Sara Mehraban

*Candidate*

Mathematics and Statistics

*Department*

This thesis is approved, and it is acceptable in quality and form for publication:

*Approved by the Thesis Committee:*

Maria Cristina Pereyra , Chairperson

Jens Lorenz

Pedro Embid

# Some Notes on Compressive Sensing

by

**Sara Mehraban**

B.S., Applied Mathematics, Ferdowsi University of Mashhad, 2007

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Mathematics

The University of New Mexico

Albuquerque, New Mexico

July, 2016

# Dedication

*I dedicate this thesis to my husband, Javad, for his remarkable patience and unwavering love and support.*

*– Sara Mehraban*

# Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Cristina Pereyra of the department of Mathematics and Statistics at the University of New Mexico, for her patience, motivation, and immense knowledge. The door to Prof. Pereyra's office was always open whenever I ran into a trouble spot or had a question about my thesis. She has been supportive since the day I started working. It was such a great experience working with her.

I would also like to acknowledge Professor Jens Lorenz and Professor Pedro Embid as the second readers of this thesis, and I am gratefully indebted to them for their very valuable comments on this thesis.

# Some Notes on Compressive Sensing

by

**Sara Mehraban**

ABSTRACT OF THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Mathematics

The University of New Mexico

Albuquerque, New Mexico

July, 2016

# Some Notes on Compressive Sensing

by

**Sara Mehraban**

B.S., Applied Mathematics, Ferdowsi University of Mashhad, 2007

Master, Mathematics, University of New Mexico, 2016

## Abstract

We are living in a world in which the growth rate of the data generated every year is almost exponential. A significant problem is how we can store this amount of data. *Compressive sensing* is giving us a clue about how we can reconstruct images and signals from frequency data, by having less samples compared to the conventional ways of data acquisition, which somehow helps us with the storage problem and gives us some other benefits that we try to present in this thesis.

The basic principle of Nyquist sampling theory has been one of the conventional ways in data acquisition and reconstruction signals and images. This so-called principle, introduces a minimum rate at which a signal can be sampled to be reconstructed without any errors. On the other side of this subject, *compressive sensing* introduces an efficient framework that enables us to derive exact reconstruction of a sparse signal from less measurements.

In this thesis we provide a few notes on mathematical insight related to this new theory which involves some proofs about the desired properties of the sampling ma-

trix and explain how we approach the problem of constructing this class of matrices and at last we move on to the recovery algorithm.

# Contents

# List of Figures

# Chapter 1

# Introduction to *compressive sensing*

## 1.1 Introduction

The advent of the digital century has brought us high resolution digital information that is generated by exploding digital components every year [1]. To be able to store or transmit this information, we need to rely on some compression algorithms that take advantage of the inherent redundancy of the information and translate it to a new recoverable format which requires lower space.[2].

In this chapter, we first focus on the transform coding that is based on the well known Nyquist theory. We will continue with some detailed discussions of the transform coding and the typical process that is generally followed to encode a signal. Then we will introduce some new exceptions to Nyquist sampling theory followed by *compressive sensing*, a new advanced coding scheme that is based on the sparsity nature of the signals. Finally we will explain the contribution of this thesis and will outline the rest of this report.

## 1.2 Transform coding

Transform coding is an efficient way to compress information, and can use either a lossless or a lossy coding technique [3]. The transform coding utilizes the knowledge of the application to determine the information to discard. For example in audio coding, since sensitivity of human ear falls within the range of (20 $Hz$ - 20 $kHz$) [4, 5], the high frequency content of the signal can safely be discarded, without a noticeable degradation in the quality of the signal at the receiver.

The block diagram in Fig. 1.1 demonstrates the general process of a transform coder. In the sampling stage, the input signal which is normally a continuous time signal is sampled at equi-spaced time intervals, generating a signal that is discrete in time. The sampling block is followed by a compression block. There are a wide variety of compression techniques which are mainly based on the redundancy of the information.

In the case of the compression of visual information, for example, the most popular techniques are based on either discrete cosine transform (DCT) or wavelets.

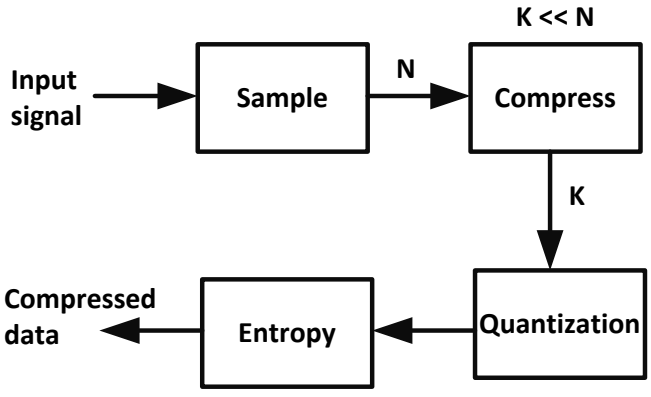


Figure 1.1: Block diagram of a general transform coder

In every frame of a video signal, there is a high probability that pixels have the same color and intensity of color as their neighbors. This feature which is called *intra-sample* redundancy is the base of image coding techniques like JPEG, video coding methods like H263, H264 and MPEG [6]. Another similar redundancy is the temporal redundancy, meaning that the pixel would have almost the same color and intensity information in consecutive frames, making the path to implement inter-frame redundancy reduction [7, 8].

Compressed information are fed to a quantization unit, aiming to be able to represent/transfer/store the compressed information using limited number of bits. The quantization unit is a critical stage in image/video/audio compression and as the analog values are quantized to some discrete levels the process is irreversible. This stage has the highest impact on the distortion of the reconstructed signal and is the main source of information loss in an encoder [6].

Figure 1.2 demonstrates the uniform quantization of a $sin(x)$ signal, when we have only 16 distinct levels of quantization. Non-uniform quantizer is the other quantization method that is selected for some signals with sensitivity which is higher at different ranges of the signal. For example for an audio signal, our ear is more sensitive to quantization error when the signal level is lower so it is preferred to use a kind of quantizer which introduces less noise to low values [6].

The final stage in a transform coder is the entropy coder. The entropy coder translates the quantized information to some new symbols that are based on the entropy (frequency) of different values at the output of the quantizer. In some coders the entropy coder is called the variable length coder (VLC), meaning that it assigns a symbol of variable length to the codes that are coming out of the quantizer. The size of the generated code is a function of the frequency for that specific symbol. The more frequent codes are assigned a smaller symbol, and the codes that are generated

lower are assigned a longer symbol. In this way the resultant code has a smaller size [6].

For the purpose of this thesis, since our target area is *compressive sensing*, the most important blocks to focus on, is the sampling and compression. In the next section we will discuss the sampling technique with more details.

## 1.3 Sampling Theorem

To detail the sampling process, we have demonstrated an example bandwidth limited signal in Fig. 1.3 (a) and its corresponding Fourier transform in 1.3 (b). As we will explain in the next section, it is of vital importance that the signal be band-limited, meaning that there is a frequency $B$ so that all the frequency components of the signal are lower or equal to $B$ and the signal doesn't have any component on the

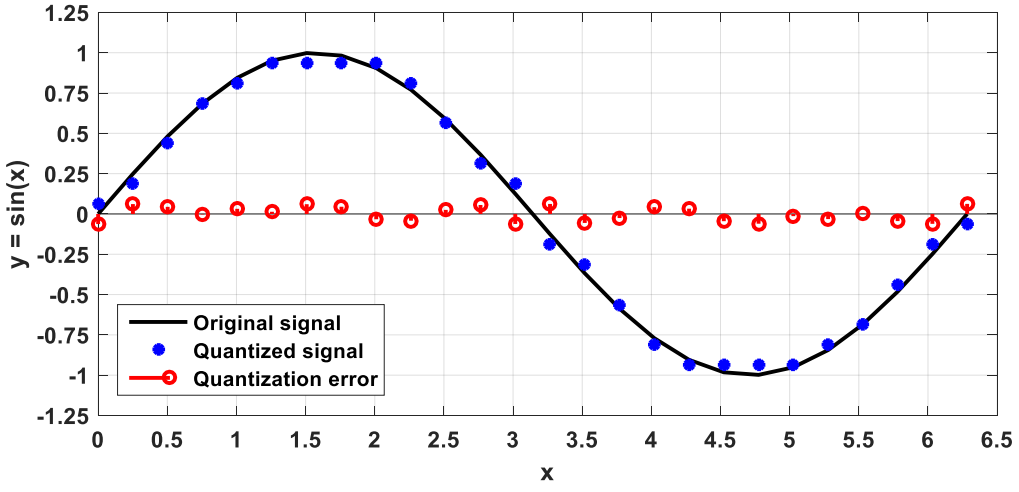


Figure 1.2: Demonstration of quantization of $sin(x)$ signal, as an example and the quantized signal when we have four bits (16 levels of uniform quantization). The quantization error which is the difference between the original signal and the quantized output which is not reversible, is also shown in red.

frequencies above $B$. In other words, a band-limited signal is a signal whose Fourier transform is compactly supported. [2].

We continue the sampling process of the signal first in the time domain and next in the frequency domain.

### 1.3.1 Sampling Theorem in the time domain

The sampling process of a signal $x(t)$ involves multiplication of the signal by a periodic sampling function. Figure 1.4 (a) depicts multiplication of the signal $x(t)$ by the sampling function $p(t)$, which is composed of periodic impulses with the period of $T_s = \frac{1}{f_s}$. The sampling fuction $p(t)$ shown in Fig. 1.4 (b) is a train of impulse signals. The resultant sampled signal $x_s(t)$ is shown on the right side of Fig. 1.4 (c). Equation 1.1 to 1.9 shows the theoretical concepts of the sampling of the signal $x(t)$ in both time and frequency domain [2, 9].

$$x_s(t) = p(t)x(t) \tag{1.1}$$

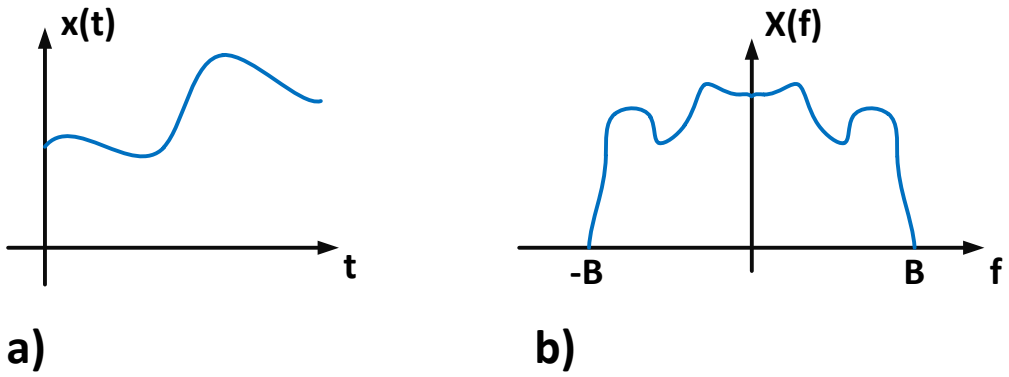


Figure 1.3: An example band-limited signal (a) time domain and (b) it's Fourier transform

Now we use a train of impulse functions as the function $p(t)$:

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s), \tag{1.2}$$

where impulse function is zero on all the horizontal axis with the exception of zero

$$\delta(t - t_0) = \begin{cases} 0 & t \neq t_0 \\ 1 & t = t_0 \end{cases} \tag{1.3}$$

And: Then we can write $x_s(t)$ as:

$$\begin{aligned} x_s(t) &= p(t)x(t) \\ &= x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \\ &= \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s) \\ &= \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \end{aligned} \tag{1.4}$$

Note that $x_s(t)$ can be considered as a discrete signal $x[n] = x_s(nT_s)$.

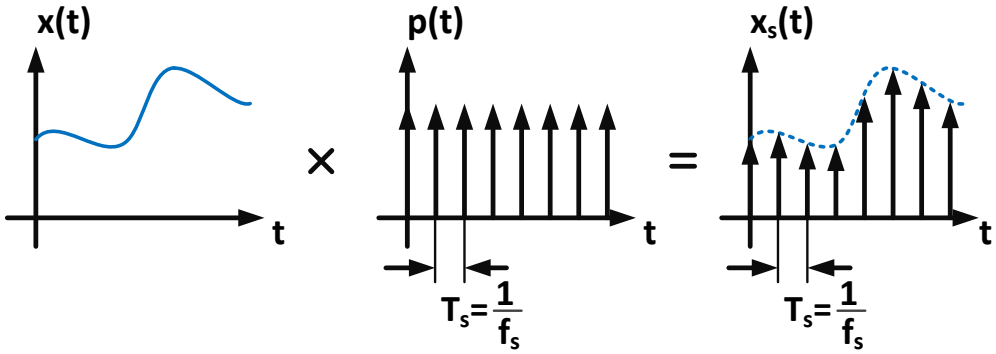


Figure 1.4: Demonstration of the sampling of continuous $x(t)$ using a periodic function $p(t)$. The resultant sampled function $x_s(t)$ which is discrete in time is shown on the right.

### 1.3.2 Frequency analysis of sampling theorem

In this part, we will review the frequency analysis of the sampling theorem and will show that the minimum sampling frequency possible for a band-limited signal is double of its highest frequency component [9].

$$x_s(t) = p(t)x(t) \qquad \overset{\mathcal{F.T.}}{\longleftrightarrow} \qquad X_s(i\omega) = \frac{1}{2\pi}X(i\omega) * P(i\omega) \tag{1.5}$$

Where:

$$X(i\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{+\infty} x(t)e^{-i\omega t}dt \tag{1.6}$$

And:

$$p(t) = \sum_{n=-\infty}^{+\infty}\big(\delta(t-nT_s)\big) \qquad \overset{\mathcal{F.T.}}{\longleftrightarrow} \qquad \frac{2\pi}{T_s}\sum_{k=-\infty}^{+\infty}\big(\delta(\omega-\frac{2\pi k}{T_s})\big) = P(i\omega) \tag{1.7}$$

So for the $X_s(\omega)$ we can write [2]:

$$\begin{aligned} X_s(i\omega) &= \frac{1}{2\pi}X(i\omega) * P(i\omega) \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(i\theta)P(i(\omega-\theta))d\theta \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(i\theta)\left\{\frac{2\pi}{T_s}\sum_{k=-\infty}^{+\infty}\big(\delta(\omega-\theta-\frac{2\pi k}{T_s})\big)\right\}d\theta \\ &= \frac{1}{T_s}\sum_{k=-\infty}^{+\infty}\Big[\int_{-\infty}^{\infty} X(i\theta)\delta(\omega-\theta-\frac{2\pi k}{T_s})d\theta\Big] \end{aligned} \tag{1.8}$$

Since we know for all $\theta \neq \omega - \frac{2\pi k}{T_s}$ that $\delta(\omega-\theta-\frac{2\pi k}{T_s})$ is zero.

$$\begin{aligned} X_s(i\omega) &= \frac{1}{T_s}\sum_{k=-\infty}^{+\infty}\Big[\int_{-\infty}^{\infty} X(i\theta)\delta(\omega-\theta-\frac{2\pi k}{T_s})d\theta\Big] \\ &= \frac{1}{T_s}\sum_{k=-\infty}^{+\infty}\Big[X(i(\omega-\frac{2\pi k}{T_s}))\int_{-\infty}^{\infty}\delta(\omega-\theta-\frac{2\pi k}{T_s})d\theta\Big] \\ &= \frac{1}{T_s}\sum_{k=-\infty}^{+\infty} X(i(\theta-\frac{2\pi k}{T_s})) \end{aligned} \tag{1.9}$$

The Equation 1.9, in short, says the multiplication of a continuous band limited signal to a period train of sampling impulses $\delta$ of period of $T_s = \frac{1}{f_s}$ is equal to infinitely replicating its Fourier transform $X(i\omega)$ and shifting of the replication by $f_s = \frac{2\pi}{T_s}$.

Figure 1.5 depicts the sampling of a continuous signal $x(t)$ in Fourier domain. The replication of the Fourier transform of the signal $x(t)$ in the frequency domain occurs because of the periodic sampling function $P(t) = \delta(t)$. The receiver/ reconstructor can easily get rid of this extra replication by use of a low pass filter (LPF) shown in red in Fig. 1.5.

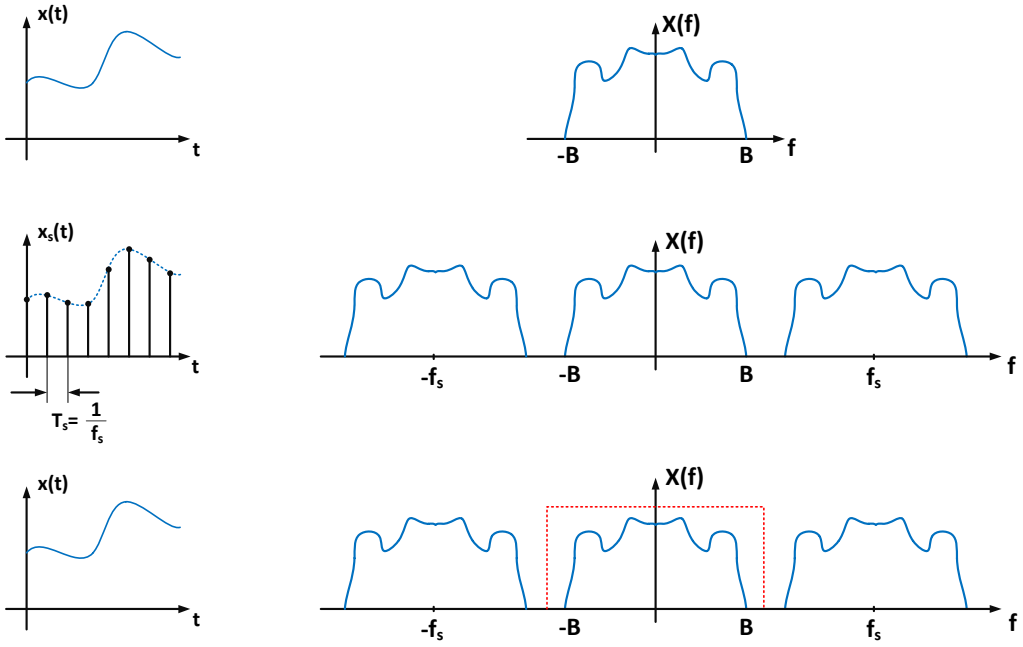


Figure 1.5: The sampling process of a continues signal $x(t)$, in Fourier domain. The red box in the last graph shows the low pass filter that is used at the reconstruction stage to get rid of the extra side replication of the Fourier transform of the signal.

