COMBINATORIAL METHODS FOR COMPRESSED SENSING

By

Abdolreza Abdolhosseini Moghadam

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Doctor of Philosophy

2014

ABSTRACT

COMBINATORIAL METHODS FOR COMPRESSED SENSING

By

Abdolreza Abdolhosseini Moghadam

The inefficiency of the classical signal acquisition systems where a signal is sampled at the Nyquist rate and then compressed, raises the question that without knowing any prior, can't we directly sense a compressed version of the signal? In a series of papers, it has been shown that under certain conditions for the sensing operation and the signal of interest, a non-adaptive linear sampling scheme called Compressed Sensing (CS) can achieve such objective. Consequently a number of decoders have been proposed to recover a signal from it compressive samples. Two extreme approaches for those decoders are convex relaxation and combinatorial methods. The former approach requires the least number of samples for an exact reconstruction at the price of high complexities while the latter approach has low complexities but requires a higher number of samples for recovery. This thesis targets to design a CS approach which keeps only the best properties of convex relaxation and combinatorial approaches. In particular, we plan to show that, under our proposed sensing operator, a fast combinatorial approach would be an instance of the convex relaxation and thus it inherits all good properties of convex relaxation approaches, namely optimality in sample requirement and robustness in the presence of noise.

Keywords: Compressed sensing, compressive sampling, sparse coding, combinatorial algorithms, image processing and signal processing.

I dedicate this dissertation to the nicest family, which I was fortunate enough to be part of, i.e. my parents (Eynollah and Farkhondeh), my brothers (Mohammad Reza, Hamid Reza –rest in peace–, Ahmad Reza and Mohammad) and my beloved wife, Sara.

ACKNOWLEDGMENTS

I would like to thank my thesis committee members Dr. Jonathan I. Hall, Dr. Selin Aviyente and Dr. Karim G. Oweiss for their time, patience and valuable comments.

I also wish to express my utmost gratitude to Dr. Hayder Radha, my advisor. His guidance, support, sensible trust and so many other positive attributes of his, have been my greatest assets in the last seven years. Truly, it was both a privilege and an honor to know him and to be his student.

Finally, I want to thank my labmates, specially Mohammad Aghagolzadeh, Iman Barjasteh and Muhammad Usman Ilyas for helpful (and sometimes fun) discussions and time that we had and spent together.

TABLE OF CONTENTS

LIST OF FIGURES . . . . **viii**

Chapter 1 Introduction **1**
1.1 A new sampling scheme 1
1.2 Notations 2
1.3 Compressed Sensing 4
1.3.1 From Compressed Sensing to under-determined system of equations 4
1.3.2 Uniqueness of the solution 5
1.3.3 Tractability of the decoder 7
1.4 Related Works 9
1.5 Motivation for the proposed research 12
1.6 Challenges and proposed solutions 14
1.6.1 Going to higher fields 16
1.6.2 Augmenting with dense random rows 17
1.6.3 Robust and deterministic solver 17
1.6.4 Changing the problem formulation 18
1.7 Proposal organization 18

Chapter 2 Compressed sensing under hybrid projections **20**
2.1 Recursive approach for CS 20
2.2 Augmenting with dense rows 24
2.3 Alignment 25
2.4 Lexical order 26

Chapter 3 An extreme case of hybrid projections **28**
3.1 Introduction 28
3.2 Hybrid Compressed Sesning 28
3.2.1 HCS Solver 29
3.3 Optimal ratio of m_s and m_d under HCS 32
3.4 Simulation Results 38

Chapter 4 Complex Sparse Projections for Compressed Sensing (CRISP) **44**
4.1 Introduction 44
4.2 Forming the projection matrix 44
4.3 CRISP Solver 47
4.4 Optimality 50
4.4.1 The Average Behavior Analysis 51
4.5 Simulation Results 55

Chapter 5 Modified SSMP **57**
5.1 SERP projection matrix 58
5.2 w-RIP 60
5.2.1 Implications of w-RIP 63
5.3 w-RIP for SERP matrices 65
5.4 A Combinatorial solver for SERP 72
5.5 Simulation Results 82

Chapter 6 Future works **88**
6.1 Introduction 88
6.2 Another look at combinatorial algorithms 88
6.3 noise 94

BIBLIOGRAPHY **98**

LIST OF FIGURES

Figure 3.1 From left to right, the k-sparse image of Monalisa, reconstructed by applying HCS, BP and OMP respectively to the whole image ($m = 150746$). The decoding times are $t_{bp} = 109.5077$, $t_{hcs} = 16.1704$ and $t_{omp} = 15.1670$ seconds. 39

Figure 3.2 From left to right, the k-sparse image of Lenna, reconstructed with $m = 28$ compressive samples by applying HCS, BP and OMP respectively on blocks of size 8×8 pixel. The decoding times are $t_{bp} = 41.0283$, $t_{hcs} = 4.4735$ and $t_{omp} = 1.1512$ seconds. 39

Figure 3.3 m, n and k represent number of samples, signal length and signal sparsity respectively. 41

Figure 3.4 m, n and k represent number of samples, signal length and signal sparsity respectively. 42

Figure 3.5 m, n and k represent number of samples, signal length and signal sparsity respectively. 43

Figure 4.1 Comparing Compressive Demosaicing with some well-known demosaicing algorithms. 54

Figure 4.2 Comparing Compressive Demosaicing with some well-known demosaicing algorithms. 55

Figure 4.3 Comparing Compressive Demosaicing with some well-known demosaicing algorithms. 56

Figure 5.1 Relative error of reconstruction ($\|\Delta(y, \boldsymbol{P}) - x\|/\|x\|$) for 50 random signals x of length $n = 10,000$ as a function of the sparsity level $k = \|x\|_0$ and sampling ratio m/k, when the projection matrix is sparse binary and the solver is SSMP. 83

Figure 5.2 Relative error of reconstruction ($\|\Delta(y, \boldsymbol{P}) - x\|/\|x\|$) for 50 random signals x of length $n = 10,000$ as a function of the sparsity level $k = \|x\|_0$ and sampling ratio m/k, when the projection matrix is a sparse real-valued sensing matrix (SERP) with $r = 1$ and the solver is MSSMP. . . . 84

Figure 5.3 Relative error of reconstruction ($\|\Delta(y, \boldsymbol{P}) - x\|/\|x\|$) for 50 random signals x of length $n = 10,000$ as a function of the sparsity level $k = \|x\|_0$ and sampling ratio m/k, when the projection matrix is a sparse real-valued sensing matrix (SERP) with $r = 2$ and the solver is MSSMP. . . . 84

Figure 5.4 Comparing the performance of the sparse real valued projection matrices and MSSMP with sparse binary projections under SSMP decoder when samples are noisy. 86

Figure 5.5 Comparing the performance of the sparse real-valued and dense Gaussian projection matrices when BP is the decoder and samples are noisy. 86

Figure 5.6 Comparing the performance of the sparse real-valued and dense Gaussian projection matrices when Cosamp is the decoder and samples are noisy. . 87

Chapter 1

Introduction

1.1 A new sampling scheme

For many decades, the Nyquist-Shannon sampling theorem was the tenet core of signal acquisition systems. According to this theorem, a signal can be completely determined by its samples if the sampling rate is twice the maximum frequency of that signal. On the other hand, there are numerous applications where sampling at Nyquist frequency is either impossible, very expensive or time consuming. For instance, sensing each sample in Magnetic resonance Imaging (MRI) is so time consuming that, a full MRI might take several hours which is clearly inconvenient for patients [5]. Another example could be the class of signals with such high frequencies (e.g. terahertz) that there is no sampling device operating at those frequencies. On the other hand, we often deal with signals with a high level of correlation in some domain. For instance, it has been known for a long time that natural images can be approximated virtually with no perceptional loss by employing the Discrete Cosine Transform (DCT) or different types of Wavelets [51, 52, 53] and by retaining a fairly low number of coefficients. Indeed, this fact has been extensively used in efficiently compressing these classes of signals where instead of storing the image, a few "significant" transform coefficients of that image are stored and the rest are discarded. This classical technique warns us about the inefficiency of current sampling approach that a highly redundant signal would be sampled at the Nyquist rate and then after compression all those samples would be thrown away. This raises the question "*why go to so much effort to acquire all the data when most of what we get will* *be thrown away? Can't we just directly measure the part that won't end up being thrown away?*" [2]. Consequently, a new sampling framework was proposed independently by Donoho in [2] and Candes, Romberg and Tao in [5]. This scheme is widely referred to as "Compressed Sensing" or "Compressive Sampling" and often abbreviated by *CS* [1]-[15].

In those papers, it has been shown that signals with high correlations in some domain can be recovered from (sometimes much) fewer samples compared to traditional approaches. This can be achieved by exploiting:

- the fact that high correlation translates into existence of some "sparsifying domain" where the transform coefficients of those signals in respective domains are mostly zero or have very low magnitudes.
- Certain conditions on the sensing operator, ensuring that each sample carries some (yet little) information about all or at least a subset of the signal of the interest.

Furthermore, as opposed to traditional sampling methods where each sample is computed by convolving the signal (to be sensed) with a shifted delta dirac function, in CS each sample is computed by correlating the signal (to be sensed) with a waveform. This immediately implies that CS (at least in its original form) is a non-adaptive linear sampling scheme. Before presenting the compressed sensing framework, let us introduce the notations which will be used in this report.

1.2 Notations

In this report, we denote vectors by bold lower case letters (e.g. $\boldsymbol{\alpha}$) and matrices by bold upper case letters (e.g. $\boldsymbol{P}$). For a natural number $q \in \mathbb{N}$, define $[q] \triangleq \{1, 2, \ldots, q\}$. The cardinality of a set S is shown by $|S|$. The set difference operation is denoted by $\backslash$. The complement of S

is represented by S^c. The l-th entry of the vector $\boldsymbol{a}$ is denoted by $\boldsymbol{a}_l$. The support of a signal is denoted by $Supp(.)$, i.e.:

$$Supp(\boldsymbol{x}) \triangleq \{j : \boldsymbol{x}_j \neq 0\} \tag{1.1}$$

We use $\boldsymbol{I}_n$ to show the identity matrix of size n. We often use MATLAB notation for identifying submatrices of a matrix. For instance, for $\boldsymbol{P} \in \mathbb{R}^{n_1 \times n_2}$, $s_1 \subseteq [n_1]$ and $s_2 \subseteq [n_2]$, by $\boldsymbol{P}_{s_1,s_2}$ we mean the submatrix of $\boldsymbol{P}$ restricted to rows of s_1 and columns of s_2. Similarly we use the notion of ':' for instance $\boldsymbol{P}_{s_1,:}$ (or $\boldsymbol{P}_{:,s_2}$) to keep only rows (columns) indexed by s_1 (s_2). The transpose of $\boldsymbol{P}$ is denoted by $\boldsymbol{P}^T$. The Moore Penrose pseudoinverse of a matrix $\boldsymbol{P}$ is $\boldsymbol{P}^\dagger$, that is

$$\boldsymbol{P}^\dagger = \left(\boldsymbol{P}^T\boldsymbol{P}\right)^{-1}\boldsymbol{P}^T \tag{1.2}$$

We use $\odot$ for Hadamard (point wise) product. That is for a field $\mathbb{C}$ and $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{C}^n$, if $\boldsymbol{a} = \boldsymbol{b} \odot \boldsymbol{c}$ then $\boldsymbol{a}_l = \boldsymbol{b}_l\boldsymbol{c}_l$ for $l \in [n]$. Also $\otimes$ stands for Kronecker tensor product. In other words, for $n_1, n_2, m_1, m_2 \in \mathbb{N}$, $\boldsymbol{A} \in \mathbb{C}^{n_1 \times n_2}$ and $\boldsymbol{B} \in \mathbb{C}^{m_1 \times m_2}$ we have:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}_{1,1}\boldsymbol{B} & \dots & \boldsymbol{A}_{1,n_2}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n_1,1}\boldsymbol{B} & \dots & \boldsymbol{A}_{n_2,n_2}\boldsymbol{B} \end{bmatrix}$$

For $p \geq 1$, the ℓ_p norm of a real valued vector $\boldsymbol{a} \in \mathbb{R}^n$ denoted by $\|\boldsymbol{a}\|_p$ is:

$$\|\boldsymbol{a}\|_p = \left(\sum_{i=1}^{n} |\boldsymbol{a}_i|^p\right)^{\frac{1}{p}}$$

Two special cases are $p = 0$ and $p = \infty$ where we define pseudo norms of ℓ_0 and ℓ_∞ by:

$$i \in [n]: \; \|\boldsymbol{a}\|_0 \triangleq \#\{i : |\boldsymbol{a}_i| \neq 0\}, \; \|\boldsymbol{a}\|_\infty \triangleq \max\{|\boldsymbol{a}_i|\}$$

In words ℓ_0 of a vector counts the number of non-zero entries of that vector and ℓ_∞ is the maximum absolute value of that vector. If necessary, we define other notations in the subsequent chapters.

1.3 Compressed Sensing

1.3.1 From Compressed Sensing to under-determined system of equations

Consider the real valued discrete signal[1] $\boldsymbol{s} \in \mathbb{R}^n$ of length $n \in \mathbb{N}$. In the traditional sensing scheme, the sensing operator is $\boldsymbol{I}_n$, the identity matrix of size n. That is $\boldsymbol{y}$, the vector of sensed samples is simply

$$\boldsymbol{y} = \boldsymbol{I}_n \boldsymbol{s} = \boldsymbol{s}$$

The theory of Compressed Sensing attempts to answer the following question: is there any linear undersampling operator $\boldsymbol{\Phi} \in \mathbb{C}^{m \times n}$ where $m < n$ and a decoder Δ such that having samples $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s}$ and $\boldsymbol{\Phi}$, the estimated solution $\hat{\boldsymbol{s}} = \Delta(\boldsymbol{\Phi}, \boldsymbol{y})$ has the minimum error in some norm ℓ_p? That is, design $\Delta(.,.)$ and the sampling operator $\boldsymbol{\Phi}$ such that:

$$\min \|\Delta(\boldsymbol{\Phi}, \boldsymbol{y}) - \boldsymbol{s}\|_p \tag{1.3}$$

Unfortunately, such solution can not be found using classical linear algebraic approaches. To see that, note $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s}$ is an underdetermined system of linear equations and has an infinite set of

[1] The following arguments can be extended to analog signals by first discretizing an analog signal.

solution. More specifically, recall that for a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$, its kernel or null-space is:

$$\ker\left(\boldsymbol{\Phi}\right) = \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{\Phi}\boldsymbol{z} = \boldsymbol{0}\} \tag{1.4}$$

where $\boldsymbol{0}$ is a vector with all zero entries. Now note that in case of our problem where $m < n$, the rank of the null-space of $\boldsymbol{\Phi}$ is at least $n - m$, meaning that $\ker\left(\boldsymbol{\Phi}\right) \neq \emptyset$. Now assume that $\boldsymbol{z} \in \ker\left(\boldsymbol{\Phi}\right)$ then for all $\alpha \in \mathbb{R}$ we have

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s} = \boldsymbol{\Phi}\boldsymbol{s} + \alpha\boldsymbol{\Phi}\boldsymbol{z} = \boldsymbol{\Phi}\left(\boldsymbol{s} + \alpha\boldsymbol{z}\right)$$

Thus given $\boldsymbol{y}$ and $\boldsymbol{\Phi}$ there are infinitely many solutions (in the form of $\boldsymbol{s} + \alpha\boldsymbol{z}$) and we do not know which one is the correct solution. Consequently, first we need a guarantee that the solution $\boldsymbol{s}$, which we are looking for, is unique.

1.3.2 Uniqueness of the solution

As noted before for many interesting classes of signals, most notably audio and images, there exist *sparsifying transforms*. More specifically, by a sparsifying transform $\boldsymbol{\Psi} \in \mathbb{R}^{q \times n}$ for the signal $\boldsymbol{s} \in \mathbb{R}^q$ we mean that:

$$\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{x},\ k = \|\boldsymbol{x}\|_0 < q \tag{1.5}$$

where $\boldsymbol{x}$ is often referred to as *transform coefficients* or *representation* of $\boldsymbol{s}$ in the $\boldsymbol{\Psi}$ domain. Usually $n \geq q$ to span all vectors in $\mathbb{R}^q$. A measure for indicating how “good” a sparsifying transform $\boldsymbol{\Psi}$ is, the ability of that transform to make representations $\boldsymbol{x}$ more sparse. In words, we expect that for all signals $\{\boldsymbol{s}^{(i)}\}_i$ belonging to a specific class $\mathbb{D}$ (e.g. images of $w_1 \times w_2$ pixels), we have:

$$\forall \boldsymbol{s}^{(i)} \in \mathbb{D},\ \exists k \ll n,\ \boldsymbol{x} \in \mathbb{R}^n :\ \boldsymbol{s}^{(i)} = \boldsymbol{\Psi}\boldsymbol{x},\ \|\boldsymbol{x}\|_0 \leq k \tag{1.6}$$

Here we present the arguments in [43] to show that how the existence of a good sparsifying transform, changes drastically the uniqueness of a solution for an under-determined system of equations $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s}$. Suppose $\boldsymbol{s}$ has a k-sparse representation $\boldsymbol{x}$ in the $\boldsymbol{\Psi}$ domain, i.e. $\|\boldsymbol{x}\|_0 = k$ and $\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{x}$. As in the same paper [43], let us use the term "*projection matrix*" for the matrix $\boldsymbol{P} = \boldsymbol{\Phi}\boldsymbol{\Psi}$. Then

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x} = \boldsymbol{P}\boldsymbol{x}$$

Now assume that every $2k$ columns of the projection matrix are linearly independent. We claim that $\boldsymbol{x}$ is the unique minimizer to the following optimization problem:

$$\arg\min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{x}}\|_0 :\ s.t.\ \boldsymbol{y} = \boldsymbol{P}\hat{\boldsymbol{x}} \tag{1.7}$$

We prove this claim by contradiction. Assume there exists another solution, say $\boldsymbol{v} \neq \boldsymbol{x}$ such that $\boldsymbol{v}$ is also at most k-sparse ($\|\boldsymbol{v}\|_0 \leq k$). Thus

$$\boldsymbol{P}\boldsymbol{x} = \boldsymbol{P}\boldsymbol{v}\ ,\ \boldsymbol{P}\left(\boldsymbol{x} - \boldsymbol{v}\right) = \boldsymbol{0}$$

Consequently the vector $\boldsymbol{x} - \boldsymbol{v}$ is in the null-space of $\boldsymbol{P}$. However $\boldsymbol{x} - \boldsymbol{v}$ is at most $2k$ sparse. This violates the assumption that every $2k$ columns of the projection matrix are linearly independent. Hence $\boldsymbol{x}$ is the unique minimizer of (1.7). Note that, the immediate consequence of this proof is that $m = 2k$ is the minimum number of samples that one can sense from a signal with a k-sparse representation and does not lose any information. Since otherwise rank of $\boldsymbol{P}$ would be less than

$2k$ and one can find $2k$ linearly dependent columns of $\boldsymbol{P}$.

Now one can hope for solving (1.7) and then reconstruct the original signal $\boldsymbol{s}$ by $\boldsymbol{s} = \boldsymbol{\Psi x}$. Unfortunately, solving (1.7) is exponentially hard. Since, one must verify that can $\boldsymbol{y}$ be spanned by only one column of $\boldsymbol{P}$. Then he (she) tries any two columns of $\boldsymbol{P}$ to span $\boldsymbol{y}$ and so on. Consequently the next question after the uniqueness of the solution, is that: does there exists any decoder $\Delta(.,.)$ with polynomial time complexity for (1.7)?

1.3.3 Tractability of the decoder

The problem of the form (1.7) has been around in the signal processing and also physics communities for a long time, specifically for finding the transform coefficients of a signal in an overcomplete sparsifying transform where the number of atoms (or basis functions) of that transform is more than the dimension of the signal [17, 26, 52, 53]. Heuristically, the problem of (1.7) has been relaxed to the following optimization problem:

$$\arg\min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{x}}\|_1 : \ s.t.\ \boldsymbol{y} = \boldsymbol{\Phi}\hat{\boldsymbol{x}} \tag{1.8}$$

The reason behind changing the ℓ_0 norm in (1.7) to ℓ_1 norm formulation of (1.8) is that, ℓ_1 norm is the *closest convex* and valid norm to the non-convex pseudo norm of ℓ_0. Furthermore, the solution to (1.8) often has many small values, since any other $\ell_{p>1}$ norm, all the values with the magnitudes in the range of $(-1, 1)$ would be *damped*. Thus if any other norm than ℓ_1 was used in (1.8), then the solution would more likely have many values in the range of $(-1, 1)$ which is not desirable (recall that we are interested in finding a sparse solution).

Independently, Donoho in [2] and Candes, Romberg and Tao in [5] have proved that under certain conditions (to be noted shortly) for the projection matrix $\boldsymbol{P}$ the solution of (1.8) is indeed equal to the solution of (1.7). This is a major breakthrough since the problem of (1.8) is a convex optimization problem (more precisely linear programming) and can be solved tractably. Let us discuss the necessary conditions on the projection matrix.

For a matrix $\boldsymbol{P}$ and each integer $k = 1, 2, \ldots$, define the "*isometry constant*" [5, 3] of order k as the smallest number δ_k such that:

$$(1 - \delta_k)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{Px}\|_2^2 \leq (1 + \delta_k)\|\boldsymbol{x}\|_2^2 \tag{1.9}$$

holds for all k-sparse vectors $\boldsymbol{x}$. Note that a small isometry constant is equivalent to these statements:

1. That matrix (approximately) preserves the geometry (norm) of the signal $\boldsymbol{x}$, i.e. $\|\boldsymbol{x}\|_2 \approx \|\boldsymbol{Px}\|_2$.
2. That matrix approximately preserves the distances between vectors in the compressed space $\|\boldsymbol{x_1} - \boldsymbol{x_2}\|_2 \approx \|\boldsymbol{Px}_1 - \boldsymbol{Px}_2\|_2$.
3. All eigen-values of the sub-matrices (with k columns) of $\boldsymbol{P}$ are in the range of $[1-\delta_k, 1+\delta_k]$.
4. Submatrices (with k columns) of $\boldsymbol{P}$ approximately behave like an orthonormal frame.

We say that a matrix $\boldsymbol{P}$ has a "*Restricted Isometry Property*" (RIP) of order k, if $\delta_{2k} < \sqrt{2} - 1$. Now we can present the main theorem of Compressed Sensing (in a compact form) [3].

Theorem 1.3.1. *Assume* $\boldsymbol{y} = \boldsymbol{Px} + \boldsymbol{e}$, $\|\boldsymbol{e}\|_2 \leq \epsilon$, $\delta_{2k} < \sqrt{2} - 1$ *and let* $\hat{\boldsymbol{x}} = \arg\min \|\bar{\boldsymbol{x}}\|_1$ *subject to* $\|\boldsymbol{y} - \boldsymbol{P}\bar{\boldsymbol{x}}\|_2 \leq \epsilon$. *Then*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq C_0 k^{-1/2}\|\boldsymbol{x} - \boldsymbol{x}^{\star}\|_1 + C_1\epsilon$$

where C_0 and C_1 are constants (independent from n) and $\boldsymbol{x}^\star$ is the best k-sparse approximation to $\boldsymbol{x}$.

The theorem is extremely strong, since:

- It is robust to noise $\boldsymbol{e}$.
- Even if the signal of interest $\boldsymbol{x}$ is not exactly sparse, it returns the best sparse approximation of that signal.

The only question to be answered is: which matrices have RIP? Generally verifying RIP for matrices is either impossible or the bounds are not very tight. However, there are proofs of the existence of RIP for dense matrices with entries coming from (a) uniformly random (b) normal distribution (c) Bernoulli matrices and (d) partial Fourier matrices [16, 5, 2]. Furthermore the best bound on the number of rows or equivalently the number of samples required to guarantee RIP is for dense matrices with entries from uniformly random or normal when $m = O\left(k \log(n/k)\right)$. In other words, to make the recovery of a k-sparse vector $\boldsymbol{x}$ from an under-determined system of linear equations $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$ tractable, we have to increase the number of samples from $m = 2k$ to $m = O\left(k \log(n/k)\right)$. Furthermore, this bound is tight. In other words, if the order of the number of samples is less than $O\left(k \log(n/k)\right)$, then either the recovery is not possible for "all" sparse signals or there is no tractable decoding algorithm for that amount of samples.

1.4 Related Works

Broadly speaking there are three approaches to the problem compressed sensing:

1. **Convex relaxation:** As stated before,the first proposed protocol for CS was relaxing the non-convex cost function of $\min\ \|.\|_0$ to the convex one $\min\ \|.\|_1$ and use random dense

matrices for the projection matrix (guaranteeing RIP). Since the derived bounds in [5, 2] are tight, only a few number of research directions remain in the convex relaxation approaches. One is to design more efficient (in terms of complexity) linear programming solver to find the solution to (1.8). This can range from using gradient descent methods [19]-[24] to recent interior points in [17]. The other research direction in this category is to reformulating (1.8) such that either the complexity decreases or the quality of reconstruction increases. For instance the algorithm Basis Pursuit (BP) [17] exactly solves (1.8). While there are other formulations such as Lasso [18] which solves the following problem:

$$\arg\min_{\hat{\boldsymbol{x}}} \|\boldsymbol{y} - \boldsymbol{\Psi}\hat{\boldsymbol{x}}\|_2 : \; s.t. \; \|\hat{\boldsymbol{x}}\|_1 \leq k \tag{1.10}$$

In practice, the complexity of Lasso is slightly lower than BP but it usually leads to higher errors of reconstruction as well. Furthermore, usually the sparsity level k of the unknown vector $\boldsymbol{x}$ is not known beforehand, except in possibly some applications. Consequently BP has become the most popular solver in this category of decoders within the CS community.

2. **Greedy pursuit:** In these approaches [26, 28, 30, 29], a greedy approach based on the original Matching Pursuit (MP) [25] is adapted for finding the solution to (1.7) where in each step of the algorithm, one coordinate of the solution would be set to a non-zero value such that the residue of the approximation up to that point would be minimized. That is, having samples $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$, projection matrix $\boldsymbol{P}$ and current estimate $\hat{\boldsymbol{x}}$, the following problem is solved:

$$\arg\max_{i} \frac{\boldsymbol{P}_{:,i}^{T}(\boldsymbol{y} - \boldsymbol{P}\hat{\boldsymbol{x}})}{\|\boldsymbol{P}_{:,i}^{T}\|_2 \|\boldsymbol{y} - \boldsymbol{P}\hat{\boldsymbol{x}}\|_2} \tag{1.11}$$

and then $\hat{\boldsymbol{x}}_i$ would be updated accordingly to minimize $\|\boldsymbol{R}\|_2$ where $\boldsymbol{R} = \boldsymbol{y} - \boldsymbol{P}\hat{\boldsymbol{x}}$ is the *residue* of approximation till that step. The reason behind this greedy decision is that, if the correlation of the residue with the i-th column of the projection matrix is high, then there is a good chance that $\boldsymbol{x}_i$ has large magnitude. Note that these greedy approaches directly attempt to solve the ℓ_0 minimization problem of (1.7). Thus proving the convergence of these approaches to the correct solution can be challenging. In fact except for a few methods such as [28, 30], there is either no proof (of convergence) for many greedy methods or certain conditions on $\boldsymbol{P}$ are required (e.g. very small values for δ_{3k}, etc.) which validating their existence or satisfaction for matrices might be virtually impossible.

