproblem. In Section 4.5 we give an algorithm based on the spectral projected gradient that solves (LS_τ) with worst-case complexity of $O(n \log n)$. Numerical experiments in [10] show that the cost is typically much smaller than the worst case.

Throughout the rest of this chapter we will make the following assumption without loss of generality, in order to simplify the discussion:

Assumption 4.1.1. The vector $\mathbf{b} \in \mathrm{range}(A)$, and $\mathbf{b} \neq \mathbf{0}$.

4.2 Convex Analysis

Since the SPGL1 algorithm relies heavily on convex optimization, we will begin by reviewing some elementary convex analysis. We follow the exposition in [14].

Convex Optimization

A mathematical optimization problem has the general form

$$\begin{aligned} \text{minimize} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq c_i \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = d_i \quad i = 1, \ldots, p. \end{aligned} \tag{4.1}$$

where f_0 is the objective function, $\mathbf{x}$ is the optimization variable, f_i are the inequality constraint functions and h_i the equality constraint functions. In general, optimization problems on this form are difficult to solve. However, there are families of special cases for which efficient solving algorithms and methods exist. A well known example of such a family is *convex optimization problems.*

In a convex optimization problem, the $f_0, \ldots, f_m$ are convex functions and $h_1, \ldots h_p$ affine functions. It can be shown that for a convex optimization problem, any locally optimal point is also globally optimal.

Convex Sets and Functions

A *convex set* is a set $C \subseteq \mathbb{R}^N$ where the line segment between any two points in C also lies in C, i.e., for all points $\mathbf{x}, \mathbf{y}$ in C and any β such that $0 \leq \beta \leq 1$ we have

$$\beta\mathbf{x} + (1 - \beta)\mathbf{y} \in C. \tag{4.2}$$

We call $\beta\mathbf{x} + (1 - \beta)\mathbf{y}$ a *convex combination* of $\mathbf{x}$ and $\mathbf{y}$. Thus, a set is convex if and only if it contains every convex combination of its points.

Let dom f denote the domain of a function f. A *convex function* is a function f where dom f is a convex set and the line segment between any two points $(\mathbf{x}, f(\mathbf{x}))$ and $(\mathbf{y}, f(\mathbf{y}))$ on the graph of f lies above the graph. Intuitively, this means the function "opens up". More formally, a function f is convex if for all $\mathbf{x}$ and $\mathbf{y}$ in the domain and β such that $0 \leq \beta \leq 1$,

$$f(\beta\mathbf{x} + (1-\beta)\mathbf{y}) \leq \beta f(\mathbf{x}) + (1-\beta)f(\mathbf{y}). \tag{4.3}$$

Similarly, a function f is *concave* if the line segments lie below the graph, or the inequality in (4.3) is the opposite way. If f is convex, then $-f$ is concave and vice versa.

If the inequality in (4.3) is a strict inequality, the function is *strictly convex*. The minimizer of a strictly convex function is unique [18, Proposition B.14].

An example of a convex function is any norm on $\mathbb{R}^N$. Assume that the function $f\colon \mathbb{R}^N \to \mathbb{R}$ is an arbitrary norm on $\mathbb{R}^N$ and that $0 \leq \beta \leq 1$. Let $\mathbf{x}$ and $\mathbf{y}$ be two arbitrary points in dom f. Then, first by the triangle inequality and then by the homogeneity of a norm, we have

$$f(\beta\mathbf{x} + (1-\beta)\mathbf{y}) \leq f(\beta\mathbf{x}) + f((1-\beta)\mathbf{y}) = \beta f(\mathbf{x}) + (1-\beta)f(\mathbf{y}).$$

Thus, f satisfies (4.3) and is convex.

Another useful example of a convex function is the *conjugate function*. The conjugate function $f^*\colon \mathbb{R}^N \to \mathbb{R}$ is defined as

$$f^*(\mathbf{y}) \coloneqq \sup_{\mathbf{x} \in \operatorname{dom} f} (\mathbf{y}^T\mathbf{x} - f(\mathbf{x})), \tag{4.4}$$

where the domain of f^* consists of $\mathbf{y} \in \mathbb{R}^N$ for which the supremum is finite.

The conjugate function is convex because it is the pointwise supremum of a family of convex functions of $\mathbf{y}$, and the pointwise supremum preserves convexity.

Lagrange Duality

Sometimes it can be useful to look at an optimization problem's *dual problem*. The solution to the dual problem can provide information about the solution to the original problem, or the *primal problem*. In certain cases, we can even find the exact solution to the primal problem by solving the dual problem. A useful property of the dual problem is that it is always convex, even when the primal problem is not.

The dual problem is based on the *Lagrange function*, which incorporates the constraints of the problem into the objective function through a weighted sum. For an optimization problem on the form (4.1), the Lagrange function is defined as

$$L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{y}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} y_i h_i(\mathbf{x}), \tag{4.5}$$

where λ_i are the Lagrangian multipliers for the inequality constraints, and y_i are the Lagrangian multipliers for the equality constraints. The variables for the dual problem become $\boldsymbol{\lambda}$ and $\mathbf{y}$.

The *Lagrange dual function*, often called just the *dual function*, is obtained by minimizing the Lagrange function over $\mathbf{x} \in \mathcal{D} \coloneqq \bigcap_{i=0}^{m} \operatorname{dom} f_i \cap \bigcap_{i=1}^{p} \operatorname{dom} h_i$,

$$\mathcal{L}(\boldsymbol{\lambda}, \mathbf{y}) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{y}). \tag{4.6}$$

For each dual pair $(\boldsymbol{\lambda}, \mathbf{y})$, the dual function gives a lower bound on the optimal value of the primal problem. If we denote the optimal primal value by p^*, we have $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{y}) \leq p^*$.

We can also use the dual function to determine the *best* lower bound on the optimal value by solving

$$\underset{\mathbf{y}}{\text{maximize}} \quad \mathcal{L}(\boldsymbol{\lambda}, \mathbf{y}) \quad \text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{4.7}$$

This is called the *Lagrange dual problem*, and solving the dual problem gives us information about the optimal primal problem. Under certain conditions, we can use the dual problem to find the exact solution to the primal problem.

Weak and Strong Duality

By definition, the dual optimal value d^* is the best lower bound on the primal optimal value p^*. This is called *weak duality* and can be written as the inequality

$$d^* \leq p^*.$$

The *duality gap* δ is the difference between the two optimal values:

$$\delta = p^* - d^* \geq 0.$$

If the duality gap is zero, i.e., $p^* = d^*$, we have *strong duality*. Then we can use the dual problem to solve the primal problem, or vice versa. The following theorem gives conditions on the optimization problem that imply strong duality holds.

Theorem 4.2.1. *[18, Theorem B.26] Assume that $f_0, f_1, \ldots, f_m$ are convex functions with $\operatorname{dom}(f_0) = \mathbb{R}^N$ and $h_1, \ldots, h_p$ are affine functions. If there exists $\mathbf{x} \in \mathbb{R}^N$ such that $f_\ell(\mathbf{x}) < c_\ell$ for all $\ell = 1, \ldots, m$ and $h_\ell(\mathbf{x}) = d_\ell$ for all $\ell = 1, \ldots, p$ then strong duality holds for the optimization problem* (4.1).

For a proof of this theorem, see Section 5.3.2 in [14].

4.3 The Pareto Curve

Keeping the convex analysis in mind, we are ready to study the SPGL1 algorithm. As previously stated, we are going to solve (BP_σ) by finding the roots of a non-linear, single-variable equation. The function we will be studying is given by

$$\phi(\tau) \coloneqq \|\mathbf{r}_\tau\|_2 \quad \text{with} \quad \mathbf{r}_\tau \coloneqq \mathbf{b} - A\mathbf{x}_\tau, \tag{4.8}$$

where $\mathbf{x}_\tau$ is the optimal solution to (LS_τ) for a given τ. We see then that $\phi(\tau)$ gives the optimal value of (LS_τ) for each $\tau \geq 0$.

The function in (4.8) is a parameterization in τ of a so-called *Pareto curve*. Pareto curves trace the trade-off between two conditions. Our Pareto curve will

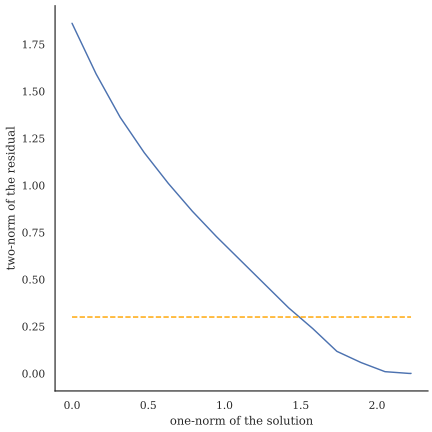


Figure 4.1: Example of the Pareto curve using a random matrix A and vector $\mathbf{b}$.

trace the trade-off between the one-norm of the solution $\mathbf{x}_\tau$ and the two-norm of its corresponding residual $\mathbf{r}_\tau$. Points on Pareto curves are *Pareto optimal*, meaning that we cannot make an improvement in one condition without lessening the other condition. Indeed, as $\|\mathbf{x}_\tau\|_1$ increases, $\|\mathbf{r}_\tau\|_2$ decreases, and vice versa. Figure 4.1 shows an example of the Pareto curve. The figure was generated by solving (LS_τ) with a random matrix A and vector $\mathbf{b}$, for 20 values of τ linearly spaced between 0 and 3. The function `spg_lasso` from [9] was used to solve (LS_τ). The output from `spg_lasso` includes the solution $\mathbf{x}_\tau$ and its residual $\mathbf{r}_\tau$. The one-norm of the solution ($\|\mathbf{x}_\tau\|_1$) was plotted against the two-norm of the corresponding residual ($\|\mathbf{r}_\tau\|_2$) for each value of τ.

Note that (BP_σ) is another parameterization of the Pareto curve, and we can therefore find points on the curve where the solutions to (LS_τ) and (BP_σ) coincide.

If we use Newton's method on the equation

$$\phi(\tau) = \sigma, \tag{4.9}$$

we get a sequence of parameters τ_k that converge to τ_σ, with $\|\mathbf{r}_{\tau_\sigma}\|_2 = \sigma$. We will see that τ_σ is the parameter for which the solutions to (LS_τ) and (BP_σ) coincide. First we note that the optimal solution of (LS_τ) for τ_σ, i.e., $\mathbf{x}_{\tau_\sigma}$, satisfies the constraint in (BP_σ), because $\|\mathbf{r}_{\tau_\sigma}\|_2 = \|A\mathbf{x}_{\tau_\sigma} - \mathbf{b}\|_2 = \sigma$. We see that $\mathbf{x}_{\tau_\sigma}$ must also be the optimal solution to (BP_σ), because we can only decrease $\|\mathbf{x}_{\tau_\sigma}\|_1$ further by increasing $\|\mathbf{r}_{\tau_\sigma}\|_2$. But increasing $\|\mathbf{r}_{\tau_\sigma}\|_2$ means that the constraint $\|A\mathbf{x}_{\tau_\sigma} - \mathbf{b}\|_2 \leq \sigma$ no longer holds. Thus finding a solution to (4.9) means finding an optimal solution to (BP_σ).

Using Newton's method to solve (4.9) means finding the point on the Pareto curve that intersects with a horizontal line with value σ. In Figure 4.1, we have chosen $\sigma = 0.3$ and it is represented by a dashed orange line.

Derivation of the Dual Variables

In order to use Newton's method on (4.9), we must be able to evaluate the derivative ϕ'. In the next section we will show that ϕ is differentiable and that $\phi' = -\lambda_\tau$, where $\lambda_\tau \geq 0$ is the unique dual solution of (LS_τ). In this section, we will see that the dual solution λ_τ is easily obtained as a by-product of solving (LS_τ).

