Distributed Data Aggregation for Sparse Recovery in Wireless Sensor Networks

A Thesis Presented for

The Master of Science

Degree

The University of Tennessee, Knoxville

Shuangjiang Li

December 2011

© by Shuangjiang Li, 2011
All Rights Reserved.

Acknowledgements

I would like to extend my sincere gratitude and appreciation to all the individuals who made this thesis possible.

First and foremost, I would like to thank my advisor, Dr. Hairong Qi for assisting and encouraging me and for always being there to lend a helping hand in my research. Without her excellent guidance and support, this work would not have been possible.

I also want to give thanks to Dr. Li and Dr. Cao. I greatly appreciate their time and input to this Thesis. I would like to give my special thanks to my parents, my parents-in-law, my sister, and my wife, who provide the encouragement and the strongest support during my study life in every possible way.

Finally, I thank all the members in the AICIP Lab for their useful advise and suggestions presented at AICIP group meetings.

Abstract

We consider the approximate sparse recovery problem in Wireless Sensor Networks (WSNs) using Compressed Sensing/Compressive Sampling (CS). The goal is to recover the n-dimensional data values by querying only $m \ll n$ sensors based on some linear projection of sensor readings. To solve this problem, a two-tiered sampling model is considered and a novel distributed compressive sparse sampling (DCSS) algorithm is proposed based on sparse binary CS measurement matrix. In the two-tiered sampling model, each sensor first samples the environment independently. Then the fusion center (FC), acting as a pseudo-sensor, samples the sensor network to select a subset of sensors (m out of n) that directly respond to the FC for data recovery purpose. The sparse binary matrix is designed using unbalanced expander graph which achieves the state-of-the-art performance for CS schemes. This binary matrix can be interpreted as a sensor selection matrix-whose fairness is analyzed. Extensive experiments on both synthetic and real data set show that by querying only the minimum amount of m sensors using the DCSS algorithm, the CS recovery accuracy can be as good as dense measurement matrices (e.g., Gaussian, Fourier Scrambles). We also show that the sparse binary measurement matrix works well on compressible data which has the closest recovery result to the known best k-term approximation. The recovery is robust against noisy measurements. The sparsity and binary properties of the measurement matrix contribute, to a great extent, the reduction of the in-network communication cost as well as the computational burden.

Contents

List of Tables vi

List of Figures vii

1 Introduction 1

1.1 General Sampling model 1

1.2 Motivations 2

1.3 Contribution 3

1.4 Related Work 3

1.5 Outline of the Thesis 5

2 Background on Compressed Sensing 7

2.1 Theory of Compressed Sensing 7

2.1.1 RIP Property in ℓ_2-norm 8

2.1.2 RIP Property in ℓ_1-norm 10

2.2 Sparse Recovery Algorithms 10

2.2.1 Direct optimization 11

2.2.2 General unconstrained optimization 12

2.2.3 Homotopy 13

2.2.4 Combinatorial approach 14

3 Problem Setup 15

3.1 Two Tier Sampling Model 15

3.2 Sparse Signal Classes . . . 16
3.2.1 k-Sparse Signal . . . 16
3.2.2 Compressible Signal . . . 17
3.3 The Sensor Subset Selection Problem . . . 18
3.4 Sparse Recovery Problem . . . 18

4 Graphs and Sparse Measurement Matrices **20**
4.1 Bipartite Graph . . . 20
4.2 Expander Graph . . . 23
4.2.1 Some Facts on Expander Graphs . . . 24
4.3 Connection between Expander Graph and Compressed Sensing Matrices . . . 26
4.4 Design of Sparse Measurement Matrices . . . 29

5 Distributed Compressive Sparse Sampling Algorithm **31**
5.1 Distributed Compressed Sensing Algorithm . . . 31
5.2 Concern on the Fairness of the Subset Selection Process . . . 34

6 Experiments **35**
6.1 Experiment using Synthetic Data . . . 36
6.1.1 Exact Sparse Signal Recovery . . . 36
6.1.2 Noisy Sparse Signals Recovery . . . 36
6.1.3 Compressible Signal Recovery . . . 38
6.2 Experiment using Real Data . . . 41

7 Conclusions and future work **44**
7.1 Conclusions . . . 44
7.2 Future Work . . . 45

Bibliography **46**

Vita **56**

List of Tables

6.1 Signal to Noise Ratio (SNR) of different measurement matrices. 41

List of Figures

2.1 The relations between the combinatorial and geometric approaches and the two main classes of decoding algorithms [6] 14

3.1 Two tier sampling model 16
3.2 The power-law decay curve and k-term approximation. 17
3.3 An Example of Sparse Recovery 19

4.1 An example of a bipartite graph. 21
4.2 An example of the adjacency matrix of a graph. 22
4.3 An unbalanced expander graph: any sufficiently small subset X on the left has a neightborhood $N(X)$ of size at least $(1-\epsilon)d|X|$ 24
4.4 An Example of 200×1024 binary sparse matrix with degree $d = 8$ 30

5.1 Sensor readings projection can be seen as generating a bipartite graph between the n data nodes and the m encoding nodes. 32
5.2 An example of observations from sensor data projection using sparse binary matrix. 32

6.1 An example of an exact sparse signal with $n = 1024$ and sparsity $k = 30$. . 36
6.2 Recovery result of an n = 1024, sparsity k=30 sparse signal x, with an average of 100 experiments using LP recovery method. 37
6.3 Noisy recovery result of an n = 1024, sparsity k=30 sparse signal x, with different SNRs and an average of 100 experiments using LP recovery method and $m = 200$ measurements. 38

6.4 Noisy recovery result of an n = 1024, sparsity k=30 sparse signal x, with different SNRs and an average of 100 experiments using LP recovery method and $m = 700$ measurements. 39
6.5 Noisy recovery result of an n = 1024, sparsity k=30 sparse signal x, with different SNRs and an average of 100 experiments using LP recovery method evaluated by different measurements. 39
6.6 Recovery result of an sampled compressible signal $x = 4n^{-\frac{7}{10}}$, with an average of 100 experiments using LP recovery method evaluated by different measurements. 40
6.7 Recovery results comparison of different methods. 40
6.8 Recovery result of real Intel Lab signal using 100 wavelet coefficients and 400 CS measurements with different measurement matrices. 42
6.9 Recovery result of real Intel Lab signal, with an average of 100 experiments using LP recovery method evaluated by different measurements. 43
6.10 Recovery results comparison of different methods. 43

Chapter 1

Introduction

This thesis explores the application of Compressed Sensing/Compressive Sampling (CS) to the sensor selection for data recovery problem in Wireless Sensor Networks (WSNs). In Section 1.1, we will outline our mainly contribution in solving this problem. Next, in Section 1.2 we briefly review the general sampling model for data recovery and a specific two-tiered sampling model will be described later in Chapter 2. In Section 1.3 we discuss the applications of data recovery in WSN, the same problem has also been proposed in [44]

1.1 General Sampling model

The Shannon/Nyquist sampling theorem tells us that in order to not lose information when uniformly sampling a signal we must sample at least two times faster than its bandwidth. Consider a real-valued, finite-length, one-dimensional, discrete-time signal x, which we view as an $n \times 1$ column vector in $\mathbb{R}^n$ with elements $x[n], n = 1, 2, \cdots, n$. We treat an image or higher-dimensional data by vectorizing it into a long one-dimensional vector. More often, people use transform coding in data acquisition systems like digital cameras where the number of samples is high but the signals are compressible. In this framework, we acquire the full $n-s$ample signal x; compute the complete set of transform coefficients $\{s_i\}$ via $s = \Psi^T x$; locate the k largest coefficients and discard the $(n-k)$ smallest coefficients; and encode the k values and locations of the largest coefficients. (In practice,

we also convert the values and locations to digital bits.) Unfortunately, the sample-then-compress framework suffers from three inherent inefficiencies: First, we must start with a potentially large number of samples n even if the ultimate desired k is small. Second, the encoder must compute all of the k transform coefficients $\{s_i\}$, even though it will discard all but k of them. Third, the encoder faces the overhead of encoding the locations of the large coefficients.

Instead of thinking in the traditional way, compressive sensing promises to recover the high-dimensional signals exactly or accurately, by using a much smaller number of non-adaptive linear samplings or measurements. In general, signals in this context are represented by vectors from linear spaces, many of which in the applications will represent images or other objects. However, the fundamental theorem of linear algebra,"as many equations as unknowns," tells us that it is not possible to reconstruct a unique signal from an incomplete set of linear measurements. As we said before, many signals such as real-world images or audio signals are often sparse or compressible over some basis, such as smooth signals or signals whose variations are bounded. This opens the room for recovering these signals accurately or even exactly from incomplete linear measurements. Some background on Compressed Sensing will be address in Chapter 2.

1.2 Motivations

In a typical wireless sensor network measuring an environmental reading, each with finite battery lifetime and thus limited computing and communication capabilities. In order to gain an understanding of the environment, information processing among the sensor nodes is of ultimate important. How can we collaboratively communicating within the network and efficiently encoding and sending the data with constrained energy and limited bandwidth? Or in other words, how can we select subset of sensors for the data approximation in WSNs. In collaborative processing, most of time the functions of data (i.e., sums, averages, mean etc) are communicated. The limited bandwidth and

computational capabilities mandate that there should be a trade-off between how much data communicated and the accuracy of the data approximation.

1.3 Contribution

The main contributions of this paper are as follows:

Firstly, a two-tiered sampling model is designed where each individual sensor first samples the environment to obtain sensor readings and then the Fusion Center (FC), acting as a pseudo sensor, samples the sensor network based on compressive sampling, to obtain a subset of sensors (m out of n) for CS recovery. Thus the traditional problem that sensing will cause data deluge is alleviated.

Secondly, a Distributed Compressive Sparse Sampling (DCSS) algorithm that efficiently selects and aggregates the sensor reading is developed. FC chooses m designated sensors within the network based on the CS measurement matrix, which provides a necessary number of m measurements for accurate data recovery. The distributed algorithm enjoys the sparsity of the measurement matrix that reduces the communication cost and relieves the computational burden of the network.

Lastly, a sparse binary matrix based on unbalanced expander graph is proposed. Although such measurement matrix has been developed for network streaming and only recently in compressed sensing [49], this paper is the first attempt in designing the sparse binary matrix with a fixed number of 1's in each column of the measurement matrix such that it can be applied in WSNs for data recovery purpose. Theoretical analysis shows the guaranteed sparse recovery result.

1.4 Related Work

Data aggregation has been put forward as an essential paradigm for wireless routing in sensor networks. An important topic addressed by the WSN community over the last several years has been in-network aggregation. The notion is that cumulants, moments,

or summaries can be computed directly in the network by sensor nodes endowed with computational power, thus avoiding the expensive transmission of all sensor data to the FC [52, 9, 51, 39, 72]. For example, [55] used a tree-like structure to disseminate queries and collect results back to the FC. Various algorithmic techniques have been proposed to allow efficient aggregation without increasing the message size [27, 28]. While most of the existing work assumes that the goal of the aggregation is to accumulate data from all the network nodes, [43] studied the problem of aggregating data from a sparse set of nodes in WSNs. A protocol that simultaneously integrates the data aggregation process with the sparse tree formation process was designed where data aggregation can be conducted by having the data flow up the tree and aggregated in the standard way. Our goal in this paper is to aggregate data in-network with a minimum number of sensor nodes for accurate recovery. Each node knows locally whether it needs to be aggregated or not based on the designed sparse binary matrix. An interesting aspect of our method is that we can actually integrate the data aggregation process with the sensor subset selection process so that they both happen simultaneously.