3. **Combinatorial Algorithms:** In these approaches [31]-[42], a carefully designed sensing matrix $\boldsymbol{\Phi}$ is deployed in the sensing stage such that the effective projection matrix $\boldsymbol{P} = \boldsymbol{\Phi\Psi}$ would be a highly structured and often binary sparse matrix. Then the sparsity and structured attributes of $\boldsymbol{P}$ are utilized usually in a group testing like algorithm to solve (1.7). Similar to greedy approaches, the proof of converging to the correct solution in these approaches is very difficult. Thus we do not have strong proofs as in convex relaxation methods guaranteeing this methods would converge to the correct solution. We discuss more about this approach later in this report.

One might ask while there are available linear programming solvers to solve (1.8), why should we look for alternative solutions? The answer to this question is solely due to the complexity of decoders, belonging to the convex relaxation methods. In fact, the complexity of a linear programming solver equals to solving an inverse problem of the same size. That is if the unknown sparse vector $\boldsymbol{x} \in \mathbb{R}^n$, then the complexity of solving (1.8) is $O(n^3)$. Although this order of complexity is a polynomial time, it is so high such that BP (or any other ℓ_1 minimizer) can be only utilized in small to medium size problems where the length of the signal is less than a few thousands. On the other hand, for many classes of signals (e.g. digital images) the length is in the order of millions (mega pixel images). This puts a big restriction on the usability of BP in those applications. Now let us present a brief comparison of different CS approaches in here.

The set of algorithms belonging to convex relaxation approaches works great as long as the projection matrix holds RIP. Not only they have the highest level of robustness in the presence of noise, they also exhibit excellent results even if the underlying signal $\boldsymbol{x}$ is not "exactly sparse", meaning that if the signal has many small and not zero entries. Furthermore, in practice, they require the least number of samples to recover $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$. However, as stated before, the main disadvantage of these solvers is their high complexities. Furthermore, this class of solvers usually only operates on dense projection matrices $\boldsymbol{P}$. Thus in applications where storage is of concern, storing the projection matrix suitable for convex relaxation methods is another challenge ($O(mn)$). On the other extreme, approaches belonging to the class of combinatorial algorithms have the lowest complexities, usually either linear or sub-linear in the signal length $\approx O(n)$. Furthermore, the sparsity of the projection matrix translates into fast matrix operations and efficient storage ($\approx O(n)$). However, these algorithms (virtually always) requires the highest number of samples for the recovery compared to the other classes of decoders. Furthermore, these algorithms are (usually) not robust in the presence of noise and only works with exactly sparse signals. Finally, greedy approaches are intermediate in their running time, sampling efficiency and robustness. However, many greedy algorithms require parameter tuning which are often signal dependent [30, 28].

1.5 Motivation for the proposed research

Indeed, the ultimate goal in the compressed sensing framework is to design an effective projection matrix $\boldsymbol{P}$ and the associated decoder (solver) Δ with the following characteristics:

1. The required number of samples for the recovery of $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$ is optimal, i.e. $m = O(\,k \log(n/k))$.

2. The complexity of the solver is minimal, for example $O(n)$ or even lower.

3. The scheme is robust to noise.

4. The algorithm works even with approximately sparse signals.

So far all of these properties have not been satisfied by a single scheme. We plan to achieve this goal by utilizing the valuable lessons which CS approaches provide us. Specifically we plan to design a CS scheme with the best properties of convex relaxation and combinatorial approaches. At first glance, this task seems impossible since convex relaxation methods (virtually always) operates on random dense projection matrices while combinatorial approaches only works with highly structured sparse binary matrices. However in this thesis, we plan to show and prove that with some modifications of the projection matrices suitable for combinatorial algorithms and designing associated solver, we might be able to achieve the aforementioned goals. To that end, we must slightly deviate from classical combinatorial approaches. More specifically we investigate the effect of substituting the binary entries in a sparse structured matrix with ones from larger fields, say real or complex numbers. Note that, this technique is reminiscent of modifying codes in classical channel coding theory [54, 55]. Consequently new respective solvers must be developed for this new class of projection matrices. One of our objectives is to emulate the Luby Transform (LT) decoder [50] (which is a rateless, near optimal erasure correcting code) for solving the problem of CS. In fact, it has been observed that the problem of compressed sensing (solving an underdetermined system of equations when the solution is sparse) has very close relationship with the problem of channel coding where we have a noisy over-complete system of linear equations [9]. Thus, another objective of this thesis is to establish a strong link between the problems of channel coding and CS. Finally, we introduce a systematic perspective to the problem of CS under combinatorial approaches which might later help us in finding rigorous proof for the convergence of the proposed solver.

1.6 Challenges and proposed solutions

To design a low complexity yet robust and optimal (in terms of sample requirement) compressed sensing framework, it would be beneficial to find the reasons why combinatorial approaches to CS fail to achieve the same goal. A typical combinatorial algorithm usually follows this simple algorithm to recover a sparse signal $\boldsymbol{x}$ of length n from $m < n$ samples $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$:

1. Initialize with the estimation of $\hat{\boldsymbol{x}} = \boldsymbol{0}$, a zero vector of length n.

2. Let s_i be the column indices where the i-th row of the projection matrix is non-zero (and in fact one):

   $$s_i \triangleq \{j : \boldsymbol{P}_{i,j} \neq 0\} \tag{1.12}$$

   We say that the i-th sample *spans* signal coordinates s_i.

3. Find a subset of sample indices $\omega \subseteq [m]$ with the following properties: (a) the size of this set is more than a threshold c ($|\omega| > c$) (b) all samples with indices in ω have the same value $\forall i, j \in \omega : \boldsymbol{y}_i = \boldsymbol{y}_j$ and (b) all the rows indexed by ω coincide in only one column index: $|\bigcap_{i \in \omega} s_i| = 1$.

4. Let $a = \bigcap_{i \in \omega} s_i$ then update the estimate with $\hat{\boldsymbol{x}}_a \leftarrow \hat{\boldsymbol{x}}_a + \boldsymbol{y}_b$ where b can be any member of the set ω.

5. Update the sample values to $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{P}_{:,a}\boldsymbol{y}_b$.

6. Go to step 3 and continue this operation until $\boldsymbol{y}$ is an all zero vector.

Let us briefly explain the task of this algorithm. After initialization in the first two steps, the algorithm finds a subset of samples which all of them *vote* the same value and all of them span only one common signal coefficient $\boldsymbol{x}_a$. Since the projection matrix is binary, thus the algorithm assumes the same votes comes from $\boldsymbol{x}_a$. Consequently, the signal at that coordinate would be estimated as the votes. Similar to many message passing algorithm (e.g. LDPC codes [57, 58]) and after subtracting the effect of that estimation from the samples, the algorithm iterates the same procedure. It might be the case that the estimate is wrong. However, one would wish that the number of correct estimates is higher than the wrong ones and hence the algorithm would converge to the correct solution.

Here a few notes should be highlighted. The assumption that "if a number of samples have equal values and they span only one common signal coordinate, therefore that common coordinate has to be non-zero", is not valid and one can find many counter examples. For instance, if a number of signal coefficients are equal and there exists a subset of samples that span only one of those non-zeroes then those samples would be equal as well. In fact, that assumption is a necessary but not a sufficient condition, that is if a number of samples spans only one common non-zero then they would be equal. This problem is overcome in traditional combinatorial algorithm by choosing the threshold c (the minimum number of samples that vote the same) large enough to increase the level of confidence. However as we increase c, we have to increase the number of samples as well, since otherwise, no c samples can be found with the aforementioned properties. This observation justifies one of the reasons why in practice, combinatorial algorithms require a higher number of samples (m) when compared to other classes of CS decoders. Furthermore, if the samples are contaminated by noise then exactly equal samples would not be found. A remedy for this problem can be finding a subset of samples where they are within a small radius of each others. Although this correction might work where the noise has very small magnitudes, it usually fails. Consequently, designing robust combinatorial CS approaches still remains an open problem. Another negative result about the projection matrices of traditional combinatorial algorithm is that, those binary sparse matrices do not hold RIP [44]. Indeed, RIP is the most important reason why convex relaxation methods are stable under the presence of noise. Thus designing alternative robust combinatorial solvers seems challenging (if not impossible), if we keep the sampling process intact. Here we present our solution to these inherent problems with combinatorial CS approaches that employ binary sparse projection matrices.

1.6.1 Going to higher fields

The first strategy that we employ to improve the characteristics of combinatorial approaches is to replace binary entries of the projection matrix with numbers from larger fields, e.g. real or complex numbers. As we show later in this report, this would aid to make milder assumptions about the underlying signal for the purpose of recovery. On the other hand, this modification introduces other problems. For instance, even if a subset of samples span only one (unique) non-zero signal coefficient then those samples would not be equal. We propose a simple and efficient solution for this problem in the next chapter. More specifically, we (jointly) look at a subset of size $l < m$ samples as a vector in $\mathbb{H}^l$ where $\mathbb{H}$ is the field from which the entries of $\boldsymbol{P}$ are. Then, we look for a (subset of) columns of the projection matrix which aligns in that direction. Furthermore, we can carefully choose entries of the projection matrix such that sub-matrices of $\boldsymbol{P}$ would have a lexical order for the decoder. This technique reduces the complexity of our solver significantly as we show in the following chapters.

1.6.2 Augmenting with dense random rows

As noted before, there are interesting links between the problem of CS and channel coding. This link is more strong when we consider the class of combinatorial algorithms. It has been shown in some codes [50], for instance the robust soliton distribution in LT codes that adding "dense encoding symbols" helps in the stability of the decoder in deviating from the average behavior. We consider this option in the design of our projection matrix where it might be the case that some random dense rows would be added to the projection matrix. Finding the optimal ratio of dense and sparse rows (in $\boldsymbol{P}$) is one of our objectives.

1.6.3 Robust and deterministic solver

Recall that in classical combinatorial algorithms some wrong decisions might happen in finding the support of the signal. These wrong decisions would be corrected in a message passing algorithm. However, this permission (of wrong decisions) leads to extra iterations of the algorithm. In this thesis, we plan to design our solver such that all the decisions would be correct which translates into decreasing the complexity of the solver and running time. As we show later in this report, we require a RIP condition on the projection matrix to guarantee that no mistake can be made during the recovery. Meanwhile RIP translates into robustness (in the presence of noise) at least for convex relaxation algorithms. We present our numerical experiments that our proposed projection matrices holds RIP and provide guidelines how to prove that claim. Hence, we can hope for a robust optimal combinatorial CS framework.

1.6.4 Changing the problem formulation

Another contribution of this work is to introduce a systematic formulation for the problem of CS under a combinatorial framework. In our formulation (as we present in the next chapter), we show that combinatorial algorithms are in fact a divide and conquer approach to find the sparsest solution to an under-determined system of equations. Then in the last chapter of this report, we present necessary and sufficient conditions for convergence and robustness.

1.7 Proposal organization

This thesis has two major parts. In the first part, we solve the problem of CS in the ideal case. i.e. the signals of interest are exactly sparse and the measurements are noise free. Furthermore, some of the analysis in the first part predicts only the average behavior of the proposed schemes. The second part of this thesis includes more realistic scenario where the signals are no longer exactly sparse and measurements are noisy. Also all the proofs will hold true either with probability one (either strictly or asymptotically) or with overwhelming probability. Nevertheless, we review the following chapters in here:

- In Chapter 2, we introduce “hybrid projection matrices” as a candidate for projection matrices in the problem of CS. This class of matrices is composed of two parametric dense and sparse matrices. Then we present our generic solver for this class of matrices.
- In Chapter 3, we consider the most simple case of hybrid projection matrices where rows of the spars sub-matrix span disjoint coordinates of the signal. We derive a bound on the sample requirements, optimal ratio of the dense rows to sparse rows and complexity of the scheme. Our simulation results also verify that this simple scheme, in certain applications,

can outperform many state of the art algorithms in the ideal case.

- In Chapter 4, we consider the case where rows of the sparse sub-matrix of our projection matrix coincide in some column indices, however there is no dense submatrix present in the projection matrix. Two interesting types of incidence are those from the adjacency matrices of (a) expander graphs [63] and (b) block designs, especially Balanced Incomplete Block Designs or BIBD [49]. We show that expander graphs lead to an optimal (in terms of sampling requirements) scheme.

- Chapter 5 is devoted to future works and our plans to extend our work to the case of compressible signals and noisy measurements.

Chapter 2

Compressed sensing under hybrid projections

2.1 Recursive approach for CS

In this section, we present our systematic approach to the problem of CS. Although most combinatorial approaches follow a similar path, to the best of our knowledge, the abstract formulation presented below is novel and unique. For the simplicity of notation and without any loss of generality, assume the signal of interest is sparse in the standard basis $\boldsymbol{\Psi} = \boldsymbol{I}_n$. That is the sensing matrix and the projection matrix are the same $\boldsymbol{P} = \boldsymbol{\Phi}\boldsymbol{\Psi} = \boldsymbol{\Phi}$ and $\boldsymbol{x} = \boldsymbol{s}$.

Consider two disjoint non-empty sets of $T \subset [n]$ and T^c where by definition $T \cup T^c = [n]$. For all vectors $\boldsymbol{x} \in \mathbb{R}^n$ and norms ℓ_p we have:

$$\|\boldsymbol{x}\|_p^p = \|\boldsymbol{x}_T\|_p^p + \|\boldsymbol{x}_{T^c}\|_p^p$$

However norms of ℓ_0 and ℓ_1 are special since the equality changes to:

$$\|\boldsymbol{x}\|_0 = \|\boldsymbol{x}_T\|_0 + \|\boldsymbol{x}_{T^c}\|_0, \; \|\boldsymbol{x}\|_1 = \|\boldsymbol{x}_T\|_1 + \|\boldsymbol{x}_{T^c}\|_1$$

and a similar equality generally does not hold true for other norms. We show that this special property of ℓ_0 and ℓ_1 enables us to follow a divide and conquer approach to the problem of CS.

Recall that the goal of CS is to find the vector $\boldsymbol{x} \in \mathbb{R}^n$ with minimum sparsity and which is the solution to the under-determined system of linear equations $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} \in \mathbb{R}^m$ where $m < n$, i.e.:

$$\boldsymbol{x} = \arg\min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{x}}\|_0 : \; s.t. \; \boldsymbol{y} = \boldsymbol{P}\hat{\boldsymbol{x}} \tag{2.1}$$

One might follow a divide and conquer approach to this problem where coordinates of $\boldsymbol{x}$ are partitioned into two disjoint non-empty sets of $T, T^c \subset [n]$ and then solve the following problem:

$$\min \|\boldsymbol{x}_T\|_0 + \min \|\boldsymbol{x}_{T^c}\|_1, \; s.t. \; \boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} \tag{2.2}$$

It can be easily verified (e.g. by contradiction) that the solution to this optimization problem equals to the solution to the original format of CS (2.1). This divide and conquer approach has all advantages of a recursive program. For instance, after dividing the signal coordinates into two disjoint sets T and T^c, one might partition each of these two sets into two more disjoint non-empty subsets. This division can be continued until we reach to a point where finding the sub-solution with the minimum ℓ_0 can be done tractably and then traverse that tree upward. However, this approach implies some undesirable conditions on the projection matrix. More specifically, to implement this divide and conquer algorithm even in the first level, we require that the sample $\boldsymbol{y}$ can be divided into disjoint subsets Q and Q^c where samples $\boldsymbol{y}_Q$ only span[1] $\boldsymbol{x}_T$ and $\boldsymbol{y}_{Q^c}$ only span $\boldsymbol{x}_{T^c}$. This is the case, since otherwise the sub-problems ($\min \|\boldsymbol{x}_T\|_0$ and $\min \|\boldsymbol{x}_{T^c}\|_0$) would not be independent and a divide and conquer approach will not work. Thus in this case, there exist

[1] Recall that we say the i-th sample spans coordinates ω if $\omega = \{j : \boldsymbol{P}_{i,j}\}$.

permutation matrices $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ such that:

$$\mathbf{\Pi_1}\boldsymbol{P}\mathbf{\Pi_2} = \begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B} \end{bmatrix} \tag{2.3}$$

where $\boldsymbol{A} \in \mathbb{R}^{|Q|\times|T|}$ and $\boldsymbol{B} \in \mathbb{R}^{|Q^c|\times|T^c|}$. Unfortunately, this highly structured matrix most likely does not hold RIP [3, 16] and hence it might lead into a non robust solution. To avoid this problem we have to consider some overlaps among the span sets of different samples.

Knowing that samples span overlapping subsets of the signal coordinates, we slightly change the formulation of (2.2) to the following:

$$\min \|\boldsymbol{x}_{T\cup a}\|_0 + \min \|\boldsymbol{x}_{T^c\cup a}\|_1, \; s.t. \; \boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} \tag{2.4}$$

where a is a (small) non-empty subset of $[n]$. Generally, the solution to this problem is not equal to the original problem of (2.1) unless $\boldsymbol{x}_a$ is an all zero vector. In this paper, we attempt to solve the problem of CS using this formulation. That is find two overlapping subsets of signal coefficients, say $T \cup a$ and $T^c \cup a$ such that $\boldsymbol{x}_a$ is zero and solve each of two sub-problems independently.

Suppose the i-th sample ($i \in [m]$) spans indices s_i of a k-sparse signal of length n. Without loss of generality, assume that support of the signal is distributed uniformly. Therefore, the probability that $\boldsymbol{y}_i$ spans j non zeros is:

$$\Pr\left(|s_i \cap Supp(\boldsymbol{x})| = j\right) = \frac{\binom{k}{j}\binom{n-k}{|s_i|-j}}{\binom{n}{|s_i|}} \tag{2.5}$$

which is a hypergeometric distribution with the mean of $|s_i|k/n$. Note that for a dense matrix where all entries are non-zero $|s_i| = n$, we have

$$\Pr\left(|s_i \cap Supp(\boldsymbol{x})| = k\right) = 1$$

(meaning that all samples spans all non-zeros), which is not surprising! However, for sparse matrices where $|s_i| \ll n$ and $k \ll n$ this probability (and the corresponding mean) is very small. In other words, for a sparse projection matrix (even considering the variance and deviation from the mean) most samples span only a few non-zeros (if any).

It turns out that finding samples which spans only zero valued coefficient is not trivial. Since otherwise, we would have known the support of the signal and the problem has been already solved by forming a full-rank system of equations. As we show in the following chapters finding *isolated* non-zero coefficients seems to be a simpler task. We say that the non-zero coefficient $\boldsymbol{x}_i$ is isolated in the j-th sample if: $Supp(\boldsymbol{x}_{s_j}) \cap s_j = i$, i.e. the only non-zero coefficient sampled by $\boldsymbol{y}_j$ is $\boldsymbol{x}_i$. In the scenario when the isolated non-zeros can be identified , one may apply the following algorithm to solve the problem of Compressed Sensing.

```
Algorithm 1: A recursive decoder $\hat{\boldsymbol{x}} = \Delta_s(\boldsymbol{P}, \boldsymbol{y})$
    input  : Samples $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$, Projection matrix $\boldsymbol{P}$
    output: $\hat{\boldsymbol{x}}$
    Find a (number of) sample(s) $\boldsymbol{y}_j$ that spans an isolated non-zero, for instance $\boldsymbol{x}_i$ ;
    Let $a$ be the estimate of that isolated non-zero;
    $\hat{\boldsymbol{x}}_i \leftarrow a$;
    $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{P}_{:,i}a$ (subtract the effect of estimation from samples);
    $\boldsymbol{P} \leftarrow \boldsymbol{P}_{[m]\backslash j,[n]\backslash s_j}$, $\boldsymbol{y} \leftarrow \boldsymbol{y}_{[m]\backslash j}$ (throw away used samples and respective rows and
    columns of the projection matrix);
    $\hat{\boldsymbol{x}}_{[n]\backslash s_j} \leftarrow \Delta(\boldsymbol{P}, \boldsymbol{y})$;
```

Note that Algorithm 1 is exactly our previous recursive formulation (2.4) when $\|\boldsymbol{x}_T\|_0 = 1$. The same algorithm can be expressed in a non-recursive formulation which is presented in

Algorithm 2.

Algorithm 2: Non recursive formulation of decoder $\hat{\boldsymbol{x}} = \Delta_s(\boldsymbol{P}, \boldsymbol{y})$

```
input  : Samples $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$, Projection matrix $\boldsymbol{P}$
output: $\hat{\boldsymbol{x}}$
while $\|\boldsymbol{y}\| \neq 0$ do
    Find a (number of) sample(s) $\boldsymbol{y}_j$ that spans an isolated non-zero, for instance $\boldsymbol{x}_i$ ;
    Let $a$ be the estimate of that isolated non-zero;
    $\hat{\boldsymbol{x}}_i \leftarrow \hat{\boldsymbol{x}}_i + a$;
    $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{P}_{:,i}a$;
end
```

2.2 Augmenting with dense rows

Algorithm 1 succeeds in recovering the sparse vector $\boldsymbol{x}$ if we always can find a number of samples that span an isolated non-zero. This is equivalent to guarantee the ripple in the LT codes never gets empty [50] before complete recovery. In LT codes, this guarantee comes from a carefully designed degree distribution over message and parity nodes. However unlike the channel codes, in CS we are not aware of the support of the signal. Consequently, it would be extremely difficult (if not impossible) to design span sets $\{s_i\}$ such that we can always find a number of samples spanning an isolated non-zero. To overcome this problem and for other reasons to be discussed in the following chapters, we augment the projection matrix with $m_d \geq 0$ random dense rows (hence Hybrid projection matrices). To avoid ambiguity, from now on, we use m_s to denote the number of sparse rows (samples) of a projection matrix and use m_d to denote the number of dense rows of the same projection matrix. Now, assume that after a number of iteration, the algorithm $\Delta_s(.,.)$ can not find any other isolated non-zero and halts. Furthermore, assume that up to that point signal values in indices of $\omega \subseteq [n]$ have been recovered and samples $\boldsymbol{y}_q$ coming from m_s sparse rows of the projection matrix have been used. If the number of undiscovered signal coefficients $n - |\omega|$ is less than or equal to the number of unused samples $m_d + (m_s - |q|)$, then we have a sufficient number of equations to determine the signal values in the remaining indices. Algorithm 3 summarizes this case.

Algorithm 3: The hybrid solver $\hat{\boldsymbol{x}} = \Delta_h(\boldsymbol{P}, \boldsymbol{y})$

```
input  : Samples $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$, Projection matrix $\boldsymbol{P}$
output: $\hat{\boldsymbol{x}}$
$\hat{\boldsymbol{x}} \leftarrow \Delta_s(\boldsymbol{P}, \boldsymbol{y})$;
Let $q \subseteq [m_s]$ be the sample indices which have been used from the sparse part;
Let $\omega \subseteq [n]$ be the indices where the signal has been recovered;
if $n - |\omega| \leq m_d + (m_s - |q|)$ then
    $\hat{\boldsymbol{x}}_{[n]\backslash\omega} \leftarrow \boldsymbol{P}^{\dagger}_{[n]\backslash\omega}(\boldsymbol{y} - \boldsymbol{P}\hat{\boldsymbol{x}})$;
end
```

2.3 Alignment

So far we have assumed that there exists an algorithm which can find isolated non-zeros. In here, we present a simple algorithm for such task. Assume $\boldsymbol{A} \in \mathbb{R}^{a \times b}$ and $\boldsymbol{z} \in \mathbb{R}^b$ is a vector of size b where $a < b$. Furthermore define $\boldsymbol{h} = \boldsymbol{A}\boldsymbol{z}$. We say that $\boldsymbol{h}$ is *aligned* with the i-th column of $\boldsymbol{A}$ if $\boldsymbol{h}$ is in the direction of $\boldsymbol{A}_{:,i}$, that is:

$$\frac{|\boldsymbol{A}_{:,i}^T \boldsymbol{h}|}{\|\boldsymbol{A}_{:,i}\|_2 \|\boldsymbol{h}\|_2} = 1 \tag{2.6}$$

Note that if $\|\boldsymbol{z}\|_0 = 1$, then $\boldsymbol{h}$ would be aligned with one of columns of $\boldsymbol{A}$ since:

$$\boldsymbol{z}_j = \begin{cases} 0 & j \neq i \\ \alpha & j = i. \end{cases} \Rightarrow \boldsymbol{h} = \alpha \boldsymbol{A}_{:,i} \tag{2.7}$$

Now if we can prove that alignment of $\boldsymbol{h}$ with a column of $\boldsymbol{A}$ means that $\|\boldsymbol{z}\|_0 = 1$

$$\boldsymbol{h} = \alpha \boldsymbol{A}_{:,i} \overset{?}{\Rightarrow} \boldsymbol{z}_j = \begin{cases} 0 & j \neq i \\ \alpha & j = i. \end{cases} \tag{2.8}$$

then we can utilize the concept of alignment in finding isolated non-zeros in the problem of CS. More specifically, we can substitute $\boldsymbol{A}$ with $\boldsymbol{P}_{\Omega,T}$, $\boldsymbol{h}$ with $\boldsymbol{y}_\Omega$ and $\boldsymbol{z}$ with $\boldsymbol{x}_T$ where $T \subseteq [n]$ and $\Omega \subseteq [m]$ is indices of the set of samples spanning $\boldsymbol{x}_T$ and then verify whether $\|\boldsymbol{x}_T\|_0 = 1$ or not.

In our simulations, we have observed that the aforementioned statement holds true in case of our projection matrices. Providing a rigorous proof to (2.8) is one of our future work. In Chapter 5, we present some guidelines on how we are going to prove it.

2.4 Lexical order

Indeed the concept of alignment for the matrix $\boldsymbol{A} \in \mathbb{R}^{a \times b}$ and a vector $\boldsymbol{h} \in \mathbb{R}^a$ makes sense[2] only when $a > 1$. As stated before, $\boldsymbol{A}$ shall be a sub-matrix of the projection matrix $\boldsymbol{P}$ and $\boldsymbol{h}$ will be subsets of compressive samples $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$. Clearly, we need some properties in the entries of the projection matrix to make the process of verifying the alignment, tractable. Since otherwise, we have to check all sub-matrices of the projection matrix and respective subsets of samples to find alignments, which has an exponential complexity. This problem can be solved by either of these two strategies:

1. Slightly modifying the decoder $\Delta(.,.)$.

2. Using lexical order, meaning that columns of the projection matrix have a logical order to

[2] We always have alignment for a=1.

the decoder. That is we can find a *scoring* function $\mathcal{S} : \mathbb{R}^b \to \mathbb{R}$ such that:

$$i, j \in [b], i < j : \mathcal{S}\left(\boldsymbol{A}(:, i)\right) < \mathcal{S}\left(\boldsymbol{A}(:, j)\right) \tag{2.9}$$

In the latter case, a quick search algorithm, similar to Algorithm 4 might be used to find the existence of an alignment. Examples for such concepts will be presented with further details in the following chapters.

