Compressed Sensing for Networked Data

Sokhar Samb (sokhar.samb@aims-senegal.org)
African Institute for Mathematical Sciences (AIMS)
Senegal
Supervised by: Dr. Bubacarr Bah (bubacarr@aims.ac.za)
AIMS-South Africa, University of Stellenbosch.
Dr. Franck Kalala Mutombo (franckm@aims-senegal.org)
AIMS-Senegal, University of Lubumbashi, DRC.

February 3, 2019

Submitted in Partial Fulfillment of a Masters II at AIMS

Abstract

The problem of storing, transferring and retrieving a large amount of data remains a great challenge. Often, the space needed is not available in our memory disc, we don't have enough developed tools for distributed data processing and even some times, we don't have the possibility to access all of the information. Compress and sampling are two possible solutions that aims to reduce the dimension of the full information. But, whichever of the two techniques one chooses, it is necessary to recovery full or original information from the compressed data set. However, how to recovery data without any lose of information? Compressed sensing is a technique of recovering the original signal/image from the compressed signal/image and it has been apply in many domains especially in electrical engineering, computer science and physics. In physics for instance, it involves recovering signal (video, audio, image ...) from a small sample.

In this essay, we are interested in how to apply this technique for recovering missing data from a small sample in a network. Base on l_1 optimization problem and considering algorithms such as the Greedy iterative algorithms: Compressive Sampling Matching Pursuit (CoSaMP) and Iterative Hard Thresholding, we show the process of compressed sensing for a network data and give an example of a piratical example.

Keywords: Compressed sensing, networked data, complex optimization, linear equation, nullspace propriety, restricted isometry property, sparsity, big data.

Declaration

I, the undersigned, hereby declare that the work contained in this essay is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Sokhar Samb, February 3, 2019

Contents

1. Introduction

Data are today produce by many systems such as social media, computers, telephones, etc. in a very fast way and are very useful in many fields. But considering the frequency of production, the quantity of those data can be very high. Therefore, manipulating them remain a great problem. Many tools and techniques have been developed for distributed processing but they still don't have enough power to deal with high quantity of data. Thus, it become necessary to look for a way of reducing the size of the data, for instance, for storage in our computer which has a very limited storage space. However, the data obtained after applying those techniques of reduction of size of data, do not represent well the information contain in the full data set. So, it is clear that for a serious study about the phenomenon related to the data, we need to recover the full information.

In the other hand, during this last years, it appeared a new technique of recovering a sparse hight dimensional vector after a dimensional reduction called compressed sensing. In fact, compressed sensing was initiated in 2006 by two main papers: in 2006 David L. Donoho in [Don06], publish his first paper in compressed sensing in which he proposed a method of reconstruction of an unknown vector x $\in \mathbb{R}^n$(digital image or signal) from a measure general linear functional y of x. Thus, x is reconstruct by solving the l_0 minimization define by:

$$\min\|x\|_0 \quad subject \quad to \quad y = Ax \tag{1.0.1}$$

In [Don06], it has been shown that certain natural class of image need only $m = O(n^{1/4}\log^{5/2}(n))$ non-adaptive pixels samples for faithful recovery, as opposed to the usual pixel samples.
In the same direction, Candes and al in [CRT06], 2016 considered the model problem of reconstructing an object from incomplete frequency samples. He considered a discrete-time signal x and a randomly chosen set of frequencies Ω and propose a method of reconstructing x using the partial knowledge of its Fourier coefficients on the set Ω.

After this two papers, several researchers in the fields of mathematics, computer science and engineering have been interested in the field compressed data for a variety of applications in astronomy, biology, medicine, radar, and seismology, community detection in a network, finites elements to name a few.

Today the challenge that was launch by Jarvis Haupt and al in [HBRN08] is compressed sensing for network data. It consist of applying the principle of compressed sensing in network data.

