RICE UNIVERSITY

Bounds for Optimal Compressed Sensing Matrices and Practical Reconstruction Schemes

by

Shriram Sarvotham

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

________________________
Richard G. Baraniuk, Chair
Professor of Electrical and Computer
Engineering

________________________
Ashutosh Sabharwal
Assistant Professor of Electrical and
Computer Engineering

________________________
Yin Zhang
Professor of Computational and Applied
Mathematics

Houston, Texas
February, 2008
Bounds for Optimal Compressed Sensing Matrices and Practical Reconstruction Schemes

Shriram Sarvotham

Abstract

Compressed Sensing (CS) is an emerging field that enables reconstruction of a sparse signal $x \in \mathbb{R}^n$ that has only $k \ll n$ non-zero coefficients from a small number $m \ll n$ of linear projections. The projections can be thought of as a vector that is obtained by multiplying a $k$-sparse signal in $\mathbb{R}^n$ by a matrix (called CS matrix) of size $m \times n$ where $k < m \ll n$. The central theme of this thesis is to study the role of the CS matrix on robustness in reconstruction as well as the complexity involved in reconstruction schemes.

In the first part of the thesis, we explore the impact of the CS matrix on robustness, as measured by the Restricted Isometry Property (RIP). We derive two converse bounds for RIP of the CS matrix in terms of $n$, $m$ and $k$. For the first bound (structural bound), we employ results from algebra of Singular Value Decomposition (SVD) of sub-matrices. The second bound (packing bound) is based on sphere packing arguments which we motivate by showing the equivalence of the RIP measure and codes on grassmannian spaces. The derivation of the two bounds offer rich geometric
interpretation and illuminate the relationship between CS matrices and many diverse concepts such as equi-angular tight frames, codes on Euclidean spheres, and the generalized Pythagorean Theorem.

In the second part of the thesis, we propose strategies to design the CS matrix so that it lends itself to low-complexity reconstruction schemes. We argue that sparse matrices are a good choice in CS and present two strategies for reconstruction involving group testing and belief propagation respectively.
Contents

Abstract ii

1 Bounds for Optimal Compressed Sensing Matrices 1

1.1 Introduction .............................................. 1

1.2 Notations and Preliminaries ............................... 2

1.3 Restricted Isometry Property ............................... 3

1.3.1 RIP and Singular Values ............................... 3

1.4 Structural Bound for the RIP Ratio ......................... 4

1.4.1 The zeros of $g_k(x)$ and the singular values of $\Phi$ .... 9

1.4.2 Properties of the structural bound ....................... 17

1.4.3 Geometric Interpretation ............................... 23

1.4.4 Structural bound for $k = 2$ ......................... 25

1.5 Packing bound for RIP .................................... 29

1.6 Tightness of RIP bounds .................................. 31

1.7 Relevance of the structural bounds in stochastic RIP .... 36

1.8 Avenues for future work .................................. 40

1.9 Conclusions .............................................. 40
2 Compressed Sensing using Belief Propagation

2.1 Introduction

2.1.1 Information theory

2.1.2 Compressed sensing

2.1.3 Compressed sensing reconstruction algorithms

2.1.4 Connections between channel coding and CS

2.1.5 Lessons from LDPC codes applied to CS

2.1.6 Contributions: Connecting CS decoding to graph decoding algorithms

2.2 CS-LDPC encoding

2.3 CS-LDPC decoding of approximately sparse signals

2.3.1 Signal model: approximately sparse signals

2.3.2 Decoding via statistical inference

2.3.3 Number of measurements

2.3.4 Decoding complexity

2.3.5 Storage requirements

2.3.6 Properties of BP decoding for CS-LDPC

2.3.7 Numerical results

2.4 Variations and enhancements

2.5 Applications

2.6 Conclusions
## 3 Sudocodes: Fast Reconstruction for Strictly Sparse Signals 73

3.1 Introduction .............................................. 73
3.2 Related work ............................................ 76
3.3 Sudocodes ................................................. 77
   3.3.1 Setup .................................................. 77
   3.3.2 Sudo-encoding ........................................ 78
   3.3.3 Sudo-decoding ........................................ 78
3.4 Analysis of Sudo-Decoding .............................. 80
   3.4.1 Choice of $L$ ........................................... 80
   3.4.2 Number of measurements $M$ ......................... 81
   3.4.3 Computational complexity ............................ 82
3.5 Numerical results ........................................ 83
3.6 Discussion ............................................... 84

## 4 Compressed Sensing Rate Distortion 86

4.1 Introduction .............................................. 86
4.2 Set up .................................................... 89
   4.2.1 Notation and preliminaries .......................... 89
   4.2.2 Measurement and reconstruction .................... 90
   4.2.3 Measurement and reconstruction quality ............ 90
4.3 Lower bound on reconstruction error .................... 91
Chapter 1

Bounds for Optimal Compressed Sensing Matrices

1.1 Introduction

In many signal processing applications the focus is often on identifying and estimating a few significant coefficients from a high dimension vector. The wisdom behind this is the ubiquitous compressibility of signals: most of the information contained in a signal resides in a few large coefficients. Traditional sensing and processing first acquires the entire data, only to throw away most of the coefficients and retaining the small number of significant coefficients. Clearly, it is wasteful to sense or compute all of the coefficients when most of it will be discarded at a later stage. This naturally begs the question: can we sense compressible signals in a compressible way? In other words, can we sense only that part of the signal that will not be thrown away? The ground-breaking work in compressed sensing (CS) pioneered by Candés et al. [13] and Donoho [28] answers the above question in the affirmative. They demonstrate that the information contained in the few significant coefficients can be captured (encoded) by a small number of random linear projections. The original signal can then be reconstructed (decoded) from these random projections using an appropriate decoding scheme.

Consider a discrete-time signal $x \in \mathbb{R}^n$ that has only $k \ll n$ non-zero coefficients. CS asserts that it is unnecessary to measure all the $n$ values of $x$; rather, we can recover $x$ from a small number of projections onto an *incoherent* basis [13, 28]. To measure (encode) $x$, we compute the measurement vector $y \in \mathbb{R}^m$ as $m$ linear projections of $x$.
via the matrix-vector multiplication \( y = \Phi x \). Our goal is to reconstruct (decode) \( x \) – either accurately or approximately – given \( y \) and \( \Phi \) using \( m \ll m \) measurements.

The initial discovery has led to a vibrant activity in the area of CS research, opening many intriguing questions both in theoretical and practical aspects of CS. In this chapter, we address the following crucial problem. For a given problem size \( n \), \( m \) and \( k \), we study the properties of the optimal CS measurement matrices. We use the Restricted Isometry Property (RIP) \([13]\) as a metric to measure the quality of a CS measurement matrix. We provide deterministic bounds on the RIP and derive structural properties of the optimal matrices.

1.2 Notations and Preliminaries

In the sequel, \( \Phi \) represents the CS matrix of size \( m \times n \). While we focus mainly on real valued \( \Phi \), we make mention of results that hold for complex valued \( \Phi \) also. For a given positive integer \( k < m \), we denote the submatrices of \( \Phi \) of size \( m \times k \) as \( \Phi_p \), where \( p \in \{1, 2, ..., \binom{n}{k}\} \) is the index that determines the set of columns of \( \Phi \) that are included in \( \Phi_p \).

We denote the set of all \( k \)-sparse signals in \( \mathbb{R}^n \) by \( \Sigma_k \), ie., \( \Sigma_k = \{ x : \|x\|_0 = k \} \).

Define \( \rho_{\max}(\Phi, k) \) and \( \rho_{\min}(\Phi, k) \) as

\[
\rho_{\max}(\Phi, k) = \max_{x \in \Sigma_k, x \neq 0} \frac{\|\Phi x\|_2^2}{\|x\|_2^2}, \tag{1.1}
\]

and

\[
\rho_{\min}(\Phi, k) = \min_{x \in \Sigma_k, x \neq 0} \frac{\|\Phi x\|_2^2}{\|x\|_2^2}. \tag{1.2}
\]

We define the RIP ratio \( R \) for a given CS matrix \( \Phi \) as

\[
R(\Phi, k) = \frac{\rho_{\max}(\Phi, k)}{\rho_{\min}(\Phi, k)}. \tag{1.3}
\]
Finally, we denote the differential operator $\frac{d}{dx}$ by $D$, ie., $D[f(x)] = \frac{d}{dx} f(x)$, $D^2[f(x)] = \frac{d^2}{dx^2} f(x)$ and so on.

1.3 Restricted Isometry Property

The Restricted Isometry Property (RIP) was introduced by Candes and Tao [13] as a metric to measure the quality of the CS matrix $\Phi$ in terms of robustness of reconstruction to noise in the measurements. We have by definition

$$\rho_{\text{min}}(\Phi, k) \|x\|_2^2 \leq \|\Phi x\|_2^2 \leq \rho_{\text{max}}(\Phi, k) \|x\|_2^2$$

(1.4)

for all $x \in \Sigma_k$. Restricted Isometry demands that the ratio $R(\Phi, k) = \rho_{\text{max}}/\rho_{\text{min}}$ is close to unity.\(^1\) The RIP condition requires that every submatrix $\Phi_p$ has all its singular values in the range $[\sqrt{\rho_{\text{max}}}, \sqrt{\rho_{\text{min}}}]$. The metric $R(\Phi, k)$ indicates the quality of the matrix $\Phi$ in terms of robustness and resolvability in Compressed Sensing: “good” CS matrices have $R$ close to unity.

1.3.1 RIP and Singular Values

We exploit the fact that the measurement $y$ depends only on $k$ out of the $m$ columns of $\Phi$ when $x$ is $k$-sparse. This observation motivates us to consider the singular values of summatrices $\Phi_p \in \mathbb{R}^{m \times k}$, formed by selecting only $k$ of the $n$ columns of $\Phi$. The index $p \in \{1, 2, ..., \binom{n}{k}\}$ identifies the set of columns chosen from the parent matrix $\Phi$. Let the singular values of the matrix $\Phi$ be $S_1, S_2, ..., S_m$, where $S_1 \geq S_2 \geq ... \geq S_m \geq 0$.\(^2\) Likewise we let the singular values of $\Phi_p$ be $s_{p,1}, s_{p,2}, ..., s_{p,k}$, where

\(^1\)In [13] the RIP is expressed in terms of the constant $\delta_K$ where $R(\Phi, k) = \frac{1+\delta_K}{1-\delta_K}$.

\(^2\)We restrict our attention only to the largest $m$ singular values of $\Phi$ because the remaining $n - m$ singular values are zero.
$s_{p,1} \geq s_{p,2} \geq \ldots \geq s_{p,k} \geq 0$. Note that from our definition of $\rho_{\text{max}}$ and $\rho_{\text{min}}$ in (1.4), we have $\rho_{\text{max}}(\Phi, k) = (\max_p\{s_{p,1}\})^2$ and $\rho_{\text{min}}(\Phi, k) = (\min_p\{s_{p,1}\})^2$.

The key insight we use to derive bounds on $\rho_{\text{max}}$ and $\rho_{\text{min}}$ is that the singular values $\{s_{p,i}\}$ are severely constrained by the singular values $\{S_j\}$ of the parent matrix. For example, interlacing inequality [80] requires

$$s_{p,i} \leq S_i \text{ for all } i = 1, 2, \ldots, k.$$  

We explore such intimate relationships between the two sets of singular values to derive bounds on the RIP.

### 1.4 Structural Bound for the RIP Ratio

In this section, we present a deterministic lower bound for $R(\Phi, k)$ in terms of $n$, $m$ and $k$ based on the properties of singular values of submatrices. The main result in this section is stated below in Theorem 1.

**Theorem 1** Let $\Phi \in \mathbb{K}^{m \times n}$ be a matrix over $\mathbb{K}$ where $\mathbb{K}$ is the real number field or the complex number field, and $0 < m < n$. Let $0 < k < m$ and define the $k$’th degree polynomial

$$f_k(x) \triangleq D^{n-k} \left[ x^{n-m}(x-1)^m \right]. \quad (1.5)$$

The following results are true.

1. The $k$ zeros of $f_k(x)$ are real, and lie in the interval $(0, 1]$.

2. Let $r_1^2 \geq r_2^2 \geq \ldots r_k^2$ be the zeros of $f_k(x)$. Then, the following lower bound on the RIP ratio holds

$$R(\Phi, k) \geq \frac{r_1^2}{r_K^2}. \quad (1.6)$$
3. Let $S_1 \geq S_2 \geq ... \geq S_m > 0$ be the $m$ singular values of $\Phi$. Equality in (1.6) is achieved if and only if the following three conditions are satisfied:

(a) $S_1 = S_2 = ... = S_m$, and

(b) The largest singular values of every $m \times k$ submatrix of $\Phi$ are all equal.

(c) The smallest singular values of every $m \times k$ submatrix of $\Phi$ are all equal.

The remainder of this Section, we establish Theorem 1 by presenting a series of results. We also uncover the properties of the proposed lower bound on the RIP ratio. The following result serves as the first step to prove Theorem 1. We propose a lower bound on the RIP ratio when the singular values of $\Phi$ are known.

**Theorem 2** Let $\Phi \in \mathbb{K}^{m \times n}$ be a matrix over $\mathbb{K}$ where $\mathbb{K}$ is the real number field or the complex number field, and $0 < m < n$. Let $S_1 \geq S_2 \geq ... \geq S_m$ be the $m$ singular values of $\Phi$. Likewise we let the singular values of the submatrix $\Phi_p$ be $s_{p,1} \geq s_{p,2} \geq ... \geq s_{p,K}$, for all $p \in \{1, 2, ..., \binom{n}{k}\}$. Let

$$g_k(x) = D^{n-k} [x^{n-m}(x - S_1^2)(x - S_2^2)...(x - S_m^2)].$$  \hfill (1.7)

Then,

1. The $k$ zeros of $g_k(x)$ are real and lie in the interval $(0, S_1^2]$.

2. Let $r_1^2 \geq r_2^2 \geq ... r_k^2$ be the zeros of $g_k(x)$. Then,

$$\rho_{\text{max}}(\Phi, k) \geq r_1^2, \quad \rho_{\text{min}}(\Phi, k) \leq r_k^2, \quad \text{and} \quad R(\Phi, k) \geq \frac{r_1^2}{r_k^2}.$$  \hfill (1.8)
3. Equality in all three inequalities of (1.8) is attained if and only if the following two conditions are satisfied.

(a) The largest singular values of every $m \times k$ submatrix of $\Phi$ are all equal.

(b) The smallest singular values of every $m \times k$ submatrix of $\Phi$ are all equal.

The proof of the above result hinges on a Theorem of Robert C. Thompson [80] that relates the singular values of all the submatrices of a given size to the singular values of the parent matrix. We present the statement of Thompson’s Theorem here for completeness; for proof of the Theorem and other powerful results on singular values and eigenvalues of submatrices, we refer the reader to the series of papers by R.C. Thompson [74, 82, 75, 78, 76, 79, 77, 81, 80].

**Theorem 3** [80, Theorem 4, page 11] Let $A$ be an $m \times n$ matrix with singular values $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{\min\{m,n\}}$. Let $B_p$ be an $l \times k$ submatrix of $A$ for some fixed $l \in \{1, 2, \ldots, m\}$ and $k \in \{1, 2, \ldots, n\}$, with singular values $\beta_{p,1} \geq \beta_{p,2} \geq \ldots \geq \beta_{p,\min\{l,k\}}$, and index $p$ that identifies the submatrix amongst all $l \times k$ submatrices of $A$. Set

$$f_p(x) = (x - \beta_{p,1}^2)(x - \beta_{p,2}^2)\ldots(x - \beta_{p,\min\{l,k\}}^2), \quad (1.9)$$

$$f(x) = (x - \alpha_1^2)(x - \alpha_2^2)\ldots(x - \alpha_{\min\{m,n\}}^2). \quad (1.10)$$

Then,

$$\sum_p x^{l-\min\{l,k\}} f_p(x) = \frac{1}{(m-l)!} \frac{1}{(n-k)!} D^{m-l} [x^{m-k} D^{n-k} [x^{n-\min\{m,n\}} f(x)]] , \quad (1.11)$$

where the summation is taken over all the submatrices of $A$ of size $l \times k$. 
Proof of Theorem 2: The zeros of \( g_k(x) \) are real and lie in the closed interval \([0, S^2_1]\) as a consequence of the Gauss-Lucas Theorem \([62]\). Recall the assertion of Gauss-Lucas Theorem that every convex set in the complex plane containing all the zeros of a polynomial also contains all its critical points (the zeros of the derivative of the chosen polynomial). In our setting, we consider the polynomial \( G(x) = x^{n-m}(x - S^2_1)(x - S^2_2)...(x - S^2_m) \), which has all its zeros in the closed interval \([0, S^2_1]\) on the real number line. Differentiating the polynomial \( G(x) \) \((n-k)\) times and applying the Gauss-Lucas Theorem at each step, we infer that the \( k \) zeros of \( g_k(x) \) are real and lie in the interval \([0, S^2_1]\). Finally, we observe that \( x = 0 \) cannot be a zero of the polynomial \( g_k(x) \) because \( x = 0 \) is a zero of \( G(x) \) of order \((n-m)\), whereas we differentiate \((n-k) > (n-m)\) times. Thus we have established statement (1) of Theorem 2.

To prove statement (2), we apply Theorem 3 (Thompson’s Theorem) to our setting where we consider the parent matrix \( \Phi \in \mathbb{K}^{m \times n} \) and the collection of submatrices \( \Phi_p \in \mathbb{K}^{m \times k} \) for \( p = 1, 2, ..., \binom{n}{k} \). We get

\[
\frac{1}{(n-k)!} D^{n-k} \left[ x^{n-m}(x - S^2_1)(x - S^2_2)...(x - S^2_m) \right] = \sum_p (x - s^2_{p,1})(x - s^2_{p,2})...(x - s^2_{p,k}).
\]

Equation (1.12)

The above equation relates the singular value polynomial of \( \Phi \) to the singular value polynomials of the submatrices \( \Phi_p \). Recall that the singular value polynomial of a matrix \( A \) is the polynomial whose roots are the squares of the singular values of the matrix.\(^4\) Equation (1.12) asserts that the sum of the singular value polynomials of the submatrices \( \Phi_p \) is a constant multiple of the \((n-k)\)th derivative of the singular value polynomial of \( \Phi \).

\(^3\)Additionally, note that when \( n-k > m \), \( x = S^2_1 \) cannot be a zero of \( g_k(x) \).

\(^4\)Equivalently, the singular value polynomial of \( A \) is the characteristic polynomial of the Grammian \( A^H A \) or \( AA^H \), whichever has the higher degree.
Since \( r_1^2, r_2^2, \ldots, r_k^2 \) are the roots of \( g_k(x) \), we have

\[
\frac{n!}{k!}(x - r_1^2)(x - r_2^2)\ldots(x - r_k^2) = D^{n-k} \left[ x^{n-m} (x - S_1^2)(x - S_2^2)\ldots(x - S_m^2) \right],
\]  

where the constant \( n!/k! \) is needed to equalize the coefficients of \( x^k \) in LHS and RHS of the above equation. From (1.12) and (1.13),

\[
\binom{n}{k} (x - r_1^2)(x - r_2^2)\ldots(x - r_k^2) = \sum_p (x - s_{p,1}^2)(x - s_{p,2}^2)\ldots(x - s_{p,k}^2).
\]  

We are now in a position to prove statement (2) of Theorem 2. First, we note that \( \rho_{\text{max}}(\Phi, k) = \max_p \{s_{p,1}^2\} \) and \( \rho_{\text{min}}(\Phi, k) = \min_p \{s_{p,1}^2\} \). The first inequality in statement (2) implies that \( \max_p \{s_{p,1}^2\} \geq r_1^2 \). For the sake of a contradiction, assume that \( s_{p,1}^2 < r_1^2 \) for all \( p \in \{1, 2, \ldots, \binom{n}{k}\} \). Under this assumption, \( (r_1^2 - s_{p,i}^2) > 0 \) for all \( p \) and \( i \) and hence each of the polynomial \( (x - s_{p,1}^2)(x - s_{p,2}^2)\ldots(x - s_{p,k}^2) \) is strictly positive when evaluated at \( x = r_1^2 \). Consequently, the sum in the RHS of Equation (1.14) evaluated at \( x = r_1^2 \) is strictly positive, which is a contradiction because the LHS of (1.14) evaluates to zero for \( x = r_1^2 \). Therefore, we require \( r_1^2 \leq \max \{s_{p,1}^2\} = \rho_{\text{max}}(\Phi, k) \).

We can show that \( r_k^2 \geq \max \{s_{p,k}^2\} = \rho_{\text{min}}(\Phi, k) \) using a similar argument as above. Note that we need to consider the additional nuance of the sign of \( (x - s_{p,1}^2)(x - s_{p,2}^2)\ldots(x - s_{p,k}^2) \) when evaluating at \( x = r_k^2 \) with the assumption \( s_{p,k}^2 > r_k^2 \): the sign is either strictly positive or strictly negative depending on whether \( k \) is even or odd, respectively. Finally, since \( \rho_{\text{max}}(\Phi, k) \geq r_1^2 \) and \( \rho_{\text{min}}(\Phi, k) \leq r_k^2 \), we have \( R(\Phi, k) \geq \frac{r_1^2}{r_k^2} \), and hence statement (2) is established.

We now turn to prove statement (3). When \( s_{p,1}^2 \) are equal for all \( p \in \{1, 2, \ldots, \binom{n}{k}\} \), we observe that with \( x = s_{p,1}^2 \), the RHS of Equation (1.14) evaluates to zero. Hence \( s_{p,1}^2 \) must equal \( r_i^2 \) for some \( i \in \{1, 2, \ldots, k\} \). Since we have established in statement (2) that we cannot have a zero of \( (x - r_1^2)(x - r_2^2)\ldots(x - r_k^2) \) greater than \( s_{p,1}^2 \), it follows...
that \( r_i^2 = s_{p,1}^2 \) for all \( p \). Similarly, when \( s_{p,k}^2 \) are equal for all possible \( p \), we have \( r_k^2 = s_{p,k}^2 \).

Conversely, suppose \( r_i^2 = \max \{ s_{p,1}^2 \} \) is true. For the sake of a contradiction assume that the values of \( s_{p,1}^2, p \in \{1, 2, \ldots, (n/k)\} \) are not all equal. In particular, pick \( p_1 \) such that \( s_{p_1,1}^2 < \max \{ s_{p,1}^2 \} \). Then the polynomial corresponding to \( p = p_1 \) within the summation in RHS in (1.14) is strictly positive when evaluated at \( r_i^2 \). Consequently, \( r_i^2 \) cannot be a zero of (1.14), yielding a contradiction. A similar argument applies to the case when \( r_k^2 = \min \{ s_{p,k}^2 \} \). Thus we have established statement (3) and the proof of Theorem 2 is complete.

While Theorem 2 applies to the case where the singular values of the CS matrix \( \Phi \) are known, Theorem 1 applies to all matrices \( \Phi \) of size \( m \times n \). In our quest for deterministic bounds for the RIP ratio, Theorem 1 therefore is of central importance. To establish Theorem 1, we explore the following question: what choice of \( S_1^2, S_2^2, \ldots, S_m^2 \) gives the most conservative bound for the RIP ratio when we invoke Theorem 2? We show in the sequel (Theorem 7) that the ratio \( \frac{r_i^2}{r_k^2} \) is minimized when all the singular values of \( \Phi \) are equal. Therefore this minimum ratio serves as a universal bound for RIP applicable to all CS matrices \( \Phi \) of size \( m \times n \) and leads to the proof of Theorem 1.\(^5\) Toward this goal, we investigate the nature of the dependence of \( r_i^2 \) on \( S_1^2, S_2^2, \ldots, S_m^2 \) in Section 1.4.1.

1.4.1 The zeros of \( g_k(x) \) and the singular values of \( \Phi \)

In this section, we treat each \( r_i^2, i \in \{1, 2, \ldots, k\} \), as a function of \( m \) variables \( S_1^2, S_2^2, \ldots, S_m^2 \) with the goal to determine the optimal choice of \( S_j \)'s that minimize

\(^5\)Of course, Theorem 2 provides a tighter bound for the RIP ratio when the singular values of \( \Phi \) are known.
the ratio $\frac{r_2}{r_k}$. First, we present how an infinitesimal increment in one of the $S_2^j$’s affects the values of $r_2^i$’s.