Algorithm 4: $M = \mathcal{B}\left(\boldsymbol{A}, \boldsymbol{h}, l, u\right)$: A binary search for alignment when a lexical order exists

```
input  : $\boldsymbol{A}$, $\boldsymbol{h} = \boldsymbol{A}\boldsymbol{z}$, $l$, $u$
output: $M$ index of alignment
if $u < l$ then
    return -1 (no alignment found);
end
$M = l + [(u - l)/2]$ ;
if $\mathcal{S}\left(\boldsymbol{A}_{:,\mathcal{M}}\right) < \mathcal{S}\left(\boldsymbol{h}\right)$ then
    return $\mathcal{B}\left(\boldsymbol{A}, \boldsymbol{h}, M + 1, u\right)$ ;
end
if $\mathcal{S}\left(\boldsymbol{A}_{:,\mathcal{M}}\right) > \mathcal{S}\left(\boldsymbol{h}\right)$ then
    return $\mathcal{B}\left(\boldsymbol{A}, \boldsymbol{h}, 1, M - 1\right)$ ;
end
return M;
```

Chapter 3

An extreme case of hybrid projections

3.1 Introduction

In this chapter, we present Hybrid Compressed Sensing (HCS), an instance of class of hybrid projection matrices, presented in Chapter 2. HCS is an extreme example of such class since in its sparse sub-matrix, each sample spans disjoint subset of signal coefficients. As we show in this chapter, HCS is not able to achieve the optimal sample requirements of $m = O(k \log n/k)$. However, it can be proved that the non-optimal HCS outperforms even the state of the art solutions in certain settings of the signal length n and sparsity k.

3.2 Hybrid Compressed Sesning

Recall that, a matrix belonging to the class of hybrid projection matrices (after some permutation on its rows) can be decomposed into:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}^{(s)} \\ \boldsymbol{P}^{(d)} \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{P}^{(s)}$ is a $m_s \times n$ sparse and $\boldsymbol{P}^{(d)}$ is a $m_d \times n$ dense matrix. In HCS, we set the non-zero entries of $\boldsymbol{P}^{(s)}$ and $\boldsymbol{P}^{(d)}$ to be respectively complex and real numbers. Note that, there exists some devices such as radar or medical imaging which capture complex valued samples. Nevertheless, we count each complex valued compressive sample as two real samples (one for the imaginary part and one for the real part). Hence, the projection matrix $\boldsymbol{P}$ in (3.1) is sensing $m = 2m_s + m_r$ measurements.

For simplicity of analysis and without loss of generality assume m_s counts n. Now let $s = \{s_1, \ldots, s_{m_s}\}$ be a set of equal size, random subsets over $[n] = 1, 2, \ldots, n$ such that s forms a *partition*:

$$\forall i, j \in [n], i \neq j : |s_i| = w, s_i \cap s_j = \emptyset, \cup_{i=1}^{m_s} s_i = [n]$$

Then the i-th row of $\boldsymbol{P}^{(s)}$ is non-zero, only in the column indices of s_i. Also let $\boldsymbol{f}^{(i)}$ be a unit norm $1 \times w$ row vector with random complex entries having distinct phases in the range of $[0, \pi)$. Then we insert the entries of $\boldsymbol{f}^{(i)}$ in the i-th row of $\boldsymbol{P}^{(s)}$ in the column indices of s_i. This can be written as:

$$\forall i \in [m_s] : \boldsymbol{P}^{(s)}_{i,s_i} = \boldsymbol{f}^{(i)},\ \boldsymbol{P}^{(c)}_{i,[n]\setminus s_i} = \mathbf{0}$$

Thus the i-th sample ($i \in [m_s]$) is a function of the signal coefficients only in the indices of s_i: $\boldsymbol{y}_i = \boldsymbol{f}^{(i)} \boldsymbol{x}_{s_i}$. Appending m_d real-valued, dense random rows to this matrix yields the final projection matrix.

3.2.1 HCS Solver

Since there is no overlap among different rows of $\boldsymbol{P}^{(s)}$, finding the isolated non-zeros can not be robust unless we made some assumptions about the signal. Let us assume that the distribution of signal coefficients is such that for all $\Omega \subset [n]$ and $|\Omega| = w$, the inner product of $\boldsymbol{x}_\Omega$ with a unit norm random vector of complex numbers is not zero. Our simulations show that this assumption holds true for test signals and images in CS community. Using this assumption we have:

Proposition 1. Let $s_i \subset [n],\ |s_i| = w$ be the (ordered) set of indices of non-zero coefficients for the i-th row of a sparse matrix $\boldsymbol{P}^{(s)}$. And let $\boldsymbol{P}^{(s)}_{i,s_i} = \boldsymbol{f}^{(i)} = [e^{j\phi_1^{(i)}} \ \ldots \ e^{j\phi_w^{(i)}}]$ be the corresponding vector of complex exponentials (on the unit circle) such that the random phases of $\phi^{(\cdot)}_{\cdot}$ are distinct in the range of $[0, \pi)$:

$$\forall l, t \in [w], l \neq t : \phi_l^{(i)} \in [0, \pi), \phi_t^{(i)} \neq \phi_l^{(i)}$$

If $\boldsymbol{x}_{s_i}$ is real-valued and non-zero in at most one index ($\|\boldsymbol{x}_{s_i}\|_0 \leq 1$) then having $\boldsymbol{y}_i = \boldsymbol{P}^{(s)}_{i,:}\boldsymbol{x} = \boldsymbol{f}^{(i)}\boldsymbol{x}_{s_i}$ and $\boldsymbol{f}^{(i)}$, one can recover $\boldsymbol{x}_{s_i}$ deterministically. Moreover having $\boldsymbol{y}_i$ and $\boldsymbol{f}^{(i)}$, we can detect the event that $\|\boldsymbol{x}_{s_i}\|_0 \geq > 2$. Specifically if $\boldsymbol{y}_i = 0$ then $\boldsymbol{x}_{s_i} = \boldsymbol{0}$. If the phase of $\boldsymbol{y}_i$ equals to the phase of l-th entry of $\boldsymbol{f}^{(i)}$ ($\angle \boldsymbol{y}_i = \angle \boldsymbol{f}_l^{(i)} = \phi_l^{(i)}$) then $\boldsymbol{x}_{s_i}$ is non-zero only in the index of $s_{i,l}$ and has the value of $\boldsymbol{x}_{s_{i,l}} = \boldsymbol{y}_i / \boldsymbol{f}_l^{(i)} = \|\boldsymbol{y}_i\|$ where $s_{i,l}$ is the l-th member of the ordered list s_i. If the phase of $\boldsymbol{y}_i$ is not among the phases of $\boldsymbol{f}^{(i)}$ ($\{\phi_1^{(i)}, \ldots, \phi_w^{(i)}\}$), then $\boldsymbol{x}_{s_i}$ is non-zero in at least two indices.

Recall that hybrid solvers have two major stages: 1) finding and recovering subsets of the signal coefficients spanning at most one non-zero coefficient and 2) forming a full-rank system of linear equations and solving it. For the first m_s complex valued compressive samples ($\boldsymbol{y}_i, i \in [m_s]$), the solver looks for the phase of each compressive sample ($\angle \boldsymbol{y}_i$) among the available phases in the associated random vector. As before let $\boldsymbol{f}^{(i)} = \boldsymbol{P}^{(s)}_{i,s_i}$, $w = |s_i|$ and define:

$$\angle \boldsymbol{f}^{(i)} \triangleq [\angle \boldsymbol{f}_1^{(i)} \ \ldots \ \angle \boldsymbol{f}_w^{(i)}]$$

Assume the phase of sample $\boldsymbol{y}_i$ equals to the phase of t-th entry of $\boldsymbol{f}^{(i)}$. Then under our assumptions and by Proposition 1, $\boldsymbol{x}_{s_i}$ is zero in all indices except in the index of $s_{s,t}$. This can be written as:

$$\angle \boldsymbol{y}_i = \angle \boldsymbol{f}_t^{(i)} \Leftrightarrow \boldsymbol{x}_l = \left\{ \begin{array}{cc} \|\boldsymbol{y}_i\| & l = s_{i,t} \\ 0 & l \in s_i \backslash t \end{array} \right\} \tag{3.2}$$

Moreover, if the compressive sample $\boldsymbol{y}_i$ is zero, then $\boldsymbol{x}_{s_i}$ is a zero vector. Otherwise (if the sample is neither zero nor its phase is found in $\angle \boldsymbol{f}^{(i)}$), this subset of the coefficients ($\boldsymbol{x}_{s_i}$) spans at least two non-zeros and Proposition 1 is not applicable to this subset. Also note that, the idea of finding the phase of a compressive sample among the available phases in a row of the projection matrix is an example of the concept of the alignment, introduced in the previous chapter. More specifically, in the formulation of (2.6), we can re-cast the proposition 1 as finding the alignment of the vector:

$$\boldsymbol{h} = \begin{bmatrix} \Re\left(\boldsymbol{y}_i\right) \\ \mathcal{I}\left(\boldsymbol{y}_i\right) \end{bmatrix} \tag{3.3}$$

in the matrix of:

$$\boldsymbol{A} = \begin{bmatrix} \Re\left(\boldsymbol{P}_{i,s_i}\right) \\ \mathcal{I}\left(\boldsymbol{P}_{i,s_i}\right) \end{bmatrix} \tag{3.4}$$

where for a complex number $c = a + bi$ where $i^2 = -1$, $\Re(c) = a$ and $\mathcal{I}(c) = b$.

Define I' as the set of sample indices, such that each of these samples is either zero or the phase of that sample can be found among the available phases in the respective random vector:

$$I' = \left\{ i \in [m_s] : \boldsymbol{y}_i = 0 \; or \; \angle \boldsymbol{y}_i \in \angle \boldsymbol{f}^{(i)} \right\} \tag{3.5}$$

Thus the first stage of the HCS solver determines signal values on indices of:

$$I = \cup_{i \in I'} s_i \tag{3.6}$$

Consequently $\boldsymbol{P}_{:,I}\hat{\boldsymbol{x}}_I$ is the effect of identified coefficients on the compressive samples $\boldsymbol{y}$; and by subtracting $\boldsymbol{P}:, I\hat{\boldsymbol{x}}_I$ from the compressive samples $\boldsymbol{y}$ we are nullifying such effect:

$$\boldsymbol{y} - \boldsymbol{P}_{:,I}\hat{\boldsymbol{x}}_I = \boldsymbol{P}_{:,I^c}\hat{\boldsymbol{x}}_{I^c}$$

where I^c is the set complement of I ($I \cup I^c = [n]$). If we consider the n signal coefficients as unknowns and compressive samples as equations, the problem of $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$ can be viewed as a system of n unknowns and $m = 2m_s + m_r$ equations where $|I|$ coefficients have already been recovered in the first stage of the solver. Having $n - |I|$ further (independent) equations one can recover the remaining unknowns. The random dense submatrix $\boldsymbol{P}^{(d)}$ and complex samples spanning at least two non-zeros provide us such equations.

3.3 Optimal ratio of m_s and m_d under HCS

In this section, we derive the sample requirements of HCS (under perfect recovery) and also the complexity of such procedure. To that end, we first find the average number of samples required for the perfect recovery and then using the tools from the concentration of measurement [59] we show that our derived bound is tight.

Recall that complex compressive samples span disjoint subset of coefficients. Hence it is straightforward to see that the distribution of number of non-zeros spanned by each sample is multinomial. The problem of finding this multinomial distribution can be casted as the classical "balls into bins" problem [59]. Hence the probability of the event that a complex compressive sample (say $\boldsymbol{y}_i, i \in [m_s]$) spans j non-zero(s) is:

$$\Pr\left(\|\boldsymbol{x}_{s_i}\|_0 = j\right) = \binom{k}{j}\left(1 - \frac{1}{m_s}\right)^{k-j}\left(\frac{1}{m_s}\right)^j \tag{3.7}$$

Consequently, the expected number of samples spanning at most one non-zero coefficient ($|I'|$) for large values of k is:

$$E(|I'|) = m_s \Pr\left(\|\boldsymbol{x}_{s_i}\|_0 \leq 1\right) \approx m_s\left(1 + \frac{k}{m_s}\right)e^{-\frac{k}{m_s}} \tag{3.8}$$

Since each complex sample spans $w = n/m_s$ coefficients, the number of identified coefficients (I) on average is:

$$E(|I|) = wE(|I'|) = n\left(1 + \frac{k}{m_s}\right)e^{\frac{-k}{m_s}} \tag{3.9}$$

Note that we have not utilized $m_s - I'$ complex samples in the first stage of the algorithm since these samples span at least two non-zeros. Counting each complex sample as two equations, we need at least $m_d \geq n - E(|I|) - 2\left(m_s - I'\right)$ further equations to form a full rank system of equations to recover the remaining coefficients. To identify the required number of measurements, this problem can be casted as the following optimization problem:

$$\underset{m_s, m_d}{\arg\min}\ 2m_s + m_d \ s.t.\ m_d \geq n - E(|I|) - 2\left(m_s - I'\right) \tag{3.10}$$

The following two lemmas outline key results for solving this problem and determine the sample requirements of the HCS solver. The first lemma provides expressions for the numbers of complex samples m_s and real samples m_d that are required for the recovery of $\boldsymbol{x}$ under HCS. The second lemma simplifies these expressions and show that the HCS measurement bound is tight. A corollary is also stated below.

Lemma 3.3.1. *Consider a k-sparse signal $\boldsymbol{x}$ of length n and define $l = n/k$. Then on average, HCS requires*

$$m_s \approx k\left(\sqrt[3]{\frac{l}{2}} - \sqrt[3]{\frac{16}{729l}} - \frac{1}{3}\right) \tag{3.11}$$

complex-valued samples and

$$m_d = n - n\alpha - 2\alpha m_s \tag{3.12}$$

real-valued samples for the perfect recovery of $\boldsymbol{x}$ where $\alpha = (1 + k/m_s)\exp(-k/m_s)$.

Proof of Lemma 1.

To that end we solve the following optimization problem:

$$\operatorname*{arg\,min}_{m_s, m_d} 2m_s + m_d \; s.t. \; m_d \geq n - E(|I|) - 2\left(m_s - I'\right) \tag{3.13}$$

One can use the Lagrangian multiplier to solve (3.13). Define $\alpha = E(|I'|)/m_s < 1$, then the objective function (L) is:

$$L = 2m_s + m_d + \lambda\left(\alpha n + 2(1-\alpha)m_s + m_d - n\right) \tag{3.14}$$

Taking derivatives respect to three parameters λ, m_s and m_d gives us three equations:

$$\lambda = -1, \; \alpha n + 2(1-\alpha)m_s + m_d = n \tag{3.15}$$

$$\frac{\partial L}{\partial m_s} = 0 \Rightarrow 2m_s^3 + 2km_s^2 + 2k^2 m_s = nk^2 \tag{3.16}$$

Define $l = n/k$. Let us assume $l^2 \gg l$, then solving (3.16) yields:

$$m_s = k\left(\sqrt[3]{\frac{l}{2}} - \sqrt[3]{\frac{16}{729l}} - \frac{1}{3}\right) < k\sqrt[3]{\frac{l}{2}} \tag{3.17}$$

Hence we have:

$$m_d = n - n\alpha - 2\alpha m_s \tag{3.18}$$

Assuming $\sqrt[3]{2l^2} < 2 + 1/(1-\alpha)$ implies $m_d < m_s$. ■

Lemma 3.3.2. *If $n\left(2(\alpha - \alpha^2)^2 \ln k\right)^3 \ll k^4$ then with a probability of at least $1 - \frac{2}{k}$, $m = 2m_s + m_d$ compressive samples is sufficient for the perfect recovery of $\boldsymbol{x}$ under HCS, where $m_s = k(1+\delta)\sqrt[3]{\frac{n}{2k}}$, $m_d = n - n\alpha - 2\alpha m_s$ and $0 < \delta \ll 1$.*

Proof of Lemma 2.

We show that deviations from our derived bound is insignificant by following these steps:

1. We use Corollary 5.11 in [59] which in case of our problem, states that the distribution of number of non-zeros spanned by each sample is approximately Poisson with mean k/m_s.

2. Assuming independent Poisson distributions of non-zero coefficients in compressive samples, we compute the probability that a given sample spans more than one non-zero and find its variance and mean. Further one can easily find μ, the average number of samples spanning at least two non-zero coefficients.

3. Assuming independent Poisson distributions of non-zeroes in the compressive samples, let γ be the number of samples spanning at least two non-zero coefficients, we find the minimum t such that: $\Pr(\gamma - \mu \geq t) < 1/k$. In words, t (with high probability) is the maximum deviation of γ from μ. Then we show that under a realistic presupposition, we have: $t \ll k$.

4. Applying Corollary 5.11 from [59] to the value of t computed in the step 3, we have: in the actual distribution of non-zero coefficients among the partitions (i.e. multinomial), the maximum number of samples spanning at least two non-zeros is less than $\mu + t$ with a probability of higher than $1 - \frac{2}{k} = 1 - O(1/k)$.

Let us assume $k' = k'_1, \ldots, k'_{m_s}$ represents the number of non-zeros spanned by each complex compressive sample in Poisson case. Let the binary random variable ν_i be one when $k'_i > 1$ and zero otherwise. Then $\Pr(\nu_i = 1) = 1 - \alpha$. Hence $E(\nu_i) = 1 - \alpha$ and $\sigma^2(\nu_i) = \alpha(1 - \alpha)$. Define $\sigma^2 = \sum(\sigma_i^2)/m_s = \sigma_i^2$. By Bennet's inequality [59] we have:

$$\Delta = \Pr\left(\sum_{i=1}^{m_s} (\nu_i - E(\nu_i)) > t\right) \leq \exp\left(-m_s\sigma^2 h\left(\frac{t}{m_s\sigma^2}\right)\right)$$

where:

$$h(u) = (1+u)\log(1+u) - u > \frac{3u^2}{6+2u} \tag{3.19}$$

for $u \geq 0$. Solving $\Delta \leq 1/k$ gives:

$$t \approx \sqrt{2\sigma^2 m_s \ln k}$$

It is straightforward to see that when

$$n\left(2\sigma^2 \ln k\right)^3 \ll k^4 \tag{3.20}$$

then we have $t \ll k$ and this completes the proof. ■

Corollary 3.3.3. *If* $\sqrt[3]{2l^2} \leq 2 + 1/(1-\alpha)$ *then* $m_d \leq m_s$.

Although the HCS sampling bound is not optimal for all values of k and n, it outperforms even the most complex CS solutions over a wide range of typical and practical values of k and n (e.g., as long as k/n is not *excessively* small). Specifically let $m^\star = Ck \log n/k$ be the sample requirement for an optimal CS framework where C is a constant only depending on the projection matrix and the solver. If $\sqrt[3]{n/k} < 1.25C \log(n/k)$, then HCS requires fewer samples for perfect recovery when compared with the sampling requirement of the hypothetical optimal solver. Now let us compute the complexity of the HCS solver. Recall that the HCS decoding process consists of two stages. In the first stage, for each complex valued compressive sample ($\boldsymbol{y}_i$, $i \in [m_s]$), we look for the phase of that sample ($\angle \boldsymbol{y}_i$) among the available phases in the respective random vector $\angle \boldsymbol{f}^{(i)}$. Assume the entries of

$$\boldsymbol{f}^{(i)} = [\exp(j\phi_1^{(i)}) \ \exp(j\phi_2^{(i)}) \ \ldots \ \exp(j\phi_w^{(i)})]$$

are sorted based on phases:

$$a, b \in [w], a > b \Leftrightarrow \angle \boldsymbol{f}_a^{(i)} = \phi_a^{(i)} > \boldsymbol{f}_b^{(i)} = \angle \phi_b^{(i)} \tag{3.21}$$

Then the complexity of this search for each compressive sample is only $\log(w)$. Since we have to repeat this search for all complex valued compressive samples, thus the complexity of the first stage of the solver is $O(m_s \log(w))$. As before, let $|I'| \geq 0$, denotes the number of compressive samples spanning at most one non-zero coefficient, then we have not utilized $m_s - |I'|$ complex sample from the first stage of the solver. The second stage of the HCS solver finds the solution of a full rank system of $m_d + m_s - |I'| \leq 2m_s$ equations. Let $g(\beta)$ be the complexity of linear solver employed in the second stage of HCS solver to solve a full rank system of β equations. Then the complexity of the second stage of HCS is:

$$O\left(g(m_d + m_s - |I'|)\right) \leq O\left(g(2m_s)\right) \tag{3.22}$$

Finally the complexity of HCS solver is $O\left(\max\{m_s \log(n/m_s), g(2m_s)\}\right)$.

As a final remark, note that the phases of sample in here are the lexical order concept which we present in the previous chapter. More specifically in case of HCS and based on formulation (2.9) we have:

$$\boldsymbol{h} \in \mathbb{C} : \mathcal{S}\left(\mathbf{h}\right) = \angle \boldsymbol{h} \tag{3.23}$$

3.4 Simulation Results

We tested the proposed HCS framework on a large number of standard compressed sensing signals and images. Here, we present the results for images of Monalisa and Lenna and compare our results with two dominant solvers/projection matrices (Fig. 3.1 and Fig. 3.2).

We performed compressive sampling on the whole image of Monalisa, while for Lena, we applied a block based compressed sensing. More specifically for Lena, the target image is formed by keeping the largest $k = 8$ DCT coefficients in all 8×8 blocks and set the rest of DCT coefficients to zero. For Monalisa image, a total of $k = 57979$ DCT coefficients (with the largest absolute value) out of $n = 283554$ coefficients have been kept and we set the remaining coefficients to zero. In our simulations, we counted each complex valued compressive sample as two real samples. In these figures m, t_{bp}, t_{hcs} and t_{omp} represent the number of (real) compressive samples and the decoding time required for BP, HCS and OMP respectively. In all simulations, we have assumed that for a given number of samples m, the ratio of complex valued compressive samples and real

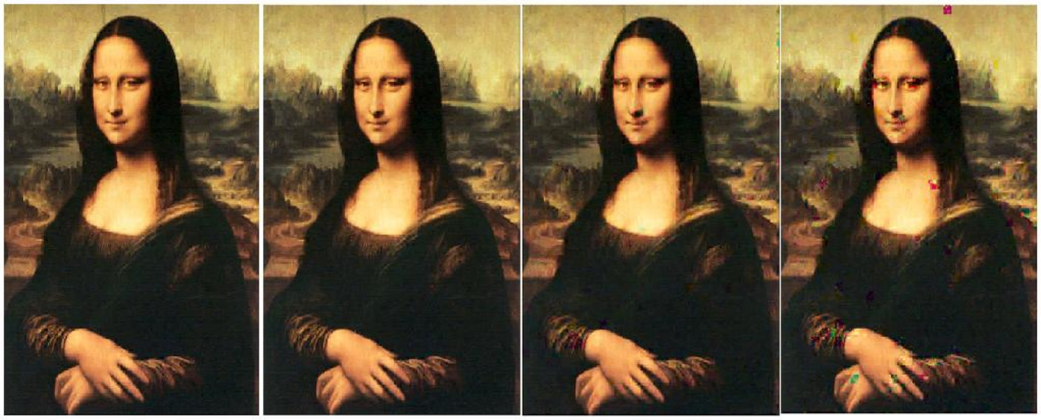


Figure 3.1: From left to right, the k-sparse image of Monalisa, reconstructed by applying HCS, BP and OMP respectively to the whole image ($m = 150746$). The decoding times are $t_{bp} = 109.5077$, $t_{hcs} = 16.1704$ and $t_{omp} = 15.1670$ seconds.