1.1 Problem Statement

Imagine that we have a large network with millions of nodes n. Each of the nodes contain a piece of information or data (signal for example) x_j, $j = 1, ..., n$. Suppose that we want to share, retrieve or store this data. The vector $x = [x_1, x_2, ..., x_n]^T$ organize into a network, constitute a networked data and just gathering x in a single point require at least n communications which is very daunting. Thus, let us consider a small sample y of length m with $m \ll n$. It is clear that it is much more easy to store y. But how do we reconstruct x from y for any need of all the original information(x)?

To give a response of the above question, the method of compressed sensing has been used by Haut in [HBRN08]. That mean that the vector x can be reconstruct from y which is in the form $y = Ax$. This method is base on two main conditions which are the sparsity of the data x and the construction of the measurement matrix A . In [HBRN08], Haut et al [HBRN08] define three methods of sparsifying networked data such as the spatial compression, the graph wavelets and the diffusion wavelets. However, they devise the compression of a networked data in two steps. The first step consist of computing the term $\sum_{j=1}^{n} A_{i,j}x_j$ for each node and the second step, using the randomized gossip algorithm, aggregate and distributed the term $A_{i,j}x_j$ across the Network. In this essay/project, we are interested on this new challenge. Base on [HBRN08], we show how to recovery Networked data from a small sample.

1.2 Outline

Our essay project is structured as follows: chapter 2, gives a review on compressed sensing for image/signal recovering follow by some illustrations. Chapter 3, reviews concepts on complex networks, definition of a network is broadly presented and its mathematical representation is given. chapter 4, presents with some details the different methods for sparsifying a network and how to compress a networked data.

2. Review on compressed sensing

Recovery an image, signal, video, data etc. from a small quantity of information (measurement) is a problem faced in many practical problems. Thus, during this last five years, compressed sensing(CS) has been one of the hot topics in the signal recovering. It has been shown that the minimum $1-norm$ solution to an under-determined system of linear equations is also the sparsest possible solution for a image recovery under quite general conditions. For instance, suppose there exists an unknown signal $x \in \mathbb{R}^n$, a measurement vector $y \in \mathbb{R}^m$ $(m \ll n)$ which is the compressed signal, and a measurement matrix $A \in \mathbb{R}^{m \times n}$ such that A is full rank and $y = Ax$. Recovering x given A and y constitutes a linear inversion problem which has a infinity solution.

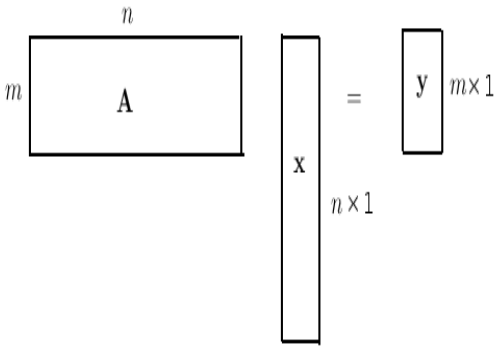


Figure 2.1: Representation of $y = Ax$

However, looking for the sparsest solution, the inverse problem is rewritten as:

$$(p_0) \quad \min_x \|x\|_0, \quad y = Ax \tag{2.0.1}$$

It has been shown that a conventional solution to this problem is the linear least squares, which finds the minimum 2-norm solution to this system. However, if x is sufficiently sparse and the sensing matrix A is incoherent with the basis under which x is sparse, then x can be exactly recovered by computing the l_1 minimization problem given by [Don06] and [CRT06]:

$$(p_1) \quad \min_x \|x\|_1, \quad y = Ax \tag{2.0.2}$$

The compressed sensing theory uses several mathematics knowledges such as norms, sparsity, nullSpace property, incoherence and the restricted isometry property. In this chapter, we give the definition of the mathematical notions used in compressed sensing. We then treat the question of how to accurately

recover a high-dimensional signal from a small set of measurements and provide performance guarantees for a variety of sparse recovery algorithms.

2.1 Definitions

2.1.1 Definition. (Norm) A function $f : \mathbb{R}^n \to \mathbb{R}$ is a norm if

1. $f(x) \geq 0, f(x) = 0 \Leftrightarrow x = 0$ (positivity).
2. $f(\alpha x) = |\alpha| f(x), \forall \alpha \in R$ (homogeneity).
3. $f(x + y) \leq f(x) + f(y)$ (triangle inequality).