An overview of several sparse random matrices and algorithms for sparse recovery problems can be found in [45]. The majority of existing work on sparse recovery in wireless sensor networks is based on the sparse random projection matrix Φ designed in [53]. In [67], the authors showed that this kind of random matrices can be successfully adopted to approximate functions of data based on their distributed sensing algorithms. The same matrix was also proposed in [59] for signal approximation to facilitate energy neutral operation in rechargeable WSNs. In the area of DNA Microarrays, sparse matrices were used for probe design and CS signal recovery. [57] proposed an algorithm called sparse matrix preprocessing (SMPP), which uses low-density parity check (LDPC) code generator matrices. The LDPC-like matrices are very sparse, with constant number of 1's in each row and column. The original spare signal can be recovered then by linear programming. The Compressive Sensing Microarrays (CSM) in [30] designed the probes in such a way that they mimic the Hamming code and signal recovery is performed through belief propagation. Also [31] established a mathematical connection between

channel coding and compressed sensing, where the binary linear channel code can be seen as good zero-one measurement matrix, thus translating the performance guarantees from information codes to CS. Wu et al. [69] employed a permutation-based multi-dimensional sparse measurement matrix, which is composed of several sub-matrices, each consisting of a block-diagonal matrix and a random permutation matrix. It has been shown that such a measurement matrix brings some useful features to the measurement symbols.

There is also a rich literature in distributed compressive sensing [5, 38, 2, 67]. For example, [5] defined a joint sparsity model on the data, and exploited both intra- and inter-signal correlation structures to reduce communications from the sensors to the FC. [2] required no prior knowledge about the sensed data and used distributed matched source-channel communication to deliver random projections of the sensor network data to the FC. Both [5, 2] assume the scenario where all sensors communicate directly to a central FC, without any in-network communication. [67] proposed that every sensor in the field computes and stores sparse random projections in a decentralized manner and sends its aggregates randomly within the network. The FC can then query any number of sensors to achieve refinable approximation. Compared to these existing works, our approach utilizes in-network communications and uses sparse binary projections for data aggregates, which reduces the data communication cost and storage burden. In addition, the projected data is only aggregated to the m designated sensors which, in contrast to [67], reduces communication overhead as well as facilitates easy measurement aggregation.

1.5 Outline of the Thesis

The outline of the thesis is as follows:

In chapter 1, we review the general sampling model for signal processing and introduce the data recovery problems in the application of Wireless Sensor Networks. We also address the motivation of this thesis and related works that have been done in compressed sensing, sparse measurement matrices design, and distributed data recovery in WSNs. At the end of the chapter, we give the synopsis of the thesis.

In chapter 2, we review some background knowledge on Compressed Sensing contains the theories of measurement matrices and different variant of recovery algorithms.

In chapter 3, we briefly discuss the problem setup throughout the thesis. We first introduce the two-tiered sampling model we will use in this thesis. We also review two classes of sparse signals, namely exact sparse signals and compressible signals, the problem of sparse recovery will be addressed here too.

In chapter 4, we investigate the connection between graphs and sparse measurement matrices by first introducing the definition and theoretic results of bipartite graphs and expander graphs, we then give proves showing that the results generated in graph codes can be adjusted to fit the compressed sensing technique. Finally, we presented a sparse measurement matrices design procedure using the adjacency matrix of the expander graph.

In Chapter 5, we propose the distributed compressive sparse sampling (DCSS) algorithm. Some practical implementation issues and the fairness of the sensor selection problem are also discussed.

In Chapter 6, we conduct various settings of experiments and tested the sparse binary matrix and our distributed algorithms on both synthetic and real data sets. The results shows the sparse binary matrix can do well on both exact sparse signal and compressible signals compared to traditional dense Gaussian and Fourier matrices.

Finally, in Chapter 7, we conclude our thesis and provide some future works.

Chapter 2

Background on Compressed Sensing

The basic CS problem is to estimate a vector $x \in \mathbb{R}^n$ from a set of linear measurements $y = \Phi x$, where $y \in \mathbb{R}^m$ and Φ is a known $m \times n$ matrix. This method greatly reduces the number of digital samples required to reconstruct from highly incomplete information, typically well below the number expected from the requirements of the Shannon/Nyquist sampling theorem. The key idea in CS is that if the signal x is constrained to be sparse or approximately sparse, then it is possible to recover x even when $m \ll n$. More precisely, one of the basic results in CS is that there exist matrices Φ with only $m = \mathbb{O}(k \log(n/k))$ rows such that for all k-sparse x, i.e., all x with at most k nonzero components, we can recover x exactly from Φx. Furthermore, [32] and [18, 19] observed that recovery can be accomplished in polynomial time via linear programming (LP), provided that the measurement matrix Φ satisfies certain technical conditions. This chapter gives a basic introduction in compressive sensing with current theories and properties.

2.1 Theory of Compressed Sensing

Without loss of generality, we consider a real-valued, finite-length, one-dimensional, discrete-time signal x ,which can be viewed as an $n \times 1$ column vector in $\mathbb{R}^n$. Since any signal in $\mathbb{R}^n$ can be represented in terms of a basis of $n \times 1$ vectors $\{\psi\}_{i=1}^n$. For simplicity,

assume that the basis is orthonormal. Using the $n \times n$ basis matrix $\Psi = [\psi_1|\psi_2 \cdots |\psi_n]$ with the vector ψ_i as columns, the signal x can be expressed as [4]

$$x = \sum_{i=1}^{n} s_i \psi_i \qquad or \qquad x = \psi s \tag{2.1}$$

where s is the $n \times 1$ column vector of weighting coefficients. $s_i = \psi_i^T x$. Clearly, x and s are equivalent representations of the signal, with x in the time or space domain and s in the ψ domain.

To begin with, we define that the signal is k-sparse if it is a linear combination of only k basis vectors, that is, only k of the s_i coefficients in (2.1) are nonzero and $(n-k)$ are zero. We call the set of indices corresponding to the nonzero entries the *support* of x and denote it by supp(x). The set of all k-sparse signals is the union of the $\binom{n}{k}$, k-dimensional subspaces aligned with the coordinate axes in $\mathbb{R}^n$. We denote this union of subspaces by $\sum_k$. The case of interest is when $k \ll n$. The signal is *compressible* if the representation in Equation (2.1) has just a few large coefficients and many small coefficients.

Now we will begin the CS problem statement in the following subsections.

2.1.1 RIP Property in ℓ_2-norm

CS integrates the signal acquisition and compression steps into a single process. In CS we do not acquire x directly but rather acquire $m < n$ linear measurements $y = \Phi x$ using an $m \times n$ measurement matrix Φ. We then recover x by exploiting its sparsity or compressibility. Our goal is to push m as close as possible to k in order to perform as much signal "compression" during acquisition as possible.

The measurement matrix Φ is very important since it largely affects the process of sampling the system as well as determines how good the recovery is. We could also consider the measurement matrix as a *sampling operator* that preserves some suitable

conditions under which the signal can be recovered exactly or with high probability. Recent years, a lot of good papers have already advanced the research on this topic [14, 3].

One of the most commonly used conditions for the measurement matrix is the *Restricted Isometry Property (RIP)* introduced by Candes and Tao [23]. RIP essentially requires that every subset of column of Φ with certain cardinality approximately behaves like an orthonormal system. For an $m \times n$ matrix Φ and an integer $k, 1 \ll k \ll p$, the measurement matrix Φ should satisfy the restricted isometry property.

Definition: An $m \times n$ matrix Φ has the k-restricted isometry property (k-RIP) with constant $\delta_k > 0$ if, for all $x \in \sum_k$,

$$(1 - \delta_k)\|x\|_2^2 \leq ||\Phi x||_2^2 \leq (1 + \delta_k)\|x\|_2^2 \tag{2.2}$$

In other words, the k-RIP ensures that all submatrices of Φ of size $m \times k$ are close to an isometry, and therefore distance (and information) preserving. Practical recovery algorithms typically require that Φ have a slightly stronger $2k$-RIP, $3k$-RIP, or higher-order RIP in order to preserve distances between k-sparse vectors (which are $2k$-sparse in general), three-way sums of k-sparse vectors (which are $3k$-sparse in general), and other higher-order structures.

Other measurement matrix conditions have been proposed for exactly recovery, like Donoho and Tanner's K-neighborly polytopes [36], expansion/randomness extraction property by Xu, et al. [71].

It is important to note that RIP conditions are difficult to verify for a given matrix Φ. A widely used technique for avoiding checking the RIP directly is to generate the matrix randomly and to show that the resulting random matrix satisfies the RIP with high probability using the well-known Johnson-Lindenstrauss Lemma. See, for example, Baraniuk, et al. [3]. This is typically done for conditions involving only the restricted isometry constant δ_k.

2.1.2 RIP Property in ℓ_1-norm

While most of the researches have been focusing on the dense matrices satisfy RIP property in ℓ_2-norm. The measurement matrix Φ that is sparse and often binary has also gained some attention. In 2008, Berinde et al [6] showed that the adjacency matrix of an expander graph satisfies a very similar property, called the restricted isometry property in the ℓ_1-norm (the so-called RIP-1 property). They used this property to show that exact recovery by basis pursuit is still possible.

2.2 Sparse Recovery Algorithms

Sparse representations have attracted a great deal of attention in signal processing and information theory [20, 15, 33, 34]. Since there are infinitely many signal coefficient vectors x' that produce the same set of compressive measurements $y = \Phi x$, to recover the "right" signal we exploit our a priori knowledge of its sparsity or compressibility. A naive approach for solving this problem is to seek the sparsest x that agrees with the measurements y:

$$\hat{x} = \arg\min_{x'} ||x||_0 \qquad s.t. \qquad y = \Phi x' \tag{2.3}$$

where the l_0 norm of a vector counts its number of nonzero entries. And essentially the l_0 minimization is to find the sparest solution in the feasible set of possible solutions. While this optimization can recover a k-sparse signal from just $m = 2k$ compressive measurements, it is unfortunately a combinatorial, NP-hard problem and thus is computationally infeasible; furthermore, the recovery is not stable in the presence of noise. We practically consider a similar program in the l_1 norm which goes by the name of *Basis Pursuit* [26].

$$\hat{x} = \arg\min_{x'} ||x||_1 \qquad s.t. \qquad y = \Phi x' \tag{2.4}$$

where we recall that $||x||_{l_1} = \sum_i |x_i|$. l_1 minimization can be viewed as a convex relaxation of l_0 minimization. A series of beautiful articles [63, 46, 35, 11] showed exact equivalence between the two optimization programs, Equations 2.3 and Equations 2.4. Also in [23] the author Candes and Tao pointed out the two optimization programs essentially have the same solution, provided that the RIP is satisfied. l_1 minimization for compressive sensing has been successfully used as an effective way for reconstructing a sparse signal in many settings. See, e.g., [13, 22, 23, 24].

Dealing with additive noise in y or x, the problem transform into:

$$\hat{x} = \arg\min_{x'} ||x||_0 \qquad s.t. \qquad ||\Phi x' - y||_2^2 \ll \varepsilon \tag{2.5}$$

Where ε is arbitrary and unknown errors. Thanks to the active researchers in the areas of Signal Processing, Machine Learning and Computer Vision. A lot of efficient and tractable algorithms were presented to solve Equation 2.5. In essence, they can be divided into following categories:

2.2.1 Direct optimization

In order to find a sparse solution (i.e., min l_0 norm) of Equation 2.5, these kinds of algorithm envision a decomposition of a signal x as in Equation 2.1 or an approximate decomposition

$$x = \sum_{i=1}^{n} s_i \psi_i + r^{(m)} \tag{2.6}$$

where $r^{(m)}$ is a residual. We start from here to call ψ_i the dictionary as it is a common word in signal processing community (i.e., the wavelet and Fourier dictionary is the best known). It is well known that most of the new dictionaries are *overcomplete*, either because they start out that way, or because we merge complete dictionaries, obtaining a new mega-dictionary consisting of several types of waveform (e.g. Fourier & Wavelets dictionaries) . The decomposition of Equation 2.1 is then nonunique, because some elements in the dictionary have representations in terms of other elements. This kind of nonuniqueness

gives us the possibility of adaptation, i.e., of choosing among many representations one that is most suited to our purposes of which *Sparsity* and *Super Resolution* are highly demanded.