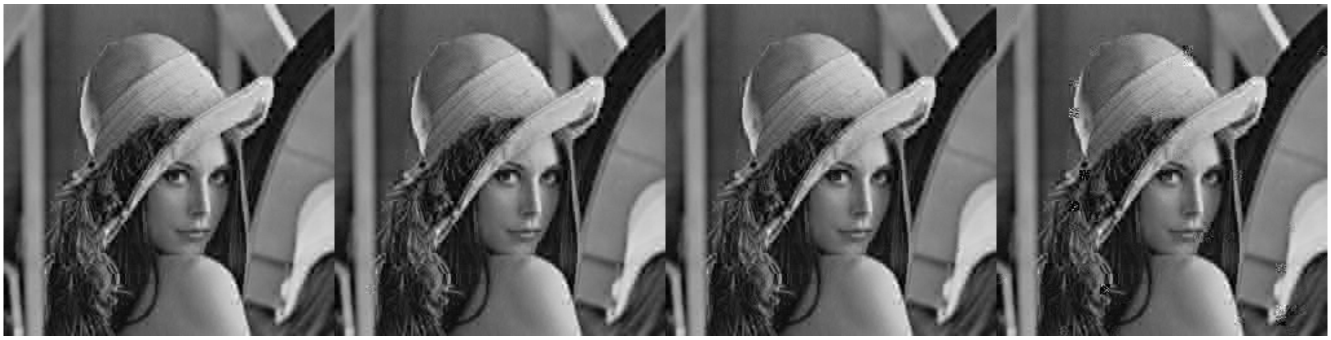
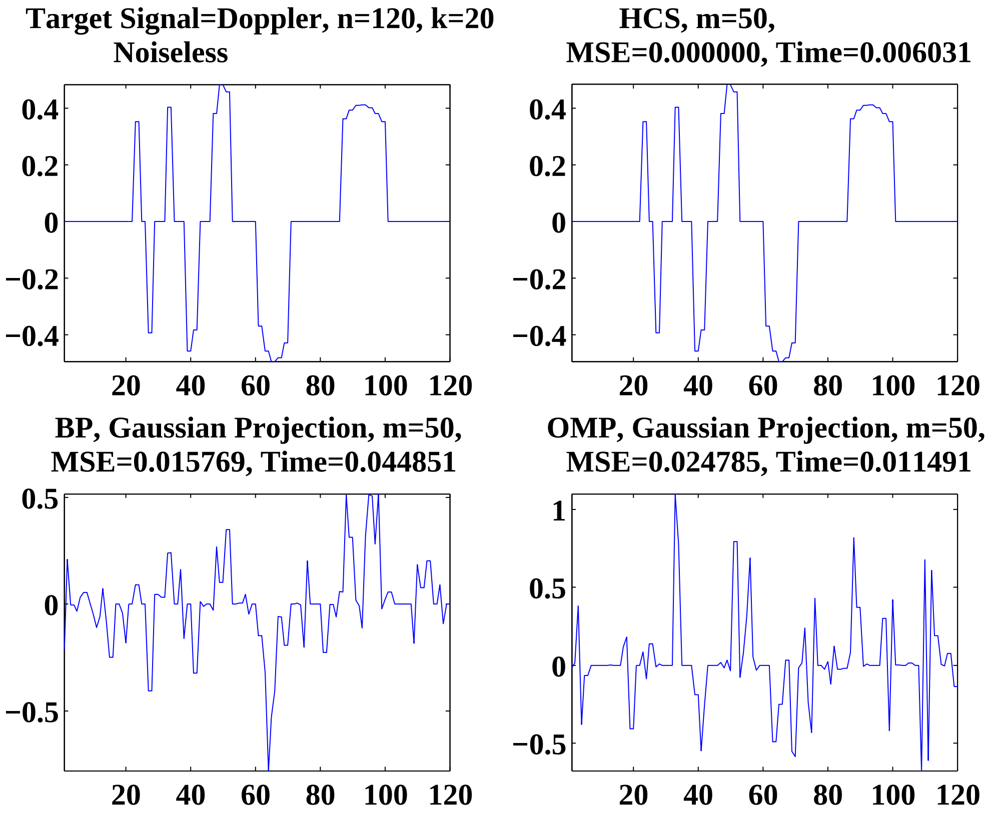


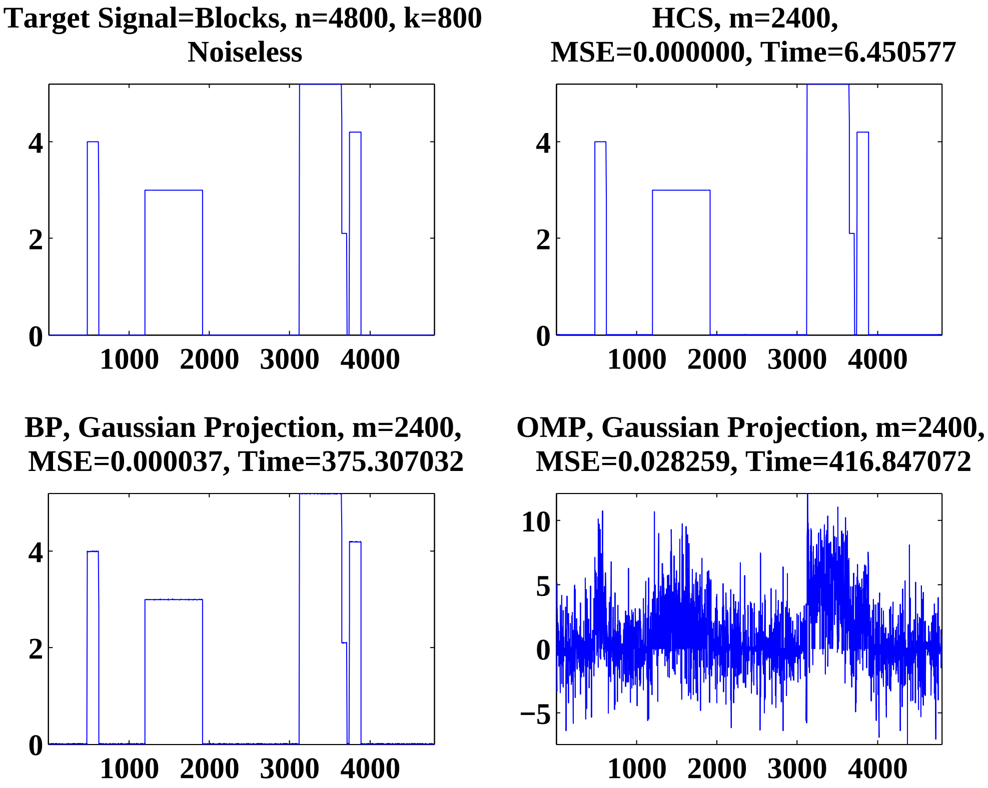
Figure 3.2: From left to right, the k-sparse image of Lenna, reconstructed with $m = 28$ compressive samples by applying HCS, BP and OMP respectively on blocks of size 8×8 pixel. The decoding times are $t_{bp} = 41.0283$, $t_{hcs} = 4.4735$ and $t_{omp} = 1.1512$ seconds.

valued dense samples (for HCS) is approximately one: ($m_s = \lceil m/3 \rceil$ and $m_d = m - 2m_s$). It is important to note that this sample assignment is not optimal and HCS needs fewer samples for the perfect recovery of the signal (see Lemma 1). However in some real world application, the number of non-zero coefficients might not be known beforehand. Although BP achieved virtually perfect reconstruction, but clearly the complexity of BP is much higher compared to HCS. Further simulation results in case of 1D signals are presented in Fig. 3.3 - Fig. 3.5.



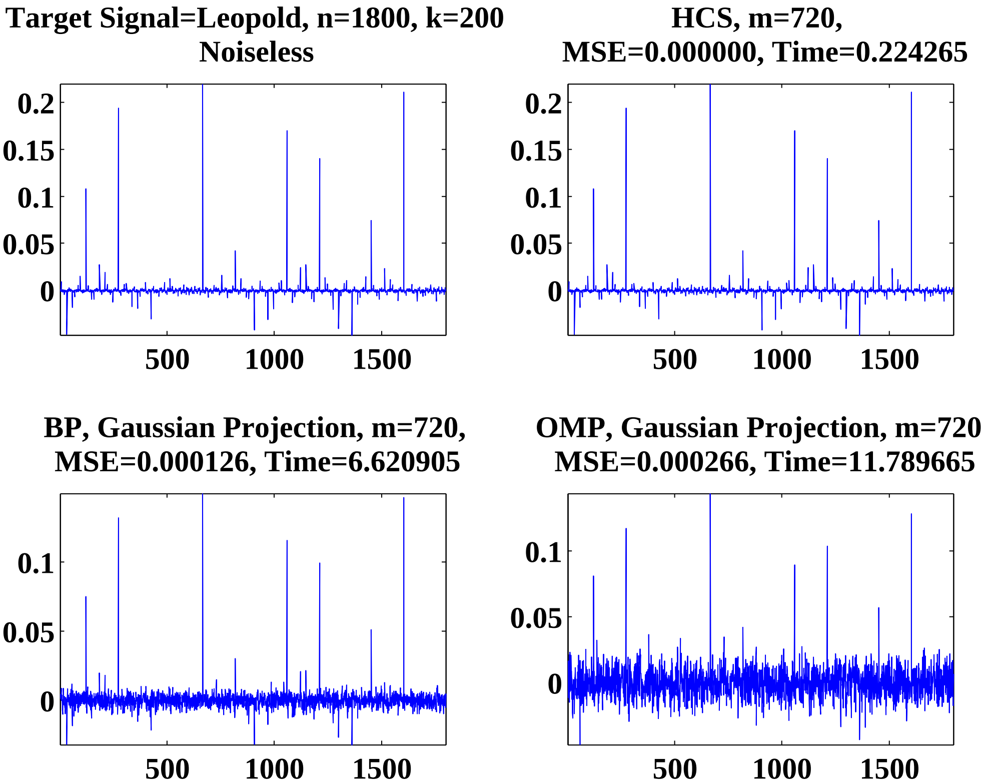
(a)

Figure 3.3: m, n and k represent number of samples, signal length and signal sparsity respectively.



(a)

Figure 3.4: m, n and k represent number of samples, signal length and signal sparsity respectively.



(a)

Figure 3.5: m, n and k represent number of samples, signal length and signal sparsity respectively.

Chapter 4

Complex Sparse Projections for Compressed Sensing (CRISP)

4.1 Introduction

In this chapter, we present Complex Sparse Projections for Compressed Sensing (CRISP), another instance of hybrid projection matrices. There are two major differences between CRISP and HCS, presented in the previous section:

- There are no dense random rows in the projection matrix, i.e. $\boldsymbol{P} = \boldsymbol{P}_S$.
- There are overlaps among rows of the (sparse part of) the projection matrix.

We show in this chapter that if overlaps among rows of the projection matrix are carefully designed, then CRISP would be an optimal CS framework.

4.2 Forming the projection matrix

Using an auxiliary matrix $\bar{\boldsymbol{P}}$ would be beneficial to describe properties of CRISP projection matrices. Let us use the term *base projection matrix* for such matrix $\bar{\boldsymbol{P}}$. Under the proposed CRISP framework, the base projection matrix $\bar{\boldsymbol{P}}$ has binary entries (ones and zeros), and hence, this base matrix provides an underlying structure that can be used to populate the "ones" entries of $\bar{\boldsymbol{P}}$ with numbers from higher fields. Similar to HCS, we again use complex numbers in the final projection matrix. This process of replacing the "ones" of $\bar{\boldsymbol{P}}$ by complex numbers generates the final projection matrix $\boldsymbol{P}$. Consequently, the underlying "structure" (and associated sparsity) of both the base matrix $\bar{\boldsymbol{P}}$ and the final projection matrix $\boldsymbol{P}$ is the same. One straightforward approach for constructing $\bar{\boldsymbol{P}}$ is to randomly select the entries of $\bar{\boldsymbol{P}}$ where "ones" are placed (Algorithm 5). This simple approach does not impose any constraints on the underlying structure of $\bar{\boldsymbol{P}}$ except for the parameter K, which is the number of "ones" that are placed randomly in each column of $\bar{\boldsymbol{P}}$. Note that K has to be sufficiently small (relative to $m = 2m_s$ and n) in order to have a sparse underlying structure for the final projection matrix $\boldsymbol{P}$. Note that the resulting $\bar{\boldsymbol{P}}$ in this case might be consider as the incidence matrix of a bipartite left regular graph of degree K. Hence with high probability [37][62] $\bar{\boldsymbol{P}}$ corresponds to an *expander graph*. A bipartite graph with n left nodes, m right nodes and regular left degree K will be called a $(\alpha n, \beta c)$ expander, for some $0 < \alpha, \beta < 1$ if for every subset of left nodes ν with cardinality less than or equal to αn ($|V| \leq \alpha n$) the number of neighbors connected to ν is larger than $\beta c|\nu|$. That is

$$|\mathcal{N}(\nu)| > \beta c|\nu| \tag{4.1}$$

where $\mathcal{N}(\nu)$ is the set of neighbors of ν.

Algorithm 5: Generating the base projection matrix $\bar{\boldsymbol{P}}$, where $\mathbf{1}$ is an all one vector.

```
Input: $m$, $n$ and $K \geq 2$
Output: The base projection matrix $\bar{\boldsymbol{P}}$
for $i = 1$ to $n$ (all columns of $\bar{\boldsymbol{P}}$) do
    Select $K$ distinct random integers ($l = \{l_1, \dots, l_K\}$) uniformly from $[m] = \{1, \dots, m\}$
    For the $i$-th column of $\bar{\boldsymbol{P}}$, set the entries in the rows indexed by $l$ to one and put zero in other indices: $\bar{\boldsymbol{P}}_{e,l} = \mathbf{1}$
```

One might impose certain constraints on the underlying structure of the base projection matrix which leads to the same structure for $\boldsymbol{P}$. Most notably, approaches that are inspired by channel coding methods for constructing parity check matrices, and in particular ones related to Low-Density-Parity-Check (LDPC) codes can be used for the binary base matrix $\bar{\boldsymbol{P}}$. For instance, one might utilize the incidence matrices of Balanced Incomplete Block Designs (BIBD) [49] (if they exist for the pair of m_s and n) or adjacency matrix of expander graphs to improve the performance of the CRSIP framework as we explain further below in this paper.

In general, one can perform some form of matrix manipulation of $\bar{\boldsymbol{P}}$ prior to replacing the "ones" of $\bar{\boldsymbol{P}}$ onto some complex numbers. Further, one can have certain constraints on the complex numbers used in the projection matrix $\boldsymbol{P}$. In this paper, we focus on applying two steps to the base matrix $\bar{\boldsymbol{P}}$, random permutation and employing complex numbers with unique phases as in HCS. For example, the overall approach used when permuting (uniformly) the columns of $\bar{\boldsymbol{P}}$ and substituting the non-zero entries of $\bar{\boldsymbol{P}}$ by normalized complex numbers with unique phases in range of $[0, \pi)$ is shown below (Algorithm 6).

Algorithm 6: Generating the projection matrix from the base projection matrix.

```
Input: The base projection matrix $\bar{\boldsymbol{P}}$
Output: The projection matrix $\boldsymbol{P}$
Permute randomly the columns of $\bar{\boldsymbol{P}}$;
for $i = 1$ to $m$ (all rows of $\boldsymbol{P}$) do
    Let $s_i$ be the non-zero entries of the $i$-th row of $\bar{\boldsymbol{P}}$ and let $w = |s_i|$.
    Generate a row vector $\boldsymbol{f}^{(i)}$ composed of $w$ normalized complex numbers with unique
    phases in range of $[0, \pi)$.
    For the $i$-th row of $\boldsymbol{P}$, insert $\boldsymbol{f}^{(i)}$ in the column indices of $s_i$: $\boldsymbol{P}_{i,s_i} = \boldsymbol{f}^{(i)}$ and
    $\forall l \notin s_i : \boldsymbol{P}_{i,l} = 0$
```

Now we say few words regarding the BIBD design procedure. As we eluded above, the incidence matrix of a Balanced Incomplete Block Design [49] represents a special case for the design of the base projection matrix $\bar{\boldsymbol{P}}$. Here, we briefly review a few key properties of such matrices. Let m, n, K and λ be positive integers such that $m > K \geq \lambda$. The incidence matrix of a (m, n, K, λ)-

BIBD is a $m \times n$, matrix with the following properties:

- Each row is non-zero exactly in $w = \lambda(m-1)/(K-1)$ column indices. Also each of the $n = mw/K$ columns is non-zero exactly in K entries[1] and thus we have $\forall i, j \in [m] : w = |s_i| = |s_j|$. Note that this matrix corresponds to the (transpose of) incidence matrix of a bipartite left-regular of degree K and right-regular of degree w.

- If we consider two distinct rows of this matrix, both are non-zero in exactly λ columns. That is: $\forall i, j \in [m], i \neq j : |s_i \cap s_j| = \lambda$.

Due to the special characteristics (listed above) of the proposed projection matrix, the following properties hold:

1. Since each column of our projection matrix is non-zero in K row indices, hence each signal coefficient $\boldsymbol{x}_i$ appears exactly in K random different equations/samples.

2. If the base projection matrix $\bar{\boldsymbol{P}}$ is an incidence matrix of a BIBD then two distinct equations (compressive samples) have exactly λ common coefficients/unknowns.

4.3 CRISP Solver

Let us assume that the distribution of magnitudes of signal coefficients is such that for all $\Omega \subset [n]$ and $|\Omega| = w$, the inner product of $\boldsymbol{x}_\Omega$ with a unit norm random vector of complex numbers is not zero (which is true for test signals and images in CS community).

Similar to other combinatorial approaches, the decoding scheme of the CRISP solver is iterative due to the overlaps among rows of the projection matrix. We examine compressive samples to find

[1] Note that the size of support of a signal (or sparsity measure) is denoted by k, while K is a parameter of a BIBD.

some subsets of the signal spanning only one non-zero coefficient. This can be achieved by looking for the phases of the compressive samples:

$$\angle \boldsymbol{y} = \{\angle \boldsymbol{y}_1, \ldots, \angle \boldsymbol{y}_{m_s}\}$$

among the phases of the entries of the random vectors $\boldsymbol{f}^{(\cdot)}$. For instance, assume that $\angle \boldsymbol{y}_i$ equals to $\angle \boldsymbol{f}_l^{(i)}$. Therefore using Proposition 1 (in Chapter 3), one can determine the signal values in the i-th subset $\boldsymbol{x}_{s_i}$. Subtracting the effect of the identified non-zero coefficient from the compressive samples leads to the following new under-determined system of equations:

$$\boldsymbol{y}^{(1)} = \boldsymbol{y} - \boldsymbol{P}_{:,s_i}\hat{\boldsymbol{x}}_{s_i} = \boldsymbol{P}\left(\boldsymbol{x} - \boldsymbol{z}\right) \tag{4.2}$$

where for $i > 1$, $\boldsymbol{y}^{(i)}$ is the residue of the approximation in the i-th iteration of the solver and $\boldsymbol{z}$ is a vector with the same size of $\boldsymbol{x}$ such that $\boldsymbol{z}_{s_i} = \boldsymbol{x}_{s_i}$ and zero otherwise. Note that:

$$\|\boldsymbol{x} - \boldsymbol{z}\|_0 = \|\boldsymbol{x}\|_0 - 1$$

. Since there are exactly K distinct samples spanning the same non-zero coefficient, this reduction affects K more samples (equations). Specifically, if the base projection matrix $\bar{\boldsymbol{P}}$ is the incidence matrix of a BIBD, then identifying the signal values in one subset reduces the number of unknowns (un-identified coefficients) in all equations:

$$\boldsymbol{y}_i = \sum_{e=1}^{w} \boldsymbol{f}^{(i)} \boldsymbol{x}_{s_{i,e}}$$

by λ. Therefore the algorithm converges to the solution very fast. More specifically, consider a compressive sample $\boldsymbol{y}_t$ spanning two non-zeros including $\boldsymbol{x}_{s_{i,l}}$:

$$\boldsymbol{y}_t = \alpha \boldsymbol{x}_{s_{i,l}} + \beta \boldsymbol{x}_e, \; e \in [n]/s_{i,l}$$

where α and β are two distinct entries of $\boldsymbol{f}^{(t)}$. Thus:

$$\boldsymbol{y}_t^{(1)} = \boldsymbol{y}_t - \boldsymbol{P}_{i,s_i} \boldsymbol{x}_{s_i} = \beta \boldsymbol{x}_e$$

Consequently after this reduction, all compressive samples spanning two non-zeros such that one of these non-zeros is among the identified coefficients in the first iteration, turn into new samples spanning only one non-zero coefficient. Therefore in the next iteration, the phases of these compressive samples can be found among the available phases in the random vectors $\boldsymbol{f}^{(\cdot)}$. This event results in identifying the signal values in new subsets (and consequently new coordinates). Algorithm 7 shows the CRISP decoding process.

Algorithm 7: CRISP solver

```
**Input**: $\boldsymbol{y}$ and $\boldsymbol{P}$
**Output**: the sparse vector $\hat{\boldsymbol{x}}$ which $\boldsymbol{y} = \boldsymbol{P}\hat{\boldsymbol{x}}$
Set $I = 0$, $\boldsymbol{y}^{(0)} = \boldsymbol{y}$ and $\hat{\boldsymbol{x}} = \boldsymbol{0}$.
**repeat**
    Find all indices $\beta = \{i : \angle \boldsymbol{y}_i^{(I)} \in \angle \boldsymbol{f}^{(i)}\}$
    **for** *all* $i \in \beta$ **do**
        Suppose $\angle \boldsymbol{y}_i^{(I)} = \angle \boldsymbol{f}_l^{(i)}$ then: $\hat{\boldsymbol{x}}_e = \left\{ \begin{array}{cc} \|\boldsymbol{y}_i^{(I)}\| & e = s_{i,l} \\ 0 & e \in s_i/s_{i,l} \end{array} \right\}$
        $\boldsymbol{y}^{(I+1)} = \boldsymbol{y}^{(I)} - \boldsymbol{P}_{:,s_i} \hat{\boldsymbol{x}}_{s_i}$
    $I = I + 1$
**until** $\boldsymbol{y}^{(I)}$ *is a zero vector*;
```

4.4 Optimality

As stated before, the projection matrix ($\boldsymbol{P}$) inherits best properties of the base projection $\bar{\boldsymbol{P}}$. Therefore, by our design and with high probability, $\boldsymbol{P}$ corresponds to (randomly permuted) the incidence matrix of an expander graph. One can use the arguments of [37] that with high probability, having $m = O\left(k \log n/k\right)$ compressive samples, 1) in the first iteration of the solver, there exists at least $1 + [K/2]$ samples spanning only one specific non-zero coefficient (where K is the number of non-zero entries in each column of the projection matrix) and 2) the algorithm never halts and finally 3) it converges in $O(k)$ iterations. Therefore the algorithm is optimal in terms of the *order* of the number of samples for perfect recovery. However now instead of real valued samples, we are sensing complex samples which (albeit keeping the order of m fixed) doubles the number of sensed real samples (each complex sample is counted as two real measurements). On the other hand, as opposed to traditional combinatorial algorithms which require at least $1 + [K/2]$ samples that span only one common non-zero coefficient (for a voting like decoding algorithm), here we only need one sample (instead of $1 + [K/2]$) that spans one non-zero, which in turn lowers the required number of samples for perfect recovery. This significant reduction in the required samples is attributed to our utility of unique complex phases that make it feasible to detect and identify isolated non-zero coefficients. And finally, this generalization (from binary matrices to matrices with complex entries) reduces the complexity of the solver significantly from $O\left(n \log(n/k)\right)$ to $O\left(k \log(n/k) \log(n/m)\right)$.

Recall that the first iteration of the solver we look for the phases of compressive samples among a predetermined set of phases (available phases in each row of the projection matrix). Similar to

HCS, we can assume that for each row of $\boldsymbol{P}$, non-zero entries are sorted by their phases:

$$a, b \in [w], a > b \Leftrightarrow \angle \boldsymbol{f}_a^{(i)} = \phi_a^{(i)} > \boldsymbol{f}_b^{(i)} = \angle \phi_b^{(i)}$$

Then one might utilize a quick search algorithm (such as binary search) in the CRISP solver. Now for each sample, we need to search among $w = O(n/m)$ distinct phases and the complexity of such search is only $\log(w)$. Hence the first iteration of CRISP has a complexity of order $O\left(m \log(w)\right)$. For the next iteration of the solver, we need to subtract the effect of identified coefficients from the compressive samples. Since each non-zero coefficient appears in K (distinct) samples, hence the complexity of such update is $O(K)$ which is a constant. However the algorithm converges in $O(k)$ iterations, hence the overall complexity of CRISP is $O\left(m \log r + kK\right) = O\left(m \log(n/m)\right)$.

4.4.1 The Average Behavior Analysis

In this subsection, we present the proof for our claim that CRISP on average needs only $k < m \leq 2k$ complex (or equivalently $2k < m \leq 4k$ real) valued compressive samples for the perfect recovery of a k-sparse signal. Note that, this does not violates the optimality of the bound $m^\star = O(k \log n/k)$. In fact, it has been observed that BP typically recovers a k-sparse signal from $\approx 4k$ compressive samples. Furthermore, this analysis is only for the average case and in practice the solver usually deviates from the mean behavior and thus higher number of samples is usually required.

Lemma 4.4.1. *On average, CRISP requires $2k < m \leq 4k$ real-valued compressive samples for the perfect recovery of a k sparse signal of length n if $k \gg 1$.*

Proof. Let $C = m/k$ be the oversampling factor. Then the probability that a given sample $\boldsymbol{y}_j$

spans exactly i non-zeros is:

$$\Pr\left(\|\boldsymbol{x}_{s_j}\|_0 = i\right) = \frac{\binom{k}{i}\binom{m-1}{K-1}^{i}\binom{m-1}{K}^{k-i}}{\binom{m}{K}^{k}} \tag{4.3}$$

where m is the number of rows of $\boldsymbol{P}$. It is important to note that since the columns of CRISP projection matrix are permuted randomly, hence the probability that any given sample spans i non-zero coefficients, is constant for all samples:

$$\forall l, j \in [m] : \Pr\left(\|\boldsymbol{x}_{s_j}\|_0 = i\right) = \Pr\left(\|\boldsymbol{x}_{s_l}\|_0 = i\right) = \eta_i \tag{4.4}$$

Thus the expected number of samples, spanning only one non-zero is $m\eta_1$. To simplify our analysis, let us consider the asymptotic case where k (and hence m and n) tends to infinity. Then the expected number of samples spanning only one non-zero would be:

$$\underset{k\to\infty}{E\left(\#j : \|\boldsymbol{x}_{s_j}\|_0 = 1\right)} \approx kKe^{-\frac{K}{C}} \tag{4.5}$$

Note that for large values of k and fixed values of $m = Ck$, the expected number of samples spanning only one non-zero is a linear function of k. These samples identify the locations of isolated non-zero coefficients which consequently lead to the recovery of the associated subsets of coefficients.