2.1.2 Example (the usual norm on $\mathbb{R}^n$)**.** For all x $\in \mathbb{R}^n$, we have:

1) p-norm

$$\|x\|_p = \left(\sum_{i=1}^{n} x_i^p \right)^{1/p}.$$

2) 1-norm

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|.$$

3) 2-norm

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

4) ∞-norm

$$\|x\| = \max |x_i|.$$

5) 0-norm

$$\|x\|_0 = \sharp\{i, x_i \neq 0, i = 1, \cdots n\}.$$

It can be view as:

$$\lim_{p \to 0} \|x\|_p.$$

This 0-norm is not really a norm. In fact it doesn't satisfy the homogeneity property of the definition of a norm. For example if we consider the vector x=(0, 1, 0, 3) and $\lambda > 0$ we have:

$$\|\lambda(0, 1, 0, 3)\| = \|0, \lambda, 0, 3\lambda)\|_0 = 2 \neq |\lambda| \|x\|_0 = 2|\lambda|.$$

2.1.3 Definition. (Basis) Let V be a space vector. A linearly independent spanning set for V is called a basis. Equivalently, a subset $S \subset V$ is a basis for V if any vector $v \in V$ is uniquely represented as a linear combination

$$v = r_1 v_1 + r_2 v_2 + ... + r_n v_n$$

where $v_1, v_2, ..., v_n$ are distinct vectors from S and $r_1, ..., r_k \in \mathbb{R}$.

2.1.4 Definition. (Rank) A rank of a matrix $A \in \mathbb{R}^{m \times n}$ is the maximum number of linear independent row or column vectors.

When all of the vectors in a matrix are linearly independent, the matrix is said to be full rank.

2.2 Sparsity and compressibility of a signal

In CS theory, the sparsity of the signal to be recovery is essential.

2.2.1 Definition (Sparsity)**.** $x \in \mathbb{R}^n$ is s-sparse if:

$$|\{j : |x_j| \neq 0\}| \leq s. \tag{2.2.1}$$

It mean a signal x of length n is s-sparse if it has a few non-zero values in comparison with its length. δ_s is call the sparse coefficient. The set of s-sparse vectors is denoted by $\sum_s^n$.

This sparsity condition constitute an essential properties in CS theory. However, the sparsity property is easy to be satisfied by a signal. For instance, natural signals such as sound, image or seismic data can be stored in compressed form, in terms of their projection on suitable basis.

2.2.2 Theorem ([QBI+13])**.** *When basis is chosen properly, a large number of projection coefficients are zero or small enough to be ignored.*

2.2.3 Theorem ([QBI+13])**.** *If a large number of projection coefficients are small enough to be ignored, then signal is said to be compressible.*

Suppose that the signal is not sparse. It has been show that it is possible to find a sparse transform Φ such that $\hat{x} = \Phi x$ is sparse. And then the matrix A can be written as:

$$A = \Psi\Phi$$

Where $\Psi \in \mathbb{R}^{m \times n}$.