**Theorem 4** Let $G(x) = x^{n-m}(x - S_1^2)(x - S_2^2)\ldots(x - S_m^2)$ and let $r_2^1 \geq r_2^2 \geq \ldots \geq r_2^k$ be the zeros of the polynomial $g_k(x) = D^k[G(x)]$. Then,

$$
\frac{\partial (r_2^i)}{\partial (S_2^j)} = \left[ \frac{D^{n-k}\left( \frac{G(x)}{x-S_2^j} \right)}{D^{n-k+1}G(x)} \right] \text{ evaluated at } x = r_2^i \quad (1.15)
$$

$$
= \left[ x \frac{D^{n-k}\left( \frac{G(x)}{x-S_2^j} \right)}{\sum_{l=1}^{m} S_2^l D^{n-k}\left( \frac{G(x)}{x-S_2^l} \right)} \right] \text{ evaluated at } x = r_2^i \quad (1.16)
$$

**Proof:** Taking the partial derivative of the Equation (1.13) with respect to the variable $S_1^2$ keeping $S_2^2, S_3^2, \ldots, S_m^2$ fixed, we have

$$
\frac{n!}{k!} \frac{\partial}{\partial (S_1^2)} (x - r_2^1)(x - r_2^2)\ldots(x - r_2^k) = \frac{\partial}{\partial (S_1^2)} D^{n-k} \left[ x^{n-m}(x - S_1^2)(x - S_2^2)\ldots(x - S_m^2) \right] \quad (1.17)
$$

Treating the quantity inside the $D^{n-k}[\cdot]$ as a product of $(x - S_1^2)$ and $x^{n-m}(x - S_2^2)(x - S_3^2)\ldots(x - S_m^2)$ and using Leibnitz’s Theorem [4] for the $(n-k)$’th derivative of a product, we get

$$
D^{n-k} \left[ x^{n-m}(x - S_1^2)(x - S_2^2)\ldots(x - S_m^2) \right]
$$

$$
= (x - S_1^2)D^{n-k-1} \left[ x^{n-m}(x - S_2^2)(x - S_3^2)\ldots(x - S_m^2) \right] + (n-k)D^{n-k-1} \left[ x^{n-m}(x - S_2^2)(x - S_3^2)\ldots(x - S_m^2) \right]. \quad (1.18)
$$

Taking partial derivative with respect to $S_1^2$ (and noting that the second term of RHS in (1.18) is independent of $S_1^2$), we have

$$
\frac{\partial}{\partial (S_1^2)} D^{n-k} \left[ x^{n-m}(x - S_1^2)(x - S_2^2)\ldots(x - S_m^2) \right]
$$
\[= -D^{n-k} \left[ x^{n-m}(x - S_2^2)(x - S_3^2)\ldots(x - S_m^2) \right] \]
\[= -D^{n-k} \left[ \frac{G(x)}{(x - S_1^2)} \right]. \quad (1.19)\]

Also,
\[
\frac{n!}{k!} \frac{\partial}{\partial (S_1^2)} \left( x - r_1^2 \right)(x - r_2^2)\ldots(x - r_k^2)
= -\frac{n!}{k!} \sum_{i=1}^{k} \left\{ \prod_{1 \leq l \leq k, l \neq i} (x - r_l^2) \frac{\partial r_i^2}{\partial (S_1^2)} \right\}. \quad (1.20)
\]

Evaluating (1.20) at \(x = r_1^2\) and noting that only one term in the summation in the RHS of (1.20) is non-zero, we get
\[
\frac{n!}{k!} \frac{\partial}{\partial (S_1^2)} \left( x - r_1^2 \right)(x - r_2^2)\ldots(x - r_k^2) \bigg|_{x=r_1^2}
= -\frac{n!}{k!} \sum_{i=1}^{k} \left\{ \prod_{1 \leq l \leq k, l \neq i} (x - r_l^2) \frac{\partial r_i^2}{\partial (S_1^2)} \right\} \bigg|_{x=r_1^2}
= -\frac{n!}{k!} \left[ \prod_{1 \leq l \leq k} (x - r_l^2) \frac{\partial r_i^2}{\partial (S_1^2)} \right] \bigg|_{x=r_1^2}
= -\frac{D^{n-k} \left[ G(x) \right]}{x - r_1^2} \frac{\partial (r_1^2)}{\partial (S_1^2)} \bigg|_{x=r_1^2}. \quad (1.21)
\]

From (1.17), (1.19), and (1.21), we infer that
\[
\frac{\partial (r_1^2)}{\partial (S_1^2)} = \left[ \frac{D^{n-k} \left( \frac{G(x)}{x - S_1^2} \right)}{D^{n-k+1}G(x)} \right] \text{ evaluated at } x = r_1^2. \quad (1.22)
\]

Finally, we use the following well known result that if a function \(f(x)\) has a zero at \(x = a\) of multiplicity 1 (i.e., a simple zero), then \(f(x)/(x - a)\) evaluated at \(x = a\) is equal to \(D \left[ f(x) \right]\) evaluated at \(x = a\). Applying this result, we have \(D^{n-k}G(x)/(x - r_1^2)\) evaluated at \(x = r_1^2\) is equal to \(D^{n-k+1}G(x)\) evaluated at \(x = r_1^2\), because \(r_1^2\) is a zero of \(D^{n-k}G(x)\). Using this in (1.22), we deduce that
\[
\frac{\partial (r_1^2)}{\partial (S_1^2)} = \left[ \frac{D^{n-k} \left( \frac{G(x)}{x - S_1^2} \right)}{D^{n-k+1}G(x)} \right] \text{ evaluated at } x = r_1^2, \quad (1.23)
\]
which proves the first part of Theorem 4. To prove the second part, denote $S(x) = (x - S_1^2)(x - S_2^2)...(x - S_m^2)$, and consider the quantity $D^{n-k+1}[xG(x)]$. We have

\[
D^{n-k+1}[xG(x)] = D^{n-k} \left[ D \{x^{n-m+1}S(x)\} \right]
= D^{n-k} \left[ (n - m + 1)x^{n-m}S(x) + x^{n-m+1}D[S(x)] \right]
= (n - m + 1)D^{n-k} \left[ x^{n-m}S(x) \right] + D^{n-k} \left[ xD[S(x)] \right].
\] (1.24)

Alternately, we can use Leibnitz Theorem to expand $D^{n-k+1}[xG(x)]$, yielding

\[
D^{n-k+1}[xG(x)] = (n - k + 1)D^{n-k} \left[ G(x) \right] +xD^{n-k+1} \left[ x^{n-m}S(x) \right].
\] (1.25)

From Equations (1.24) and (1.25), we have

\[
(n - m + 1)D^{n-k} \left[ x^{n-m}S(x) \right] + D^{n-k} \left[ xD[S(x)] \right]
= (n - k + 1)D^{n-k} \left[ G(x) \right] +xD^{n-k+1} \left[ x^{n-m}S(x) \right],
\] (1.26)

and therefore

\[
D^{n-k} \left[ x^{n-m+1}D[S(x)] \right] = (m - k)D^{n-k} \left[ G(x) \right] +xD^{n-k+1} \left[ G(x) \right].
\] (1.27)

Therefore,

\[
D^{n-k+1} \left[ x^{n-m}S(x) \right] = \frac{1}{x} \left\{ D^{n-k} \left[ x^{n-m+1}D[S(x)] \right] - (m - k)D^{n-k} \left[ G(x) \right] \right\}
= \frac{1}{x} \left\{ D^{n-k} \left[ x^{n-m+1}D[S(x)] - mG(x) \right] + kD^{n-k} \left[ G(x) \right] \right\}.
\] (1.28)

Since $D[S(x)] = S(x) \left\{ \frac{1}{x - S_1^2} + ... + \frac{1}{x - S_m^2} \right\}$, we have

\[
x^{n-m+1}D[S(x)] = x^{n-m}S(x) \left\{ \frac{x}{x - S_1^2} + ... + \frac{x}{x - S_m^2} \right\}.
\] (1.29)

Rewriting $mD^{n-k}G(x)$ as

\[
mD^{n-k}G(x) = D^{n-k} \left[ x^{n-m}S(x) \left\{ \frac{x - S_1^2}{x - S_1^2} + ... + \frac{x - S_m^2}{x - S_m^2} \right\} \right].
\] (1.30)
and subtracting (1.30) from (1.29), we get
\[
D^{n-k} \left[ x^{n-m+1} D[S(x)] \right] - mD^{n-k} G(x) = D^{n-k} \left[ x^{n-m} S(x) \left\{ \frac{S_1^2}{x - S_1^2} + \cdots + \frac{S_m^2}{x - S_m^2} \right\} \right] (1.31)
\]

Plugging the above result in (1.28), we get
\[
D^{n-k+1} \left[ x^{n-m} S(x) \right] = \frac{1}{x} k D^{n-k} [G(x)] + \frac{1}{x} \left\{ \sum_{i=1}^{m} S_i^2 D^{n-k} \left[ \frac{G(x)}{x - S_i^2} \right] \right\}. (1.32)
\]

Substituting (1.32) in (1.23) and evaluating at \( x = r_i^2 \) (noting that \( D^{n-k} [G(x)] \) is zero at \( x = r_i^2 \)), we get the result in (1.16) which completes the proof of Theorem 4.

\[ \square \]

**Remark 1** From (1.16), we have
\[
S_1 \frac{\partial (r_i^2)}{\partial (S_1^2)} + S_2 \frac{\partial (r_i^2)}{\partial (S_2^2)} + \cdots + S_m \frac{\partial (r_i^2)}{\partial (S_m^2)} = r_i^2, \tag{1.33}
\]
which we recognize as Euler’s condition for homogeneity of \( r_i^2 \) on \( \{S_1^2, S_2^2, \ldots, S_m^2\} \) of degree 1 [4].

Remark 1 implies that when we write \( r_i^2 \) as a function of \( \{S_1^2, S_2^2, \ldots, S_m^2\} \), we have
\[
r_i^2(aS_1^2, aS_2^2, \ldots, aS_m^2) = ar_i^2(S_1^2, S_2^2, \ldots, S_m^2), \tag{1.34}
\]
for all \( a \in \mathbb{R}^+ \). This result is in agreement with (1.13), because the homogeneity result can be derived from (1.13) by making a change of variables from \( x \) to \( ax \).

**Theorem 5** Under the assumptions of Theorem 4,
\[
\frac{\partial (r_i^2)}{\partial (S_j^2)} \geq 0. \tag{1.35}
\]
Theorem 5 is significant because it indicates that increasing $S_j^2$ for any $j$ can only increase the corresponding $r_i^2$‘s. This fact can be exploited if our objective is to maximize the singular values of the submatrices of $\Phi$. In Donoho’s pioneering paper on Compressed Sensing, the condition CS1 requires the smallest singular values of the submatrices of $\Phi$ to exceed a positive constant [28]. The above Theorem indicates the relationship of CS1 condition to the singular values of the parent matrix $\Phi$. Also, in any practical system, we have a bound on the maximum $S_j^2$‘s that can be used, reminiscent of coding with power constraints [61]. In such a scenario, Theorem 5 indicates that the best choice of $S_j^2$‘s is when they are all equal to the maximum allowable bound. In order to prove Theorem 5, we require a result on interlacing polynomials.

**Definition 1** Two non-constant polynomials $p(x)$ and $q(x)$ with real coefficients have weakly interlacing zeros if

- their degree are equal or differ by one,
- their zeros are all real, and
- there exists an ordering such that

$$\alpha_1 \leq \beta_1 \leq \alpha_2 \leq \beta_2 \leq \ldots \leq \alpha_\nu \leq \beta_\nu \leq \ldots, \quad (1.36)$$

where $\alpha_1, \alpha_2, \ldots$ are the zeros of one polynomial and $\beta_1, \beta_2, \ldots$ are the zeros of the other.

If, in the ordering of (1.36), no equality sign occurs, then $p(x)$ and $q(x)$ have strictly interlacing zeros.

**Theorem 6** *(Hermite-Kakeya)* Let $p(x)$ and $q(x)$ be non-constant polynomials in $x$ with real coefficients. Then, $p(x)$ and $q(x)$ have strictly
interlacing zeros if and only if, for all \( \mu, \lambda \in \mathbb{R} \) such that \( \lambda^2 + \mu^2 > 0 \), the polynomial \( g(x) = \lambda p(x) + \mu q(x) \) has simple, real zeros.

**Proof of Theorem 5:** Consider Equation (1.15). We demonstrate in this proof that the numerator and denominator polynomials in the RHS of the said equation have the same sign when evaluated at \( x = r_i^2 \). Consider the three polynomials

\[
p_1(x) = D^{n-k+1}[G(x)], \quad p_2(x) = D^{n-k}\left\{G(x)/(x - S_j^2)\right\} \quad \text{and} \quad q(x) = D^{n-k}[G(x)].
\]

Note that \( p_1(x) \) and \( p_2(x) \) are of degree \((k - 1)\), while \( q(x) \) is of degree \( k \). We assert that

1. \( p_1(x) \) and \( q(x) \) have strictly interlacing zeros, and
2. \( p_2(x) \) and \( q(x) \) have strictly interlacing zeros.

Statement (1) above is a straightforward consequence of the interlacing property of a polynomial with real zeros and its derivative \[62\]; we note that \( p_1(x) = D[q(x)]. \)

Statement (2) follows from a direct application of Hermite-Kakeya’s Theorem to \( p_2(x) \) and \( q(x) \). Consequently, for a given zero of \( q(x) \), say \( r_i^2 \), there are equal number of zeros of \( p_1(x) \) and \( p_2(x) \) to the left of \( r_i^2 \) on the real number line. Similarly, there are equal number of zeros of \( p_1(x) \) and \( p_2(x) \) to the right of \( r_i^2 \) on the real number line. Lastly, note that the leading coefficients of all three polynomials (ie., coefficient of \( x^k \) for \( q(x) \) and the coefficients of \( x^{k-1} \) for \( p_1(x) \) and \( p_2(x) \)) are positive. Therefore, \( p_1(x) \) and \( p_2(x) \) have the same sign when evaluated at \( x = r_i^2 \), and hence the RHS of (1.15) is positive, completing the proof. \( \square \)

Finally, we present the main Theorem in this Section which shows that the choice of \( S_j \)'s that minimize the ratio \( r_1^2/r_k^2 \) is when the \( S_j^2 \)'s are all equal.

**Theorem 7** Under the assumptions of Theorem 4, the ratio \( r_1^2/r_k^2 \) is minimized for \( S_1^2 = S_2^2 = \ldots = S_m^2 \).
Proof: In order to optimize \( r_2^2/r_k^2 \), we consider the partial derivatives of the said ratio with respect to the \( S_j^2 \)'s. At the optimal location, we require the partial derivatives to be zero, ie.,

\[
\frac{\partial}{\partial (S_j^2)} \left\{ \frac{r_1^2}{r_k^2} \right\} = 0,
\]

\[
\Rightarrow \frac{1}{r_k^2} \left\{ r_k^2 \frac{\partial (r_1^2)}{\partial (S_j^2)} - r_1^2 \frac{\partial (r_k^2)}{\partial (S_j^2)} \right\} = 0,
\]

\[
\Rightarrow \frac{r_1^2}{r_k^2} \left\{ \frac{1}{r_k^2} \frac{\partial (r_1^2)}{\partial (S_j^2)} - \frac{1}{r_k^2} \frac{\partial (r_k^2)}{\partial (S_j^2)} \right\} = 0,
\]

\[
\Rightarrow \frac{1}{r_k^2} \frac{\partial (r_1^2)}{\partial (S_j^2)} = \frac{1}{r_k^2} \frac{\partial (r_k^2)}{\partial (S_j^2)}.
\]

(1.37)

for all \( S_j \)'s. We show that the above condition is satisfied with the choice \( S_1^2 = S_2^2 = ... = S_m^2 \). Assume that the \( S_j^2 \)'s are all equal and non-zero, and denote their common value by \( S^2 \). Because of symmetry, the quantities \( \frac{\partial (r_1^2)}{\partial (S_j^2)} \) are independent of the choice of \( j \), and we denote the said quantity by \( \frac{\partial (r_1^2)}{\partial (S^2)} \). Consequently, the terms in the summation of LHS in the Euler homogeneity equation (1.33) are all equal, and hence

\[
\frac{\partial (r_1^2)}{\partial (S^2)} = \frac{r_1^2}{mS^2}.
\]

(1.38)

Therefore, \( \frac{1}{r_1^2} \frac{\partial (r_1^2)}{\partial (S_j^2)} \) is independent of the index \( i \) and hence the condition (1.37) is satisfied. It remains to show that the optimal point just derived is a minima. A rigorous analysis to prove minimality involves the computation of the Hessian matrix [4] of \( r_1^2/r_k^2 \) with respect to the \( S_j^2 \)'s. However, the analysis quickly turns intractable. Instead, we fix \( S_1^2 = 1 \) and note that the only choice of \( S_j^2 \)'s for \( j = 2, 3, ..., k \) that satisfy the condition (1.37) is when they are all equal to unity. Thus, we can check the maxima or minima criteria by comparing the value of \( \frac{r_1^2}{r_k^2} \) at \( S_j^2 = 1 \) to another set choice of \( S_j^2 \)'s. We pick the set \( S_1^2 = 1, S_2^2 = 0, S_3^2 = 0, ..., S_m^2 = 0 \) for the purpose of comparison, and we immediately see that \( \frac{r_1^2}{r_k^2} = \infty \) for this choice because \( r_k^2 = 0 \). Therefore, the optimal point we have determined is a minima. \( \square \)
We remark that the above proof reveals the fact that the ratio \( r_{i_1}^2 / r_{i_2}^2 \) of any two roots \( r_{i_1}^2 \) and \( r_{i_2}^2 \) is minimized for a given \( i_1 \) and \( i_2 \) such that \( i_1 < i_2 \).

Finally, we note that Theorem 2 together with Theorem 7 proves Theorem 1 which is the main result of this section.

1.4.2 Properties of the structural bound

In this section, we study the relationships between the bound given by Theorem 1 and the parameters \( n, m \) and \( k \) of Compressed Sensing. Clearly, the properties of the structural bound are tied to the properties of the polynomial \( f_k(x) \) as defined in Theorem 1. We begin by first expressing \( f_k(x) \) in the standard polynomial form, by carrying out the \((n-k)\)’th order differentiation in (1.5). We get

\[
f_k(x) = \sum_{j=0}^{k} (-1)^{k-j} \binom{m}{k-j} \binom{n + j - k}{j} x^j
\]

and \( S_i \)'s different

\[
g_k(x) = \sum_{j=0}^{k} (-1)^{k-j} \left( \sum S_1 S_2 \ldots S_{k-j} \right) \frac{(n + j - k)!}{j!} x^j.
\]

We ask if the form of \( f_k(x) \) is similar to one of classic polynomials in the literature [62]. The ratio of successive terms of the polynomial suggests that \( f_k(x) \) is a Gauss Hypergeometric function of the second kind.

\[
\frac{c_j}{c_{j+1}} = \frac{(-1)^{k-j-1} \binom{m}{k-j} \binom{n+j-k+1}{j+1} x^{j+1}}{(-1)^{k-j} \binom{m}{k-j} \binom{n+j-k}{j} x^j} = \frac{(j-k)(j+n-k+1)}{(j+m-k+1)(j+1)}.
\]

Therefore,

\[
f_k(x) = c \cdot _2F_1 (-k, n - k + 1; m - k + 1; x),
\]
for some constant $c$. There is very little known about the location of zeros of hypergeometric functions of the kind described above [46]. However, algorithms to compute the zeros have been recently studied [40].

To see the dependence of the structural bound on $n$, $m$ and $k$, we plot $r^2_1/r^2_k$ as a function of $n$ in Figures 1.1 and 1.2. We assume that $S^2_j$’s are all equal. Several properties of the bound can be guessed from the plots. In particular, 

1. For a given $m$ and $k$, the ratio $r^2_1/r^2_k$ increases when we increment $n$.

2. For a given $m$ and $k$, the ratio $r^2_1/r^2_k$ approaches a constant as we let $n \to \infty$.

In other words, $r^2_1/r^2_k$ can be upper bound by a constant that is independent of $n$.

3. For a given $n$ and $m$, the ratio $r^2_1/r^2_k$ increases when we increment $k$, and finally,

4. For a given $n$ and $k$, the ratio $r^2_1/r^2_k$ decreases when we increment $m$.

Each of the statements above is stated and proved in the form of Theorems below. Alongside each Theorem, a result of the same flavor is proved for the RIP ratio for $\Phi$, if such a result exists.

The following Theorem shows that when we keep $m$ and $k$ fixed, the RIP constant is a strictly increasing function of $n$.

**Theorem 8** Let $r^2_1 > r^2_2 > ... > r^2_k$ be the zeros of the polynomial $D^{n-k}[x^{n-m}(x-1)^m]$, and let $t^2_1 > t^2_2 > ... > t^2_k$ be the zeros of the polynomial $D^{(n+1)-k}[x^{(n+1)-m}(x-1)^m]$. Then,

$$ \frac{r^2_1}{r^2_k} \leq \frac{t^2_1}{t^2_k}. \tag{1.44} $$
Figure 1.1  Structural bound for $k = 4$ and $m = 5, 6, \ldots, 10$.

Proof: We have

$$D^{n+1-k}[x^{n+1-m}(x-1)^m] = D^{n-k}[D[x^{n+1-m}(x-1)^m]]$$

$$= D^{n-k}[(n+1-m)x^{n-m}(x-1)^m + mx^{n+1-m}(x-1)^{m-1}]$$

$$= D^{n-k} [x^{n-m}(x-1)^{m-1} \{(n+1-m)(x-1) + mx\}]$$

$$= (n+1)D^{n-k} [x^{n-m}(x-1)^{m-1} \left(x - \left(1 - \frac{m}{n+1}\right)\right)] \ (1.45)$$

Comparing equation (1.45) to (1.5) reveals that the polynomial $f_k(x)$ evaluated for $n + 1$ and $m$ is of the same form (up to a constant) of $g_k(x)$ evaluated for $n$ and $k$ with all singular values equal except one with value $S^2_{m} = \left(1 - \frac{m}{n+1}\right)$. Because the singular values are not all equal, Theorem 7 ensures that $r_{k}^{2} \leq r_{k}^{2}$. This completes the proof of Theorem 8.  □
Figure 1.2  Structural bound for $k = 10$ and $m = 11, 12, ..., 15$.

Theorem 9  Let $\Phi_1 \in \mathbb{K}^{m \times n}$ be a submatrix of $\Phi_2 \in \mathbb{K}^{m \times (n+1)}$. Then, the RIP ratios of the two matrices satisfy $R(\Phi_1, k) \leq R(\Phi_2, k)$.

Proof: Let $\Lambda_1$ be the set of all squared singular values ($s^2_{\rho,i}$) of all submatrices of $\Phi_1$, and let $\Lambda_2$ be the set of all squared singular values of all submatrices of $\Phi_2$. Since every $m \times k$ submatrix of $\Phi_1$ is also a submatrix of $\Phi_2$, we have $\Lambda_1 \subset \Lambda 2$. Therefore, $\max\{\Lambda_1\} \leq \max\{\Lambda_2\}$ and $\min\{\Lambda_1\} \geq \min\{\Lambda_2\}$. Because $R(\Phi_1, k) = \max\{\Lambda_1\}/\min\{\Lambda_1\}$ and $R(\Phi_2, k) = \max\{\Lambda_2\}/\min\{\Lambda_2\}$, the result of Theorem 9 follows as a consequence.

The following Theorem shows that the structural bound can itself be bound by a quantity that is independent of $n$. 
Theorem 10  Let $r_1^2 > r_2^2 > \ldots > r_k^2$ be the zeros of the polynomial \(D_{n-k}[x^{n-m}(x-1)^m]\). Then,

\[
\frac{r_1^2}{r_k^2} \leq \frac{(mk)^k(m-k)!}{m!}. \tag{1.46}
\]

Proof: We establish the result by deriving an upper bound for $r_1^2$ and a lower bound for $r_k^2$. By Viète’s Theorem, we have

\[
r_1^2 + r_2^2 + \ldots + r_k^2 = \frac{mk}{n}. \tag{1.47}
\]

Since each term in the above equation is positive, we have

\[
r_1^2 \leq \frac{mk}{n}, \tag{1.48}
\]

giving us an upper bound on $r_1^2$.

To derive a lower bound on $r_k^2$, we begin by using Viète’s Theorem for the constant term of Equation (1.39), involving the product $r_1^2r_2^2\ldots r_k^2$:

\[
r_1^2r_2^2\ldots r_k^2 = \binom{m}{k} \frac{n}{k}. \tag{1.49}
\]

Since $r_i^2 > r_k^2$ for all $i < k$, we have the inequality

\[
r_k^2(r_1^2)^{k-1} \geq \binom{m}{k} \frac{n}{k}. \tag{1.50}
\]

Applying (1.48) to the above inequality,

\[
r_k^2 \left( \frac{mk}{n} \right)^{k-1} \geq r_k^2(r_1^2)^{k-1} \geq \binom{m}{k} \frac{n}{k}. \tag{1.51}
\]
Therefore,
\[
\begin{align*}
\frac{r_k^2}{k} & \geq \frac{\binom{m}{k}}{\binom{n}{k}} \cdot \frac{n^{k-1}}{(mk)^{k-1}} \\
& = \frac{m!(n-k)!n^{k-1}}{(m-k)!n!(mk)^{k-1}} \\
& = \frac{m!}{n(m-k)!(mk)^{k-1}} \cdot \frac{(n-k)!n^k}{n!} \\
& \geq \frac{m!}{n(m-k)!(mk)^{k-1}},
\end{align*}
\]
where the last inequality (1.52) is a consequence of \( n^k \geq n!/(n-k)! \). Combining the inequalities (1.48) and (1.52) by taking the ratio (noting that both inequalities have positive LHS and RHS), we get the inequality in (1.46), completing the proof of Theorem 10.

Note that we cannot make a similar assertion about the RIP ratio of a matrix \( \Phi \) as we increase \( n \). In other words, \( R(\Phi, k) \) necessarily depends on \( n \). This is a consequence of the packing bound, to be discussed in Section 1.5.

**Theorem 11** Let \( r_1^2 > r_2^2 > \ldots > r_k^2 \) be the zeros of the polynomial \( D^{n-k}[x^{n-m}(x-1)^m] \), and let \( t_1^2 > t_2^2 > \ldots > t_k^2 \) be the zeros of the polynomial \( D^{n-k}[x^{n-m+1}(x-1)^{m-1}] \). Then,
\[
\frac{r_1^2}{r_k^2} \leq \frac{t_1^2}{t_k^2},
\]
(1.53)

**Proof**: Reducing \( m \) by one is equivalent to the following operation: set \( S_m^2 = 0 \). The proof of the above Theorem follows therefore from a straightforward application of Theorem 7.