## 1.4 Nyquist rate

To process analog signals, the sampling theorem is the fundamental key to the gate of digital signal processing [10]. The state of the art analog to digital converters (ADCs) sample their inputs at equi-spaced time intervals. There are many applications of this kind of sampling in radio frequency (RF) communications [11, 12].

The famous Nyquist rate that is named after electronic engineer Harry Nyquist has become a mathematical statement that has a high impact in industrial applications. By definitions, the Nyquist rate is the frequency, twice the highest frequency content of the signal. When working with band-limited (A band-limited signal is a signal whose Fourier transform has compact support) continuous (analog) signals, the Nyquist rate determines the lower limit to the sampling frequency at which the signal can be recovered without losing information [13].

In the example continuous signal that is shown in Fig. 1.4 and Fig. 1.5, when the signal is sampled with a sampling frequency of $f_s$, there are infinite number of functions that can fit to the sampled data, but as for all $f \geqslant f_s$ , $X(f) = 0$, there is a unique function that is band-limited to $B$ and fits to the sampled data. Based on the Nyquist criterion, in the definitions above, we must have $f_s \geqslant 2B$, otherwise the recovered signal from the same data would be different from the original signal [14, 15].

## 1.5 Aliasing

Figure 1.6 shows the situation when the sampling frequency is smaller then the rate offered by Nyquist ($f_s < 2B$). As a result, in the sampled data part of the Fourier transform of the signal is overlapped with the side replication of Fourier transform. This effect which is called *aliasing* destroys the signal and it will not be possible to

reconstruct the signal.

## 1.6 Sub-Nyquist sampling

As we explained in the previous section, sampling a signal at a slower rate than twice of its band limit will cause it being impossible to have a proper reconstruction, making the Nyquist criterion a must condition to avoid aliasing [16]. Although the Nyquist sampling theorem offers a solid upper limit to the criterion to the sampling and reconstruction of a band-limited signal, however under specific conditions sampling below the rate that is offered by the Nyquist rate, is still possible [17, 18].

Recent studies have proven [19] that the Nyquist criterion is not a necessary condition, if there is enough restriction applied to the desired signal. *Compressive sensing* (CS), as an example is a new field that offers significant decrease in the number of samples that are needed to be acquired, if the signal is sparse [20]. While Nyquist sampling theorem poses restrictions based on the bandwidth of the signal, by taking the largest frequency component of the band-limited signal as the base,

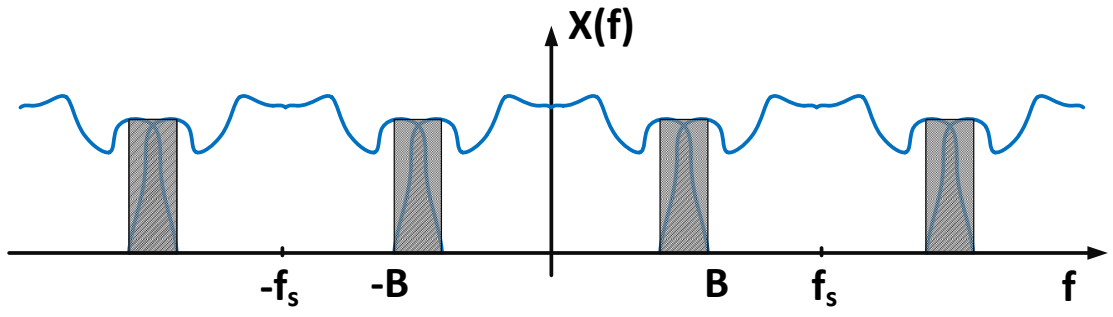


Figure 1.6: Sampling of a signal at a sampling frequency below the rate offered by Nyquist caused distortion of the signal and since part of the Fourier transform is lost, will make the reconstruction impossible.

compressive sensing offers a new parameter named effective bandwidth which can be much lower than the well-studied Nyquist rate.

Another benefit of *compressive sensing* is the fact that it generates compressed information at the same place that it samples the information and removes the need for subsequent blocks to comprehend the sampled data. When it comes to the hardware implementation, this would lead to much simpler systems. Figure 1.7 shows the block diagram of a *compressive sensing* based encoder. One could see that in the recent advancements the coding has eliminated the need for the sampling-coding overhead, by taking already compressed information.

And finally, what is making *compressed sensing* remarkable is that, using this technique, sensors can gather signal information very briefly, without trying to comprehend them.

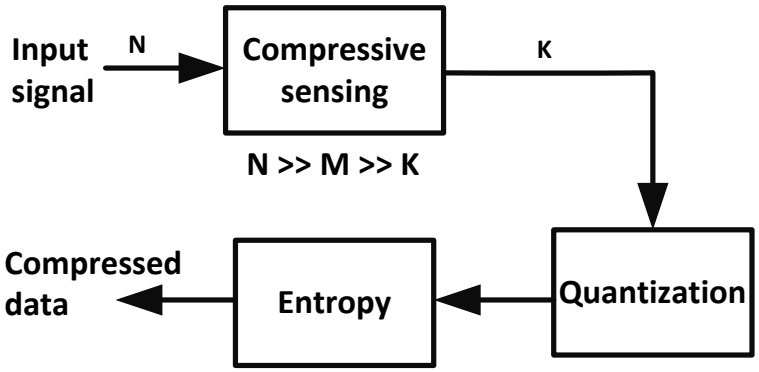


Figure 1.7: Block diagram of an encoder which is based on *compressive sensing*.

## 1.7 Sampling in compressive sensing

Although the sampling rate that is offered by Nyquist, works for every band-limited signal, it still is so high that is not affordable by means of storage or transmission. To address these challenges, we depend on some compression algorithms that find and remove redundant information by transforming it to a new basis that the signal has a more concise form. The new basis is selected with the goal of providing a sparse or compressible representation for the signal [21]. Note that a compressible signal is a signal which is sparse in at least one domain.

According to the *compressed sensing* theory, the encoding of a given signal does not involve a complex process. Linear sampling/measurement process gets done by simple matrix multiplication.

## 1.8 Conclusion

In this chapter we had a brief overview on the concept of the well-known Nyquist sampling theory and also the specification of a typical transform coder is explained. We also have introduced a recent coding technique which offers sampling of the signal at rates that can be much lower than the double of its bandwidth. In the next chapter we will review the properties of the sensing matrix. We will explain the problem of the sensing matrix construction in Chapter 3.

# Chapter 2

# Sensing Matrix Properties

## 2.1 Introduction

As highlighted in the previous chapter, *compressive sensing*, also known as *compressive sampling* or *sparse sampling* is a technique in signal processing for efficiently acquiring and reconstructing a sparse signal. Sparse representation in *compressed sensing* establishes a classified mathematical pattern for studying high dimensional sparse signals and ways to reconstruct them, introducing a large collection of efficient algorithms. In a recent research by Candes et. al. [22] it has been shown that using *compressed sensing* a sparse representation can be reconstructed exactly from small set of linear measurements. This result proves the possibility of reconstruction of a sparse signal by taking much lower number of measurements.

Additionally, *compressed sensing* is designed to measure sparse signals directly in a compressed form. Most signals of interest are only "approximately sparse" in contrast to "ideally sparse" signals which means, despite the fact that the signal contains only a small fraction of large entries, the other entries are not strictly equal to zero, however they are very close to zero.

A formal mechanism to sample the signal is at the heart of compressive sensing. The measurement matrix forms a standard mechanism in which the information about a signal $x$ is obtained. In other words the measurement matrix $A$ samples a high dimensional signal $x \in \mathbb{R}^n$ using limited linear measurements and plays a central role in reducing the dimension of the signal $x$ such that:

$$y = Ax \quad .$$

Where $y \in \mathbb{R}^m$ is the dimensionally reduced form of the input vector $x$. And $A$ is an $m \times n$ matrix which is the dimensionality reduction matrix that maps vector $x$ to vector $y$ noticing that $m \ll n$. For instance if $x$ is a sinusoid waveform, then $y$ may be the lower dimensional vector of its Fourier coefficients.

In this chapter we will specify some important properties corresponding to the class of measurement/sensing matrices. We restrict our focus to discrete signals in $\mathbb{R}^n$, however one could extend the same *compressive sensing* method to continuous signals as well [23].
The material in this chapter and the next is mostly based on the survey by Davenport et. al. [24].

## 2.2 Sparse representation and exact recovery

To start, we introduce a significant concept in the field of *compressive sensing*, which is the notion of sparsity.

### 2.2.1 Sparse signals

As explained in the previous chapter, *compressive sensing* works on a specific class of signals named sparse signals. A *sparse signal* is a signal which can be

represented as a linear combination of relatively few base elements in a basis. In other words a vector at which there is a large number of zero elements is considered to be a *sparse vector*.

Sparse signal representation has been proven to be a drastically powerful tool in acquiring, representing and compressing high-dimensional signals. The fact that important classes of signals like images or audio signals, have sparse representations in some fixed basis (i.e. time or frequency), has been very important in proving this matter. Sparse representation has been found as a very attractive subject in fields of signal processing, image processing, pattern recognition and computer vision.

In fact many signals are naturally sparse in the sense that they have concise sparse representation in a proper basis. In mathematics, a signal $x$ is *k-sparse* if the number of its non-zero entries i.e. $\|x\|_0$, is less than or equal to k. Set of all *k-sparse* signals can be denoted as:

$$\Sigma_k = \{x : \|x\|_0 \leqslant k\} \tag{2.1}$$

Note that however we are using the norm symbol to count the nonzero entries of a vector, we should notice that $\|x\|_0$ is not a norm. Because $\|\alpha x\|_0 \neq |\alpha| \|x\|_0$.

Let's see how k-sparse signals are related with 2k-sparse signals.

**Lemma 2.1.** *For any non-zero 2k-sparse signal $h$, there are two different k-sparse signals, $x$ and $x'$, such that $h$ can be represented as the difference of $x$ and $x'$. In other words for any non-zero $h$, $h \in \Sigma_{2k}$, if and only if $h = x - x'$ for some $x, x' \in \Sigma_k$ with $x \neq x'$.*

*Proof.*
($\Rightarrow$)
Assume that $h = (h_1, h_2, ...., h_n)$ is a vector in $\Sigma_{2k}$ where every $h_i \in \mathbb{R}$ represents an entry of $h$. Let the set $S = \left\{i \in \{1, 2, ..., n\} : h_i \neq 0\right\}$, where the set $S$ represents the

positions of the nonzero entries of $h$. Since $h \neq 0$ there is at least one non-zero entry in it, and since $h \in \Sigma_{2k}$, $h$ has at most $2k$ non-zero entries. So $S \neq \varnothing$. Notice that $1 \leqslant |S| \leqslant 2k$ ($|S|$ denotes the cardinality of $S$).

Partition the set S into two subsets $S_1$ and $S_2$ so that their cardinalities are at most k. Hence the subsets $S_1$ and $S_2$ are such that:

$$S_1 \cup S_2 = S,\ S_1 \cap S_2 = \varnothing, \quad \textit{and} \quad |S_1| \leqslant k, |S_2| \leqslant k. \tag{2.2}$$

Necessarily $S_1 \neq \varnothing$ or $S_2 \neq \varnothing$, because $S \neq \varnothing$.

Let $x = (x_1, x_2, ...., x_n)$, be a vector in $\Sigma_k$ where:

$$x_i = \begin{cases} h_i & i \in S_1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

and $x' = (x'_1, x'_2, ..., x'_n)$, be a vector in $\Sigma_k$ where

$$x'_i = \begin{cases} -h_i & i \in S_2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

so by construction $h = x - x'$, with $x, x' \in \Sigma_k$ and $x \neq x'$.

($\Leftarrow$)

We are assuming $x \neq x'$ and $x, x' \in \Sigma_k$. Let $h = x - x'$ so $h = (h_1, h_2, \cdots, h_n)$ where $h_i = x_i - x'_i$. We want to show that $h \in \Sigma_{2k}$. Consider the sets $K$, $K_1$ and $K_2$ which represent the positions of the non-zero entries of $h$, $x$ and $x'$ respectively:

$$K = \Big\{ i \in \{1, 2, \cdots, n\} : h_i \neq 0 \Big\},$$

$$K_1 = \Big\{ i \in \{1, 2, \cdots, n\} : x_i \neq 0 \Big\},$$

and

$$K_2 = \Big\{ i \in \{1, 2, \cdots, n\} : x'_i \neq 0 \Big\}.$$

Note that since $x', x \in \Sigma_k$, the cardinality of $K_1$ and $K_2$ is less than or equal to $k$ or in other words $|K_1| \leqslant k$ and $|K_2| \leqslant k$. We claim that $K \subseteq K_1 \cup K_2$, then $|K| \leqslant |K_1| + |K_2| \leqslant k + k = 2k$, which implies that $h \in \Sigma_{2k}$.

**Proving the claim:** Suppose $i \in K$, then $h_i \neq 0$, since $h_i = x_i - x'_i$ either $x_i \neq 0$ or $x'_i \neq 0$, otherwise $h_i = 0$. This means that $i \in K_1$ or $i \in K_2$, therefore $i \in K_1 \cup K_2$. So $K \subseteq K_1 \cup K_2$ and since $K_1 \cup K_2$ has at most $2k$ non-zero entries, we conclude that:

$$h \in \Sigma_{2k} \quad .$$

□

### 2.2.2 Null space condition

One important specification of the sensing matrix that we use to sample the data, is its null space. Considering matrix $A$, the null space of $A$ is denoted by $\mathcal{N}(A)$ and is defined by:

$$\mathcal{N}(A) = \{z \in \mathbb{R}^n \mid Az = 0\}. \tag{2.5}$$

In order to be able to recover all k-sparse signals $x$ from $Ax$, $A$ must uniquely represent all $x \in \Sigma_k$ meaning:

$$\begin{aligned} &\forall x,\ x' \in \Sigma_k \\ &\qquad \text{if:} \quad x \neq x' \\ &\qquad \text{then} \quad Ax \neq Ax'. \end{aligned} \tag{2.6}$$

Otherwise it is not possible to distinguish $x$ from $x'$ based on $y$.
The following theorem specifies a property that makes a matrix represent all $k$ – *sparse* vectors, uniquely.

**Theorem 2.1.** *A sensing matrix* A, *uniquely represents all* $x \in \Sigma_k$ *if and only if* $\mathcal{N}(A)$ *contains no non-zero vector in* $\Sigma_{2k}$.

*Proof.* We have to show $p \iff q$ where:

p: $A$ uniquely represents all $x \in \Sigma_k$.

q: $\mathcal{N}(A)$ contains no non-zero vector in $\Sigma_{2k}$ .

$\Rightarrow$ We try to show the contrapositive of the statement $p \Rightarrow q$: $\sim q \Rightarrow \sim p$
So we are assuming there is a non-zero vector $h$ in $\Sigma_{2k}$ which is also in the null-space. Since by Lemma 2.1, $h$ can be written as a difference of two vectors in $\Sigma_k$.

$$\begin{cases} h \neq 0 \\ h \in \Sigma_{2k} \\ h \in \mathcal{N}(A) \end{cases} \Rightarrow h = (x - x') \quad \text{where} \begin{cases} x, x' \in \Sigma_k \\ x \neq x' \end{cases} . \tag{2.7}$$

Noticing $h \in \mathcal{N}(A)$ then $Ah = 0$, so $A(x - x') = 0$. And since $A$ is a linear transformation $Ax = Ax'$. This is $\sim p$

$\Leftarrow$ To show $q \Rightarrow p$ again we prove the contrapositive of the statement $\sim p \Rightarrow \sim q$. So there are $x, x' \in \Sigma_k$ and $x \neq x'$ such that $Ax = Ax'$ this implies $A(x - x') = 0$. By Lemma 2.1 $h = x - x'$ is a vector in $\Sigma_{2k}$ since $x, x' \in \Sigma_k$ and also $h \neq 0$ since $x \neq x'$. So $h$ is a nonzero vector in $\Sigma_{2k}$ that is in $\mathcal{N}(A)$. This is $\sim q$.

So the statement $p \iff q$ holds. □

## 2.3 Spark and coherence

In this section we aim to explain two important matrix properties in the field of *compressed sensing*, which together with some other properties explained in next chapter, will provide the exact recovery of all $k - sparse$ vectors.

### 2.3.1 Spark of a matrix

Before we introduce the notion of the *spark* of a matrix, it is wise to start with a more familiar concept about matrices in linear algebra:

**Rank of a matrix (dimension of the column space)**

The rank of a matrix $A$ is the dimension of the vector space spanned by its columns. So the maximum number of linearly independent column (row) vectors is the rank of the matrix. If you have the *row-echelon* form of a matrix, the number of non-zero rows is the rank.

In the following examples we have evaluated the *row-echelon form (REF)* of the matrix $A$ to determine the $rank(A)$:

**Example 2.1.**

$$A = \begin{pmatrix} 2 & -1 & 3 \\ 1 & 0 & 1 \\ 0 & 2 & -1 \\ 1 & 1 & 4 \end{pmatrix} \rightarrow REF \rightarrow \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

*The rank is the number of non-zero rows in REF form, so.* $\Rightarrow$ $rank(A) = 3$.

**Full rank matrices:** When all of the column vectors in a matrix are linearly independent, the matrix is said to be *full-rank*. See the following example of a *full-rank* matrix:

**Example 2.2.**

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}$$

*A is a full rank matrix, because all the columns are linearly independent. If we compute the row-echelon form of A we get:*

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{pmatrix}$$

*We have 3 non-zero rows so the rank is 3 and the matrix A is considered to be a full-rank matrix.*

**Definition 2.1.** *The spark of a matrix is the smallest number of columns of A, that are linearly dependent.*

**Remark:** *If we have a zero column in A, then* $spark(A) = 1$. *And that is because a zero column is considered to be linearly dependent by itself. In general any set of vectors, containing the vector 0 is linearly dependent, because for any* $c \neq 0, c\,0 = 0$.