From all the above, one would find the connection between sparse representation* and compressing sensing as they all do the job of finding as sparse as possible the basis to represent the signal and trying to decompose to recover the exact signal with few significant coefficients. However, CS differs from sparse representation, the later is a sample-then-compress framework meaning that we first acquire a sufficient data sample and then we apply a compression algorithm on the data while CS as like its name directly acquiring a compressed signal representation.

Several methods have been proposed for solving Equation 2.6. They are typical call greedy algorithms ranging from general approaches like the method of Matching Pursuit [56], to clever schemes derived for specialized dictionaries, like Orthogonal Matching Pursuit [64]. Elsewhere about the algorithm, [26] gives explanation in more depth.

2.2.2 General unconstrained optimization

The CS problem falls into the unconstrained optimization problem of the form

$$\min_{x} \Phi(x) := f(x) + \tau c(x), \tag{2.7}$$

where f: $\mathbb{R}^{\aleph} \to \mathbb{R}$ is a smooth function, and c: $\mathbb{R}^{\aleph} \to \mathbb{R}$, usually called the *regularizer* or *regularization function*, is finite for all $x \in \mathbb{R}^{\aleph}$. Problem 2.7 generalizes the now famous $l_2 - l_1$ problem (called *basis pursuit denoising (BPDN)* in [26])

$$\min_{x \in \mathbb{R}^{\aleph}} \frac{1}{2}||y - \Phi x||_2^2 + \tau||x||_1 \qquad (\tau > 0) \tag{2.8}$$

Problem 2.8 is closedly related to the following two formulations:

*namely basis pursuit in signal processing community and l_1 regularization in statistical community

$$\min_x ||y - \Phi x||_2^2 \qquad s.t. \qquad ||x||_1 \leq K \tag{2.9}$$

frequently referred to as the *least absolute shrinkage and selection operator* (LASSO) [62], and

$$\min_x ||x||_1 \qquad s.t. \qquad ||y - \Phi x||_2^2 \leq \varepsilon \tag{2.10}$$

The precise relationship between Problem 2.8, Problem 2.9, and Problem 2.10 is discussed in [41] and [66], for example.

Also, the *Dantzig Selector* has the form

$$\min_x ||x||_1 \qquad s.t. \qquad ||\Phi^*(\Phi x - y)||_\infty \leq \gamma \tag{2.11}$$

where γ is a user specified parameter, relaxes the Equation 2.4 in a different way. Equation 2.11 requires that the residual $(\Phi x - y)$ of a candidate vector x not be too correlated with any of the columns of Φ (the product $\Phi^*(\Phi x - y)$ measures each of these correlations) [21].

Soft thresholding based algorithms are most popular in dealing with these kinds of formulation, like coordinate descent [42, 70], Bregman iterative algorithms [73], Fast Linearized Bregman Iteration [12], SpaRSA [68], GPSR [41] and log-barrier algorithm [10]. Equation 2.4 can be recast as an LP, a primal-dual algorithm for linear programming would solve this [17].

2.2.3 Homotopy

Let reconsider Equation 2.8, if the Lagrangian multiplier $\tau > 0$ is also a known parameter. The optimization problem could be formulated as

$$\min_{x \in \mathbb{R}^\aleph, \tau} \frac{1}{2}||y - \Phi x||_2^2 + \tau||x||_1 \qquad (\tau > 0) \tag{2.12}$$

Homotopy continuation methods could be used in this setting, for example [47, 37, 65].

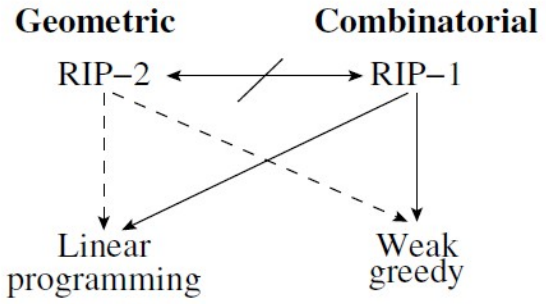


Figure 2.1: The relations between the combinatorial and geometric approaches and the two main classes of decoding algorithms [6] .

2.2.4 Combinatorial approach

In the combinatorial approach, the measurement matrix Φ is sparse and oftern binary. Typically, it is obtained from an adjacency matrix of a sparse bipartite random graph. The recovery algorithm proceeds by iteratively identifying and eliminating large coefficients1 of the vector x. The identification uses nonadaptive binary search techniques. Examples of combinatorial sketching and recovery algorithms include [29, 48, 71, 60]. The typical advantages of the combinatorial approach include fast recovery (often sub-linear in the signal length n if $k \ll n$), as well as fast and incremental (under coordinate updates) computation of the observation vector Φx. In addition, it is possible to construct efficient (albeit suboptimal) measurement matrices explicitly, at least for simple type of signals. For example, it is known [48, 71], how to explicitly construct matrices with $k2^{(\log\log n)O(1)}$ measurements, for signals x that are exactly k-sparse. The main disadvantage of the approach is the suboptimal measurement length. Figure 2.1 summarizes the relationship between the combinatorial and geometric approaches for sparse signal recovery.

Chapter 3

Problem Setup

We consider a wireless sensor network of n sensors, each of which measures a real data value x_i. Suppose the signal is compressible, so that it can be well approximated using $k \leq n$ coefficients of some orthonormal transform. For simplicity, we assume that each sensor stores one real value. At last, we want to be able to query only m designated sensors and recover an approximation of the n data values, with reconstruction error comparable to the best k-term approximation.

3.1 Two Tier Sampling Model

Consider a wireless network of n sensors, all connected through 1-hop wireless communication, each measures a real data value x_i, $i = 1, 2, \cdots, n$, which is k-sparse or compressible. The network topology is modeled as an undirected graph G where $G = (V, E)$. V represents the nodes in the network and E represents an edge which connects two nodes if they are within radio transmission range of each other.

Our distributed compressive sparse sampling (DCSS) algorithm is based on a two-tiered sampling model which described in Figure 3.1. In tier one each sensor unit samples the environment to obtain the discrete data readings x_i. In tier two the FC, acting as a pseudo sensor, samples the entire sensor networks using compressive sparse sampling to obtain (or

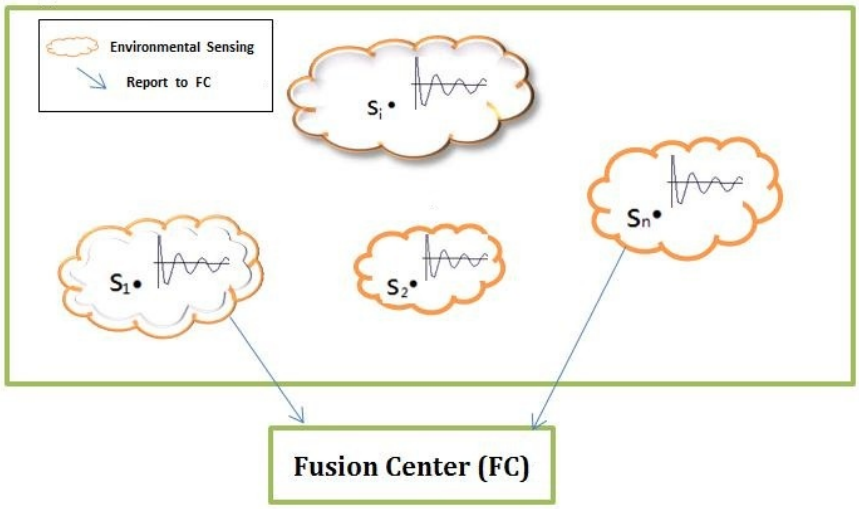


Figure 3.1: Two tier sampling model

select) m designated sensors which serve as aggregators and directly respond to queries sent from the FC.

3.2 Sparse Signal Classes

We categorize the general signals used in this thesis into two classes.

3.2.1 k-Sparse Signal

We say that a n-dimensional signal x is k-sparse if it has k or fewer non-zero components:

$$x \in \mathbb{R}^n,\ ||x||_0 := |\text{supp}(x)| \leq k \ll m,$$

where $|\text{supp}(x)|$ denotes the cardinality of the support set of x, and thus $||x||_0$, namely the number of non-zero components, is a quasi-norm.

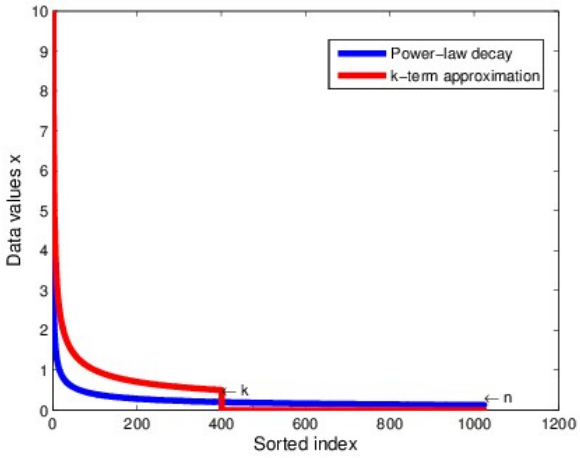


Figure 3.2: The power-law decay curve and k-term approximation.

3.2.2 Compressible Signal

We consider a real data vector $x \in \mathbb{R}^n$, and fix an orthonormal transform $\Psi \in \mathbb{R}^{n\times n}$ consisting of a set of orthonormal basis vectors $\{\psi_1, \cdots, \psi_n\}$. Ψ can be, for example, a Wavelet or a Fourier transform. The transform coefficients $\Theta = [\psi_1^T x, \cdots, \psi_n^T x]^T$ of the data can be ordered in magnitude, so that $|\theta_1| \geq |\theta_2| \geq \cdots \geq |\theta_n|$. The best k-term approximation keeps the largest k transform coefficients and discards the remaining as zero. The approximation error is $||x - \hat{x}||_2^2 = ||\Theta - \hat{\Theta}||_2^2 = \sum_{i=k+1}^{n} ||\theta_i||^2$.

According to CS literature [40, 32], the data is compressible if the magnitude of its transform coefficients decay like a power law. That is, the ith largest transform coefficient satisfies $|\theta_i| \leq Li^{-\frac{1}{p}}$, where L is some positive constant, and $0 < p \leq 1$. Note that p controls the compressibility (or rate of decay) of the transform coefficients (i.e., smaller p implies faster decay). The approximation error , is then

$$||x - \hat{x}||_2 = ||\Theta - \hat{\Theta}||_2 \leq \alpha_p L k^{-1/p+1/2}$$

where α_p is a constant that only depends on p. Of course, sparse signals are special cases of compressible signals. Figure 3.2 shows the power-law decay curve and k-term approximation with respect to the sorted index of the coefficients.

3.3 The Sensor Subset Selection Problem

By definition, the sensor subset selection problem is to choose a set of m sensor measurements, from a set of n possible or potential sensor measurements. Solving this problem by evaluating the performance for each of the C_n^m possible choices of sensor measurements is not practical unless m and n are small. Broadly speaking, this problem belongs to traditional feature selection problem.

A large class of algorithms have been developed that search for optimal solutions (e.g., exhaustive search, branch and bound, genetic algorithm) and deterministic suboptimal solutions (e.g., sequential forward selection, sequential backward selection). We now consider the zero-one matrix with elements $\{\phi_{ij}\}$ and the feature set F with elements, $f_j, j = 1, 2 \cdots, n$. Defining $\phi_{ij} = 1$ if we select the set element f_j and $\phi_{ij} = 0$ otherwise. In the sensor subset selection problem, we view $\sum_{i=1}^{m} \phi_{ij} f_j$, $j = 1, 2, \cdots, n$ as the selection of m sensor measurements from totally n number of sensors. Thus the sparse binary measurement matrix naturally severs as a sensor selection matrix.

3.4 Sparse Recovery Problem

In Chapter 2 Section 2.2 we introduce some flavors on sparse recovery algorithms. Figure 3.3 shows the problem of estimating an unknown signal based on m noisy observations, when the number of samples m is much less than the ambient signal dimension n. This is typically sparse signal recovery problem given that the signal model is either exact k-sparse or compressible.