**Theorem 12** Let \( \Phi_1 \in \mathbb{K}^{m \times n} \) be a submatrix of \( \Phi_2 \in \mathbb{K}^{(m+1) \times n} \). Then, the RIP ratios of the two matrices satisfy \( R(\Phi_1, k) \leq R(\Phi_2, k) \).
Proof: Let \( p \in \{1, 2, ..., \binom{n}{k}\} \) be the index that identifies the set of \( k \) columns that are picked from the parent matrix to form a submatrix with \( k \) columns. Then, \((\Phi_1)_p\) is a submatrix of \((\Phi_2)_p\), and consequently, the maximum (minimum) singular value of \((\Phi_1)_p\) is smaller (greater) than the maximum (minimum) singular value of \((\Phi_2)_p\) by the interlacing Theorem for matrices [80]. Thus Theorem 12 is established.

Theorem 13 Let \( r_1^2 > r_2^2 > ... > r_k^2 \) be the zeros of the polynomial \( D^{n-k}[x^{n-m}(x-1)^m] \), and let \( t_1^2 > t_2^2 > ... > t_{k+1}^2 \) be the zeros of the polynomial \( D^{n-k-1}[x^{n-m+1}(x-1)^{m-1}] \). Then,

\[
\frac{r_1^2}{r_k^2} \leq \frac{t_1^2}{t_{k+1}^2}.
\]

Proof: Since \( D^{n-k-1}[x^{n-m+1}(x-1)^{m-1}] = D[D^{n-k}[x^{n-m}(x-1)^m]] \), the roots of the two polynomials weakly interlace, as a consequence of the interlacing theorem for a polynomial and its derivative [62]. Therefore, \( r_1^2 \leq t_1^2 \) and \( r_k^2 \geq t_{k+1}^2 \) and the proof of Theorem 13 follows as a consequence.

Finally, we state a similar Theorem for matrices.

Theorem 14 Let \( \Phi_1 \in \mathbb{K}^{m \times n} \). Then, the RIP ratios satisfy \( R(\Phi_1, k) \leq R(\Phi_2, k + 1) \).

Proof of the above Theorem is identical to the proof of Theorem 12.

1.4.3 Geometric Interpretation

Recall the geometric interpretation of the SVD: The matrix \( \Phi \) with SVD \( \Phi = U S V^T \) can be represented as a hyperellipse of dimension \( m \) embedded in \( \mathbb{R}^n \). The axes of the hyperellipse are aligned with the column vectors of \( V \) and the length of each semi-axis
is equal to the corresponding singular value. Let us denote this hyperellipse by $E(\Phi)$. It is well known [49] that $\|\Phi x\|_2$ is equal to the magnitude of the projection of the vector $x$ onto the hyperellipse $E(\Phi)$. Furthermore, the column vectors of $U$ describe the orientation of the hyperellipse in $\mathbb{R}^m$ which is the image of $\Phi x$ of the unit sphere in $\mathbb{R}^n$, ie., for $\|x\|_2 = 1$.

$$\binom{n-q}{k-q} \sum_{q\text{-wise}} (S_1S_2...S_q)^2 = \sum_p \left\{ \sum_{q\text{-wise}} (s_{p,1}s_{p,2}...s_{p,q})^2 \right\}. \quad (1.55)$$

Equation (1.55) relates the dimensions of $E(\Phi)$ to the dimensions of the collection of $k$-dimensional hyperellipses $E(\Phi_p)$ corresponding to $\Phi_p$. Note that each of the hyperellipses $E(\Phi_p)$ lie in a $k$-dimensional subspace spanned by $k$ canonical basis vectors. A particularly interesting case is when $m = k = q$, for which (1.55) reduces to

$$(S_1S_2...S_k)^2 = \sum_p (s_{p,1}s_{p,2}...s_{p,k})^2. \quad (1.56)$$

The above result for $m = k = q$ is equivalent to the Generalized Pythagorean Theorem (GPT) [19, 58]. GPT states that the square of the $k$-volume of a $k$-dimensional parallelepiped embedded in an $n$-dimensional Euclidean space is equal to the sum of the squares of the $k$-volumes of the $\binom{n}{k}$ projections of the parallelepiped on to the distinct $k$-dimensional subspaces spanned by the canonical basis vectors. Equation (1.56) implies that the statement of the GPT can be directly carried over from parallelepipeds to hyperellipses.

Equation (1.55) extends GPT to arbitrary values of $m, k$ and $q$. The form of the equation motivates us to define the $q$-volume of an ellipse of intrinsic dimension that is greater than $q$ as follows.

\footnote{Note that the hyperellipses $E(\Phi_p)$ are the projections of $E(\Phi)$ onto the canonical $k$-subspaces only when $m = k$.}
**Definition 2** Consider a hyperellipse $H$ of dimension $d > q$ with semi-axes $a_1, a_2, ..., a_d$. The $q$-volume of a $H$, denoted by $\text{Vol}_q(H)$ is defined as

$$\text{Vol}_q(H) \triangleq \sqrt{\sum_{q\text{-wise}} (a_1 a_2 ... a_q)^2}.$$  \hfill (1.57)

Equation (1.55) therefore relates the $q$-volumes of $E(\Phi)$ to the $q$-volumes of $E(\Phi_p)$ in the following manner: the square of the $q$-volume of $E(\Phi)$ is proportional to the sum of the squares of the $q$-volumes of $E(\Phi_p)$.

**1.4.4 Structural bound for $k = 2$**

In this Section, we study the structural bound for the specific case of $k = 2$. The motivation for studying this case are many fold. First, $k = 2$ is the smallest non-trivial case to investigate the RIP ratio. For the case $k = 1$, any matrix $\Phi$ that has equi-normed columns satisfy $R(\Phi, 1) = 1$ and therefore the structural bound is trivially 1 for any $n$ and $m$. Secondly, the roots of the polynomial (1.39) can be explicitly evaluated, providing avenue for analysis. Lastly and most importantly, designing good CS matrices for $k = 2$ can be shown to be equivalent to well-known problems in coding theory.

For $k = 2$, the form of (1.39) and (1.40) reduce to

$$\frac{n(n - 1)}{2} x^2 - \left(\sum_{j=1}^m S_j^2\right) (n - 1)x + \left(\sum_{1 \leq j_1 < j_2 \leq m} S_{j_1}^2 S_{j_2}^2\right) = 0 \hfill (1.58)$$

and

$$\frac{n(n - 1)}{2} x^2 - (n - 1)mx + \frac{m(m - 1)}{2} = 0. \hfill (1.59)$$

We focus on (1.59), because are interested in universal bounds for RIP. The roots of (1.59) can be computed as

$$r_1^2, r_2^2 = \frac{m}{n} \pm \frac{1}{n} \sqrt{\frac{m(n - m)}{n - 1}}.$$  \hfill (1.60)
and the structural bound is thus

\[
\frac{r_1^2}{r_2^2} = \frac{1 + \sqrt{\frac{n-m}{m(n-1)}}}{1 - \sqrt{\frac{n-m}{m(n-1)}}}.
\]

(1.61)

Note that as \(n \to \infty\), the above equation reduces to

\[
\lim_{n \to \infty} \frac{r_1^2}{r_2^2} = \frac{1 + \sqrt{\frac{1}{m}}}{1 - \sqrt{\frac{1}{m}}} = \left(1 + \sqrt{\frac{1}{m}}\right)^2
\]

(1.62)

(1.63)

From our bound before (1.46), we had

\[
\frac{r_1^2}{r_k^2} \leq \frac{(2m)^2(m - 2!)}{m!} = \frac{4m^2}{m(m-1)} = \frac{4}{1 - \frac{1}{m}}
\]

(1.64)

(1.65)

(1.66)

which shows that the asymptotic bound in Theorem 10 is loose by a maximum factor of four for the case \(k = 2\).

We now make important connections of good CS matrices for \(k = 2\) to coding theory. The key result that provides the segue is the following Theorem.

**Theorem 15** Let \(A = [a_1 \ a_2]\) be a matrix in \(\mathbb{K}^{m \times 2}\) comprising of two columns \(a_1, a_2 \in \mathbb{K}^{m \times 1}\), with \(\|a_1\|_2 > 0\) and \(\|a_2\|_2 > 0\). Create the matrix

\(B \in \mathbb{K}^{m \times 2}\) as \(B = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \). Then, the RIP ratios of \(A\) and \(B\) satisfy

\(R(B, 2) \leq R(A, 2)\).

**Proof:** To simplify the proof, we transform the matrix \(A\) by performing two operations that both preserve the RIP ratio. First, we pre-multiply \(A\) by a unitary matrix such that the first column of \(A\) is transformed to a vector that is aligned with a
coordinate axis, say the first coordinate without loss of generality. Note that this operation preserves the RIP ratio. Second, we scale the matrix such that the second column has \( \ell_2 \) norm of unity, noting that scaling preserves the RIP ratio.

Assume that the two transformations described above has been performed on \( A \). We thus have \( \|a_2\|^2 = 1 \) and \( a_1^H = [a \ 0 \ 0...0] \) for some scalar \( a \). Let \( a_2^H = [u_1 \ u_2...u_m] \) such that \( u_1^2 + u_2^2 + ... + u_m^2 = 1 \).

Let \( s_1^2 \) and \( s_2^2 \) be the squared singular values of \( A \). Since the squared singular values of \( A \) are the eigen values of the matrix \( A^H A \), we compute \( A^H A \) as

\[
A^H A = \begin{pmatrix} a^2 & au_1 \\ au_1 & 1 \end{pmatrix}
\]

Thus \( s_1^2 \) and \( s_2^2 \) are the zeros of the characteristic polynomial of (1.67) given by

\[
x^2 - (1 + a)^2 + a^2(1 - u_1^2) = 0.
\]

Computing the roots yield

\[
s_1^2, s_2^2 = \frac{1}{2} \left\{ (1 + a^2) \pm \sqrt{1 + a^4 - 2a^2 + 4a^2u_1^2} \right\}.
\]

Setting \( a \) as a variable parameter, we carry out an exercise in calculus to minimize the ratio \( s_1^2/s_2^2 \) as a function of \( a \), by setting \( (d/da)(s_1^2/s_2^2) = 0 \). Omitting the algebraic details, we arrive at the condition \( a(a^2 - 1)(u_1 - 1) = 0 \). The cases \( a = 0 \) and \( u_1 = 0 \) gives rise to an RIP ratio of infinity (a maxima). The interesting case is when \( a = \pm 1 \), which minimizes the RIP ratio. The condition \( a = \pm 1 \) implies that the column vectors \( a_1 \) and \( a_2 \) are of equal norm, and thus we have proved Theorem 15.

The above Theorem has important consequences for RIP ratio of order \( k = 2 \), and the observations are given below.

**Remark 2** Theorem 15 reveals that among the set of all matrices that can be obtained by scaling each column of a given matrix independently, the RIP ratio is minimized when the column norms are equal.
Remark 3 Let $\Phi_1 \in \mathbb{K}^{m \times n}$. Construct another matrix $\Phi_2$ of the same size by scaling each column of $\Phi_1$ yielding equi-normed columns for $\Phi_2$. Theorem 15 guarantees that $R(\Phi_2, 2) \leq R(\Phi_1, 2)$.

We use the above results to study the properties of a CS matrix $\Phi$ that attains the structural bound for $k = 2$. First, Theorem 1 (statement 3) requires that $\Phi_p$ has the same pair of squared singular values for all $p$. Second, we deduce from Theorem 15 that the columns of $\Phi$ are all equi-normed. If the contrary were true, then equalizing the column norms will yield a matrix with smaller RIP ratio, which violates Theorem 1. Since this result is of importance, we state it in the form of a Theorem.

**Theorem 16** If $\Phi \in \mathbb{K}^{m \times n}$ satisfies (1.5) with equality, then $\Phi$ has equi-normed columns.

Furthermore, we show that $\Phi$ is an equi-angular tight frame (ETF) [73]. From [73], we list the three conditions for a matrix $A \in \mathbb{K}^{m \times n}$ to be an ETF:

1. The columns of $A$ are unit normed,

2. The absolute value of the dot product of every pair of columns of $A$ are same, ie., the columns are equi-angular. And lastly,

3. $AA^* = (n/m)I_{m \times m}$.

If only conditions (1) and (2) are satisfied the columns of $A$ form a system of equi-angular lines [86, 52].

**Theorem 17** Let $\Phi \in \mathbb{K}^{m \times n}$ satisfy (1.6) with equality, and let $\Phi$ be scaled such that its $m$ squared singular values are equal to $n/m$ each. Then, $\Phi$ is an ETF.
**Proof:** Let $\Phi = USV^*$ be the singular value decomposition for $\Phi$. We check the three conditions for ETF, starting with the third condition. We have $AA^* = USV^*VSU^* = US^2U^* = (n/m)UIU^* = (n/m)I$, satisfying condition (3). Condition (2) is satisfied because every pair of columns have the same set of singular values. In fact, one can show that of the angle between two equi-normed columns is $\theta$ and the common norm is $d$, then the squared singular values of the matrix comprising only of the two said columns are $d^2(1 + \cos(\theta))$ and $d^2(1 - \cos(\theta))$. To verify condition (1), we first note that the norms of each column of $\Phi$ are the same, say $d^2$. It remains to show that this norm is unity. To see this, we use Equation (1.58) and set $S_1^2 = S_2^2 = n/m$. For this choice of $S_1$ and $S_2$, we see that $r_1^2$ and $r_2^2$ are the roots of the equation $x^2 - 2x + \frac{n(m-1)}{m(n-1)} = 0$, yielding

$$r_1^2 + r_2^2 = 2.$$  \hfill (1.69)

We have $r_1^2 = d^2(1 + \cos(\theta))$ and $r_2^2 = d^2(1 - \cos(\theta))$ and so

$$r_1^2 + r_2^2 = 2d^2.$$  \hfill (1.70)

From (1.69) and (1.70), we have $d^2 = 1$, and thus $\Phi$ satisfies all three conditions for an ETF.

The relationship of structural bound to the concept of ETF’s can be used to make statements about the set of allowable pairs $(m, n)$ that meet the structural bound. Results from [73] reveal that an ETF of size $m \times n$ exist only when $n \leq \frac{1}{2}m(m + 1)$ for real ETF’s and $n \leq m^2$ for a complex ETF. In addition, $n$ and $m$ should satisfy strict integer constraints.

### 1.5 Packing bound for RIP

The motivation for another bound for RIP is that we can view the CS matrix $\Phi$ as a collection of column vectors in $\mathbb{R}^m$, and we need to spread these vectors as far away
from each other as possible in order to ensure that the singular values of its $k$-column submatrices have good condition numbers. Increasing the number of columns (ie., increasing $n$) leads to crowding of these vectors in $\mathbb{R}^m$ that leads to deterioration of the RIP ratio. We make these notions precise for $k = 2$. For $k > 2$, the exact nature of the packing bounds are yet elusive.

As we saw from Theorem 15, we need to restrict our attention only to matrices of equi-normed columns for designing good CS matrices for $k = 2$. Therefore, the problem of designing good CS matrices for $k = 2$ is equivalent to finding arrangements of $n$ lines in $\mathbb{R}^m$ such that the minimum angle between the pairs of lines is maximized. This problem has been studied extensively by Conway, Sloane et al. [20]. Furthermore, converse and achievable bounds can been derived by studying a related problem, namely of arrangements of $2n$ points on a Euclidean sphere in $\mathbb{R}^m$. The latter problem has been studied independently by Shannon, Chabauty and Wyner [35]. We state the main results that are relevant to our problem of bounding RIP. For a detailed description and derivation of the packing bounds, refer [35].

**Definition 3** The area of a spherical cap of radius $\beta$ on an $m$ dimensional Euclidean sphere of unit length is given by

$$C_m(\beta) = k_m \int_0^{\beta} \sin^{m-2} \alpha d\alpha,$$

where $k_m$ is a constant for a given $m$.

Note that $2C_m(\pi)$ gives the surface area of a unit sphere in $\mathbb{R}^m$. Packing the surface of the $m$-dimensional sphere using spherical caps gives rise to an upper bound on the minimum angle $\theta$ that can be attained between the pairs of $2n$ points on the sphere. The upper bound is given by $C_m(\theta_{\text{max}}/2) = \frac{C_m(\pi)}{n}$. Consequently, we get a lower bound on the RIP ratio, and we state it in the following Theorem.
Theorem 18  Let $\theta$ satisfy
\[
C_m(\theta/2) = \frac{C_m(\pi)}{n}.
\]
(1.72)

Then, there exist no $\Phi \in \mathbb{R}^{m \times n}$ such that $\rho(\Phi, 2) \leq \frac{1 + \cos(\theta)}{1 - \cos(\theta)}$.

Proof: The statement follows from the fact that for any arrangement of $2n$ points on the surface of the unit sphere in $\mathbb{R}^m$, the minimum angle between pairs of points is less than $\theta$ defined above, as a result of packing. 

Furthermore, covering arguments can be used to make a statement on achievability. The following Theorem is based on the result of Shannon, Chabauty and Wyner [35, 21] which asserts that there exists an arrangement of $2n$ points on the Euclidean sphere in $\mathbb{R}^m$ whose minimum angle is at least as large as $\theta$, where $C_m(\theta) = \frac{C_m(\pi)}{n}$. To use this result for an achievable bound for RIP, we need to account for the fact that the $2n$ points in our case are antipodal, and therefore the above condition should be modified to $C_m(\theta) = \frac{C_m(\pi - \theta/2)}{n}$.

Theorem 19  Let $\theta$ satisfy
\[
C_m(\theta) = \frac{C_m(\pi - \theta/2)}{n}
\]
and let $q = \frac{1 + \cos(\theta)}{1 - \cos(\theta)}$. Then, there exists a $\Phi \in \mathbb{R}^{m \times n}$ such that $\rho(\Phi, 2) \leq q$.

1.6 Tightness of RIP bounds

In this Section, we study the tightness of the structural and packing bounds by comparing with RIP ratios for actual constructions of $\Phi$. We perform an extensive study for $k = 2$ using the data for the best Grassmannian packings maintained by Sloane [69, 70].
Figures 1.3, 1.4, 1.5 and 1.6 depict the plots for the RIP ratio for $k = 2$ for different values of $m$. We note that for small $n$, the structural bound dictates the RIP ratio of the best CS matrix. As we increase $n$, the packing bound overtakes the structural bound in value and thus controls the behavior of the RIP ratio for the best $\Phi$. Note also that the iid Gaussian construction of $\Phi$ has a much higher RIP ratio (up to two orders of magnitude) which points to the value of deterministic constructions of $\Phi$. 

**Figure 1.3**  RIP bounds for $m = 3$, $k = 2$. 

- Gaussian
- C–S–W achievable (UB)
- Best known Phi
- Structural (LB)
- Packing (LB)
Figure 1.4  RIP bounds for $m = 6, k = 2$.

Figure 1.5  RIP bounds for $m = 8, k = 2$. 
Figure 1.6 RIP bounds for $m = 16$, $k = 2$.

Figure 1.7 RIP ratio for $m = 4$ for Gaussian matrices.
Figure 1.8  RIP ratio for $m = 8$ for Gaussian matrices.
1.7 Relevance of the structural bounds in stochastic RIP

In this section, we explore the notion of stochastic RIP, where we consider only a small number of the submatrices of $\Phi$ and compute the maximum and minimum squared singular values of these matrices. The motivation to consider this measure is the fact that the number $\binom{n}{k}$ of submatrices of $\Phi$ is huge. It is very unlikely that a randomly selected $k$-sparse signal has the exact sparsity pattern corresponding to the submatrices of $\Phi$ with extreme values for the squared singular values. Toward this goal, we perform the following simple yet illuminating simulation study. We pick a $\Phi \in \mathbb{R}^{m \times n}$ at random having prescribed singular values, and pick a fixed number (we used 10 in our study) of submatrices of $\Phi$ of size $m \times k$. We compute the stochastic RIP ratio as the ratio of the maximum and minimum squared singular values for this set of submatrices. We repeat for several trials of $\Phi$ (we used 5000 trials in our study) and averaged the stochastic RIP.

Figures 1.9 and 1.10 show the results of this study for $k = 2$ and $m = 4$ and 8. The plots reveal that the expected value of the stochastic RIP is very close to the structural bound, and always exceeds the structural bound. Furthermore, the packing bound seems to have no impact on the expected stochastic RIP even for large values of $n$.

The above observations hint at the possibility that the zeros of the polynomial in (1.5) are in fact estimates of the squared singular values of a randomly picked submatrix of $\Phi$. Additional theoretical evidence stems from statement (2) of Theorems 1 and 2 indicate $r_1^2$ and $r_k^2$ as estimates of $s_{p,1}^2$ and $s_{p,k}^2$ respectively. We ran the following simulation to test our suspicion. We pick a single $\Phi$ of moderate size with prescribed set of singular values, and select a large number of submatrices of $\Phi$ of size $m \times k$. We compute the $k$ singular values of each of the submatrices. Finally, we
plot the $k$ histograms of the respective squared singular values and compare against the $r_i^2$'s. Figures 1.11, 1.12 and 1.13 show these plots for a set of 4 out of $k$ singular values (to prevent clutter in the plot). Notice that the $r_i^2$'s provide remarkably good estimate for the $s_{p,i}^2$'s.
Figure 1.10  Stochastic RIP ratio $m = 8$ for Gaussian matrices, for one submatrix.
Figure 1.11  Histogram of the squared singular values for $n = 100$, $m = 25$, $k = 12$. The histogram is plotted for the 1st, 4th, 8th and 12th singular values of 20000 randomly chosen submatrices of $\Phi$. The parent matrix $\Phi$ is chosen with $m$ prescribed singular values all equal to unity. The vertical lines correspond to the 1st, 4th, 8th and 12th zeros of the polynomial in (1.5).
1.8 Avenues for future work

The present study of deterministic bounds for RIP opens up a lot of interesting research questions. First, we are in pursuit of the relationships between the structural and packing bounds to the Johnson-Lindenstrauss Lemma. Second, we plan to formalize the notion of stochastic RIP described in Section 1.7 and identify the precise nature of the estimate $r_i^2$ of the singular values $s_{p,i}^2$. Third, we explore how the results can be extended to universal Compressed Sensing. In the current work, we assume that the signal is sparse in the canonical basis. How do the bounds on RIP change when we sparsity basis is unknown? Finally, we plan to derive the packing bounds for $k > 2$.

1.9 Conclusions

In this chapter, we have derived two converse bounds for RIP – the first based on structural bounds for singular values of submatrices and the second based on packing arguments. The derivation of the two bounds offer rich geometric interpretation and illuminate the relationships between CS matrices and equi-angular tight frames, codes on grassmannian spaces and Euclidean spheres, and the generalized Pythagorean Theorem.
Figure 1.12  Histogram of the squared singular values for $n = 100$, $m = 60$, $k = 12$. The histogram is plotted for the first, fourth, eighth and twelfth singular values of 20000 randomly chosen submatrices of $\Phi$. The parent matrix $\Phi$ is chosen with $m$ prescribed singular values all equal to unity. The vertical lines correspond to the 1st, 4th, 8th and 12th zeros of the polynomial in (1.5).
Figure 1.13  Histogram of the squared singular values for $n = 100$, $m = 14$, $k = 12$. The histogram is plotted for the 1st, 4th, 8th and 12th singular values of 20000 randomly chosen submatrices of $\Phi$. The parent matrix $\Phi$ is chosen with $m$ prescribed singular values all equal to unity. The vertical lines correspond to the 1st, 4th, 8th and 12th zeros of the polynomial in (1.5).
Chapter 2

Compressed Sensing using Belief Propagation

2.1 Introduction

In many signal processing applications the focus is often on identifying and estimating a few significant coefficients from a high dimension vector. The wisdom behind this is the ubiquitous compressibility of signals: most of the information contained in a signal resides in a few large coefficients. Traditional sensing and processing first acquires the entire data, only to throw away most of the coefficients and retaining the small number of significant coefficients. Clearly, it is wasteful to sense or compute all of the coefficients when most of it will be discarded at a later stage. This naturally suggests the question: can we sense compressible signals in a compressible way? In other words, can we sense only that part of the signal that will not be thrown away? The ground-breaking work in compressed sensing (CS) pioneered by Candés et al. [13] and Donoho [28] answers the above question in the affirmative. They demonstrate that the information contained in the few significant coefficients can be captured (encoded) by a small number of random linear projections. The original signal can then be reconstructed (decoded) from these random projections using an appropriate decoding scheme.

The initial discovery has led to a vibrant activity in the area of CS research, opening many intriguing questions both in theoretical and practical aspects of CS. Two fundamental questions naturally emerge in compressed sensing. The first question concerns efficiency: what is the minimum number of projections required to capture
the information contained in the signal (either losslessly or with a certain fidelity)? Clearly, the number of measurements needed depends on the signal model as well as the measurement model. The second question relates to algorithmic achievability: can we construct practical CS coding schemes that approach the performance limits? Questions such as these have been addressed in various other contexts. The vision of our work is to leverage the insights from information theory [25] to obtain new CS coding algorithms. In particular, we draw insights from low density parity check (LDPC) codes.

2.1.1 Information theory

The main problem that information theory deals with is reliable transmission of information over communication channels. A fundamental result by Shannon states that the upper-limit on the rate at which we can send information over a channel is given by the channel capacity [68]. Since Shannon’s seminal work, several approaches to building practical codes have been proposed. The emphasis has been on imposing structure to the codewords; for example, explicit algebraic construction of very good codes were designed for some channels. Although the decoding of such codes has polynomial complexity (and thus practical), most of these codes fared poorly in the asymptotic regime – they achieved arbitrarily small probabilities of decoding errors only by decreasing the information rate to zero. The grand breakthrough occurred quite recently, with the advent of turbo codes [9] and the rediscovery of LDPC codes [39]. These codes belong to a class of linear error correcting codes that use sparse parity check matrices and achieve information rates close to the Shannon limit. In addition to their excellent performance, turbo and LDPC codes lend themselves to simple and practical decoding algorithms, thanks to the sparse structure of the parity check matrices.
2.1.2 Compressed sensing

Consider a discrete-time signal $x \in \mathbb{R}^N$ that has only $K \ll N$ non-zero coefficients. The core tenet of CS is that it is unnecessary to measure all the $N$ values of $x$; rather, we can recover $x$ from a small number of projections onto an incoherent basis [13, 28]. To measure (encode) $x$, we compute the measurement vector $y \in \mathbb{R}^M$ as $M$ linear projections of $x$ via the matrix-vector multiplication $y = \Phi x$. Our goal is to reconstruct (decode) $x$ – either accurately or approximately – given $y$ and $\Phi$ using $M \ll N$ measurements.