To find an expression for λ_τ, we start by recasting the constrained Lasso problem (LS_τ) as the equivalent problem

$$\underset{\mathbf{r},\mathbf{x}}{\text{minimize}} \quad \|\mathbf{r}\|_2 \quad \text{subject to} \quad A\mathbf{x} + \mathbf{r} = \mathbf{b}, \quad \|\mathbf{x}\|_1 \leq \tau. \tag{4.10}$$

By incorporating the constraints into the objective function, we get the associated Lagrangian

$$L(\lambda, \mathbf{y}) = \|\mathbf{r}\|_2 - \mathbf{y}^T(A\mathbf{x} + \mathbf{r} - \mathbf{b}) + \lambda(\|\mathbf{x}\|_1 - \tau), \tag{4.11}$$

where $\mathbf{y}$ and λ are the Lagrangian multipliers. The Lagrangian dual function is then

$$\mathcal{L}(\lambda, \mathbf{y}) = \inf_{\mathbf{x},\mathbf{r}} \{L(\lambda, \mathbf{y})\}, \tag{4.12}$$

so that the dual of the convex problem (4.10) is given by

$$\max_{\mathbf{y},\lambda} \quad \mathcal{L}(\lambda, \mathbf{y}) \quad \text{subject to} \quad \lambda \geq 0. \tag{4.13}$$

By using the separability of the infimum in $\mathbf{r}$ and $\mathbf{x}$ and the definition of the conjugate function, we can simplify the dual problem and find expressions for $\mathbf{y}$ and λ. We start by distributing the $\mathbf{y}^T$ and λ in (4.11), and then group the terms that are dependent on $\mathbf{r}$, dependent on $\mathbf{x}$, and independent of both. We also use the fact that $\mathbf{y}^T\mathbf{b} = \mathbf{b}^T\mathbf{y}$, since this is the inner product of two vectors, and that $-\sup\{f\} = \inf\{-f\}$ for an affine function f on $\mathbb{R}^N$. Thus,

$$\begin{aligned}
\mathcal{L}(\lambda, \mathbf{y}) &= \inf_{\mathbf{x},\mathbf{r}} \{\|\mathbf{r}\|_2 - \mathbf{y}^T A\mathbf{x} - \mathbf{y}^T\mathbf{r} + \mathbf{y}^T\mathbf{b} + \lambda\|\mathbf{x}\|_1 - \lambda\tau\} \\
&= \inf_{\mathbf{r},\mathbf{x}} \{\mathbf{b}^T\mathbf{y} - \lambda\tau\} + \inf_{\mathbf{r},\mathbf{x}} \{-\mathbf{y}^T\mathbf{r} + \|\mathbf{r}\|_2\} + \inf_{\mathbf{r},\mathbf{x}} \{-\mathbf{y}^T A\mathbf{x} + \lambda\|\mathbf{x}\|_1\} \\
&= \mathbf{b}^T\mathbf{y} - \tau\lambda - \sup_{\mathbf{r}} \{\mathbf{y}^T\mathbf{r} - \|\mathbf{r}\|_2\} - \sup_{\mathbf{x}} \{\mathbf{y}^T A\mathbf{x} - \lambda\|\mathbf{x}\|_1\}.
\end{aligned}$$

Using the general definition of the conjugate function (4.4), we get that the conjugate function of $f(\mathbf{x}) = \alpha\|\mathbf{x}\|$ for any $\alpha \geq 0$ and arbitrary norm $\|\cdot\|$ with dual norm $\|\cdot\|_*$, is given by

$$f_*(\mathbf{z}) \coloneqq \sup_{\mathbf{x}} \{\mathbf{z}^T\mathbf{x} - \alpha\|\mathbf{x}\|\} = \begin{cases} 0, & \text{if } \|\mathbf{z}\|_* \leq \alpha; \\ \infty, & \text{otherwise.} \end{cases} \tag{4.14}$$

From this we see that the suprema in $\mathcal{L}(\lambda, \mathbf{y})$ are conjugate functions of $\|\mathbf{r}\|_2$ and $\lambda\|\mathbf{x}\|_1$, respectively. The first supremum has $\alpha = 1$ and the norm is the self-dual two-norm. In the latter supremum, $\alpha = \lambda$ and the norm is the one-norm, with dual norm the infinity-norm.

It follows that (4.13) stays bounded if and only if the dual variables satisfy the constraints of the conjugate functions, namely $\|\mathbf{y}\|_2 \leq 1$ and $\|A^T\mathbf{y}\|_\infty \leq \lambda$.

Otherwise, the conjugate functions will evaluate to ∞, making $\mathcal{L}$ infinitely small, and thus infeasible. Therefore, we can remove the suprema from the dual problem and add the constraints, so that the dual of (4.10) is given by

$$\max_{\mathbf{y},\lambda} \quad \mathbf{b}^T\mathbf{y} - \tau\lambda \quad \text{subject to} \quad \|\mathbf{y}\|_2 \leq 1,\ \|A^T\mathbf{y}\|_\infty \leq \lambda. \tag{4.15}$$

Since $\|\cdot\|_\infty$ is a norm, the non-negative constraint on λ is still enforced by the second constraint in (4.15).

We are now ready to find the expressions for $\mathbf{y}$ and λ. We will see that they are expressions of the primal solution $\mathbf{r}$, with $\mathbf{y} = \mathbf{r}/\|\mathbf{r}\|_2$ and $\lambda = \|A^T\mathbf{y}\|_\infty$. The expression for $\mathbf{y}$ is found by noting that from (4.14), we have

$$\sup_{\mathbf{r}} \{\mathbf{y}^T\mathbf{r} - \|\mathbf{r}\|_2\} = 0 \quad \text{if} \quad \|\mathbf{y}\|_2 \leq 1. \tag{4.16}$$

We use the Cauchy-Schwartz inequality and the constraint in (4.16) to find

$$\mathbf{y}^T\mathbf{r} - \|\mathbf{r}\|_2 \leq \|\mathbf{y}\|_2\|\mathbf{r}\|_2 - \|\mathbf{r}\|_2 \leq \|\mathbf{r}\|_2 - \|\mathbf{r}\|_2 = 0.$$

Let $\mathbf{r}$ be fixed as the $\mathbf{r}$ that achieves the supremum. Then, since the supremum is equal to 0, we must also have $\mathbf{y}^T\mathbf{r} - \|\mathbf{r}\|_2 = 0$, and the inequalities above thus become equalities. If $\mathbf{r} = 0$, the choice of $\mathbf{y}$ is arbitrary. If $\mathbf{r} \neq 0$, then $\mathbf{y} = \mathbf{r}/\|\mathbf{r}\|_2$ would satisfy $\|\mathbf{y}\|_2\|\mathbf{r}\|_2 - \|\mathbf{r}\|_2 = 0$. Then $\mathbf{y}$ is a unit vector, so we have $\|\mathbf{y}\|_2 = 1$.

The expression for λ can be seen from the dual problem (4.15). Let $\mathbf{y}$ be fixed. To maximize $\mathbf{b}^T\mathbf{y} - \tau\lambda$, we must minimize $\tau\lambda$. If $\tau > 0$, λ must be at its lower bound, i.e. $\lambda = \|A^T\mathbf{y}\|_\infty$. If $\tau = 0$, then the choice of λ is arbitrary.

Since the dual variable $\mathbf{y}$ only depends on the primal variable $\mathbf{r}$, we can eliminate $\mathbf{y}$. Resulting are necessary and sufficient optimality conditions for the primal-dual solution $(\mathbf{r}_\tau, \mathbf{x}_\tau, \lambda_\tau)$ of the problem (4.10):

$$A\mathbf{x}_\tau + \mathbf{r}_\tau = \mathbf{b}, \quad \|\mathbf{x}_\tau\|_1 \leq \tau \quad \text{(primal feasibility)}; \tag{4.17}$$

$$\|A^T\mathbf{r}_\tau\|_\infty \leq \lambda_\tau\|\mathbf{r}_\tau\|_2 \quad \text{(dual feasibility)}; \tag{4.18}$$

$$\lambda_\tau(\|\mathbf{x}_\tau\|_1 - \tau) = 0 \quad \text{(complimentarity)}. \tag{4.19}$$

Differentiability of the Pareto Curve

In this section we will show some properties of the Pareto curve ϕ for points $\tau \in [0, \tau_{BP}]$, where τ_{BP} is the optimal objective value of the problem (BP). Note that because of our assumption $0 \neq \mathbf{b} \in \text{range}(A)$, we know that $A\mathbf{x} = \mathbf{b}$ has a solution, so that (BP) is feasible, and that $\tau_{BP} > 0$.

When $\tau = 0$, we have that $\|\mathbf{x}_\tau\|_1 = 0$, which means that $\mathbf{x}_\tau = 0$. Then $\mathbf{r}_\tau = \mathbf{b}$, and consequently $\phi(0) = \|\mathbf{b}\|_2$. That τ_{BP} is the optimal objective value of (BP) means that it is the smallest value of τ such that $\|A\mathbf{x}_\tau - \mathbf{b}\|_2 = \|\mathbf{r}_\tau\|_2 = 0$. In other words, it is the smallest value of τ such that (LS_τ) has a zero objective value, or that $\phi(\tau_{BP}) = 0$. Therefore, we have that the function values at the endpoints of our interval are

$$\phi(0) = \|\mathbf{b}\|_2 > 0 \quad \text{and} \quad \phi(\tau_{BP}) = 0. \tag{4.20}$$

The properties of ϕ we want to show are listed in the following theorem.

Theorem 4.3.1. *[10, Theorem 2.1]*

a) The function ϕ is convex and non-increasing.

b) For all $\tau \in (0, \tau_{BP})$, ϕ is continuously differentiable, $\phi' = -\lambda_\tau$, and the optimal dual variable $\lambda_\tau = \|A^T \mathbf{y}_\tau\|_\infty$, where $\mathbf{y}_\tau = \mathbf{r}_\tau / \|\mathbf{r}_\tau\|_2$.

c) For $\tau \in [0, \tau_{BP}]$, $\|x_\tau\| = \tau$, and ϕ is strictly decreasing.

Proof.

a) To prove that $\phi(\tau) = \|\mathbf{r}_\tau\|_2$ is convex, we must show that $\phi(\beta\tau_1 + (1-\beta)\tau_2) \leq \beta\phi(\tau_1) + (1-\beta)\phi(\tau_2)$ for any τ_1, τ_2 in $\operatorname{dom}\phi$ and all $\beta \in [0,1]$. In order to do this, we restate ϕ as

$$\phi(\tau) = \inf_{\mathbf{x}} f(\mathbf{x}, \tau), \tag{4.21}$$

where

$$f(\mathbf{x}, \tau) \coloneqq \|A\mathbf{x} - \mathbf{b}\|_2 + \psi_\tau(\mathbf{x}) \quad \text{and} \quad \psi_\tau(\mathbf{x}) \coloneqq \begin{cases} 0, & \text{if } \|\mathbf{x}\|_1 \leq \tau \\ \infty, & \text{otherwise.} \end{cases} \tag{4.22}$$

With $\alpha = \tau$ and $\|\cdot\|_* = \|\cdot\|_1$ in (4.14), we note that $\psi_\tau(\mathbf{x}) = \sup_{\mathbf{z}}\{\mathbf{x}^T\mathbf{z} - \tau\|\mathbf{z}\|_\infty\}$. Then $\psi_\tau(\mathbf{x})$ is a pointwise supremum of an affine function in $(\mathbf{x}, \tau)$, and thus convex. Since f then is the sum of two convex functions, f is also convex. Now, let τ_1, τ_2 be any non-negative scalars, and let $\mathbf{x}_1, \mathbf{x}_2$ be the corresponding minimizers of (4.21). For any $\beta \in [0,1]$,

$$\begin{aligned} \phi(\beta\tau_1 + (1-\beta)\tau_2) &= \inf_{\mathbf{x}} f(\mathbf{x}, \beta\tau_1 + (1-\beta)\tau_2) \\ &\leq f(\beta\mathbf{x}_1 + (1-\beta)\mathbf{x}_2, \beta\tau_1 + (1-\beta)\tau_2) \\ &\leq \beta f(\mathbf{x}_1, \tau_1) + (1-\beta) f(\mathbf{x}_2, \tau_2) \\ &= \beta\phi(\tau_1) + (1-\beta)\phi(\tau_2). \end{aligned}$$

In the first inequality, we have used that the infimum is less than or equal to $f(\mathbf{z})$ for any $\mathbf{z} \in \operatorname{dom}(f)$. Since f is convex, its domain is a convex set. We can therefore pick a point $\mathbf{z} = \beta\mathbf{x}_1 + (1-\beta)\mathbf{x}_2$. The second inequality follows from the convexity of f. The last equality shows that ϕ is convex in τ.

Furthermore, ϕ is nonincreasing, because the feasible set for (LS_τ) extends as τ increases. When the feasible set extends, the optimal value of (LS_τ) will stay the same or become smaller. But because $\phi(\tau)$ is the optimal value of (LS_τ) at τ, we have that ϕ decreases as τ increases.

b) We know from part (a) that ϕ is a convex, non-increasing function. We also know that $\tau \in (0, \tau_{BP})$ are points where ϕ is finite, since ϕ is finite at the endpoints of this interval. Then by Theorem 25.1 in [23], ϕ is differentiable at τ if and only if its subgradient at τ is unique. By Proposition 6.5.8a) in [11], if λ_τ is a geometric multiplier, we have that $-\lambda_\tau \in \partial\phi(\tau)$, i.e., $-\lambda_\tau$ is a subgradient of $\phi(\tau)$. We know that λ_τ is a Lagrangian multiplier for (4.10). Since (4.10) is a convex problem, we have by Proposition 6.1.2b)

in [11] that geometric and Lagrangian multipliers coincide. Ergo, $-\lambda_\tau$ is a subgradient of $\phi(\tau)$.

To prove the differentiability of ϕ, it suffices to show that λ_τ is unique. Since $\tau > 0$, the dual variable λ_τ is only optimal at its lower bound, as discussed in the previous section. Hence $\lambda_\tau = \|A^T \mathbf{y}_\tau\|_\infty$. By Proposition 2.6.1c), the solution $\mathbf{x}_\tau$ of (LS_τ) is also a solution to the strictly convex (U-LASSO). Then $\mathbf{x}_\tau$ is unique and thus the optimal value $\mathbf{r}_\tau = \mathbf{b} - A\mathbf{x}_\tau$ of (LS_τ) is unique. Since $\tau < \tau_{BP}$ and ϕ is nonincreasing, we have $\|\mathbf{r}_\tau\|_2 > 0$. We can then take $\mathbf{y}_\tau = \mathbf{r}_\tau / \|\mathbf{r}_\tau\|_2$. The uniqueness of $\mathbf{r}_\tau$ implies the uniqueness of $\mathbf{y}_\tau$, which again implies the uniqueness of λ_τ. Thus ϕ is differentiable with $\phi' = -\lambda_\tau$. Finally, since ϕ is both convex and differentiable, we have that its derivative is continuous.

c) Recall that $\mathbf{x}_\tau$ is the optimal solution of (LS_τ), i.e., $\|\mathbf{x}_\tau\|_1 \leq \tau$. For $\tau = 0$, the assertion $\|\mathbf{x}_\tau\|_1 = \tau$ holds by the definiteness of a norm. The assertion holds for $\tau = \tau_{BP}$ by the definition of τ_{BP}. What remains to show is that part (c) holds for the interior of the interval.