Here we should emphasize that not all of these isolated coefficients provide new information. For instance, it is possible that there are several samples spanning only one specific non-zero coefficient $\boldsymbol{x}_i$. So it is important to exclude this redundant information from our analysis. Let $\mho$ be the probability of the event that one sample $\boldsymbol{y}_a$ spans only one non-zero coefficient $\boldsymbol{x}_i$, given that there exists another compressive sample $\boldsymbol{y}_b$ ($b \neq a$) spanning the same coefficient $\boldsymbol{x}_i$:

$$\mho \triangleq \Pr\left(s_a \cap Supp(\boldsymbol{x}) = i | \exists b \in [m] : s_b \cap Supp(\boldsymbol{x}) = i\right) \tag{4.6}$$

Then we have:

$$\mho = \frac{k(m-1)\binom{m-2}{K-2}\binom{m-2}{K}^{k-1}}{\binom{m}{K}^{k}} \tag{4.7}$$

That is choose a non-zero coefficient $\boldsymbol{x}_i$ among k non-zeros and choose another sample $\boldsymbol{y}_a$ (except $\boldsymbol{y}_b$) and let $\boldsymbol{y}_a$ be only function of $\boldsymbol{x}_i$. Now randomly select $K-2$ other samples which would be functions of $\boldsymbol{x}_i$ and distribute the remaining $k-1$ non-zeros (counting their K repetitions) among remaining $m-2$ samples. Fix $m = Ck$, for large values of k (4.7) can be approximated to:

$$\underset{k\to\infty}{\mho} \approx \frac{k(K^2-K)}{m} e^{-\frac{2K}{C}} \tag{4.8}$$

which is a linear function of sparsity measure k. Thus the expected number of samples which do not result to discovery of new coefficients (μ) is:

$$\mu = \frac{m}{K}\mho \approx \frac{m}{K}\frac{kK(K-1)}{m} e^{-\frac{2K}{C}} = k(K-1)e^{-\frac{2K}{C}} \tag{4.9}$$

Recall that (on average) $kKe^{-\frac{K}{C}}$ samples span only one non-zero coefficient. Hence, the expected number of compressive samples spanning distinct non-zeros in the first iteration of the solver is:

$$\mu' = kKe^{-\frac{K}{C}} - \mu = k\left(Ke^{-\frac{K}{C}} - (K-1)e^{-\frac{2K}{C}}\right) \tag{4.10}$$

In words, CRISP solver (on average) recovers μ' non-zero coefficients in the first iteration. Since the columns of the CRISP projection matrix are permuted independently and uniformly, we can

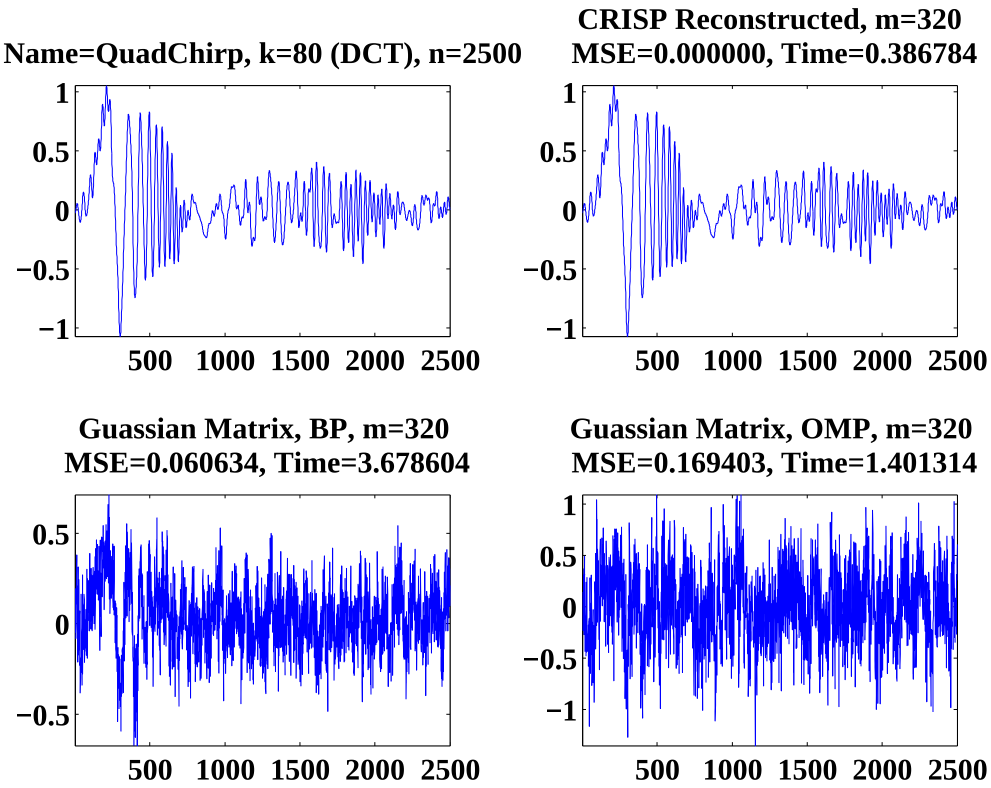


Figure 4.1: Comparing Compressive Demosaicing with some well-known demosaicing algorithms.

say that in the next iteration, there are $k^{(1)}$ non-zeros left to recover from m samples, where:

$$k^{(1)} = k - \mu' = kq, \; q = \left(1 - Ke^{-\frac{K}{C}} + (K-1)e^{-\frac{2K}{C}}\right) \tag{4.11}$$

Note that for fixed values of $C = m/k$, q is independent of k. It is straightforward to see that, (on average) in the i-th iteration of the algorithm, there are: $k^{(i)} = kq^i$ non-zero coefficients which have not been recovered. Also note that even for $1 < C \leq 2$, we have $q < 1$. In other words, on average, $k < m \leq 2k$ complex valued samples (and hence $2k < m \leq 4k$ real valued samples) is sufficient for the perfect recovery. ■

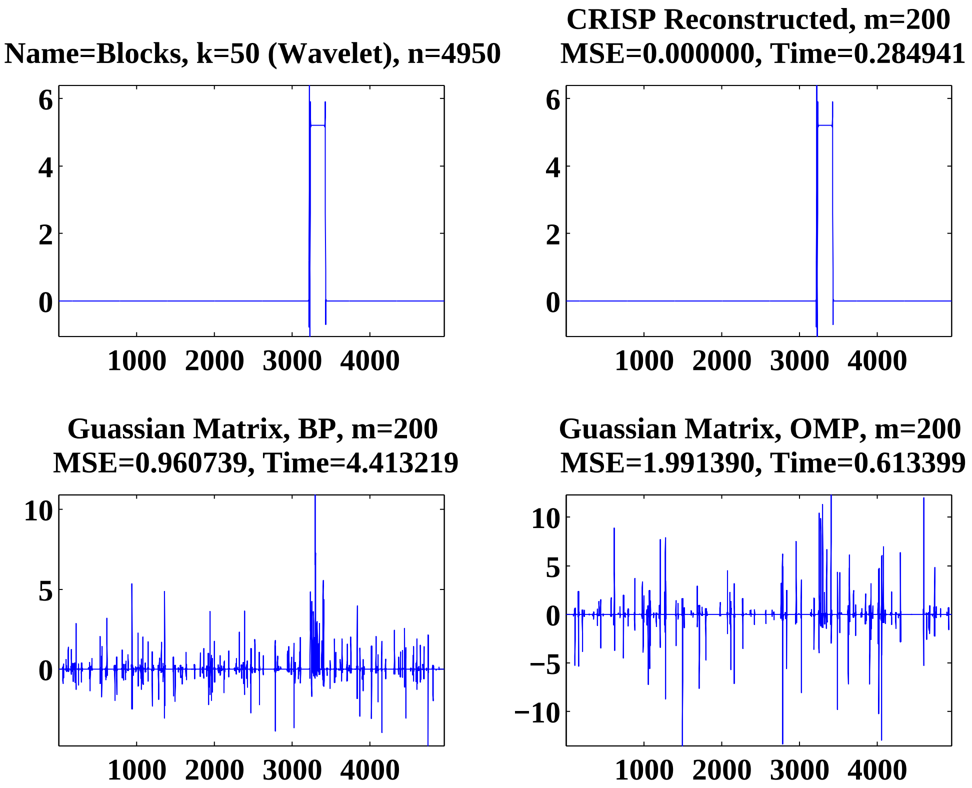


Figure 4.2: Comparing Compressive Demosaicing with some well-known demosaicing algorithms.

4.5 Simulation Results

We tested the proposed CRISP on a large number of standard signals from SparseLab [45]. Specifically we compared the performance of CRISP (in terms of quality of the recovered signal as a function of the number of compressive samples and the required time for recovery) in comparison with the popular Gaussian random projection matrix and dominant Basis Pursuit (BP) and OMP solvers. In all plots m, n and k denote the number of compressive samples, the signal length and the sparsity of the signal in the sparsifying domain (Wavelet/DCT). To provide fairness, we have counted each complex sample as two samples. We performed our simulations under various configurations of signal length and sparsity ratio which we present some of them in Fig. 4.1- Fig. 4.3.. As clearly demonstrated CRISP has significantly lower complexity compare to BP and OMP.

Here we should highlight an important note: due to the strong structures and rich properties of

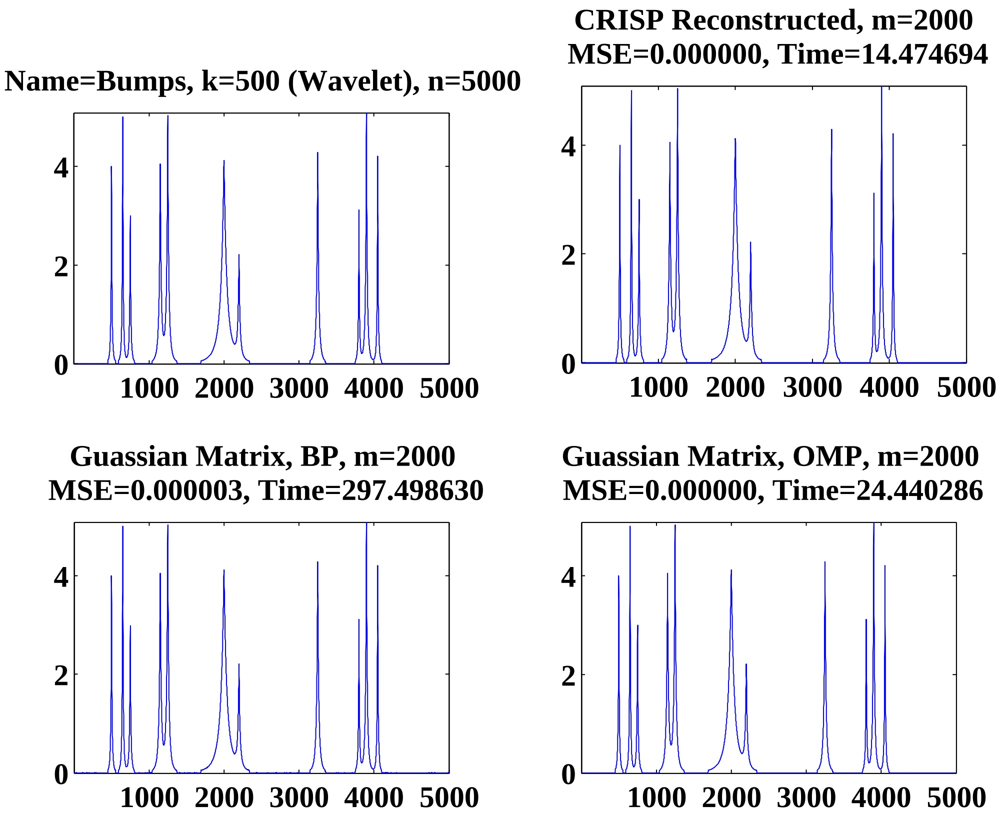


Figure 4.3: Comparing Compressive Demosaicing with some well-known demosaicing algorithms.

BIBDs, we expect that a sensing matrix generated based on an incidence matrix of a BIBD boosts CRISP performances in terms of the solver complexity and also the sample requirements for the perfect recovery. For instance, in Fig. 4.2, we have used a Hamming matrix of weight two (all m tuples with $K = 2$ non-entries in each column) as the base projection matrix $\bar{\boldsymbol{P}}$. Hence $\bar{\boldsymbol{P}}$ is an incidence matrix of a $(m, m(m-1)/2, 2, 1)$-BIBD. In that simulation, we observed that if the incidence of a $(200, 4950, 2, 1)$-BIBD was used as the base projection matrix, then $m = 4k$ (real valued) samples was enough for the perfect recovery. On the other hand, if we have used Algorithm 5 to generate the base projection, then $m \approx 6k$ samples was required for the perfect recovery and also the decoding time would increase as well. However we should recall that: although there are numerous methods for generating a BIBD with some given parameters [49], it is possible that there are no BIBD designs that exist in some configurations of (m, n, K, λ).

Chapter 5

Modified SSMP

In this chapter, we are interested in finding a sensing operator $\boldsymbol{P}$ which can be utilized in all three types of CS approaches (combinatorial, convex relaxation, and greedy); and at the same time, $\boldsymbol{P}$ can provide recovery guarantees in ℓ_2 norm sense. We show that by some modification in the adjacency matrices of expander graphs, one can construct a class of sensing matrices which exhibits some "*weak*" form of RIP. Let us clarify what we mean by weak in here. As stated before, the formal definition of RIP demands that $||\boldsymbol{P}x||_2$ must be close to $||x||_2$ (with high probability) for *all* possible k-sparse signals. Although RIP is known to be a sufficient condition to guarantee the success of geometrical approaches in solving CS, it has been reported in numerous papers (e.g. [64] [65] [66]) that the empirical results of linear programming solvers shows that there might be some less constraining properties than RIP, governing the success of linear programming in CS context. This observation raised to investigation for alternative sufficient conditions which consequently led to introduction of different types of RIP, for instance statistical RIP [64], weak RIP [67] and generalized RIP [69] to name a few. This chapter is along with those efforts. More specifically, we focus on the probability of recovering any arbitrary but fixed signal and not the probability of failure of the decoder for all possible k-sparse signals which clearly is a much stronger one. Such notion of fixing the signal support has been introduced and used before, for instance in [67]. By using basic arguments, we show that under this measure (the probability of failure for a fixed arbitrary signal), a simpler and weaker version of RIP which we refer to as w-RIP (that is very similar to the concept Weak-RIP in [67]) is sufficient to recover any arbitrary fixed signal under convex relaxation methods and at least one optimal greedy algorithm called Cosamp [28] with high probability. The proposed w-RIP possibly widens the admissibility of many sensing matrices which do not possess the original RIP, to be considered as "good" sensing mechanism at least in the sense of *any arbitrary but fixed signal*. We show that if a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times n}$ is constructed by replacing the non-zero entries of the sparse binary adjacency matrix of an expander graph with a Gaussian random vector then P would benefit from w-RIP and hence could be utilized in geometrical and greedy frameworks. Also, we show that there exists at least a combinatorial algorithm (based on Sequential Sparse Matching Pursuit [68]) for the proposed $\boldsymbol{P}$ with the $\ell_2 < C\ell_2$ guarantee of reconstruction. To the best of our knowledge, no other class of sensing projection matrices has been introduced in prior works which can provide $\ell_2 < C\ell_2$ guarantee of reconstruction under all three approaches to CS.

This chapter is organized as follows: in Section 5.1 and 5.2, the proposed SERP and our definition of w-RIP are introduced. We prove that if one targets for recovering an arbitrary but fixed signal, then w-RIP is a sufficient condition of success for at least one greedy and all geometrical CS frameworks. In section 5.3, we prove that SERP matrices adheres to such w-RIP and thus can be safely utilized in the aforementioned CS frameworks. In Section 5.4, we show that for the proposed class of SERP matrices, there exists a fast combinatorial algorithm which can robustly recover a signal from its noisy samples. Finally, simulation results are presented in section 5.5.

5.1 SERP projection matrix

Consider matrix $\boldsymbol{P} = G(m, n, r, d, D)$ where the function G is defined in Algorithm 8 D is a probability distribution and m, n, r and d are some natural numbers with the condition that m is divisible by r. In words, Algorithm 8 generates the matrix $\boldsymbol{P}$ by following these operations: (i) for each column $i \in [n]$, the algorithm selects uniformly at random d indices among $[m/r] = \{1, 2, \ldots, m/r\}$. Let us denote those indices by h. (ii) Then for each entry of h, for instance the l-th entry h_l, the algorithm generates another r indices $\{(h_l - 1)r + 1, (h_l - 1)r + 2, \ldots, h_l r\}$. The collection of these rd indices forms the row-indices where the i-th column will be non-zero (i.e. Ω_i). (iii) Finally, each non-zero entry of the i-th column of $\boldsymbol{P}$ would be a random variable with distribution D. The output of Algorithm 8 could be also viewed as a matrix which each of its column is non-zero in exactly d random indices and each non-zero entry is a random vector $(t_1, t_2, \ldots, t_r)$ with $t_i \sim D$ for $i \in [r]$. Note that, for all i and j where $\lfloor i/r \rfloor = \lfloor j/r \rfloor$ we have $\omega_i = \omega_j$. It is straightforward to verify that for sufficiently large n if (with high probability) the output of $G(m/r, n, 1, d, D)$ corresponds to a high quality expander graph, then $G(m, n, r, d, D)$ would also (with high probability) correspond to a high quality expander. Also it should be clear that permuting columns and rows of the generated projection matrix has no effect on our arguments nor the quality of the matrix.

A wide range of popular projection matrices (in CS context) could be generated by Algorithm 8. For instance if $m = O\left(k \log n/k\right)$ and $D = N(0, 1/m)$, then $G(m, n, 1, m, D)$ outputs a random dense Gaussian projection matrix which has RIP with high-probability [16]. On the other hand, for $d = \theta\left(\log n\right)$, $r = 1$ and a distribution with delta-Dirac at one and zero otherwise for D (i.e. $t \sim D$ is a constant number with value of one), the output of the Algorithm 8 (with high probability [38, 37, 68]) will correspond to a bipartite expander graph and hence can be utilized in a combinatorial solver for CS.

In this chapter, we are interested in the output of the Algorithm 8 when $r = O\left(1\right)$ is a small natural number, $d = \theta\left(\log n\right)$, $m = O(k \log n/k)$ and $D = N(0, 1/rd)$. We call an output instance of $G(m, n, r, d, D)$ with those values as “Sparse Expander-like Real-valued Projection” or SERP. With some realistic approximations, we show that this class of matrices adheres to a property

Algorithm 8: The function $G(m, n, r, d, D)$.

```
input  : $m$, $n$, $r$, $d$ and $D$
output: $\boldsymbol{P} \in \mathbb{R}^{m \times n}$
for $i = 1$ to $n$ do do
    $T = \mathbf{0}_m$ ;
    Let $h$ be a random subset of $[m/r]$ of size $d$ ($Card\{h\} = d$) ;
    $\Omega_i = \cup_{l=1}^{m/r}\{(h_l - 1)r + 1, (h_l - 1)r + 2, \ldots, h_l r\}$ ;
    $\forall j \in \Omega_i : \boldsymbol{P}_{j,i} \sim D$ ;
end
```

which we refer to as w-RIP (to be defined in follow) for an arbitrary signal x with a fixed support set $T = Support\{x\}$. Then, we show that w-RIP is in fact sufficient to guarantee the recovery of that fixed signal under convex relaxation and Cosamp (an optimal greedy CS decoder) and there is no need for the stronger RIP condition. Also by using the same property, we prove that modifying only one line of the SSMP decoder [68] suffices to come up with a combinatorial algorithm with a $\ell_2 < C\ell_2$ guarantee of recovery for that particular signal under SERP sensing matrices. To the best of our knowledge, this is the first class of matrices which can provide $\ell_2 < C\ell_2$ recovery under all three approaches to the problem of compressed sensing at least in a "for any arbitrary but fixed signal" setting. Throughout this paper, we assume that parameters m, d and r are chosen properly such that, if non-zero entries of $\boldsymbol{P} = G(m, n, r, d, D)$ are replaced by one, then it would correspond to a $((c + 1)k, d, q)$ expander for small constants $c \geq 1$ and $0 < q < 1$.

5.2 w-RIP

Since Restricted Isometry Property (RIP) is an essential tool in the analysis of different algorithms in Compressed Sensing, we review that concept in here. A matrix $\boldsymbol{P} \in \mathbb{R}^{m \times n}$ has RIP of order $k \in \mathbb{N}$ [3], if for every k-sparse signal $x \in \mathbb{R}^n$, $||x||_0 = k$, there exists a Restricted Isometry

Constant (RIC) constant δ'_k such that:

$$\left(1+\delta'_k\right)||x||_2^2 \leq ||\boldsymbol{P}x||_2^2 \leq \left(1-\delta'_k\right)||x||_2^2$$

No deterministic construction is available so far for a matrix to achieve RIP of order k with the minimal number of rows (samples) of $m = O\left(k \log n/k\right)$. However, it has been proved that some random matrices satisfy RIP with an optimal number of rows with high probability. For instance, it can be shown [16] that Gaussian random matrices have that property with a probability of at least $1 - cn^{-\gamma}$ for some positive constants c and γ. To establish a clear connection between RIP and our proposed w-RIP let us define an auxiliary RIP which we refer to as probabilistic-RIP and abbreviate by p-RIP in this paper.

Definition 1. *We say, that a matrix $\boldsymbol{P} \in \mathbb{R}^{m\times n}$ has p-RIP of order $(k,p) \in \mathbb{N}\times[0,1]$, if P has RIP of order k with a probability of at least p.*

Hence Gaussian random matrices with $m = O\left(k \log n/k\right)$ rows have p-RIP of order (k,p) with $p = 1 - cn^{-\gamma}$ for some constants γ and c. Now let us present our definition of w-RIP in here.

Definition 2. *Definition 2. We say, that a matrix $\boldsymbol{P} \in \mathbb{R}^{m\times n}$ has w-RIP of order $(k,p) \in \mathbb{N}\times[0,1]$, if for any arbitrary but fixed k-sparse signal $x \in \mathbb{R}^n$, $||x||_0 = k$, there exists a w-RIP constant $0 \leq \delta_{k,p} \leq 1$ such that:*

$$\left(1+\delta_{k,p}\right)||x||_2^2 \leq ||\boldsymbol{P}x||_2^2 \leq \left(1-\delta_{k,p}\right)||x||_2^2$$

with a probability of at least p.

In some parts of this chapter when p and k are known from the context, we drop subscripts and simply denote $\delta_{k,p}$ by δ. Note that, this definition is slightly different from the definition of weak-RIP in [67] and the one in [69]. The connection between RIP and p-RIP is obvious from their definitions. The connection between w-RIP and p-RIP can be justified as follow. Assume the set of $T = \{T_1, T_2, \ldots\}$ contains all subsets of $[n] = \{1, 2, \ldots, n\}$ with the cardinality of k, i.e:

$$\forall i \in \{1, 2, \ldots, Card\{T\}\} : \; T_i \subseteq [n], \; Card\{T_i\} = k$$

Furthermore, suppose that $\boldsymbol{P}$ has w-RIP of order (k, p). Then it means that for each T_i, all singular values of $\boldsymbol{P}_{:,T_i}$ are in the range of $[\sqrt{1-\delta_k}, \sqrt{1+\delta_k}]$ with a probability of p. If columns of $\boldsymbol{P}$ are independently distributed (which is usually the case) then $\boldsymbol{P}$ would have RIP of order k with a probability of $p^{Card\{T\}}$, i.e. p-RIP of order $\left(k, p^{Card\{T\}}\right)$.

The first papers on Compressed Sensing (e.g. [2, 8, 4, 5]) were focusing on the failure probability of decoder for all possible k sparse signals. For instance, to target a probability of success q in case of a convex relaxation decoder, one needs a p-RIP of order $(2k, q)$. Consequently, to show that one matrix has this property, that matrix must have w-RIP of order $(2k, p)$ where $p \geq \sqrt[Card\{T\}]{q}$ and T contains all subsets of $[n]$ with the cardinality of k. Note that cardinality of the superset T is $Card\{T\} = \binom{n}{2k}$. Even for small values of n, $Card\{T\}$ is enormously large, demanding $\sqrt[Card\{T\}]{q}$ to be extremely close to one. This theoretically deprives many sensing matrices from having RIP or p-RIP for the optimal number of samples $m = O\left(k \log n/k\right)$ even though they exhibit good empirical results in practice. This has been one of main motives that some recent efforts (e.g. [64, 67]) has focused on deriving "*the probability of failure of a CS framework for any arbitrary but fixed signal*" instead of the "*the probability of failure of a CS framework for all possible signals*".

One expects that this relaxation should have some implications on RIP as well and fortunately that is the case. By looking more closely at proofs in [bp] and [28], one can easily notice that under both these frameworks, to recover an arbitrary signal x with a fixed support from its compressive samples, there is no need for the full RIP or p-RIP. Instead, to prove that an arbitrary signal with a fixed support set could be recovered from its compressive samples with a target probability of q under those decoders, it only suffices to have w-RIP of order $\left(k, p = \sqrt[z]{q}\right)$ where the constant z is solver-dependent and by far smaller than $\binom{n}{2k}$. For instance as we show in proof of Lemma 5.2.1, the constant z in case of a linear program based decoder is only $O\left(n/k\right)$. Also later in this paper, we show that the proposed class of SERP matrices can easily achieve w-RIP of a proper order with some realistic approximations.

5.2.1 Implications of w-RIP

The following two lemmas prove that, if a matrix has w-RIP of proper orders, then it will be a "*good*" sensing mechanism under Cosamp (a greedy algorithm) and also any convex relaxation decoder when one targets for recovering any arbitrary signal with a fixed support.

Lemma 5.2.1. *Consider any arbitrary signal $x \in \mathbb{R}^n$ with the fixed support set T of cardinality k. Assume that, $\mathbf{P} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ has w-RIP of order $(2k, p)$ with the constant $\delta_{2k,p} < \sqrt{2} - 1$ for $p = 1 - 1/n^{\beta+1}$ where $\beta \geq 1$. If $\Delta\left(\boldsymbol{P}, y\right)$ belongs to the class of convex relaxation methods, then $\Delta\left(\boldsymbol{P}, y\right)$ provides a $\ell_2 < C\ell_1$ guarantee of recovery for x from $y = \boldsymbol{P}x$ with a probability of at least $p^{2n} \geq 1 - 2/n^{\beta}$.*

Proof: Let $x^* = \Delta\left(\boldsymbol{P}, y\right)$ where Δ is any convex relaxation based decoder. In other words, x^* is the solution to optimization problem (1.8). As in [3] define $h = x^* - x$ and partition $[n]$ into sets of cardinality k, denoted by $T_0, T_1, T_2, \ldots$ such that T_0 corresponds to k largest entries of x, T_1 corresponds to k largest entries of $h_{T_0^c}$, T_1 corresponds to the next k largest entries of $h_{T_0^c}$ and so on. Fixing the support of x, the arguments of [3] requires that RIP inequalities to be active on sets

$$T_0 \cup T_1,\ T_0 \cup T_2, T_0 \cup T_3, \ldots, T_1 \cup T_2,\ T_1 \cup T_3, T_1 \cup T_4, \ldots,$$

Since k could be as low as one, there are at most $2n$ of these sets. Now assume that the matrix P has w-RIP of order $(2k, p)$. Then $\boldsymbol{P}$ has the required properties in [3] with a probability of p^{2n}. Recalling that

$$\forall n \geq 1 : \left(1 - \frac{1}{n^{\beta+1}}\right)^{2n} \geq 1 - \frac{2}{n^{\beta}}$$

it can be concluded that if one targets for recovery of a fixed signal with a probability of $q = 1 - 2/n^{\beta}$ for some constant $\beta \geq 1$, then if suffices to have $p > 1 - 1/n^{\beta+1}$. ■

Lemma 5.2.2. *Consider any arbitrary signal $x \in \mathbb{R}^n$ with the fixed support set T of cardinality k. Assume that, $\mathbf{P} \in \mathbb{R}^{\mathbf{m} \times \mathbf{n}}$ has w-RIP of order $(2k, p)$ with the constant $\delta_{4k,p} < 0.1$ for $p = 1 - 1/n^{\beta+1}$ where $\beta \geq 1$. If $\Delta(\boldsymbol{P}, y)$ is Cosamp algorithm, then $\Delta(\boldsymbol{P}, y)$ provides a $\ell_2 < C\ell_1$ guarantee of recovery for x from $y = \boldsymbol{P}x$ with a probability of at least $p^{2n} \geq 1 - 7/n^{\beta}$.*

Proof: For a k-sparse signal x, Cosamp requires at most $6(k+1) \leq 7k$ iterations (for $k \geq 6$). Assume that in the start of the i-th iteration, the algorithm estimates x by $x^{\{i\}}$. Define:

$$R^{\{i\}} = x - x^{\{i\}},\ d^{\{i}\} = y - \boldsymbol{P}x^{\{i\}}$$

In words, $R^{\{i\}}$ is the residue of estimation and $d^{\{i}\}$ is the discrepancy of estimation in that iteration. Finally let $\Omega^{\{i\}}$ be indices corresponding to $2k$ largest values of $P^T d^{\{i\}}$. Then on the i-th iteration of Cosamp, one needs RIP inequalities only on column indices

$$\forall i \in [6(k+1)] : T_1^{\{i\}} = Support\{\Omega^{\{i\}}\} \backslash Support\{R^{\{i\}}\}$$
$$\forall i \in [6(k+1)] : T_2^{\{i\}} = Support\{\Omega^{\{i\}}\} \cup Support\{x\} \cup Support\{x^{\{i\}}\}$$

Note that for all i, $Card\{T_1^{\{i\}}\} \leq 2k$ and $Card\{T_2^{\{i\}}\} \leq 4k$. Based on the proof of [28], Cosamp recovers a fixed sparse signal x when the restricted isometery constant on sets $T_1^{\{1\}}, T_1^{\{2\}}, \ldots, T_1^{\{6(k+1)\}}$ and $T_2^{\{1\}}, T_2^{\{2\}}, \ldots, T_2^{\{6(k+1)\}}$ are all smaller than 0.1. Therefore, if one wants to recover any arbitrary but fixed signal x with a target probability of q then the matrix $\boldsymbol{P}$ has to have a w-RIP of order $(4k, p)$ with the constant $\delta_{k,p} \leq 0.1$ for $p \geq \sqrt[6(k+1)]{q}$. For instance, if $p \geq 1 - 1/n^{\beta+1}$ then $q \geq 1 - 7/n^{\beta}$. ■

5.3 w-RIP for SERP matrices

The following two lemmas prove that SERP matrices have w-RIP and hence could be utilized in Cosamp and convex relaxation methods in a “for any arbitrary but fixed signal” scenario.