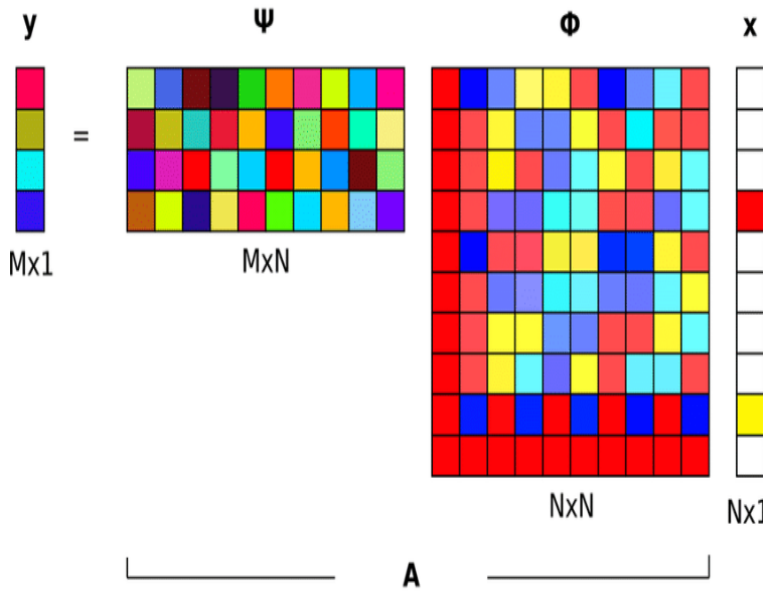


Figure 2.2: sparse transformation of a signal, source: Sourabh Bhattacharya

The sparse Fourier transform and the wavelets transform give a sparse approximation of a signal respectively in Fourier basis and Wavelet basis.

2.2.4 Discrete Fourier Transform .

2.2.5 Definition. (Discrete Fourier Transform (DFT)) DFT is a fundamental transform in digital signal processing with applications in frequency analysis, signal processing etc. DFT is the transformation of the discrete signal taking in time domain into its discrete frequency domain representation. The nth DFT coefficient of length N sequence x (n) is defined as follows [Zha16]:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi i n k/N}, \quad k = 0, ..., N-1$$

And the inverse Fourier transform as:

$$x(k) = \frac{1}{N} \sum_{k=0}^{N-1} x(n) e^{-2\pi i n k/N}, \quad n = 0, ..., N-1$$

Discrete Fourier Transform is a decomposition in a Fourier orthonormal basis $\{e^{-i2nt}\}_{n\in\mathbb{Z}}$ of $L_2([0,1])$. If $x(t)$ is uniformly regular, then its Fourier transform coefficients also have a fast decay when the frequency 2πn increases, so it can be easily approximated with few low-frequency Fourier coefficients. The Fourier transform therefore defines a sparse representation of uniformly regular functions [Ste99]. The matrix Φ formed by the Fourier transform coefficients is called the Fourier transform matrix.

2.2.6 Discrete Wavelets transform.

2.2.7 Definition. A wavelet transform of real signal x with respect to the wavelet function ψ is define as:

$$X(b,a) = \frac{1}{\sqrt{a}} \sum_{k=0}^{n} \psi'(\frac{k-b}{a})x(k). \tag{2.2.2}$$

Where ψ' is the complex conjugate of ψ.

The support of a wavelet is a square of with proportional to the scale a. Two-dimensional wavelet bases are discretized to define orthonormal bases of images including N pixels If the signal x is locally regular and a is small, then it is nearly constant over this interval and the wavelet coefficient $\Phi(a,b)$ is nearly zero [Ste99]. The matrix wavelets transform Φ is the matrix of entries the wavelets coefficients.

The error due to the sparse approximation of x is given by

$$\frac{\|x-\hat{x}\|_2^2}{n} \leq const.s^{-2\alpha} \tag{2.2.3}$$

where $\alpha > 0$. In this case, the signal x is say $\alpha-$compressible. The result guarantees that even when signals are only approximately sparse, consistent estimation is still possible [HBRN08].

2.3 Measurement matrix

One of the key elements in compressed sensing is the creation of the measurement matrix. In this section, we define the properties of sensing matrix and give some types of sensing matrix.

Measurement matrix call also sensing matrix should satisfy the properties of nullSpace property condition, incoherence condition and restricted isometry property condition. In the following line, we give the definitions of those conditions.

2.3.1 Definition (Nullspace Property [EK12])**.** Let $A \in \mathbb{R}^{m\times n}$ a matrix and $1 \leq s \leq n$. A satisfy the nullspace property of order s if for all $J \subset \{1,...,n\}$ such that $|J| = s$ and for all $v \in Ker(A) - \{0\}$ we have

$$\|v_J\|_1 < \|v_{J^c}\|_1. \tag{2.3.1}$$

With v_J is the J-restriction of v and J^c is the complement of J.