Although the recovery of the signal $x$ from the measurements $y = \Phi x$ appears to be a severely ill-posed inverse problem, the strong prior knowledge of sparsity in $x$ gives us hope to reconstruct $x$ using $M \ll N$ measurements. In fact the signal recovery can be achieved using optimization by searching for the sparsest signal that agrees with the $M$ observed measurements in $y$. The key observation is that the signal $x$ is the solution to the $\ell_0$ minimization

$$\hat{x} = \arg \min_x \|x\|_0 \text{ s.t. } y = \Phi x$$

with overwhelming probability as long as we have sufficiently many measurements, where $\| \cdot \|_0$ denotes the $\ell_0$ “norm” that counts the number of non-zero elements. Unfortunately, solving the $\ell_0$ optimization is known to be an NP-complete problem [15]. In order to recover the signal, the decoder needs to perform combinatorial enumeration of all the $\binom{N}{K}$ possible sparse subspaces.

The practical revelation that supports the CS theory is that it is not necessary to resort to combinatorial search to recover the set of non-zero coefficients of $x$ from the ill-posed inverse problem $y = \Phi x$. A much easier problem yields an equivalent solution. We need only solve for the $\ell_1$-sparsest coefficients that agree with the
measurements $y$ [13, 28]

$$\hat{x} = \arg \min \|x\|_1 \quad \text{s.t. } y = \Phi x,$$

(2.2)
as long as $\Phi$ satisfies the restricted isometry (RIP) condition [13]. The RIP condition is shown to be satisfied by a measurement strategy using independent and identically distributed (iid) Gaussian entries for $\Phi$. Furthermore, the $\ell_1$ optimization problem, also known as Basis Pursuit [17], is significantly more approachable and can be solved with linear programming techniques. The decoder based on linear programming requires $cK$ projections for signal reconstruction where $c \approx \log_2(1 + N/K)$ [7] and reconstruction complexity is $\Omega(N^3)$ [15, 30, 29].

### 2.1.3 Compressed sensing reconstruction algorithms

While linear programming techniques figure prominently in the design of tractable CS decoders, their $\Omega(N^3)$ complexity still renders them impractical for many applications. We often encounter sparse signals with large $N$; for example, current digital cameras acquire images with the number of pixels $N$ of the order of $10^6$ or more. For such applications, the need for faster decoding algorithms is critical. Furthermore, the encoding also has high complexity; encoding with a full Gaussian $\Phi$ requires $\Theta(MN)$ computations.\(^7\) This realization has spawned a large number of decoding schemes in the CS research community in search of new measurement strategies and accompanying low-complexity decoders. We briefly review some of the previous work.

At the expense of slightly more measurements, iterative greedy algorithms have been developed to recover the signal $x$ from the measurements $y$. Examples include the iterative Orthogonal Matching Pursuit (OMP) [83], matching pursuit (MP), and

\(^7\)For two functions $f(n)$ and $g(n)$, $f(n) = O(g(n))$ if $\exists c, n_0 \in \mathbb{R}^+$, $N \leq U(\infty) \leq \delta(\infty)$ for all $n > n_0$. Similarly, $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$ and $f(n) = \Theta(g(n))$ if $\exists c_1, c_2, n_0 \in \mathbb{R}^+$, $N \leq \nu \delta(\infty) \leq U(\infty) \leq \nu \delta(\infty)$ for all $n > n_0$. 
tree matching pursuit (TMP) \cite{34} algorithms. In CS applications, OMP requires \(c \approx 2 \ln(N)\) \cite{83} to succeed with high probability; decoding complexity is \(\Theta(NK^2)\). Unfortunately, \(\Theta(NK^2)\) is cubic in \(N\) and \(K\), and therefore OMP is also impractical for large \(K\) and \(N\).

Donoho et al. recently proposed the Stage-wise Orthogonal Matching Pursuit (StOMP) \cite{31}. StOMP is an enhanced version of OMP where multiple coefficients are resolved at each stage of the greedy algorithm, as opposed to only one in the case of OMP. Moreover, StOMP takes a fixed number of stages while OMP takes many stages to recover the large coefficients of \(x\). The authors show that StOMP with fast operators for \(\Phi\) (such as permuted FFT’s) can recover the signal in \(N \log N\) complexity. Therefore StOMP runs much faster than OMP or \(\ell_1\) minimization and can be used for solving large-scale problems.

While the CS algorithms discussed above use a full \(\Phi\) (all the entries of \(\Phi\) are non-zero in general), a class of techniques has emerged that employ sparse \(\Phi\) and use group testing to decode \(x\). Cormode and Muthukrishnan proposed a fast algorithm based on group testing \cite{24, 23}. Their scheme considers subsets of the signal coefficients in which we expect at most one large coefficient to lie. Within this set, they locate the position and value of the large coefficient using a Hamming code construction. Their decoding scheme has \(O(K \log^2(N))\) complexity, but they require \(M = O(K \log^2(N))\) measurements.

Gilbert et al. \cite{41} propose the Chaining Pursuit (CP) algorithm for reconstructing compressible signals. CP reconstructs with \(O(K \log^2(N))\) non-adaptive linear measurements in \(O(K \log^2(N) \log^2(K))\) time. Simulations reveal that CP works best for “super-sparse” signals, where the sparsity rate \(S \triangleq K/N\) is very small \cite{6}. As we increase \(S\), CP requires an enormous number of measurements; for some ranges of \(S\) the number of measurements \(M\) exceeds the signal length \(N\), which is undesirable.
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Setting</th>
<th>$M_{\text{min}}$</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_0$ Optimization</td>
<td>noise-free</td>
<td>$K + 1$</td>
<td>NP complete</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\ell_1$ Optimization</td>
<td>noise-free</td>
<td>$K \log(1 + N/K)$</td>
<td>$\Omega(N^3)$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\ell_1$ regularization</td>
<td>noisy</td>
<td>$K \log(1 + N/K)$</td>
<td>$\Omega(N^3)$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMP</td>
<td>noiseless</td>
<td>$2K \log N$</td>
<td>$NK^2$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StOMP</td>
<td>noiseless</td>
<td>$K \log N$</td>
<td>$N \log N$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cormode-Muthu</td>
<td>noiseless</td>
<td>$K \log^2 N$</td>
<td>$K \log^2 N$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chaining Pursuit</td>
<td>noiseless</td>
<td>$K \log^2 N$</td>
<td>$K \log^2 N \log^2 K$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sudocodes [6]</td>
<td>noiseless</td>
<td>$K \log N$</td>
<td>$K(\log K)(\log N)$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-LDPC</td>
<td>noisy</td>
<td>$K \log N$</td>
<td>$N \log N$</td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2.1 summarizes the number of measurements required and computational complexity of several CS schemes proposed in the literature. Note that the efficiency and complexity depends on the signal class considered as well as the reconstruction scheme. The results suggest that we can reduce the CS coding complexity only at the expense of more measurements.

2.1.4 Connections between channel coding and CS

The world of error control coding and CS share intriguing connections. For example, a fully Gaussian/Rademacher CS measurement matrix [13, 28] is reminiscent of Shannon’s fully random code construction. Although a fully random CS matrix efficiently captures the information content of sparse signals with very few projections, they are not amenable to fast encoding and decoding schemes. On the other hand, the grand success of sparse codes in error control coding such as LDPC codes strongly suggests the use of sparse compressed sensing matrices. This leading insight inspires the following questions:

- Can sparse CS matrices efficiently encode sparse signals?

- Can we take advantage of the sparse structure to reduce the encoding and decoding complexity?

To address these questions, we overview LDPC codes and describe how we can leverage various insights to CS.

2.1.5 Lessons from LDPC codes applied to CS

An LDPC code is a block code that has a parity check matrix $H$ that is sparse along every row and every column [54, 39]. The parity check matrix of an LDPC code can be conveniently represented as a bipartite graph (Figure 2.1). The nodes in the
graph are one of two classes: the first class of nodes represent the transmitted bits (called the bit nodes) and the second class of nodes represent the constraints (called check nodes). Although any linear code can be represented by a graph, what makes LDPC codes special is that the bipartite graph is sparse: the number of edges in the graph scales roughly linearly with \( N \), rather than quadratically. The sparsity in the graph representation of LDPC codes simplifies the decoding process: LDPC codes can be decoded by running a low-complexity message-passing algorithm (such as belief-propagation) over the graph.

Message-passing algorithms start with the input consisting of the prior probabilities of the bits. The algorithms use the parity check relationship among the bits to iteratively pass messages between the nodes in the graph and extract the posterior probabilities for the codeword bits. Although these algorithms yield accurate posterior probabilities only when the bipartite graph is cycle free, it turns out that the algorithms compute astonishingly accurate estimates for LDPC decoding in spite of the presence of cycles in the graph.

LDPC codes offer an important insight to the area of CS: namely that sparse coding matrices offer great promise in efficiently encoding signals, as well as enabling low complexity algorithms based on the sparse graph structure. We will later deliver on these promises.

### 2.1.6 Contributions: Connecting CS decoding to graph decoding algorithms

The key idea underlying this chapter is to extend LDPC measurement techniques over a finite alphabet to matrix multiplications over the real numbers. Surprisingly, the usage of similar matrix structures provides superior processing results even in this setting. Despite the significant benefit provided by this extension to real valued
signals, the LDPC matrices that we use are not universal in the sense that they do not work for signals that are sparse or approximately sparse in any basis. Although universality can be addressed by modifying the measurement matrix and reconstruction techniques (Section 2.4), this is only of secondary concern here.

**Encoder**: We have developed strategies to encode (measure) the signal using sparse \( \{0, 1\} \) or \( \{0, 1, -1\} \) LDPC-like matrices. We compute the measurements \( y = \Phi x \) using a sparse CS matrix \( \Phi \), with the entries of \( \Phi \) are restricted to \( \{0, 1, -1\} \). Hence the measurements are just sums and differences of small subsets of the coefficients of \( x \). A sparse \( \Phi \) can be represented as a sparse bipartite graph (Figure 2.1). The graph shown in the figure could corresponds to the following \( \Phi \):

\[
\Phi = \begin{pmatrix}
-1 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 1 & 0 & -1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & -1 \\
0 & -1 & 0 & 1 & 0 & 1 & 1
\end{pmatrix}
\]
The design of $\Phi$ (such as column weight, row weight, etc) is based on the properties of the signal as well as the accompanying decoding algorithm. The goal is to construct a decoder (reconstruction algorithm) and $\Phi$ structure for which the decoder consumes minimum number of measurements and operates at low complexity. To achieve this goal, we have developed two accompanying low-complexity reconstruction schemes.

**Decoder for strictly sparse signals:** Our first decoding algorithm appears in [6] and is called *Sudocodes*. This algorithm is based on *graph reduction*, and is tailor-made for strictly sparse signals: signals in $\mathbb{R}^N$ that contain exactly $K \ll N$ non-zero coefficients. The key idea of Sudocodes is that decoding the coefficients of $x$ leads to a series of graph reduction steps. The decoding times of sudo-decoding are comparable to other low-complexity schemes [24, 23, 41]. In contrast, Sudocodes require fewer measurements for perfect reconstruction, compared to the other schemes.

**Decoder for approximately sparse signals:** The highlight of this chapter is a second reconstruction algorithm based on *message passing* over a graph. This algorithm is tailor-made for approximately sparse signals – signals in $\mathbb{R}^N$ that have $K \ll N$ large coefficients and the remaining coefficients are small but non-zero. Examples of such signals are strictly sparse signals in noise and compressible signals.\(^8\)

The decoding approach is based on message passing over graphs to solve a Bayesian inference problem. Using the two-state mixture Gaussian distribution as a prior model for the signal coefficients, we compute an estimate of $x$ that explains the measurements and best matches the prior. We use belief propagation, similar to the decoding in LDPC codes and turbo codes [39, 54, 9]. Our technique for approximately sparse signals has $O(N \log(N))$ encoding complexity and $O(N \log(N))$ worst case decoding complexity. We also show that only $M = O(K \log(N))$ measurements are needed.

\(^8\)A signal is compressible if its coefficients sorted by magnitude obey a power law; it can be approximated well by a sparse signal.
The remainder of this chapter is organized as follows. Our CS-LDPC encoder and decoder are described in Sections 2.2 and 2.3, respectively. Variations and applications are described in Sections 2.4 and Section 2.5, and Section 2.6 concludes. We have also included some details in an appendix.

### 2.2 CS-LDPC encoding

A CS matrix $\Phi$ can be represented as a bipartite graph $G$ (Figure 2.1). The edges of $G$ connect a coefficient node $x(i)$ to a measurement node $y(j)$. The neighbors of a given measurement node is the set of coefficient nodes that are used to generate that measurement. Our techniques rely on the sparse nature of $G$; and so we choose $\Phi$ to be a sparse matrix. Thus, a measurement $y(j)$ is computed using a small subset of the coefficients of $x$. We impose an additional constraint on $\Phi$ to enable even simpler encoding: the entries of $\Phi$ are restricted to $\{0, 1, -1\}$. Hence the computation of $y$ involves only sums and differences of small subsets of the coefficients of $x$. The advantage of representing $\Phi$ as a graph is that the accompanying decoding procedure can employ graph algorithms over $G$; these algorithms can either be a graph reduction or a message passing scheme over the edges of $G$.

In addition to the core structure of the encoding matrix described above, we may introduce other constraints to tailor-make the measurement process to the signal model that generated the input signal. Some of these additional constraints that we use in this chapter are listed below.

1. Each row of $\Phi$ contains exactly $L$ non-zero entries (1 or $-1$) placed randomly.

   The row weight $L$ is chosen based on the properties of the signal that we are trying to encode (such as sparsity), and also the decoding process.
2. In some situations, we fix the column weight for each column of $\Phi$ to be a constant $R$.

3. While sparse $\Phi$ is well suited for sparse signals in the canonical “spike” basis, the encoding can be generalized to the case where $x$ is sparse in an arbitrary basis $\Psi$. In this case, we multiply the encoding matrix $\Phi$ with the sparsifying matrix $\Psi$ and use $\Phi \Psi$ to encode $x$ as $y = (\Phi \Psi)x$.

**Encoding complexity.** Consider the case where the row weight of $\Phi$ is a constant given by $L$. Each measurement requires $O(L)$ additions/subtractions. Therefore, encoding requires $O(ML) = O(N \log(N))$ computations to generate the $M$ measurements. The above computation assumes that the sparsifying basis $\Psi = I$. For the general case $\Psi \neq I$, the encoding matrix $\Phi \Psi$ may not be sparse. Therefore, the encoding must perform extra computations; this extra cost is $O(N^2)$ in general. Fortunately, in many practical situations $\Psi$ is structured (e.g., Fourier or wavelet bases) and thus amenable to fast computation. Therefore, extending our techniques to such bases is feasible.

### 2.3 CS-LDPC decoding of approximately sparse signals

The decoding process for LDPC measurement of strictly sparse signals is provided using our Sudocode approach (Chapter 3). The focus here is on decoding *approximately* sparse signals.

#### 2.3.1 Signal model: approximately sparse signals

We use the two-state Gaussian mixture distribution to model the coefficients of approximately sparse signals [57, 18, 27]. This stochastic model for the signal succinctly captures our prior knowledge about its sparsity. We describe the signal model below.
Let $X = [X(1), X(2), ..., X(N)]$ be a random vector in $\mathbb{R}^N$, and let us consider the signal $x = [x(1), x(2), ..., x(N)]$ as an outcome of $X$. The key observation we exploit is that a sparse signal consists of a small number of large coefficients and a large number of small coefficients. Thus each coefficient of the signal can be associated with a state variable that can take on two values: “high”, corresponding to a coefficient of large magnitude and “low”, representing a coefficient of small magnitude. Let the state associated with the $j$’th coefficient be denoted by $q(j)$, where either $q(j) = 1$ (high) or $q(j) = 0$ (low). We view $q(j)$ as an outcome of the state random variable $Q(j)$ that can take values from $\{0, 1\}$. Let $Q = [Q(1), Q(2), ..., Q(N)]$ be the state random vector associated with the signal. The actual state configuration $q = [q(1), q(2), ..., q(N)] \in \{0, 1\}^N$ is one of $2^N$ possible outcomes of $Q$. We associate with each possible state of coefficient $j$ a probability density for $X(j)$, resulting in a two state mixture distribution for that coefficient. For the “high” state, we choose a high-variance zero mean Gaussian distribution, and for the “low” state, we choose a low-variance zero-mean Gaussian distribution to model the coefficients. Thus the distribution of $X(j)$ conditioned on $Q(j)$ is given by

$$X(j)|Q(j) = 1 \sim \mathcal{N}(t, \sigma_1^2),$$

and

$$X(j)|Q(j) = 0 \sim \mathcal{N}(t, \sigma_0^2),$$

where $\sigma_1^2 > \sigma_0^2$. For simplicity, we assume independence between coefficients: the outcome of the state or value of a coefficient does not influence the state or value of another coefficient. For simplicity, we assume independence between coefficients: the outcome of the state or value of a coefficient does not influence the state or value of another coefficient. Finally, to ensure that we have about $K$ large magnitude coefficients, we choose the probability mass function (pmf) of the state variable $Q(j)$

\[ The model can be extended to capture the dependencies between coefficients, if any. We leave this for future work. \]
to be Bernoulli with \( P [Q(j) = 1] = S \) and \( P [Q(j) = 0] = 1 - S \), where \( S = K/N \) is the sparsity rate (we assume that the sparsity rate is known).

The two-state Gaussian mixture model model is completely characterized by three parameters: the distribution of the state variable (parametrized by the sparsity rate \( S \)) and the variances \( \sigma_1^2 \) and \( \sigma_0^2 \) of the Gaussian pdf’s corresponding to each state. Figure 2.2 provides a pictorial illustration of the two state mixture Gaussian model.

Mixture Gaussian models have been successfully employed in image processing and inference problems because they are simple yet effective in modeling real-world signals [57, 18, 27]. Theoretical connections have also been made between wavelet coefficient mixture models and the fundamental parameters of Besov spaces – function spaces that have proved invaluable for characterizing real-world images [3]. Furthermore, by increasing the number of states and allowing non-zero means, we can make the fit arbitrarily close for densities with with a finite number of discontinuities [71].

Although we study the use of two state mixture Gaussian distributions for modeling the coefficients of the signal, the decoding techniques proposed in the sequel can be used with other prior distributions as well.

2.3.2 Decoding via statistical inference

MMSE and MAP estimation for CS

Decoding for approximately sparse signals is a Bayesian inference problem that can be solved efficiently using a message passing algorithm over factor graphs.

The key idea in Bayesian inference is to determine that signal that is consistent with the observed measurements that best matches our signal model. As before, we observe the linear projections \( y \in \mathbb{R}^M \) of a sparse signal \( x \in \mathbb{R}^N \) given by \( y = \Phi x \),
where $\Phi$ is the compressed sensing matrix of dimension $M \times N$, with $M \ll N$. The goal is to estimate $x$ given $\Phi$ and $y$. Resolving $x$ that satisfy $y = \Phi x$ describes an under-determined set of equations and has infinitely many solutions. The solutions lie along a hyperplane of minimum dimension $N - M$ [7]. We use Bayesian inference to locate the solution in this hyperplane that best matches our prior signal model. We pose the CS reconstruction as the following inference problems (we consider MAP and MMSE estimates):

\[
\hat{x}_{\text{MMSE}} = \arg \min_{x'} E \| X - x' \|^2_2 \quad \text{s.t.} \quad y = \Phi x', \quad \text{and} \\
\hat{x}_{\text{MAP}} = \arg \max_{x'} P[X = x'] \quad \text{s.t.} \quad y = \Phi x',
\]

where the expectation is taken over the prior distribution $P$ of $X$. The MMSE estimate can also be expressed as the conditional mean, given by $\hat{x}_{\text{MMSE}} = E[X|Y = y]$, where $Y$ is the random vector in $\mathbb{R}^M$ corresponding to the measurement. In the sequel, we first present a scheme to determine the exact MMSE estimate of the signal (Section 2.3.2). This scheme has exponential complexity, and hence is impractical for large $N$. However, the sparse structure of $\Phi$ enables us to use low-complexity message-passing schemes to estimate $x$ approximately; this procedure is described in Section 2.3.2.

**Exact solution to CS statistical inference**

To compute the MAP and MMSE estimates of the signal, let us first consider the simple case in which we know the state configuration $q$ of the signal. In this setting, the MMSE estimate of $x$ can be computed using the pseudo-inverse of $\Phi$ after applying the covariance matrix of the associated state configuration (as described below). The covariance matrix $\Sigma_q$ associated with a state configuration $q$ is given by $\Sigma_q = E[XX^T|Q = q]$. The covariance matrix is diagonal (because the coefficients are
independent), with the \( j \)’th diagonal entry equal to \( \sigma_0^2 \) if the state \( q(j) = 0 \), or \( \sigma_1^2 \) if \( q(j) = 1 \). We have the following Theorem:

**Theorem 20** Given the measurements \( y = \Phi x \), then the MMSE and the MAP estimate of \( x \) conditioned on knowing the state configuration \( q \) is given by

\[
\hat{x}_{\text{MAP},q} = \hat{x}_{\text{MMSE},q} = \Sigma_q \Phi^T \left( \Phi \Sigma_q \Phi^T \right)^{-1} y,
\]

where \( \Sigma_q \) is the covariance of \( X \) conditioned on knowing the state configuration \( q \).

**Proof sketch:** Conditioned on knowing the state configuration \( q \), the distribution of \( X \) is a multivariate Gaussian with covariance \( \Sigma_q \). Therefore,

\[
P[X = x|Q = q] = \frac{1}{(2\pi)^{N/2} \det(\Sigma_q)^{1/2}} e^{-\frac{1}{2} x^T \Sigma_q^{-1} x}.
\]

The probability \( P(x) \) is maximized when \( x^T \Sigma_q^{-1} x \) is minimum. Consider the MAP estimate \( \hat{x}_{\text{MAP},q} \): this is given by \( \hat{x}_{\text{MAP},q} = \arg \min_{x'} x'^T \Sigma_q^{-1} x' \) such that \( y = \Phi x' \).

Now consider change of variables, by setting \( \theta = \Sigma_q^{-1/2} x' \). In this setting, we have \( \theta_{\text{MAP}} = \arg \min_{\theta} \theta^T \theta \) such that \( y = \Phi \Sigma_q^{1/2} \theta \). This is a simple least squares problem and the solution can be computed using the pseudo-inverse: \( \theta_{\text{MAP}} = (\Phi \Sigma_q^{1/2})^+ y = \Sigma_q^{1/2} \Phi^T (\Phi \Sigma_q \Phi^T)^{-1} y \). Because \( \hat{x}_{\text{MAP},q} = \Sigma_q^{1/2} \theta_{\text{MAP}} \), we have proved the result for \( \hat{x}_{\text{MAP},q} \). Finally, the MMSE and MAP estimates are identical for multivariate Gaussian random variables, and this proves the Theorem. \( \square \)

We now present a scheme with exponential complexity that computes \( \hat{x}_{\text{MMSE}} \) when the states are unknown. We use this solution as a yardstick to compare the approximate solutions we propose later. To determine \( \hat{x}_{\text{MMSE}} \) accurately, we compute the conditional MMSE estimate for each of the \( 2^N \) state configurations using Theorem 20.
The following Theorem relates the overall MMSE estimate to the conditional MMSE estimates.

**Theorem 21** The MMSE estimate of $x$ when the state configuration is unknown is given by

$$
\hat{x}_{\text{MMSE}} = \sum_{q \in [0,1]^N} P[Y = y|Q = q] \cdot P[Q = q] \cdot \hat{x}_{\text{MMSE},q}.
$$

**Proof sketch:** The expected value of the MMSE error obeys the equalities

$$
\hat{x}_{\text{MMSE}} = E[X|Y = y] = \sum_{q \in [0,1]^N} E[X, Q = q|Y = y]
$$

$$
= \sum_{q \in [0,1]^N} P[Q = q|Y = y] \cdot E[X|Q = q, Y = y]
$$

$$
= \sum_{q \in [0,1]^N} P[Q = q|Y = y] \cdot \hat{x}_{\text{MMSE},q}
$$

$$
= \sum_{q \in [0,1]^N} P[Y = y|Q = q] \cdot P[Q = q] \cdot \hat{x}_{\text{MMSE},q},
$$

which completes the proof.

The signal estimate can be interpreted geometrically. As a simple example, consider a 2D signal ($N = 2$) where each coefficient is modeled by a mixture Gaussian distribution. Figure 2.3 shows a sample sketch of contours of equi-probable values of $x$. This enables to visualize the MAP and MMSE estimates of $x$.

It is well known that exact inference in arbitrary graphical models is an NP hard problem [22]. However, sparse matrices $\Phi$ lend themselves to easy approximate inference using message passing algorithms such as BP. In the sequel, we present a message-passing approach for CS reconstruction, where we compute the approximate marginals for $[X(j)|Y]$ and thus evaluate the MMSE or MAP estimate for each coefficient.
Approximate solution to CS statistical inference via message passing

We now employ belief propagation (BP), an efficient scheme to solve inference problems by message passing over graphical models [56, 47, 37, 88, 48, 54, 26]. We employ BP by message passing over factor graphs. Factor graphs enable fast algorithms to compute global functions of many variables by exploiting the way in which the global function factors into a product of simpler local functions, each of which depends on a subset of the variables [51].