Lemma 5.3.1. *Let* $\mathbf{P} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ *be an output instance of* $G(m, n, r, d, D)$*. If* $d = \theta(\log n)$*,* $m = O(k \log n/k)$*,* $D = N(0, 1/rd)$ *and* $r = O(1)$ *is a constant, then for any arbitrary vector* $x \in \mathbb{R}^n$ *with a fixed support, there exist constants* δ *and* $\alpha = O(r)$ *such that*

$$Pr\left(||\boldsymbol{P}x||_2^2 \geq (1+\delta)\,||x||_2^2\right) \leq \frac{1}{n^{\alpha}} \tag{5.1}$$

Proof: Without loss of generality, assume that x has a unit norm ($||x||_2^2 = 1$). To bound $||y||_2^2 = ||\boldsymbol{P}x||_2^2$ (similar to [16]) one can resort to moment generating function of the random variable y:

$$\Pr\left(||y||_2^2 \geq (1+\delta)\right) = \Pr\left(\exp\left(t||y||_2^2\right) \geq \exp\left(t(1+\delta)\right)\right) \tag{5.2}$$

$$\leq \frac{E\left(\exp\left(t||y||_2^2\right)\right)}{\exp\left(t(1+\delta)\right)} \tag{5.3}$$

for a positive value of t, where the last inequality is due to Markov's. Clearly $E\left(\exp\left(t||y||_2^2\right)\right)$ is the moment generating function of the random variable $||y||_2^2$. Note that (see [16]):

$$y_i \sim \frac{||x_{\omega_i}||_2 N(0,1)}{\sqrt{rd}}$$

where $N(0,1)$ is standard normal distribution. Consequently

$$y_i^2 \sim \frac{||x_{\omega_i}||_2^2 \chi^2(1)}{rd}$$

where $\chi^2(1)$ is a chi-squared distribution with one degree of freedom. Define

$$\forall i \in [m] : z_i = \frac{||x_{\omega_i}||_2^2}{rd} = \frac{\sum_{j \in \omega_i} x_j^2}{rd}$$

Then the random variable $||y||_2^2 = \sum_{i=1}^m z_i \chi^2(1)$ is a weighted sum of chi-squares. It is known and easy to verify that manipulating the moment generating function of a random variable with the distribution of weighted sum of chi-squares does not lead to neat or usable formulas. Hence, a number of estimations are suggested to approximate that random variable with another one which can be manipulated more easily (e.g. [70, 71]). Here in this paper, we use the approximation in [70]. A commonly practiced approximation of a weighted chi-squared random variable is a Gamma distribution with the same mean and variance of the original random variable. In [70], the authors proved that, this approximation leads to acceptable results and more importantly, the gamma approximation tends to be an over-estimate of the true distribution of $||y||_2^2$ when variance of $\{z_i\}$ is not small[1]. Since values of $\{z_i\}$ in SERP are linearly proportional to $||x_{\omega_i}||_2^2$ (i.e. norms of random subsets of the signal) and the magnitudes of many signals of the interest (e.g. DCT coefficients of images) almost follow a power law, hence at least for these classes of signals, it is very unlikely that $||x_{\omega_i}||_2^2$ to have small variances. Consequently, this approximation seems to be reasonable for those practical signals of interest. Define

$$a = \frac{\sum_{i=1}^{m} z_i}{2\sum_{i=1}^{m} z_i^2},\ \lambda = \frac{(\sum_{i=1}^{m} z_i)^2}{2\sum_{i=1}^{m} z_i^2} \tag{5.4}$$

Then based on the approximation of [70], $||y||_2^2$ can be approximated by a Gamma distribution with the shape parameter λ and scale parameter $1/a$. Note that

$$\sum_{i=1}^{m} z_i = \frac{\sum_{i=1}^{m} c_i x_i^2}{rd},\ c_i = Card\{j : j \in \omega_i\} \tag{5.5}$$

where c_i is the number of times that $i \in [n]$ occurs in ω_i. Recall that, all columns of $\boldsymbol{P}$ are non-zero in exactly rd row indices. Hence, for all $i \in [n]$ we have $c_i = rd$. Since it is assumed that x is unit norm, we get

$$\sum_{i=1}^{m} z_i = \left(\sum_{i=1}^{m} z_i\right)^2 = 1 \Rightarrow a = \lambda$$

Now let us find upper and lower bounds for $\sum_{i=1}^{m} z_i^2$.

It can be proved by contradiction that given the properties of $\{z_i\}$, the vector $z = [z_1\ z_2\ z_3\ \ldots\ z_m]$ has to be maximally sparse when the objective function $\sum_{i=1}^{m} z_i^2$ attains its maximum. More specifically, assume that n is not extremely small such that $rd > 1$. Furthermore, assume that a

[1] Here we are deriving the upper bound for equation (5.1). Hence, the derived probability from the approximation in here is higher than the actual one.

given sequence $\{z_i\}$ leads to maximum value for $\sum_{i=1}^{m} z_i^2$, however the vector $z = [z_1\ z_2\ z_3\ \ldots\ z_m]$ is not maximally sparse. Then, one can find two indices $a, b \in Support\{z\}$ such that $z_a \neq 0$ and $z_a \neq 0$ (since $rd > 1$ and $||z||_0 \geq rd$) and form a new sequence of $\{w_1\ w_2\ \ldots\ w_m\}$ where

$$w_c = \begin{cases} z_c & c \neq a,\ c \neq b \\ 0 & c = a \\ z_a + z_b & c = b. \end{cases} \tag{5.6}$$

Define $w = [w_1\ w_2\ \ldots\ w_m]$. It is easy to verify that

$$\sum_{i=1}^{m} z_i = \sum_{i=1}^{m} w_i,\ ||w||_0 = ||z||_0 - 1,\ \sum_{i=1}^{m} z_i^2 < \sum_{i=1}^{m} w_i^2$$

which contradicts our assumption that this sequence of $\{z_i\}$ leads to maximum value for $\sum_{i=1}^{m} z_i^2$. Performing the re-distribution of signal energies successively, one can conclude that the vector $z = [z_1\ z_2\ z_3\ \ldots\ z_m]$ has to be maximally sparse when $\sum_{i=1}^{m} z_i^2$ attains its upper bound. However, the sparsity of the vector z can never be less than rd, due to two facts: (a) the matrix $\boldsymbol{P}$ corresponds to an expander and (b) each column of $\boldsymbol{P}$ is non-zero at exactly rd indices. This maximally sparse case happens when the signal is non-zero in exactly one location. Since, we have assumed that the signal is unit-norm, then it means that the signal is one at one index and zero in the rest of indices. It is straightforward to verify that in this case, $\sum_{i=1}^{m} z_i^2 = 1/rd$. This concludes the upper bound for $\sum_{i=1}^{m} z_i^2$.

It is known and easy to verify that the when the constraint $\sum_{i=1}^{m} z_i = 1$ is active, the minimum value of $\sum_{i=1}^{m} z_i^2$ occurs when all z_i are equals to $1/m$. In that case, $\sum_{i=1}^{m} z_i^2$ equals to $1/m$ as well.

Putting together upper and lower bounds, we have $rd/2 \leq a = \lambda = m/2$. Based on arguments above, the following holds

$$\Pr\left(||y||_2^2 \geq (1+\delta)\right) \leq \frac{E\left(\exp\left(t||y||_2^2\right)\right)}{\exp\left(t(1+\delta)\right)} = \tag{5.7}$$

$$\frac{\left(1-\frac{t}{a}\right)^{-a}}{e^{t(t(1+\delta))}} = f(t) \tag{5.8}$$

for $0 < t < a$. By setting $\partial f(t)/\partial t = 0$, we get $t = \frac{a\delta}{1+\delta} < a$ which is in a valid range for the corresponding moment generating function. Substituting back this value into (5.8) leads to

$$\Pr\left(||y||_2^2 \geq (1+\delta)\right) \leq \left(\frac{1+\delta}{e^{\delta}}\right)^a \tag{5.9}$$

Let

$$\tau = \frac{\delta^2}{2\left(1+\delta+\frac{\delta^2}{2}\right)} \tag{5.10}$$

Then by writing Taylor series expansion of $(1+\delta)/\exp\delta$ around $\delta = 0$ upto δ^2 and recalling that $1 - s \leq \exp(-s)$, it can be concluded that

$$\Pr\left(||y||_2^2 \geq (1+\delta)\right) \leq \exp(a)^{-\tau} \tag{5.11}$$

Now recall that $rd/2 \leq a$ and $d = c_1 \log n$ for a constant c_1. Thus

$$\Pr\left(||y||_2^2 \geq (1+\delta)\right) \leq \frac{1}{n^{rc_1\tau/2}} \tag{5.12}$$

In summary, there exists a constant $\alpha = rc_1\tau/2$ such that $Pr\left(||y||_2^2 \geq (1+\delta)\right) \leq 1/n^{\alpha}$. Clearly parameter r (in the projection matrix) can be tuned such that $\alpha \geq 2$. ■

Lemma 5.3.2. *Let* $\mathbf{P} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ *be an output instance of* $G(m, n, r, d, D)$. *If* $d = \theta(\log n)$, $m =$

$O(k \log n/k)$, $D = N(0, 1/rd)$ and $r = O(1)$ is a constant, then for any arbitrary vector $x \in \mathbb{R}^n$ with a fixed support, there exist constants δ and $\alpha = O(r)$ such that

$$\Pr\left(||\boldsymbol{P}x||_2^2 \geq (1+\delta)\,||x||_2^2\right) \leq \frac{1}{n^{\alpha}} \tag{5.13}$$

Proof: Assume that x has a unit norm. Similar to proofs of Lemma 5.3.1, one can conclude:

$$\begin{aligned} \Pr\left(||y||_2^2 \leq (1-\delta)\right) &= Pr\left(\exp\left(-t||y||_2^2\right) \geq \exp\left(-t(1-\delta)\right)\right) && (5.14)\\ &\leq \frac{E\left(\exp\left(-t||y||_2^2\right)\right)}{\exp\left(-t(1-\delta)\right)} && (5.15) \end{aligned}$$

for a positive value of t. We again use the approximation of [70] to estimate $E\left(\exp\left(-t||y||_2^2\right)\right)$. More specifically by using arguments provided in the proof of Lemma 5.3.1, we have

$$\frac{E\left(\exp\left(-t||y||_2^2\right)\right)}{e^{-t(1-\delta)}} \approx \frac{e^{t(1-\delta)}}{\left(1+\frac{t}{a}\right)} \tag{5.16}$$

where $rd/2 \leq a \leq m/2$. For finding the tightest bound for (5.15), one should set the derivative of (5.16) with respect to t to zero and solve for t. That value is $t = a\delta/(1-\delta)$. It is straightforward to verify that if $0 < \delta < 1/2$ then this computed value for t is in the valid range of the corresponding moment generating function. Plugging back this value to (5.15) yields

$$Pr\left(||y||_2^2 \leq (1-\delta)\right) \leq \left((1-\delta)e^{\delta}\right)^a \tag{5.17}$$

Note that Taylor series expansion of $(1-\delta)e^{\delta}$ around $\delta = 0$ up-to degree 4 is

$$(1-\delta)e^{\delta} = 1 - \frac{\delta^2}{2} - \frac{\delta^3}{3} - \frac{\delta^4}{8} + O\left(\delta^5\right) \tag{5.18}$$

Hence

$$\left((1-\delta)e^{\delta}\right)^{a} \leq e^{-\frac{a\delta^2}{2}} \tag{5.19}$$

Considering that $rd/2 \leq a \leq m/2$ and $d = c_1 \log n$ for a constant c_1, the following holds

$$Pr\left(||y||_2^2 \leq (1-\delta)\right) \leq \frac{1}{n^{\alpha}} \tag{5.20}$$

for $\alpha = rc_1\delta^2/4$. Again r can be tuned such that $\alpha \geq 2$. ■

Theorem 5.3.3. *Let* $\mathbf{P} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ *be an output instance of* $G(m,n,r,d,D)$*. If* $d = \theta(\log n)$*,* $m = O(k \log n/k)$*,* $D = N(0, 1/rd)$ *and* $r = O(1)$ *is a constant, then* $\boldsymbol{P}$ *would have w-RIP of order* (k,p) *with the constant* δ *for* $p \geq 1 - \frac{2}{n^{\alpha}}$ *where* $\alpha = O(r)$*.*

Proof: Lemmas 5.3.1, Lemma 5.3.2 and the Definition 2 directly imply that $\boldsymbol{P}$ has w-RIP of order (k,p) for $p \geq 1 - \frac{2}{n^{\alpha}}$ where

$$\alpha = \min\left(\frac{rc_1\delta^2}{4\left(1+\delta+\frac{\delta^2}{2}\right)}, \frac{rc_1\delta^2}{4}\right) = \frac{rc_1\delta^2}{4\left(1+\delta+\frac{\delta^2}{2}\right)} = O(r)$$

■

Theorem 5.3.3 proves that, the proposed SERP can be utilized in at least one greedy solver and all convex relaxation methods and provide a “for any arbitrary but fixed” guarantee of recovery in $\ell_2 < c\ell_q$ norm sense even in case of noisy samples and compressible signals. In the next section, we show that the proposed projection matrix could be utilized in combinatorial solvers as well.

5.4 A Combinatorial solver for SERP

As before, assume $\mathbf{P} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ is generated by using Algorithm 8. In this section, we show how an already available combinatorial solver for binary projection matrices could be easily modified to support the proposed sparse real-valued projections. To that end, we have considered the Sequential Sparse Matching Pursuit (SSMP) decoder algorithm, introduced in [68] due to its low computational costs and also optimality of sample requirements $m = O(k \log n/k)$. We prove that by changing only one line in the SSMP algorithm, one can come up with a Modified version of Sequential Sparse Matching Pursuit (MSSMP) with a $\ell_2 < C\ell_2$ guarantee of recovery which is stronger than the $\ell_1 < C\ell_q$ guarantee in case of the original SSMP decoder algorithm. Algorithm 9 outlines the proposed MSSMP decoder for the class of sparse real-valued projection (SERP) matrices. The only difference between the MSSMP algorithm and SSMP is in line 5 of Algorithm 9, where in [68] for a fixed index i, $||P\left(x^{\{j\}} + ze_i\right) - y||_1$ is minimized. Note that for a fixed index i, the minimizer z to $||P\left(x^{\{j\}} + ze_i\right) - y||_2$ is

$$z = \frac{\boldsymbol{P}^T_{:,i}\left(y - \boldsymbol{P}x^{\{j\}}\right)}{||\boldsymbol{P}^T_{:,i}||_2^2} = \frac{\boldsymbol{P}^T_{\Omega_i,i}\left(y_{\Omega_i} - \boldsymbol{P}_{\Omega_i,:}x^{\{j\}}\right)}{||\boldsymbol{P}^T_{\Omega_i,i}||_2^2} \tag{5.21}$$

while the minimizer to $||P\left(x^{\{j\}} + ze_i\right) - y||_1$ in [68] is the median of the vector $y_{\Omega_i} - \boldsymbol{P}_{\Omega_i,:}x^{\{j\}}$.

To prove the $\ell_2 < C\ell_2$ guarantee of recovery for the proposed pair of sensing operator and combinatorial solver, we essentially follow the proof in [68]. However, here the w-RIP feature of the proposed projection matrix is utilized in the proof, which has different characteristics than its RIP-1 counterpart in the case of the binary projections in [68, 38, 37]. To provide an easy comparison between the presented proof and the one in SSMP, we tried to adapt the notations used in [68] as much as possible in here.

Algorithm 9: The Modified version of the Sequential Sparse Matching Pursuit (MSSMP), a combinatorial algorithm for class of sparse real-valued projections. For any vector x, the hard-thresholding operator $H_k[x]$ returns a vector which keeps the k largest coefficients of x and is zero in the rest of entries. In other words, $H_k[x]$ is the best k sparse approximation of the vector/signal x.

```
input  : $\boldsymbol{P} \in \mathbb{R}^{m \times n}$, $y \in \mathbb{R}^m$, $c \in \mathbb{N}$ and $k \in \mathbb{N}$
output: $x \in \mathbb{R}^n$
$x^{\{0\}} \leftarrow \mathbf{0}_n$ ;
for $j = 1$ to $T = O\left(\log ||x||\right)$ do
    $x' \leftarrow x^{\{j-1\}}$ ;
    for $S = 1$ to $(c-1)k$ do
        Find a coordinate $i$ and increment $z$ that minimizes $||\boldsymbol{P}\left(x^{\{j\}} + ze_i\right) - y||_2$ ;
        $x' \leftarrow x' + ze_i$ ;
    end
    $x^{\{j\}} \leftarrow H_k[x']$ ;
end
```

The algorithms SSMP and hence MSSMP are composed of two nested loops. The objective of the inner loop is to reduce the discrepancy of the estimation in the j-th iteration $||\boldsymbol{P}x^{\{j\}} - y||_l$, where $l = 1$ in case of SSMP and $l = 2$ for MSSMP. The outer loop just enforces that the signal estimation to be sparse. The following theorem (which itself is built based upon some other lemmas) proves that the modified SSMP can handle the proposed real valued projection matrix and guarantee $\ell_2 < C\ell_2$ recovery for a fixed but arbitrary signal.

Theorem 5.4.1. *Given the projection matrix $\boldsymbol{P}$, noisy samples $y = \boldsymbol{P}x + \mu$, the sparsity level $k = ||x||_0$, the combinatorial Algorithm 9 returns an estimate $\hat{x}$ such that $||\hat{x} - x||_2 = O(\mu)$ with a probability of at least $1 - \log ||x||_2/n^{\alpha}$ (where α is a constant) if w-RIP constant of $\boldsymbol{P}$ $\delta_{2k,p}$ is sufficiently small for $p \geq 1 - 1/n^{\alpha+1}$.*

Proof: Let us use $x^{\{j\}}$ to denote the estimated value of the true signal x in the beginning of the j-th iteration of the outer loop. Note that in the start of the outer loop, $x^{\{j\}}$ is k-sparse. By Lemma 5.4.2 (stated and proved below), if for a fixed regularization parameter c_l, the discrepancy of the estimation is still high (i.e. $||\boldsymbol{P}x^{\{j\}} - y||_2 \geq c_l||\mu||_2$) then there exists an index $i \in [n]$ and a step size z such that

$$||\boldsymbol{P}\left(x^{\{j\}} + ze_i\right) - y|| \leq \sqrt{1 - \frac{c_u}{k}}||\boldsymbol{P}x^{\{j\}} - y|| \tag{5.22}$$

where c_u is a function of w-RIP constant of P and also the expander quality of the corresponding bipartite graph. Thus after $O(k)$ iterations of the inner loop

$$||\boldsymbol{P}x' - y|| \leq \beta||\boldsymbol{P}x^{\{j-1\}} - y||,\ \beta = e^{-cu/2} \tag{5.23}$$

Noting that $\boldsymbol{P}x' - y = \boldsymbol{P}(x' - x) - \mu$, triangle inequality can be applied to get

$$||\boldsymbol{P}\left(x' - x\right)|| \leq \beta||\boldsymbol{P}(x^{\{j-1\}} - x)|| + (1 + \beta)\,||\mu|| \tag{5.24}$$

To relate the improvement on discrepancy to the improvement on signal estimation, one could utilize w-RIP on the set of columns corresponding to $Support\{x' - x\}$ to conclude

$$||x' - x|| \leq \beta\sqrt{\frac{1 + \delta}{1 - \delta}}||x^{\{j-1\}} - x|| + \frac{1 + \beta}{\sqrt{1 - \delta}}||\mu|| \tag{5.25}$$

Define constants

$$\alpha = 2\beta\sqrt{\frac{1 + \delta}{1 - \delta}},\ c_\mu = \frac{2\,(1 + \beta)}{\sqrt{1 - \delta}} \tag{5.26}$$

Then by Lemma 5.4.3 (stated and proved below), we have:

$$||x^{\{j\}} - x|| \leq \alpha||x^{\{j-1\}} - x|| + c_\mu||\mu|| \tag{5.27}$$

Now if the expander quality of the corresponding bipartite graph and also the number of samples are sufficiently high (leading to smaller values for β and δ respectively), then α would be less than one. In that case, running the outer loop for z iterations, we would have

$$||x^{\{z\}} - x|| \leq \alpha^z + \frac{2c_\mu}{1-\alpha}||\mu|| \tag{5.28}$$

Thus running the outer loop only for $O\left(\log ||x||\right)$ iterations leads to $||x^{\{z\}} - x|| = O(\mu)$.

As a final remark, note that Algorithm 9 requires $\tau = O(k \log ||x||)$ iterations and in each iteration it relies on w-RIP inequalities of $\boldsymbol{P}$ for signals which are non-zero at most in $2k$ indices (see proof of Lemma 5.4.2). Hence to recover an arbitrary but fixed signal x from its compressive samples with a target probability of q, $\boldsymbol{P}$ must have w-RIP of order $(2k, p)$ where $p^\tau \geq q$.■

In the following, we state the two lemmas used in the proof of the Theorem 5.4.1.

Lemma 5.4.2. *(Equivalent of Lemma 3 in [68]) Let* $y = \boldsymbol{P}x + \mu$. *If* x' *is* k*-sparse and* $||\boldsymbol{P}x' - y|| \geq c_l||\mu||$ *and* $\boldsymbol{P}$ *has a w-RIP of a proper order, then there exist an index* i *and a positive constant* c_u *such that*

$$||\boldsymbol{P}\left(x' - x'_i e_i + x_i e_i\right) - y||^2 \leq \left(1 - \frac{c_u}{k}\right)||\boldsymbol{P}x' - y||^2 \tag{5.29}$$

Proof: Define

$$\Delta = x' - x \tag{5.30}$$

Since $y = \boldsymbol{P}x + \mu$ then (5.29) can be re-expressed as:

$$||\boldsymbol{P}\left(\Delta - \Delta_i e_i\right)|| \leq \sqrt{1 - \frac{c_u}{k}}||\boldsymbol{P}\Delta - \mu|| \tag{5.31}$$

Note that from the assumption in the lemma

$$(c_l - 1)\,||\mu|| \leq ||\boldsymbol{P}x' - y|| - ||\mu|| \leq ||\boldsymbol{P}x' - y + \mu|| = ||\boldsymbol{P}\Delta|| \tag{5.32}$$

Since $||\Delta||_0 \leq 2k$ hence w-RIP implies that $||\boldsymbol{P}\Delta|| \leq \sqrt{1+\delta}||\Delta||$. Consequently:

$$||\Delta|| \geq \frac{c_l - 1}{\sqrt{1+\delta}}||\mu|| \tag{5.33}$$

Define

$$T = \mathit{Support}\{\Delta\} = \{i : \Delta_i \neq 0\} \tag{5.34}$$

Note that $0 \geq \mathit{Card}\{T\} = ||x - x'||_0 \leq 2k$. Assume sort entries of Δ in a decreasing order, i.e.

$$\forall i \in [\mathit{Card}\{\Delta\} - 1] : |\Delta_i| \geq |\Delta_{i+1}| \tag{5.35}$$

Consider G, the bi-partite graph corresponding to the projection matrix $\boldsymbol{P}$. Assume one enumerate the edges of the graph G in lexicographic order and forms the sets Ω_i^+ and Ω_i^- by following this rule: if (i, j) is the first edge going to the vertex j, then j would be included in Ω_i^-, otherwise it goes to the set Ω_i^- [68]. More formally

$$\Omega_i^+ = \{j : j \in \Omega_i,\ s.t.\ \nexists z < i, \boldsymbol{P}_{j,z} \neq 0\},\ \Omega_i^- = \Omega_i \backslash \Omega_i^+ \tag{5.36}$$

Note that sets Ω_i^+ are pair-wise disjoint. As before, let $D = dr$. With a slight abuse and for the simplicity of notations, let us use $|A|$ for a set to denote its cardinality. Since, $\boldsymbol{P}$ corresponds to an

$((c+1)k, D, q)$ expander graph (where $c \geq 1$), one has

$$\forall s \leq (c+1)k : \sum_{i=1}^{s} |\Omega_i^-| \leq qDs \tag{5.37}$$

For a fixed constant $c_s > 1$, define:

$$T^+ = \{i : i \in T \; and \; |\Omega_i^-| < qDc_s\}, \; T^- = T \backslash T^+ \tag{5.38}$$

As in [16], our goal is to show that (a) most of energy of Δ is in coordinates T^+ and (b) there is an index in T^+, such that (5.29) holds true for that index.