We note that $\phi(\tau) \equiv \|\mathbf{r}_\tau\|_2 > 0$ for all $\tau \in [0, \tau_{BP})$. Then $\mathbf{r}_\tau \neq 0$, which implies that $\mathbf{y}_\tau \neq 0$ and $\lambda_\tau \neq 0$. More precisely, $\lambda_\tau > 0$. Since $\phi'(\tau) = -\lambda_\tau$, we have $\phi'(\tau) < 0$ on $(0, \tau_{BP})$, which implies that ϕ is strictly decreasing on this interval. Since both $\mathbf{x}_\tau$ and λ_τ satisfy the complimentarity in (4.19), we must have that $\|\mathbf{x}_\tau\|_1 - \tau = 0$, or $\|\mathbf{x}_\tau\|_1 = \tau$.

■

4.4 Root Finding

We will generate a sequence of parameters τ_k that converge to τ_σ by applying the Newton iteration

$$\tau_{k+1} = \tau_k + \Delta\tau_k \quad \text{with} \quad \Delta\tau_k \coloneqq (\sigma - \phi(\tau_k))/\phi'(\tau_k) \tag{4.23}$$

to (4.9). Then the corresponding solutions $\mathbf{x}_{\tau_k}$ of (LS_τ) will converge to $\mathbf{x}_\sigma$, which is the solution to (BP_σ).

By Theorem 4.3.1, we have that ϕ is convex, strictly decreasing and continuously differentiable for $\sigma \in (0, \|\mathbf{b}\|_2)$. Then by Proposition 1.4.1 in [12], this convergence is superlinear for all initial values $\tau_0 \in (0, \tau_{BP})$. This analysis is based on standard Newton methods, where we solve the Newton equations

$$\phi'(\tau_k)\Delta\tau_k = \sigma - \phi(\tau_k) \tag{4.24}$$

exactly at each iteration. This involves solving (LS_τ) exactly at each iteration. For large numbers of unknowns, this can be very computationally expensive.

For systems of non-linear equations in general, *inexact Newton methods* would solve (4.24) only approximately, reducing the cost of the algorithm. The residual is a fraction of the right-hand side. Analysis of inexact Newton methods shows that they are convergent and that the rate of convergence depends on the fraction. For example, by Theorem 7.2 in [22], if the fraction tends to zero, the method will have a superlinear convergence rate.

Unfortunately, this analysis does not apply when the right-hand side in (4.24) is only known approximately, which is the case we are working with here.

We will show later that the inexact Newton method still converges when we do not know the exact function value of ϕ, although this convergence is sublinear. However, we can make this convergence as close to superlinear as we want, by increasing the accuracy with which we compute ϕ. As we will see in the following section, we can use the duality gap to bound this accuracy.

Bounds on Approximate Solutions

In this section we find a bound on the accuracy of the computed function value of ϕ, using the duality gap. We later use this bound to establish the rate-of-convergence guarantee for the Newton iteration.

The algorithm for solving (LS_τ) that we outline in Section 4.5 preserves the feasibility of the iterates at all iterations. That is, an approximate solution $\overline{\mathbf{x}}_\tau$ will satisfy the primal feasibility condition. Because $\overline{\mathbf{x}}_\tau$ is an approximate solution and therefore suboptimal, the norm of its residual $\overline{\mathbf{r}}_\tau \coloneqq \mathbf{b} - A\overline{\mathbf{x}}_\tau$ must be greater than or equal to the true minimum $\|\mathbf{r}_\tau\|_2$. Because $\tau < \tau_{BP}$, both norms are greater than 0. We thus have the following inequalities:

$$\|\overline{\mathbf{x}}_\tau\|_1 \leq \tau, \quad \text{and} \quad \|\overline{\mathbf{r}}_\tau\|_2 \geq \|\mathbf{r}_\tau\|_2 > 0. \tag{4.25}$$

Then we can construct the approximations to the dual variables that are dual feasible,

$$\overline{\mathbf{y}}_\tau \coloneqq \overline{\mathbf{r}}_\tau / \|\overline{\mathbf{r}}_\tau\|_2 \quad \text{and} \quad \overline{\lambda}_\tau \coloneqq \|A^T \overline{\mathbf{y}}_\tau\|_\infty.$$

By weak duality, the value of the dual problem at the dual feasible point $(\overline{\lambda}_\tau, \overline{\mathbf{y}}_\tau)$ is a lower bound on the primal optimal value $\|\mathbf{r}_\tau\|_2$. Because the primal problem is a minimization problem, the value of the primal problem at the primal feasible point $\overline{\mathbf{x}}_\tau$ is an upper bound on the primal optimal value. Hence,

$$\mathbf{b}^T \overline{\mathbf{y}}_\tau - \tau\overline{\lambda}_\tau \leq \|\mathbf{r}_\tau\|_2 \leq \|\overline{\mathbf{r}}_\tau\|_2. \tag{4.26}$$

Now we use the duality gap

$$\delta_\tau \coloneqq \|\overline{\mathbf{r}}_\tau\|_2 - (b^T \overline{\mathbf{y}}_\tau - \tau\overline{\lambda}_\tau) \geq 0, \tag{4.27}$$

to measure the quality of an approximate solution $\overline{\mathbf{x}}_\tau$.

Let $\overline{\phi}(\tau) \coloneqq \|\overline{\mathbf{r}}_\tau\|_2$ be the objective value of (LS_τ) at the approximate solution $\overline{\mathbf{x}}_\tau$. Then, using (4.26), we get

$$\overline{\phi}(\tau) - \phi(\tau) = \|\overline{\mathbf{r}}_\tau\|_2 - \|\mathbf{r}_\tau\|_2 \leq \|\overline{\mathbf{r}}_\tau\|_2 - (b^T \overline{\mathbf{y}}_\tau - \tau\overline{\lambda}_\tau) = \delta_\tau, \tag{4.28}$$

so that δ_τ is a bound on the accuracy of $\overline{\phi}(\tau)$.

According to [10, p. 9], with the added assumption that A has full rank and thus finite condition number, we can also use the duality gap to bound the difference between the derivatives $\overline{\phi'}(\tau)$ and $\phi'(\tau)$. Using the following definition of the condition number of A,

$$\mathrm{cond}(A) \coloneqq \frac{\sigma_{\max}}{\sigma_{\min}}, \tag{4.29}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values of A respectively, we have that $\mathrm{cond}(A)$ is bounded if A has full rank. The bound on the derivatives can be seen in (4.30):

$$\overline{\phi}(\tau) - \phi(\tau) < \delta_\tau \quad \text{and} \quad |\overline{\phi'}(\tau) - \phi'(\tau)| < \gamma\delta_\tau \tag{4.30}$$

where $\gamma > 0$ is independent of τ and proportional to the condition number of A. It is unclear from [10] why the second inequality holds and how it is connected to the condition number.

Rate-of-convergence Results

In this section we will show that the inexact Newton method converges even if we only know ϕ and ϕ' approximately. The main theorem in this section establishes the local convergence rate. Before we get to this theorem, we show the following useful lemma.

Lemma 4.4.1. *Suppose that A has full rank and $\sigma \in (0, \|\mathbf{b}\|_2)$. Then there exists positive constants γ_1 and γ_2 independent of τ_k such that*

$$\left| \frac{\phi(\tau_k) - \sigma}{\phi'(\tau_k)} - \frac{\overline{\phi}(\tau_k) - \sigma}{\overline{\phi'}(\tau_k)} \right| \leq \gamma_1 \delta_k \quad \textit{and} \quad |\phi'(\tau_k)^{-1}| < \gamma_2. \tag{4.31}$$

Proof. First we note that $\sigma \in (0, \|\mathbf{b}\|_2)$ implies that $\tau_k \in (0, \tau_{BP})$. Since $\phi(\tau_k) = \sigma$, we have that $\phi(\tau_k) \in (0, \|\mathbf{b}\|_2)$. Since $\phi(0) = \|\mathbf{b}\|_2 > 0$ and $\phi(\tau_{BP}) = 0$, and ϕ is non-increasing by Theorem 4.3.1a), we have $\tau_k \in (0, \tau_{BP})$.

We will show the second inequality first. Since A has full rank and is of dimension $(m \times N)$ with $m < N$, we have $\text{rank}(A) = m$. Then $\text{rank}(A^T) = m$, as well. By the Rank Theorem, the dimension of the null space of A^T is

$$\dim(\text{Null}(A^T)) = m - \text{rank}(A^T) = m - m = 0.$$

Thus, $A^T\mathbf{y} = 0$ only when $\mathbf{y} = 0$.

From Theorem 4.3.1b), we have that $\phi'(\tau_k) = -\|A^T\mathbf{y}_{\tau_k}\|_\infty$, where $\mathbf{y}_{\tau_k}$ is a unit vector. So $\mathbf{y}_{\tau_k}$ cannot be zero, and by the positive definiteness of a norm, $\|A^T\mathbf{y}_{\tau_k}\|_\infty \neq 0$. Then, and because of the non-negativity of a norm,

$$\gamma_3 \coloneqq \min_{\mathbf{y}:\|\mathbf{y}\|_2=1} \{\|A^T\mathbf{y}\|_\infty\} > 0.$$

This gives us

$$|\phi'(\tau_k)| = \|A^T\mathbf{y}_{\tau_k}\|_\infty > \gamma_3,$$

which yields $|\phi'(\tau_k)^{-1}| < \gamma_2$ with $\gamma_2 \coloneqq 1/\gamma_3$.

To prove the first inequality in (4.31), we introduce the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(u, v) = \frac{u - \sigma}{v}. \tag{4.32}$$

Taylor's formula in two variables for this function is expressed as

$$f(\overline{u}, \overline{v}) = f(u, v) + \nabla f(\mathbf{c})^T((\overline{u}, \overline{v}) - (u, v)), \tag{4.33}$$

where $\mathbf{c} = (c_1, c_2)$ lies on the line segment between (u, v) and $(\overline{u}, \overline{v})$, i.e.,

$$u \leq c_1 \leq \overline{u} \quad \text{and} \quad v \leq c_2 \leq \overline{v}. \tag{4.34}$$

We subtract $f(u, v)$ and take the absolute value on both sides in (4.33). Recognizing $\nabla f(\mathbf{c})^T((\overline{u}, \overline{v}) - (u, v))$ as the inner product between $\nabla f(\mathbf{c})$ and $((\overline{u}, \overline{v}) - (u, v))$, we can use Cauchy-Schwartz' inequality to obtain

$$|f(\overline{u}, \overline{v}) - f(u, v)| \leq \|\nabla f(\mathbf{c})\|_2 \|((\overline{u}, \overline{v}) - (u, v))\|_2.$$

Let $u = \phi(\tau_k)$, $v = \phi'(\tau_k)$, $\overline{u} = \overline{\phi}(\tau_k)$ and $\overline{v} = \overline{\phi'}(\tau_k)$. Making these substitutions and using the definition of f, we get

$$\left| \frac{\overline{\phi}(\tau_k) - \sigma}{\overline{\phi'}(\tau_k)} - \frac{\phi(\tau_k) - \sigma}{\phi'(\tau_k)} \right| \leq \|\nabla f(\mathbf{c})\|_2 \|([\overline{\phi}(\tau_k) - \phi(\tau_k)], [\overline{\phi'}(\tau_k) - \phi'(\tau_k)])\|_2,$$

where we recognize the left-hand side as the expression we are trying to restrict. The second norm on the right-hand side can be bounded by using the inequalities in (4.30),

$$\begin{aligned} \|([\overline{\phi}(\tau_k) - \phi(\tau_k)], [\overline{\phi'}(\tau_k) - \phi'(\tau_k)])\|_2 &= \sqrt{(\overline{\phi}(\tau_k) - \phi(\tau_k))^2 + (\overline{\phi'}(\tau_k) - \phi'(\tau_k))^2} \\ &< \sqrt{\delta_k^2 + \gamma^2 \delta_k^2} \\ &= \delta_k \sqrt{1 + \gamma^2}. \end{aligned}$$

To bound the first norm, $\|\nabla f(\mathbf{c})\|_2$, we compute the gradient of f and insert $\mathbf{c} = (c_1, c_2)$,

$$\nabla f(\mathbf{c}) = \left(\frac{1}{c_2}, \frac{\sigma - c_1}{c_2^2} \right). \tag{4.35}$$

From the second inequality in (4.31), we have $|v^{-1}| = |\phi'(\tau_k)^{-1}| < \gamma_2$. Then, because of (4.34) and that $u = \phi(\tau_k)$ is non-increasing and bounded below by 0, we have

$$\begin{aligned} \|\nabla f(\mathbf{c})\|_2 &= \sqrt{\frac{1}{(c_2)^2} + \frac{(\sigma - c_1)^2}{(c_2^2)^2}} \\ &\leq \sqrt{\frac{1}{v^2} + \frac{(\sigma - u)^2}{v^4}} \\ &\leq \sqrt{\gamma_2^2 + \sigma^2 \gamma_2^4} \\ &= \gamma_2 \sqrt{1 + \sigma^2 \gamma_2^2}. \end{aligned}$$

Combining everything, we get

$$\left| \frac{\phi(\tau_k) - \sigma}{\phi'(\tau_k)} - \frac{\overline{\phi}(\tau_k) - \sigma}{\overline{\phi'}(\tau_k)} \right| \leq \delta_k \gamma_1,$$

where $\gamma_1 \coloneqq \gamma_2 \sqrt{1 + \sigma^2 \gamma_2^2}\, \sqrt{1 + \gamma^2}$. ■

We are now ready to prove the main result, which states that the Newton iteration still converges when we only know ϕ and ϕ' approximately.