*If there is no zero column in A, then*

$$2 \leqslant spark(A) \leqslant rank(A) + 1 \tag{2.8}$$

*because* $rank(A)$ *is the maximum number of linearly independent columns, so if we have* $rank(A) + 1$ *columns, they will always be* linearly dependent.

The following examples will illustrate the concept of the spark of a matrix in different cases:

**Example 2.3.**

*Consider the given matrix A :*

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & -1 & +1 \end{pmatrix}$$

*Any two columns are linearly independent, for instance if we consider the first two columns which are linearly independent, as soon as we add the third or the fourth*

*column, we won't have the linear independence anymore.*

$$\text{The first and the second columns: } \underbrace{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix}}_{\text{Linearly independent}} \Rightarrow \text{The first three columns: } \underbrace{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix}}_{\text{Linearly dependent}}$$

$$\text{The second and the third columns: } \underbrace{\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix}}_{\text{Linearly independent}} \Rightarrow \text{The second, the third and the first columns: } \underbrace{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix}}_{\text{Linearly dependent}}$$

*Since the maximum number of linearly independent columns is 2, the rank is* 2*, and since the smallest number of linearly dependent columns is* 3

$$\Rightarrow \quad spark(A) = 3 = rank(A) + 1$$

**Example 2.4.**

*Consider the given matrix* $B$:

$$B = \begin{pmatrix}1 & 0 & 1 & -1\\ 0 & 1 & 1 & 0\end{pmatrix}$$

*Since the maximum number of linearly independent columns is* 2*, the rank is two, like the first two columns. However we also can find pairs of columns, for example:* $\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix}$*, which are linearly dependent. So the* $spark(A)$ *cannot be 3, like the previous example, since there are 2 linearly dependent columns in* $A$*, and since we have no zero columns each column is linearly independent so the* $spark(A) = 2$ *in this case.*

**Example 2.5.**

$$A = \begin{pmatrix}1 & 0 & 0 & -1\\ 0 & 1 & 0 & 0\end{pmatrix}$$

*Since we have a zero column and a zero column is linearly dependent, then we get* $spark(A) = 1$. *Despite the fact that* $rank(A) = 2$ *also in this example.*

**To reiterate:**

Assuming we have no zero columns in matrix $A$:
$2 \leqslant spark(A) \leqslant rank(A) + 1$
If *spark* is *1*, we have one zero column at least.

The definition of the spark of a matrix allows us to pose the following theorem to check whether or not a sensing matrix is representing every k-sparse signal, uniquely.

**Theorem 2.2.** *For any vector* $y \in \mathbb{R}^m$, *there is at most one signal* $x \in \Sigma_k$ *such that*: $Ax = y$ *if and only if* $spark(A) > 2k$ .

*Proof.* We are showing $p \iff q$ where:
p: A uniquely represents all k-sparse vectors.
q: $Spark(A) > 2k$.

$\Rightarrow$ To prove $p \Rightarrow q$ we try to show the contrapositive of the statement, $\sim q \Rightarrow \sim p$:
$\sim p$ : There is more than one $x \in \Sigma_k$ such that $Ax = y$
$\sim q : Spark(A) \leqslant 2k$

The proposition $\sim q$ implies that any set of up to $(2k - 1)$ columns or, more generally, any set of up to $spark(A) - 1$ columns of $A$ is *linearly independent*. There is a permutation matrix $P$ such that $P$ separates linearly independent columns from linearly dependent ones: Since $spark(A) \leqslant 2k$, there is a set of $2k$ columns that are linearly dependent. We use permutation $P$ to put them as the first $2k$ columns.

Consider:

$$\mathrm{AP} = \left( \underbrace{\begin{array}{c|c|c|c} A_1 & A_2 & \cdots & A_{2k} \end{array}}_{\text{linearly dependent}} \middle| \begin{array}{c|c|c} A_{2k+1} & \cdots & A_n \end{array} \right).$$

Since $\{A_1, A_2, \cdots, A_{2k}\}$ are linearly dependent there are constants $c_1$, $c_2$, $\cdots$, $c_{2k}$ not all equal to zero such that:

$$c_1 A_1 + c_2 A_2 + \cdots + c_{2k} A_{2k} = \overrightarrow{0}. \tag{2.9}$$

Let $C = (c_1, c_2 \cdots, c_{2k}, 0 \cdots, 0)^t$, then the equation 2.9 is equal to $APC = 0$. Let $PC = h$, since C has at most $2k$ non-zero entries, so does $h$, because $P$ is just a permutation. Also $h \in \mathcal{N}(A)$, because $Ah = APC = 0$. So by Theorem 2.1, $A$ will not represent all $\Sigma_{2k}$ uniquely. This is $\sim p$.

$\Leftarrow$ We are to prove $q \Rightarrow p$. By contradiction assume that, there are $x, x' \in \Sigma_k$ and $x \neq x'$ such that $Ax = Ax'$, so $A(x - x') = 0$. By Lemma 2.1 if we set $x - x' = h$ then $h \neq 0$, also $h \in \Sigma_{2k}$. Since $spark(A) > 2k$ all sets of up to $2k$ columns are linearly independent. Then since $Ah = 0$, if $A_1$, $A_2$, $\cdots$, $A_n$ denote the columns of $A$ (without permutation), then:

$$0 = Ah = h_1 A_1 + h_2 A_2 + \cdots + h_n A_n, \tag{2.10}$$

Where $h = (h_1, h_2, \cdots, h_n)^t$, since $h \in \Sigma_{2k}$ we have at most $2k$ non-zero entries, Consider the non-zero entries labeled by, say $\{n_1, n_2, \cdots, n_s\}$ where $1 \leqslant n_1 < n_2 < \cdots < n_s \leqslant n$ and $1 \leqslant s \leqslant 2k$ thus (2.10) becomes

$$h_{n_1} A_{n_1} + h_{n_2} A_{n_2} + \cdots h_{n_s} A_{n_s} = 0. \tag{2.11}$$

Knowing that $s \leqslant 2k$ and since the $spark(A) > 2k$, we conclude that the columns $A_{n_1}, A_{n_2}, \cdots, A_{n_s}$ are linearly independent, so $h_{n_1} = h_{n_2} = \cdots = h_{n_s} = 0$. Therefore we end up with $h = 0$ which contradicts with $h$ being non-zero.

□

### 2.3.2 Coherence of a matrix

Coherence of a matrix is another property of matrices that we need for the exact recovery of k-sparse signals:

**Definition 2.2.** *The coherence of a matrix A, $\mu(A)$:*

$$\mu(A) = \max_{1 \leqslant i < j \leqslant n} \frac{|\langle a_i, a_j \rangle_{\mathbb{R}^m}|}{\|a_i\|_2 \|a_j\|_2}, \quad i \neq j$$

Is the largest absolute value of the inner product of any two normalized columns of A, here we are denoting by $a_i \in \mathbb{R}^m$ the $i$th column of $A$.
Note that $\langle X, Y \rangle_{\mathbb{R}^m} = X.Y$ denotes the inner product, or dot product of two vectors in $\mathbb{R}^m$, where $X = (x_1, x_2, \cdots, x_m)$ and $Y = (y_1, y_2, \cdots, y_m)$, defined as

$$\langle X, Y \rangle_{\mathbb{R}^m} = X.Y = \sum_{i=1}^{m} x_i y_i.$$

And $\|X\|_2 = \sqrt{\langle X, X \rangle}$ denotes the $\ell^2 - norm$ in $\mathbb{R}^n$. Also, notice that $\mu(A) \leqslant 1$ by Cauchy-Schwarz inequality: $|A.B| \leqslant \|A\|_2 \|B\|_2$ and the equality holds when the vectors $A$ *and* $B$ are parallel or in other words $A = \lambda B$ for $\lambda \in \mathbb{R}$
**Remark**: If all pairs of columns are orthogonal then $\mu(A) = 0$, and if we have a pair of identical or proportional columns (parallel vectors) then $\mu(A) = 1$.

### 2.3.3 Spark versus coherence

Coherence and spark of a matrix are somehow related. In order to prove this we need Gershgorin's Disk Theorem which is stated and proved below.

**Theorem 2.3.** *The eigenvalues of an $n \times n$ matrix $A$, lie in the union of $n$ discs $d_i = d_i(c_i, r_i)$, $1 \leqslant i \leqslant n$, centered at $c_i = a_{ii}$ and with radius $r_i = \sum_{i \neq j} |a_{ij}|$.*

*Proof.* let $\lambda$ be an eigenvalue of $A$ and $x$ be a corresponding eigenvector. And let $x_i$ be the the largest entry of x in absolute value. Since $x$ is an eigenvector, $Ax = \lambda x$. In other words $\sum_j a_{ij}x_j = \lambda x_i$ for $i = 1, 2, \cdots, n$, pulling out $a_{ii}$:

$$\sum_{j:\, i \neq j} a_{ij}x_j = \lambda x_i - a_{ii}x_i$$

Now we divide both sides by $x_i$, and also we take absolute value from both sides of the equation, by triangle inequality:

$$|\lambda - a_{ii}| = \left| \frac{\sum_{i \neq j} a_{ij}x_j}{x_i} \right| \leqslant \sum_{i \neq j} \frac{|a_{ij}x_j|}{|x_i|} \leqslant \sum_{i \neq j} |a_{ij}| = r_i \ .$$

Since we chose $x_i$ to be the largest entry of the vector $x$, in absolute value, for $i \neq j$, $\left|\frac{x_j}{x_i}\right| \leqslant 1$, so the last inequality is valid. □

Recall that $\Lambda \subset \{1, 2, \ldots, n\}$ with $|\Lambda| < n$, is a subset of indices and let $\Lambda^c \subset \{1, 2, \ldots, n\} \backslash \Lambda$. For a matrix $A$, $A_\Lambda$ means the $m \times p$ matrix derived by deleting columns of $A$ indexed in $\Lambda^c$.

**Remark**: Scaling by a non-zero parameter or in particular normalizing columns of a matrix, does not change the columns linearity so the spark and also the rank of a matrix do not depend on scaling. If we start with $\vec{v_1}, \vec{v_2}, \cdots, \vec{v_n}$ are linearly independent if for any $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ such that:

$$\alpha_1\vec{v_1} + \alpha_2\vec{v_2}\cdots + \alpha_n\vec{v_n} = 0$$

Then

$$\alpha_1 = \alpha_2 = \cdots\alpha_n = 0 \quad .$$

In case of normalizing we have that if $\beta_1 \frac{\vec{v_1}}{\|v_1\|_2} + \beta_2 \frac{\vec{v_2}}{\|v_2\|_2} + \cdots + \beta_n \frac{\vec{v_n}}{\|v_n\|_2}$, then $\frac{\beta_1}{\|v_1\|_2}\vec{v_1} + \frac{\beta_2}{\|v_2\|_2}\vec{v_2}\cdots + \frac{\beta_n}{\|v_n\|_2}\vec{v_n} = 0$, since $v_1, v_2, \cdots, v_n$ are linearly independent, $\alpha_j = \frac{\beta_j}{\|v_j\|} = 0$ for all

j. Recall that linearly independent vectors can not be equal to zero vector so $\|v_j\|_2 > 0$.

If we apply the Theorem 2.3 to the Gram matrix $G \coloneqq A_\Lambda^t A_\Lambda$, we will have the following lemma which connects the concepts of spark and the coherence of a matrix:

**Lemma 2.2.** *For any matrix A with non-zero columns,*

$$spark(A) \geqslant 1 + \frac{1}{\mu(A)}.$$

*Proof.* Without loss of generality we assume that the columns of A are normalized, or in other words, A has unit-norm columns. Note that, scaling won't change the $spark(A)$. See remark 2.3.3.

Consider $\Lambda = \{1, 2, \cdots, n\}$ as a set of indices with $|\Lambda| = p$. Also consider the restricted $p \times p$ Gram Matrix $G \coloneqq A_\Lambda^t A_\Lambda$. Where G is symmetric $(G^t = (A_\Lambda^t A_\Lambda)^t = A_\Lambda^t A_\Lambda = G)$ with entries: $g_{ij} = \langle a_{n_i}, a_{n_j} \rangle$ so $g_{ii} = 1$ for $1 \leqslant i \leqslant p$. Also $|g_{ij}| \leqslant \mu(A)$ for $1 \leqslant i, j \leqslant p,\ i \neq j$ .

If we choose $p < 1 + 1/\mu(A)$ then equivalently $(p-1)\mu(A) < 1$. But since $|g_{ij}| \leqslant \mu(A)$, we get

$$\sum_{i \neq j} |g_{ij}| \leqslant (p-1)\mu(A) < 1 \quad .$$

Now by Gershgoring theorem since $\sum_{i \neq j} |g_{ij}| < 1 = |g_{ii}|$ which means the matrix G is diagonally dominant, G also is a positive definite matrix so all of its columns are linearly independent, and so are the columns of $A_\Lambda$, as we show:

$A_\Lambda^t A_\Lambda = G$ is positive definite $\iff \langle Gx, x \rangle > 0, x \neq 0, thus \quad Gx = 0 \iff x = 0$. For $x \in \mathbb{R}^p$

So null space of G only contains the zero vector. And we conclude that the columns of G must be linearly independent

Suppose $c_1 a_1 + c_2 a_2 + \cdots + c_p a_p = 0$, where $a_1, a_2, \cdots, a_p$ are the columns of $A_\Lambda$. Then

$\forall i = 1, ..., p: \quad c_1\langle a_1, a_i\rangle + \cdots + c_p\langle a_p, a_i\rangle = 0$ so

$$c_1 \begin{pmatrix} \langle a_1, a_1\rangle \\ \langle a_1, a_2\rangle \\ \vdots \\ \langle a_1, a_p\rangle \end{pmatrix} + \cdots + c_p \begin{pmatrix} \langle a_p, a_1\rangle \\ \langle a_p, a_2\rangle \\ \vdots \\ \langle a_p, a_p\rangle \end{pmatrix} = 0.$$

But the columns of G are linearly independent, so $c_1 = c_2 = \cdots = c_p = 0$. Now we conclude that columns of $A_\Lambda$ are linearly independent as well. So we have shown that given $\Lambda$ with $|\Lambda| = p$ and $p < 1 + 1/\mu(A)$ then the columns of $A_\Lambda$ are linearly independent. So in conclusion $spark(A) > p$. Let $p_0 = max\{p \in \mathbb{N} : p < 1 + 1/\mu(A)\}$ we are going to have 2 cases depending on whether $1 + 1/\mu(A)$ is a natural number or not. Recall $\lfloor x \rfloor$, the integer part of $x$, is the smallest integer less than or equal to $x$.

So

$$p_0 = \begin{cases} \lfloor 1 + 1/\mu(A) \rfloor & if \quad \lfloor 1 + 1/\mu(A) \rfloor < 1 + 1/\mu(A) \\ 1/\mu(A) & if \quad 1 + 1/\mu(A) = \lfloor 1 + 1/\mu(A) \rfloor \end{cases} \tag{2.12}$$

Note that by definition $p_0 < 1 + 1/\mu(A)$, we have shown that $spark(A) > p_0$.

Case I. $(1 + 1/\mu(A) \notin \mathbb{N})$:

$$p_0 = \lfloor 1 + 1/\mu(A) \rfloor < 1 + 1/\mu(A) < \lfloor 1 + 1/\mu(A) \rfloor + 1 = p_0 + 1 \leqslant spark(A)$$

So $1 + 1/\mu(A) < spark(A)$

Case II. $(1 + 1/\mu(A) \in \mathbb{N})$:

$p_0 = 1/\mu(A)$ so $\mu(A) = 1/k$ where $k \in \mathbb{N}$, knowing that $p_0 < spark(A)$. Then, $p_0 + 1 = 1/\mu(A) + 1 \leqslant spark(A)$

So in both cases we get $1 + 1/\mu(A) \leqslant spark(A)$. □

As a corollary of Lemma 2.2 and Theorem 2.2 we conclude that:

**Corollary 2.1.** *If* $1 + 1/\mu(A) > 2k$ *then for each* $y \in \mathbb{R}^n$ *there is at most one signal* $x \in \sigma_k$ *such that* $Ax = y$.

# Chapter 3

# Null space and restricted isometry properties

While we are dealing with ideally or exactly sparse vectors, the spark gives us a full characterization of when sparse recovery is possible. However, when dealing with approximately sparse vectors, we must consider a more restrictive condition on the null-space $\mathcal{N}(A)$ [25].

As mentioned before, most signals of interest are only "approximately sparse" rather than "ideally sparse". When a signal contains only a small fraction of large entries, and the other entries are not strictly equal to zero, but are only close to zero, the signal is considered to be an *approximately sparse signal*.

Note that the approximately sparse signals are generalizations of the exactly sparse signals when some of the zero entries of the exactly sparse signals are set to some small magnitude numbers.
The material in this chapter and the previous is mostly based on the survey by Davenport et. al. [24].

## 3.1 Null space property (NSP)

Suppose $\Lambda \subset \{1, 2, \ldots, n\}$ is a subset of indices and let $\Lambda^c \subset \{1, 2, \ldots, n\} \backslash \Lambda$, $x_\Lambda$ means the length $n$ vector derived by setting entries of $x$ indexed by $\Lambda^c$ to zero.

**Definition 3.1.** *Matrix $A$ satisfies the null space property (NSP) of order $k$, if there exists a non-negative constant $C$ such that for all $h \in \mathcal{N}(A)$,*

$$\|h_\Lambda\|_2 \leqslant C\frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}} \quad \textit{for all} \quad \Lambda \subset \{1, 2, \cdots, n\} \textit{ with } |\Lambda| \leqslant k. \tag{3.1}$$

Where the $\ell^1$ norm and the $\ell^2$ norm are defined by:

$$\|h\|_2 = (\sum_{i=1}^{n} |h_i|^2)^{1/2} \textit{ and } \|h\|_1 = \sum_{i=1}^{n} |h_i| \quad \textit{for } h = (h_1, h_2, \cdots, h_n).$$

If a vector $h$ is k-sparse, then there exists a $\Lambda$ with $|\Lambda| \leqslant k$ such that $h_{\Lambda^c} = 0$ and $\|h_{\Lambda^c}\|_1 = 0$, and therefore (3.1) implies that $h_\Lambda = 0$ also. Hence if $A$ satisfies the $NSP$ of order $k$, then the only k-sparse vector in $\mathcal{N}(A)$ is $h = 0$.