The factor graph shown in Figure 2.4 captures the relationship between the signal coefficients, their states, and the observed CS measurements. The factor graph contains two types of vertices: variable nodes (black) and constraint nodes (white). The factor graph is bipartite: all edges connect a variable node to a constraint node. A constraint node encapsulates the dependencies (constraints) that its neighbors (variable nodes) are subjected to. The factor graph has three types of variable nodes: states $Q(j)$, coefficients $X(j)$ and the measurements $Y(i)$. Furthermore it has three types of constraint nodes. The first type – prior constraint nodes – impose the Bernoulli prior on the state variables. The second type of constraint nodes connect the state variables and the coefficient variables. These constraints impose the conditional distribution of the coefficient value given the state. The third type of constraint nodes connect one measurement variable to a set of coefficient variables that are involved in computing that measurement. In the CS inference problem, the observed variables are the outcomes of the measurement variables $Y$. We employ BP to (approximately) infer the probability distributions (beliefs) of the coefficient and the state variables conditioned on the measurements. In other words, BP provides the mechanism to infer the marginal distributions $P[X(j)|Y = y]$ and $P[Q(j)|Y = y]$. Knowing these distributions, one can extract either the MAP estimate (the value of $x(j)$ that maxi-
mizes $P[X(j)|Y = y])$ or the MMSE estimate (the mean of $P[X(j)|Y = y])$ for each coefficient.

An overview of BP is provided in A. Let us describe some particularly interesting details:

1. There are only 2 types of message processing in CS reconstruction using BP: multiplication of beliefs (at the variable nodes) and convolution (as the constraint node).

2. We need a strategy to encode the beliefs. We propose 2 strategies:
   
   - Approximating the continuous distribution by a mixture Gaussian with a given number of components. We use the parameters of the mixture Gaussian as the messages.
   
   - We sample the pdf’s uniformly and use the samples as the messages. We present results for both schemes and discuss the pros and cons for each.

We could use alternate schemes to encode continuous distributions such as particle filtering, importance sampling and Monte Carlo methods [1]. These directions are left for future work.

**Mixture Gaussian parameters as messages**: In this method, we approximate a distribution by a mixture Gaussian with a maximum number of components. We then use the mixture Gaussian parameters as messages.

For both multiplication and convolution, the resulting number of components in the mixture is multiplicative. To keep the message representation tractable, employ model order reduction on mixture Gaussian distributions. We use the Iterative Pairwise Replacement Algorithm (IPRA) scheme for model order reduction [67].
The advantage of using mixture Gaussian parameters to encode continuous distributions is that the messages are short and hence do not consume large amounts of memory. However, this encoding scheme could be too limiting. The scheme is tailor made to work for mixture Gaussian priors. In particular, this scheme cannot be extended to priors such as $\ell_p$ compressible signals. Also, the model order reduction – a necessary step to keep the representation tractable – introduces errors in the messages, that can affect the quality of the solution as well as impact the convergence of BP [72]. Furthermore, the model reduction algorithms (such as IPRA) used to limit the number of components in the mixture can be computationally expensive.

**Samples of the pdf as messages:** An alternate approach to encode the beliefs is to sample the pdf and send the samples as the messages. In this scheme, multiplication of pdf’s correspond to point-wise multiplication of messages. Convolution of the pdf’s can be computed efficiently in the frequency domain.\(^\text{10}\)

The primary advantage of using pdf samples as messages is that it can be applied to a larger class of prior distributions for the coefficients. Also, both multiplication and convolution of the pdf’s can be computed efficiently. However, the drawback of this scheme is the large memory requirement: we require finer sampling of the pdf for more accurate reconstruction. As a rule of thumb, we sample the pdf’s with a spacing less than $\sigma_0$, the standard deviation of the noise level in the signal. Furthermore, the mere process of sampling the pdf introduces errors in the messages that can impact accuracy and convergence of BP [72].

**Stabilizing BP using damping:** We employ damping using message damped belief propagation (MDBP) [59] to stabilize BP in the face of loopy graphs and inaccurate belief representations. We refer the reader to A for the reasons to use damping.

---

\(^\text{10}\)This approach of computing the FFT of the message to enable fast convolution has been used in LDPC decoding over $GF(2^q)$ using BP [54].
2.3.3 Number of measurements

The number of measurements required for reconstruction of approximately sparse signals with our decoding technique is $O(N \log(N))$ using $L = O(S^{-1})$ where $S = K/N$.

2.3.4 Decoding complexity

The complexity of our decoding technique for approximately sparse signals is $O(N \text{polylog}(N))$ when we use sampled pdf as messages, or $O(m^2 N S \log(S^{-1}) \log N)$ when we use mixture Gaussian parameters as messages, where $m$ is the maximal model order (details below).

First consider the implementation of BP with sampled pdf. Let each message be a vector of $p$ samples. During each iteration, we perform 2 sets of operations: multiplication on the coefficient nodes and convolution on the constraint nodes. For a given coefficient node, an outgoing message is given by (A.1), which can be re-written in the following form to reduce complexity:

$$
\mu_{v \rightarrow c}(v) = \prod_{u \in n(v) \setminus \{c\}} \mu_{u \rightarrow v}(v)
= \frac{\prod_{u \in n(v)} \mu_{u \rightarrow v}(v)}{\mu_{c \rightarrow v}(v)},
$$

as long as the condition that the denominator contains no zeros in the beliefs is satisfied. This condition is satisfies in our setting because a mixture Gaussian distribution has non-zero pdf in the real number domain. The modified form of message computation given in Equation (2.3) makes computation simple, because the numerator can be computed once and can be reused for all the message outgoing from the variable node $v$. In a similar manner, the messages outgoing from a constraint node
(Equation (A.2)), can be re-written as:

\[
\mu_{c \rightarrow v}(v) = \sum_{\sim\{v\}} \left( c(n(c)) \prod_{w \in n(c) \setminus \{v\}} \mu_{w \rightarrow c}(w) \right) \\
= \sum_{\sim\{v\}} \left( c(n(c)) \prod_{w \in n(c)} \frac{\mu_{w \rightarrow c}(w)}{\mu_{c \rightarrow v}(v)} \right).
\]

The computational complexity for the message processing at a variable node for one iteration is \(O(Rp)\), because we are multiplying \(R + 1\) vectors of length \(p\). For all \(N\) nodes, the combined complexity per iteration is \(O(NRp)\). Now consider a constraint node associated with the measurements. We perform the convolution operation in the DFT domain, so the computational cost per node is \(O(Lp + Lp \log p) = O(Lp \log p)\), to account for the FFT and the multiplications. Therefore, the combined complexity per iteration for all the \(M\) constraint nodes is \(MLp \log p\). Thus the total cost per iteration is \(O(NRp + MLp \log p) = O(MLp \log p)\), because \(NR = ML\). For \(\log N\) iterations, we get the overall complexity of \(O(MLp \log p \log N)\). As a rule of thumb, we choose \(p = \sigma_1 / \sigma_0\) and hence the overall complexity can be expressed as \(O(ML^{2\sigma_1} \log \sigma_1 \log \sigma_0 \log N)\), which is \(O(N\text{polylog}(N))\).

Now let us consider the BP implementation where the messages are mixture Gaussian parameters. In this scenario, we cannot use the modified Equations (2.3) and (2.4) because it is impossible to undo the effect of multiplication. Furthermore, we perform component-reduction for products and convolutions of mixtures (using schemes like IPRA [67]) to make the representation tractable. The component-reduction makes it impossible to undo the effect of multiplication. Let \(m\) be the maximum model-order that we deal with. The complexity per coefficient node per iteration is given by \(O(m^2 R^2)\). For all \(N\) nodes per iteration, the complexity is \(O(m^2 R^2 N)\). Similarly, the complexity for all \(M\) constraint nodes associated with the measurement is given by \(O(m^2 L^2 M)\). Thus the overall complexity for \(\log N\) iterations
is $O(m^2LM(R+L)\log N)$. Substituting $L = O(S^{-1}) = O(N/K)$, $R = O(\log(N/K))$, and $M = O(K\log(N/K))$, the overall complexity is $O(m^2\frac{N}{S}\log(S^{-1})\log N)$.

### 2.3.5 Storage requirements

The bulk of the storage is for message representations for each edge. The number of edges is given by $ML = NR = O(N\log(N/K))$. For implementation of BP with pdf samples, the message length is $p$, and so the storage requirement is $O(MLp)$. For implementation of BP with mixture Gaussian parameters, the message length is $m$, and so the storage requirement is $O(MLm) = O(mN\log(N/K))$.

### 2.3.6 Properties of BP decoding for CS-LDPC

We now briefly describe several properties of our BP decoding approach for CS-LDPC measurements:

1. **Progressive reconstruction**: Whatever the value of $M$, we can always find the posterior probabilities.

2. **Information scalability**: If we are interested only in the states of the coefficients but not their value, i.e., estimating the support set, we would need less measurements.

3. **Robustness to noise**: We can incorporate noisy measurements into our model. This could add an additional component to the Gaussian mixtures of our current signal model.
2.3.7 Numerical results

We study the performance of BP for CS reconstruction for several levels of sparsity and $\sigma_1/\sigma_0$ ratios. Figure 2.5 shows a sample reconstruction of a sparse signal using BP.

Figure 2.6 shows sample simulation results depicting the reconstruction quality for $N = 6000$, $\sigma_1/\sigma_0 = 30$ for different values of $K$ and $L$. Note that as we increase $L$, we require less measurements to obtain the same accuracy. However, there is an optimal value of $L$ ($L_{\text{OPT}}$) beyond which there is the additional gain in performance is marginal. Furthermore, values of $L > L_{\text{OPT}}$ gives rise to divergence in BP, even with damping.

2.4 Variations and enhancements

While we present our techniques for sparse signals in the canonical “spike” basis, they can be generalized to the case where $x$ is sparse in an arbitrary basis $\Psi$. In this case, we multiply the LDPC matrix $\Phi$ with the sparsifying matrix $\Psi$ and use $\Phi \Psi$ to encode $x$ as $y = (\Phi \Psi)x$. The decoder uses $y$ to determine the signal represented in the sparsifying basis (namely $\Psi x$), and $x$ is then recovered by multiplying by $\Psi^T$. The encoding matrix $\Phi \Psi$ may not be sparse. Therefore, the encoding and decoding must perform extra computations; this extra cost is $O(N^2)$ in general. Fortunately, in many practical situations $\Psi$ is structured (e.g., Fourier or wavelet bases) and thus amenable to fast computation. Therefore, extending our techniques to such bases is feasible.

Exploiting statistical dependencies: In many signal representations, the coefficients of the representation are not statistically independent but have correlations. For example, natural images that are sparsely represented in a wavelet basis have
significant coefficients that can be arranged on a tree structure, and there is strong
correlation between the magnitudes of coefficients that are arranged on parent and
child nodes. It is possible to decode signals from fewer measurements via an ex-
tended algorithm that allocates different probabilities of different coefficients being
non-zero. For example, the wavelet coefficients of the low frequency components have
a higher probability of appearing in the sparse representation than coefficients for
high-frequency components [34]. The LDPC measurement technique can compen-
sate for these different probabilities by allocating different probabilities that these
coefficients are measured in any given row.

Feedback: Feedback from the decoder to the encoder can allow the measure-
ments to be adapted according to the coefficients that have been reconstructed thus
far. Consider the Sudocode scheme for strictly sparse signals as an example [6].
Notions reminiscent of closed-loop iterative doping [12] can be utilized here. The
Sudocode encoder knows what it has transmitted. Consequently, the encoder can
track the decoder’s progress and send the decoder “hints” regarding problematic sig-
nal coefficients. These hints will cause many other unknown coefficient values to be
resolved.

2.5 Applications

Compressive signal acquisition: The obvious application of our ideas is to sense
signals in an implicitly compressive manner using a reduced number of measurements.
This approach can be especially appealing on the encoding side, because the analog
measurement process can be run much slower than in a traditional signal acquisition
system.
Partial reconstruction: Our decoding scheme resolves the signal several coefficients at a time. This property can be used in applications that look for information in parts of the signal, where revelation of the entire signal is not necessary. For example, problems in detection do not require complete signal revelation [33].

Lossy channels: In some applications, measurements may be lost because of transmissions over faulty channels. In such a setup, one can stream measurements until the decoder has sufficiently many to reconstruct the signal. The decoder can notify the encoder that the reconstruction is complete.

p2p and distributed file storage applications: Our techniques can be used in distributed settings where multiple encoders send measurements to a single reconstruction center. Consider peer-to-peer (p2p) applications for example. A plurality of p2p sources can all stream digital content such as video, audio, images, etc. These data are typically in compressed form and thus strictly sparse. Otherwise, we enforce strict sparsity by thresholding the transform coefficients. When an individual user wishes to download some content, these sources transmit LDPC measurements towards the user. Once the user has received enough measurements, the original signal can be reconstructed. This strategy does not require complicated synchronization of the transmissions arriving from the large number of servers. All that is required for reconstruction is that enough measurements arrive at the decoder, no matter from which transmitter. The decoding algorithm must also know the seeds of all the pseudo-random number generators in all the encoders. This is reminiscent of a digital fountain [11] but we use real valued coefficients and strictly sparse signals.
2.6 Conclusions

The extension of LDPC matrices from a finite alphabet setting to measurement of real-valued inputs is very promising, and leads to effective encoding and decoding schemes. We employed belief propagation techniques to reconstruct approximately sparse signals. A comparison of our results to several CS schemes proposed in the literature appears in Table 2.1. Our techniques can be extended to sensing and reconstructing signals that are sparse in other bases, additional signal models, and also handle noise.
Figure 2.2 Illustration of the 2-state mixture Gaussian model for the Random Variable $X$. The distribution of $X$ conditioned on the state variable $Q$ are depicted. Also shown is the overall non-Gaussian distribution for $X$. We use this mixture distribution to model the prior for the coefficients of a sparse signal.

Figure 2.3 Equi-probable surfaces using the mixture Gaussian model for the coefficients. The line represents the set of $x$’s that satisfy $\Phi x = y$. As we expand the equi-probable surface, it touches the line at the MAP estimate.
Figure 2.4  Factor graph depicting the relationship between the variables involved in our CS problem. The variable nodes are depicted in black and the constraint nodes in white.

Figure 2.5  Example reconstruction for $N = 600$, $K = 18$, $M = 180$, $L = 15$. 
Figure 2.6  Reconstruction versus the number of measurements, for different values of $L$ using belief propagation.
Chapter 3

Sudocodes: Fast Reconstruction for Strictly Sparse Signals

3.1 Introduction

In this chapter, we introduce sudocodes, an efficient lossless measurement and reconstruction scheme for compressive sampling (CS) of sparse signals. Consider a discrete-time signal \( x \in \mathbb{R}^N \) that contains only \( K \) non-zero values. The core tenet of CS is that it is unnecessary to measure all the \( N \) values of \( x \); rather, we can recover \( x \) from a small number of projections onto an incoherent basis [13, 28]. To measure \( x \), we compute the measurement vector \( y \in \mathbb{R}^M \) as \( M \ll N \) linear projections of \( x \) via the matrix-vector multiplication \( y = \Phi x \). Candès et al. [13] and Donoho [28] proposed a measurement strategy using independent and identically distributed (iid) Gaussian entries for \( \Phi \) and a tractable reconstruction procedure based on linear programming. The reconstruction based on linear programming requires \( M = cK \) projections for signal reconstruction, where \( c \approx \log_2(1+N/K) \) [7], and has computational complexity at least \( O(N^3) \) [28].

While linear programming techniques figure prominently in the design of practical CS reconstruction schemes, their \( O(N^3) \) complexity renders them impractical for many applications. We often encounter sparse signals with large \( N \); for example, current digital cameras acquire images with the number of pixels \( N \) of the order of \( 10^6 \) or more. For such applications, the need for faster reconstruction algorithms is
critical. Furthermore, the measurement process also has high complexity; encoding with a full Gaussian or Bernoulli $\Phi$ requires $O(MN)$ computations.

In this chapter, we develop a framework we term *sudocoding* to measure (or encode) the signal $x$ so that the reconstruction (or decoding) of $x$ given $y$ and $\Phi$ has low computational complexity. Sudocodes are applicable to strictly sparse signals whose non-zero values satisfy certain simple conditions and can be extended to signals that are sparse in arbitrary bases.

The key idea behind sudocoding is to construct measurements by summing subsets of the coefficient values $x(i)$ akin to group testing [32]. We consider subsets comprising $L$ coefficients of $x$ at a time. In other words, the entries of $\Phi$ are either 0 or 1, with 1 appearing $L$ times in each row of $\Phi$ (sudo-decoding is robust to a relatively large range of $L$). The sparse structure of $\Phi$ lends itself to fast measurement and reconstruction algorithms. The sudo-decoder receives a stream of measurements and the corresponding rows of $\Phi$ and performs reconstruction.\[11\]

The name “sudocode” is inspired by Sudoku puzzles, where the challenge is to determine unknown values of cells in a grid [2];\[12\] our sudo-decoder works in a similar manner. Decoding is based on the key insight that both zero measurements and matching measurements can be used to infer coefficient values. As in group testing [32], if the decoder receives a zero measurement, then it knows that all coefficients that were involved in generating the measurement are zero. Sudocodes extend group testing by also resolving coefficient values when the decoder locates matching measurements. We use a Binary Search Tree (BST) data structure to store the measurements to perform search operations in logarithmic time. We show that sudocodes require

\[11\] Pseudo-random construction of $\Phi$ obviates the need to transmit the rows of $\Phi$; the decoder can easily reproduce $\Phi$ using the same seed.

\[12\] Thanks to Ingrid Daubechies for pointing out the connection.
$M = O(K \log(N))$ measurements for perfect reconstruction with high probability and that the worst-case decoding complexity is $O(K \log(K) \log(N))$. Sudocodes thus substantially expand the horizon for CS applications. We present simulation results for problem sizes of practical interest and show that the number of measurements and decoding times outperform other recently proposed schemes (Table 3.1).

One of the highlights of sudocoding is that it employs an avalanche strategy to accelerate the decoding process. The inference of a coefficient value triggers an avalanche of coefficient revelations and thus enables the signal to be recovered using far fewer measurements.

Sudocodes are applicable beyond data acquisition systems. They can be applied as a sort of erasure code [10] for data in $\mathbb{R}^N$. We can use this property in distributed settings where multiple encoders send measurements to a single reconstruction center. Once enough measurements have been received at the reconstruction center, the original signal can be recovered.

<table>
<thead>
<tr>
<th>Table 3.1</th>
<th>Comparison of Sudocodes with Chaining Pursuit [41].13</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 10000$</td>
<td>$M = 5915$</td>
</tr>
<tr>
<td>$K = 10$</td>
<td>$T = 0.16s$</td>
</tr>
<tr>
<td>$N = 10000$</td>
<td>$M = 90013$</td>
</tr>
<tr>
<td>$K = 100$</td>
<td>$T = 2.43s$</td>
</tr>
<tr>
<td>$N = 100000$</td>
<td>$M = 17398$</td>
</tr>
<tr>
<td>$K = 10$</td>
<td>$T = 1.13s$</td>
</tr>
<tr>
<td>$N = 100000$</td>
<td>$M &gt; 10^b$</td>
</tr>
<tr>
<td>$K = 1000$</td>
<td>$T &gt; 30s$</td>
</tr>
</tbody>
</table>

13The decoding time $T$ is based on simulations in Matlab 7.1.0 running on a Linux platform with a 2.2GHz AMD Athlon 64X2 Dual Core Processor 4400+ and 2GB memory. The value of $M$ shown for sudocodes includes measurements in both decoding phases.
While we present sudocodes for sparse signals in the canonical “spike” basis, they can be generalized to the case where $x$ is sparse in an arbitrary basis $\Psi$. In this case, we multiply the sudocode matrix $\Phi$ with the sparsifying matrix $\Psi$ and use $\Phi \Psi$ to encode $x$ as $y = (\Phi \Psi)x$. The decoder uses $y$ to determine the signal represented in the sparsifying basis (namely $\Psi x$), and $x$ is then recovered by multiplying by $\Psi^T$. The encoding matrix $\Phi \Psi$ may not be sparse. Therefore, the encoding and decoding must perform extra computations; this extra cost is $O(N^2)$ in general. Fortunately, in many practical situations $\Psi$ is structured (e.g., Fourier or wavelet bases) and thus amenable to fast computation. Therefore, extending sudocodes to such bases is feasible.

3.2 Related work

Encoding and decoding of sparse signals in $\mathbb{R}^N$ has been addressed in the compressed sensing (CS) literature. The practical revelation that supports CS is that the much easier problem of $\ell_1$ minimization $\hat{x} = \arg \min \|x\|_1$ s.t. $y = \Phi x$ can be used to reconstruct $x$ [13, 28]. This optimization problem can be solved with linear programming techniques. Yet, the $O(N^3)$ complexity of these techniques is impractical for many applications.

At the expense of slightly more measurements, iterative greedy algorithms have been developed. Examples include the iterative Orthogonal Matching Pursuit (OMP) [83], Matching Pursuit (MP), and Tree Matching Pursuit (TMP) [34] algorithms. In CS applications, OMP requires $c \approx 2 \ln(N)$ [83] to succeed with high probability; decoding complexity is $O(NK^2)$. Unfortunately, $O(NK^2)$ is cubic in $N$ and $K$, and therefore OMP is also impractical for large $N$.

Cormode and Muthukrishnan recently proposed a fast CS algorithm based on group testing [23]. Their scheme considers subsets of the signal coefficients in which we expect at most one large coefficient to lie. Within this set, they locate the position
and value of the large coefficient using a Hamming code construction. Their decoding scheme has low $O(K \log^2(N))$ complexity, but they require $M = O(K \log^2(N))$ measurements.

Gilbert et al. [41] propose the Chaining Pursuit (CP) algorithm for reconstructing $K$-sparse signals. Unlike sudocodes, CP can be applied to approximately sparse and compressible signals as well.\(^{14}\) CP reconstructs with $O(K \log^2(N))$ non-adaptive linear measurements in $O(K \log^2(N) \log^2(K))$ time. CP works best for “super-sparse” signals, where the ratio $K/N$ is very small (see Table 3.1). Define the sparsity rate as the fraction of non-zero coefficients in $x$: $S \triangleq K/N$. As we increase $S$, CP requires an enormous number of measurements; for some ranges of $S$ the number of measurements $M > N$, which is undesirable. In contrast, sudocodes require only $M = O(K \log(N))$ measurements and offer $O(K \log(K) \log(N))$ worst-case computational complexity for a wide range of sparsity rates $S$.

### 3.3 Sudocodes

#### 3.3.1 Setup

The sudocode scheme presented in this chapter is for strictly sparse signals. Without loss of generality, we make the following mild assumption. Let $\Omega \triangleq \{i : i \in [1, \ldots, N], x_i \neq 0\}$ be the support set for the non-zero entries of $x$ and denote its cardinality $|\Omega|$ by $K$. We assume that for any two subsets $\Omega_1, \Omega_2 \subseteq \Omega$ with $\Omega_1 \neq \Omega_2$, the following condition holds: $|\sum_{i \in \Omega_1} x_i - \sum_{i \in \Omega_2} x_i| > \epsilon$; that is, we assume that the sum of $x(i)$’s corresponding to each subset of $\Omega$ is unique up to a given precision $\epsilon$. This assumption is valid with high probability if the non-zero signal coefficients are drawn from a continuous random distribution such as uniform or Gaussian. For

\(^{14}\)A signal is compressible if its coefficients sorted by magnitude obey a power law; it can be approximated well by a sparse signal.
signals that do not conform to the above condition, we add pseudo-random perturbations to the non-zero coefficients to make it so. The decoder can then regenerate the pseudo-random perturbations using the same seed and subtract them off after reconstruction. Extensions of sudocodes to other signal classes such as sparse signals in noise and compressible signals are part of our ongoing work (see Section 3.6).

### 3.3.2 Sudo-encoding

A sudo-encoding matrix $\Phi$ is sparse with entries 0 or 1. Each row of $\Phi$ contains exactly $L$ ones placed randomly, where $L$ is a constant chosen based on the values of $N$ and $K$. We show below that $L$ proportional to $N/K$ yields good performance (Section 3.4). The encoder constructs the measurement vector $y$ via the matrix-vector multiplication $y = \Phi x$, which merely sums different sets of $L$ coefficients of $x$.

### 3.3.3 Sudo-decoding

The decoding scheme is iterative; we process each measurement $y(j)$ in succession, $j = 1, 2, \ldots, M$. If we have identified some of the values of the $L$ coefficients that make up measurement $y(j)$ from previous iterations, then we account for those values by subtracting them from $y(j)$ and then considering the modified measurement, which we call $y^*(j)$. Effectively, we are decreasing the size of the problem. The modified measurement captures the unexplained portion of the original measurement $y(j)$.

Now, $y^*(j)$ is a sum of $L^*(j)$ coefficients of $x$, where $L^*(j) \leq L$. The decoding process proceeds based on whether $y^*(j)$ is zero or non-zero.

**Case 1:** $y^*(j) = 0$: We conclude that the corresponding values of the $L^*(j)$ coefficients of $x$ are all zero. This follows from our uniqueness assumption on the non-zero coefficients of $x$.