Theorem 4.4.2. *[10, Theorem 3.1] Suppose that A has full rank, $\sigma \in (0, \|b\|_2)$, and $\delta_k \coloneqq \delta_{\tau_k} \to 0$. Then if τ_0 is close enough to τ_σ, the iteration (4.23)–with ϕ and ϕ' replaced by $\overline{\phi}$ and $\overline{\phi'}$–generates a sequence $\tau_k \to \tau_\sigma$ that satisfies*

$$|\tau_{k+1} - \tau_\sigma| = \gamma \delta_k + \eta_k |\tau_k - \tau_\sigma|, \tag{4.36}$$

where $\eta_k \to 0$ and γ is a positive constant.

Proof. Similar to the proof of Lemma 4.4.1, $\sigma \in (0, \|b\|_2)$ implies that $\tau_\sigma \in (0, \tau_{BP})$. Then, by Theorem 4.3.1b), ϕ is continuously differentiable for all τ close enough to τ_σ. By the fundamental theorem of calculus, we have

$$\phi(\tau_k) - \sigma = \phi(\tau_k) - \phi(\tau_\sigma) = \int_{\tau_\sigma}^{\tau_k} \phi'(\tau)\, d\tau.$$

We will make the following substitution in the integral above. Let $\tau = \tau_\sigma + \alpha(\tau_k - \tau_\sigma)$. Then $d\tau = (\tau_k - \tau_\sigma)\, d\alpha$. If $\tau = \tau_\sigma$, then $\alpha = 0$. If $\tau = \tau_k$, then $\alpha = 1$. We get

$$\phi(\tau_k) - \sigma = \int_0^1 \phi'(\tau_\sigma + \alpha[\tau_k - \tau_\sigma])\, d\alpha \cdot (\tau_k - \tau_\sigma).$$

Next, we add and subtract $\phi'(\tau_k)(\tau_k - \tau_\sigma)$ on the right-hand side. By noting that $\int_0^1 \phi'(\tau_k)(\tau_k - \tau_\sigma)\, d\alpha = \phi'(\tau_k)(\tau_k - \tau_\sigma)$ and combining the integrals, we get

$$\begin{aligned}
\phi(\tau_k) - \sigma &= \phi'(\tau_k)(\tau_k - \tau_\sigma) - \phi'(\tau_k)(\tau_k - \tau_\sigma) \\
&\quad + \int_0^1 \phi'(\tau_\sigma + \alpha[\tau_k - \tau_\sigma])\, d\alpha \cdot (\tau_k - \tau_\sigma) \\
&= \phi'(\tau_k)(\tau_k - \tau_\sigma) + \int_0^1 [\phi'(\tau_\sigma + \alpha[\tau_k - \tau_\sigma]) - \phi'(\tau_k)] \cdot d\alpha(\tau_k - \tau_\sigma).
\end{aligned} \tag{4.36}$$

We solve the integral in (4.36),

$$\begin{aligned}
&\int_0^1 [\phi'(\tau_\sigma + \alpha[\tau_k - \tau_\sigma]) - \phi'(\tau_k)] \cdot d\alpha(\tau_k - \tau_\sigma) \\
&\quad = \left[\frac{\phi(\tau_\sigma + \alpha[\tau_k - \tau_\sigma])}{(\tau_k - \tau_\sigma)} - \phi'(\tau_k) \cdot \alpha \right]_0^1 (\tau_k - \tau_\sigma) \\
&\quad = \left(\frac{\phi(\tau_\sigma + [\tau_k - \tau_\sigma])}{(\tau_k - \tau_\sigma)} - \phi'(\tau_k) \cdot 1 - \frac{\phi(\tau_\sigma)}{(\tau_k - \tau_\sigma)} + \phi'(\tau_k) \cdot 0 \right) (\tau_k - \tau_\sigma) \\
&\quad = \left(\frac{\phi(\tau_k)}{(\tau_k - \tau_\sigma)} - \phi'(\tau_k) - \frac{\phi(\tau_\sigma)}{(\tau_k - \tau_\sigma)} \right) (\tau_k - \tau_\sigma) \\
&\quad = \phi(\tau_k) + \phi'(\tau_k)(\tau_\sigma - \tau_k) - \phi(\tau_\sigma) \\
&\quad = \omega(\tau_k, \tau_\sigma).
\end{aligned}$$

In the last equation, we have used that we can express $\phi(\tau_\sigma)$ as a version of the first order Taylor expansion around τ_k,

$$\phi(\tau_\sigma) = \phi(\tau_k) + (\tau_\sigma - \tau_k)\phi'(\tau_k) + \omega(\tau_k, \tau_\sigma),$$

with the remainder $\omega(\tau_k, \tau_\sigma)$ satisfying

$$\omega(\tau_k, \tau_\sigma)/|\tau_k - \tau_\sigma| \to 0 \quad \text{as} \quad |\tau_k - \tau_\sigma| \to 0. \tag{4.37}$$

Inserting $\omega(\tau_k, \tau_\sigma)$ for the integral in (4.36), we get $\phi(\tau_k) - \sigma = \phi'(\tau_k)(\tau_k - \tau_\sigma) + \omega(\tau_k, \tau_\sigma)$.

Next, because $\Delta\tau_k = (\sigma - \overline{\phi}(\tau_k))/\overline{\phi'}(\tau_k)$ from (4.23), we have

$$\begin{aligned}|\tau_{k+1} - \tau_\sigma| &= |\tau_k + \Delta\tau_k - \tau_\sigma| \\ &= \left|\frac{\sigma - \overline{\phi}(\tau_k)}{\overline{\phi'}(\tau_k)} + \tau_k - \tau_\sigma\right|.\end{aligned}$$

Solving $\phi(\tau_k) - \sigma = \phi'(\tau_k)(\tau_k - \tau_\sigma) + \omega(\tau_k, \tau_\sigma)$ for $(\tau_k - \tau_\sigma)$ and using Lemma 4.4.1, we get

$$\begin{aligned}|\tau_{k+1} - \tau_\sigma| &= \left|\frac{\sigma - \overline{\phi}(\tau_k)}{\overline{\phi'}(\tau_k)} + \frac{1}{\phi'(\tau_k)}(\phi(\tau_k) - \sigma - \omega(\tau_k, \tau_\sigma))\right| \\ &= \left|-\frac{(\overline{\phi}(\tau_k) - \sigma)}{\overline{\phi'}(\tau_k)} + \frac{\phi(\tau_k) - \sigma}{\phi'(\tau_k)} - \frac{\omega(\tau_k, \tau_\sigma)}{\phi'(\tau_k)}\right| \\ &\leq \left|\frac{\phi(\tau_k) - \sigma}{\phi'(\tau_k)} - \frac{(\overline{\phi}(\tau_k) - \sigma)}{\overline{\phi'}(\tau_k)}\right| + \left|\frac{\omega(\tau_k, \tau_\sigma)}{\phi'(\tau_k)}\right| \\ &\leq \gamma_1\delta_k + \gamma_2|\omega(\tau_k, \tau_\sigma)| \\ &= \gamma_1\delta_k + \eta_k|\tau_k - \tau_\sigma|,\end{aligned}$$

where $\eta_k \coloneqq \gamma_2|\omega(\tau_k, \tau_\sigma)|/|\tau_k - \tau_\sigma|$. Since $\gamma_2 > 0$, we have $\eta_k > 0$. With τ_k sufficiently close to τ_σ, (4.37) implies that $\eta_k \to 0$ and $\eta_k < 1$.

To show that the sequence of τ_k converges to τ_σ, we apply the inequality above recursively $\ell \geq 1$ times and get

$$|\tau_{k+\ell} - \tau_\sigma| \leq \gamma_1 \sum_{i=1}^{\ell} (\eta_k)^{\ell-i}\, \delta_{k+i-1} + (\eta_k)^\ell |\tau_k - \tau_\sigma|. \tag{4.38}$$

Because $\eta_k < 1$, we have that $(\eta_k)^\ell \to 0$ and $(\eta_k)^{\ell-i} \to 0$ as $\ell \to \infty$. We also have by assumption that $\delta_k \to 0$, so the whole right-hand side goes to zero as $\ell \to \infty$. Thus $\tau_k \to \tau_\sigma$. For a derivation of (4.38), refer to Appendix A.2. ■

We remark that the convergence rate of the algorithm depends on the rate at which δ_k approaches zero. If δ_k is exactly zero, which is the case if (LS_τ) is solved exactly at each iteration, then (4.36) becomes

$$\begin{aligned}|\tau_{k+1} - \tau_\sigma| &= \eta_k|\tau_k - \tau_\sigma| \\ \frac{|\tau_{k+1} - \tau_\sigma|}{|\tau_k - \tau_\sigma|} &= \eta_k.\end{aligned}$$

Taking the limit as $k \to \infty$ on both sides yields $|\tau_{k+1} - \tau_\sigma|/|\tau_k - \tau_\sigma| \to 0$. The convergence rate is superlinear. This agrees with the convergence analysis of a standard Newton iteration.

Theorem 4.4.2 assumes that A has full rank. If A is rank deficient, then we would have slow convergence unless $\delta_k = 0$. To see this, assume A is rank deficient. Then $\gamma_3 \coloneqq \min_{\mathbf{y}:\|\mathbf{y}\|_2=1}\{\|A^T\mathbf{y}_{\tau_k}\|_\infty\}$ is zero. But then $\gamma_2 \coloneqq 1/\gamma_3$ is infinite. Since γ_1 is an expression of γ_2, we have that γ_1 is infinite as well. In the proof for Theorem 4.4.2 we see that $\gamma = \gamma_1$. Thus the right-hand side in (4.36) is infinite.

4.5 Solving the Lasso Problem

In order to solve (LS_τ), we can use an altered version of the *gradient descent method*. Traditional gradient descent only works for a non-constrained problem. Our Lasso problem has a constraint on the variable $\mathbf{x}$, namely $\|\mathbf{x}\|_1 \leq \tau$. We can alter the gradient descent method to work for a constrained problem. By moving in the direction of the negative *projected gradient* instead of the negative gradient, we ensure that our iterate candidate satisfies the constraint by projecting the iterate candidate into the feasible set. We will do this using the operator

$$P_\tau[\mathbf{c}] \coloneqq \{\underset{\mathbf{x}}{\operatorname{argmin}} \quad \|\mathbf{c}-\mathbf{x}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq \tau\}, \tag{4.39}$$

which gives the projection of an N-dimensional vector $\mathbf{c}$ onto the one-norm ball with radius τ.

Finding the ideal step length is an important and non-trivial part of gradient methods. The step length can determine whether or not our sequence will converge. In the SPGL1 algorithm, we use the Barzilai-Borwein step length, which was introduced by Jonathan Barzilai and Jonathan M. Borwein in [7]. According to their analysis, gradient descent algorithms with this step length achieve better performance and are cheaper to compute than the standard steepest-descent method.

Let $\mathbf{x}_{k+1} = P_\tau[\mathbf{x}_k - \alpha_k \mathbf{g}_k]$ be the iteration and $\mathbf{g}_k = -A^T\mathbf{r}_k$ be the current gradient. We denote the difference between two consecutive iterate candidates and two consecutive gradients as $\Delta\mathbf{x} = \mathbf{x}_k - \mathbf{x}_{k-1}$ and $\Delta\mathbf{g} = \mathbf{g}_k - \mathbf{g}_{k-1}$, respectively.

The step length α_k is chosen as the α that minimizes $\|\Delta\mathbf{x} - \alpha\Delta\mathbf{g}\|_2^2$. We note that

$$\|\Delta\mathbf{x} - \alpha\Delta\mathbf{g}\|_2^2 = (\Delta x_1 - \alpha\Delta g_1)^2 + (\Delta x_2 - \alpha\Delta g_2)^2 + \cdots + (\Delta x_N - \alpha\Delta g_N)^2,$$

where Δx_i is the i-th element in $\Delta\mathbf{x}$ and Δg_i is the i-th element in $\Delta\mathbf{g}$. Since the norm is continuous and differentiable, in order to find the α that minimizes this norm, we set the derivative equal to 0 and solve for α. We use the chain rule to find the derivative and get:

$$\begin{aligned}
\sum_{i=1}^{N} -2(\Delta x_i - \alpha\Delta g_i)\cdot\Delta g_i &= 0 \\
-2\sum_{i=1}^{N}\Delta x_i\Delta g_i + 2\sum_{i=1}^{N}\alpha\Delta g_i^2 &= 0 \\
\sum_{i=1}^{N}\Delta x_i\Delta g_i &= \sum_{i=1}^{N}\alpha\Delta g_i^2 \\
\alpha &= \sum_{i=1}^{N}\Delta x_i\Delta g_i \Big/ \sum_{i=1}^{N}\alpha\Delta g_i^2 \\
\alpha &= \langle\Delta\mathbf{x},\Delta\mathbf{g}\rangle/\langle\Delta\mathbf{g},\Delta\mathbf{g}\rangle.
\end{aligned}$$

The final equality uses that $\langle\mathbf{x},\mathbf{y}\rangle = \mathbf{x}^T\mathbf{y} = \sum_{i=1}^{N} x_i y_i$. Our step length is given by

$$\alpha_k = \langle\Delta\mathbf{x},\Delta\mathbf{g}\rangle/\langle\Delta\mathbf{g},\Delta\mathbf{g}\rangle. \tag{4.40}$$

Listing 4.1 is a translation of Algorithm 1 from [10] into MATLAB syntax. It also closely follows Algorithm 2.1 in [13]. Note that the following is not how the actual code for the SPGL1 algorithm is written. It merely outlines the SPG procedure that we have just described. A very important difference between Listing 4.1 and the actual SPGL1 code is that here the matrix A is assumed to be an explicit matrix, whereas in the SPGL1 code A can be an operator.