**Definition 3.2.** *(Informal definition): A recovery method is an algorithm by which a sampled low-dimensional signal can be retrieved to a human readable format after being transfered to a low-dimensional form by the encoder.*

Note that not all recovery method are necessarily linear.

Let $\triangle : \mathbb{R}^m \to \mathbb{R}^n$ represent our recovery method, then the following property

$$\begin{gathered} \|\triangle(Ax) - x\|_2 \leqslant C\frac{\sigma_k(x)_1}{\sqrt{k}}, \quad \text{for all} \quad x \in \mathbb{R}^n, \\ \text{where } \sigma_k(x)_p = \inf_{\hat{x} \in \Sigma_k} \|x - \hat{x}\|_p \quad \text{for } p = 1. \end{gathered} \tag{3.2}$$

guarantees exact recovery of all possible signals in $\Sigma_k$. Because if $x \in \Sigma_k$ then $\sigma_k(x)_1 = 0$, which implies that $\|\Delta(Ax) - x\|_2 = 0$, and so $\Delta(Ax) = x$. Indeed, if

$x \in \Sigma_k$ we can choose $\hat{x} = x$, from definition of infimum we conclude:

$$0 \leqslant \sigma_k(x)_1 = \inf_{\hat{x} \in \Sigma_k} \|x - \hat{x}\|_1 \leqslant \|x - x\|_1 \text{ so } \sigma_k(x)_1 = 0, \text{ when } x \in \Sigma_k.$$

**Note**: The infimum will be reached by a vector $\hat{x}_0 \in \Sigma_k$ so we could write min instead of inf. See the Appendix for justification.

The following theorem suggests that if there is any recovery algorithm satisfying (3.2), the matrix $A$ satisfies $NSP$ of order $2k$:

**Theorem 3.1.** *Let $A_{m\times n} : \mathbb{R}^m \to \mathbb{R}^n$ (where $m \ll n$), be a sensing matrix, and $\Delta : \mathbb{R}^m \to \mathbb{R}^n$ be an arbitrary recovery algorithm. If the pair $(A, \Delta)$ satisfies (3.2), then $A$ satisfies the NSP of order $2k$.*

*Proof.* Let $h \in \mathcal{N}(A)$ and suppose $\Lambda$ be the indices corresponding to the $2k$ largest entries of $h$. So that $h = h_\Lambda + h_{\Lambda^c}$ .
Split $\Lambda$ into $\Lambda_0$ and $\Lambda_1$ where $|\Lambda_0| = |\Lambda_1| = k$, since $|\Lambda| = 2k$, so that $h = h_{\Lambda_0} + h_{\Lambda_1}$.
Set $x = h_{\Lambda_1} + h_{\Lambda^c}$ and let $h = x - x'$, so we are going to have, $x' = -h_{\Lambda_0}$ .
This setting works because:
We know, $h = h_\Lambda + h_{\Lambda^c}$, if we split $\Lambda$ by the construction mentioned above, we will have: $h = h_{\Lambda_0} + h_{\Lambda_1} + h_{\Lambda^c}$, since $h = x - x'$ and also $x = h_{\Lambda_1} + h_{\Lambda^c}$ then $x = h_{\Lambda_1} + h_{\Lambda^c} - h_{\Lambda_0} - h_{\Lambda_1} - h_{\Lambda^c}$ so we get $x' = -h_{\Lambda_0}$.
Since by construction $x' \in \Sigma_k$ we can apply (3.2) to get $x' = \Delta(Ax')$. Plus, since $h \in \mathcal{N}(A)$: $Ah = A(x - x') = 0$, so $Ax = Ax'$ and we conclude that $x' = \Delta(Ax)$
*Moreover* $\|h_\Lambda\|_2 \leqslant \|h\|_2 = \|x - x'\|_2 = \|x - \Delta(Ax)\|_2$, and finally by (3.2)
$\|x - \Delta(Ax)\|_2 \leqslant \frac{C\sigma_k(x)_1}{\sqrt{k}} \leqslant \frac{C\sqrt{2}\|h_{\Lambda^c}\|_1}{\sqrt{2k}}$.
The last inequality is correct because since $\sigma_k(x)_1 = \inf_{\hat{x} \in \Sigma_k} \|x - \hat{x}\|_1$, and also by construction we know that $x = h_{\Lambda_1} + h_{\Lambda^c}$ and $h_{\Lambda_1} \in \Sigma_k$ so $\sigma_k(x)_1 \leqslant \|x - h_{\Lambda_1}\|_1 = \|h^c_\Lambda\|_1$, thus we get $\|h_\Lambda\|_2 < \frac{C\sqrt{2}\|h_{\Lambda^c}\|_1}{\sqrt{2k}}$
Which implies that A satisfies the NSP of order 2k by (3.2). □

**Definition 3.3.** *Suppose $A : \mathbb{R}^n \to \mathbb{R}^m$ denotes a sensing matrix and let $\Delta : \mathbb{R}^m \to \mathbb{R}^n$ be a recovery algorithm. Then $(A, \Delta)$ is said to be (C,k)-stable if for any $x \in \Sigma_k$ and $e \in \mathbb{R}^m$ we have:*

$$\|\Delta(Ax + e) - x\|_2 \leqslant C \|e\|_2 . \tag{3.3}$$

This definition implies that if we add a small amount of noise to the measurements, then its impact on the recovered signal should not be very large.

**Theorem 3.2.** *If the pair $(A, \Delta)$ is (C,k)-stable, then*

$$\frac{1}{C} \|x\|_2 \leqslant \|Ax\|_2 \quad \textit{for all} \quad x \in \Sigma_{2k}. \tag{3.4}$$

*Proof.* Let $y, z \in \Sigma_k$, and note that $y - z$ and $z - y \in \Sigma_{2k}$. We define

$$e_y = \frac{A(z - y)}{2} \quad \textit{and} \quad e_z = \frac{A(y - z)}{2}, \tag{3.5}$$

Then $Ay + e_y = Az + e_z = \frac{A(y+z)}{2}$. Let $\hat{y} = \Delta(Ay + e_y) = \Delta(Az + e_z)$. Using the (C,k)-stability and triangle inequality we have:

$$\begin{aligned} \|y - z\|_2 &= \|y - \hat{y} + \hat{y} - z\|_2 \\ &\leqslant \|y - \hat{y}\|_2 + \|\hat{y} - z\|_2 \\ &\leqslant C \|e_y\|_2 + C \|e_z\|_2 \\ &= C \|Ay - Az\|_2 . \end{aligned}$$

Since this hold for any $y, z \in \Sigma_k$ the result follows for any $x \in \Sigma_{2k}$ by Lemma 2.1. □

## 3.2 Restricted isometry property (RIP)

Matrix $A$ satisfies the restricted isometry property (RIP) of order $k$ if:
There exist a $\delta_k \in (0,1)$ such that:

$$(1-\delta_k)\|x\|_2^2 \leqslant \|Ax\|_2^2 \leqslant (1+\delta_k)\|x\|_2^2, \tag{3.6}$$

for every $x \in \Sigma_k$. If $A$ satisfies the RIP of order $2k$ then it means that $A$ approximately preserves the distance between any pair of k-sparse vectors.
Note that while we are using the given definition for RIP, we are considering bounds which are symmetric about 1, only for notational convenience.
In practice we can always use:

$$\alpha\|x\|_2^2 \leqslant \|Ax\|_2^2 \leqslant \beta\|x\|_2^2. \tag{3.7}$$

where $0 < \alpha < \beta < \infty$. To evaluate $\alpha$ and $\beta$ assume, $A = \frac{1}{\theta}(\theta A)$ where $\theta A = \widetilde{A}$ and $\theta$ is a finite scalar.
Then:

$$a.\theta^2\|x\|_2^2 \leqslant \left\|\widetilde{A}x\right\|_2^2 \leqslant \beta.\theta^2\|x\|_2^2.$$

So:

$$\begin{cases} \alpha\theta^2 = 1-\delta_k \\ \beta\theta^2 = 1+\delta_k. \end{cases} \Rightarrow (\alpha+\beta)\theta^2 = 2 \quad \Rightarrow \quad \theta = \sqrt{\frac{2}{\alpha+\beta}} \quad .$$

Having $\theta$, we can use it to get $\delta_k$:

$$\begin{aligned} \alpha.\theta^2 = 1-\delta_k &\Rightarrow \alpha.\frac{2}{\alpha+\beta} = 1-\delta_k \\ &\Rightarrow \frac{2\alpha}{\alpha+\beta} - 1 = -\delta_k \\ &\Rightarrow \frac{2\alpha}{\alpha+\beta} - \frac{\alpha+\beta}{\alpha+\beta} = \frac{\alpha-\beta}{\alpha+\beta} = -\delta_k \quad \Rightarrow \quad \delta_k = \frac{\beta-\alpha}{\alpha+\beta} \\ &\Rightarrow 0 < \delta_k < 1 \end{aligned}$$

Since $0 < \alpha < \beta$ then $\frac{\beta-\alpha}{\alpha+\beta}$ is positive and less than 1, so it can be our desired $\delta_k$. Hence using any $0 < \alpha < \beta < \infty$, we can always scale $A$ such that it satisfies the symmetric bounds about 1 in equation (3.6)

Note that if $A$ satisfies the RIP of order $k$ with the constant $\delta_k$, $0 < \delta_k < 1$, then $A$ satisfies the RIP of order $k'$ where $k' < k$. Then the constant $\delta_{k'}$ will be less than $\delta_k$ $(\delta_{k'} < \delta_k)$.

**Lemma 3.1.** *Let $k < n/2$ be given. Then there exists a set $X \subset \Sigma_k$ such that for any $x \in X$ we have $\|x\|_2 \leqslant \sqrt{k}$ and for any $x, z \in X$ with $x \neq z$*

$$\|x - z\|_2 \geqslant \sqrt{k/2}$$

*and*

$$\log|X| \geqslant \frac{k}{2}\log(\frac{n}{k})$$

The proof of this lemma could be found in Appendix B.

For an $m \times n$ sensing matrix A, to have the $RIP$ of order $2k < n$ with $\delta \leqslant 1/2$ there are some size constraints that must be met. To get these constraints we prove the following theorem and to do so we need to apply Lemma 3.1

**Theorem 3.3.** *Let $A$ be an $m \times n$ matrix that satisfies the RIP of order $2k$ where $2k < n$, with constant $\delta \in (0, 1/2]$ then*

$$m > Ck\log(n/k) \quad .$$

*Where* $C = 1/(2\log(\sqrt{96}+1))$ .

*Proof.* Since $A$ satisfies the RIP of order 2k, for any $x$ and $z$ with, $x \neq z \in X \subset \Sigma_k$ -we know $X \subset \Sigma_k$ implies $X \subset \Sigma_{2k}$, since $\Sigma_k \subset \Sigma_{2k}$- for the set of points $X$ in Lemma

3.1, then for $\delta \leqslant 1/2$ we have:

$$(1-\delta)\left\|x-z\right\|_2^2 \leqslant \left\|Ax-Az\right\|_2^2 \leqslant (1+\delta)\left\|x-z\right\|_2^2 \ .$$

Then by Lemma 3.1:

$$\left\|Ax-Az\right\|_2 \geqslant \sqrt{1-\delta}\left\|x-z\right\|_2 \geqslant \sqrt{k/4},$$

for all $x, z \in X$, since $x - z \in \Sigma_{2k}$.

We also have:

$$\left\|Ax\right\|_2 \leqslant \sqrt{1+\delta}\left\|x\right\|_2 \leqslant \sqrt{3k/2},$$

for all $x \in X$, since in that case we have, $\left\|x\right\|_2 \leqslant \sqrt{k}$.

From the lower bound we can get that for any points $x, z \in X$, if we center closed balls of radius $\sqrt{k/4}/4 = \sqrt{k/64}$ at $Ax$ and $Az$, then these balls will be disjoint. That is because the distance between centers is $d = \sqrt{k/4}$ then dividing by 4 makes us sure that the close balls centered at $Ax$ and $Az$ are disjoint. In the case of having open balls it suffices to divide $d$ by 2.

Also the upper bounds tells us that the entire set of balls is itself contained within a larger ball of radius $\sqrt{3k/2} + \sqrt{k/64}$ if we let $B^m(r) = \{x \in \mathbb{R}^m : \left\|x\right\|_2 \leqslant r\}$, then this implies that:

$$\begin{aligned}
vol(B^m(\sqrt{3k/2}+\sqrt{k/64}) &\geqslant |X|\,.vol(B^m(\sqrt{k/64}) \\
&\iff \\
(\sqrt{3k/2}+\sqrt{k/64})^m &\geqslant |X|\,.(\sqrt{k/96})^m \\
&\iff \\
(\sqrt{96}+1)^m &\geqslant |X| \\
&\iff \\
m &\geqslant \frac{\log|X|}{\log(\sqrt{96}+1)}.
\end{aligned}$$

Where the $|X|$ is the number of balls with radius $\sqrt{k/64}$.
By Lemma 3.1 we know $\log|X| \geqslant \frac{k}{2}\log\frac{n}{k}$. So merging it with $m \geqslant \frac{\log|X|}{\log(\sqrt{96}+1)}$ We get:

$$m \geqslant Ck\log(n/k) \quad .$$

Where $C = 1/(2\log(\sqrt{96}+1))$. □

## 3.3 RIP versus NSP

In this section we will show that if a matrix satisfies the $RIP$ it also satisfies the $NSP$. So we can conclude that the $RIP$ is a stronger condition than $NSP$.

**Theorem 3.4.** *Suppose that the matrix $A$ satisfies the $RIP$ of order $2k$ with $\delta_{2k} < \sqrt{2}-1 \leqslant \frac{1}{2}$. Then $A$ satisfies the $NSP$ of order $2k$ with constant:*

$$C = \frac{\sqrt{2}\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}} \quad .$$

The proof of this theorem involves proving the following lemmas:

**Lemma 3.2.** *Suppose $u \in \Sigma_k$. Then:*

$$\frac{\|u\|_1}{\sqrt{k}} \leqslant \|u\|_2 \leqslant \sqrt{k}\,\|u\|_\infty \quad .$$

*Proof.* For any u we have $\|u\|_1 = |\langle u, sgn(u)\rangle|$. Note that $sgn(x) = \begin{cases} 1 & x > 1 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$.
And by this definition: $x\,sgn(x) = |x|$.
Applying Cauchy-Schwarz inequality we obtain:

$$|\langle u, sgn(u)\rangle| \leqslant \|u\|_2\,\|sgn(u)\|_2$$

Then

$$\|u\|_1 \leqslant \|u\|_2 \|sgn(u)\|_2$$

Note that $sgn(u)$ has no more than $k$ non-zero entries equal to $\pm 1$ (since $u \in \Sigma_k$) and thus $\|sgn(u)\|_2 \leqslant \sqrt{k}$

$\Rightarrow$ We get the lower bound

$$\frac{\|u\|_1}{\sqrt{k}} \leqslant \|u\|_2 \ .$$

For the upper bound just notice that since $u \in \Sigma_k$ and by definition we know $\|u\|_2 = (\sum_{i=1}^{n}(u_i^2)^{1/2})$, and $\|u\|_\infty = \max_{i=1,2,\cdots,n} |u_i|$, thus we can observe that each of $k$ nonzero entries of $u$ can be upper bounded by $\|u\|_\infty$.

$\Rightarrow$ We get the upper bound:

$$\|u\|_2 \leqslant \sqrt{k}\,\|u\|_\infty \ .$$

And over all we have proven:

$$\frac{\|u\|_1}{\sqrt{k}} \leqslant \|u\|_2 \leqslant \sqrt{k}\,\|u\|_\infty \ .$$

□

Next Lemma is the second key to prove Theorem 3.4. This result holds for any arbitrary h, not only vectors in the null space of the matrix $A$. It is obvious that the argument is a lot simpler for $h \in \mathcal{N}(A)$

**Lemma 3.3.** *Suppose that $A$ satisfies RIP of order $2k$ and let $h \in \mathbb{R}^n$, $h \neq 0$. Let $\Lambda_0$ be any subset of $\{1, 2, \cdots, n\}$ such that $|\Lambda_0| \leqslant k$. Define $\Lambda_1$ as the index set corresponding to the $k$ entries of $h_{\Lambda_0^c}$ with the largest magnitude,and set $\Lambda = \Lambda_0 \cup \Lambda_1$. Then:*

$$\|h_\Lambda\|_2 \leqslant \alpha \frac{\|h_{\Lambda_{01}^c}\|}{\sqrt{k}} + \beta \frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}$$

*where*

$$\alpha = \frac{\sqrt{2}\delta_{2k}}{1-\delta_{2k}}, \quad \beta = \frac{1}{1-\delta_{2k}} \ .$$

In order to prove this lemma, we establish the following preliminary lemmas.