**Case 2:** $y^*(j) \neq 0$: We compare $y^*(j)$ to the past measurements (i.e., $y^*(k)$, $k < j$). If there is a perfect match (within the precision $\epsilon$) between $y^*(j)$ and a past
measurement then we infer that the two measurements originate from the same set of non-zero coefficients of $x$. Therefore the indices present in one set but not in the other set have zero coefficient values. Furthermore, we also determine the coefficient value of a non-zero coefficient if the intersection of the two index sets contains only one element; this occurs with overwhelmingly high probability as we increase $N$.  

**Accelerated decoding**: In order to search the past measurements for matches efficiently, we store the past measurements in a Binary Search Tree (BST) data structure. For a BST containing $n$ elements, the search, insert, delete, and update operations require $O(\log(n))$ time on average. We also introduce the history matrix $H$ of size $R \times N$, where $R = O(ML/N)$. The $i$'th column of $H$ stores the list of row numbers of $\Phi$ that measured $x(i)$ up to the current value of $j$. The history matrix makes it easier to update the measurements as the coefficients of $x$ are recovered. When $x(i) \neq 0$ is resolved, we subtract off its value from the previous measurements to which it contributed.

**Avalanche**: An avalanche strategy further accelerates the decoding process and also reduces the number of measurements. Whenever a coefficient is resolved, we search the past measurements that measured the coefficient value (using the history matrix) to induce further coefficient revelations. This triggers an avalanche of coefficient revelations, and thus the signal can be recovered using fewer measurements.

**Two phase coding**: It is useful to break the encoding and decoding processes into two phases. First consider encoding. Phase 1 encoding proceeds as above by computing sums of sets of $L$ coefficients of $x$. Phase 2 measurements are encoded differently: we compute $\tilde{K}$ measurements of $x$ using a non-sparse $\tilde{K} \times N$ matrix $\tilde{\Phi}$. The only requirement for $\tilde{\Phi}$ is that every submatrix formed by taking $\tilde{K}$ columns from $\tilde{\Phi}$ is invertible (the value of $\tilde{K}$ will be described soon). Now consider decoding. In Phase 1 decoding, we proceed as above until we have reconstructed most of the coefficients
in $x$. When at least $N - \tilde{K}$ coefficient values have been determined, we transition to Phase 2 decoding. In Phase 2 decoding, we have at most $\tilde{K}$ remaining unknown coefficients. We extract the columns of $\Phi$ corresponding to the unresolved coefficients to form a $\tilde{K} \times \tilde{K}$ matrix $\tilde{\Phi}_K$. The decoder solves for the unknown coefficients by computing the matrix inverse of $\tilde{\Phi}_K$ and multiplying it with the Phase 2 measurements. Since matrix inversion has cubic complexity, we select $\tilde{K}$ proportional to $K^{1/3}$, thereby requiring $O(K)$ complexity for Phase 2 decoding.

The reason for introducing Phase 2 is as follows. As we decode more coefficients of $x$ in Phase 1, we see more frequent occurrences of measurements that provide no new information; this occurs when all of the indices corresponding to a row of $\Phi$ are already known. Phase 2 en/decoding is introduced before the number of such measurement explodes.

### 3.4 Analysis of Sudo-Decoding

#### 3.4.1 Choice of $L$

The performance of sudo-decoding (the number of measurements $M$ as well as the encoding and decoding time) depends on the choice of $L$. The optimal $L$ depends on the probability that we capture all zero coefficients in a set of size $L$. For large $N$, this probability depends only on the sparsity rate $S = K/N$. Figure 3.1 shows the number of measurements needed ($M$) as a function of $L$. This plot is based on simulations for $N = 16000, 64000, \text{and } 256000$ and sparsity rate $S = 0.02$. Figure 3.2 plots the optimal $L$ (which requires the least number of measurements) for a variety of sparsity rates.
3.4.2 Number of measurements $M$

The following theorem characterizes the trade-off between $M$ and the fraction of coefficients recovered.

**Theorem 22** Using $L = O(N/K)$, Phase 1 sudo-decoding requires

$$M = O(K \log(1/\epsilon))$$

measurements for exact reconstruction of $N(1 - \epsilon)$ coefficients.

**Proof:** Let $L = c_1/S = c_1N/K$. Consider a single measurement. We have

$$\Pr(\text{zero measurement}) = f_1(c_1) + o(1),$$

where the $o(1)$ term vanishes as $N$ increases and the binomial distribution is approximated as Poisson. Similarly, the probability that a single coefficient was measured satisfies

$$\Pr(\text{single coefficient}) = f_2(c_1) + o(1).$$

Let Phase 1 take $M = c_2K$ measurements in total. In the remainder of the proof we ignore the $o(1)$ terms, because they are easily accounted for by modifying $M$ slightly. Phase 1 encounters a zero measurement $f_1(c_1)c_2K$ times; each time $L = c_1N/K$ zero coefficients are determined, for a total of $c_1c_2f_1(c_1)N$ zero coefficients. However, the same zero coefficient may have been determined multiple times. Because there are $N - K$ zero coefficients in total, each was determined $c_1c_2f_1(c_1)N/(N - K) > c_1c_2f_1(c_1)$ times. Therefore, the probability that any specific zero coefficient was determined is greater than

$$f_3(c_1c_2f_1(c_1))$$

where the Poisson approximation gives

$$f_3(\theta) = 1 - e^{-\theta} + o(1).$$

We now analyze the total number of non-zero coefficients recovered in Phase 1. The total number of times that a single coefficient was measured is $c_2f_2(c_1)K$. If the same non-zero coefficient is measured multiple times without other coefficients interfering, then that coefficient is recovered by Phase 1. This happens with probability

$$f_4(c_2f_2(c_1)),$$

where the Poisson approximation gives

$$f_4(\theta) = 1 - e^{-\theta}(1 + \theta) + o(1).$$
We complete the proof by noting that $c_1$, $f_1(c_1)$, and $f_2(c_1)$ are constants, and so the proportion of coefficients that are not recovered decays as $e^{-O(c_2)}$. \hfill \Box$

Phase 1 must cover all but the last $O(K^{1/3})$ coefficients, which corresponds to $\epsilon = O(K^{1/3}/N)$. Applying Theorem 22, we have $M = O(K \log(N/K^{1/3})) = O(K \log(N))$. We express this in the following.

**Corollary 1** Using $L = O(N/K)$, Phase 1 sudo-decoding requires $M = O(K \log(N))$ measurements for exact reconstruction of $N - \tilde{K}$ coefficients, where $\tilde{K} = O(K^{1/3})$.

Finally, Phase 2 requires $O(K^{1/3})$ measurements, which is much less than the $O(K \log(N))$ measurements required for Phase 1. Therefore, the total number of measurements required for exact reconstruction by sudo-decoding is $M = O(K \log(N))$.

### 3.4.3 Computational complexity

**Encoding complexity.** Each measurement requires $O(L)$ additions. Therefore, sudo-encoding requires $O(ML) = O(N \log(N))$ computations in Phase 1. In Phase 2, sudo-encoding requires $O(\tilde{K}N)$ complexity in general; however special measurements such as DCT or wavelets can be used to reduce the complexity to $O(N \log(N))$ or $O(N)$.

**Decoding complexity.** Sudo-decoding can be implemented efficiently using the following data structures. The measurement matrix $\Phi$ must be stored sparsely. By storing the indices of the $L$ coefficients measured by each row, we can efficiently perform operations such as comparing the support sets of two rows when two non-zero measurements are found to be equal. Using the BST data structure to store past measurements, we can search for matching measurements with $O(\log(M))$ complexity. Using the history matrix $H$ makes it easier to update the measurements as the coefficients of $x$ are recovered, as described in Section 3.3.3. Every search for
past identical measurements using the BST costs $O(\log(M))$. Because the aggregate number of accesses to the heap is $O(M)$, the aggregate decoding complexity is $O(M \log(M)) = O(K \log(N) \log(K \log(N))) = O(K \log(K) \log(N))$.\footnote{We assume that $K$ grows faster than $\log(N)$. Otherwise, the decoding complexity is $O(K \log(N) \log \log(N))$.}

Phase 2 inverts a matrix of size $\tilde{K} \times \tilde{K}$, and so its complexity is $O(\tilde{K}^{1/3}) = O(K)$, which is smaller than the computation in Phase 1. We summarize the discussion with the following theorem.

**Theorem 23** The computation complexity of the sudo-encoder is $O(N \log(N))$. The computation complexity of the sudo-decoder is $O(K \log(K) \log(N))$.

### 3.5 Numerical results

The decoding performance of sudocodes (number of measurements $M$ and decoding time $T$) is presented in Table 3.1 for a range of $N$ and $K$. For comparison, the corresponding performance of Chaining Pursuit (CP) [41] is also shown. Sudocodes

![Figure 3.1](image)

**Figure 3.1** Over sampling factor vs. $L$ for sparsity rate $S = 0.02$. There is an optimal choice of $L$ that minimizes $M$.\footnote{We assume that $K$ grows faster than $\log(N)$. Otherwise, the decoding complexity is $O(K \log(N) \log \log(N))$.}
Figure 3.2 Optimal of $L$ for different sparsity rates. The optimal $L$ is independent of $N$ and is $O(1/S)$, where $S = K/N$.

perform very well in all sparsity rate regimes shown in the Table. CP is well suited for “super-sparse” (very small $S = K/N$) signals. For moderately sparse signals, CP requires a very large number of measurements; in particular, $M > N$ in some simulations.

3.6 Discussion

Sudocodes could lend themselves to a plethora of applications. Since sudocodes resolve several signal coefficients with each measurement they can be used in applications where partial reconstruction of a signal is sufficient (e.g., in detection). Sudocodes can be used as erasure codes for signals in p2p and distributed file storage applications. For example, multiple p2p sources can stream compressed digital content such as video, audio, images, etc. When an individual user wishes to download the content, the sources take sudocode measurements and transmit them toward the user. Once the user has received enough measurements, the original signal can be reconstructed. The thresholded DCT/wavelet coefficients used in JPEG/JPEG2000 compression are excellent candidates for sudocoding.
Sudocodes can be enhanced in a variety of ways. Sudocodes can be made to leverage the *statistical dependencies* between the non-zero coefficients in the signal. For example, natural images that are sparsely represented in a wavelet basis exhibit strong correlation between the parent and the child nodes [34]. Sudocodes can be enhanced to provide resiliency against *unknown sparsity* order $K$. If the encoder does not have the knowledge of $K$, then it can begin with a small estimate for $K$ and then refine the estimate as it transmits measurements. The number of ones $L$ in each row can be set based on the best estimate of $K$. *Feedback* from the decoder to the encoder can allow the measurements to be adapted to the coefficients that remain unresolved at the decoder.

Finally, modifications can be made to make sudocodes robust to the presence of noise in the signal or the measurements. We are investigating the use of message passing algorithms like belief propagation [88] in this scenario.
Chapter 4

Compressed Sensing Rate Distortion

4.1 Introduction

Consider a discrete-time real-valued signal $X$ of length $n$ that has only $k$ non-zero coefficients for some $k \ll n$. The core tenet of Compressed Sensing (CS) is that it is unnecessary to measure all $n$ values of the sparse signal; rather, we can recover the signal using a small number of linear projections onto an incoherent basis [13, 28]. To measure (encode) $X$, we compute the ideal (noiseless) measurement vector $Y_0 \in \mathbb{R}^m$ as $m$ linear projections of $X$ via the matrix-vector multiplication $Y_0 = \Phi X$. The goal in CS is to reconstruct (decode) $X$ — either accurately or approximately — given the measurements.

CS reconstruction can be performed with $O(n^3)$ computation via $\ell_1$ minimization by applying linear programming techniques.\footnote{For two functions $f(n)$ and $g(n)$, $f(n) = O(g(n))$ if $\exists c, n_0 \in \mathbb{R}^+, \forall n > n_0$, $f(n) \leq cn$ for all $n > n_0$. Similarly, $f(n) = o(g(n))$ if for all $c > 0$, $\exists n_0 \in \mathbb{R}^+$, $f(n) \leq cn$ for all $n > n_0$, and $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$. Finally, $f(n) = \Theta(g(n))$ if $\exists c_1, c_2, n_0 \in \mathbb{R}^+, \forall n > n_0$, $c_1 \frac{g(n)}{n} \leq f(n) \leq c_2 \frac{g(n)}{n}$ for all $n > n_0$.} This approach requires approximately $k \log(n/k)$ measurements [28, 13, 7], where we use the base-two logarithm.

The CS community has also studied acquisition of signals that are not $k$-sparse but compressible, meaning that their coefficient magnitudes decay according to a power law. In this case too there are polynomial complexity algorithms that achieve $\|\hat{X} - X\|_2^2 \leq c_1 \sigma_k^2$, where the operator $\| \cdot \|_2^2$ denotes the squared $\ell_2$ norm, $\sigma_k^2$ is the squared $\ell_2$ error in the best $k$-term approximation to the signal $X$, and $c_1$ is a
constant. For compressible signals that lie in an $\ell_1$ ball, these algorithms require $m = O(k \log(n/k))$ measurements [13]. Additionally, it has been shown that at least $m = \Omega(k \log(n/k))$ measurements are required in this case [42, 43, 50].

In order to reduce resource consumption, a problem of considerable interest is to seek practical CS measurement and reconstruction schemes that require fewer measurements. Indeed, as Donoho wrote [28], “Why go to so much effort to acquire all the data when most of what we get will be thrown away?”

Recalling that $m = \Omega(k \log(n/k))$ measurements are required for compressible signals, let us consider more restrictive signal classes. It has been shown that we cannot hope to perform CS reconstruction of a $k$-sparse signal using fewer than $m = k + 1$ measurements [7, 84]. However, the only approach for CS reconstruction using $m = k + 1$ is via $\ell_0$ minimization, which is known to be NP-complete and therefore impractical [15]. This raises the question: can we construct practical CS schemes that require $m = o(k \log(n/k))$ measurements for $k$-sparse signals?

In addition to computational issues and appropriate signal classes, any performance evaluation of CS reconstruction techniques must account for imprecisions in the measurement process. We emphasize that all analog measurement systems are imperfect and add various artifacts. Furthermore, any CS hardware system that relies on analog-to-digital conversion will contain quantization noise, which must also be accounted for in the performance analysis.

In this chapter we show that the performance limits of CS reconstruction systems subject to additive white Gaussian noise obey

$$\delta \geq \frac{2R(\mathcal{E}(D_X))}{\log(1 + \text{SNR})},$$

where $\delta = m/n$ is the measurement rate, $R(\cdot)$ is the rate-distortion function of the signal source, $\mathcal{E}(D_X)$ is the distortion level using decoder $D_X$ to reconstruct $X$, and SNR
is the measurement signal-to-noise ratio (details in the sequel). We show that each CS measurement is similar to one usage of a communication channel and contributes $\frac{1}{2} \log(1 + \text{SNR})$ bits toward the resolution of the signal. The revelation here is the direct relationship in this noisy setting between the measurement rate required and the rate-distortion function of the signal source being acquired. We apply information theoretic tools such as channel coding and rate-distortion to illuminate these points. These information theoretic tools shed new light on CS techniques, enabling us to realize the crucial significance of rate-distortion tradeoffs and measurement noise.

To illustrate this principle, consider an observed measurement vector $Y$ that incorporates additive white Gaussian noise $Z$ such that $Y = Y_0 + Z$. We will derive a lower bound (a converse result) on the CS measurement rate as a function of the squared reconstruction error and the measurement signal-to-noise ratio. This bound dictates the minimum reconstruction distortion that the user must contend with in the face of Gaussian measurement noise. To derive our result, we model the measurement process as ideal measurements passing through a noisy stochastic channel. Because the capacity of this measurement channel is finite, perfect signal recovery is impossible. The lower bound is obtained by relating the capacity of the measurement channel to the rate-distortion function of the input signal.

One of the grand successes of information theory has been the rise of low-complexity capacity-achieving channel codes [9, 63]. Our broader vision is to develop similar CS reconstruction algorithms. Indeed, our recent work on sparse signal acquisition and recovery using LDPC measurement matrices [66], which borrows ideas from the channel coding community [39, 9, 63], indicates that reconstruction of $k$-sparse signals, which have $\Theta(k \log(n/k))$ rate-distortion content, can be performed using $m = O(k \log(n/k))$ noisy measurements with modest $O(n \log(n))$ complexity.
The remainder of the chapter is organized as follows. Section 4.2 describes the mathematical setting formally, followed by a derivation of our lower bound on the measurement rate in Section 4.3. Examples are provided in Section 4.4. We discuss our work on reconstruction via LDPC measurement matrices in Section 4.5, and conclude the chapter in Section 4.6.

4.2 Set up

4.2.1 Notation and preliminaries

We denote random variables by upper-case letters and realizations of random variables by their corresponding lower-case letters, e.g., $x$ is a realization of the random variable $X$. We also use lower-case letters to represent deterministic variables, e.g., $n$ to denote the signal length. Vectors are denoted in boldface in order to distinguish them from scalars, e.g., the vector $\mathbf{x}$ is a realization of the random vector $\mathbf{X}$. We refer to the entries of a vector using the subscript notation, and so $\mathbf{X} = [X_1, X_2, ..., X_n]^T$ is a length $n$ random vector, where $[\cdot]^T$ is the transpose operator. We assume that all vectors are column vectors. We denote the probability density function (pdf) of a random variable $X$ as $p_X(x)$ and the cumulative distribution function (cdf) as $P_X(x)$. We represent estimators of variables using the “hat” notation, e.g., the random variable $\hat{X}$ is an estimator for $X$.

A source associated with a random variable $X$ (called “source $X$”) produces a random vector $\mathbf{X} = [X_1, X_2, ..., X_n]^T$ of a specified length $n$, where the entries $X_i, i = 1, 2, ..., n$ are independent and identically distributed (i.i.d.) and $X_i \sim p_X(x)$.

Finally, we use standard notation to refer to information theoretic measures such as the channel capacity ($C$) and rate-distortion function ($R(\cdot)$) [25].
4.2.2 Measurement and reconstruction

We acquire $m$ real-valued measurements of a signal $X$ by multiplying $X$ with a measurement matrix $\Phi \in \mathbb{R}^{p \times k}$. We denote the ideal (uncontaminated) measurement vector by $Y_0$ so that $Y_0 = \Phi X$. The observed measurements are obtained by corrupting the ideal measurements with additive noise. The noise vector $Z \in \mathbb{R}^p$ consists of $m$ i.i.d. $\mathcal{N}(t, \infty)$ random variables. We denote the observed measurement vector by $Y$ so that

$$Y = Y_0 + Z = \Phi X + Z.$$ 

Therefore, the statistics of the measurement process are fully characterized by specifying the measurement matrix $\Phi$.

A reconstruction scheme uses a decoder to estimate the signal $X$ using the observed measurements $Y$. The decoder $D_X$ is a mapping $D_X : \mathbb{R}^p \rightarrow \mathbb{R}^k$ that takes the measurement vector $Y$ as an input and produces an estimate $\hat{X}$ of $X$. It is assumed that the signal statistics and the measurement matrix $\Phi$ are known to the decoder.

4.2.3 Measurement and reconstruction quality

We define the measurement signal-to-noise ratio SNR as the ratio between the expected noiseless measurement energy and the expected noise energy:

$$\text{SNR} \triangleq \frac{E[||Y_0||^2]}{E[||Z||^2]} = \frac{E[||Y_0||^2]}{m}.$$ 

We also define a metric to gauge the quality of reconstruction for a given pair of CS measurement matrix $\Phi$ and reconstruction scheme $D_X$. We use the normalized squared reconstruction error

$$\mathcal{E}(D_X) \triangleq \frac{E[||X - \hat{X}||^2]}{E[||X||^2]},$$ 

where $\hat{X}$ is the estimate produced by the decoder.
where the expectation in the numerator is over the joint distribution of $X$ and $\hat{X}$. Finally, we define the measurement rate as

$$\delta \triangleq \frac{m}{n}.$$  

### 4.3 Lower bound on reconstruction error

The goal of this section is to find the minimum measurement rate needed to reconstruct the signal $X$ to achieve a given fidelity using CS measurement and reconstruction schemes as described above. Specifically, we seek to determine an asymptotic \textit{lower bound} on $\delta$ in order to achieve a reconstruction quality $\mathcal{E}(D_X)$, \textit{irrespective} of $\Phi$ and the decoding scheme $D_X$.

#### 4.3.1 Approach

To probe the performance limits of CS, we draw inspiration from information theory. Our motivation stems from the following insights. The source $X$ is a discrete-time continuous amplitude source. The measurements $Y$ are modeled as outputs of a Gaussian channel, where the channel inputs are the ideal CS measurements $Y_0$ (the precise nature of the channel is described in Section 4.3.3). This channel has a finite capacity and so each measurement only extracts a finite amount of information. Therefore, perfect signal recovery is impossible [14]. We seek to find a lower bound on the measurement rate $\delta$ in terms of the distortion $\mathcal{E}(D_X) > 0$ and measurement signal-to-noise ratio SNR.

In order to apply information theoretic insights to solve this problem, we investigate the amount of information that can be extracted from the CS measurements. This quantity is determined by the capacity of the measurement channel.
Having upper bounded the information contained in the measurements, we investigate the minimum information (in bits) needed to reconstruct the signal with distortion $\mathcal{E}(\mathcal{D}_X)$. This result can be obtained from the rate-distortion function for the source $X$ using the mean squared error distortion measure. We have thus characterized $\delta$ and $\mathcal{E}(\mathcal{D}_X)$ in terms of a common currency, namely bits. Invoking the source-channel separation theorem [8], we obtain a lower bound on $\delta$ as a function of $\mathcal{E}(\mathcal{D}_X)$ and SNR.

### 4.3.2 Main result

**Theorem 24** For a signal source with rate-distortion function $R(\cdot)$ and measurement scheme specified above, the lower bound on the CS measurement rate required to obtain normalized reconstruction error $\mathcal{E}(\mathcal{D}_X)$ subject to a fixed SNR is given by

$$\delta \geq \frac{2R(\mathcal{E}(\mathcal{D}_X))}{\log(1 + \text{SNR})}$$

as $n \to \infty$.

Before providing the proof for Theorem 24, let us pause to reflect on the theorem statement. First, the theorem is valid for any i.i.d. source for which the rate-distortion function can be characterized. In particular, the scope of the theorem extends beyond signals that are exactly sparse. Second, the theorem is valid in the asymptotic regime when $n \to \infty$. Finally, the theorem reveals the crucial dependence of $\delta$ on the measurement SNR.

### 4.3.3 Proof of Theorem 24

In the first part of the proof, we derive the capacity of the measurement channel. This result enables us to compute the maximum information that can be extracted from $m$
noisy real valued measurements $Y$. The second part of the proof applies the source-channel separation theorem for discrete-time continuous-amplitude ergodic sources to the rate-distortion function of the source and the aforementioned capacity.

**Capacity of the measurement channel**

We consider noisy measurements $Y$ obtained by passing the noise free measurements $Y_0$ through an additive Gaussian noise channel with i.i.d. noise components $N(t, \infty)$. We call this the *measurement channel*, and it is characterized by the input-output relationship

$$Y = Y_0 + Z.$$  

In order to calculate the capacity of the measurement channel, we consider the covariance matrices $\Sigma_{Y_0}$ and $\Sigma_Z = I_{m \times m}$ of $Y_0$ and $Z$ respectively. Note that we can write the SNR in terms of $\Sigma_{Y_0}$ as $\text{SNR} = (1/m) \text{tr}(\Sigma_{Y_0})$, where $\text{tr}(\cdot)$ refers to the trace of a matrix.

To compute the capacity of this channel, we use the well-known result [25]

$$C = \max_{\text{tr}(\Sigma_{Y_0}) \leq m \text{SNR}} \frac{1}{2m} \log \frac{|\Sigma_{Y_0} + \Sigma_Z|}{|\Sigma_Z|},$$

where $C$ is the channel capacity in bits per measurement, and $|\cdot|$ denotes the determinant. Because $\Sigma_Z = I_{m \times m}$ we have $|\Sigma_Z| = 1$, and the above equation reduces to

$$C = \max_{\text{tr}(\Sigma_{Y_0}) \leq m \text{SNR}} \frac{1}{2m} \log |\Sigma_{Y_0} + I_{m \times m}|.$$

To maximize the channel capacity, we seek to determine the correlation matrix $\Sigma_{Y_0}$ that maximizes $|\Sigma_{Y_0} + I_{m \times m}|$ subject to the constraint $\text{tr}(\Sigma_{Y_0}) \leq m \text{SNR}$. For this, we apply Hadamard’s inequality [25] which states that the determinant of any
positive definite matrix $\Lambda$ is less than or equal to the product of the diagonal elements, i.e., $|\Lambda| \leq \prod_i \Lambda(i, i)$, with equality if and only if the matrix is diagonal. Since $(\Sigma_Y + I_{m \times m})$ is a positive definite matrix (it is a sum of two covariance matrices that are each positive definite), we have $|\Sigma_Y + I_{m \times m}| \leq \prod_i (1 + \Sigma_Y(i, i))$.

Finally, the maximum value for the product $\prod_i (1 + \Sigma_Y(i, i))$ under the constraint $\text{tr}(\Sigma_Y) \leq m \text{SNR}$ is attained when the diagonal entries $\Sigma_Y(i, i)$ all equal SNR. Using the above arguments, we have

$$C \leq \max_{\text{tr}(\Sigma_Y) \leq m \text{SNR}} \frac{1}{2m} \log \prod_i (1 + \Sigma_Y(i, i))$$

$$\leq \frac{1}{2} \log(1 + \text{SNR}),$$

where equality is attained when $Y_0$ is diagonal and the diagonal entries are all equal to SNR. Therefore, the best CS measurement system has statistically independent measurements $Y_0$, with all measurements having the same variance. This revelation can be used as a practical guiding principle in order to construct good compressed sensing matrices $\Phi$. We summarize the important conclusions in the form of a Lemma.