Listing 4.1: Spectral projected gradient for (LS_τ)

```
function [x_tau, r_tau] = spgl(b, A, x_in, tau, delta, varargin)

    alpha_min = options.step_min; % set min step length
    alpha_max = options.step_max; % set max step length
    gam = 1e-4; % set sufficient descent parameter gamma
    alpha_curr = 1; % set initial step length
    M = options.max_its; % set linesearch history

    % history of iterates
    x = zeros(M,1);
    r = zeros(M,1);
    g = zeros(M,1);

    % initial iterates
    x(1) = real_proj(x_in, tau); r(1) = b - A*x(1); g(1) = -A.'*r(1);

    i = 1;
    while 1
        % duality gap
        delta_curr = norm(r(i),2) - (b.'*r(i) - tau*norm(g(i), inf))/norm(r(i),2);
        if delta_curr < delta
            break
        end

        alpha = alpha_curr; % initial step length

        while 1
            x_bar = real_proj(x(i)-alpha*g(i));
            r_bar = b - A*x_bar;

            % find armijo condition
            armijo = norm(r(i),2)^2 + gam*(x_bar-x(i).'*g(i));
            for j=1:min(i,M-1)
                tmp = norm(r(i-j),2)^2 + gam*(x_bar-x(i).'*g(i));
                if armijo < tmp;
                    armijo = tmp;
                end
            end

            if norm(r_bar, 2)^2 <= armijo
                break
            else
                alpha = alpha/2; % decrease step length
            end
        end

        % update iterates
        x(i+1) = x_bar; r(i+1) = r_bar; g(i+1) = -A.'*r(i+1);
        dx = x(i+1) - x(i); dg = g(i+1) - g(i);

        % update Barzilai-Borwein step length
        if dx.' * dg <= 0
            alpha_curr = alpha_max;
```

```
54          else
55              bb_step = (dx.' * dx)/(dx.' * dg);
56              alpha_curr = min(alpha_max, max(alpha_min, bb_step));
57          end

59          i = i + 1;
60      end

62      x_tau = x(i); r_tau = r(i);
63  end
```

In this procedure, we update the iterates until the duality gap is acceptably small. If the current duality gap δ_i is less than our desired duality gap δ, the method has converged and we can return with our solution $\mathbf{x}_\tau = \mathbf{x}_i$ and corresponding residual $\mathbf{r}_\tau = \mathbf{r}_i$.

In lines 27 to 45 we perform a linesearch for the next iterate candidate. In line 28 we compute the projection of the iterate candidate into the feasible set. Since this is a potentially expensive step, a separate function `real_proj` has been written to efficiently perform this task. We will outline the algorithm behind this function in the next section. Lines 31 to 38 find the *non-monotone Armijo condition* (see, e.g., [22]). It ensures a sufficient decrease in the objective function at least every M iterations.

Lines 52 to 56 update the Barzilai-Borwein step length and ensure that α_{i+1} stays within the limits of $\alpha_{\min}$ and $\alpha_{\max}$.

One-norm Projection

As mentioned, the step of computing the projected gradient (4.39) can potentially be costly. We will now give an algorithm for computing this projection, with worst-case complexity of $O(n \log n)$. Numerical experiments in [10] show that on average the cost is much less than the worst-case cost.

To simplify the discussion, we assume that the entries of the vector $\mathbf{c}$ are non-negative, which we can do without loss of generality. If $\mathbf{c}$ had any negative entries, we could change the optimization problem in (4.39) to the equivalent problem

$$\underset{\mathbf{x}}{\operatorname{argmin}} \quad \|D\mathbf{c} - D\mathbf{x}\|_2 \quad \text{subject to} \quad \|D\mathbf{x}\|_1 \leq \tau, \tag{4.41}$$

where $D = \operatorname{diag}(\operatorname{sgn}(\mathbf{c}))$. We have used the convention from [14], where two optimization problems are equivalent if the solution of one problem can be readily found from the solution of the other problem. In this case, the two problems are related by a change of variable, $\mathbf{z} = D\mathbf{x}$, so the solution to (4.39) can be found from the solution to (4.41) by applying D^{-1}.

We will start by giving the motivation for the one-norm projection algorithm. The smallest possible value the norm in (4.39) could have is zero. It becomes zero if $\mathbf{x} = \mathbf{c}$, and so we start with this as our trial solution. If this is feasible, i.e., $\|\mathbf{c}\|_1 \leq \tau$, we have found the solution and can exit with $P_\tau[\mathbf{c}] \coloneqq \mathbf{x}^* = \mathbf{c}$. If not, we will try to reduce the norm of the trial $\mathbf{x}$ by the amount of infeasibility, which is

$$\nu \coloneqq \|\mathbf{x}\|_1 - \tau. \tag{4.42}$$

We want to find a correction vector $\mathbf{d}$ such that $\|\mathbf{x} - \mathbf{d}\|_1 = \tau$. We see that

$$\|\mathbf{c} - (\mathbf{x} - \mathbf{d})\|_2 = \|\mathbf{c} - \mathbf{x} + \mathbf{d}\|_2 \leq \|\mathbf{c} - \mathbf{x}\|_2 + \|\mathbf{d}\|_2,$$

so in order to minimize the potential increase in the objective value, we have to choose $\mathbf{d}$ such that we minimize $\|\mathbf{d}\|_2$. The vector $\mathbf{d}$ we are looking for is then a solution to

$$\underset{\mathbf{d}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{d}\|_2 \quad \text{subject to} \quad \mathbf{d} \geq 0,\ \|\mathbf{d}\|_1 = \nu. \tag{4.43}$$

With $\mathbf{e}$ a vector of ones, it is straightforward to verify that

$$\mathbf{d}^* = \gamma\mathbf{e} \quad \text{with} \quad \gamma = \nu/N, \tag{4.44}$$

is a solution to (4.43). For an intuitive argument, we look at the case where $\mathbf{d} \in \mathbb{R}^2$. Then the constraint $\mathbf{d} \geq 0$ confines us to the first quadrant. The constraint $\|\mathbf{d}\|_1 = \nu$ is a line segment from $(0, \nu)$ to $(\nu, 0)$. Minimizing $\|\mathbf{d}\|_2$ under these constraints then means finding the point along this line which is closest to the origin. This point is the center of the line segment, namely the point $(\nu/2, \nu/2)$.

Unfortunately, the solution $\mathbf{x} = \mathbf{c} - \mathbf{d}^*$ could still be infeasible. If some of the entries in $\mathbf{c} - \mathbf{d}^*$ are negative, the value of $\|\mathbf{x}\|_1$ would increase. To combat this problem, we ensure that our solution preserves the sign pattern of $\mathbf{c}$. If every element in $\mathbf{d}^*$ is smaller than the smallest element in $\mathbf{c}$, i.e., if each

$$d_i^* < c_{\min} \coloneqq \min_i c_i, \tag{4.45}$$

none of the elements in $\mathbf{x} = \mathbf{c} - \mathbf{d}^*$ can be negative, and we exit with this as the solution to (4.39). Otherwise, we set all the elements that would be negative to zero, i.e.,

$$x_i = 0 \quad \text{for all} \quad i \in \mathcal{I} \coloneqq \{i : d_i^* \geq c_{\min}\}. \tag{4.46}$$

Listing 4.2 shows a translation of Algorithm 2 in [10] into MATLAB syntax. In order to improve the efficiency of the algorithm and reduce cost from bookkeeping, the procedure is applied to a growing subvector of $\mathbf{c}$. This way, we do not need to sort the entire vector $\mathbf{c}$. For the first iteration, we start with a single element which is the largest element in magnitude from $\mathbf{c}$. For each subsequent iteration we add the next element from $\mathbf{c}$ that is largest in magnitude. The variable name `c_min` can be a little confusing, as we are extracting the largest element of $\mathbf{c}$. However, this name was chosen because it corresponds to the name used in (4.45) and (4.46), and because the element we are extracting will become the minimum element in the current subvector.

Listing 4.2: Real projection onto the feasible set

```
1  function [x] = real_proj(c, tau)
2      n = size(c,2);

4      if norm(c,1) <= tau % c is feasible
5          x = c;
6          return
7      end

9      delta = 0; nu = -tau; gam = 0; % gamma

11     c_bar = build_heap(abs(c));

13     for j=1:n
14         c_min = c_bar(1); % extract largest element
```

```
15          nu = nu + c_min; % accumulate infeasibility
16          gam = nu/j; % define current solution

18          if gam >= c_min % remaining iterations satisfy soft thresholding condition
19              break
20          end

22          c_bar = delete_max(c_bar);
23          delta = gam; % element in d
24      end
25      x = soft_threshold(c, delta);
26  end
```

This MATLAB script assumes a few functions exist. The function `build_heap` should build a binomial heap with the largest element in magnitude as the first element. The function `delete_max` removes the current largest element and restores the heap property. The function `soft_threshold` corresponds to (4.46).

The soft-thresholding operation would behave differently if $\mathbf{c} \in \mathbb{C}^N$ and the algorithm outlined here would then have to be modified. For an algorithm for the complex one-norm projection, see Algorithm 3 in [10].

CHAPTER 5

New Compressed Sensing Theory

In Chapter 2 we discussed the main concepts in traditional CS. In Chapter 4 we discussed an algorithm that efficiently solves the main optimization problem in CS. Now we are interested in how the sampling process and recovery work in practice.

As it turns out, there is a gap between the theory that we outlined and its use in practice. The theory is built on three important pillars: sparsity, incoherence and uniform random subsampling. There are cases where we have all of these and recovery is successful, but quite often at least one of these properties is missing. If that is the case, we have to adapt our measurement process and sampling model to achieve satisfactory recovery results.

In Figure 5.1, we see two reconstructions of a 512×512 gray-scale image of the Oslo Opera House. The image on the left was achieved using uniform random subsampling, which is what the CS theory suggests. This poor reconstruction is an example of the actual process not aligning with the theory. The image on the right is clearly a much better reconstruction. This image is the result of a multilevel sampling pattern.

In this chapter we will discuss why the multilevel sampling pattern works and define a new CS theory (first established in [2, Part III] and [3, 24]) that better supports what is happening.

5.1 Sampling Structure

As we see in Figure 5.1, the structure of the sampling operator is important. The multilevel sampling pattern performed much better than uniform random sampling. We can explain this by examining the coherence of the matrix $\Phi\Psi^{-1}$, with Φ being the Hadamard matrix and Ψ the DWT matrix. This is seen in Figure 3.4 from Section 3.3.

The matrix $\Phi\Psi^{-1}$ has a block structure, where the non-zero elements lie along the main diagonal. Even though most of the elements in the matrix are zero, the matrix has a high global coherence because there are elements with magnitudes close to 1. A high global coherence means we need a higher number of samples to get uniform recovery, according to Theorem 2.5.4. The measurements corresponding to blocks with high coherence are likely to contain important information about the signal. Therefore, we are most interested in drawing these samples. Even if the number of samples satisfies the lower bound in Theorem 2.5.4, if we sample uniformly at random we could end up drawing

Uniform random Circle

Figure 5.1: Two recoveries of a 512×512 image of the Oslo Opera House from 20% Hadamard samples using DB4 wavelets. Original image courtesy of Alexandra von Gutthenbach-Lindau from Pixabay.

most of our samples from areas outside the main diagonal, and thus get a poor reconstruction.

We need a new, local notion of coherence so that we can sample with respect to the structure in the coherence. That way, we can design sampling patterns that sample densely in the blocks with high coherence and less densely in the blocks with smaller coherence. We will define local coherence and this new way of sampling later in the chapter.

5.2 Sparsity Structure

Figure 5.1 demonstrates that the structure in the sampling operator affects the recovery, but what about the structure in the sparsifying operator, i.e., the wavelets? The standard CS theory tells us that the core assumption for successfull recovery is *sparsity*. That is, as long as we have a signal where most of the coefficients are zero, we should be able to recover it. In practice, we see that the regular notion of sparsity alone is not sufficient to guarantee recovery. Information about the structure of the sparsity, i.e., where the non-zero elements are located, is in fact necessary.

To illustrate this, we perform an experiment called the *flip test*. The flip test is based on the fact that sparsity is independent of permutation. Recall that a vector is s-sparse if it has at most s non-zero elements. The number s does not depend on the location of these non-zero elements. Thus, an s-sparse vector $\mathbf{x}$ and a vector $\mathbf{x}' = Q\mathbf{x}$, where Q is a permutation matrix, will have the same sparsity. If sparsity alone is sufficient for recovery, then we would expect the same recovery quality whether we recover the original signal $\mathbf{x}$ from $\mathbf{y} = A\mathbf{x}$ or $\mathbf{y}' = A\mathbf{x}'$. With the flip test, we will attempt to recover the original signal from the permutation and compare the results to a standard recovery.

By varying how we define the permutation matrix Q, we can investigate different ways the sparsity might be structured. We perform two different

versions of the flip test, which we call the *global flip test* and the *flip test in levels*. In the former version, we let Q be a so-called global permutation, corresponding to flipping all the wavelet coefficients at once. That is, the first and last coefficients switch places, the second and second to last coefficients switch places, and so on. With the global flip test, we find that the reconstruction quality is drastically worse than for a standard recovery, meaning that the structure of the sparsity does matter.