**Lemma 3.4.** *Suppose $u, v$ are orthogonal vectors. Then*

$$\|u\|_2 + \|v\|_2 \leqslant \sqrt{2}\,\|u+v\|_2 \quad .$$

*Proof.* We begin by defining the $2 \times 1$ vector $w = [\|u\|_2, \|v\|_2]^t \in \mathbb{R}^2$. By applying Lemma 3.2 with $k = 2$, we are having

$$\|w\|_1 \leqslant \sqrt{2}\,\|w\|_2 .$$

Using the identity $(a+b)^2 \leqslant 2(a^2+b^2)$ we have:

$$\|u\|_2 + \|v\|_2 \leqslant \sqrt{2}\sqrt{\|u\|_2^2 + \|v\|_2^2}. \tag{3.8}$$

Since $u$ and $v$ are orthogonal, $\|u\|_2^2 + \|v\|_2^2 = \|u+v\|_2^2$.
Using this in (3.8) we get the desired result:

$$\|u\|_2 + \|v\|_2 \leqslant \sqrt{2}\,\|u+v\|_2 \quad .$$

□

**Lemma 3.5.** *If $A$ satisfies the $RIP$ of order $2k$ then for any pair of vectors $u$,$v \in \Sigma_k$ with disjoint support,*

$$|\langle Au, Av\rangle| \leqslant \delta_{2k}\,\|u\|_2\,\|v\|_2 \ .$$

*Proof.* Suppose $u$,$v \in \Sigma_k$ with disjoint support and that $\|u\|_2 = \|v\|_2 = 1$, moreover $u \perp v$. Note that $u \pm v \in \Sigma_{2k}$ and $\|u \pm v\|_2^2 = 2$ (due to having disjoint support), then using the $RIP$

$$\|u \pm v\|_2^2\,(1-\delta_{2k}) \leqslant \|Au \pm Av\|_2^2 \leqslant \|u \pm v\|_2^2\,(1+\delta_{2k}) \ .$$

Notice that the support of a vector is defined as $supp(x) = \{i : x_i \neq 0\}$. Considering what we assumed:

$$2(1-\delta_{2k}) \leqslant \|Au \pm Av\|_2^2 \leqslant 2(1+\delta_{2k}) \ .$$

Knowing that

$\|Au+Av\|_2^2 = \langle Au+Av, Au+Av\rangle = \langle Au, Au\rangle + \langle Au, Av\rangle + \langle Av, Au\rangle + \langle Av, Av\rangle.$

And also

$\|Au-Av\|_2^2 = \langle Au-Av, Au-Av\rangle = \langle Au, Au\rangle - \langle Au, Av\rangle - \langle Av, Au\rangle + \langle Av, Av\rangle.$

So we have

$|\langle Au, Av\rangle| = \frac{1}{4}\left|\|Au+Av\|_2^2 - \|Au-Av\|_2^2\right| \leqslant \delta_{2k}$ .

So far we have shown that if $\|u_0\|_2 = \|v_0\|_2 = 1$, and they are in $\Sigma_k$ and have disjoint support thus we have:

$$|\langle Au_0, Av_0\rangle| \leqslant \delta_{2k} \ .$$

Suppose $u, v \in \Sigma_k$, are non-zero and have disjoint support and let $u_0 = \dfrac{u}{\|u\|}, v_0 = \dfrac{v}{\|v\|}$, so $\|u_0\| = \|v_0\| = 1$ with disjoint support. Then:

$$|\langle Au_0, Av_0\rangle| = \frac{|\langle Au, Av\rangle|}{\|u\|\,\|v\|} \ .$$

Using the properties of inner product we have:

$$|\langle Au, Av\rangle| \leqslant \delta_{2k}\,\|u\|_2\,\|v\|_2 \ .$$

□

**Lemma 3.6.** *let $\Lambda_0$ be an arbitrary subset of $\{1, 2, \cdots, n\}$ such that $|\Lambda_0| \leqslant k$. For any vector $u \in \mathbb{R}^n$, define $\Lambda_1$ as the index corresponding to the k largest entries of $u_{\Lambda_0^c}$ (in absolute value), $\Lambda_2$ as the index set corresponding to the next k largest entries, and so on. Then:*

$$\sum_{j\geqslant 2} \left\|u_{\Lambda_j}\right\|_2 \leqslant \frac{\left\|u_{\Lambda_0^c}\right\|_1}{\sqrt{k}} \ .$$

*Proof.* First Observe that for $j \geqslant 2$,

$$\left\|u_{\Lambda_j}\right\|_\infty \leqslant \frac{\left\|u_{\Lambda_{j-1}}\right\|_1}{k}. \tag{3.9}$$

And that's because the $\Lambda_j$ sort $u$ to have decreasing magnitude. We have:

$$\sum_{j\geqslant 2}\left\|u_{\Lambda_j}\right\|_2 \leqslant \sqrt{k}\sum_{j\geqslant 2}\left\|u_{\Lambda_j}\right\|_\infty \leqslant \frac{1}{\sqrt{k}}\sum_{j\geqslant 1}\left\|u_{\Lambda_j}\right\|_1 = \frac{\left\|u_{\Lambda_0^c}\right\|_1}{\sqrt{k}}\ .$$

Where the first inequality is obtained by applying Lemma 3.2, and the second inequality comes from the observation 3.9, and $\Lambda_0^c = \Lambda_1 \cup \Lambda_2 \cup \cdots \cup \Lambda_n$ □

Now we have all the keys we need, to prove Lemma 3.7:

**Lemma 3.7.** *Suppose that $A$ satisfies $RIP$ of order $2k$ and let $h \in \mathbb{R}^n$, $h \neq 0$. Let $\Lambda_0$ be any subset of $\{1, 2, \cdots, n\}$ such that $|\Lambda_0| \leqslant k$. Define $\Lambda_1$ as the index set corresponding to the $k$ entries of $h_{\Lambda_0^c}$ with the largest magnitude,and set $\Lambda = \Lambda_0 \cup \Lambda_1$. Then:*

$$\|h_\Lambda\|_2 \leqslant \alpha\frac{\left\|h_{\Lambda_0^c}\right\|_1}{\sqrt{k}} + \beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}$$

*where*

$$\alpha = \frac{\sqrt{2}\delta_{2k}}{1-\delta_{2k}}, \quad \beta = \frac{1}{1-\delta_{2k}}\ .$$

*Proof.* Since $h_\Lambda \in \Sigma_{2k}$ and the fact that $A$ satisfies the $RIP$ of order $2k$ we have:

$$(1-\delta_{2k})\|h_\Lambda\|_2^2 \leqslant \|Ah_\Lambda\|_2^2\ ,$$

From the lower bound on the $RIP$.

Define $\Lambda_j$ as in Lemma 3.6. Then since by definition $h_\Lambda = h - \Sigma_{j\geqslant 2}h_{\Lambda_j} = h_{\Lambda_0} + h_{\Lambda_1}$ and by linearity of A, $Ah_\Lambda = Ah - \sum_{j\geqslant 2} Ah_{\Lambda_j}$, we can write the lower bound on the $RIP$ as:

$$(1-\delta_{2k})\|h_\Lambda\|_2^2 \leqslant \langle Ah_\Lambda, Ah_\Lambda\rangle = \left\langle Ah_\Lambda, Ah - \sum_{j\geqslant 2} Ah_{\Lambda_j}\right\rangle .$$

So

$$(1 - \delta_{2k}) \|h_\Lambda\|_2^2 \leqslant \langle Ah_\Lambda, Ah \rangle - \left\langle Ah_\Lambda, \sum_{j \geqslant 2} Ah_{\Lambda_j} \right\rangle .$$

Now to bound $\left\langle Ah_\Lambda, \sum_{j \geqslant 2} Ah_{\Lambda_j} \right\rangle$ using Lemma 3.5, since we have that for $i \neq j, h_{\Lambda_i}, h_{\Lambda_j} \in \Sigma_k$ and have disjoint supports, hence they are orthogonal:

$$\left|\left\langle Ah_{\Lambda_i}, Ah_{\Lambda_j}\right\rangle\right| \leqslant \delta_{2k} \left\|h_{\Lambda_i}\right\|_2 \left\|h_{\Lambda_j}\right\|_2 \tag{3.10}$$

for any $i \neq j$ . Plus Lemma 3.4 yields: $\left\|h_{\Lambda_0}\right\|_2 + \left\|h_{\Lambda_1}\right\|_2 \leqslant \sqrt{2}\left\|h_{\Lambda}\right\|_2$. Now substituting it in (3.10) we have:

$$\begin{aligned}
\left|\left\langle Ah_{\Lambda}, \sum_{j\geqslant 2} Ah_{\Lambda_j}\right\rangle\right| &= \left|\sum_{j\geqslant 2}\left\langle Ah_{\Lambda_0}, Ah_{\Lambda_j}\right\rangle + \sum_{j\geqslant 2}\left\langle Ah_{\Lambda_1}, Ah_{\Lambda_j}\right\rangle\right| \\
&\leqslant \sum_{j\geqslant 2}\left|\left\langle Ah_{\Lambda_0}, Ah_{\Lambda_j}\right\rangle\right| + \sum_{j\geqslant 2}\left|\left\langle Ah_{\Lambda_1}, Ah_{\Lambda_j}\right\rangle\right| \\
&\leqslant \delta_{2k}\left\|h_{\Lambda_0}\right\|_2 \sum_{j\geqslant 2}\left\|h_{\Lambda_j}\right\|_2 + \delta_{2k}\left\|h_{\Lambda_1}\right\|_2 \sum_{j\geqslant 2}\left\|h_{\Lambda_j}\right\|_2 \\
&= \delta_{2k}\sum_{j\geqslant 2}\left\|h_{\Lambda_j}\right\|_2 \left(\left\|h_{\Lambda_0}\right\|_2 + \left\|h_{\Lambda_1}\right\|_2\right) \\
&\leqslant \sqrt{2}\delta_{2k}\left\|h_{\Lambda}\right\|_2 \sum_{j\geqslant 2}\left\|h_{\Lambda_j}\right\|_2 .
\end{aligned}$$

Using Lemma 3.6 this reduces to:

$$\left|\left\langle Ah_{\Lambda}, \sum_{j\geqslant 2} Ah_{\Lambda_j}\right\rangle\right| \leqslant \sqrt{2}\delta_{2k}\left\|h_{\Lambda}\right\|_2 \frac{\left\|h_{\Lambda_0^c}\right\|_1}{\sqrt{k}} .$$

Combining it with:

$$(1-\delta_{2k})\left\|h_{\Lambda}\right\|_2^2 \leqslant \langle Ah_{\Lambda}, Ah\rangle - \left\langle Ah_{\Lambda}, \sum_{j\geqslant 2} Ah_{\Lambda_j}\right\rangle = \left\langle Ah_{\Lambda}, Ah_{\Lambda}\right\rangle .$$

We have:

$$\begin{aligned}
(1-\delta_{2k})\left\|h_{\Lambda}\right\|_2^2 &\leqslant \left|\langle Ah_{\Lambda}, Ah\rangle - \left\langle Ah_{\Lambda}, \sum_{j\geqslant 2} Ah_{\Lambda_j}\right\rangle\right| \\
&\leqslant |\langle Ah_{\Lambda}, Ah\rangle| + \left|\left\langle Ah_{\Lambda}, \sum_{j\geqslant 2} Ah_{\Lambda_j}\right\rangle\right| \\
&\leqslant |\langle Ah_{\Lambda}, Ah\rangle| + \sqrt{2}\delta_{2k}\left\|h_{\Lambda}\right\|_2 \frac{\left\|h_{\Lambda_0^c}\right\|_1}{\sqrt{k}}.
\end{aligned}$$

Rearranging and dividing by $(1-\delta_{2k})\left\|h_{\Lambda}\right\|_2$ we get the desired result as follow:

$$\left\|h_{\Lambda}\right\|_2 \leqslant \sqrt{2}\delta_{2k}\frac{\left\|h_{\Lambda_0^c}\right\|_1}{\sqrt{k}(1-\delta_{2k})} + \frac{|\langle Ah_{\Lambda}, Ah\rangle|}{\left\|h_{\Lambda}\right\|_2(1-\delta_{2k})} .$$

Set:

$$\frac{\sqrt{2}\delta_{2k}}{1-\delta_{2k}} = \alpha, \quad \frac{1}{1-\delta_{2k}} = \beta \quad .$$

Therefore:

$$\|h_\Lambda\|_2 \leqslant \alpha \frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{k}} + \beta \frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2} \, .$$

Notice that if $\delta_{2k} < \sqrt{2}-1$ then $\alpha < 1$. Because the function $f(x) = \frac{\sqrt{2}x}{1-x}$ is strictly increasing, since $f'(x) = \frac{\sqrt{2}}{(1-x^2)} > 0$ and $f(\sqrt{2}-1) = 1$. □

Now we have all we need to prove Theorem 3.4, which we state again:

**Theorem 3.5.** *Suppose that the matrix $A$ satisfies the RIP of order $2k$ with $\delta_{2k} < \sqrt{2}-1$. Then $A$ satisfies the NSP of order $2k$ with constant:*

$$C = \frac{\sqrt{2}\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}} \, .$$

*Proof.* By definition we know that matrix $A$ satisfies the null space property, $NSP$ of order $2k$, if there exists a non-negative constant C such that for all $h \in \mathcal{N}(A)$,

$$\|h_\Lambda\|_2 < C' \frac{\|h_{\Lambda^c}\|_1}{\sqrt{2k}} \leqslant C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}} \qquad \text{where} \qquad |\Lambda| \leqslant 2k \, .$$

So it suffices to show $\|h_\Lambda\|_2 < C\frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}}$ holds for the case where $\Lambda$ is the index set corresponding to the $2k$ largest entries of $h$ . Therefore we can take $\Lambda_0$ to be the index set corresponding to the $k$ largest entries of $h$ and apply Lemma 3.7. The term $\beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}$ will be equal to zero since $Ah = 0$. So we only have:

$$\|h_\Lambda\|_2 \leqslant \alpha \frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{k}} \, .$$

Now by Lemma 3.2

$$\|h_{\Lambda_0^c}\|_1 \leqslant \sqrt{k}\,\|h_{\Lambda_0^c}\|_2 \, .$$

Noticing that $\Lambda = \Lambda_0 \cup \Lambda_1$ and also knowing that $\Lambda_0$ *and* $\Lambda_1$ are disjoint with $|\Lambda_0| = |\Lambda_1| = k$, using Lemma 3.2 we have:

$$\left\|h_{\Lambda_0^c}\right\|_1 = \|h_{\Lambda_1}\|_1 + \|h_{\Lambda^c}\|_1 \leqslant \sqrt{k}\,\|h_{\Lambda_1}\|_2 + \|h_{\Lambda^c}\|_1 \ .$$

So it results in:

$$\|h_\Lambda\|_2 \leqslant \alpha\Big(\,\|h_{\Lambda_1}\|_2 + \frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}}\Big)\ .$$

Since $\|h_{\Lambda_1}\|_2 \leqslant \|h_\Lambda\|_2$ we have :

$$\|h_\Lambda\|_2 - \alpha\,\|h_\Lambda\|_2 \leqslant \alpha\frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}}$$

$\Rightarrow$

$$(1-\alpha)\,\|h_\Lambda\|_2 \leqslant \alpha\frac{\|h_{\Lambda^c}\|_1}{\sqrt{k}}\ .$$

By assumption $\delta_{2k} < \sqrt{2}-1$ so we know that $\alpha < 1$ . Therefore we can divide by $(1-\alpha) > 0$ preserving the direction of the inequality to get the desired result.

$$\|h_\Lambda\|_2 \leqslant \frac{\alpha}{1-\alpha}\frac{\left\|h_{\Lambda_0^c}\right\|_1}{\sqrt{k}}\ .$$

So the theorem is proved by setting:

$$C = \frac{\alpha}{1-\alpha} = \frac{\sqrt{2}\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}}\ .$$

□

By above theorem we showed that the $RIP$ is a stronger property than $NSP$. In next chapter we will introduce a type of matrices that cover all the properties we need for our **sensing matrix**.

## 3.4 Conclusion

In this chapter we defined and proved some theorems and lemmas about the desired properties of a measuring/sensing matrix. We also showed that if a matrix

satisfies the $RIP$ it also satisfies the $NSP$ property. So we can conclude that the $RIP$ is a stronger condition than $NSP$. Knowing this fact, we move on to the problem of sensing matrix construction in the following chapter.

# Chapter 4

# Sensing Matrix Construction and Signal Recovery

## 4.1 Introduction

In the previous chapters we have outlined the properties of a sensing matrix which is one of the major problems in the world of *compressive sensing*. More specifically we showed that in order to be able to successfully reconstruct a signal, the sensing matrix must satisfy 1) Null space, 2) Spark or 3) RIP properties. In this chapter we first propose some practical methods which enables creation of the sensing matrix using some very low cost algorithms. Then we will briefly overview the recovery algorithm which uses convex optimization to reconstruct a signal.

## 4.2 Sensing Matrix Construction

Now that we found the relevant properties of a sensing matrix $A$, we need to construct a matrix which satisfies these properties. It is pretty clear that an $m \times n$ Vandermonde matrix $V$ from $m$ *distinct scalars* has full rank and so the spark(V) is going to be $m+1$. However when the number of columns $n$ is large, these matrices are poorly conditioned facing the recovery algorithm [26].

It has been shown that a random matrix of size $m \times n$ whose entries are independent and identically distributed with continuous distributions satisfies the properties of a sensing matrix [27, 28, 29, 30]. In the rest of this chapter as a proof to the concept, we take the random function from Matlab to generate a random Sparse signal and also a random sensing matrix and using a Library of Matlab functions that are reported in [31] we reconstruct the signal.

## 4.3 Deterministic UUP matrices

In this section we are going to look at the class of $RIP$ matrices from another point of view. In *compressive sensing* the problem of suggesting a deterministic method to construct a sensing matrix, is an example of a *derandomization problem,* Notice that the process of removing randomness is called derandomization.
As explained in above section a random matrix whose entries are identically randomized based on Gaussian or Bernoulli distribution, satisfies the desired properties of the measuring/sensing matrix. However these constructions all are probabilistic in nature. Meaning that these constructions are not 100% guaranteed to practically generate $UUP$ matrices. But what is a $UUP$ matrix?
$UUP$ stands for *uniform uncertainty principle*, which are generalization of (rectangular) orthogonal matrices where for all k-sparse sets B, the columns of the matrix

corresponding to B, are almost orthogonal, rather than being perfectly and globally orthogonal.

Such matrices will still have the important properties of orthogonal matrices like computable invertibility as long as we restrict our attention to sparse vectors. $UUP$ matrices are able to diminish high-dimensional sparse vectors into low-dimensional vectors and they are also able to reconstruct those high-dimensioned vectors. However, there is no fast algorithm known to test if a given matrix is $UUP$ or not. As mentioned above, the construction of the class of $UUP$ matrices involves some failures while the failure rate can be proven exponentially small. Still, since many applications of *compressed sensing* are for noisy environments, even very small inaccuracy will be dissatisfying. So it's inevitable to look for a deterministic way which can give us a reasonably fast algorithm to construct this class of matrices.