**Lemma 1** The upper bound on the capacity of the CS measurement channel is given by

$$C \leq \frac{1}{2} \log(1 + \text{SNR})$$

bits per measurement. Equality in the above equation requires that the measurements in $Y_0$ are statistically independent and all measurements have the same variance equal to SNR.
Computing the error bound using the source-channel separation theorem

So far, we have described the information provided by the measurements $Y$. We can also characterize the information content of the requisite signal reconstruction quality using rate-distortion formulae.

The source-channel separation theorem for discrete-time continuous amplitude stationary ergodic signals [8] states that a source $X$ can be communicated up to distortion quality $D$ via $m$ channel uses if and only if the information content $mC$ that can be extracted from the channel exceeds the information content $nR(D)$ of the quantized source.

We complete the proof of Theorem 24 by applying the converse portion of the separation theorem with Lemma 1 and the rate-distortion function $R(D)$. □

4.4 Examples

Consider a $k$-sparse signal $X$ where the spikes have uniform amplitude. In this case, it is well known that precise description of $X$ would require $\log\left(\binom{n}{k}\right) \approx k \log(n/k)$ bits, where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$  

This problem can be extended to a lossy description of $X$ using a rate-distortion approach for a binary asymmetric source [25] or recent results on the rate-distortion of spike processes [85, 16]. For small distortion values, the rate-distortion content remains roughly $k \log(n/k)$ bits. Combining this result with our lower bound, we obtain the following condition on the number of measurements

$$m \gtrsim \frac{2k \log(n/k)}{\log(1 + SNR)},$$
where the approximation is due to the asymptotic nature of the information theoretic tools that were used in our proof.

Let us now examine specific numerical examples. First, suppose that the signal is of length $n = 10^7$ and contains $k = 10^3$ spikes. If the measurement signal-to-noise ratio satisfies $\text{SNR} = 10$ dB, then the number of measurements must satisfy

$$m \gtrapprox 2 \cdot 10^3 \log(10^7/10^3) = 7,682.$$  

Therefore, a “reasonable” SNR requires a modest number of measurements. In contrast, if we choose $\text{SNR} = -20$ dB, then $m$ must exceed $1.85 \cdot 10^6$. Although in this case $m$ is still much smaller than $n$, a poor signal-to-noise ratio prevents a drastic reduction in the number of measurements.

Our lower bound can also be used to provide results for denoising sparse signals from a reduced number of measurements. Fletcher et al. [36] provided bounds on denoising sparse signals using redundant frames. Their work considered several specific sparse signal models. Although our work does not consider the extension to redundant frames, our result is more general in the sense than any i.i.d. signal model can be used.

### 4.5 Practical reconstruction scheme

We have extended the success of LDPC codes [39, 63] to the problem of CS measurement and reconstruction. The crucial principle is the use of low density structure for the CS matrix $\Phi$. This special structure for $\Phi$ can be leveraged in three ways. First, the LDPC structure enables low-complexity computation of measurements, because each measurement depends only on a small set of coefficients. Second, the sparse structure of $\Phi$ can be used to provide low-complexity reconstruction schemes
by deploying message passing algorithms [39, 9, 63]. Third, these LDPC-based CS reconstruction schemes can operate close to the theoretical limits of Theorem 24.

4.5.1 Measurement process

We compute the CS measurements using a sparse CS matrix $\Phi$, where the entries of $\Phi$ are drawn from the set $\{0, 1, -1\}$. Note that in addition to using the elements 0 and 1 as in LDPC codes, we also include the element $-1$ in order to ensure that the expected values of the elements in each row (and column) of $\Phi$ are zero. In this setting, (uncorrupted) measurements $Y_0(i)$ are just sums and differences of small subsets of the coefficients of the signal $X$. The design of $\Phi$ (such as fixing the row weight, column weight, and so on) is based on the properties of the sparse signal $X$ as well as the accompanying decoding algorithm. The goal is to imbibe the reconstruction algorithm with low-complexity yet require a modest number of measurements. We have argued that choosing the row weight to be inversely proportional to the sparsity rate of the input signal yields good performance [66].

4.5.2 Reconstruction via belief propagation

The use of sparse CS matrix facilitates the application of message passing algorithms for signal reconstruction. The key property that enables us to use message passing algorithms is that the sparse structure of $\Phi$ can be represented as a sparse bipartite graph. Signal reconstruction can be viewed as a Bayesian inference problem, and can be solved by iteratively exchanging messages over the edges of the said graph using the well known belief propagation algorithm. The stochastic signal model (where we model the coefficients of the input signal as i.i.d. outcomes of a distribution $p_X(\cdot)$) can be used as a prior to model the input signal. Belief propagation allows us to estimate the signal that explains the measurements and best matches the signal prior. We have
shown that this technique exhibits $O(n \log(n))$ complexity and empirically observed that $m = O(k \log(n))$ measurements are required.

### 4.5.3 “One shot” reconstruction

We have also considered a simplified “one shot” reconstruction algorithm for sparse signals. The algorithm is based on a median filter approach. The complexity of this approach is also $O(n \log(n))$. Surprisingly, this algorithm achieves decent reconstruction fidelity for $k$-sparse signals using $m = O(k \log(n/k))$ measurements. Again, because the rate-distortion content of a $k$-sparse signal is $\Theta(k \log(n/k))$, the combination of this result with our lower bound (Theorem 24) indicates the potential to develop low-complexity “capacity approaching” CS schemes.

### 4.6 Conclusions

In this chapter, we presented information theoretic arguments to lower bound the number of noisy compressed sensing (CS) measurements required for signal reconstruction. The key idea is to model the noisy signal acquisition process as a communication channel. The capacity of this channel allows us to express the information contained in the measurements in terms of bits. Using this result along with the rate-distortion function of the source yields a converse result on the measurement rates required. We considered the example of using a spike process as input, and presented numerical results given by the theoretical bound.

This work further strengthens the connections between information theory and CS. As part of our ongoing work, we are investigating the best achievable CS reconstruction scheme. In particular, we are striving to provide converse and achievable
bounds that are tight. The resolution of this problem will determine the effectiveness of CS in encoding discrete-time real-valued signals.
Chapter 5

Analysis of the One-Stage Greedy Algorithm

Setup

We denote the signals by $x_j, j \in \{1, \ldots, J\}$, and assume that each signal $x_j \in \mathbb{R}^N$. Our DCS II model [7] for joint sparsity concerns the case of multiple sparse signals that share common sparse components, but with different coefficients. For example,

$$x_j = \Psi \theta_j,$$

where each $\theta_j$ is supported only on $\Omega \subset \{1, 2, \ldots, N\}$, with $|\Omega| = K$. The matrix $\Psi$ is orthonormal, with dimension $N \times N$ (we consider only signals sparse in an orthonormal basis).

We denote by $\Phi_j$ the measurement matrix for signal $j$, where $\Phi_j$ is of dimension $M \times N$, where $M < N$. We let $y_j = \Phi_j x_j = \Phi_j \Psi \theta_j$ be the observations of signal $j$.

We assume that the measurement matrix $\Phi_j$ is random with i.i.d entries taken from a $\mathcal{N}(0,1)$ distribution. Clearly, the matrix $\Phi_j \Psi$ also has i.i.d $\mathcal{N}(0,1)$ entries, because $\Psi$ is orthonormal. For convenience, we assume $\Psi$ to be identity $I_{N \times N}$. The results presented can be easily extended to a more general orthonormal matrix $\Psi$ by replacing $\Phi_j$ with $\Phi_j \Psi$. 
Recovery

After gathering all of the measurements, we compute the following statistic for each $n \in \{1, \ldots, N\}$:

$$\xi_n = \frac{\sum_{j=1}^{J} \langle y_j, \phi_{j,n} \rangle^2}{J},$$

(5.1)

where $\phi_{j,n}$ denotes column $n$ of measurement matrix $\Phi_j$. To estimate $\Omega$ we choose the $K$ largest statistics $\xi_n$. We have the following results.

**Theorem 25** Assume the $M \times N$ measurement matrices $\Phi_j$ contain i.i.d. $\mathcal{N}(0,1)$ entries and that the coefficient vectors $x_j$ contain i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. Let $y_j = \Phi_j x_j$ and let $\xi_n$ be defined as in Equation (5.1). The mean and variance of $\xi_n$ are given by

$$E\xi_n = \begin{cases} m_b & \text{if } n \notin \Omega \\ m_g & \text{if } n \in \Omega \end{cases}$$

and

$$Var(\xi_n) = \begin{cases} \sigma_b^2 & \text{if } n \notin \Omega \\ \sigma_g^2 & \text{if } n \in \Omega \end{cases},$$

where

$$m_b = MK\sigma^2,$$

$$m_g = M(M + K + 1)\sigma^2,$$

$$\sigma_b^2 = \frac{2MK\sigma^4}{J}(MK + 3K + 3M + 6), \quad \text{and}$$

$$\sigma_g^2 = \frac{M\sigma^4}{J}(34MK + 6K^2 + 28M^2 + 92M + 48K + 90 + 2M^3 + 2MK^2 + 4M^2 K).$$
Theorem 26  The one shot algorithm recovers Ω with a probability of success $p_s$ given by approximately

$$p_s \approx \frac{1}{2^{N-1}} \frac{(N - K)}{\sigma_b \sqrt{2\pi}} \int_{-\infty}^{\infty} \left[1 + erf\left(\frac{x - m_b}{\sigma_b \sqrt{2}}\right)\right]^{N-K-1} \left[1 - erf\left(\frac{x - m_g}{\sigma_g \sqrt{2}}\right)\right]^K \exp \left[-\left(\frac{x - m_b}{\sigma_b \sqrt{2}}\right)^2\right] dx.$$ 

Corollary 2  The one-stage algorithm recovers Ω with probability approaching 1 as $J \to \infty$.

Remark 4  The mean and variance of $\xi_n$ are independent of $N$.

Remark 5  The variance of $\xi_n$ goes to zero as $J \to \infty$.

Proof of Theorem 25

We first present a short sketch of the strategy we adopt to prove the result. The main idea is to compute the statistics of $\langle y_j, \phi_{j,n} \rangle$ up to first four moments, for $n \in \Omega$ and $n \notin \Omega$. Based on these results, we derive the mean and variance of $\xi_n$.

We use the following ideas in our proof. Let $X_1$ and $X_2$ be two independent random variables. Define random variables $Y$ and $Z$ as $Y = X_1 \times X_2$ and $Z = X_1 + X_2$. Then, the $p^{th}$ moment of $Y$ — which we denote by $m_p(Y)$ — is given by $m_p(Y) = m_p(X_1) \times m_p(X_2)$. Furthermore, the $p^{th}$ cumulant [55] of $Z$ — denoted by $c_p(Z)$ — is given by $c_p(Z) = c_p(X_1) + c_p(X_2)$. When we multiply independent random variables, we work with their moments. While working with the sum of independent random variables, we work with their cumulants. We use the standard formulae [55] to convert from moments to cumulants and vice versa.

We use the notation $X = \text{Moments}(m_1, m_2, ..., m_p)$ to keep track of the first $p$ moments of the random variable $X$. Likewise, we denote $X = \text{Cumulants}(c_1, c_2, ..., c_p)$
to keep track of the first $p$ cumulants of $X$. The conversion from cumulants to moments and vice versa for up to two orders is as follows:

$$\text{Cumulants}(c_1, c_2) \equiv \text{Moments}(m_1, m_2)$$

if $c_1 = m_1$, and $m_2 = c_2 + c_1^2$ (or equivalently $c_2 = m_2 - m_1^2$). The first and second cumulants correspond, respectively, to the mean and variance.

We also use the results for the moments of a Gaussian Random variable $X \sim \mathcal{N}(0, 1)$: $EX^4 = 3$ and $EX^6 = 15$.

We begin by computing statistics of operations on the columns of the matrix $\Phi_j$. These results are presented in the form of five Lemmas.

**Lemma 2** For $1 \leq j \leq J$, $1 \leq n, l \leq N$ and $n \neq l$,

$$E\langle \phi_{j,n}, \phi_{j,l} \rangle^2 = M.$$

**Proof of Lemma:** Let $\phi_{j,n}$ be the column vector $[a_1, a_2, ..., a_M]^T$, where each element in the vector is iid $\mathcal{N}(0, 1)$. Likewise, let $\phi_{j,l}$ be the column vector $[b_1, b_2, ..., b_M]^T$ where the elements are iid $\mathcal{N}(0, 1)$. We have

$$\langle \phi_{j,n}, \phi_{j,l} \rangle^2 = (a_1 b_1 + a_2 b_2 + ... a_M b_M)^2 = \sum_{q=1}^{M} a_q^2 b_q^2 + 2 \sum_{q=1}^{M-1} \sum_{r=q+1}^{M} a_q a_r b_q b_r.$$
Taking expectations,

\[
E \left[ (\langle \phi_{j,n}, \phi_{j,l} \rangle)^2 \right] = E \left[ \sum_{q=1}^{M} a_q^2 b_q^2 \right] + 2E \left[ \sum_{q=1}^{M-1} \sum_{r=q+1}^{M} a_q a_r b_q b_r \right] 
\]

\[
= \sum_{q=1}^{M} E(a_q^2 b_q^2) + 2 \sum_{q=1}^{M-1} \sum_{r=q+1}^{M} E(a_q a_r b_q b_r) 
\]

\[
= \sum_{q=1}^{M} E(a_q^2) E(b_q^2) + 2 \sum_{q=1}^{M-1} \sum_{r=q+1}^{M} E(a_q) E(a_r) E(b_q) E(b_r) 
\]

(because the random variables are independent)

\[
= \sum_{q=1}^{M} (1) + 0 
\]

(because \(E(a_q^2) = E(b_q^2) = 1\) and \(E(a_q) = E(b_q) = 0\))

\[
= M. 
\]

This completes the proof of the Lemma.

**Lemma 3** For \(1 \leq j \leq J\), \(1 \leq n, l \leq N\) and \(n \neq l\),

\[
E(\langle \phi_{j,n}, \phi_{j,l} \rangle)^4 = 3M(M + 2). 
\]

**Proof of Lemma:** As before, let \(\phi_{j,n}\) be the column vector \([a_1, a_2, ..., a_M]^T\), where each element in the vector is iid \(\mathcal{N}(0,1)\). Likewise, let \(\phi_{j,l}\) be the column vector
$[b_1, b_2, ..., b_M]^T$ where the elements are iid $\mathcal{N}(0,1)$. We have

$$E\langle \phi_{j,n}, \phi_{j,l} \rangle^4 = E(a_1 b_1 + a_2 b_2 + ... a_M b_M)^4$$

$$= E \left[ \sum_{q=1}^M a_q^4 b_q^4 \right] + \binom{4}{2} E \left[ \sum_{q=1}^{M-1} \sum_{r=q+1}^M (a_q b_q)^2 (a_r b_r)^2 \right]$$

$$+ E(\text{cross terms with zero expectation})$$

$$= \sum_{q=1}^M E a_q^4 E b_q^4 + 6 \sum_{q=1}^{M-1} \sum_{r=q+1}^M E a_q^2 E b_q^2 E a_r^2 E b_r^2 \quad \text{(by independence)}$$

$$= 9M + 6 \frac{M(M-1)}{2} \quad \text{(because $E a_q^4 = 3$ and $E a_q^2 = 1$)}$$

$$= 3M(M + 2).$$

This completes the proof of the Lemma.

**Lemma 4** For $1 \leq j \leq J$, $1 \leq n, l, q \leq N$ and unique $n, l$ and $q$,

$$E \left[ \langle \phi_{j,n}, \phi_{j,l} \rangle^2 \langle \phi_{j,n}, \phi_{j,q} \rangle^2 \right] = M(M + 2).$$

**Proof of Lemma:** As before, let $\phi_{j,n}$ be the column vector $[a_1, a_2, ..., a_M]^T$, where each element in the vector is iid $\mathcal{N}(0,1)$. Likewise, let $\phi_{j,n}$ be the column vector $[b_1, b_2, ..., b_M]^T$ and $\phi_{j,q}$ be the column vector $[c_1, c_2, ..., c_M]^T$. From the statement of the Lemma,

$$LHS = E \left[ \langle \phi_{j,n}, \phi_{j,l} \rangle^2 \langle \phi_{j,n}, \phi_{j,q} \rangle^2 \right]$$

$$= E \left[ (a_1 b_1 + a_2 b_2 + ... a_M b_M)^2 (a_1 c_1 + a_2 c_2 + ... a_M c_M)^2 \right]$$

$$= E \left[ \left( \sum_{r=1}^M a_r^2 b_r^2 + \sum_{r=1}^M \sum_{s=1, s \neq r}^M a_r b_r a_s b_s \right) \left( \sum_{r=1}^M a_r^2 c_r^2 + \sum_{r=1}^M \sum_{s=1, s \neq r}^M a_r c_r a_s c_s \right) \right].$$
Collecting only those terms with non-zero expectations,

\[
LHS = E \left[ \sum_{r=1}^{M} a_r^4 b_r^2 c_r^2 + \sum_{r=1}^{M} \sum_{s=1, s \neq r}^{M} a_r^2 a_s^2 b_r^2 c_s^2 \right]
\]

\[
= \sum_{r=1}^{M} E a_r^4 E b_r^2 E c_r^2 + \sum_{r=1}^{M} \sum_{s=1, s \neq r}^{M} E a_r^2 E a_s^2 E b_r^2 E c_s^2
\]

\[
= 3M + M(M - 1)
\]

\[
= M(M + 2).
\]

This completes the proof of the Lemma.

**Lemma 5** For 1 ≤ j ≤ J, 1 ≤ n, l ≤ N and n ≠ l,

\[
E \left[ \|\phi_{j,l}\|^4 \langle \phi_{j,n}, \phi_{j,l} \rangle^2 \right] = M(M + 2)(M + 4).
\]

**Proof of Lemma:** As before, let \( \phi_{j,n} \) be the column vector \([a_1, a_2, \ldots, a_M]^T\), where each element in the vector is iid \( \mathcal{N}(0, 1) \). Likewise, let \( \phi_{j,l} \) be the column vector \([b_1, b_2, \ldots, b_M]^T\) where the elements are iid \( \mathcal{N}(0, 1) \). We have

\[
E \left[ \|\phi_{j,l}\|^4 \langle \phi_{j,n}, \phi_{j,l} \rangle^2 \right] = E \left[ (a_1^2 + a_2^2 + \ldots a_M^2)^2 (a_1 b_1 + a_2 b_2 + \ldots a_M b_M)^2 \right]
\]

\[
= E \left[ \left( \sum_{r=1}^{M} a_r^4 + \sum_{r=1}^{M} \sum_{s=1, s \neq r}^{M} a_r^2 a_s^2 \right) \left( \sum_{r=1}^{M} b_r^2 + \sum_{s=1, s \neq r}^{M} a_r b_r a_s b_s \right) \right].
\]

Collecting only the terms with non-zero expectations,

\[
E \left[ \|\phi_{j,l}\|^4 \langle \phi_{j,n}, \phi_{j,l} \rangle^2 \right] = E \left[ \sum_{r=1}^{M} a_r^6 b_r^2 + \sum_{r=1}^{M} \sum_{s=1, s \neq r}^{M} a_r^4 a_s^2 b_r^2 + 2 \sum_{r=1}^{M} \sum_{s=1, s \neq r}^{M} a_r^4 a_s^2 b_r^2 b_s^2 \right]
\]

\[
= 15M + 3M(M - 1) + 6M(M - 1) + M(M - 1)(M - 2)
\]

(because for \( X \sim \mathcal{N}(0, 1) \), \( EX^4 = 3 \) and \( EX^6 = 15 \))

\[
= M(M + 2)(M + 4).
\]
This completes the proof of the Lemma.

**Lemma 6** For $1 \leq j \leq J$, $E\|\phi_{j,l}\|^4 = M(M + 2)$, and $E\|\phi_{j,l}\|^8 = M(M + 2)(M + 4)(M + 6)$.

**Proof of Lemma:** Let $\phi_{j,n}$ be the column vector $[a_1, a_2, ..., a_M]^T$, Define the random variable $Z = \|\phi_{j,l}\|^2 = \sum_{q=1}^{M} a_q^2$. From the theory of random variables, we know that $Z$ is chi-squared distributed with $m$ degrees of freedom. Thus, $EZ^2 = M(M + 2)$ and $EZ^4 = M(M + 2)(M + 4)(M + 6)$, which proves the lemma.

**Statistics of $\xi_n$ when $n \notin \Omega$**

Assume without loss of generality that $\Omega = \{1, 2, \ldots, K\}$ for convenience of presentation. Let us compute the mean and variance of the test statistic $\xi_n$ for the case when $n \notin \Omega$. Consider one of these statistics by choosing $n = K + 1$.

Let $B = \langle y_j, \phi_{j,K+1} \rangle = \sum_{l=1}^{K} x_i(l)\langle \phi_{j,l}, \phi_{j,K+1} \rangle$. Clearly, the expectations of odd powers of $B$ are zero, because $E(x_j(l)) = 0$ and $x_j(l)$ is independent of the other factors in each term of the summation. We will now compute $EB^2$ and $EB^4$. First,
consider $EB^2$.

$$EB^2 = E \left[ \sum_{l=1}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,K+1} \rangle \right]^2$$

$$= E \left[ \sum_{l=1}^{K} (x_j(l))^2 (\langle \phi_{j,l}, \phi_{j,K+1} \rangle)^2 \right]$$

$$+ E \left[ \sum_{q=1}^{K} \sum_{l=1, l\neq q}^{K} x_j(l)x_j(q) \langle \phi_{j,l}, \phi_{j,K+1} \rangle \langle \phi_{j,q}, \phi_{j,K+1} \rangle \right]$$

$$= \sum_{l=1}^{K} E (x_j(l))^2 E (\langle \phi_{j,l}, \phi_{j,K+1} \rangle)^2 +$$

$$\sum_{q=1}^{K} \sum_{l=1, l\neq q}^{K} E(x_j(l))E(x_j(q))E(\langle \phi_{j,l}, \phi_{j,K+1} \rangle \langle \phi_{j,q}, \phi_{j,K+1} \rangle)$$

(because the terms are independent)

$$= \sum_{l=1}^{K} E (x_j(l))^2 E (\langle \phi_{j,l}, \phi_{j,K+1} \rangle)^2 \quad \text{(because } E(x_j(l)) = E(x_j(q)) = 0)$$

$$= \sum_{l=1}^{K} \sigma^2 M \quad \text{(from Lemma 2)}$$

$$= MK\sigma^2.$$

Next, consider $EB^4$.

$$EB^4 = E \left[ \sum_{l=1}^{K} x_i(l) \langle \phi_{j,l}, \phi_{j,K+1} \rangle \right]^4$$

$$= E \left[ \sum_{l=1}^{K} (x_j(l))^4 (\langle \phi_{j,l}, \phi_{j,K+1} \rangle)^4 \right] +$$

$$\binom{4}{2} E \left[ \sum_{q=1}^{K} \sum_{l=1, l\neq q}^{K} (x_j(l))^2(x_j(q))^2 (\langle \phi_{j,l}, \phi_{j,K+1} \rangle)^2 (\langle \phi_{j,q}, \phi_{j,K+1} \rangle)^2 \right].$$

The cross terms that involve $x_j(l)$, $x_j(q)$, $(x_j(l))^3$, $(x_j(q))^3$ factors have zero expectation, and hence not shown in the above equation. To explain the $\binom{4}{2}$ factor in the above expression, note that we have $\binom{4}{2}$ ways of obtaining the product of two squared
factors when we expand $EB^4$. Thus,

\[
EB^4 = \sum_{l=1}^{K} E\left( (x_j)^4 \right) E\left( (\phi_{j,l}, \phi_{j,K+1})^4 \right) + 
\]

\[
6 \sum_{q=1}^{K} \sum_{l=1, l \neq q}^{K} E(x_j(l))^2 E(x_j(q))^2 E((\phi_{j,l}, \phi_{j,K+1})^2 (\phi_{j,q}, \phi_{j,K+1})^2)
\]

(because the terms are independent)

Let us consider the two terms in the above equation separately. Simplifying the first term, we get

\[
\sum_{l=1}^{K} E\left( (x_j)^4 \right) E\left( (\phi_{j,l}, \phi_{j,K+1})^4 \right) = 3k\sigma^2 E\left( (\phi_{j,l}, \phi_{j,K+1})^4 \right)
\]

\[
= 9K\sigma^4 M(M + 2) \quad \text{(from Lemma 3)}.
\]

The second term can be reduced to

\[
6 \sum_{q=1}^{K} \sum_{l=1, l \neq q}^{K} E(x_j(l))^2 E(x_j(q))^2 E((\phi_{j,l}, \phi_{j,K+1})^2 (\phi_{j,q}, \phi_{j,K+1})^2)
\]

\[
= 6 \frac{K(K - 1)}{2} \sigma^4 M(M + 2) \quad \text{(from Lemma 4)}
\]

\[
= 3\sigma^4 K(K - 1)M(M + 2).
\]

Summing the two terms, we get

\[
EB^4 = 9K\sigma^4 M(M + 2) + 3\sigma^4 K(K - 1)M(M + 2)
\]

\[
= 3MK\sigma^4 (M + 2)(K + 2).
\]

Thus, we have

\[
B = \langle y_j, \phi_{j,K+1} \rangle = \text{Moments}(0, MK\sigma^2, 0, 3MK\sigma^4(M + 2)(K + 2)).
\]
Thus the first two moments for $\langle y_j, \phi_{j,K+1} \rangle^2$ are

$$\langle y_j, \phi_{j,K+1} \rangle^2 = \text{Moments}(MK\sigma^2, 3MK\sigma^4(M + 2)(K + 2)).$$

Writing in terms of cumulants,

$$\langle y_j, \phi_{j,K+1} \rangle^2 = \text{Cumulants}(MK\sigma^2, 3MK\sigma^4(M + 2)(K + 2) - M^2K^2\sigma^4) = \text{Cumulants}(MK\sigma^2, 2MK\sigma^4(MK + 3K + 3M + 6)).$$

Summing $J$ such independent random variables,

$$\sum_{j=1}^{J} \langle y_j, \phi_{j,K+1} \rangle^2 = \text{Cumulants}(MKJ\sigma^2, 2MKJ\sigma^4(MK + 3K + 3M + 6)).$$

Dividing by $J$,

$$\frac{1}{J} \sum_{j=1}^{J} \langle y_j, \phi_{j,K+1} \rangle^2 = \text{Cumulants}(MK\sigma^2, \frac{2MK\sigma^4}{J}(MK + 3K + 3M + 6)).$$

The above equation gives the mean and the variance of the test statistic $\xi_n$ when $n \notin \Omega$.