We can then view the sparse recovery problem as the recovery of the k-sparse signal $x \in \mathbb{R}^n$ from its measurement vector $y = \Phi x \in \mathbb{R}^m$.

If a matrix Φ satisfies certain RIP property, then the recovery process can be accomplished by finding a vector $x^\star$ using the following linear program:

$$\min ||x^\star||_1 \quad \Phi x^\star = \Phi x. \qquad \text{(Sparse Recovery)}$$

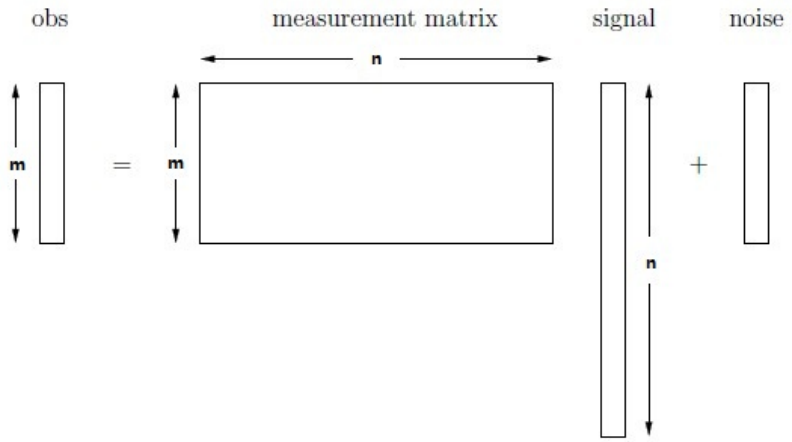


Figure 3.3: An Example of Sparse Recovery

A well-studied phenomenon in signal processing is that many natural classes of signals, such as smooth signals with bounded derivatives and bounded variation signals, are compressible in some transform domain [61]. Sensor networks measuring a smooth temperature field, for example, may efficiently represent the data using only a few large transform coefficients, which record useful structure such as average temperature and sharp temperature changes. The main goal in this thesis is to recover a vector $\hat{x}$ corresponds to the physical phenomenon recorded by the senor nodes such that the ℓ_p approximation error $||x - \hat{x}||_p$ is at most $C > 0$ times the smallest possible ℓ_q approximation error $||x - x^\star||_q$, where $x^\star$ ranges over all $k-$sparse vectors. That is,

$$||x - \hat{x}||_p \leq C \cdot \inf ||x - x^\star||_q$$

For the algorithms given in this thesis we have $p = q$.

Chapter 4

Graphs and Sparse Measurement Matrices

In this chapter, we explore the use of sparse graph codes for compressed sensing measurement matrices design. Compressed Sensing has many properties in common with coding theory. The recovery algorithm is similar to the decoding algorithms of error correcting codes but over $\mathbb{R}^m$ instead of a finite field. As a result, several results from coding theory have been generalized to fit compressed sensing techniques. In 4.1, we formally gives the definition of Bipartite Graph. We then move on to the definition and theory about Exapnder Graphs in 4.2 and the connections between the Expander Graph and Compressed Sensing are presented in 4.3, we show that high-quality expander graph can yield matrices with RIP-p property. Finally, we present our sparse measurement matrices design procedure using the adjacency matrix of the expander graph in 4.4.

4.1 Bipartite Graph

A graph is an ordered pair $G = (V, E)$ comprising a set V of vertices or nodes together with a set E of edges or lines. The edges may be directed (asymmetric) or undirected

(symmetric), which corresponding to directed graph and undirected graph. We begin by first give a formal definition of the Bipartite graph:

Definition: A bipartite graph $G = (U, V, E)$ is a graph whose vertices can be divided into two disjoint sets U and V such that every edge in E connects a vertex in U to one in V; that is, U and V are independent sets. Denote m and n respectively the cardinality of set U and set V (i.e., $|U| = m, |V| = n$). See Fig. 4.1 for an example.

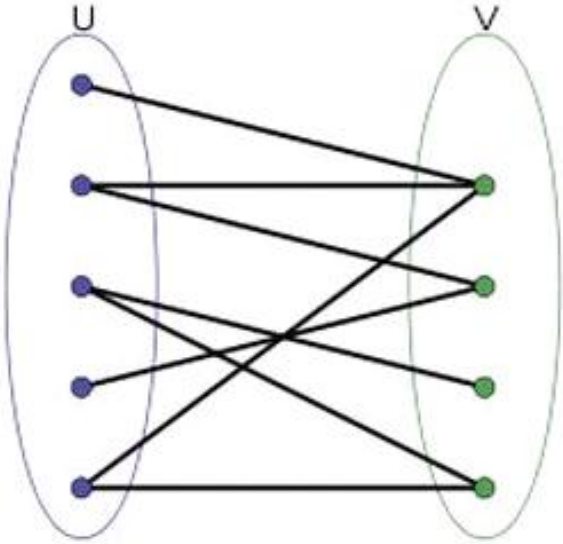


Figure 4.1: An example of a bipartite graph.

A regular graph is a graph where each vertex has the same number of neighbors, i.e., every vertex has the same degree or valency. Here, we define the *degree* of a vertex of a graph as the number of edges incident to the vertex, with loops counted twice. The degree of a vertex v is denoted $deg(v)$. A regular graph with vertices of degree d is called a $d-$regular graph or regular graph of degree d. A bipartite graph G is said to be d-left regular if all vertices in the left partition of G have degree d. Similarly, we can define a bipartite graph to be d-right regular.

A matrix is usually associated with a graph to store some of the information about the graph. This matrix can be used to obtain more detailed information about the graph. If a graph has n vertices, we may associate an $n \times n$ matrix A which is called adjacency matrix. The adjacency matrix A is defined by

$$A = \begin{cases} 1 & \text{if } v_i \to v_j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Below in Fig. 4.2 shows an example of three graphs and there corresponding adjacency matrices. It also indicates that for an undirected graph, the adjacency matrix is symmetric.

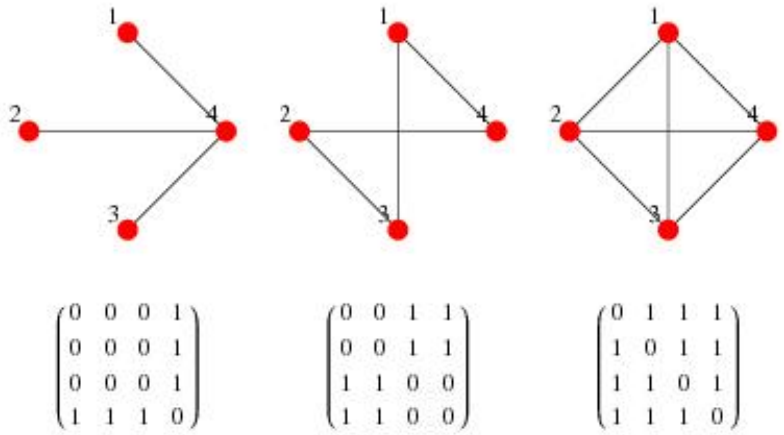


Figure 4.2: An example of the adjacency matrix of a graph.

The adjacency matrix A of a bipartite graph whose parts have r and s vertices has the form

$$A = \begin{pmatrix} O & B \\ B^T & O \end{pmatrix},$$

where B is an $r \times s$ matrix and O is an all-zero matrix. Clearly, the matrix B uniquely represents the bipartite graphs. It is sometimes called the biadjacency matrix. Formally, let $G = (U, V, E)$ be a bipartite graph with parts $U = u_1, \cdots, u_r$ and $V = v_1, \cdots, v_s$. The biadjacency matrix is the $r \times s$ 0-1 matrix B in which $b_{i,j} = 1$ iff $(u_i, v_j) \in E$. For example, the biadjacency matrix B in Fig. 4.1 is a 5×4 matrix

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

So the adjacency matrix A of this bipartite matrix is

$$A = \begin{pmatrix} O & B \\ B^T & O \end{pmatrix} = \left(\begin{array}{ccccc|cccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right)$$

4.2 Expander Graph

Definition: A $(k, \epsilon)-$*unbalanced expander* is a bipartite graph $G = (U, V, E), |U| = n, |V| = m (n \neq m$ thus unbalanced) with left degree d such that for any $X \subset U$ with $|X| \leq k$, the set of neighbors $N(X)$ of X has size $|N(X)| \geq (1-\epsilon)d|X|$, for all $0 < \epsilon < 1$.

For example, the bipartite graph $G = (U, V, E)$ in Fig. 4.3 is called a (k, ϵ) *expander* if for any subset of left nodes X, with cardinality $|X| \leq k$, are connected to at least $(1-\epsilon)|E(X)|$ righthand side nodes (namely the neighbors of X, denoted by $N(X)$), where $|E(X)|$ is the set of links that go from X to the righthand side (namely the degree d of left node U). In other words, $|E(X)|$ is the total number of nonzero elements in the columns corresponding to X in the biadjacency matrix B of the bipartite graph G, $|N(X)|$ is the

number of nonzero rows in the biadjacency B. Put it mathematically, $|N(X)| \geq (1-\epsilon)d|X|$.

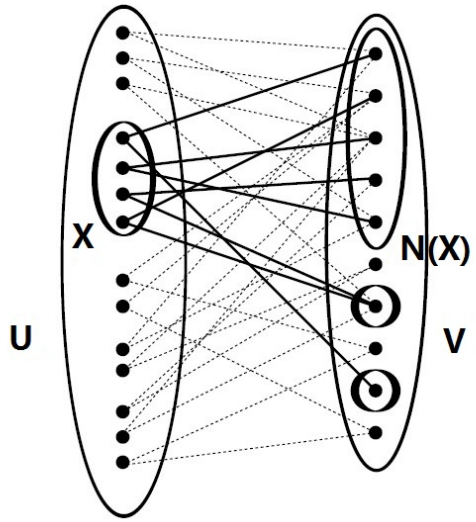


Figure 4.3: An unbalanced expander graph: any sufficiently small subset X on the left has a neightborhood $N(X)$ of size at least $(1-\epsilon)d|X|$

4.2.1 Some Facts on Expander Graphs

The following claim follows from the chernoff bounds [7].

Proposition 1: For any $n/2 \geq k \geq 1, \epsilon > 0$, there exists a $(k,\epsilon)-$unbalanced expander with left degree

$$d = \mathbb{O}(\frac{\log(\frac{n}{k})}{\epsilon})$$

and right set size

$$m = \mathbb{O}(\frac{k\log(\frac{n}{k})}{\epsilon^2})$$

The RIP-1 property of the expander graph is stated as follows:

Lemma 4.0.1 (RIP-1 property of the expander graphs [49])**.** *: Let $m \times n$ matrix Φ be the adjacency matrix of a $(k, 1-\epsilon)$ expander graph E, then for any $k-$sparse vector $x \in \mathbb{R}^n$ we have:*

$$(1-2\epsilon)d||x||_1 \leq ||\Phi x||_1 \leq d||x||_1$$

Proof: The upper bound is trivial using the triangle inequality, so we only prove the lower bound. The left side inequality is not influenced by changing the position of the coordinates of x, so we can assume that they are in a non-increasing order: $|x_1| \geq |x_2| \geq \cdots \geq |x_n|$. Let E be the set of edges of G and $e_{ij} = (x_i; y_j)$ be the edge that connects x_i to y_j. Define

$$E_2 = \{e_{ij} : \exists k < i \; s.t. \; e_{kj} \in E\},$$

.Intuitively E_2 is the set of the collision edges. Let

$$T_i = \{e_{i'j} \in E_2 \; s.t. \; i' < i\},$$

and $a_i = |T_i|$.