For the flip test in levels, we will define Q as local permutations within each wavelet scale, or *level*. The first and last coefficients within a level switch places, the second and second to last coefficients within a level switch places, and so on.

Let $\mathbf{x}$ be the original signal we seek to recover in vector form. Let $\mathbf{d} = \Psi\mathbf{x}$ be the coefficients of $\mathbf{x}$ in the sparsifying transform, which we have chosen to be the DWT. The matrix $A = P_\Omega \Phi \Psi^{-1}$ is as usual the measurement matrix and Q is the permutation matrix. The steps of the flip test are summarized below.

1. Compute the coefficients $\mathbf{d} = \Psi\mathbf{x}$.
2. Flip the coefficients, resulting in $\mathbf{d}' = Q\mathbf{d}$.
3. Reconstruct $\hat{\mathbf{d}}$ from $\mathbf{y} = A\mathbf{d}$.
4. Reconstruct $\hat{\mathbf{d}}'$ from $\mathbf{y}' = A\mathbf{d}'$.
5. Flip $\hat{\mathbf{d}}'$ back, resulting in $\check{\mathbf{d}}$.
6. Perform the inverse transformation, resulting in $\hat{\mathbf{x}} = \Psi^{-1}\hat{\mathbf{d}}$ and $\check{\mathbf{x}} = \Psi^{-1}\check{\mathbf{d}}$.
7. Compare $\hat{\mathbf{x}}$ and $\check{\mathbf{x}}$.

For the flip test experiments, we have used the test images in Figure 5.2, which have been converted to gray-scale and resized to 512×512. We have sampled using the Hadamard transform and sparsified by a DB4 wavelet. We have used three different sampling patterns, seen in Figure 5.3.

The results of the flip test experiments are found in Figure 5.4 and Figure 5.5. The middle columns correspond to the global flip test. The right-most columns correspond to the flip test in levels. For comparison, a standard recovery of the images can be found in the left-most columns.

The quality of the reconstruction for the global flip test is clearly worse than for a standard CS recovery. It is obvious by examining the results visually, but we can also compare the peak signal-to-noise ratio (PSNR), which is a measure of the quality of the recovery. Take for example the recovery of the pig image using the circle sampling pattern. For the standard CS recovery, the PSNR is 25.0. For the global flip test, the PSNR is only 7.5. If the structure of the sparsity had no impact on recovery, we would expect the PSNR scores to be equal.

We note that for the flip test in levels, the PSNR scores are much closer to being equal to those of the standard recoveries. If we look at the recovery of the pig image using the circle sampling pattern again, both the standard CS recovery and the flip test in levels recovery have a PSNR of 25.0. This means that the structure of the sparsity likely corresponds to the scales of the wavelet we are using. In the next section, we will define a new notion of sparsity that can describe this behavior.

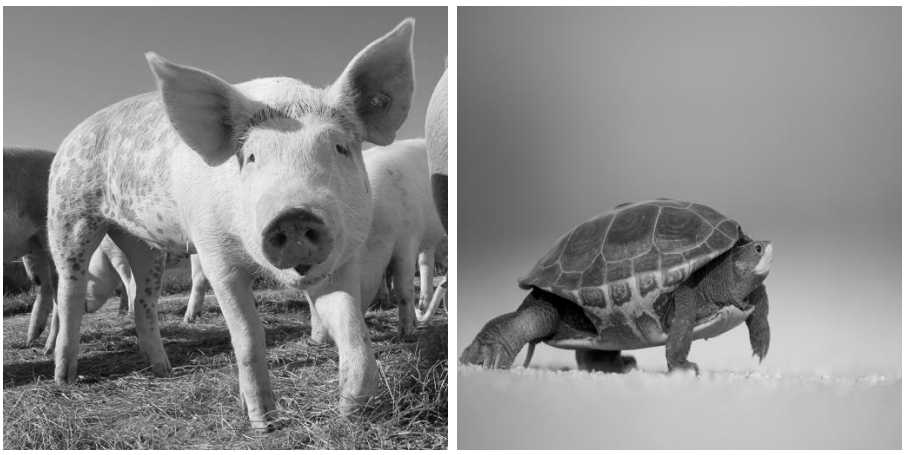
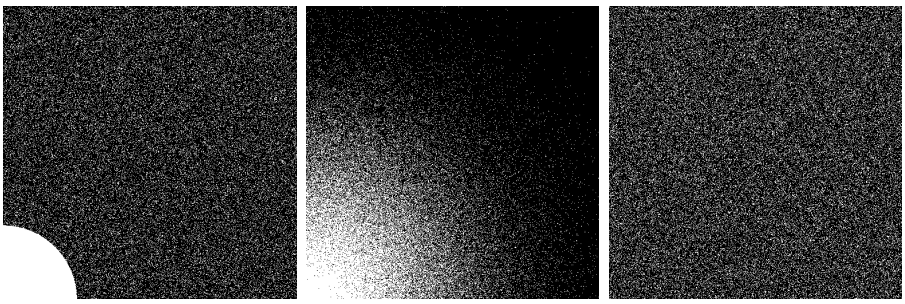


Figure 5.2: Two 512×512 test images of a pig and a turtle. The pig image is by Marion Streiff and the turtle image by Pexels from Pixabay.



2 level Circle Uniform

Figure 5.3: The sampling patterns for the flip test.

The structure in the sparsity that we have observed here aligns with the structure in the coherence that we discussed in the previous section. Like how the Hadamard-wavelet matrix is asymptotically incoherent, the wavelet coefficients are *asymptotically sparse*. That is, they are more sparse at coarse wavelet scales and less and less sparse for finer and finer scales. This is why the reconstruction quality in the flip test in levels was better for the 2 level and circle sampling patterns than for the uniform random sampling pattern. The first two patterns are what we call multilevel subsampling patterns, which take the structure in the coherence into account. Section 5.5 gives a definition of multilevel subsampling patterns.

5.3 Asymptotic Sparsity

As mentioned in the introduction to this chapter, the standard CS theory is insufficient to describe what we have just seen. This is because the standard theory is based on global notions of sparsity and incoherence. We will now introduce local versions of these properties.

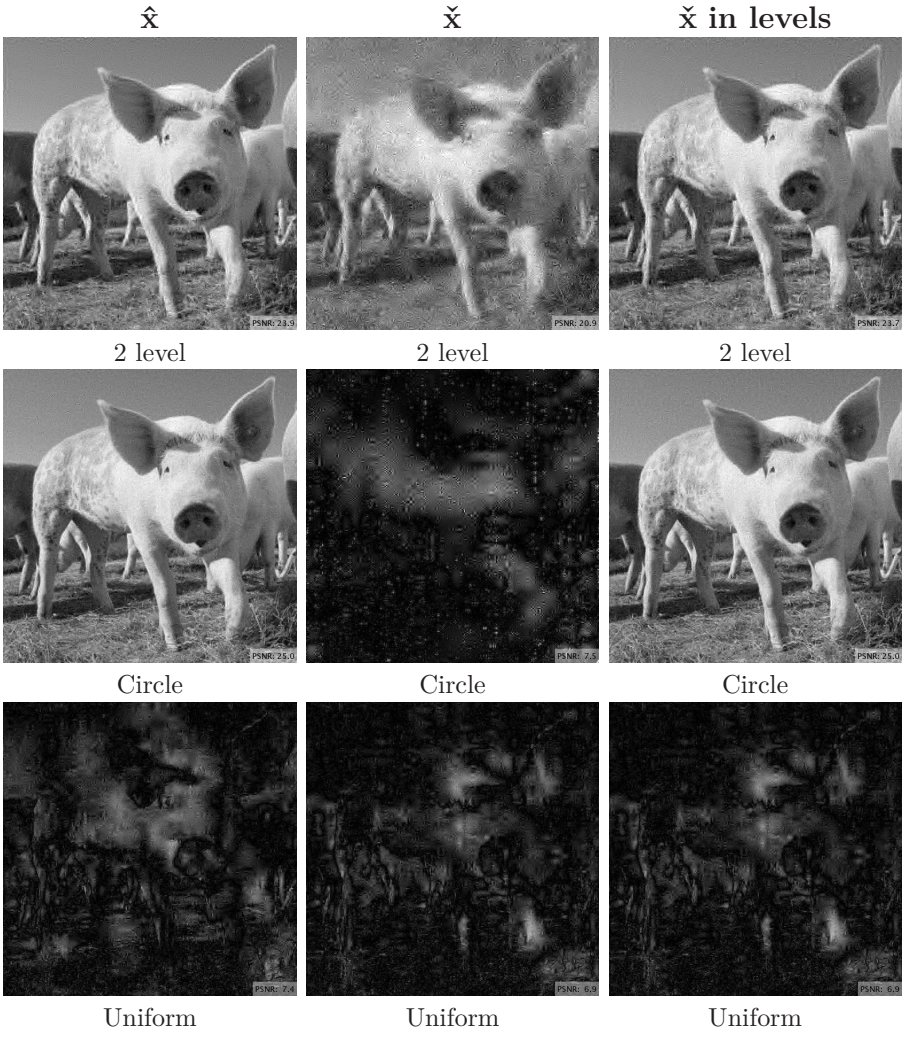


Figure 5.4: The flip test experiments using Hadamard sampling with Daubechies wavelets on the pig image. The first column corresponds to standard CS recovery, the second column to the global flip test and the last column to the flip test in levels. Figure 5.3 shows the sampling patterns that were used.

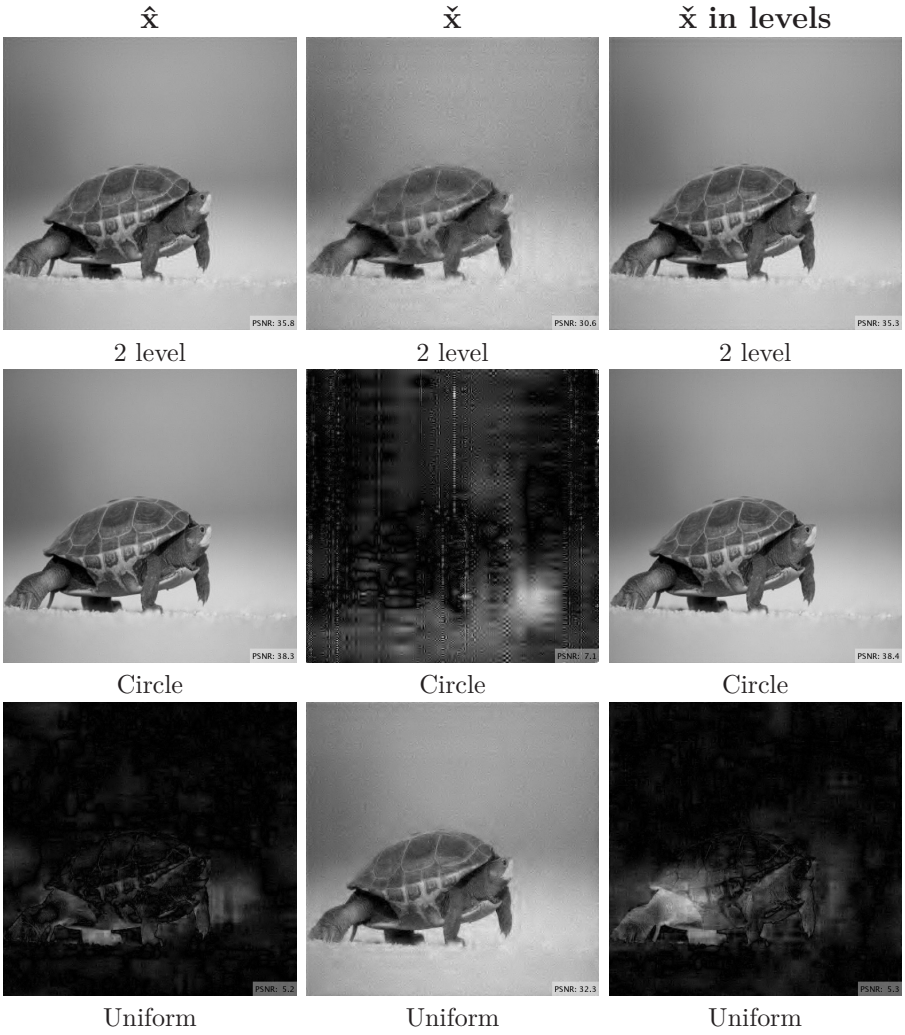


Figure 5.5: The flip test experiments using Hadamard sampling with Daubechies wavelets on the turtle image. The first column corresponds to standard CS recovery, the second column to the global flip test and the last column to the flip test in levels. Figure 5.3 shows the sampling patterns that were used.

Definition 5.3.1. [2, Definition 11.1] Let $1 \leq r \leq N$ and $M_0 = 0$. Further, let $\mathbf{M} = (M_1, \ldots, M_r)$ where $1 \leq M_1 < \cdots < M_r = N$ and $\mathbf{s} = (s_1, \ldots, s_r)$ where $s_k \leq M_k - M_{k-1}$ for $k = 1, \ldots, r$. A vector $\mathbf{x} \in \mathbb{R}^N$ is $(\mathbf{s}, \mathbf{M})$-*sparse in levels* if

$$|\text{supp}(\mathbf{x}) \cap \{M_{k-1} + 1, \ldots, M_k\}| \leq s_k \quad k = 1, \ldots, r.$$

The total sparsity is $s = s_1 + \cdots + s_r$.