Intuitively, condition (2.3.1) says that if v is a non-zero element of Ker(A) then there can not exist s coordinated with v that contain more than half of its "mass" in the sense of 1. So, we can rewrite (2.3.1) by

$$2\|v_J\|_1 \leq \|v\|_1 \tag{2.3.2}$$

for all $J \subset 1, ..., N$ such that $|J| = s$. In other words, if a matrix satisfy the NSP then the only s-sparse vector in ker(A) is $v = 0$.

2.3.2 Definition (Incoherence)**.** The coherence between two matrices represents the highest correlation between any two columns/rows vectors of the matrices. Given two matrices A and B such that A is $n \times n$ with $A_1, A_2, \cdots A_n$ as columns and B is $m \times n$ with $B_1, B_2, \cdots B_m$ as rows then the coherence μ is define as:

$$\mu(B, A) = \sqrt{n} \max_{k,j} | < B_k, A_j > |. \tag{2.3.3}$$

For $1 \leq j \leq n$ and $1 \leq k \leq m$.

With $< B_k, A_j >$ the inner product between any two columns of A and B.

Since any columns have a l_2-norm equal to $\sqrt{n}$ the coherence take values from the interval.

$$1 \leq \mu(B, A) \leq \sqrt{n}.$$

The value of the coherence is greater if the two matrices are more correlated. In the CS scenario the value of the coherence should be as low as possible and in that case we talk about incoherence.

2.3.3 Definition (Restricted Isometry Property(RIP))**.** Let A be an $m \times n$ matrix. Then A has the Restricted Isometry Property (RIP) of order s, if there exists a $\delta_s \in (0, 1)$ such that:

$$(1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2. \tag{2.3.4}$$

For all $x \in \sum_s^n$.

This three properties verify by the measurement matrix are linked by the following theorem.

2.3.4 Theorem ([CDD09])**.** *Given $A \in \mathbb{R}^{m \times n}$, the following properties are equivalent:*

1) *Every s-sparse vector $x \in \mathbb{R}^n$ is the unique s-sparse solution of $Az = Ax$, that is, if $Ax = Az$ and both x and z are s-sparse, then $x = z$.*

2) *The null space ker(A) does not contain any 2s-sparse vector other than the zero vector, that is, $ker(A) \cap z \in \mathbb{R}^n : |z|_0 \leq 2s = 0$.*

3) *Every set of 2s columns of A is linearly independent.*

Proof. $1) \Rightarrow 2)$: let $z \in \sum_{2s} \bigcap ker(A)$. we can rewrite $z = z_1 - z_2$, $z_1, z_2 \in \sum_s \Rightarrow A(z_1 - z_2) = 0$ we have by $1)$ $z_1 = z_2 \Rightarrow z = 0$.

$2) \Rightarrow 1)$: let $x, z \in \sum_s$: $Az = Ax \Rightarrow A(z-x) = 0 \Rightarrow z - s \in ker(A)$ or $z - x \in \sum_{2s}$ we have by $2),\ z = x$.

$1) \Rightarrow 3)$: suppose there are 2s columns that are linearly dependent, this means that $\exists y$, such that $y \in \sum_{2s}$ and $Ay = 0$ Write $y = x_1 - x_2$ where $x_1, x_2 \in \sum_s \Rightarrow Ax_1 = Ax_2$,by $1)y = 0$, every set of 2s columns of A is linearly independent .

$3 \Rightarrow 1)$ suppose by contradiction that $Ax = y$ has two sparse solutions $x_1,\ x_2$. Let $z = x1 - x2$. We have $z \in \sum_{2s}$, by $2)$ $Az = 0 \Rightarrow x_1 = x_2$ which is a contradiction. □

This last theorem give some information in the number of measurement necessary to reconstruct a signal. In fact, the point (3) of the theorem implies that $rank(A) \geq 2s$. We also have rank(A)$\leq m$, because the rank is at most equal to the number of rows. Therefore, considering the point (1) of the theorem, the number of measurements needed to reconstruct every s-sparse vector always satisfies $m \geq 2s$. Some research have show that for m exactly equal to 2s, the signal x can be reconstruct from the sample y.

However, for some properties like the restricted isometry, the research have show that it is not easy to find matrices that verify this property. Thus, in the following line we define random matrix which is one of the matrix that can verify the RIP.