Clearly $a_1 = 0$; moreover by the expansion property of the graph for any k' less than or equal to k, a'_k is less than or equal to $\epsilon d'_k$. Finally since the graph is k-sparse we know that for each k'' greater than k, x''_k is zero. Therefore

$$\begin{aligned}
\sum_{e_{ij} \in E_2} &= \textstyle\sum_{i=1}^{n} |x_i|(a_i - a_{i-1}) \\
&= \textstyle\sum_{i \leq k} a_i(|x_i| - |x_{i+1}|) \\
&\leq \textstyle\sum_{i \leq k} \epsilon d i(|x_i| - |x_{i+1}|) \\
&\leq \textstyle\sum_{i \leq k} |x_i| \epsilon d \\
&= \epsilon d||x||_1.
\end{aligned}$$

Now the triangle inequality, and the definition of E_2 imply

$$
\begin{aligned}
||\Phi x||_1 &= \sum_{j=1}^{m} |\sum_{e_{ij}\in E} x_i| \\
&= \sum_{j=1}^{m} |\sum_{e_{ij}\in E_2} x_i + \sum_{e_{ij}\notin E_2} x_i| \\
&\geq \sum_{j=1}^{m} (|\sum_{e_{ij}\notin E_2} x_i| - |\sum_{e_{ij}\in E_2} x_i|) \\
&= \sum_{j=1}^{m} (|\sum_{e_{ij}\notin E_2} x_i| + ||\sum_{e_{ij}\in E_2} x_i|| - 2||\sum_{e_{ij}\in E_2} x_i||) \\
&= \sum_{e_{ij}\notin E_2} |x_i| + \sum_{e_{ij}\in E_2} |x_i| - 2\sum_{e_{ij}\in E_2} |x_i| \\
&\geq d||x||_1 - 2\epsilon d||x||_1 \\
&= (1-2\epsilon)d||x||_1.
\end{aligned}
$$

4.3 Connection between Expander Graph and Compressed Sensing Matrices

In this section we recall the main results by Berinde et al. [6].It shows that RIP-p matrices for $p \approx 1$ can be constructed using high-quality expanders.

We consider the CS matrices that are *binary* and *sparse*. They have only a small number (i.e., d) of ones in each column, and all the other entries are equal to zero. It has been shown recently [25] that such matrices cannot satisfy the RIP property with parameters k and δ, unless the number of rows is $\Omega(k^2)$. Recent result in [6] demonstrates that such matrices satisfy a different form of the RIP property, namely RIP-p property if for any k-sparse vector x and $||\Phi x||_p = (1 \pm \delta)||x||_p$ holds. In particular, it shows that this property holds for $1 \leq p \leq 1 + \mathbb{O}(1)/\log n$ if matrix Φ is an adjacency matrix of a high-quality *unbalanced expander graph*, where "high-quality" refers to an expander with ϵ as small as possible, creating an expansion as large as possible.

In the following, we discuss the relationship between the adjacency matrix, the (k, ϵ)-unbalanced expander graph and the RIP-p property of the adjacency matrix that have been extensively studied in [6]. Theorem 4.1 states the fact that the adjacency matrix of a (k, ϵ)-unbalanced expander graph satisfies the RIP-p property, which can be used for guaranteed CS recovery problem in (Sparse Recovery).

Theorem 4.1 ([6]). *Consider any $m \times n$ matrix Φ that is the adjacency matrix of an $(k, \epsilon)-$unbalanced expander $G = (U, V, E), |U| = n, |V| = m$, with left degree d, such that $1/\epsilon, d$ are smaller than n. Then the scaled matrix $\Phi/d^{\frac{1}{p}}$ satisfies the RIP$_{p,k,\delta}$ property, for $1 \leq p \leq 1 + \mathbb{O}(1)/\log n$ and $\delta = C\epsilon$ for some absolute constant $C > 1$.*

Proof: **The case of** $p = 1$ is proved in Section 4.3 in this thesis. **The case of** $p \leq 1 + 1/\log n$ is proved in the full version in [6].

In the following, Theorem 4.3 demonstrates that any binary matrix Φ with each column having d 1's and satisfies the RIP-p property with proper parameters, must be an adjacency matrix of a good unbalanced expander. That is, a RIP-p matrix and the adjacency matrix of an unbalanced expander are essentially equivalent.

Theorem 4.2 ([6]). *Consider any $m \times n$ binary matrix Φ such that each column has exactly d ones. If for some scaling factor $S > 0$ the matrix $S\Phi$ satisfies the RIP$_{p,k,\delta}$ property, then the matrix Φ is an adjacency matrix of an $(k, \epsilon)-$unbalanced expander, for*

$$\epsilon = (1 - \frac{1}{1+\delta})/(2 - \sqrt{2})$$

Next,we want to show that if Φ is an adjacency matrix of an expander graph, then the LP decoding procedure can be used for recovering sparse approximations. Let Φ be an $mtimesn$ adjacency matrix of an unbalanced $(2k, \epsilon)$-expander G with left degree d. Let $\alpha(\epsilon) = (2\epsilon)/(1 - 2\epsilon)$. We also define $E(U : V) = E \cap (X \times Y)$ to be the set of edges between the sets U and V. The following theorem provides recovery guarantees for the problem Sparse Recovery by setting $u = x$ and $v = x^\star$

Theorem 4.3 ([6]). *Consider any two vectors u, v, such that for $y = u - v$ we have $\Phi y = 0$, and $||v||_1 \leq ||u||_1$. Let S be the set of k largest (in magnitude) coefficients of u, then*

$$||v - u||_1 \leq 2/(1 - 2\alpha(\epsilon)) \cdot ||u - u_S||_1$$

Proof: We have

$$\begin{aligned} ||u||_1 \geq ||v||_1 &= ||(u+y)_S||_1 + ||(u+y)_{S^c}||_1 \\ &= ||u_S||_1 - ||y_S||_1 + ||y_{S^c}||_1 - ||u_{S^c}||_1 \\ &= ||u||_1 - 2||u_{S^c}||_1 + ||y||_1 - 2||y_S||_1 \\ &\geq ||u||_1 - 2||u_{S^c}||_1 + (1 - 2\alpha(\epsilon))||y||_1 \end{aligned}$$

Lemma 4.3.1 (L1 Uncertainty principle)**.** *: Consider any $y \in \mathbb{R}^n$ such that $\Phi y = 0$, and let S be any set of k coordinates of y. Then we have*

$$||y_S||_1 \leq \alpha(\epsilon)||y||_1$$

Proof: See [6].

where we used Lemma 4.3.1 in the last line. It follows that

$$2||u_{S^c}||_1 \geq (1 - 2\alpha(\epsilon))||y||_1$$

By combining Theorem 4.3 with some basic facts of high quality expanders described in Section 4.2.1 we conclude that the CS recovery scheme can achieve best known bounds on the number of measurements which is $\mathbb{O}(k \log(n/k))$, the update time, $\mathbb{O}(\log(n/k))$, and the encoding time, $\mathbb{O}(n \log(n/k))$.

4.4 Design of Sparse Measurement Matrices

We will now begin to give the procedure on how to design the binary sparse matrices to serve for the measurement matrix used in compressed sensing. Followed that, we observed that, if we use this $\{0, 1\}$- matrix as an indicator set for the sensor subset selection, with each row corresponding to the total number of n sensors in the field, interestingly, each sensor will be selected with equal number of times during the measurement process. In other word, we have an uniform energy consumption on every sensor and guaranteed fairness on the subset of sensor selected. We will illustrate more in Chapter 5 Section 5.2.

A binary sparse matrix Φ of m rows and n columns is generated in the following steps:

- Step 1: For each column, randomly generate d integers whose values are between 1 and m and place $1's$ in those rows indexed by the d numbers;
- Step 2: If the d numbers are not distinct, repeat Step 1 until they are (this is not really an issue when $d \ll m$).

By Theorem 4.3 with some proper value of d, we see that such a matrix is the adjacency matrix of an expander graph of degree d with high probability.

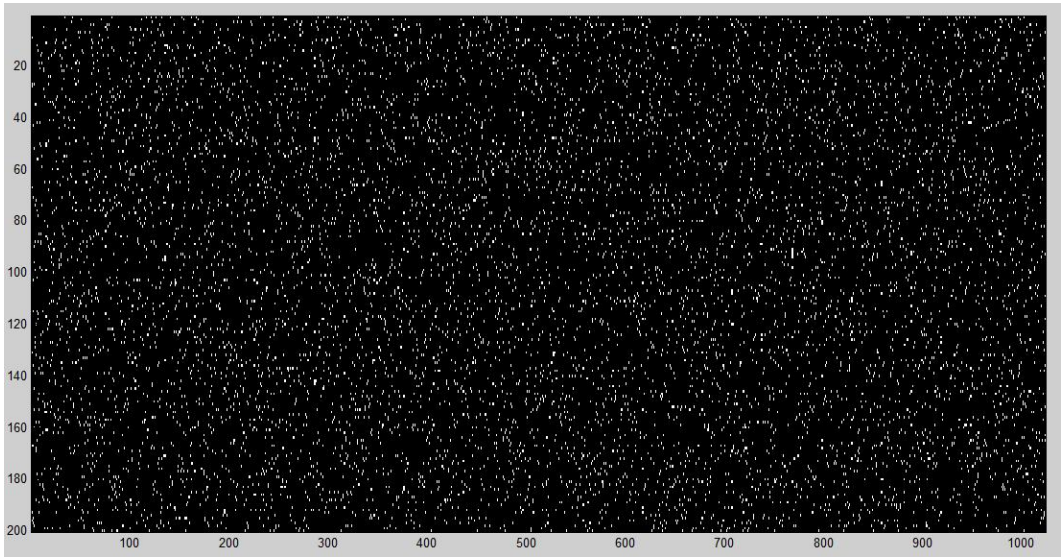


Figure 4.4: An Example of 200×1024 binary sparse matrix with degree $d = 8$

Chapter 5

Distributed Compressive Sparse Sampling Algorithm

In this chapter, the detail of our the distributed compressive sparse sampling (DCSS) algorithm will be described with the averaged per sensor communication cost analyzed. Some practical implementation issues and the fairness of the sensor selection problem are also discussed then after.

5.1 Distributed Compressed Sensing Algorithm

We now describe an algorithm by which every sensor in the wireless network each measures a data value $x_i \in \mathbb{R}^n$, and each computes and stores one sparse random projection of the aggregate data x. Consider a $m \times n$ sparse binary matrix Φ with entries as defined in Chapter 4. Each sensor will compute and store the inner product $\sum_{j=1}^{n} \Phi_{ij} x_j$ between the aggregate data x and one row of Φ. See Figure 5.1, we think of this as generating a bipartite graph between the n data nodes and the m encoding nodes. Figure 5.2 shows that the m encoding nodes can be used as the m observations to reconstruct an approximation of x. The m encoding nodes will corresponding to m designated sensors in the network. Below is the description of the distributed compressive sparse sampling (DCSS) algorithm.

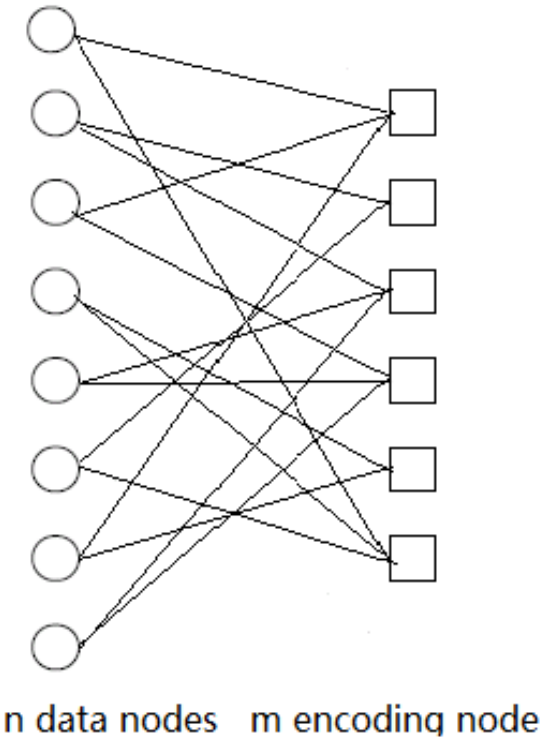


Figure 5.1: Sensor readings projection can be seen as generating a bipartite graph between the n data nodes and the m encoding nodes.