But what is rectangular orthogonal matrix? Let $a_1, a_2, a_3, \cdots, a_n$ be the $m-$ *dimensional* columns of $A$ which is an $m \times n$ matrix. Such a matrix is called orthogonal if these column vectors are orthonormal meaning that they have unit length $\|a_j\|_2 = 1$ and $\langle a_i, a_j \rangle = 0$ whenever $i \neq j$ (this requires $m \geqslant n$). One of the properties of orthonormal matrices is Pythagoras' theorem. If we look at the linear transformation of an $n-$ *dimensional* vector $(b_1, b_2, \cdots, b_n)$, by an orthonormal matrix $A$, according to Pythagoras' we will have:

$$\left\| \sum_{j=1}^{n} b_j a_j \right\|^2 = \sum_{j=1}^{n} |b_j|^2 . \tag{4.1}$$

The above mentioned is valid for all complex -and so for all real numbers- $b_1, b_2, \cdots, b_n$. Thus $(b_1, b_2, \cdots, b_n) \to \sum_{j=1}^{n} b_j a_j$ is an isometry so it can be inverted. Meaning that if we have $c = \sum_{j=1}^{n} b_j a_j$ the original coefficients $b_1, b_2, \cdots, b_n$ can be uniquely recovered. And plus, a small changes in $c$ will not cause huge oscillations in $b_1, b_2, \cdots, b_n$. We obviously see that the original coefficients can be recovered explicitly

by:

$$b_j = \langle c, a_j \rangle \ .$$

In *compressed sensing* we wish to make n as large as possible -because we are dealing with *high−dimensional* sparse vectors- and m as small as possible -to transfer those large vectors into smaller ones- . We have, however, a basic problem here. which is that a rectangular orthogonal matrix exists if the number of columns is less than the number of the rows, while we wish to have $m \ll n$. Therefore, in order to have such a matrix, it's necessary that $n \leqslant m$ otherwise we will have too many columns to still have the linear independence and therefore one must have a non-trivial solution for the linear combination:

$$b_1 a_1 + \cdots + b_n a_n = 0 \ .$$

So we get for some $(b_1, \cdots, b_n) \neq (0, \cdots, 0)$ which does not match with 4.1. One can make this condition weaker or in other words Pythagoras' theorem needs to hold approximately instead of exactly and we call it *almost orthogonality* condition:

$$0.9 \sum_{j=1}^{n} |a_j|^2 \leqslant \left\| \sum_{j=1}^{n} b_j a_j \right\|^2 \leqslant 1.1 \sum_{j=1}^{n} |a_j|^2 \ .$$

Considering this, the class of matrices we need to look into will become larger knowing that the constants 0.9 and 1.1 are not accurately important, but it still does not remove the condition $n \leqslant m$. It turns out that when we fix $k < m$ as sparsity parameter we can say the matrix $A$ or in other words, the set of columns: $(a_1, a_2, \cdots, a_n)$ is a $UUP$ matrix with sparsity $k$ if we have the almost orthogonality condition for any sets of coefficients $(b_1, b_2, \cdots, b_n)$ assuming at most $k$ of them are non-zero. Note that the constants 0.9 and 1.1 will become worse in case that k increases. So as long as we consider not more than $k$ vectors at a time, the set of vectors $a_1, a_2, \cdots, a_n$ will be almost the same as a set of perfectly orthogonal vectors. Almost orthogonality condition turns out to be the same as $RIP$ property which we showed if a matrix satisfies $RIP$ then it will satisfy the null space property. We showed it in

Theorem 3.4. The deterministic construction of $UUP$ matrices is still an open question But taking advantage of the randomness one can simply select the $a_1, a_2, \cdots, a_n$ randomly for instance using random normalized Gaussian vectors. Noticing that for very large-dimensional spaces any pairs of random vectors are almost orthogonal with high probability. [32]

## 4.4 Signal Recovery via $\ell_1$ Minimization

There now exist a wide variety of approaches to recover a signal $x$ from a restricted number of measurements, we begin with considering a natural first approach to the problem of recovering a sparse signal. Assume that we have a sparse, or at least compressible signal $x$ and $y$ is the signal sampled by the sensing matrix A or in other words $Ax = y$. It is natural to attempt to recover $x$ by solving the following optimization problem:

$$\hat{x} = \arg\min_z \|z\|_0 \quad \textit{subject to} \quad z \in B(y). \tag{4.2}$$

Where $B(y)$ ensures that $\hat{x}$ is consistent with the measurement $y$. For instance in the case of having noise-free measurements we can set $B(y) = \{z : Az = y\}$. When the measurements have been contaminated with a small amount of bounded noise we should instead consider $B(y) = \{z : \|Az - y\|_2 \leqslant \epsilon\}$.
Notice that in (4.2) it is assumed that the original signal $x$ is sparse. One way to translate this problem to something more tractable is to replace $\|.\|_0$ by its convex approximation $\|.\|_1$,Thus the problem will be

$$\hat{x} = \arg\min_z \|z\|_1 \quad \textit{subject to} \quad z \in B(y). \tag{4.3}$$

Provided that $B(y)$ is convex (4.3) is feasible. While it is clear that replacing (4.2) with (4.3) transforms an intractable problem into a tractable one, it may not be

obvious that the solution of (4.3) will be similar to the solution of (4.2) . However there are intuitive reasons to expect that the use of $\ell_1$ minimization will promote sparsity for sure.

Figure 4.1 shows that the solution to $\ell_1$ minimization problem is exactly the same as the solution to the $\ell_p$ minimization for any $p < 1$;

$$\hat{x} = \arg\min_{z} \|z\|_p \quad \textit{subject to} \quad z \in B(y).$$

Note for $p < 1$ the $\ell_p$ norm, is not a norm, because it does not satisfy the triangle inequality and also the unit ball is not convex.

The following lemma establishes an error bound for $\ell_1$ minimization algorithm described by (4.3) when combined with the sensing matrix satisfying RIP.

**Lemma 4.1.** *Suppose that the matrix $A$ satisfies the RIP of order $2k$ with $\delta_{2k} < \sqrt{2}-1$ let $x, \hat{x} \in \mathbb{R}^n$ be given, and define $h = \hat{x} - x$. Let $\Lambda_0$ denote the index set corresponding to the $k$ entries of $x$ with largest magnitude and $\Lambda_1$ the index set corresponding to the $k$ entries of $\Lambda_0^c$ with largest magnitude. Set $\Lambda = \Lambda_0 \cup \Lambda_1$. If $\|\hat{x}\|_1 \leqslant \|x\|_1$, then*

$$\|h\|_2 \leqslant C_0 \frac{\sigma_k(x)_1}{\sqrt{k}} + C_1 \frac{\langle Ah_\Lambda, Ah \rangle}{\|h_\Lambda\|_2} \quad .$$

*Where*

$$C_0 = 2\frac{1-(1-\sqrt{2})\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}}, C_1 = \frac{2}{1-(1+\sqrt{2})\delta_{2k}} \ .$$

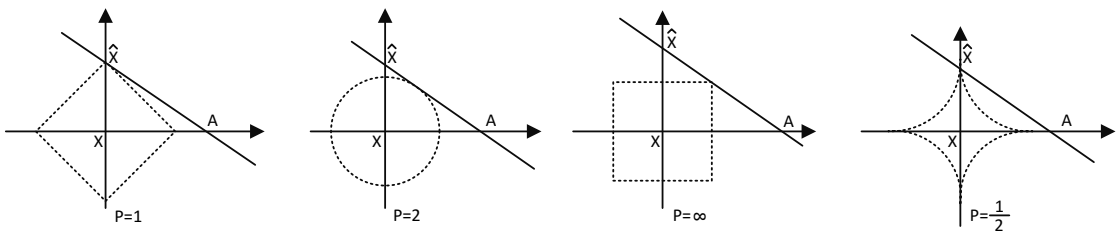


Figure 4.1: Demonstration of the general solution to $\ell_1$ minimization problem [24]

Recall $\sigma_k(x)_1 = \inf_{\hat{x}\in\Sigma_k} \|x - \hat{x}\|_1$ .

*Proof.* Let's start with noticing that $h = h_\Lambda + h_{\Lambda^c}$ so using the triangle inequality we have

$$\|h\|_2 \leqslant \|h_\Lambda\|_2 + \|h_{\Lambda^c}\|_2 . \tag{4.4}$$

First using Lemma 3.6 to bound $\|h_{\Lambda^c}\|$

$$\|h_{\Lambda^c}\|_2 = \left\|\sum_{j\geqslant 2} h_{\Lambda_j}\right\|_2 \leqslant \sum_{j\geqslant 2} \left\|h_{\Lambda_j}\right\|_2 \leqslant \frac{\left\|h_{\Lambda_0^c}\right\|_1}{\sqrt{k}}. \tag{4.5}$$

Where $\Lambda_j$ are defined as in Lemma 3.6, so $\Lambda_1$ are the index corresponding to the next k largest entries of $h_{\Lambda_0^c}$. $\Lambda_2$ as the index set corresponding to the next $k$ largest entries. and so on. We now wish to bound $\left\|h_{\Lambda_0^c}\right\|$. Since $\|x\|_1 \geqslant \|\hat{x}\|_1$, and $\hat{x} = x + h$, applying triangle inequality we obtain

$$\begin{aligned} \|x\|_1 \geqslant \|x+h\|_1 &= \|x_{\Lambda_0} + h_{\Lambda_0}\|_1 + \left\|x_{\Lambda_0^c} + h_{\Lambda_0^c}\right\|_1 \\ &\geqslant \|x_{\Lambda_0}\|_1 - \|h_{\Lambda_0}\|_1 + \left\|h_{\Lambda_0^c}\right\|_1 - \left\|x_{\Lambda_0^c}\right\|_1 . \end{aligned}$$

Rearranging and applying triangle inequality we have:

$$\begin{aligned} \left\|h_{\Lambda_0^c}\right\|_1 &\leqslant \|x\|_1 - \|x_{\Lambda_0}\|_1 + \|h_{\Lambda_0}\|_1 + \left\|x_{\Lambda_0^c}\right\|_1 \\ &\leqslant \|x - x_{\Lambda_0}\|_1 + \|h_{\Lambda_0}\|_1 + \left\|x_{\Lambda_0^c}\right\|_1 . \end{aligned}$$

Recalling that $\sigma_k(x)_1 = \left\|x_{\Lambda_0^c}\right\|_1 = \|x - x_{\Lambda_0}\|_1$,

$$\left\|h_{\Lambda_0^c}\right\|_1 \leqslant \|h_{\Lambda_0}\|_1 + 2\sigma_k(x)_1. \tag{4.6}$$

Combined with (4.5) we have:

$$\|h_{\Lambda^c}\|_2 \leqslant \frac{\|h_{\Lambda_0}\|_1 + 2\sigma_k(x)_1}{\sqrt{k}} \leqslant \|h_{\Lambda_0}\|_2 + 2\frac{\sigma_k(x)_1}{\sqrt{k}} \quad .$$

Where the last inequality comes from Lemma 3.2, and also observing that $\|h_{\Lambda_0}\|_2 \leqslant \|h_\Lambda\|_2$, this merged with (4.4) yields:

$$\|h\|_2 \leqslant 2\|h_\Lambda\|_2 + 2\frac{\sigma_k(x)_1}{\sqrt{k}}. \tag{4.7}$$

Now to establish a bound for $\|h_\Lambda\|_2$, we use a combination of Lemma 3.7, (4.6) and Lemma 3.2 to obtain:

$$\begin{aligned}\|h_\Lambda\|_2 &\leqslant \alpha\frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{k}} + \beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2} \\ &\leqslant \alpha\frac{\|h_{\Lambda_0}\|_1 + 2\sigma_k(x)_1}{\sqrt{k}} + \beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2} \\ &\leqslant \alpha\|h_{\Lambda_0}\|_2 + 2\alpha\frac{\sigma_k(x)_1}{\sqrt{k}} + \beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}\,.\end{aligned}$$

Since $\|h_{\Lambda_0}\|_2 \leqslant \|h_\Lambda\|_2$,

then

$$(1-\alpha)\|h_\Lambda\|_2 \leqslant 2\alpha\frac{\sigma_k(x)_1}{\sqrt{k}} + \beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}\quad.$$

We assumed that $\delta_{2k} < \sqrt{2}-1$ it ensures that $\alpha < 1$. Dividing by $(1-\alpha)$ and using (4.7) results in:

$$\|h\|_2 \leqslant 2\|h_\Lambda\|_2 + 2\frac{\sigma_k(x)_1}{\sqrt{k}} \leqslant \frac{2}{(1-\alpha)}\Big(2\alpha\frac{\sigma_k(x)_1}{\sqrt{k}} + 2\beta\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}\Big) + \frac{2\sigma_k(x)_1}{\sqrt{k}}$$

$$\|h\|_2 \leqslant \Big(\frac{4\alpha}{1-\alpha} + 2\Big)\frac{\sigma_k(x)_1}{\sqrt{k}} + \Big(\frac{4\beta}{1-\alpha}\Big)\frac{|\langle Ah_\Lambda, Ah\rangle|}{\|h_\Lambda\|_2}\quad.$$

Where

$$\alpha = \frac{\sqrt{2}\delta_{2k}}{1-\delta_{2k}},\quad \beta = \frac{1}{1-\delta_{2k}}\quad.$$

So after plugging $\alpha$ and $\beta$ we get the:

$$C_0 = 2\frac{1-(1-\sqrt{2})\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}}, C_1 = \frac{4}{1-(1+\sqrt{2})\delta_{2k}}$$

□

# Chapter 5

# $\ell_1$-Magic Construction Algorithm

In Chapter 2 we proved some important sensing matrix properties and in Chapter 4 we explained that to generate a sensing matrix it is enough to generate a normal random matrix, which is computationally easy to generate.

In this chapter, as a proof of the concept, we use the $\ell_1$-Magic algorithm that uses standard interior-point methods [33] to solve the *compressive sensing* based on convex optimization [31]. Using a set of Matlab library routines which are reported in [31], it is demonstrated that as long as $k \leq m$, the standard interior-point methods can successfully reconstruct sparse vectors that are sampled using a random sampling matrix.

The Algorithm 1 explained below is based on the recovery procedure developed by Candes and Romberg [31] and the encoding/decoding Matlab code is based on a code is reported at codeproject.com [34]. In this project we have modified the code to sample and reconstruct a sparse image, instead of a vector which is originally reported in [31, 34]. The modified Matlab code that generates a random sparse vector and a random sampling matrix is shown in Appendix C.

**Algorithm 1** Sampling an sparse image and reconstruct using $\ell_1$-Magic Algorithm

**Procedure** $\ell_1$-MagicImageCodingAndReconstruction

1: $w \leftarrow 50$ ▷ width of image
2: $l \leftarrow 50$ ▷ length of image
3: $n \leftarrow w \times l$ ▷ the length of the signal that is to sample
4: $k \leftarrow 100$ ▷ number of non-zero pixels
5: $m \leftarrow 500$ ▷ number of measurements
6: $signal \leftarrow [random]_{k \times 1}$ ▷ location non-zero pixels, $k$ random number
7: $x \leftarrow$ random sparse image with *peaks* non-zero random number.
8:
9: plot (x) ▷ plotting original signal
10:
11: $A \leftarrow [normal\ random]_{m \times n}$ ▷ the sampling matrix
12:
13: plot (y) ▷ plotting measured signal $y$
14:
15: $y \leftarrow A \times x$ ▷ sample matrix $x$ using sampling matrix $A$
16: $x_0 \leftarrow A^T \times y$ ▷ initial guess for the recovered signal
17: $x_p \leftarrow l1eq\_pd(x_0, A, y)$ ▷ call $\ell_1$-Magic reconstruction algorithm
18:
19: plot $(x_p)$ ▷ plotting reconstructed signal
20:
13: $x_{error} \leftarrow x - x_p$ ▷ the error signal
21:
22: plot $(x_{error})$ ▷ plotting the error signal
23:

We have demonstrated the signals generated using the above algorithm in Fig. 5.1 (a). The graph on the top which is the original signal is a vector of 2500 elements, represents the serialized form of a $50 \times 50$ image. The Image is sparse and only 100 of the pixels are not zero. While the Nyquist requires $50 \times 50$ measurements, compressive sensing relaxes and let having a percentage of it which in our case is only 500 measurements. The graph on Fig. 5.1 (b) shows the measured signal which is the inner-product of the sampling matrix to the original signal.

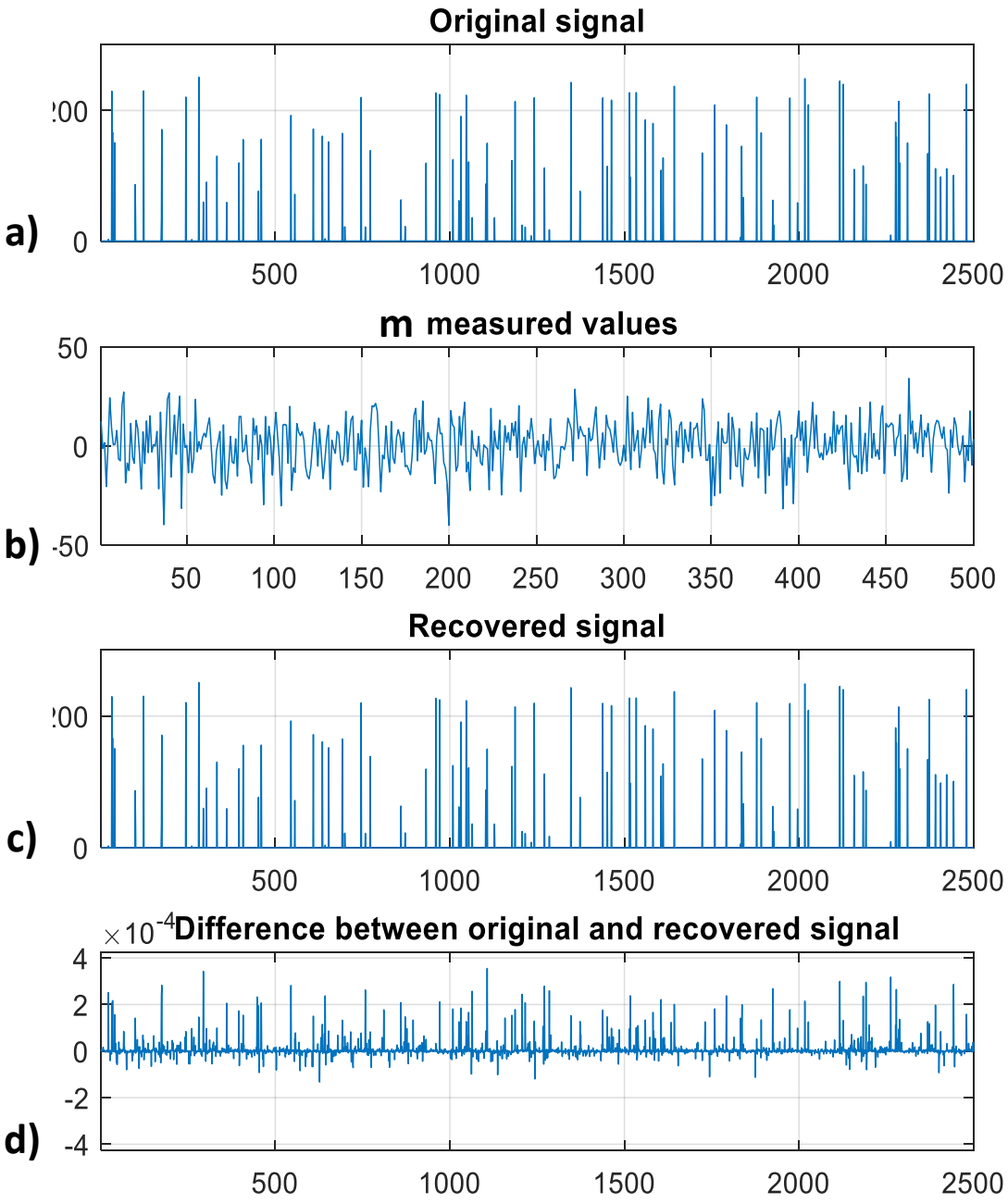


Figure 5.1: Demonstration of coding and decoding of an sparse random vector using $\ell 1$ Magic code.