Suppose that, entries of T are sorted increasingly, i.e. $T^- = \{u_1, u_2, \ldots\}$ where $u_i < u_{i+1}$. Now since $u_k \geq k$, hence

$$\sum_{i=1}^{k} |\Omega_{u_i}^-| \leq \sum_{i=1}^{u_k} |\Omega_i^-| \tag{5.39}$$

By definition $\forall i \in T^-$ we have $|\Omega_i^-| > qDs$ and therefore

$$kqDc_s < \sum_{i=1}^{k} |\Omega_{u_i}^-| \tag{5.40}$$

On the other hand since the graph G is $D = rd$ left regular, thus

$$\sum_{i=1}^{u_k} |\Omega_i^-| \leq qDu_k \tag{5.41}$$

From the last two equations, we get

$$u_k \geq kc_s \tag{5.42}$$

The immediate consequence is that for all $i \in [k]$:

$$|\{1, 2, \ldots, u_i\} \cap T^+| \geq k\left(c_s - 1\right) \tag{5.43}$$

As in [68] for any $i \in [k]$, let S_i be the set containing the smallest $(c_s - 1)$ elements of $\{1, 2, \ldots, u_i\} \cap T^+ \backslash \{\cup_{i=1}^{k-1} S_i\}$. Noting that $\forall j \in S_i : |\Delta_j| \geq |\Delta_{u_i}|$ and $|S_i| \geq (c_s - 1)$, one has

$$||\Delta||^2 \geq ||\Delta_{T^- \cup \{\cup_i S_i\}}||^2 = \sum_{u_i \in T^-} \left(\Delta_{u_i}^2 + \sum_{j \in S_i} \Delta_j^2\right) \geq \tag{5.44}$$

$$c_s \sum_{u_i \in T^-} \Delta_{u_i}^2 = c_s ||\Delta_{T^-}||^2 \tag{5.45}$$

Recalling that $||\Delta||^2 = ||\Delta_{T^-}||^2 + ||\Delta_{T^+}||^2$, it can be concluded that

$$||\Delta_{T^+}|| \geq \sqrt{1 - \frac{1}{c_s}} ||\Delta|| \tag{5.46}$$

Define:

$$gain_i = ||\boldsymbol{P}\Delta - \mu||^2 - ||\boldsymbol{P}\left(\Delta - \Delta_i e_i\right) - \mu||^2 = \tag{5.47}$$

$$||\boldsymbol{P}\Delta||^2 - ||\boldsymbol{P}\left(\Delta - \Delta_i e_i\right)||^2 - 2\mu^T \boldsymbol{P}\Delta_i e_i \tag{5.48}$$

Since the i-th column of $\boldsymbol{P}$ is non-zero only in row indices Ω_i, hence $\forall j \in [m] \backslash \Omega_i : (\boldsymbol{P}\Delta)_j = (\boldsymbol{P}\Delta - e_i \Delta_i)_j$. It is straightforward to verify that:

$$gain_i = 2\left(\boldsymbol{P}\Delta\right)_{\Omega_i}^T \left(\boldsymbol{P}\Delta_i e_i\right)_{\Omega_i} - ||\left(\boldsymbol{P}\Delta_i e_i\right)_{\Omega_i}||^2 - 2\mu^T \boldsymbol{P} e_i \Delta_i \tag{5.49}$$

Using the fact that $\Delta = \Delta_{T+} + \Delta_{T-}$ and summing $gain_i$ over all $i \in T^+$ yield:

$$\sum_{i \in T^+} gain_i = 2\left(\boldsymbol{P}\Delta\right)_{\Omega_i}^T \boldsymbol{P}\Delta_{T+} - \sum_{i \in T^+} ||\left(\boldsymbol{P}\Delta_i e_i\right)_{\Omega_i}||^2 - 2\mu^T \boldsymbol{P}\Delta_{T+} = \tag{5.50}$$

$$2||\boldsymbol{P}\Delta_{T+}||^2 + 2\left(\boldsymbol{P}\Delta_{T-}\right)^T \left(\boldsymbol{P}\Delta_{T+}\right) - \sum_{i \in T^+} ||\left(\boldsymbol{P}\Delta_i e_i\right)_{\Omega_i}||^2 - 2\mu^T \boldsymbol{P}\Delta_{T+} \tag{5.51}$$

By w-RIP on support of T^+, one has:

$$||\left(\boldsymbol{P}\Delta_i e_i\right)_{\Omega_i}||^2 \leq (1+\delta) \sum_{i \in T^+} ||\Delta_i||^2 = (1+\delta)\, ||\Delta_{T+}||^2 \tag{5.52}$$

Using the lower bound of w-RIP for T^+ one has $||\boldsymbol{P}\Delta_{T+}||^2 \geq (1-\delta)||\Delta_{T+}||^2$ and hence (5.51) can be simplified to

$$\sum_{i \in T^+} gain_i \geq (1-3\delta)\, ||\Delta_{T+}||^2 + 2\left(\boldsymbol{P}\Delta_{T-}\right)^T \left(\boldsymbol{P}\Delta_{T+}\right) - 2\mu^T \boldsymbol{P}\Delta_{T+} \tag{5.53}$$

Note that by Cauchy-Schwarz inequality

$$\mu^T \boldsymbol{P}\Delta_{T+} \leq \frac{1+\delta}{c_l - 1}||\Delta_{T+}||^2 \tag{5.54}$$

Combining two last equations we have:

$$\sum_{i \in T^+} gain_i \geq \left(1 - 3\delta - \frac{2\,(1+\delta)}{c_l - 1}\right) ||\Delta_{T+}||^2 + 2\left(\boldsymbol{P}\Delta_{T-}\right)^T \left(\boldsymbol{P}\Delta_{T+}\right) \tag{5.55}$$

Using w-RIP of $\boldsymbol{P}$ on two partitions of T, we have (see [28])

$$||\boldsymbol{P}_{T+}^T \boldsymbol{P}_{T-}|| \leq \delta \tag{5.56}$$

and hence

$$|\left(\boldsymbol{P}\Delta_{T-}\right)^T\left(\boldsymbol{P}\Delta_{T+}\right)| \leq \delta||\Delta_{T+}||||\Delta_{T-}|| \tag{5.57}$$

Substituting (5.57) into (5.56) yields

$$\sum_{i\in T^+} gain_i \geq \left(1-3\delta-\frac{2\left(1+\delta\right)}{c_l-1}\right)||\Delta_{T+}||^2 - 2||\Delta_{T+}||||\Delta_{T-}|| \tag{5.58}$$

By (5.46) and (5.58), we have

$$\sum_{i\in T^+} gain_i \geq \left(1-3\delta-\frac{2\left(1+\delta\right)}{c_l-1}\right)||\Delta_{T+}||^2 - 2\delta\sqrt{\frac{1}{c_s}-\frac{1}{c_s^2}} \geq C||\Delta||^2 \tag{5.59}$$

for

$$\bar{C} = \left(\left(1-\frac{1}{c_s}\right)\left(1-3\delta-\frac{2\left(1+\delta\right)}{c_l-1}\right)-2\delta\sqrt{\frac{1}{c_s}-\frac{1}{c_s^2}}\right) \tag{5.60}$$

Recall that $|T^+| \leq 2k$. Thus there exists an index $j \in [n]$ such that:

$$gain_j \geq \frac{\bar{C}||\Delta||^2}{|T^+|} \geq \frac{\bar{C}||\Delta||^2}{2k} \tag{5.61}$$

To relate the gain to discrepancy $||\boldsymbol{P}x'-y||$, it can be shown that

$$\sqrt{1+\delta}||\Delta|| \geq ||\boldsymbol{P}\Delta|| = ||\boldsymbol{P}x'-\boldsymbol{P}x-\mu+\mu|| \geq ||\boldsymbol{P}x'-y||-||\mu|| \tag{5.62}$$

By formula (5.33)

$$||\Delta|| \geq \frac{||\boldsymbol{P}x'-y||}{\sqrt{1+\delta}\left(1+\frac{1}{c_l-1}\right)} \tag{5.63}$$

Now one can simplify (5.61) to

$$\exists j \in [n] : gain_j \geq \frac{c_u}{k} ||\boldsymbol{P}x' - y||^2 \tag{5.64}$$

for

$$c_u = \frac{\bar{C}}{2\left(1+\delta\right)\left(1 + \frac{1}{c_l - 1}\right)^2} \tag{5.65}$$

Note that, to improve the descrepancy of estimation and hence reducing the error of estimation, we require that c_u to be strictly positive. Since the denominator of c_u is strictly positive, it only suffices to have $\bar{C} > 0$. This can be achieved if δ (w-RIP constant) is sufficiently small. For instance, for $c_s = 2$ and $c_l = 4$, $\bar{C}$ would be positive if $\delta \leq 0.058$. ■

Lemma 5.4.3. *(Equivalent of Lemma 6 in [68]) For any k-sparse vector x and any vector x' we have:*

$$||H_k\left[x'\right] - x|| \leq 2||x' - x|| \tag{5.66}$$

Proof: The proof of Lemma 6 in [68] is for ℓ_1 norm, however it holds for any other valid norm as well. Nevertheless for the sake of completeness, here we present the proof. Note that, $H_k\left[x'\right]$ is the closest k-sparse vector to x'. Hence $||H_k\left[x'\right] - x'|| \leq ||x - x'||$. Consequently by triangle-inequality we have:

$$||H_k\left[x'\right] - x|| \leq ||H_k\left[x'\right] - x'|| \leq ||x' - x|| + ||x' - x|| \leq 2||x' - x|| \tag{5.67}$$

Let us conclude this section with some discussion about the complexity of the MSSMP decoder. Computing $\boldsymbol{P}x$ in case of a sparse binary projection is achieved by adding some entries of x (since all non-zero entries of $\boldsymbol{P}$ are one). However, for a sparse real-valued projection, computing $\boldsymbol{P}x$

must be performed through computing a series of inner products. Clearly, this update/computation is heavier for a sparse real-valued projection matrix than the one in case of a sparse binary projection matrix. However, it is only a constant factor worse (the complexity of the multiplication divided by the complexity of summation). On the other hand, computing $\boldsymbol{P}x$ in case of a sparse real-valued projection matrix is much lighter than when $\boldsymbol{P}$ is a random-dense matrix. Also as stated before, the algorithms of MSSMP and SSMP are exactly the same, except line 5 of Algorithm 9. In that line of SSMP, the median of d numbers are computed while the same line in MSSMP demands the orthogonal projection of two vectors in $\mathbb{R}^{rd}$. Although this operation is heavier than computing the median of d numbers, order-wise, these two operations have exactly the same complexity (since $r = O(1)$). Consequently, the complexity order of MSSMP and SSMP are the same.

5.5 Simulation Results

In this section, we present some numerical results regarding the performance of the proposed sparse real-valued projection matrices (SERP) under combinatorial, greedy, and convex-relaxation approaches. Here, we have considered two scenarios: (i) when compressive samples are noiseless and (ii) the samples are contaminated by noise.

For the noiseless case, the performance of the pair of traditional sparse binary projection matrices and SSMP as the decoder is compared with the proposed pair of sparse real-valued projection and MSSMP in the decoder. More specifically, we fixed the signal length to $n = 10,000$ and considered signal sparsities $k = ||x||_0 = 100, 200, 300, \dots, 800$. For each sparsity level, we considered six levels of sampling $m = 3k, 4k, \dots, 8k$. For each configuration of k and m, we generated a signal $x \in \mathbb{R}^n$ and uniformly at random selected k of its indices and let the signal values at those indices be a uniform random variable in the range of $[-0.5, 0.5]$. Then, we recorded the

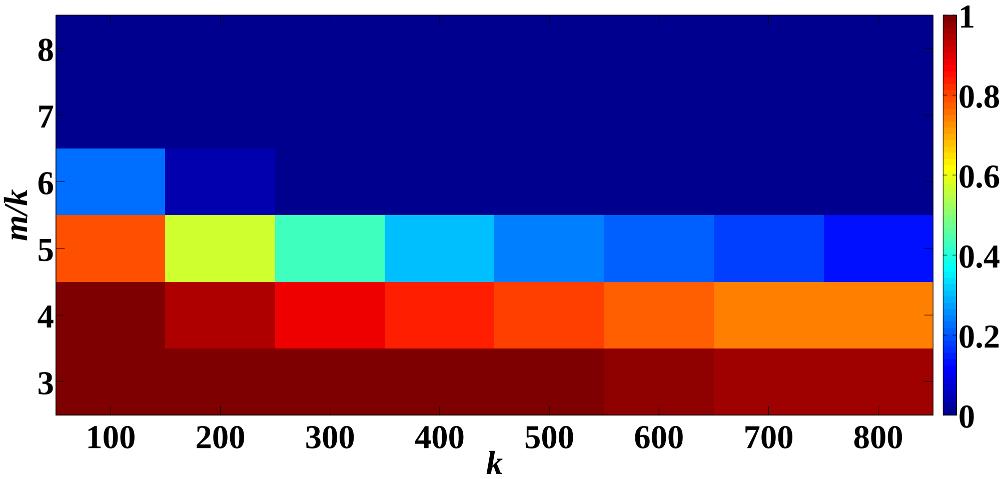


Figure 5.1: Relative error of reconstruction ($\|\Delta(y, \boldsymbol{P}) - x\|/\|x\|$) for 50 random signals x of length $n = 10,000$ as a function of the sparsity level $k = \|x\|_0$ and sampling ratio m/k, when the projection matrix is sparse binary and the solver is SSMP.

average of relative reconstruction error ($||\Delta(y, \boldsymbol{P}) - x||/||x||$) in 50 independent trials. The results for the pair of a sparse binary projection matrix with $d = 18$ and SSMP decoder are presented in Figure 5.1. For the same value of d, the results for MSSMP and sparse real-valued projection matrices with $r = 1$ and $r = 2$ are respectively shown in Figure 5.2 and Figure 5.3. As shown in those plots, MSSMP and the sparse real-valued projections lead to (sometimes much) lower levels of distortions in recovery when the number of samples are not sufficiently large (e.g. $m = 4k$ and $m = 5k$ for all values of k). However, both of these two approaches (MSSMP and SSMP) introduce virtually no error in the recovery process when $m > 6k$. Note that under the MSSMP decoder, the results from sensing matrices with $r = 1$ are often less distorted than those from matrices with $r = 2$.

Now let us focus on the scenario when samples are noisy, i.e. $y = \boldsymbol{P}x + \mu$ where $||\mu|| > 0$. Fixing the signal length, sparsity and number of samples respectively to $n = 5000$, $k = 250$ and $m = 5k = 1250$, we varied the relative error in sampling ($||\mu||/||\boldsymbol{P}x||$) from 0.1 to one in steps of size 0.1. For each relative error in sampling, we generated 100 random signals and projection

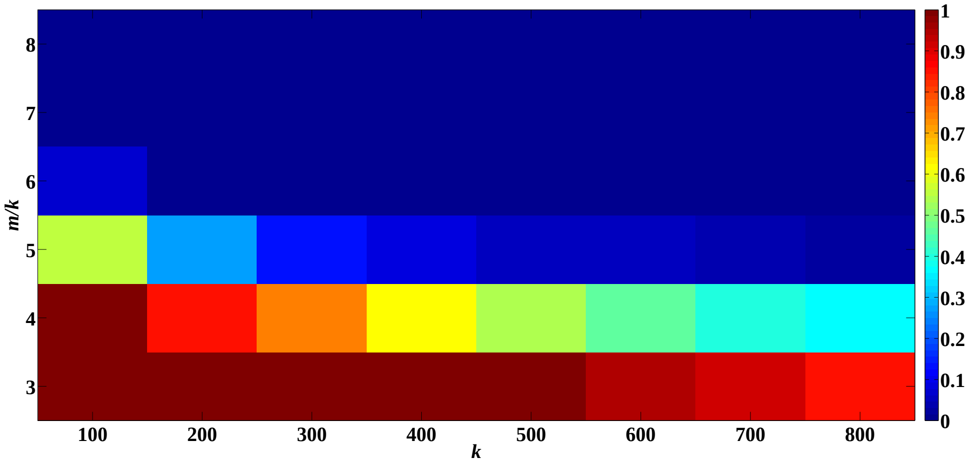


Figure 5.2: Relative error of reconstruction ($\|\Delta(y, \boldsymbol{P}) - x\|/\|x\|$) for 50 random signals x of length $n = 10,000$ as a function of the sparsity level $k = \|x\|_0$ and sampling ratio m/k, when the projection matrix is a sparse real-valued sensing matrix (SERP) with $r = 1$ and the solver is MSSMP.

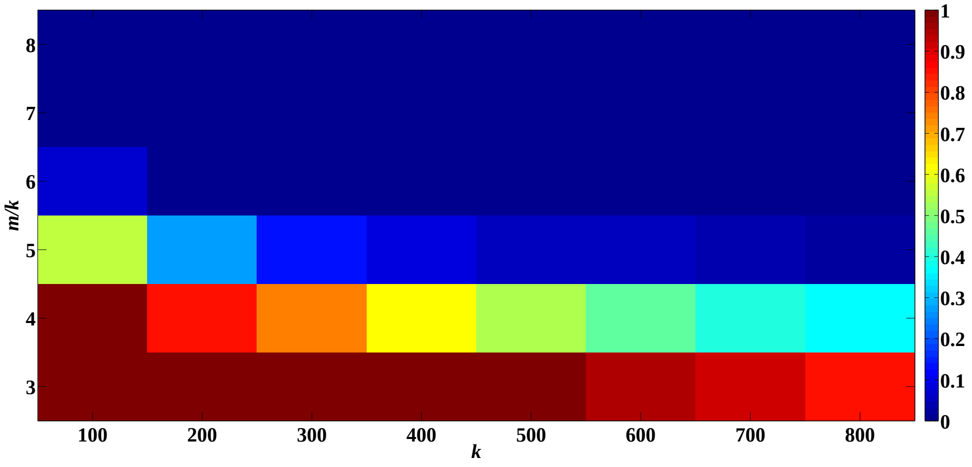


Figure 5.3: Relative error of reconstruction ($\|\Delta(y, \boldsymbol{P}) - x\|/\|x\|$) for 50 random signals x of length $n = 10,000$ as a function of the sparsity level $k = \|x\|_0$ and sampling ratio m/k, when the projection matrix is a sparse real-valued sensing matrix (SERP) with $r = 2$ and the solver is MSSMP.

matrices as in the noiseless case scenario (i.e. uniformly at random selecting the indices of the signal to be non-zero and populate those indices uniformly in the range of $[-0.5, 0.5]$). Then, the averages of relative errors in reconstruction as a function of the relative error in sampling were recorded for all three types of CS-decoders and are presented in Figure 5.4-Figure 5.6. Let us, we briefly highlight the key results of these plots. As shown in Figure 5.4, the pair of sparse real-valued projections and MSSMP decoder leads to higher qualities of recovery, when compared to the pair of sparse binary projections and SSMP. Within the class of sparse real-valued projections, when the relative noise energy is small, projection matrices with $r = 1$ seems to be better sensing operators. However, as the noise energy increases, we introduce less error in the recovery process if non-zero entries of the sparse projection matrix are Gaussian random variables in $\mathbb{R}^2$ (i.e. $r = 2$). When BP [17] is employed as the decoder, the class of sparse real-valued projection matrices are virtually as good as dense Gaussian projection matrices (see Figure 5.5). Within the class of sparse real-valued projections and for fixed value of d, BP consistently introduces less errors in the approximation for matrices with $r = 2$. For evaluating the performance of the sparse real-valued projection in a greedy CS framework, we selected the algorithm of Cosamp [28] for the decoder side. Figure 5.6 shows that Cosamp is more sensitive to the deployed projection matrix compared to BP when the projection matrix is from the class of sparse real-valued projections. Figure 5.6 and Figure 5.5 suggest that for a fixed value of d and under non-combinatorial class of solvers, deploying sparse real-valued projection matrices with parameter r being slightly greater than one, leads to less error in the recovery process.

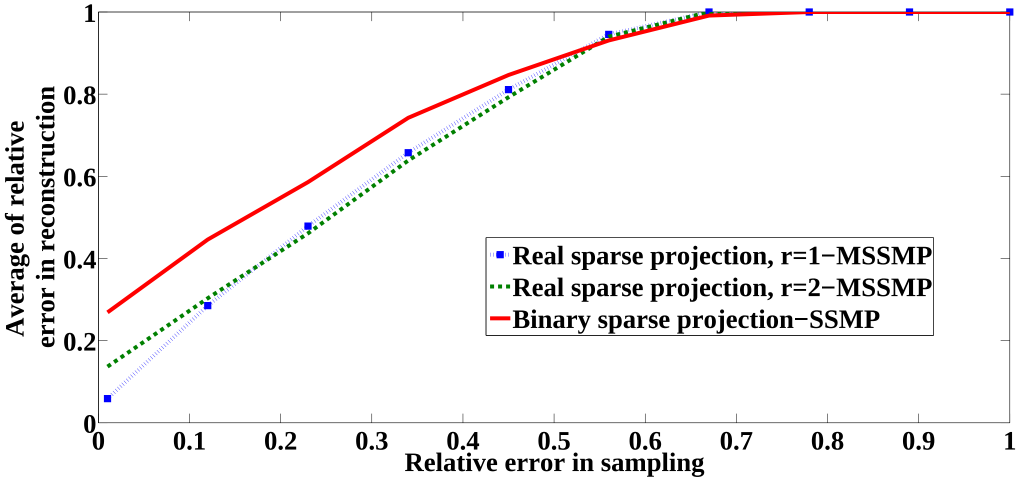


Figure 5.4: Comparing the performance of the sparse real valued projection matrices and MSSMP with sparse binary projections under SSMP decoder when samples are noisy.

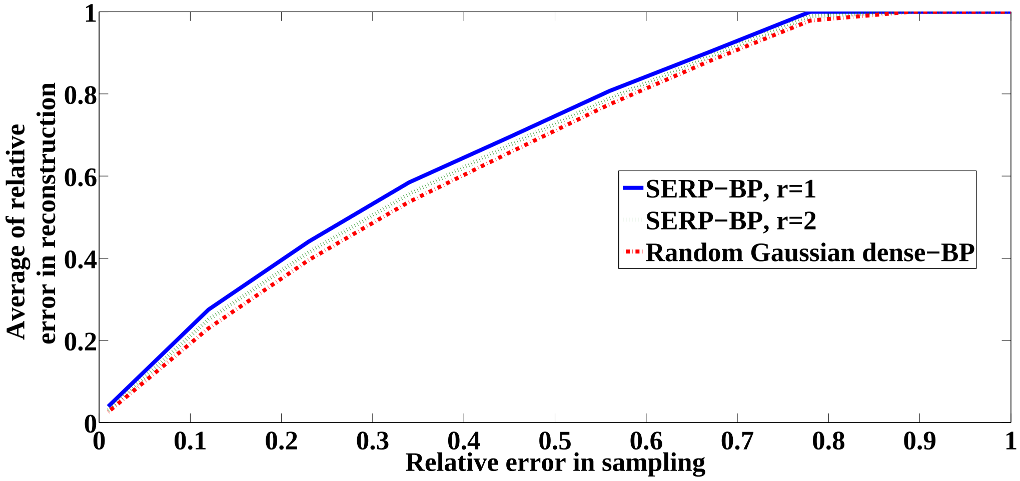


Figure 5.5: Comparing the performance of the sparse real-valued and dense Gaussian projection matrices when BP is the decoder and samples are noisy.

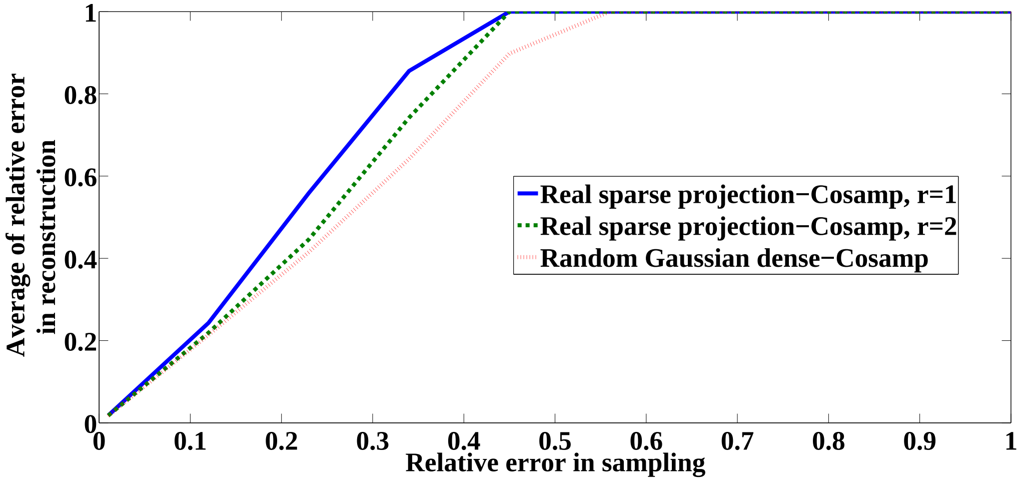


Figure 5.6: Comparing the performance of the sparse real-valued and dense Gaussian projection matrices when Cosamp is the decoder and samples are noisy.

Chapter 6

Future works

6.1 Introduction

CRISP, HCS and many other fast combinatorial algorithms (e.g. [38, 37]) follow the traditional (i.e. not message passing) approach of combinatorial algorithms where each coefficient of the unknown signal is estimated only once. The main advantage of such approach is that the decoding process would be extremely fast. On the negative side, such algorithms suffer from high sample requirements. Algorithm 10 shows how typical combinatorial approaches to CS operate for solving (6.1) under the zero pseudo norm $p = 0$. Our initial effort was to utilize the w-RIP property of SERP matrices to show that one can find an extremely low complexity traditional combinatorial algorithm under SERP encoding matrices, without sacrificing sample requirements. As we show in the rest of this chapter, such algorithm indeed exists when samples are not affected by noise. However, the moment that samples are contaminated by noise, we would be faced with some challenging problems. We leave solving those challenges for future works and propose some possible research paths to that end.

6.2 Another look at combinatorial algorithms

Algorithm 10 summarizes a traditional approach for combinatorial algorithms to CS. In the following, we show that Algorithm 11 could be perceived as the recursive formulation of Algorithm

10 for solving the equation (6.1) when p is set to zero in the third line of Algorithm 11.

$$\arg\min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{x}}\|_p : \ s.t. \ \boldsymbol{y} = \boldsymbol{P}\hat{\boldsymbol{x}} \tag{6.1}$$

Algorithm 10: A typical combinatorial algorithm for CS. The set of coefficients which their values are determined, is denoted by z.

```
input  : $\boldsymbol{P} \in \mathbb{R}^{m\times n}$ and $y \in \mathbb{R}^m$
output: $\hat{x} \in \mathbb{R}^n$
$\hat{y} = y, \ z = \emptyset$ ;
while $z \neq [n]$ do
    Find $l \subseteq [m]$ such that $||x_{\omega_l}||_0 = 1$ ;
    Let $\hat{x}_{\omega_l}$ denotes estimate of values $x_{\omega_l}$;
    For $\theta = \omega_l : \hat{y}_{\Omega_\theta} \leftarrow y_{\Omega_\theta} - \boldsymbol{P}_{\Omega_\theta,\omega_l} x_{\omega_l}, \ z \leftarrow z \cup \omega_l$ ;
end
```

Algorithm 11: $\Delta(\boldsymbol{P}, \theta, y)$ the recursive formulation of Algorithm 10.

```
input  : $\boldsymbol{P} \in \mathbb{R}^{m\times n}$, $\theta$ and $y \in \mathbb{R}^m$
output: $\hat{x} \in \mathbb{R}^n$
$\hat{x} = \mathbf{0}_n$ ;
Find $l \subseteq [m]$ where $||x_{\omega_l}||_0 = 1$ ;
$\hat{x}_{\theta_1} = \arg\min_{\xi} \|\xi\|_p : \ s.t. \ y_l = \boldsymbol{P}_{l,\theta_1}\xi$ ;
$\hat{x}_{\theta\backslash\theta_1} \leftarrow \Delta\left(\boldsymbol{P}, \theta\backslash\theta_1, y - \boldsymbol{P}\hat{x}\right)$ ;
```

Initializing with $\theta = [n]$ and after finding θ_1 such that $\|\hat{x}_{\theta_1}\|_0 = 1$ (i.e. finding an isolated non-zero), Algorithm 11 recursively finds a subset of signal coefficients in the un-recovered part $\theta_{i+1} \subseteq [n] \backslash \cup_{j\in[i]} \theta_j$ such that $\|x_{\theta_{i+1}}\|_0 = 1$ where θ_j denotes the value of θ_1 (line 2 in Algorithm 11) in the j-th level of the stack. This process continues until all non-zero coefficients are recovered. Note that, for a vector x with a single non-zero entry ($\|x\|_0 = 1$) and $y = \boldsymbol{P}x$, the solutions to (6.1) for $p = 0$ and $p = 1$ are equal by the triangle inequality. Also note that for any signal x and two disjoint subsets of $a \subseteq [n]$ and $a^c := [n]\backslash a$ one has: $\|x_a\|_p + \|x_{a^c}\|_p = \|x\|_p$ both for $p = 0$ and $p = 1$ (see line 3 and 4 in Algorithm 11). Combining this observation with the equality of solutions of (6.1) for $p = 0$ and $p = 1$, one can conclude that in Algorithm 11, the solutions for $p = 0$ and $p = 1$ are exactly the same. Hence, typical combinatorial approaches to CS (Algorithm 10) could be recast as a recursive ℓ_1 minimizer. The success of the aforementioned recursive algorithm depends on:

- The ability of step 3 in Algorithm 10 to find isolated non-zeros.
- The quality of the projection matrix $\boldsymbol{P}$ in recoverability

More specifically, the first condition demands that the decoder at the i-th iteration could truly find $\theta_i = \omega_l$, a subset of signal coefficients, containing only one non-zero ($\|x_{\theta_i}\|_0 = 1$). Meanwhile, the second requirement states that, at least one isolated non-zero must exist throughout the recovery process and hence such process shall not be halted before the perfect recovery. It has been shown [37, 38] that certain sparse matrices, for instance the adjacency matrices of high quality expander graphs provide the second condition. In the following, we assume that the employed sparse projection matrix has the second condition, meaning that if isolated non-zeros are correctly identified in step 3 of Algorithm 10, then the recovery process will not halt. To that end and to make our analysis simple, we adapt the SERP matrices, introduced in the previous chapter. That is, each column of $\boldsymbol{P}$ is non-zero in exactly $rd = O\left(\log n\right)$ random indices for a constant $r \geq 1$.