The vector $\mathbf{x}$ is $(\mathbf{s}, \mathbf{M})$-sparse in levels if for each level i, there are at most s_i non-zero elements. The vector $\mathbf{M}$ defines the r sparsity levels. The sparsity in levels definition is general, but in the case where we use the wavelet transform for sparsifying, the sparsity levels would correspond to the wavelet scales. For the Haar wavelet, $M_k = 2^k$. The k-th level would consist of indices $\{2^{k-1} + 1, 2^{k-1} + 2, \ldots, 2^k\}$.

A signal has *asymptotic sparsity* if

$$s_k/(M_k - M_{k-1}) \to 0$$

rapidly as $k \to \infty$.

We can also redefine the error of best approximation,

$$\sigma_{\mathbf{s},\mathbf{M}}(\mathbf{x})_p \coloneqq \{\|\mathbf{x} - \mathbf{z}\|_p : \mathbf{z} \text{ is } (\mathbf{s}, \mathbf{M})\text{-sparse}\}.$$

5.4 Asymptotic Incoherence

As we have previously discussed, the matrix $\Phi\Psi^{-1}$, with Φ being the Hadamard matrix and Ψ the wavelet matrix, will have a high global coherence. Even though most of the matrix elements are zero, the matrix has elements with magnitude close to 1 in the top left corner. Therefore choosing the maximum of the magnitudes of the elements will yield a global coherence close to 1. We can define a *local* coherence that is the maximum element in magnitude within a given *region* of the matrix. With this local version, we can design sampling patterns that utilizes the block structure in the coherence.

Definition 5.4.1. [20, Definition 2.8] Let $\mathbf{N} = (N_1, \ldots, N_r)$ be the sampling levels and $\mathbf{M} = (M_1, \ldots, M_r)$ the sparsity levels. The (k, l)-th *local coherence* of an isometry $U \in \mathbb{R}^{N \times N}$ is given by

$$\mu_{k,l} = \mu_{k,l}(U) \coloneqq \max\{|u_{i,j}|^2 : i = N_{k-1} + 1, \ldots, N_k, j = M_{l-1} + 1, \ldots, M_l\}.$$

We say that a matrix has *asymptotic incoherence* if we can remove either the first K columns or rows, and get new matrices with small global coherence. We recall again Figure 3.4. The combination of Hadamard and Daubechies wavelets is asymptotically incoherent, because the highest values are concentrated in the first K rows and columns.

5.5 Multilevel Subsampling

If we have a matrix with asymptotic incoherence, we would get poor performance if we sampled uniformly at random from the whole matrix. We saw an example of this in Figure 5.1. We could end up under-sampling in the areas with high

coherence and sampling unnecessarily in areas with low coherence. By sampling in levels, we can sample fully in the first levels with the highest coherence, and sample less and less in the consecutive levels.

Definition 5.5.1. [2, Definition 11.5] Let $N_0 = 0$ and $\mathbf{N} = (N_1, \dots, N_r)$ where $1 \leq N_1 < \cdots < N_r = N$ and $\mathbf{m} = (m_1, \dots, m_r)$ where $m_k \leq N_k - N_{k-1}$ for $k = 1, \dots, r$. An $(\mathbf{m}, \mathbf{N})$-*multilevel random subsampling scheme* is a set $\Omega = \Omega_1 \cup \cdots \cup \Omega_r$ of $m = m_1 + \cdots + m_r$ indices, where for each k the following holds: If $m_k = N_k - N_{k-1}$, then $\Omega_k = \{N_{k-1} + 1, \dots, N_k\}$, otherwise Ω_k consists of m_k indices chosen independently and uniformly at random from the set $\{N_{k-1} + 1, \dots, N_k\}$.

The first two sampling patterns in Figure 5.3 are examples of multilevel subsampling patterns.

5.6 Restricted Isometry Property in Levels

We can also define an in-levels version of the restricted isometry property, the RIPL. We will show later that with enough samples, the measurement matrix will with high probability satisfy the RIPL and consequently yield uniform recovery. This is analogous to Theorem 2.5.4 in Chapter 2.

Definition 5.6.1. [20, Definition 2.12] Let $\mathbf{M} = (M_1, \dots, M_r)$ be sparsity levels and $\mathbf{s} = (s_1, \dots, s_r)$ be local sparsities. The $\mathbf{s}$-th *restricted isometry constant in levels (RICL)* $\delta_{(\mathbf{s},\mathbf{M})}$ of a matrix $A \in \mathbb{R}^{m \times N}$ is the smallest constant $\delta \geq 0$ such that

$$(1 - \delta)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1 + \delta)\|x\|_2^2, \quad \text{for all } (\mathbf{s}, \mathbf{M})\text{-sparse } \mathbf{x}.$$

If $0 < \delta_{(\mathbf{s},\mathbf{M})} < 1$, we say that A satisfies the *restricted isometry property in levels (RIPL)*.

As with the standard RIP, the RIPL implies uniform recovery.

Theorem 5.6.2. *[20, Theorem 2.13] Let $r \in \mathbb{N}$. Suppose that $A \in \mathbb{R}^{m \times N}$ satisfies the RIPL of order $(2\mathbf{s}, \mathbf{M})$ with*

$$\delta_{(2\mathbf{s},\mathbf{M})} < \frac{1}{\sqrt{r(\sqrt{\alpha} + 1/4)^2 + 1}}$$

where

$$\alpha = \max_{k,l=1,\dots,r} \{s_k / s_l\}.$$

Then for any $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^m$ with $\|A\mathbf{x} - \mathbf{y}\|_2 \leq \eta$, a solution $\mathbf{x}^\# \in \mathbb{R}^N$ of

$$\underset{\mathbf{z} \in \mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \textit{subject to} \quad \|A\mathbf{z} - \mathbf{y}\|_2 \leq \eta$$

approximates the vector $\mathbf{x}$ with errors

$$\|\mathbf{x} - \mathbf{x}^\#\|_1 \lesssim \sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1 + \sqrt{s}\eta$$

$$\|\mathbf{x} - \mathbf{x}^\#\|_2 \lesssim (1 + (r\alpha)^{1/4}) \frac{\sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1}{\sqrt{s}} + (1 + (r\alpha)^{1/4})\eta,$$

where $s = s_1 + \cdots + s_r$.

Since the RIPL implies uniform recovery, we are interested in knowing how many samples we need to achieve the RIPL. For technical purposes, we need the following scaling of the measurement matrix. Let $U \in \mathbb{R}^{N\times N}$ be an isometry and $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N},\mathbf{m})$-multilevel subsampling scheme. We then define the matrix

$$A = P_\Omega D U \in \mathbb{R}^{m\times N}, \tag{5.1}$$

where $D \in \mathbb{R}^{N\times N}$ is a diagonal scaling matrix with

$$d_{i,i} = \begin{cases} \sqrt{\frac{N_k - N_{k-1}}{m_k}}, & \text{if } m_k \neq 0 \\ 1, & \text{if } m_k = 0 \end{cases} \qquad N_{k-1} < i \leq N_k,\ k = 1,\dots,r.$$

Theorem 5.6.3. *[20, Theorem 3.1] Let $U \in \mathbb{R}^{N\times N}$ be an isometry, $r \in \mathbb{N}$, $\epsilon > 0$ and $\delta < 1$. Let $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N},\mathbf{m})$-multilevel subsampling scheme, $\mathbf{M}$ be the sparsity levels and $\mathbf{s}$ be the local sparsities. Suppose that*

$$m_k \gtrsim \delta^{-2}\cdot(N_k - N_{k-1})\cdot\left(\sum_{l=1}^{r}\mu_{k,l}\cdot s_l\right)\cdot(r\log(2m)\log(2N)\log^2(2s)+\log(1/\epsilon)) \tag{5.2}$$

for $k = 1,\dots,r$ where $m = m_1 + \cdots + m_r$. Then with probability at least $1-\epsilon$, the matrix A as defined in (5.1) satisfies the RIPL of order $(\mathbf{s},\mathbf{M})$ with constant $\delta_{(\mathbf{s},\mathbf{M})} \leq \delta$.

Finally, we have reached our main result in the new CS theory. It gives an estimate on the sampling size m_k for each sampling level in order to achieve uniform recovery with high probability.

Theorem 5.6.4. *Let $U \in \mathbb{R}^{N\times N}$ be an isometry, $r \in \mathbb{N}$ and $\epsilon > 0$. Let $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N},\mathbf{m})$-multilevel subsampling scheme, $\mathbf{M}$ be the sparsity levels and $\mathbf{s}$ be the local sparsities. Let $A \in \mathbb{R}^{m\times N}$ be defined as in (5.1). Suppose that*

$$m_k \gtrsim (r(\sqrt{\alpha}+1/4)^2+1)\cdot(N_k - N_{k-1})\cdot\left(\sum_{l=1}^{r}\mu_{k,l}\cdot s_l\right)\cdot L \tag{5.3}$$

for $k = 1,\dots,r$ where $m = m_1 + \cdots + m_r$, L is the same log-factor as in (5.2), and

$$\alpha = \alpha_{(\mathbf{s},\mathbf{M})} = \max_{k,l=1,\dots,r}\{s_k/s_l\}.$$

Then with probability at least $1-\epsilon$, for any $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^m$ with $\|A\mathbf{x}-\mathbf{y}\|_2 \leq \eta$, a solution $\mathbf{x}^\# \in \mathbb{R}^N$ of

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \textit{subject to} \quad \|A\mathbf{z}-\mathbf{y}\|_2 \leq \eta$$

approximates the vector $\mathbf{x}$ with errors

$$\|\mathbf{x}-\mathbf{x}^\#\|_1 \lesssim \sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1 + \sqrt{s}\eta$$
$$\|\mathbf{x}-\mathbf{x}^\#\|_2 \lesssim (1+(r\alpha)^{1/4})\frac{\sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1}{\sqrt{s}} + (1+(r\alpha)^{1/4})\eta,$$

where $s = s_1 + \cdots + s_r$.

Proof. Assume that (5.3) holds. Let $\delta^{-2} > r(\sqrt{\alpha} + 1/4)^2 + 1$ where $\alpha = \max_{k,l=1,\dots,r}\{s_k/s_l\}$. Then by Theorem 5.6.3, the matrix $A = P_\Omega DU$ satisfies the RIPL of order $(2\mathbf{s}, \mathbf{M})$ with $\delta_{(2\mathbf{s},\mathbf{M})} < \frac{1}{\sqrt{r(\sqrt{\alpha}+1/4)^2+1}}$. By Theorem 5.6.2, we have

$$\|\mathbf{x} - \mathbf{x}^\#\|_1 \lesssim \sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1 + \sqrt{s}\eta,$$
$$\|\mathbf{x} - \mathbf{x}^\#\|_2 \lesssim (1 + (r\alpha)^{1/4})\frac{\sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1}{\sqrt{s}} + (1 + (r\alpha)^{1/4})\eta,$$

where $s = s_1 + \cdots + s_r$, which is the desired result. ■

The appearance of the sparsity ratio α in this estimate is unfortunate. When there is a significant difference between the biggest and smallest sparsity levels, the ratio α and consequently the number of required samples m_k will become quite large. However, we can remove α from this estimate if we replace the ℓ_1-minimization problem with a weighted ℓ_1-minimization problem. This has been shown in [26].

The estimate for m_k is dependent on the local sparsity s_k. This provides an explanation for the results of the flip test experiments. We observed that the recovery quality depends on the structure of the sparsity. When we flip all the coefficients at once we get poor recovery, whereas flipping the coefficients within the wavelet scales achieves the same quality as a standard CS recovery. If we flip all the coefficients at once, the local sparsities s_k may change, and we could end up with a greater lower bound on m_k. Then the number of samples we have drawn may be too low and we get a poor reconstruction. For permutations within the levels, the local sparsities do not change and the number of samples drawn still meets the criteria for uniform recovery.

The estimate (5.3) also formalizes the need to sample fully in levels with high coherence and less densely in levels with small coherence. For levels with high local coherence, i.e., $\mu_{k,l}$ close to 1, the factor $\sum_{l=1}^r \mu_{k,l} \cdot s_l$ makes a considerable contribution to the sample size estimate. For levels with small coherence, $\sum_{l=1}^r \mu_{k,l} \cdot s_l$ will be close to zero, making the sample size estimate close to zero as well. For completely incoherent levels, we do not need to sample at all.

The final theorem in this chapter is a restatement of Theorem 5.6.4 for the case where U is the Haar-Hadamard matrix.

Theorem 5.6.5. *Let $U = \Phi\Psi^{-1} \in \mathbb{R}^{N\times N}$ with Φ as the Hadamard matrix and Ψ as the Haar wavelet matrix. Let $r \in \mathbb{N}$ and $\epsilon > 0$. Let $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be an $(\mathbf{N},\mathbf{m})$-multilevel subsampling scheme, and let $\mathbf{N} = \mathbf{M} = (2^1,\dots,2^r)$ be the sampling and sparsity levels, respectively. Let $\mathbf{s}$ be the local sparsities and $A \in \mathbb{R}^{m\times N}$ be defined as in (5.1). Suppose that*

$$m_k \gtrsim (r(\sqrt{\alpha} + 1/4)^2 + 1) \cdot s_k \cdot (r\log(2m)\log(2^{r+1})\log^2(2s) + \log(1/\epsilon)) \quad (5.4)$$

for $k = 1,\dots,r$ where $m = m_1 + \cdots + m_r$ and

$$\alpha = \alpha_{(\mathbf{s},\mathbf{M})} = \max_{k,l=1,\dots,r}\{s_k/s_l\}.$$

Then with probability at least $1-\epsilon$, for any $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^m$ with $\|A\mathbf{x} - \mathbf{y}\|_2 \le \eta$, a solution $\mathbf{x}^\# \in \mathbb{R}^N$ of

$$\underset{\mathbf{z}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{z}\|_1 \quad \textit{subject to} \quad \|A\mathbf{z} - \mathbf{y}\|_2 \le \eta$$

approximates the vector $\mathbf{x}$ *with errors*

$$\|\mathbf{x}-\mathbf{x}^{\#}\|_1 \lesssim \sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1 + \sqrt{s}\eta$$
$$\|\mathbf{x}-\mathbf{x}^{\#}\|_2 \lesssim (1+(r\alpha)^{1/4})\frac{\sigma_{(\mathbf{s},\mathbf{M})}(\mathbf{x})_1}{\sqrt{s}} + (1+(r\alpha)^{1/4})\eta,$$

where $s = s_1 + \cdots + s_r$.