Figure 5.1 (c) shows the result of the compressive sensing reconstruction of the signal. The reconstruction error is shown in Fig. 5.1 (d). One can see that the coding error is smaller than $2 \times 10^{-4}$ and considering that the magnitude of the signal, the errors are six orders of magnitude smaller than the original signal, and we have used only 500 measurements.

Figure 5.2 (a) and (b) shows the image form of the signals shown in Fig. 5.1 (a) and (c) respectively. The reformatting of the signals shown in Fig. 5.1 to an image is based on reshaping a vector of 2500 elements to a $50 \times 50$ matrix. One could visually confirm that the reconstructed image at Fig. 5.2 (b) is the same as the original image, Fig. 5.2 (a), with no distortion.

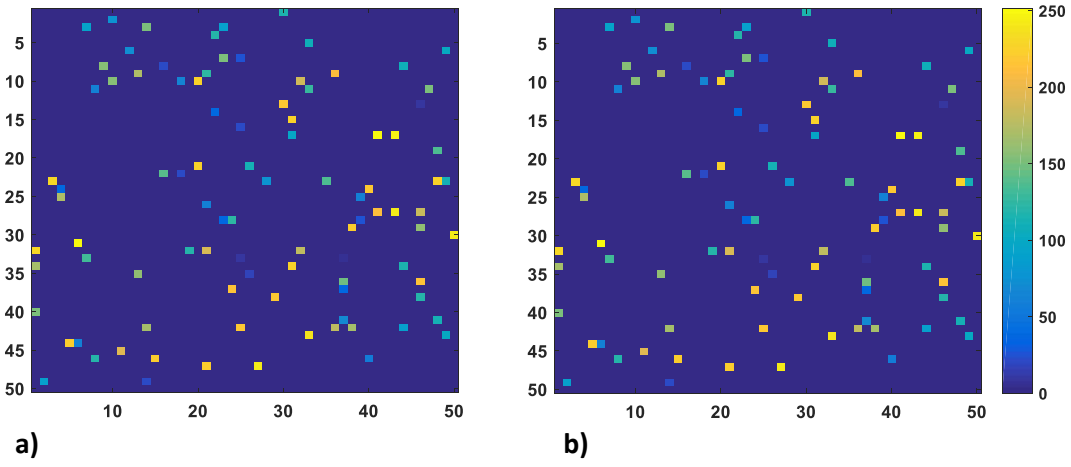


Figure 5.2: The original random image used as the input of $\ell 1$ Magic compressive sensing Matlab code (a) and the image after reconstruction (b)

# Chapter 6

# Conclusions

In this thesis we have studied *compressive sensing*, a new signal acquisition method which potentially reduces the number of measurements needed for a signal which is sparse or compressible in some basis. The compressive sensing which challenges the well-known Shannon-Nyquist theorem, instead of measuring the signal at uniform equi-spaced sampling points and then sending the signal to compression, measures the inner-product of a sensing matrix to a test-function to complete the sampling process, in order to reduce the dimension of the given signal or image, drastically. In this work we have detailed the most important properties of a sensing matrix that supports reduction of the number of measurements required for a reliable reconstruction. Then we have outlined that the sensing matrix properties, 1) Null space, 2) Spark and 3) RIP, can be generated using a normal random function. And finally we generated a random sparse image and employed the $\ell^1$ Magic algorithm which is a set of Matlab library codes for compressive sensing reconstruction, to recover the sparse image using convex optimization.

# Appendix A

# Justification for (3.2)

Justification for (3.2)

Minimum is reached in the definition of $\sigma_k(x)_p$:

$$\sigma_k(x)_p =: \inf_{\hat{x}\in\Sigma_k} \|x - \hat{x}\|_p$$

We are to justify that infimum is reached so it's the desired Minimum. Thus we need to show there is a $\hat{x}_0 \in \Sigma_k$ such that

$$\|x - \hat{x}_0\|_p = \sigma_k(x)_p$$

If so, we can replace inf by min.

By definition of infimum, there exists a sequence $\hat{x}_j \in \Sigma_k$ such that

$$\lim_{j\to\infty} \|x - \hat{x}_j\|_p = \sigma_k(x)_p$$

By **pigeonhole principle** there is a subsequence $\{\hat{x_{j_i}}\}_{i=1}^{\infty}$ such that all at most k non-zero terms for each $\hat{x_{j_i}}$ lie in the same position for all $i \geqslant 1$. Notice that there are finitely many possible positions for k nonzero entries (since the vectors are in $\mathbb{R}^n$ there are $\binom{n}{k}$ such positions, which gives us a finite number) considered to be holes

here, and infinitely many $\hat{x_{j_i}}$s (pigeons) to fill out the holes. By this principle, some hole must have infinitely many pigeons.

We assume without loss of generality that the initial sequence $\hat{x_j}$, has the following property: There is a set $S \subseteq \{1, 2, \cdots, n\}$ with $|S| = k$, such that the $\ell th$ position of $\hat{x_j}$ for all j, is zero if $\ell \in S^c$. In other words:

$$\hat{x_j} = (x_1^j, x_2^j, \cdots, x_n^j), \quad x_\ell^j = 0 \quad \text{if} \quad \ell \in S^c \ \ \forall j.$$

By passing to a subsequence we can ensure convergence along each entry if we have some compactness in $\mathbb{R}$ or $\mathbb{R}^n$. Knowing that compact sets in $\mathbb{R}$ are closed and bounded sets, all we need is to ensure that the sequence $\{x_i^j\}_{i=1}^{\infty}$, for each fixed i is a bounded sequence. Say $\forall j = 1, 2, 3, ..., \left|x_i^j\right| \leqslant M_i$, *and for each* $i =, 1, 2,$. Because then the sequence $\{x_i^j\}_{i=1}^{\infty}$ will be contained in the compact interval $[-M_i, M_i]$, and we know that equivalently every sequence in a compact space has a convergent subsequence that converges to some point in the space.

The fact that the subsequence $\{x_i^j\}_{j=1}^{\infty}$ for each fixed i is bounded hence it is contained in a closed interval $[a_i, b_i]$, comes from the knowledge that $\sigma_k(x)_p = \lim_{j\to\infty} \|x - \hat{x}_j\|_p$ is finite, which implies boundedness of $\|x - \hat{x}_j\|_p$.

$\|x\|_p - \|\hat{x}\|_p \leqslant \|x - \hat{x}_j\|_p = \sqrt[p]{(x_1 - x_1^j)^p + \cdots (x_n - x_n^j)^p} \leqslant M$

So each $\left|x_i^j - x_i\right| \leqslant M \Rightarrow \left|x_i^j\right| - |x_i| \leqslant M$

$$\Rightarrow \left|x_i^j\right| \leqslant M + |x_i| =: M_i \quad \forall j \geqslant 1$$

So for each i there is a subsequence of $\{x_i^j\}_{j=1}^{\infty}$ convergent on $[-M_i, M_i]$ But we need a subsequence so that for all i, $\{x_i^{j_k}\}_{i=1}^{\infty}$ converges. We know that if $i \in S^c$ then $x_i^{j_k} = 0$ for all k. So we need to take care of those terms with $i \in S$ . Let us order the elements of $S = \{\ell_1, \ell_2, \cdots, \ell_k\}$ where $\ell_1 < \ell_2 < \cdots < \ell_K$. We first locate a subsequence of the first non-zero terms, such that $\lim_{x\to\infty} x_{\ell_1}^{j_k} = x_{\ell_1}^0$ .We toss the term not n the convergent sequence and relable the sequence $\{\hat{x}_j\}$. Next locate a subsequence $\{\hat{x}_j\}$ such that $\{x_{\ell_2}^{j_k}\}_{k=1}^{\infty}$ converges to $x_{\ell_2}^0$. Notice that for this new subsequence still $\lim_{x\to\infty} x_{\ell_1}^{j_k} = x_{\ell_1}^0$. Toss all non-selected terms and repeat k times. At the end of this finite process,

whe will have a subsequence $\{\hat{x}_{j_k}\}_{k=1}^{\infty}$ of the original sequence $\{\hat{x}_j\}$ with the property that $\lim_{x\to\infty} x_i^{j_k} = x_i^0$, for each $i = 1, 2..., n$. Let $\hat{x}_0 = (x_1^0, x_2^0, \cdots, x_n^0)$, and we know for $\ell \in S^c$, $x_\ell^0 = 0$.

Without loss of generality we can assume by relabeling, that the original sequence $\{\hat{x}_j\}$ had the property that $\lim_{x\to\infty} x_i^j = x_i^0$; and necessarily $\hat{x}_j \to \hat{x}_0$, where the convergence is in $\mathbb{R}^n$ with any $\ell^p$ norm that is as $j \to \infty$:

$$\|\hat{x}_j - \hat{x}_0\|_p \to 0$$

so

$$\|x - \hat{x}_0\|_p = \lim_{j\to\infty} \|x - \hat{x}_j\|_p = \sigma_k(x)_p$$

For instance assume we have the given k-sparse vectors which have their non zero entries at same positions:

$$\begin{array}{lcccccccccccccccc}
\hat{x_1} & = ( & x_1^1 & 0 & 0 & x_4^1 & x_5^1 & 0 & 0 & \cdots & x_\ell^1 & 0 & 0 & 0 & 0 & 0 & ) \\
\hat{x_2} & = ( & x_1^2 & 0 & 0 & x_4^2 & x_5^2 & 0 & 0 & \cdots & x_\ell^2 & 0 & 0 & 0 & 0 & 0 & ) \\
\vdots & & \vdots & & & \vdots & \vdots & & & & \vdots & & & & & & \\
\hat{x_j} & = ( & x_1^j & 0 & 0 & x_4^j & x_5^j & 0 & 0 & \cdots & x_\ell^j & 0 & 0 & 0 & 0 & 0 & ) \\
\downarrow & & \downarrow & & & \downarrow & \downarrow & & & & \downarrow & & & & & & \\
\hat{x_0} & = ( & x_1^0 & 0 & 0 & x_4^0 & x_5^0 & 0 & 0 & \cdots & x_\ell^0 & 0 & 0 & 0 & 0 & 0 & )
\end{array}$$

Here in this example the set $S = \{1, 4, 5, ..., \ell\}$. Due to compactness as explained, we get a convergent subsequence along each entry. So we will be end up with a vector which is the desired $\hat{x}_0$.

# Appendix B

# Proof of Lemma 3.1

**Lemma 3.1:** Let $k < n/2$ be given. Then there exists a set $X \subset \Sigma_k$ such that for any $x \in X$ we have $\|x\|_2 \leqslant \sqrt{k}$ and for any $x, z \in X$ with $x \neq z$

$$\|x - z\|_2 \geqslant \sqrt{k/2}$$

and

$$\log |X| \geqslant \frac{k}{2} \log(\frac{n}{k})$$

*Proof.* Consider the set $U = \{x \in \{0, +1, -1\}^n : \|x\|_0 = k\}$ So by construction $\|x\|_2^2 = k$ for all $x \in U$. Therefore if we construct X by using elements in U we are going to have $\|x\|_2 = \sqrt{k}$. Also notice that $|U| = \binom{n}{k} 2^k$. And that's because the dimension of the vectors is $n$ and we have $k$ nonzero entries, so we have $\binom{n}{k}$ number of ways that we can choose $k$ locations out of $n$ where order does not matter. And since each non zero entry is either +1 or -1 then the cardinality of U is $|U| = \binom{n}{k} 2^k$. Note also that for all $x, z \in U$, $\|x - z\|_0 \leqslant \|x - z\|_2^2$, because:

*Appendix B. Proof of Lemma 3.1*

$$x_i - z_i = \begin{cases} 0 \quad if \quad x_i = z_i \\ 1 \quad if \quad x_i = 1, z_i = 0 \text{ or } x_i = 0, z_i = -1 \\ -1 \quad if \quad x_i = -1, z_i = 0 \text{ or } x_i = 0, z_i = 1 \\ 2 \quad if \quad x_i = 1, z_i = -1 \\ -2 \quad if \quad x_i = -1, z_i = 1. \end{cases}$$

Also

$$|x_i - z_i| = \begin{cases} 0 \quad if \quad x_i = z_i \\ 1 \quad if \quad x_i = 0, z_i \neq 0 \text{ or } x_i \neq 0, z_i = 0 \\ 2 \quad if \quad x_i z_i = -1. \end{cases}$$

Observing that $\|x_i - z_i\|_0 = \sum_{x_i \neq z_i} \frac{|x_i - z_i|}{|x_i - z_i|}$ and $\|x_i - z_i\|_2^2 = \sum_{x_i \neq z_i} |x_i - z_i|^2$, it is clear that when $|x_i - z_i| = 1$ then $\frac{|x_i - z_i|}{|x_i - z_i|} = 1 = |x_i - z_1|^2$ but when $|x_i - z_i| = 2$ then $\frac{|x_i - z_i|}{|x_i - z_i|} = 1 < 2^2 = |x_i - z_i|^2$.

So in conclusion $\|x - z\|_0 \leqslant \|x - z\|_2^2$.

Now if we take $\|x - z\|_2^2 \leqslant k/2$ then $\|x - z\|_0 \leqslant k/2$ and by this we see that for each fixed $x \in U$, it matters if $k$ is even or odd, so let: $m = \begin{cases} \frac{k}{2} & k, \text{ even} \\ \frac{k-1}{2} & k, \text{ odd} \end{cases} = \lfloor \frac{k}{2} \rfloor$.

$$\left|\left\{z \in U : \|x - z\|_2^2 \leqslant k/2\right\}\right| \leqslant \left|\left\{z \in U : \|x - z\|_0 \leqslant k/2\right\}\right| \leqslant \binom{n}{m} 3^{k/2} \quad .$$

The first inequality justification:

Let $\left\{z \in U : \|x - z\|_2^2 \leqslant k/2\right\}$ be set A and $\left\{z \in U : \|x - z\|_0 \leqslant k/2\right\}$, set B. If $z \in A$ then $\|x - z\|_2^2 \leqslant k/2$, but $\|x - z\|_0 \leqslant \|x - z\|_2^2 \leqslant k/2$, then $\|x - z\|_0 \leqslant k/2$ and we conclude that $z \in B$, or more generally $A \subseteq B$, and this implies $|A| \leqslant |B|$.

The second inequality justification:

If $\|x - z\|_0 \leqslant \frac{k}{2}$ then $x$ and $z$ must have at most $\lfloor \frac{k}{2} \rfloor = m$ different entries.Thus the cardinality of the set $z \in U$ such that $z$ and $x$ have at most $\lfloor \frac{k}{2} \rfloor$ different entries will be counted in $\binom{n}{m} 3^{k/2}$.