2.3.5 Random Matrix. Random matrix are easy to create and ensure high probability reconstruction. It exist several random matrix such as Bernoulli matrix, Gaussian matrix, partial random Fourier matrix etc.

But, Gaussian and Bernoulli sensing matrices are the most prominent random sensing matrices used in Compressive Sensing (CS). They are known by their simplicity of construction and efficiency of reconstruction. In addition, they verify the RIP with high probability. Here are the main characteristics of their elements:

- Gaussian Matrix: has elements following Gaussian distribution with zero mean and variance equal to 1.
- Bernoulli Matrix: has elements following Bernoulli distribution. They take values +1 or -1 with equal probability.

2.4 Principle and algorithms of compressed sensing

The (p_1) problem de-script above is a l_1 convex optimization problem. A convex optimisation problem is a problem consisting to minimize or maximize a convex function over a convex set.

2.4.1 Definition (Convex set)**.** A set E is convex if $\forall x, y \in E$, $\forall \lambda \in [0, 1]$:

$$\lambda x + (1 - \lambda)y \in E. \tag{2.4.1}$$

2.4.2 Definition (Convex function)**.** A function $f : E \to R$ is convex if its domain is convex and $\forall x, y \in E$ and $\forall 0 \leq \lambda \leq 1$ we have:

$$f(\lambda x + (1 - \lambda y)) \leq \lambda f(x) + (1 - \lambda) f(y). \tag{2.4.2}$$

2.4.3 Theorem. *$p - norm$ are convex for $P \geq 1$.*

2.4.4 Theorem. *$0 - norm$ is not convex.*

2.4.5 proof(Theorem 2.4.4). For proving this theorem, we look for a counter-example.
Let consider $x = (0, 1),\ y = (1, 0)$ and $\lambda = 1/2$, we have:

$$\|1/2x + (1 - 1/2y)\|_0 = 2 \neq 1 = 1/2\|x\|_0 + (1 - 1/2)\|y\|_0.$$

In the following section, we define some well know algorithms in compressed sensing.

2.4.6 Algorithms. There are several algorithms available for image recovery. In this section, we will define three well known algorithms which are l_1 minimization algorithm, the greedy algorithms: Compressive Sampling Matching Pursui(CoSaMP), Iterative Hard and soft Thresholding algorithm (IHT).

1. l_1 **minimisation:** Given measurements y and the knowledge that our original signal x is sparse or compressible, it is natural to attempt to recover x by solving an optimization problem of the form:

   $$\hat{x} = min\|x\|_0 \text{ subject to } z \in \mathcal{B}(y) \tag{2.4.3}$$

   where $\mathcal{B}(y)$, in the case where y is exact and noise-free is: $\mathcal{B}(y) = \{x : Ax = y\}$. When the measurements have been contaminated with a bounded noise, we could instead consider $\mathcal{B}(y) = \{x : \|Ax - y\|_2 \leq \epsilon\}$.

   However, it turns out that, 2.4.3 is a NP problem. In fact, as we show in the theorem (2.1.4), the l_0 is not convex norm and computationally, solving a non-convex optimisation problem is hard to

be done in a polynomial time. For s-sparse signal of length n, the total number of s position subset is $\binom{n}{s}$ which is not computationally feasible if n and s are large number. A alternative solution is to perform a l_1 minimization problem in place of a l_0-norm minimization. This l_1 minimization also call basis pursuit problem is define as follow:

$$\hat{x} = min\|x\|_1 \quad \text{subject to } z \in \mathcal{B}(y) \tag{2.4.4}$$

The choice of using the l_1-norm doesn't depend only on the fact that the l_1 is convex. In fact, recent literature have demonstrated that when the representation solution obtained by using the l_1-norm minimization constraint is also content with the condition of sparsity and the solution using l_1-norm minimization with sufficient sparsity can be equivalent to the solution obtained by l_0-norm minimization with full probability[OY14]. The figure bellow give a comparison of the l_1 and the l_2 minimization solution. Observing the figure above, we see that the l_1 ball fit the