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \phi_{ij} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{m1} & \phi_{m2} & \cdots & \phi_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{bmatrix}$$

Sparse binary matrix Sensor readings Observations

Figure 5.2: An example of observations from sensor data projection using sparse binary matrix.

Algorithm 1 Distributed Compressive Sparse Sampling Algorithm (DCSS)

Input:

Sparse binary matrix $\Phi \in \mathbb{R}^{m \times n}$;

Environmental reading $x \in \mathbb{R}^n$;

Output:

Recovered vector $x^* \in \mathbb{R}^n$;

1: The fusion center (FC) generates a sparse binary matrix Φ and randomly chooses m designated sensors then broadcasts Φ and m sensor IDs, each sensor node in the network stores them locally;

2: For each individual sensor j, $(1 \leq j \leq n)$, if $\phi_{ij} \neq 0$, then sensor node j sends its reading x_j to designated sensor ID i. Repeat for all $1 \leq j \leq n$;

3: For m designated sensors, each computes and stores the sum of the reading it receives;

4: Finally, the m designated sensors send directly their results to FC and FC performs compressed sensing recovery for $x^\star$.

From above DCSS algorithm, we see that for each individual sensor, $\phi_{ij} \cdot x_j$ means that sensor id j sends its readings x_j to the designated sensor i, if $\phi_{ij} \neq 0$. Correspondingly, for m designated sensors, designated sensor i computes and stores the summation of the sensor reading it receives, repeat for all $1 \leq i \leq m$ and then report $y_1, y_2, \cdots, y_m$, as observations $y \in \mathbb{R}^m$ to FC. After collecting all the m observations, FC will recover the original signal $x \in \mathbb{R}^n$ using Linear Programming. For the communication cost, the dense CS matrices require $\mathbb{O}(n)$ packets per sensor to compute the projection in a network of size n, while the sparse binary matrix with fix d ones in each column requires $\mathbb{O}(1)$ packets per sensor to compute the projection in a network of size n and the m designated sensors need an average of $\frac{1}{m} \cdot \mathbb{O}(m \cdot d) = \mathbb{O}(1)$ communications to compute the observations. We see that using sparse binary matrix greatly reduces the overall communication cost.

We want to talk a little bit more on implementation of the algorithms. The main issues are how to choose d, that is the number of ones in each column, and m, which is the number of designated sensor that directly communicate to fusion center. We already give some theoretic facts in Chapter 4 Section 4.2.1, where $d = \mathbb{O}(\log(n/k)$ and $m = \mathbb{O}(k \log(n/k)$. In our experiments, the dimension of data is 1024 and d is set equal to 8. Vary d a little does not affect too much the recovery results. While for the purpose of testing the performance

of recovery, we vary the m from 100 to 1000 depending different experiments settings. But clearly $m = \mathbb{O}(k \log(n/k)$ should be a touchstone for choosing the number of observations in order to achieve both energy efficiency and communication reduction.

5.2 Concern on the Fairness of the Subset Selection Process

Since sparse binary projection matrix we proposed can also be seen as a sensor subset selection matrix for choosing m designated sensors to communicate with the FC directly, it is important to evaluate its fairness in sensor selection to avoid leaving "holes" in the network due to extensive energy consumption for computation and communication. In this case, a balanced energy consumption among all the sensor nodes is obvious since the proposed sparse binary projection matrix has exactly d 1's along each column confirming that every sensor nodes in the WSN will be selected to be active d times during the entire process of distributed data aggregation for recovery of n-dimensional data value x. The only imbalance energy consumption comes from these designated sensors which computes and reports the projections to the FC. However, since $m \ll n$ and the measurement matrix can be re-generated each time a query task is performed, in the long run, the energy consumption can still be balanced.

Chapter 6

Experiments

Following chapter 5′s procedures, we carried out the experiments on both synthetic and real datasets using distributed compressed sensing algorithms for data recovery. For the synthetic data experiment part we test the recovery of exact k-sparse signals, noisy sparse signals with different noise decibels and as well as compressible signals. In the real experiment part, the real data is collected from a WSN deployed in the Intel Berkeley Research Laboratory. The network recorded temperature, humidity, light and voltage measurements at 31 seconds intervals. In our experiments, we use the light intensity readings at node 19. The data values are represented by wavelet coefficients to get a compressible representation. Throughout the experiments, we assume the data value x we want to recover is in 1024 dimension. For the CS recovery algorithm that processed in the FC, we use linear programming method in the popular ℓ_1 magic [16] package. The random Gaussian and Fourier scramble measurement matrices are often used in the CS community. We will compared our results using sparse binary matrix with both of them.

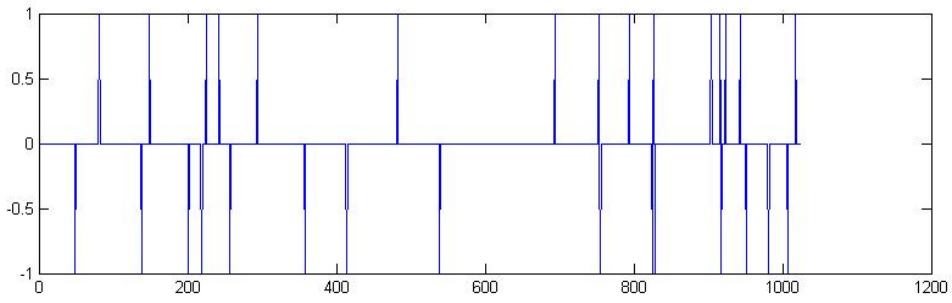


Figure 6.1: An example of an exact sparse signal with $n = 1024$ and sparsity $k = 30$.

6.1 Experiment using Synthetic Data

6.1.1 Exact Sparse Signal Recovery

We consider a sparse vector x with length $n = 1024$ contains only ± 1, and set the sparsity level $k = 30$ (i.e., x only have 30 non-zero entries). Figure 6.1 shows the exact sparse signal x.

Figure 6.2 shows the recovery performance compared to the Gaussian Fourier scramble measurement matrix. The recovery performance is evaluated with various number of measurements m, and we use the relative approximation error $\frac{||x-x^*||_2}{||x||_2}$ as the metric, where x^* is output recovered result of signal x. We run the recovery process 100 times with the same sparse signal x each round and different measurement matrices. The result demonstrates that the sparse binary matrix performs as good as these dense matrices that used more often.

6.1.2 Noisy Sparse Signals Recovery

We consider the same sparse signal x but with different noise levels that occurred during the sensor data recording and data transmission stage. In our experiments we manually add different noise levels to observation y. Figure 6.3 shows the recovery performance of the noisy measurements y. We use five different noise levels with the signal to noise ratio (SNR) equal to 5, 15, 25, 35, 45, respectively. The number of measurements is chosen

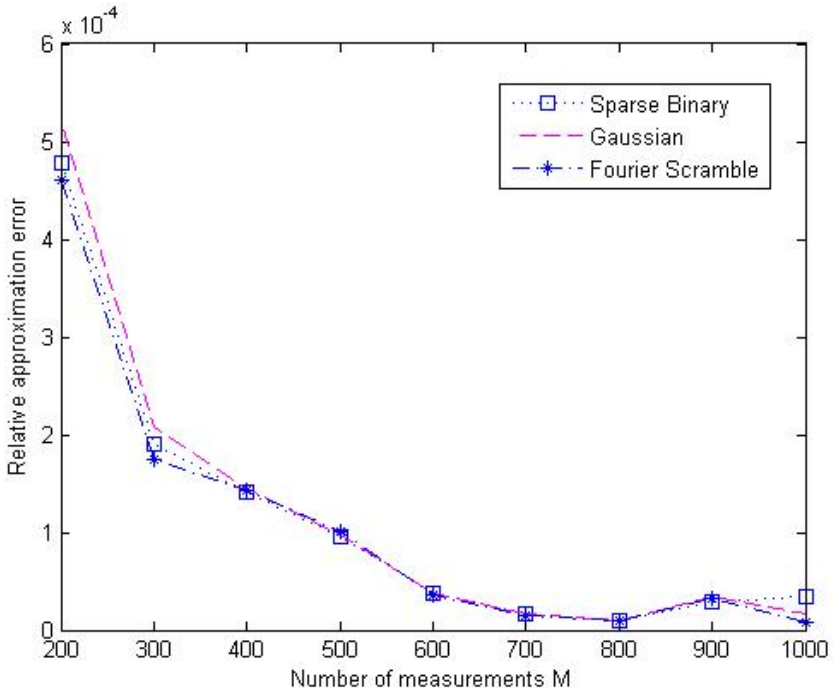


Figure 6.2: Recovery result of an n = 1024, sparsity k=30 sparse signal x, with an average of 100 experiments using LP recovery method.

as $m = 200, 700$ respectively. We use the ℓ_2 distance between the original signal and recovered one (i.e., $||x - x^\star||_2$) as the metric. The results are obtained by an average of 100 experiments using the same noisy measurements y. We see that the proposed sparse binary matrix performs better than Gaussian matrix, while Fourier scramble works the best.

In order to be more fair on the evaluation, we increase the total number of measurements to account for the error induced by insufficient measurements. Figure 6.4 shows the result using $m = 700$ measurements. We see that since the observations y are corrupted by noise, the more observations we get the worse the recovered result. But the result from using the sparse binary matrix still lies in between the Gaussian matrix and the Fourier Scrambles.

We also conduct experiment on the recovery performance of our sparse binary matrix with different noise SNRs added on different number of measurements. With an average of 100 experiments, Figure 6.5 demonstrates that the higher the noise level, the worse the

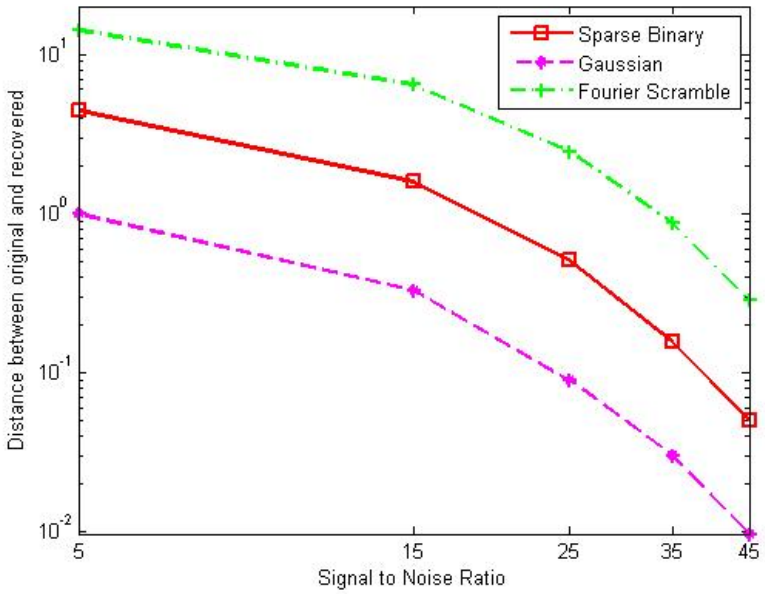


Figure 6.3: Noisy recovery result of an n = 1024, sparsity k=30 sparse signal x, with different SNRs and an average of 100 experiments using LP recovery method and $m = 200$ measurements.

recovery accuracy. It is true because CS normally requires the sparsity feature of the signals while adding more noise causes the signal to be less sparse.

6.1.3 Compressible Signal Recovery

While a strict sparse signal is rare in reality, we conduct experiments on compressible signal which obey a decay signal model, $x = 4n^{-\frac{7}{10}}$, we then sample the signal x to get $n = 1024$ data points. We compare the recovery result using different matrices and various measurements. The best k-term approximation is obtained by only keeping the first k coefficients while setting the rest to zero. It can serve as a baseline of the recovery performance evaluation. Figure 6.6 demonstrates that sparse binary matrices performs the best for compressible signals in the sense that it has the closest recovery result to the known best k-term approximation. Table 6.7 shows detailed result of the recovered error.