Having the guarantee of recoverability, now our concern will be to satisfy the first condition, i.e. correctly finding isolated non-zeros. The most common approach for finding isolated non-zeros (in case of a binary sparse projection matrix) is to apply a voting mechanism (see [68, 35, 38, 37]). More specifically, assume that there exists a set of sample indices of $l \subseteq [m]$ such that all of them vote the same value and they span the same single coefficient i.e.:

$$\exists l \subseteq [m] : \; \forall p, q \in l : y_p = y_q, \; \cap_{j \in l} \omega_j = i \qquad (6.2)$$

Then typical combinatorial algorithms infer that $\hat{x}_i$ equals to that vote and $\hat{x}_j = 0$ for $j \in \omega_l \backslash i$. Such conclusion might imply certain presumptions about the underlying signal. For example, it has been assumed in [35] that no two subsets $\theta_1, \theta_2 \subseteq [n]$ exist such that $\|x_{\theta_2}\|_1 \approx \|x_{\theta_2}\|_1$ or other assumptions in some other methods such as non-negativity of the signal and so on. Consequently, the applicability of the aforementioned approaches will be restricted to signals, satisfying corresponding conditions. On the other hand, due to the non-binary nature of the proposed sparse projection matrix $\boldsymbol{P}$, the mechanism of voting is not applicable in case of $\boldsymbol{P}$. We introduced the notions of "alignment" and "isolation" in Chapter 2. In this chapter, we show that the method of voting is a special case of such notions (alignment and isolation) in case of binary matrices. Furthermore, we present necessary and sufficient conditions for detecting isolated non-zeros under real-valued sparse projection matrices.

As before, assume $\boldsymbol{P} \in \mathbb{R}^{m \times n}$ is a matrix where each of its columns is non-zero in exactly $rd = O(\log n)$ random indices. Also assume that, $\boldsymbol{P}$ is an instance of SERP matrices. Since $\boldsymbol{P}$ corresponds to the adjacency matrix of a high quality expander graph (with high probability) [37] and the analysis of [37] is not based on values of $\boldsymbol{P}$, therefore the recoverability holds for $\boldsymbol{P}$. Consequently in here, we only discuss lines 3 and 4 in Algorithm 11. We show, how one can find subsets of signal coefficients containing only one non-zero, by examining compressive samples. To that end, it would be beneficial to have a brief review of notions of isolation and alignment in here.

We say "x_i is isolated in sample indices of $l \subseteq [m]$ if $\|x_{\omega_l}\|_0 = 1$ and $x_i \neq 0$. In other words, x_i is the only non-zero signal coefficient (among indices ω_l) spanned by samples y_l. Clearly, since $\boldsymbol{P}_{:,i}$ is non-zero only in indices Ω_i, therefore $l \subseteq \Omega_i$. For $q \subseteq [m]$ we say "y_q is aligned with the j-th column of $\boldsymbol{P}$" if: $\forall i \in q : \boldsymbol{P}_{i,j} \neq 0$ and $\exists \alpha \neq 0 : y_q = \alpha \boldsymbol{P}_{q,j}$. Note that, if $\boldsymbol{P}$ is binary (i.e. $\boldsymbol{P}_{q,j}$ is an all one vector), then the definition of alignment reads as "there exists a set of samples (y_q), all voting the same value of α". This is exactly the voting mechanism, employed in combinatorial algorithms. This verifies our claim that the voting mechanism is in fact an alignment in the special case of binary matrices. Assume that $x_i \neq 0$ is isolated in sample indices of $l \subseteq \Omega_i \subseteq [m]$. Then:

$$y_l = \boldsymbol{P}_{l,\omega_l} x_{\omega_l} = x_i \boldsymbol{P}_{l,i} \tag{6.3}$$

In words, if x_i is isolated in sample indices of l then y_l will be aligned with the i-th column of $\boldsymbol{P}$. Thus isolation is a necessary condition for alignment. Now, if isolation is a sufficient condition for alignment as well, then isolated non-zero coefficients in the signal could be found by looking for alignment of samples among the columns of $\boldsymbol{P}$. For now assume that isolation and alignment are equivalent. Then step 3 and 4 in Algorithm 11 could be implemented as follows: to see whether x_i is isolated in a set of unknown samples, one might consider $\boldsymbol{P}_{\Omega_i,i}$ and looks for $q \subseteq \Omega_i$ such that $y_q = \alpha \boldsymbol{P}_{q,j}$. If there exists any q with those conditions, then (6.3) could be applied to infer $x_i = \alpha$. In the rest of this section, we identify the underlying conditions under which isolation is a sufficient condition for alignment.

Here, it is important to highlight the impact of the size of the set $l \subseteq \Omega_i \subseteq [m]$ in our analysis. Since, in an extreme case when $|l| = 1$ in (6.3), there always exists a meaningless alignment. Recall that, each column of $\boldsymbol{P}$ is non-zero in exactly $rd = O(\log n)$ entries. In [37], it has been shown that in case of a sparse binary projection matrix based on the adjacency matrix of a left d-regular random graph, there exists at least $\tau \geq 1 + d/2$ samples which span only one (and the same) non-zero in all stages of decoding. Hence, in here we have $l = \tau > rd/2 = O(\log n)$ and ignore alignments with size smaller than $rd/2$ in the decoder.

As before, assume y_l is aligned with the i-th column of $\boldsymbol{P}$, where $l > d/2$. Since by definition $y_l = \boldsymbol{P}_{l,\omega_l} x_{\omega_l}$ and $\|y_l\|_2 > 0$, there are two possible cases here: (a) $\|x_{\omega_l}\|_0 = 1$ or (b) $\|x_{\omega_l}\|_0 > 1$.

In the first case, since y_l is non-zero in all entries ($\|y_l\|_0 = |l|$) and for all $j \in \omega_l \backslash i$ we have $\|\boldsymbol{P}_{l,j}\|_0 < |l|$ (this comes from the expander property of $\boldsymbol{P}$) thus one can conclude that, x_i is the only non-zero among indices of ω_l and its value can be easily computed by (6.3). In words, in the first case, isolation is a sufficient and necessary condition for alignment. Unfortunately, a similar argument generally does not hold true for the second case ($\|x_{\omega_l}\|_0 > 1$) for $\boldsymbol{P}$. Nevertheless, here we show how if the parameter r in SERP matrices is sufficiently large, then the second case would never happen with a very high probability. To that end, we follow these steps (a) for any subset of samples $l \subseteq [m]$ with $l \geq 1 + d/2 = O(\log n)$ first we find $Q = \max\{\|x_{\omega_l}\|_0\}$, the maximum number of non-zeros which will be spanned (with high probability) by samples l. (b) Then we find the value of r, such that every Q columns of $\boldsymbol{P}_{l,\omega_l}$ would be independent. Finally by using elementary arguments, one can show that it is impossible to have $\|x_{\omega_l}\|_0 > 1$ and y_l to be aligned with the i-th column of $\boldsymbol{P}$.

For a fixed subset of sample indices l, define the binary random variable X_i as:

$$X_i = \begin{cases} 1 & x_i \neq 0,\ i \in \omega_l \\ 0 & otherwise \end{cases} \tag{6.4}$$

Note that $Q = \sum_{i=1}^{n} X_i$. Now since the probability that $x_i \neq 0$ is independent from the probability of the event that $i \in \omega_l$, thus:

$$\Pr(X_i = 1) = \frac{k}{n}\left(1 - \frac{\binom{m-l}{d}}{\binom{m}{d}}\right) \tag{6.5}$$

For now assume that $r = 1$, later we derive how large we should set r. Assuming k (and hence m) is much larger than one, one can use the approximation: $1/(m - d + 1) \approx 1/m$. Using the inequality $1 + \alpha \leq e^{\alpha}$, it can be shown that

$$\mu = E[Q] = n \Pr(X_i = 1) \leq k \left(1 - \exp(l/m)\right)^d \tag{6.6}$$

Recalling that $d = O(\log n) = c_1 \log n$, $d/2 < l < d$ and $m = O(k \log n) = c_2 k \log n$, it can be easily derived that $\mu = O(1)$. Now it only remains to bound the deviation of Q from μ. Fortunately, since Q is the sum of Bernoulli random variables, there exists good concentration inequalities for it. For instance, by using Chernoff bound for Poisson trials, it can be easily shown that for $\delta \geq 6.389$ (which makes $\log(1+\delta) > 2$) we have:

$$\Pr(Q \geq \mu + \delta\beta \log n) \leq \frac{1}{n^{\beta}} \tag{6.7}$$

Now if the spark [43] of $\boldsymbol{P}_{l,\omega_l}$ is more then Q, then it is impossible to have y_l aligned with the i-th column of $\boldsymbol{P}$ and at the same time $\|x_{\omega_l}\|_0 > 1$. This can be proved by contradiction. More specifically, assume $1 < \|x_{\omega_l}\| < Q$ and y_l is aligned with the i-th column of $\boldsymbol{P}$ (i.e. $y_l = \boldsymbol{P}_{l,\omega_l} x_{\omega_l} = \alpha \boldsymbol{P}_{l,j}$). Define a new vector x' where $x'_j = x_j - \alpha$ and $x'_i = x_i$ for $i \in \omega_l \backslash j$. Then $\boldsymbol{P}_{l,\omega_l} x' = 0$ and $\|x'\|_0 \leq Q$ which contradicts the presumption that the spark of $\boldsymbol{P}_{l,\omega_l}$ is more than Q. In summary, if the parameter r in SERP matrices is tuned such that (6.7) is satisfied, then alignment translates into isolation with high probability .

6.3 noise

It seems that since w-RIP holds for SERP matrices and typical combinatorial decoding algorithms are in fact quick implementation of BP/linear programming [17] (minimizing ℓ_1 norm of the solution subject to a set of equations), hence such combinatorial algorithm should be robust when samples are noisy. A naive justification could be following these three steps (a) the decoding algorithm is trying to find the sparsest solution and for each sub-problem it never makes a mistake (b) using the triangle inequality, the solution found for that sub-problem has the minimum ℓ_1 norm and (c) applying induction, the algorithm should find all non-zero coefficients correctly. Since w-RIP is a sufficient condition for the robustness of BP (at least in a for any arbitrary signal), this also proves that our algorithm should be robust in the presence of noise.

On the other hand, we are facing another challenge when samples are contaminated by noise and that is: when samples are noisy, then no perfect alignment can be found. This problem might be solved by allowing ϵ-*alignment*. We say that a vector h is ϵ-aligned with the i-th column of $\boldsymbol{P}$ if:

$$|\cos^{-1}\left(\frac{|\boldsymbol{P}_{:,i}^T h|}{\|\boldsymbol{P}_{:,i}\|_2\|h\|_2}\right)| \leq \epsilon \tag{6.8}$$

Since the argument of $\cos^{-1}$ is strictly positive and $\cos^{-1}$ is strictly decreasing, one might only consider the argument inside parentheses when using the notion of ϵ-alignment during the proof.

As noted before, RIP translates into the fact that sub-matrices of a matrix behave like an orthonormal projection. Suppose $y = \boldsymbol{P}x + e$ where e is the noise vector, $\boldsymbol{P}$ has a small value of isometry constant δ_k and x is non-zero in index of i. Then we can conclude that: columns of $\boldsymbol{P}$ are so far apart from each others that the cosine of the angle between vector $h = y_{\Omega_i}$ and vector $\boldsymbol{P}_{\Omega_i,i}$ gets its maximum if the noise energy $\|\boldsymbol{e}\|_2$ does not dominate the sample energy $\|h\|_2$. Thus, still we can find the isolated non-zeros even in the presence of the noise.

Assume that we are given noisy samples $y = \boldsymbol{P}x + e$. Furthermore, suppose that a subset of samples l span only one non-zero coefficient $x_i \neq 0$ ($\|x_{\omega_l}\|_0 = 1$ and $i \in \omega_l$). The first question to be answered is how noise would distort the angle of pure (noiseless) samples $\boldsymbol{P}_{l,\omega_l}x_{\omega_l} = x_i\boldsymbol{P}_{l,i}$ with $\boldsymbol{P}_{l,i}$ compared to the noisy angel between $\boldsymbol{P}_{l,\omega_l}x_{\omega_l} + e_l$ and $\boldsymbol{P}_{l,i}$. Also, we need to find the lower bound L for $|\boldsymbol{P}_{:,i}^T y_l| / \|\boldsymbol{P}_{:,i}\|_2 \|y_l\|_2$ when there is an isolation and also find the upper bound U for the same term when there is no isolation. If $U < \delta < L$ for a constant δ (function of ϵ), then the notion of ϵ-alignment could be applied in case of noisy samples. Providing a rigorous proof is left as a future work. However, in the rest of this section, we highlight some directions and challenges in that path.

A statistical argument could be applied here, where one derives the distribution of the numerator of 6.8. Note that when an isolation is occurred, the numerator inside $\cos^{-1}$ function is

$$\boldsymbol{P}_{l,i}^T y_l = x_i \|\boldsymbol{P}_{l,i}\|_2^2 + \boldsymbol{P}_{l,i}^T e_l \tag{6.9}$$

The distribution of the first term $\|\boldsymbol{P}_{l,i}\|_2^2$ is simply Chi-square. However, the distribution of the second term $\boldsymbol{P}_{l,i}^T e_l$ is Chi. Since two different distributions are added here, finding the distribution of $\boldsymbol{P}_{l,i}^T y_l$ would be difficult. There could be two solutions for such challenge (a) using the method of moment generating functions or (b) starting with a simple assumption that the noise e itself is normally distributed, which makes the second term also Chi-squared. However even in the second case, we have to deal with the problem of weighted-Chi squares. One might think that similar approximations in the previous chapter could be utilized. However due to other approximations which might be applied later, this act should be taken with cautious.

For the denominator, we have to find the distribution of the product of $\|\boldsymbol{P}_{l,i}\|_2 \|y_l\|_2$. Although for each of multiplicands, the distribution could be derived fairly easily, finding the product distribution could be cumbersome. The problem could be even more challenging when we want to calculate the distribution of the random variable generated by division of numerator by denominator. Although a remedy in here could be utilizing w-RIP inequalities.

To find the upper bound L, the distributions of the similar terms when there is no isolation have to be calculated. For instance, when samples l are spanning more than one non-zeros then

$$\boldsymbol{P}_{l,i}^T y_l = \sum_{i \in \omega_l} x_i \|\boldsymbol{P}_{l,i}\|_2^2 + \sum_{i \in \omega_l} \sum_{j \in \omega_l \setminus i} \boldsymbol{P}_{l,i}^T \boldsymbol{P}_{l,j} \tag{6.10}$$

Again here, the first sum has a distribution of weighted Chi-squares. The second sum however is distributed based on sum of normal products. Despite the fact that there exist closed formulas for distribution of normal-product random variables, adding those terms could be challenging. However, again using the method of moment generating functions sounds attractive since it separates such sum into product of two terms where there exists either good approximations or even closed formulas for each term.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Emmanuel Cands and Justin Romberg, “Quantitative robust uncertainty principles and optimally sparse decompositions”. (Foundations of Comput. Math., 6(2), pp. 227 - 254, April 2006)

[2] David Donoho, “Compressed sensing”, IEEE Trans. on Information Theory, 52(4), pp. 1289 - 1306, April 2006

[3] Emmanuel Cands, The restricted isometry property and its implications for compressed sensing. (Compte Rendus de l'Academie des Sciences, Paris, Series I, 346, pp. 589-592, 2008).

[4] Emmanuel Cands and Justin Romberg, “Practical signal recovery from random projection”, (Preprint, Jan. 2005)

[5] Emmanuel Cands, Justin Romberg, and Terence Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. IEEE Trans. on Information Theory, 52(2) pp. 489 - 509, February 2006

[6] D. Donoho and J. Tanner, “Neighborliness of randomly projected simplices in high dimensions”, Mar. 2005, Preprint.

[7] Emmanuel Cands and Terence Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?” IEEE Trans. On Information Theory, 52(12), pp. 5406 - 5425, December 2006

[8] David Donoho and Yaakov Tsaig, “Extensions of compressed sensing”. (Signal Processing, 86(3), pp. 533-548, March 2006)

[9] Emmanuel Cands and Terence Tao, Decoding by linear programming. (IEEE Trans. on Information Theory, 51(12), pp. 4203 - 4215, December 2005)

[10] E. Candes and T. Tao, “Error correction via linear programming”, Found. of Comp. Math., 2005, Submitted.

[11] Jarvis Haupt, Waheed U. Bajwa, Gil Raz, and Robert Nowak, “Toeplitz compressed sensing matrices with applications to sparse channel estimation”. (Preprint, 2008)

[12] Dror Baron, Shriram Sarvoham, and Richard G. Baraniuk, “Bayesian compressive sensing via belief propagation”. (Preprint, 2008)

[13] A.M. Bruckstein, D.L. Donoho, and M. Elad, “From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images”, SIAM Review, Vol. 51, No. 1, Pages 34-81, February 2009.

[14] Dror Baron, Marco F. Duarte, Michael B. Wakin, Shriram Sarvotham, and Richard G. Baraniuk, “Distributed compressive sensing”. (Preprint, 2005)

[15] Shriram Sarvotham, Dror Baron, and Richard Baraniuk, “Measurements vs. bits: Compressed sensing meets information theory”. (Proc. Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 2006)

[16] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin, “A simple proof of the restricted isometry property for random matrices”. Constructive Approximation, 28(3), pp. 253-263, December 2008, [Formerly titled “The Johnson-Lindenstrauss lemma meets compressed sensing”]

[17] S. Chen, , D. Donoho, and M. Saunders. “Atomic decomposition by basis pursuit”. Technical Report 479, Department of Statistics, Stanford University, May 1995.

[18] R. Tibshirani, “Regression shrinkage and selection via the lasso”. J. Royal. Statist. Soc B., Vol. 58, No. 1, pages 267-288, 1996.

[19] Thomas Blumensath and Mike E. Davies, “Gradient pursuits”. (IEEE Trans. on Signal Processing, 56(6), pp. 2370 - 2382, June 2008)

[20] Ingrid Daubechies, Massimo Fornasier, and Ignace Loris, “Accelerated projected gradient method for linear inverse problems with sparsity constraints”. (Preprint, 2007)

[21] T. Blumensath, M. E. Davies, “Stagewise weak gradient pursuits. Part I: Fundamentals and numerical studies”. (Preprint, 2008)

[22] T. Blumensath, M. E. Davies, “Stagewise weak gradient pursuits. Part II: Theoretical properties”. (Preprint, 2008)

[23] Rahul Garg, Rohit Khandekar, “Gradient Descent with Sparsification: An iterative algorithm for sparse recovery with restricted isometry property”. (ICML 2009, Montreal, Canada)

[24] Mrio A. T. Figueiredo, Robert D. Nowak, and Stephen J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems”. (IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing, 1(4), pp. 586-598, 2007).

[25] S. G. Mallat and Z. Zhang, “Matching Pursuits with Time-Frequency Dictionaries”, IEEE Transactions on Signal Processing, December 1993, pp. 3397-3415.

[26] Y. C. Pati R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition”, Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers, Nov. 1993.

[27] Thong T. Do, Lu Gan, Nam Nguyen, and Trac D. Tran, “Sparsity adaptice matching pursuit algorithm for practical compressed sensing”. Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, October 2008

[28] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. Preprint, 2008

[29] Deanna Needell and Roman Vershynin, “Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit”. Preprint, 2007

[30] David L. Donoho, Yaakov Tsaig, Iddo Drori, and Jean-Luc Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit”. Preprint, 2007

[31] Piotr Indyk, “Explicit constructions for compressed sensing of sparse signals”. Symp. on Discrete Algorithms, 2008.

[32] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, “Combining geometry and combinatorics: A unified approach to sparse signal recovery”. (Preprint, 2008)

[33] Radu Berinde and Piotr Indyk, “Sparse recovery using sparse random matrices”. (Preprint, 2008)

[34] Ronald A. DeVore, “Deterministic constructions of compressed sensing matrices”. J. of Complexity, 23, pp. 918 - 925, August 2007

[35] Shriram Sarvotham, Dror Baron, and Richard Baraniuk, “Sudocodes - Fast measurement and reconstruction of sparse signals”. Proc. IEEE Int. Symposium on Information Theory (ISIT), Seattle,Washington, July 2006

[36] Weiyu Xu and Babak Hassibi, “Efficient compressive sensing with deterministic guarantees using expander graphs”. IEEE Information Theory Workshop, Lake Tahoe, September 2007

[37] Sina Jafarpour, Weiyu Xu, Babak Hassibi, and Robert Calderbank, “Efficient compressed sensing using high-quality expander graphs”. (to appear in IEEE Trans Info Theory, 2009)

[38] M. Amin Khajehnejad, Alexandros G. Dimakis, Weiyu Xu, Babak Hassibi, “Sparse Recovery of Positive Signals with Minimal Expansion” . (Preprint, 2009)

[39] Abdolreza Abdolhosseini Moghadam and Haydar Radha, “Hybrid Compressed Sensing”, IEEE MMSP 2010.

[40] Abdolreza Abdolhosseini Moghadam and Haydar Radha, “Complex Sparse Projections for Compressed Sensing”, Accepted for publication in Conference on Information Sciences and Systems (CISS10), Johns Hopkins University, Baltimore, MD, USA.

[41] Abdolreza Abdolhosseini Moghadam and Haydar Radha, “Complex Randomness-in-Structured Projections for Compressed Sensing”, in Proceedings of IEEE International Conference on Image Processing (ICIP09), Egypt.

[42] Abdolreza Abdolhosseini Moghadam and Haydar Radha, “Practical Compressed Sensing with Log-of-Prime Projections”, in Proceedings of Conference on Information Sciences and Systems (CISS09), Johns Hopkins University, Baltimore, MD, USA, March 18-20,2009.

[43] Michael Elad, “Optimized projections for compressed sensing”. IEEE Trans. on Signal Processing, 55(12), pp. 5695-5702, December 2007

[44] Venkat Chandar, “A negative result concerning explicit matrices with the restricted isometry property”. (Preprint, 2008)

[45] http://sparselab.stanford.edu

[46] D. Donoho, “High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension”, Jan. 2005, Preprint.

[47] http://www.mathpages.com/home/kmath199.htm

[48] Martin Raab and Angelika Steger, “”balls into bins” - A simple and tight analysis”, Lecture notes in computer science, vol. 1518, Springer, Berlin, 1998

[49] “Combinatorial Designs: Construction and Analysis” by Douglas R. Stinson, Springer-Verlag, New York inc, 2004

[50] M.Luby, “LT Codes”, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002

[51] Stphane Mallat, “A wavelet tour of signal processing” 2nd Edition, Academic Press, 1999, ISBN 0-12-466606-x

[52] M. N. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” IEEE Trans. on IP, 14(12):2091-2106.

[53] E. Cands and D. Donoho, “Curvelets a surprisingly effective nonadaptive representation for objects with edges.” In: A. Cohen, C. Rabut and L. Schumaker, Editors, Curves and Surface Fitting: Saint-Malo 1999, Vanderbilt University Press, Nashville (2000), pp. 105120.

[54] Clark, George C., Jr.; Cain, J. Bibb (1981). “Error-Correction Coding for Digital Communications”. New York: Plenum Press. ISBN 0-306-40615-2.

[55] Lin, Shu; Costello, Daniel J. Jr. (1983). “Error Control Coding: Fundamentals and Applications”. Englewood Cliffs NJ: Prentice-Hall. ISBN 0-13-283796-X.

[56] http://www.math.msu.edu/ jhall/classes/codenotes/coding-notes.html

[57] R. G. Gallager, “Low Density Parity Check Codes”, Monograph, M.I.T. Press, 1963.

[58] Amin Shokrollahi (2003) LDPC Codes: An Introduction.

[59] “Probability and Computing, Randomized Algorithms and Probabilistic Analysis“, Michael Mitzenmacher and Eli Upfal, Cambridge University Press, 2005.

[60] “Iterative Methods for Solving Linear Systems”, Anne Greenbaum, Society for Industrial and Applied Mathematics, 1997.

[61] David L. Donohoa, A. Maleki and A. Montanari, “Message-passing algorithms for compressed sensing”, preprint, 2009.

[62] L. Bassalygo and M. Pinsker, “Complexity of an Optimum Nonblocking Switching Network without Reconnections”, Problem in Information Transmission, vol 9 no 1, pp. 289-313, 1973.

[63] P. Sarnak,“What is an Expander,” Notices of American Mathematical Society, Volume 51, Number 7.

[64] R. Calderbank, S. Howard, and S. Jafarpour. (2009) “Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property.” [Online]. Available: http://www.dsp.ece.rice.edu/?les/cs/strip-?nal.pdf

[65] Lu Gan, Cong Ling, Thong T. Do, and Trac D. Tran, “Analysis of the statistical restricted isometry property for deterministic sensing matrices using Steins method”. (Preprint, 2009).

[66] David L. Donoho, Arian Maleki, Andrea Montanari, “the noise-sensitivity phase transition in compressed sensing”, IEEE transactions on information theory, Oct. 2011, vol. 57, issue 10, pages 6920-6941.

[67] E. Candes, “A probabilistic and RIPless theory of compressed sensing”, IEEE transactions on information theory, Nov. 2011, Vol. 57, Issue 11, pages 7235-7254.

[68] R. Berinde and P. Indyk, “Sequential Sparse Matching Pursuit”, IEEE Allerton 2009, pages 36-43.

[69] J. Haupt and R. Nowak, “A generalized restricted isometry property”, preprint 2007.

[70] H. Feiveson and F. C. Delaney, “the distribution and properties of a weighted sum of chi squares”, NASA technical notes, TN D-4575, May 1968.

[71] S. Gabler and C. Wolff, “A quick and easy approximation to the distribution of a sum of weighted chi-square variables”, Statistical Papers, vol. 28, pp.317 -325 1987.