**Statistics of $\xi_n$ when $n \in \Omega$**

Again, we assume without loss of generality that $\Omega = \{1, 2, \ldots, K\}$ for ease of presentation. Let us compute the mean and variance of the test statistic $\xi_n$ for the case when $n \in \Omega$. Consider one of these statistics by choosing $n = 1$.

Let $G = \langle y_j, \phi_{j,1} \rangle = x_j(1)\|\phi_{j,1}\|^2 + \sum_{l=2}^{K} x_j(l)\langle \phi_{j,l}, \phi_{j,1} \rangle$. As before, the expectations of odd powers of $G$ are zero, because of the leading $x_j(\cdot)$ factor in each term.
We will now compute $E G^2$ and $E G^4$. First, consider $E G^2$.

$$
E G^2 = E \left[ x_j(1) ||\phi_{j,1}||^2 + \sum_{l=2}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,1} \rangle \right]^2 \\
= E \left[ (x_j(1))^2 ||\phi_{j,1}||^4 \right] + E \left[ \sum_{l=2}^{K} (x_j(l))^2 \langle \phi_{j,l}, \phi_{j,1} \rangle^2 \right] \\
(All \ other \ cross \ terms \ have \ zero \ expectation) \\
= E (x_j(1))^2 E ||\phi_{j,1}||^4 + \sum_{l=2}^{K} E (x_j(l))^2 E \langle \phi_{j,l}, \phi_{j,1} \rangle^2 \quad (by \ independence) \\
= \sigma^2 M (M + 2) + (K - 1) \sigma^2 M \quad (from \ Lemmas \ 2 \ and \ 6) \\
= M (M + K + 1) \sigma^2.
$$

Next, consider $E G^4$.

$$
E G^4 = E \left[ x_j(1) ||\phi_{j,1}||^2 + \sum_{l=2}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,1} \rangle \right]^4 \\
= E \left[ x_j(1) ||\phi_{j,1}||^2 \right]^4 \\
+ E \left[ \sum_{l=2}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,1} \rangle \right]^4 \\
+ \binom{4}{2} E \left[ (x_j(1) ||\phi_{j,1}||^2)^2 \left( \sum_{l=2}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,1} \rangle \right)^2 \right] \\
(All \ other \ cross \ terms \ have \ zero \ expectation).
$$

We use the result from Lemma 6 to simplify the first term, and the result from the fourth moment of the statistic $\xi_n$ when $n \notin \Omega$ for the second term, to get

$$
E G^4 = 3 \sigma^4 M (M + 2) (M + 4) (M + 6) + 3M (K - 1) \sigma^4 (M + 2) (K + 1) \\
+ 6E \left[ (x_j(1) ||\phi_{j,1}||^2)^2 \left( \sum_{l=2}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,1} \rangle \right)^2 \right]. \quad (5.2)
$$
The last term in the above equation can be written as

\[ E \left[ (x_j(1)) \| \phi_{j,1} \|^2 \left( \sum_{l=2}^{K} x_j(l) \langle \phi_{j,l}, \phi_{j,1} \rangle \right)^2 \right] \]

\[ = E \left[ (x_j(1))^2 \| \phi_{j,1} \|^4 \sum_{l=2}^{K} (x_j(l))^2 \langle \phi_{j,l}, \phi_{j,1} \rangle^2 \right] \]

(all other cross terms have zero expectation)

\[ = \sigma^4 E \left[ \sum_{l=2}^{K} \| \phi_{j,1} \|^4 \langle \phi_{j,l}, \phi_{j,1} \rangle^2 \right] \]

\[ = \sigma^4(K - 1) E \left[ \| \phi_{j,1} \|^4 \langle \phi_{j,2}, \phi_{j,1} \rangle^2 \right] \]

\[ = \sigma^4(K - 1)M(M + 2)(M + 4) \]

(using result from Lemma 5).

Substituting this result in Equation 5.2, we get

\[ EG^4 = 3\sigma^4M(M + 2)(M + 4)(M + 6) + 3M(K - 1)\sigma^4(M + 2)(K + 1) \]
\[ + 6(K - 1)M(M + 2)(M + 4)\sigma^4 \]
\[ = 3M\sigma^4(M^3 + 10M^2 + 31M + MK^2 + 2M^2K) \]
\[ + 12MK + 2K^2 + 16K + 30) \]

Thus, we have

\[ G = \langle y_j, \phi_{j,1} \rangle = \text{Moments}(0, M\sigma^2(M + K + 1), 0, \]
\[ 3m\sigma^4(M^3 + 10M^2 + 31M + MK^2) \]
\[ + 2M^2K + 12MK + 2K^2 + 16K + 30)). \]

Thus the first two moments of \( \langle y_j, \phi_{j,1} \rangle^2 \) are

\[ \langle y_j, \phi_{j,1} \rangle^2 = \text{Moments}(M\sigma^2(M + K + 1), \]
\[ 3M\sigma^4(M^3 + 10M^2 + 31M + MK^2) \]
\[ + 2M^2K + 12MK + 2K^2 + 16K + 30)). \]
In terms of cumulants,

\[
(y_j, \phi_j, 1)^2 = \text{Cumulants}(M \sigma^2(M + K + 1),

3M \sigma^4(M^3 + 10M^2 + 31M + MK^2

+ 2M^2K + 12MK + 2K^2 + 16K + 30) - M^2 \sigma^4(M + K + 1)^2)

= \text{Cumulants}(M \sigma^2(M + K + 1),

M \sigma^4(34MK + 6K^2 + 28M^2 + 92M + 48K

+ 90 + 2M^3 + 2MK^2 + 4M^2K)).
\]

Summing \(J\) such random variables,

\[
\sum_{j=1}^{J} (y_j, \phi_j, 1)^2 = \text{Cumulants}(JM \sigma^2(M + K + 1),

JM \sigma^4(34MK + 6K^2 + 28M^2 + 92M + 48K

+ 90 + 2M^3 + 2MK^2 + 4M^2K)).
\]

Dividing by \(J\),

\[
\frac{1}{J} \sum_{j=1}^{J} (y_j, \phi_j, 1)^2 = \text{Cumulants}(M \sigma^2(M + K + 1),

\frac{M \sigma^4}{J}(34MK + 6K^2 + 28M^2

+ 92M + 48K + 90 + 2M^3 + 2MK^2 + 4M^2K)).
\]

The above equation gives the mean and the variance of the test statistic \(\xi_n\) when \(n \in \Omega\).

**Proof of Theorem 26**

The statistic \(\xi_n\) is the mean of \(J\) independent random variables \((y_j, \phi_j, n)\). For large \(J\), we can invoke the central limit theorem [5, 65, 64, 87] to argue that the distribution of \(\xi_n\) is Gaussian with mean and variance as given in Theorem 25.
The one shot algorithm successfully recovers $\Omega$ if the following condition is satisfied: $[\max(\xi_n), n \notin \Omega] < [\min(\xi_n), n \in \Omega]$. To compute the probability that the above condition holds, we derive the equations that describe the distributions for the maximum and minimum, respectively, of $\xi_n$ when $n \notin \Omega$ and when $n \in \Omega$. Define $\xi_{\text{max}} \triangleq [\max(\xi_n), n \notin \Omega]$, and $\xi_{\text{min}} \triangleq [\min(\xi_n), n \in \Omega]$.

Let $x$ be an arbitrary real number. For $n \notin \Omega$, the probability that the statistic $\xi_n$ is less than $x$ is given by its cumulative distribution function (CDF):

$$Pr[\xi_n < x] = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right) \right).$$

Since the cardinality of the set $\Omega'$ is $N - K$, the probability that all the corresponding statistics $\xi_n, n \notin \Omega$ are less than $x$ is given by the CDF of $\xi_{\text{max}}$:

$$Pr[\xi_{\text{max}} < x] = \frac{1}{2^{N-K}} \left( 1 + \text{erf} \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right) \right)^{N-K}. \tag{5.3}$$

The above equation assumes that the statistics $\xi_n$ are independent. In reality, this assumption is not valid. However, we make this assumption in order to get an approximate result.

Using similar arguments, the probability that all the corresponding statistics $\xi_n, n \in \Omega$ are greater than $x$ is given by

$$Pr[\min(\xi_n, n \in \Omega) > x] = Pr[\xi_{\text{min}} > x] = \frac{1}{2^K} \left( 1 - \text{erf} \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right) \right)^{K}.$$

For a given $x$, the probability that $\xi_{\text{max}}$ lies between $x$ and $x + dx$ can be computed using the probability density function (PDF) of $\xi_{\text{max}}$. The PDF of $\xi_{\text{max}}$ in turn can be computed by differentiating its CDF as given by Equation 5.3. Thus,

$$Pr(\xi_{\text{max}} \in [x, x + dx]) = \frac{d}{dx} \left[ \frac{1}{2^{N-K}} \left( 1 + \text{erf} \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right) \right)^{N-K} \right] dx = \frac{1}{2^{N-K-1}} \frac{(N - K)}{\sigma_b \sqrt{2\pi}} \exp \left[ - \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right)^2 \right] \left( 1 + \text{erf} \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right) \right)^{N-K} dx.$$
Thus the probability of successfully recovering $\Omega$ is given by

$$p_s = \int_{x=-\infty}^{x=\infty} Pr(\xi_{\text{max}} \in [x, x + dx]).Pr(\xi_{\text{min}} > x)$$

$$= \frac{1}{2^{N-1}} \sigma_b \sqrt{2\pi} \int_{-\infty}^{\infty} \left[ 1 + \text{erf} \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right) \right] \left[ 1 - \text{erf} \left( \frac{x - m_g}{\sigma_g \sqrt{2}} \right) \right]^{K} \exp \left[ - \left( \frac{x - m_b}{\sigma_b \sqrt{2}} \right)^2 \right] dx.$$ 

This proves Theorem 26.

Since we assumed independence between the statistics $\xi_n$, the above is only an approximation. Figure 5.1 illustrates the approximation formula given by Theorem 26 by comparing with simulation results.

**Proof**

**Setup:** For the following we use the notation $RAND(a, b)$ to denote a random variable having mean $a$ and variance $b$.

Assume without loss of generality that $\Omega = \{1, 2, \ldots, k\}$ for convenience of notation. Thus, the correct estimates are $j \leq k$, and the incorrect estimates are $j \geq k + 1$. Now consider the statistic $\xi_j$. This is the sample mean of $S$ i.i.d. variables. Hence, as $S$ grows large, $\xi_j$ will converge to $E[(y_i, \phi_{i,j})^2]$ by the Law of Large Numbers. Note that the random variable $(y_i, \phi_{i,j})^2$ has finite variance (that we compute below), and hence the Law of Large numbers can be applied. We can write

$$(y_i, \phi_{i,j})^2 = (\Phi_i x_i, \phi_{i,j})^2.$$ 

Note that $x_i$, $y_i$ and $\phi_{i,j}$ are column vectors of length $m$, and $\Phi_i$ is a matrix of dimension $m \times n$. Because $\Phi_i x_i = \sum_{\ell=1}^{n} \phi_{i,\ell} x_i(\ell)$, we can expand the RHS of the above equation as

$$(y_i, \phi_{i,j})^2 = \left( \sum_{\ell=1}^{n} \phi_{i,\ell} x_i(\ell), \phi_{i,j} \right)^2.$$
where \( x_i(\ell) \) refers to entry \( \ell \) in signal \( x_i \). The summation in the above equation computes the sum of \( n \) column vectors. Invoking the distributive law for the inner product, we can write

\[
\langle y_i, \phi_{i,j} \rangle^2 = \left( \sum_{\ell=1}^{n} x_i(\ell) \langle \phi_{i,\ell}, \phi_{i,j} \rangle \right)^2.
\]

Because the signal \( x_i \) is supported only on \( \Omega = 1, 2, \ldots, k \), we have \( x_i(\ell) = 0 \) if \( \ell \geq k+1 \). Hence,

\[
\langle y_i, \phi_{i,j} \rangle^2 = \left( \sum_{\ell=1}^{k} x_i(\ell) \langle \phi_{i,\ell}, \phi_{i,j} \rangle \right)^2.
\]
We now compute the expected value of $\langle y_i, \phi_{i,j} \rangle^2$ under two cases. In the first case, we consider $j \leq k$ (we call this the good statistics case), and in the second case, we consider $j \geq k + 1$ (bad statistics case).

**Good statistics:** Consider one of the good statistics, and choose $j = 1$ without loss of generality. We have

$$\langle y_i, \phi_{i,1} \rangle^2 = \left( x_i(1)\|\phi_{i,1}\|^2 + \sum_{\ell=2}^{k} x_i(\ell) \langle \phi_{i,\ell}, \phi_{i,1} \rangle \right)^2.$$  

The norm $\|\phi_{i,1}\|^2$ is $\chi^2$ with $m$ degrees of freedom and hence has mean $m$ and second moment $m(m + 2)$. Thus it follows that

$$x_i(1)\|\phi_{i,1}\|^2 = RAN D(0, m(m + 2)\sigma^2).$$

We have

$$\sum_{\ell=2}^{k} x_i(\ell) \langle \phi_{i,\ell}, \phi_{i,1} \rangle = \left\langle \phi_{i,1}, \sum_{\ell=2}^{k} x_i(\ell) \phi_{i,\ell} \right\rangle.$$  

The random variable $x_i(\ell)$ is $\mathcal{N}(0, \sigma^2)$ and $\phi_{i,\ell}$ is an $m$-length column vector of $\mathcal{N}(0, 1)$ entries; hence each entry of the vector $x_i(\ell)\phi_{i,\ell}$ is $RAN D(0, \sigma^2)$. The expression $\sum_{\ell=2}^{k} x_i(\ell)\phi_{i,\ell}$ is a sum of $(k - 1)$ such independent column vectors, and hence each of its entries are distributed $RAN D(0, (k - 1)\sigma^2)$. Because each entry of $\phi_{i,1}$ is $\mathcal{N}(0, 1)$, we have the distribution for $\langle \phi_{i,1}, \sum_{\ell=2}^{k} x_i(\ell)\phi_{i,\ell} \rangle$ as $RAN D(0, m(k - 1)\sigma^2)$.

Thus we have

$$x_i(1)\|\phi_{i,1}\|^2 + \sum_{\ell=2}^{k} x_i(\ell) \langle \phi_{i,\ell}, \phi_{i,1} \rangle = RAN D(0, m(k - 1)\sigma^2) + RAN D(0, m(m + 2)\sigma^2)$$

In the sequel, we show that $x_i(1)\|\phi_{i,1}\|^2$ and $\sum_{\ell=2}^{k} x_i(\ell) \langle \phi_{i,\ell}, \phi_{i,1} \rangle$ are uncorrelated, and hence the variance of their sum is the sum of their variance.
Define the random variables $A$ and $B_\ell$ as $A = x_i(1)\|\phi_{i,1}\|^2$ and $B_\ell = x_i(\ell)\langle \phi_{i,\ell}, \phi_{i,1} \rangle$, for some $\ell \in \{2, 3, ..., k\}$. We can write the product $AB_\ell$ as

$$AB_\ell = x_i(1)x_i(\ell)\|\phi_{i,1}\|^2\langle \phi_{i,\ell}, \phi_{i,1} \rangle$$

$$= |x_i(1)||x_i(\ell)||\|\phi_{i,1}\|\|\langle \phi_{i,\ell}, \phi_{i,1} \rangle|\text{sgn}(x_i(1))\text{sgn}(x_i(\ell))\text{sgn}(\langle \phi_{i,\ell}, \phi_{i,1} \rangle),$$

where $\text{sgn}(x)$ is the sign function given by

$$\text{sgn}(x) = \begin{cases} 
1 & \text{if } x > 0; \\
-1 & \text{if } x < 0; \\
0 & \text{otherwise.}
\end{cases}$$

The three $\text{sgn}(.)$ terms in Equation 5.4 are independent; furthermore, they are identically distributed with $\text{Probability}(\text{sgn}(.) = -1) = \frac{1}{2}$ and $\text{Probability}(\text{sgn}(.) = 1) = \frac{1}{2}$.

Let $C$ be the product of the three $\text{sgn}(.)$ terms, $C = \text{sgn}(x_i(1))\text{sgn}(x_i(\ell))\text{sgn}(\langle \phi_{i,\ell}, \phi_{i,1} \rangle)$. Clearly, $C$ has the distribution given by $\text{Probability}(C = -1) = \frac{1}{2}$ and $\text{Probability}(C = 1) = \frac{1}{2}$. From these arguments, it is easily seen that $E(AB_\ell) = 0$; in other words, $A$ and $B_\ell$ are uncorrelated. Consequently, $E(A \sum_{\ell=2}^{k} B_\ell) = 0$. We use the following result: for two zero mean uncorrelated random variables, the variance of their sum is the sum of their variances. Applying this result to the random variables $A$ and $\sum_{\ell=2}^{k} B_\ell$, we have

$$\text{variance}(A \sum_{\ell=2}^{k} B_\ell) = \text{variance}(A) + \text{variance}(\sum_{\ell=2}^{k} B_\ell),$$

and hence

$$x_i(1)\|\phi_{i,1}\|^2 + \sum_{\ell=2}^{k}x_i(\ell)\langle \phi_{i,\ell}, \phi_{i,1} \rangle = \text{RAND}(0, m(k - 1)\sigma^2 + m(m + 2)\sigma^2)$$

$$= \text{RAND}(0, (m + k + 1)m\sigma^2).$$

---

17 Because $x_i(1)$ and $x_i(\ell)$ are both $\mathcal{N}(0, \sigma^2)$ distributed, they have equal probability of being positive or negative. Also, the dot product of the two random vectors $\phi_{i,\ell}$ and $\phi_{i,1}$ are equally likely to be either positive or negative.
We have computed the variance of $\langle y_i, \phi_{i,1} \rangle$ and conclude that

$$\xi_1 = E[\langle y_i, \phi_{i,1} \rangle^2] = (m + k + 1)m\sigma^2.$$  

**Bad statistics:** Consider one of the bad statistics by choosing $j = k + 1$ without loss of generality. We have

$$\langle y_i, \phi_{i,k+1} \rangle^2 = \left( \sum_{\ell=1}^{k} x_i(\ell)\langle \phi_{i,\ell}, \phi_{i,k+1} \rangle \right)^2.$$  

Following similar arguments as before we quickly conclude that

$$\sum_{\ell=1}^{k} x_i(\ell)\langle \phi_{i,\ell}, \phi_{i,k+1} \rangle = \operatorname{RAND}(0,km\sigma^2).$$  

Thus we have computed the variance of $\langle y_i, \phi_{i,k+1} \rangle$ and conclude that

$$\xi_{k+1} = E[\langle y_i, \phi_{i,k+1} \rangle^2] = km\sigma^2.$$  

**Conclusion:** From these expected values we conclude that

$$\lim_{S \to \infty}\xi_j = \begin{cases} 
(m + k + 1)m\sigma^2, & j \in \Omega \\
km\sigma^2, & j \notin \Omega.
\end{cases}$$  

For any $m \geq 1$, these values are distinct, with a ratio of

$$\frac{m + k + 1}{k}$$

between them. In practice for sufficiently large $S$ we see two distinct clusters of values, so $k$ need not be known *a priori.*
Appendix A

Overview of Belief Propagation

BP solves inference problems by iteratively exchanging messages over the edges of the factor graph. BP computes approximately the marginal distribution of all the variables involved in the factor graph, conditioned on the observed data. Messages are propagated over the edges of the factor graph; the messages encode the probability distribution for the variable associated with that edge. We review the basic message passing and message processing operations involved in BP.

Consider a variable node $v$ in a factor graph and one of its neighbors, a constraint node $c$. Let $\mu_{v\rightarrow c}(v)$ denote the message sent from $v$ to $c$ in the operation of BP, and let $\mu_{c\rightarrow v}(v)$ denote the message sent from $c$ to $v$. Let $n(v)$ denote the set of neighbors of $v$, and $u(c)$ the set of neighbors of $c$.

Belief propagation, also called the sum-product algorithm, operates according to the following simple update rules: the message (belief) from a node $v$ on an edge $e$ is the product of all the messages received at $v$ on edges other than $e$. The message from node $c$ to $v$ is computed in a similar manner, but the constraint associated with $c$ is applied to the product of messages and the result is marginalized for the variable associated with $e$. In other words, the message computations performed by belief propagation is given by

$$\mu_{v\rightarrow c}(v) = \prod_{u\in n(v)\setminus\{c\}} \mu_{u\rightarrow v}(v)$$

and

$$\mu_{c\rightarrow v}(v) = \frac{1}{Z_c} \exp\left\{\sum_{u\in n(c)} \mu_{u\rightarrow c}(u) - \mu_{c\rightarrow u}(u)\right\}$$

where $Z_c$ is a normalization constant that ensures the messages are probability distributions.

(A.1)
$$\mu_{c\rightarrow v}(v) = \sum_{v \not\in \{v\}} \left( c(n(c)) \prod_{w \in n(c) \setminus \{v\}} \mu_{w\rightarrow c}(w) \right), \quad (A.2)$$

where $c(n(c))$ is the constraint on the set of variable nodes $n(c)$ (here the outer $c$ represents the constraint and the inner $c$ represents the node). The notation $\sim \{v\}$ represents the set of all nodes neighboring $c$ except $v$, and $n(v) \setminus \{c\}$ is the set of neighbors to $v$ excluding $c$.

BP enables the computation of the marginal distribution of all the variables, conditioned on the observed data. The marginal distribution for a given variable node is obtained by the product of all the most recent incoming messages along the edges connecting to that node:

$$\text{pdf}(v) = \prod_{u \in n(v)} \mu_{u\rightarrow v}(v). \quad (A.3)$$

**Protecting BP against loopy graphs and message errors:** The results of BP converge to the exact conditional distribution (function summaries) in the ideal situation where the following two conditions are met:

1) The factor graph is cycle-free, and
2) Messages are processed and propagated without errors.

In CS reconstruction with BP as described above, both these conditions are violated. In general, the factor graph is loopy. Also, our message propagation for CS involves message quantization because exact message representation is computationally intractable. These non-idealities may lead BP to converge to inexact function summaries, or more critically, lead BP to diverge [72, 38, 45]. We investigate the impact of these non-idealities and suggest damping strategies to protect BP.

BP may be used in factor graphs with cycles simply by following the same message update rules. In this scenario, the algorithm has no natural termination, with messages passed multiple times on a given edge. Furthermore, the result of BP oper-
ating in a factor graph with cycles cannot in general be interpreted as exact marginal distributions (even if the iterations converge). Despite this, the approximate nature of BP for graph with cycles is an acceptable price to pay for performing efficient inference. Indeed, BP is used in numerous applications in which the underlying factor graph does have cycles [38]. Examples of such applications include decoding of turbo codes [9] and LDPC codes [39], where BP achieves astonishing performance (within a fraction of a decibel of the Shannon limit on a Gaussian channel) [53]. The excellent performance of BP in loopy graphs motivates its use in CS reconstruction algorithms.

Furthermore, BP for CS has to deal with message quantization because exact message representation is computationally intractable. We incur message errors either by approximation by model order reduction of mixture Gaussian distributions, or by sampling the pdf. The message errors can propagate and can lead BP to diverge [45].

To protect BP in the face of loopy-graphs and message errors, we propose the use of damping schemes.

**Damping techniques for BP**: We provide a brief overview of techniques in the literature to combat divergence in the iterations of BP. Yuille proposed the Convex Concave Procedure – a class of discrete iterative algorithms that are provably convergent alternatives to BP [89]. The main idea is to obtain a discrete iterative algorithms by decomposing a cost function into a concave and a convex part. The crucial trick, is to bound the possibly concave part and solve the remaining convex problem. Heskes et al. [44] proposed the double-loop algorithm that is provably convergent for the minimization of Bethe and Kikuchi free energies [88] that represent the cost function of BP. Pretti and Pelizzola proposed a new propagation algorithm for the minimization of the cost function (Bethe free energy) for a generic lattice model with pair interactions [60]. The algorithm is shown to be more stable than belief propagation, as it reaches a fixed point also for highly frustrated systems and faster than
the provably convergent double loop algorithms. More recently, Pretti [59] proposed a modified version of BP with an over-relaxed BP dynamics, where, at each step of the algorithm, the evaluation of beliefs (messages) is taken to be a weighted average between the old estimate and the new estimate. The weighted average could either be applied to the messages (resulting in the message damped BP, or MDBP) or to the estimate of the probability distribution of the variables (probability damped BP, or PDBP). The author relates MDBP and PDBP to the double-loop algorithm of Heskes et al [44]; in particular, MDBP and PDBP are limiting cases of the double-loop algorithm. The connection between MDBP and PDBP to the double-loop algorithm sheds light about the way MDBP and PDBP work, and also provides directions for further enhancements.
Bibliography