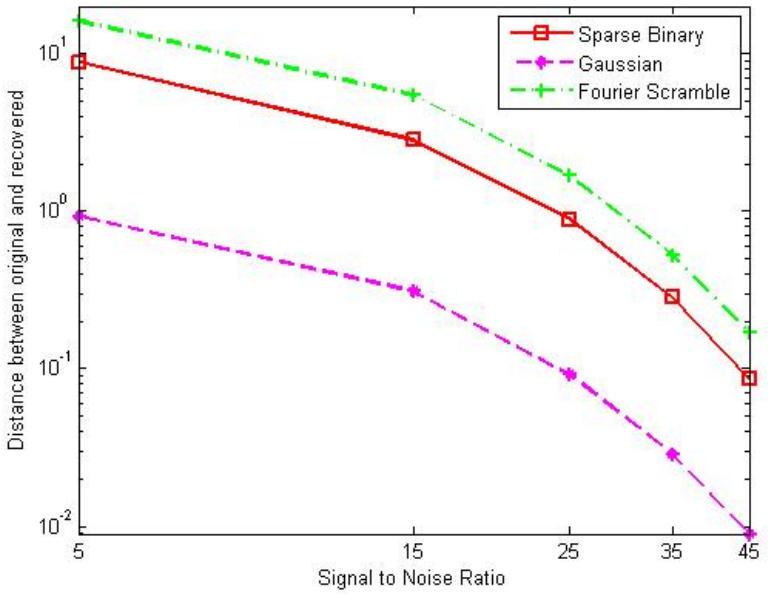


Figure 6.4: Noisy recovery result of an n = 1024, sparsity k=30 sparse signal x, with different SNRs and an average of 100 experiments using LP recovery method and $m = 700$ measurements.

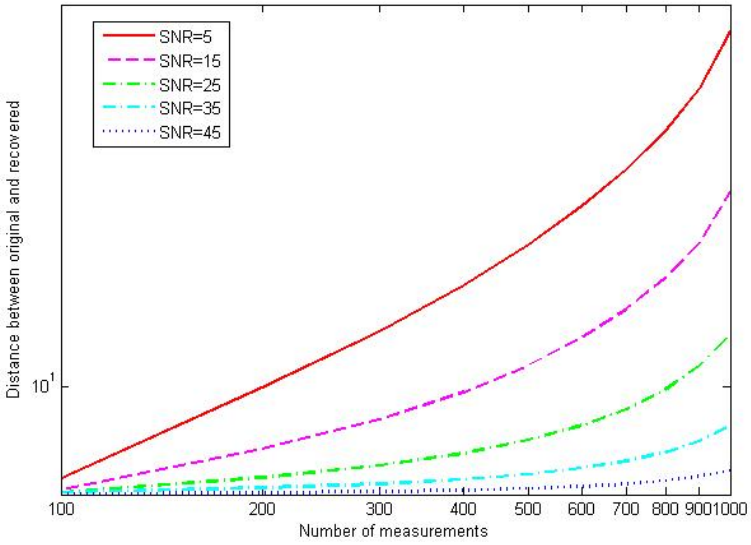


Figure 6.5: Noisy recovery result of an n = 1024, sparsity k=30 sparse signal x, with different SNRs and an average of 100 experiments using LP recovery method evaluated by different measurements.

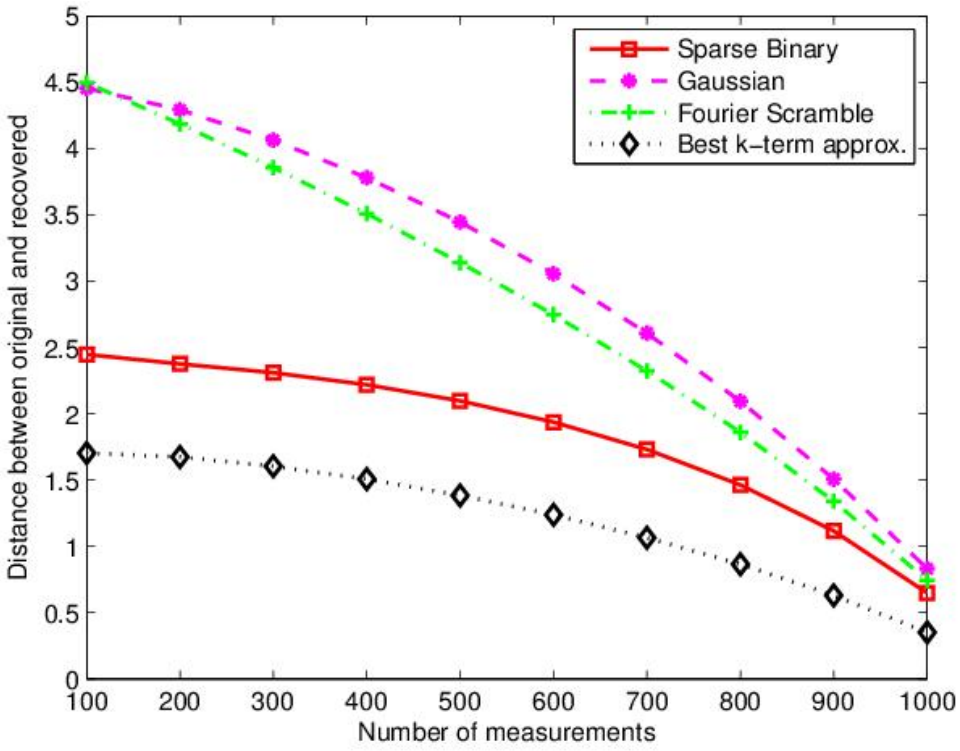


Figure 6.6: Recovery result of an sampled compressible signal $x = 4n^{-\frac{7}{10}}$, with an average of 100 experiments using LP recovery method evaluated by different measurements.

Methods	Number of Measurements									
	100	200	300	400	500	600	700	800	900	1000
Sparse Binary	2.45	2.38	2.31	2.22	2.10	1.94	1.73	1.46	1.11	0.65
Gaussian	4.46	4.30	4.07	3.78	3.45	3.06	2.61	2.10	1.51	0.83
Fourier Scramble	4.50	4.19	3.86	3.51	3.14	2.75	2.32	1.86	1.34	0.74
Best k-term	1.71	1.68	1.61	1.51	1.39	1.24	1.07	0.87	0.64	0.35

Figure 6.7: Recovery results comparison of different methods.

Table 6.1: Signal to Noise Ratio (SNR) of different measurement matrices.

Methods	Wavelet Approx.	Sparse Binary	Gaussian	Fourier Scramble
SNR	21.4735	25.5702	22.8528	25.4075

6.2 Experiment using Real Data

In this experiment, we consider the dataset [54] for temperature, humidity, and light readings from a group of nodes deployed at the offices of Intel Research Labs in Berkeley, CA. The signals were recorded in an office environment and therefore exhibit periodic behavior caused by the activity levels during day and night. Therefore, we expect the signals to be compressible in the wavelet domain.

We consider the recovery from CS measurements for the light intensity signal at node 19. We obtain different number of CS measurements for the signal using sparse binary matrix, Gaussian matrix, Fourier Scramble. Figure 6.8 plots the recovered signal using 100 coefficients for wavelet approximation and the same 400 coefficients for Gaussian, sparse binary matrix and Fourier Scrambles. We observe that wavelet approximate gives us more smooth signal profile than the other measurement matrices, however, the recovered SNR is the worst. Gaussian, Fourier Scramble and sparse binary matrices all perform well on recovering the details around data point position 200 and 500 and sparse binary matrix has the recovered SNR equals to 25.5702, which is the highest among the three. Table 6.1 shows the detailed SNRs of the recovered signal.

We also evaluate the recovery performance using the relative approximation error metric. Figure 6.9 shows the result compared to various recovery methods, since our sparse binary matrix also serve as the sensor selection matrix, we conduct experiment to compare the sensor selection result compare to random sensor selection. The experiment shows that sparse binary matrix outperforms all other methods when the number of measurements is less than 300 except for wavelet approximation. When the number of measurements increase, the sparse binary matrix still performs better than Gaussian and slightly worse than Fourier Scramble. Random selection of sensors works good around measurement

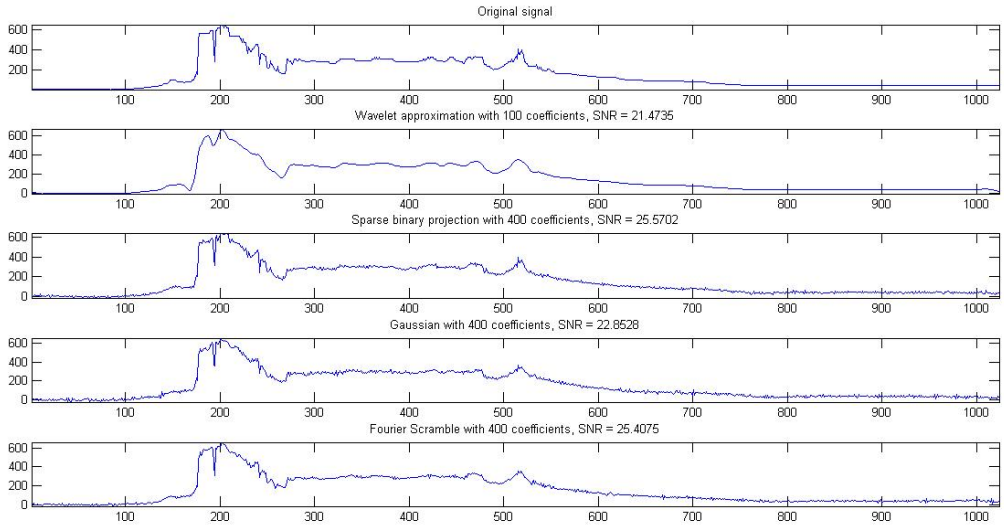


Figure 6.8: Recovery result of real Intel Lab signal using 100 wavelet coefficients and 400 CS measurements with different measurement matrices.

equals to 700 and performs poorly when the number of measurements is either too small or too large.

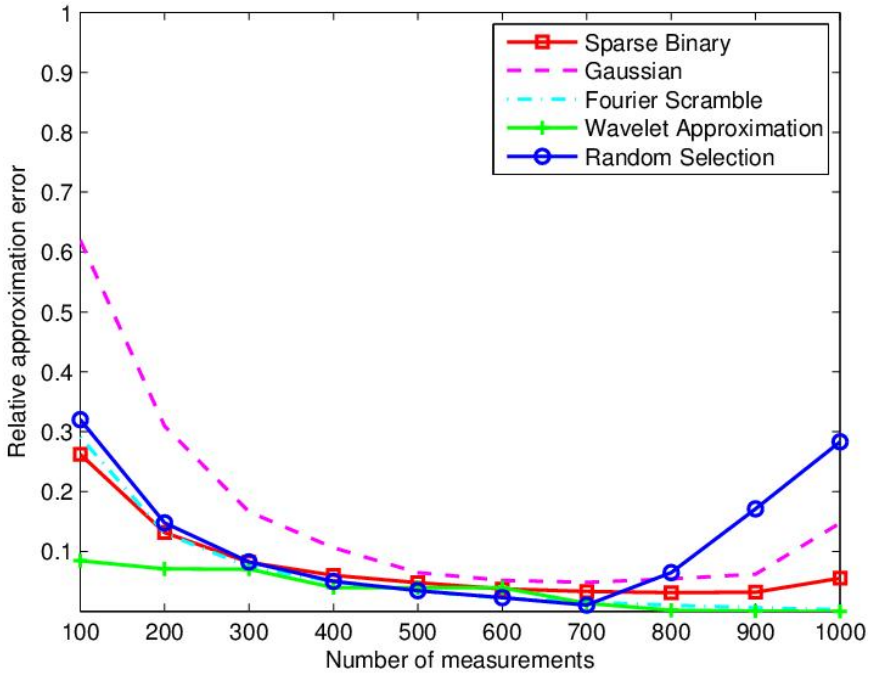


Figure 6.9: Recovery result of real Intel Lab signal, with an average of 100 experiments using LP recovery method evaluated by different measurements.