Knowing that the cardinality of $U$ is $\binom{n}{k} 2^k$, consider the set $A_0 = \{z \in U : \|x_0 - z\|_2^2 \leqslant$

$k/2\}$ with cardinality less than or equal to $\binom{n}{m}3^{k/2}$. It is clear that the set $\tilde{A}_0 = \{z \in U : \|x_0 - z\|_2^2 < k/2\}$ is a subset of $A_0$, or more generally, $\tilde{A}_0 \subset A_0 \subset U$. To construct $X$, considering the conditions of the lemma, suppose we pick $x_1 \notin \tilde{A}_0$ so $\|x_0 - x_1\|_2 \geqslant \sqrt{k/2}$. We observe the set $\tilde{A}_1 = \{z \in U : \|x_1 - z\|_2^2 < k/2\}$ has $\left|\tilde{A}_1\right| \leqslant \binom{n}{m}3^{k/2}$. Now we pick $x_2$, such that, $x_2 \notin \tilde{A}_0$ and $x_2 \notin \tilde{A}_1$ or $x_2 \in \tilde{A}_0^c \cap \tilde{A}_1^c$, and we repeat this process N times to construct the set X by iteratively choosing points that satisfy the condition of this lemma. The worst case scenario happens when after $N$ steps $\tilde{A}_0, \tilde{A}_1, \tilde{A}_2 \cdots, \tilde{A}_N$ are pairwise disjoint subsets of $U$. After adding $N$ points to the set, there are at least

$$\binom{n}{k}2^k - N\binom{n}{m}3^{k/2} \quad .$$

points left to pick from. Thus we can construct a set of size $|X| = N$ provided that

$$N\binom{n}{m}3^{k/2} \leqslant \binom{n}{k}2^k. \tag{B.1}$$

Now observe that when $k$ is **even**, $m = \frac{k}{2}$ and:

$$\frac{\binom{n}{k}}{\binom{n}{k/2}} = \frac{(k/2)!(n-k/2)!}{k!(n-k)!} = \frac{(n-k/2)(n-k/2-1)\cdots(n-k+1)}{k(k-1)\cdots(k-k/2+1)} = \prod_{i=1}^{k/2}\frac{n-k+i}{k/2+i}. \tag{B.2}$$

Consider $\phi(i) = \dfrac{n-k+i}{k/2+i}$, then $\phi'(i) = \dfrac{(k/2+i)-(n-k+i)}{(k/2+i)^2} = \dfrac{3k/2-n}{(k/2+i)^2}$, and since by hypothesis

$$k < n/2 \Rightarrow 3k/2 < 3n/4 < n \quad \text{so we conclude that } 3k/2 - n < 0.$$

Since the denominator is always positive we will have $\phi'(i) < 0$, so the function $\phi(i)$ is decreasing, therefore it gets its minimum at $i = k/2$ so

$$\prod_{i=1}^{k/2}\frac{n-k+i}{k/2+i} \geqslant (\frac{n}{k} - \frac{1}{2})^{k/2} \quad .$$

Now back to the problem of finding $N$ when:

$$N \leqslant \frac{\binom{n}{k}}{\binom{n}{k/2}}\frac{2^k}{3^{k/2}} = \frac{\binom{n}{k}}{\binom{n}{k/2}}(\frac{4}{3})^{k/2} \quad .$$

*Appendix B. Proof of Lemma 3.1*

By hypothesis $k < n/2$ or equivalently $\frac{n}{k} > 2$, so $\frac{-n}{k} < -2$, and dividing by 4 we have: $-\frac{n}{4k} < \frac{-2}{4} = \frac{-1}{2}$ So

$$\begin{aligned}\frac{\binom{n}{k}}{\binom{n}{k/2}}\left(\frac{4}{3}\right)^{k/2} &> \left(\frac{n}{k}-\frac{1}{2}\right)^{k/2}\left(\frac{4}{3}\right)^{k/2}\\ &> \left(\frac{n}{k}-\frac{n}{4k}\right)^{k/2}\left(\frac{4}{3}\right)^{k/2}\\ &= \left(\frac{3}{4}\frac{n}{k}\right)^{k/2}\left(\frac{4}{3}\right)^{k/2}\\ &= \left(\frac{n}{k}\right)^{k/2}.\end{aligned}$$

Now we need to justify the case when $k$ is **odd**,
By definition $m = \frac{k-1}{2}$, then the inequality *B.1* and the fact that $k = 2m+1$ will give us:

$$\begin{aligned}\frac{\binom{n}{k}}{\binom{n}{m}} &= \frac{(n-m)!m!}{(n-k)!k!}\\ &= \frac{(n-m)(n-m-1)\cdots(n-2m)}{(2m+1)(2m)\cdots(m+1)}\\ &= \prod_{i=0}^{m}\frac{n-2m+i}{m+1+i}.\end{aligned} \tag{B.3}$$

Consider $\phi(i) = \dfrac{n-2m+i}{m+1+i}$ Then $\phi'(i) = \dfrac{m+1+i-(n-2m+i)}{(m+1+i)^2} = \dfrac{3m-n+1}{(m+1+i)^2}$,

we know $k = 2m+1$ and $k < n/2$, by hypothesis, $2k-n<0$ and $2k = 4m+2$

$$\text{so} \quad 2k-n = 4m-n+2 = 3m-n+1+[m+1] < 0$$

$$\text{therefore}: \; 3m-n+1 < -[m+1] < 0$$

Since the denominator, $(m+1+i)^2$, is always positive $\phi'(i) < 0$. Or $\phi(i)$ is a decreasing

function, so it gets it's minimum at $i = m$ and we conclude that:

$$\begin{aligned}
&\prod_{i=0}^{m} \frac{n-2m+i}{m+1+i} \\
&\qquad \geqslant (\frac{n-m}{2m+1})^{m+1} \\
&\qquad = (\frac{n}{k} - \frac{m}{2m+1})^{m+1} \\
&\qquad > (\frac{n}{k} - \frac{1}{2})^{m+1}.
\end{aligned}$$

And since we are trying to find N which:

$$N \leqslant \frac{\binom{n}{k}}{\binom{n}{m}} \frac{2^k}{3^{k/2}} = \frac{\binom{n}{k}}{\binom{n}{m}} \left(\frac{4}{3}\right)^{k/2}. \tag{B.4}$$

For $k = 2m+1$, Knowing that by hypothesis $k < n/2$, or equivalently $\frac{n}{k} > 2$, so $\frac{-n}{k} < -2$, and dividing by 4 we have: $-\frac{n}{4k} < \frac{-2}{4} = \frac{-1}{2}$ and

$$\begin{aligned}
\frac{\binom{n}{k}}{\binom{n}{m}} \left(\frac{4}{3}\right)^{k/2} &\geqslant (\frac{n}{k} - \frac{m}{2m+1})^{m+1} \left(\frac{4}{3}\right)^{k/2} \\
&> (\frac{n}{k} - \frac{1}{2})^{m+1} \left(\frac{4}{3}\right)^{k/2} \qquad \text{and since} - \frac{n}{4k} < \frac{-2}{4} = \frac{-1}{2} \\
&> (\frac{n}{k} - \frac{n}{4k})^{m} + 1 \left(\frac{4}{3}\right)^{k/2} \\
&> (\frac{3n}{4k})^{m+1} (\frac{4}{3})^{m+1/2} \\
&= (\frac{3n}{4k})^{m} (\frac{3n}{4k}) (\frac{4}{3})^{m} (\frac{4}{3})^{1/2} \\
&= (\frac{n}{k})^{m} (\frac{3n}{4k} . \frac{2}{\sqrt{3}}) \\
&= (\frac{n}{k})^{m} (\frac{n\sqrt{3}}{2k}) \qquad \text{and since } \frac{n}{k} > 2 \\
&> (\frac{n}{k})^{m} \frac{\sqrt{3}}{2} 2 \\
&> (\frac{n}{k})^{m}.
\end{aligned}$$

Which is the exact same lower bound we found for the even case when $k = 2m$ What

we have done so far has all been about finding $N$ that fits in B.1 so if we consider:

$$N = \begin{cases} \lceil (\frac{n}{k})^m \rceil & when \quad \lceil (\frac{n}{k})^m \rceil \leqslant \frac{\binom{n}{k}}{\binom{n}{m}} (\frac{4}{3})^{k/2} \\ \lfloor (\frac{n}{k})^m \rfloor & Otherwise \end{cases} . \tag{B.5}$$

Considering the first case we have $N = \lceil (\frac{n}{k})^m \rceil \geqslant \left(\frac{n}{k}\right)^m$ .
And since $N = |X|$, we have:

$$\log |X| \geqslant m \log \frac{n}{k} .$$

The following charts show the surface, $\frac{\binom{n}{k}}{\binom{n}{m}} (\frac{4}{3})^{k/2} - \lceil (\frac{n}{k})^m \rceil$, is always greater than zero.

So in B.5, $N = \lfloor (\frac{n}{k})^m \rfloor$ is never the case. Therefore we always have:

$$\log |X| \geqslant m \log \frac{n}{k}$$

□

*Appendix B. Proof of Lemma 3.1*

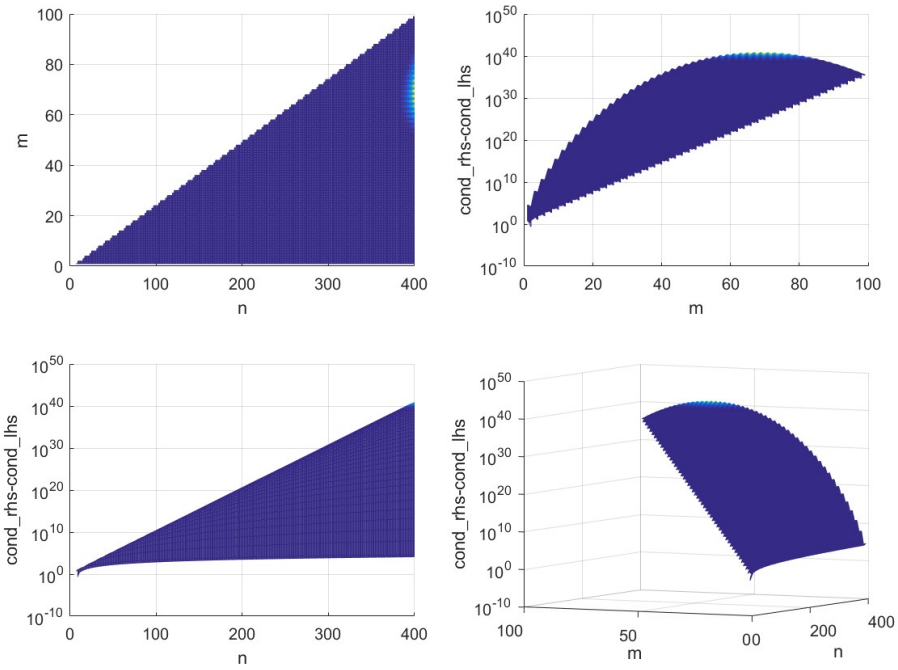


Figure B.1: The surface obtained by plotting $\frac{\binom{n}{k}}{\binom{n}{m}}(\frac{4}{3})^{k/2} - \lceil(\frac{n}{k})^m\rceil$. Notice that the surface never touches the $XY$ plane

# Appendix C

# $\ell_1$-Magic Matlab code

Listing C.1: $\ell_1$-Magic Matlab code adopted from [31] proving that that a sparse matrix that is sampled using a random matrix can be successfully reconstructed.

```
clear
path(path, './Optimization');
% Initialize constants and variables
rng(0);                  % set RNG seed
dim_x = 50;
dim_y = 50;
n    = dim_x * dim_y;    % length of signal
K    = 100;              % number of non-zero peaks
m    = 500;              % number of measurements to take (
   n < L)
x    = zeros(n , 1);     % original signal (K-sparse)

% Generate signal with K randomly spread values
peaks      = randperm(n);
peaks      = peaks(1:K);
x(peaks)   = rand(1, K)*256;

amp = 1.2*max(abs(x));
figure;
subplot(4,1,1);
plot(x);
```

```
21 title('Original signal');
22 xlim([1 n]);
23 ylim([0 amp]);
24 grid on;
25
26 x_org = x;
27 x = x ./ 128;
28
29 % Obtain m measurements
30 A = randn(m, n);
31 y = A*x;
32 subplot(4,1,2); plot(y); title('m measured values'); xlim
      ([1 m]); grid on;
33
34 % Perform Compressed Sensing recovery
35 x0 = A.'*y;
36 xp = l1eq_pd(x0, A, y);
37
38 xp = xp .* 128;
39
40 subplot(4,1,3);
41 plot(real(xp));
42 title('Recovered signal');
43 xlim([1 n]);
44 ylim([0 amp]);
45 grid on;
46
47
48 amp = 1.2*max(abs(x_org - xp));
49 subplot(4,1,4); plot(x_org - xp);
50 title('Difference between original and recovered signal');
51 xlim([1 n]);
52 ylim([-amp amp]);
53 grid on;
54
55 figure
56 imagesc(imresize(reshape(x_org,dim_x,dim_y),1)); colorbar
57 set(gca,'FontSize',12)
58 set(gca,'FontWeight','bold')
59
60 figure
```

```
61 imagesc(imresize(reshape(xp,dim_x,dim_y),1));    colorbar
62 % pos = get(gca, 'Position');
63 % pos(3) = 0.6;
64 % pos(1) = 0.5;
65 % set(gca, 'Position', pos)
66 set(gca,'FontSize',12)
67 set(gca,'FontWeight','bold')
68 % xlabel('Position of laser spot (um)', 'FontSize', 12);
```

# Appendix D

# Matlab source code for Figure B.1

Listing D.1: The Matlab source code to prove that the surface obtained by $\frac{\binom{n}{k}}{\binom{n}{m}}(\frac{4}{3})^{k/2} - \lceil(\frac{n}{k})^m\rceil$ never touches the $XY$ plane.

```
clear; warning('off','all')

n=1:400;
m=1:n(end)/2;
x = m;

k = 2 * m ;
for i=1:m(end)
    if ( mod(m(i),2) == 1 )
        k(i) = 2 * m(i)+ 1;
    end
end

f_m_n = NaN(size(m,2),size(n,2));

for i=1:m(end)
for j=2*k(i)+1:n(end)
    m_i = m(i);
    k_i = k(i);
    n_j = n(j);
    cond_lhs = ceil ((n_j / k_i) .^ m_i);
```

```
22      cond_rhs = nchoosek(n_j,k_i) / nchoosek(n_j,m_i) *
           (4.0/3.0).^(k_i/2);
23      if (cond_rhs-cond_lhs <= 0)
24          fprintf('%d %d  %d  %3.2f %3.2f %3.2f\n',m_i,k_i,
               n_j,cond_lhs,cond_rhs,cond_rhs-cond_lhs)
25      end
26      f_m_n(m_i,n_j) = cond_rhs-cond_lhs;
27  end
28  end
29
30  [X,Y] = meshgrid(n,m);
31
32  subplot(221)
33  mesh(X,Y,f_m_n)
34  set(gca, 'ZScale', 'log')
35  view([0 90]); xlabel('n'); ylabel('m');
36  zlabel('cond\_rhs-cond\_lhs')
37  set(gca,'ZTick',[1e-10 1e0 1e10 1e20 1e30 1e40 1e50]);
38  zlim([1e-10 1e50])
39
40  subplot(222)
41  mesh(X,Y,f_m_n)
42  set(gca, 'ZScale', 'log')
43  view([90 0]); xlabel('n'); ylabel('m');
44  zlabel('cond\_rhs-cond\_lhs')
45  set(gca,'ZTick',[1e-10 1e0 1e10 1e20 1e30 1e40 1e50]);
46  zlim([1e-10 1e50])
47
48  subplot(223)
49  mesh(X,Y,f_m_n)
50  set(gca, 'ZScale', 'log')
51  view([0 0]); xlabel('n'); ylabel('m');
52  zlabel('cond\_rhs-cond\_lhs')
53  set(gca,'ZTick',[1e-10 1e0 1e10 1e20 1e30 1e40 1e50]);
54  zlim([1e-10 1e50])
55
56  subplot(224)
57  mesh(X,Y,f_m_n)
58  set(gca, 'ZScale', 'log')
59  view([-63 5]); xlabel('n'); ylabel('m')
60  zlabel('cond\_rhs-cond\_lhs')
```

```
61 zlim([1e-10 1e50])
62 set(gca,'ZTick',[1e-10 1e0 1e10 1e20 1e30 1e40 1e50]);
```

# References


[1] Christine Alvarado, Jaime Teevan, Mark S Ackerman, and David Karger. Surviving the information explosion: How people find their electronic information. 2003.

[2] Alan V Oppenheim, Ronald W Schafer, John R Buck, et al. *Discrete-time signal processing*, volume 2. Prentice hall Englewood Cliffs, NJ, 1989.

[3] Guy E Blelloch. Introduction to data compression. *Computer Science Department, Carnegie Mellon University*, 2001.

[4] Chris Kyriakakis. Fundamental and technological limitations of immersive audio systems. *Proceedings of the IEEE*, 86(5):941–951, 1998.

[5] Kazuhiro Kondo. *Multimedia Information Hiding Technologies and Methodologies for Controlling Data*. IGI Global, 2012.

[6] Rafael C Gonzalez and Richard E Woods. *Digital image processing*. Prentice hall, 2002.

[7] Mohammed Ghanbari. *Standard codecs: image compression to advanced video coding*. Number 49. Iet, 2003.

[8] Ben Waggoner. *Compression for great video and audio: master tips and common sense*. Taylor & Francis, 2010.

[9] Stanley Chan. Class note for signals and systems; University of California, San Diego, August 2011. [Online; accessed 12-May-2016].

[10] Sophocles J Orfanidis. *Introduction to signal processing*. Prentice-Hall, Inc., 1995.

[11] Walt Kester. What the nyquist criterion means to your sampled data system design. *Analog Devices*, pages 1–12, 2009.

[12] Meixian Hao, Feng Han, and Yanxia Wang. Comments on nyquist rate in engineering application. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 3, pages V3–5. IEEE, 2010.

[13] Thomas Zawistowski and Paras Shah. An introduction to sampling theory. *Ejemplo de muestreo y sobremuestreo*, 2010.

[14] Bruno A Olshausen. Aliasing. *PSC 129–Sensory Processes*, pages 3–4, 2000.

[15] Tim Wescott. Sampling: What nyquist didnt say, and what to do about it. *Wescott Design Services, Oregon City, OR*, 2010.

[16] Robert J II Marks. *Advanced topics in Shannon sampling and interpolation theory*. Springer Science & Business Media, 2012.

[17] Moshe Mishali and Yonina C Eldar. From theory to practice: Sub-nyquist sampling of sparse wideband analog signals. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):375–391, 2010.

[18] Chien-Chia Chen. A survey on sub-nyquist sampling.

[19] Moshe Mishali and Yonina C Eldar. Sub-nyquist sampling. *Signal Processing Magazine, IEEE*, 28(6):98–124, 2011.

[20] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[21] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin E Kelly, Richard G Baraniuk, et al. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83, 2008.

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

[23] Emmanuel J Candè and Michael B Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

[24] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing. *Preprint*, 93(1):2, 2011.

[25] A Cohen, W Dahmen, and R DeVore. Compressed sensing and best k-term approximation,(2006). *Preprint*.

[26] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best -term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

[27] Djalil Chafaı, Olivier Guédon, Guillaume Lecué, and Alain Pajor. Interactions between compressed sensing, random matrices, and high dimensional geometry. *Panoramas et synthéses*, 37, 2011.

[28] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[29] Xuejun Liao. Understanding the isometry property of random matrices. 2009.

[30] Emmanuel J Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[31] Emmanuel Candes and Justin Romberg. l1-magic: Recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, 4:46, 2005.

[32] Terence Tao. Open question: deterministic UUP matrices. https://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices/, 2016. [Online; accessed 1-May-2016].

[33] Robert Robere. Interior point methods and linear programming. *University of Toronto*, 2012.

[34] Milliarde. Compressed Sensing Intro & Tutorial with Matlab. http://www.codeproject.com/Articles/852910/Compressed-Sensing-Intro-Tutorial-w-Matlab, 2016. [Online; accessed 11-December-2015].