Proof. By Proposition 4.10 in [1] with $J_0 = 0$, the matrix $P_N U P_N$ is an isometry. Let $L \coloneqq (r\log(2m)\log(2N)\log^2(2s) + \log(1/\epsilon))$. Then by Theorem 5.6.4, we need

$$m_k \gtrsim (r(\sqrt{\alpha}+1/4)^2+1)\cdot(N_k - N_{k-1})\cdot\left(\sum_{l=1}^{r}\mu_{k,l}\cdot s_l\right)\cdot L \tag{5.5}$$

in order to achieve the desired error estimates. We have that $N_k = 2^k$, so $N_k - N_{k-1} = 2^k - 2^{k-1} = 2^{k-1}$. We can rewrite the sum above as

$$\sum_{l=1}^{r}\mu_{k,l}\cdot s_l = \mu_{k,k}\cdot s_k + \sum_{l=1,l\neq k}^{r}\mu_{k,l}\cdot s_l. \tag{5.6}$$

By Proposition 4.11 in [1] with $J_0 = 0$, we have that the local coherences are

$$\mu_{k,l} = \begin{cases} 2^{-k+1}, & \text{if } k = l; \\ 0, & \text{if } k \neq l. \end{cases} \tag{5.7}$$

Then the sum (5.6) becomes $2^{-k+1}\cdot s_k$. We have $(N_k - N_{k-1})\cdot\sum_{l=1}^{r}\mu_{k,l}\cdot s_l = 2^{k-1}\cdot 2^{-k+1}\cdot s_k = s_k$. Inserting this into (5.5) gives the desired result. ■

We conclude this section by making a note on the total number of measurements m required in the Haar-Hadamard case. For readability, we let $B \coloneqq (r(\sqrt{\alpha}+1/4)^2+1)$ and $L \coloneqq (r\log(2m)\log(2^{r+1})\log^2(2s)+\log(1/\epsilon))$. Then,

$$\begin{aligned} m &= m_1 + m_2 + \cdots + m_r \\ &\gtrsim Bs_1L + Bs_2L + \cdots + Bs_rL \\ &= BL(s_1 + s_2 + \cdots + s_r) \\ &= BLs. \end{aligned}$$

We see that the number of measurements m needed for uniform recovery scales linearly in the total sparsity s with a mild log-factor.

CHAPTER 6

Conclusion

In this thesis we have presented some key concepts, results and concerns from the field of compressed sensing (CS). We have also discussed what is necessary to be able to apply this theory to real-world applications.

We started with a review of the traditional CS theory in Chapter 2. Here we discussed the main principles of CS: sparsity, incoherence and uniform random subsampling. We also proved some important recovery results, including our main result, which gives an estimate on how many samples m are required for uniform recovery.

In Chapter 3 we gave a brief introduction to wavelets and the Hadamard transform, and discussed the coherence between the wavelets and the Hadamard matrices. We saw that the coherence had an asymptotic structure.

Chapter 4 provided the mathematical motivation for the SPGL1 algorithm and gave pseudocode for critical parts of the algorithm.

Finally, in Chapter 5 we performed numerical experiments that showed issues with the global principles from Chapter 2. The numerical experiments showed that there is an asymptotic structure in the sparsity that aligns with the previously mentioned asymptotic structure in the coherence. To better explain the way CS works in practice, we introduced a new local CS theory, which included four concepts: sparsity in levels, coherence in levels, the restricted isometry property in levels and multilevel random sampling. The last section in Chapter 5 provided a proof for an important recovery result, namely that for a certain number of samples in each sampling level, we have uniform recovery.

APPENDIX A

Extra derivations

A.1 Telescoping Series

We will show that by applying the inequality

$$|\tau_{k+1} - \tau_\sigma| \leq \gamma_1 \delta_k + \eta_k |\tau_k - \tau_\sigma| \tag{A.1}$$

(with $\eta_k \to 0$ and $\eta_k < 1$) recursively $\ell \geq 1$ times, we obtain

$$|\tau_{k+\ell} - \tau_\sigma| \leq \gamma_1 \sum_{i=1}^{\ell} (\eta_k)^{\ell-i} \, \delta_{k+i-1} + (\eta_k)^\ell |\tau_k - \tau_\sigma|.$$

Ideally, we would prove this by induction. However, for the sake of readability, we will show that the inequality holds for $\ell = 1, 2, 3$ and from the pattern that emerges, it will become clear that it must hold for any $\ell \geq 1$.

For the case $\ell = 1$, we have

$$\begin{aligned} |\tau_{k+1} - \tau_\sigma| &\leq \gamma_1 \delta_k + \eta_k |\tau_k - \tau_\sigma| \\ &= \gamma_1 (\eta_k)^0 \delta_{k+1-1} + (\eta_k)^1 |\tau_k - \tau_\sigma| \\ &= \gamma_1 \sum_{i=1}^{1} (\eta_k)^{1-i} \delta_{k+i-1} + (\eta_k)^1 |\tau_k - \tau_\sigma|. \end{aligned}$$

For $\ell = 2$, we have

$$\begin{aligned} |\tau_{k+2} - \tau_\sigma| &\leq \gamma_1 \delta_{k+1} + \eta_{k+1} |\tau_{k+1} - \tau_\sigma| \\ &\leq \gamma_1 \delta_{k+1} + \eta_{k+1} (\gamma_1 \delta_k + \eta_k |\tau_k - \tau_\sigma|) \\ &= \gamma_1 \delta_{k+1} + \eta_{k+1} \gamma_1 \delta_k + \eta_{k+1} \eta_k |\tau_k - \tau_\sigma| \\ &\leq \gamma_1 \delta_{k+1} + \eta_k \gamma_1 \delta_k + (\eta_k)^2 |\tau_k - \tau_\sigma| \end{aligned}$$

where we have used the fact that $\eta_k < 1$ implies that $\eta_{k+1} < \eta_k$. If we factor out γ_1 from the first two terms, we get the expression we are trying to obtain.

The case where $\ell = 3$ is much like the previous case, but we include it to make the pattern more clear. Applying (A.1) recursively 3 times, we get

$$\begin{aligned} |\tau_{k+3} - \tau_\sigma| &\leq \gamma_1 \delta_{k+2} + \eta_{k+2} |\tau_{k+2} - \tau_\sigma| \\ &\leq \gamma_1 \delta_{k+2} + \eta_{k+2} (\gamma_1 \delta_{k+1} + \eta_{k+1} |\tau_{k+1} - \tau_\sigma|) \\ &\leq \gamma_1 \delta_{k+2} + \eta_{k+2} (\gamma_1 \delta_{k+1} + \eta_{k+1} (\gamma_1 \delta_k + \eta_k |\tau_k - \tau_\sigma|)) \end{aligned}$$

$$
\begin{aligned}
&= \gamma_1\delta_{k+2} + \eta_{k+2}\gamma_1\delta_{k+1} + \eta_{k+2}\eta_{k+1}(\gamma_1\delta_k + \eta_k|\tau_k - \tau_\sigma|) \\
&= \gamma_1\delta_{k+2} + \eta_{k+2}\gamma_1\delta_{k+1} + \eta_{k+2}\eta_{k+1}\gamma_1\delta_k + \eta_{k+2}\eta_{k+1}\eta_k|\tau_k - \tau_\sigma| \\
&\leq \gamma_1\delta_{k+2} + \eta_k\gamma_1\delta_{k+1} + (\eta_k)^2\gamma_1\delta_k + (\eta_k)^3|\tau_k - \tau_\sigma|.
\end{aligned}
$$

Again, factoring out γ_1 gives the desired expression.

A.2 Column Coherence

Consider the column coherence from Definition 2.4.2. We observe that for $1 \leq s \leq N-1$, we have

$$
\begin{aligned}
\mu_c &= \max_{1\leq i\neq j\leq N} |\langle \mathbf{a}_i, \mathbf{a}_j\rangle| \\
&\leq \max_{i\in[N]} \max\{\sum_{j\in S} |\langle \mathbf{a}_i, \mathbf{a}_j\rangle|, S \subset [N], \mathrm{card}(S) = s, i \notin S\} = \mu_1(s) \\
&\leq s \cdot \max_{1\leq i\neq j\leq N} |\langle \mathbf{a}_i, \mathbf{a}_j\rangle| \\
&= s\mu_c.
\end{aligned}
$$

That is, $\mu_c \leq \mu_1(s) \leq s\mu_c$.

Bibliography

[1] Adcock, B., Antun, V. and Hansen, A. C. "Uniform recovery in infinite-dimensional compressed sensing and applications to structured binary sampling". In: *Appl. Comput. Harmon. Anal.* vol. 55 (2021), pp. 1–40.

[2] Adcock, B. and Hansen, A. C. *Compressive Imaging: Structure, Sampling, Learning*. Cambridge University Press, 2021.

[3] Adcock, B., Hansen, A. C., Poon, C. and Roman, B. "Breaking the coherence barrier: A new theory for compressed sensing". In: *Forum of Mathematics, Sigma* vol. 5 (2017), e4.

[4] Antun, V. *CIlib – A software library for compressive imaging*. https://github.com/vegarant/cilib. 2020.

[5] Antun, V. "Coherence estimates between Hadamard matrices and Daubechies wavelets". MA thesis. University of Oslo, 2016.

[6] Antun, V. and Ryan, Ø. "On the unification of schemes and software for wavelets on the interval". In: *Acta Appl. Math.* vol. 173, no. 7 (2021), pp. 1–25.

[7] Barzilai, J. and Borwein, J. M. "Two-point step size gradient methods". In: *IMA J. Numer. Anal.* vol. 8, no. 1 (1988), pp. 141–148.

[8] Beauchamp, K. G. *Walsh functions and their applications*. Techniques of Physics, No. 3. Academic Press [Harcourt Brace Jovanovich, Publishers], London-New York, 1975, pp. xiii+236.

[9] Berg, E. van den and Friedlander, M. P. *SPGL1: A solver for large-scale sparse reconstruction*. https://friedlander.io/spgl1. Dec. 2019.

[10] Berg, E. van den and Friedlander, M. P. "Probing the Pareto frontier for basis pursuit solutions". In: *SIAM J. Sci. Comput.* vol. 31, no. 2 (2008), pp. 890–912.

[11] Bertsekas, D. P. *Convex analysis and optimization*. With Angelia Nedić and Asuman E. Ozdaglar. Athena Scientific, Belmont, MA, 2003, pp. xvi+534.

[12] Bertsekas, D. P. *Nonlinear programming*. Second. Athena Scientific Optimization and Computation Series. Athena Scientific, Belmont, MA, 1999, pp. xiv+777.

[13] Birgin, E. G., Martínez, J. M. and Raydan, M. "Nonmonotone spectral projected gradient methods on convex sets". In: *SIAM J. Optim.* vol. 10, no. 4 (2000), pp. 1196–1211.

[14] Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

[15] Cohen, A., Daubechies, I. and Vial, P. "Wavelets on the Interval and Fast Wavelet Transforms". In: *Applied and Computational Harmonic Analysis* vol. 1, no. 1 (1993), pp. 54–81.

[16] Daubechies, I. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.

[17] Flores, I. "Reflected Number Systems". In: *IRE Transactions on Electronic Computers* vol. EC-5, no. 2 (1956), pp. 79–82.

[18] Foucart, S. and Rauhut, H. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013, pp. xviii+625.

[19] Gardner, M. *Knotted doughnuts and other mathematical entertainments*. W. H. Freeman and Company, New York, 1986, pp. xvi+278.

[20] Li, C. and Adcock, B. "Compressed sensing with local structure: Uniform recovery guarantees for the sparsity in levels class". In: *Applied and Computational Harmonic Analysis* vol. 46, no. 3 (2019), pp. 453–477.

[21] Mallat, S. *A wavelet tour of signal processing*. Third. The sparse way, With contributions from Gabriel Peyré. Elsevier/Academic Press, Amsterdam, 2009, pp. xxii+805.

[22] Nocedal, J. and Wright, S. J. *Numerical Optimization*. Second. New York, NY, USA: Springer, 2006.

[23] Rockafellar, R. T. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970, pp. xviii+451.

[24] Roman, B., Hansen, A. and Adcock, B. *On asymptotic structure in compressed sensing*. 2014. arXiv: 1406.4178 [math.FA].

[25] Ryan, Ø. *Linear Algebra, Signal Processing, and Wavelets - A Unified Approach. MATLAB Version*. Springer International Publishing, 2019.

[26] Traonmilin, Y. and Gribonval, R. "Stable recovery of low-dimensional cones in Hilbert spaces: One RIP to rule them all". In: *Applied and Computational Harmonic Analysis* vol. 45, no. 1 (2018), pp. 170–205.