Methods	Number of Measurements									
	100	200	300	400	500	600	700	800	900	1000
Sparse Binary	0.262	0.132	0.082	0.060	0.048	0.038	0.033	0.031	0.032	0.055
Gaussian	0.620	0.309	0.166	0.106	0.065	0.052	0.049	0.054	0.062	0.148
Fourier Scramble	0.291	0.131	0.075	0.050	0.034	0.023	0.016	0.010	0.006	0.003
Wavelet Approx.	0.084	0.071	0.071	0.040	0.040	0.040	0.014	0.002	0.0004	5*10^-13
Rand. Sampling	0.320	0.148	0.082	0.050	0.034	0.023	0.011	0.065	0.171	0.283

Figure 6.10: Recovery results comparison of different methods.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this thesis, we presented a novel distributed compressed sensing algorithm based on sparse binary matrix for data recovery in WSNs. The connection between the sparse graph and CS measurement matrix was explored. We constructed a sparse binary matrix that resemble the unbalance expander graph with high probability for CS measurement matrix. We showed that the sparse binary matrix can be used to select active sensor nodes and project the sensor node readings to the designated sensors in the network simultaneously. The experimental results showed that using sparse binary matrix, the accuracy of recovery can be as good as the traditional dense CS measurement matrices and the proposed sparse binary measurement matrix worked well on compressible data. Results from both synthetic and real data experiments demonstrated the usefulness of the proposed distributed CS algorithms using sparse binary matrix while the average communication cost per sensor reduced from $\mathbb{O}(n)$ to $\mathbb{O}(1)$.

7.2 Future Work

In the future, several aspects can be improved. Optimal design of the sparse binary measurement matrix using expander graph or other sparse graph codes. For the recover methods, we use LP decoding only, other fast online might be more realistic for wireless sensor network application.

Bibliography

Bibliography

[1] Ahn, S. and Kim, D. (2006). Proactive context-aware sensor networks. *Wireless Sensor Networks*, pages 38–53.

[2] Bajwa, W., Haupt, J., Sayeed, A., and Nowak, R. (2006). Compressive wireless sensing. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 134–142. ACM.

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

[4] Baraniuk, R. G. (2007). Compressive sensing. *IEEE Signal Processing Magazine*, pages 118–124.

[5] Baron, D., Wakin, M., Duarte, M., Sarvotham, S., and Baraniuk, R. (2005). Distributed compressed sensing.

[6] Berinde, R., Gilbert, A., Indyk, P., Karloff, H., and Strauss, M. (2008). Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 798–805. IEEE.

[7] Berinde, R. and Indyk, P. (2008). Sparse recovery using sparse random matrices. *Technical Report, MIT*.

[8] Bian, F., Kempe, D., and Govindan, R. (2006). Utility based sensor selection. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 11–18. ACM.

[9] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *Information Theory, IEEE Transactions on*, 52(6):2508–2530.

[10] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge Univ Pr.

[11] Bruckstein, A. and Elad, M. (2002). A generalized uncertainty principle and sparse representation in pairs of R^N bases. *IEEE Trans. Inf. Theory*, 48(9):2558–2567.

[12] Cai, J., Osher, S., and Shen, Z. (2008). Fast Linearized Bregman Iteration for Compressed Sensing. *and Sparse Denoising, 2008. UCLA CAM Reprots*, pages 08–37.

[13] Cai, T., Wang, L., and Xu, G. (2009). Shifting inequality and recovery of sparse signals. *IEEE Transactions on Signal Processing*.

[14] Cai, T., Wang, L., and Xu, G. (2010). New bounds for restricted isometry constants. *Information Theory, IEEE Transactions on*, 56(9):4388–4394.

[15] Candes, E. and Randall, P. (2008). Highly Robust Error Correction byConvex Programming. *IEEE Transactions on Information Theory*, 54(7):2829–2840.

[16] Candes, E. and Romberg, J. (2005a). l 1-magic: Recovery of sparse signals via convex programming.

[17] Candes, E. and Romberg, J. (2005b). l1-magic: Recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*.

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

[19] Candes, E., Romberg, J., and Tao, T. (2006a). Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223.

[20] Candes, E., Rudelson, M., Tao, T., and Vershynin, R. (2005). Error correction via linear programming. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 668–681. IEEE.

[21] Candes, E. and Tao, T. (2007). The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351.

[22] Candes, E. J., Romberg, J., and Tao, T. (2006b). Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, pages 1207–1223.

[23] Candes, E. J. and Tao, T. (2005). Decoding by linear programming. *IEEE Trans. Inf. Theory*, pages 4203–4215.

[24] Candes, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inf. Theory*, pages 5406–5425.

[25] Chandar, V. (2008). A negative result concerning explicit matrices with the restricted isometry property. *preprint*.

[26] Chen, S., Donoho, D., and Saunders, M. (2001). Atomic decomposition by basis pursuit. *SIAM review*, pages 129–159.

[27] Considine, J., Li, F., Kollios, G., and Byers, J. (2004). Approximate aggregation techniques for sensor databases. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 449–460. IEEE.

[28] Cormode, G., Garofalakis, M., Muthukrishnan, S., and Rastogi, R. (2005). Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 25–36. ACM.

[29] Cormode, G. and Muthukrishnan, S. (2006). Combinatorial algorithms for compressed sensing. *Structural Information and Communication Complexity*, pages 280–294.

[30] Dai, W., Sheikh, M., Milenkovic, O., and Baraniuk, R. (2009). Compressive sensing dna microarrays. *EURASIP journal on bioinformatics and systems biology*, 2009:1–12.

[31] Dimakis, A., Smarandache, R., and Vontobel, P. (2010). Ldpc codes for compressed sensing. *Arxiv preprint arXiv:1012.0602*.

[32] Donoho, D. (2006a). Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306.

[33] Donoho, D. (2006b). For most large underdetermined systems of equations, the minimal l_1-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics*, 59(7):907–934.

[34] Donoho, D. (2006c). For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829.

[35] Donoho, D. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2197.

[36] Donoho, D. and Tanner, J. (2005). Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9446.

[37] Drori, I. and Donoho, D. (2006). Solution of l1 minimization problems by lars/homotopy methods. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Citeseer.

[38] Duarte, M., Wakin, M., Baron, D., and Baraniuk, R. (2006). Universal distributed sensing via random projections. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 177–185. ACM.

[39] Eager, D., Feng, J., and Makaroff, D. (2011). Asynchronous data aggregation for real-time monitoring in sensor networks. *IFIP Lecture Notes in Computer Science (LNCS)*, 4479(4479):73–84.

[40] E.J., C. and Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425.

[41] Figueiredo, M., Nowak, R., and Wright, S. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal on selected topics in Signal Processing*, 1(4):586–597.

[42] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

[43] Gao, J., Guibas, L., Milosavljevic, N., and Hershberger, J. (2007). Sparse data aggregation in sensor networks. In *Proceedings of the 6th international Conference on information Processing in Sensor Networks*, pages 430–439. ACM.

[44] Ghassemi, F. and Krishnamurthy, V. (2011). Separable approximation for solving the sensor subset selection problem. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1):557–568.

[45] Gilbert, A. and Indyk, P. (2010). Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947.

[46] Gribonval, R. and Nielsen, M. (2003). Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325.

[47] Hale, E., Yin, W., and Zhang, Y. (2007). A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *TR07-07, Dept. of Comput. and Appl. Math., Rice University*.

[48] Indyk, P. (2008). Explicit constructions for compressed sensing of sparse signals. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 30–33. Society for Industrial and Applied Mathematics.

[49] Jafarpour, S., Xu, W., Hassibi, B., and Calderbank, R. (2009). Efficient and robust compressed sensing using optimized expander graphs. *Information Theory, IEEE Transactions on*, 55(9):4299–4308.

[50] Joshi, S. and Boyd, S. (2009). Sensor selection via convex optimization. *Signal Processing, IEEE Transactions on*, 57(2):451–462.

[51] Kalpakis, K. (2010). Everywhere sparse approximately optimal minimum energy data gathering and aggregation in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 7(1):9.

[52] Krishnamachari, L., Estrin, D., and Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 575–578. IEEE.

[53] Li, P., Hastie, T., and Church, K. (2006). Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 296. ACM.

[54] Madden, S. (2004). Intel lab data. web page. http://db.lcs.mit.edu/labdata/labdata.html. [Online; Accessed on 10/17/2011].

[55] Madden, S., Franklin, M., Hellerstein, J., and Hong, W. (2002). Tag: a tiny aggregation service for ad-hoc sensor networks. *ACM SIGOPS Operating Systems Review*, 36(SI):131–146.

[56] Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

[57] Parvaresh, F., Vikalo, H., Misra, S., and Hassibi, B. (2008). Recovering sparse signals using sparse measurement matrices in compressed dna microarrays. *Selected Topics in Signal Processing, IEEE Journal of*, 2(3):275–285.

[58] Perillo, M., Cheng, Z., and Heinzelman, W. (2004). On the problem of unbalanced load distribution in wireless sensor networks. In *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pages 74–79. IEEE.

[59] Rana, R., Hu, W., and Chou, C. (2010). Energy-aware sparse approximation technique (east) for rechargeable wireless sensor networks. *Wireless Sensor Networks*, pages 306–321.

[60] Sarvotham, S., Baron, D., and Baraniuk, R. (2006). Compressed sensing reconstruction via belief propagation. *preprint*.

[61] S.G., M. (1999). *A wavelet tour of signal processing*. Academic Pr.

[62] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[63] Tropp, J. (2004). Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242.

[64] Tropp, J. and Gilbert, A. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655.

[65] Trzasko, J. and Manduca, A. (2009). Highly Undersampled Magnetic Resonance Image Reconstruction via Homotopic $ell_\{0\}$-Minimization. *IEEE Transactions on Medical Imaging*, 28(1):106–121.

[66] van den Berg, E. and Friedlander, M. (2007). In pursuit of a root. *Dept. Comput. Sci., Univ. British Columbia, Canada, Tech. Rep. TR-2007-19, Jun*.

[67] Wang, W., Garofalakis, M., and Ramchandran, K. (2007). Distributed sparse random projections for refinable approximation. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 331–339. ACM.

[68] Wright, S., Nowak, R., and Figueiredo, M. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493.

[69] Wu, K. and Guo, X. (2011). Compressive sensing with sparse measurement matrices. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE.

[70] Wu, T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Annals*, 2(1):224–244.

[71] Xu, W. and Hassibi, B. (2007). Efficient compressive sensing with deterministic guarantees using expander graphs. In *IEEE Information Theory Workshop, 2007. ITW'07*, pages 414–419.

[72] Ye, F., Luo, H., Cheng, J., Lu, S., and Zhang, L. (2002). A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 148–159. ACM.

[73] Yin, W., Osher, S., Goldfarb, D., and Darbon, J. (2008). Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM J. Imaging Sci*, 1(1):143–168.

Appendix

Vita

Shuangjiang Li was born in ShuCheng, Anhui, P.R. China in 1988. After graduating in 2005 from ShuCheng High School, he attended University of Science and Technology Beijing, where he received a Bachelor of Science degree in 2009 from the school of Information Engineering. In the fall of 2009, Shuangjiang enrolled into the doctoral program at the University of Tennessee at knoxville in the department of Electrical Engineering and Computer Science. At the same time, he joined the Advanced Imaging and Collaborative Information Processing (AICIP) group as a graduate teaching assistant and research assistant where he will complete his Master degree in the Winter of 2011. His major research areas are compressed sensing, wireless sensor networks, pattern recognition, and image processing.