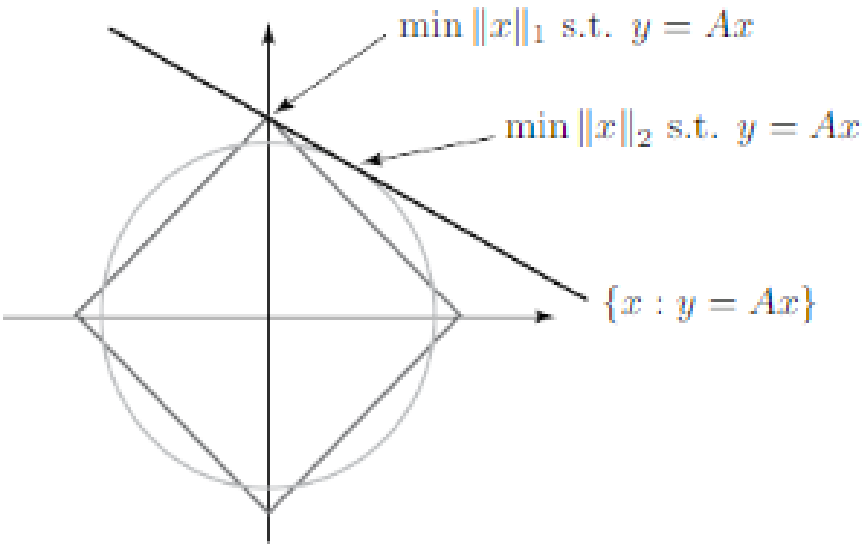


Figure 2.3: l_1 versus l_2 minimization, source [OY14]

constraint $y = Ax$ exactly on the line $x = 0$ in contrary to the l_2 ball which fit the constraint in a point which coordinate $x \neq 0$ and $y \neq 0$. It mean the solution obtain with the l_2 is non-sparse. There exist different method of solving a l_1 minimization problem such that linear programming. Because of the absolute values, the objective function is not linear. Thus, we observe two cases. First suppose that the unknown signal is non-negative component then the l_1 minimization is:

$$\sum_{i}^{n} x_i : \ y = Ax, \quad x \geq 0$$

In this case, objective function is linear and all the two constraint are linear. And then we obtain a linear minimization problem which can be solve using for instance the l_1 penalty algorithm. The other which is the more general case is when the signal contain negative values. In this case, we need to include auxiliary decision variables $y_1, y_2, ..., y_n$ to linearise the objective function. The intent is for y_j to represent $|xj|$.

Greedy Algorithms: Greedy algorithms are group of algorithm used to find the sparsest solution of the p_1 problem. The greedy algorithms are based on finding the elements of the transform matrix called dictionary that best matches the signal through iterations. The greedy algorithm have much attention this last years and offer intermediate performance in term of execution time and required number of measurement. They are very fast and easy to implement.
Most used greedy algorithms are Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), Compressive Sampling Matching Pursuit (CoSaMP), Iterative Hard Thresholding (IHT) etc. In the following line, we give the pseudo-code of the CoSaM algorithm and the IHT algorithm which are two well-know greedy algorithm for signal recovery.

CoSaM algorithm The CoSaM algorithm is an iterative method which demonstrate that each iteration reduces the error in the current signal approximation. Bellow the pseudo-code of CoSaMP algorithm.

The CoSAMP algorithm is composed of five principles step de-script bellow [NT09].

Identification. Using the current samples, the algorithm computes a vector that is highly correlated with the signal, called the signal proxy.

Support Merger The set of newly identified components is united with the set of components that appear in the current approximation.

Estimation. The algorithm solves a least-squares problem to approximate the target signal on the merged set of components.

Pruning. The algorithm produces a new approximation by retaining only the largest entries in this least squares signal approximation.

Sample Update. Finally, the samples are updated so that they reflect the residual, the part of the signal that has not been approximated.

Iterative Hard Thresholding (IHT): This algorithm is also a method of resolution of the 2.4.4 by iteration. We consider $\hat{x}_0 = 0$ and use the iteration:

$$\hat{x}_i = H_s(\hat{x}_{i-1} + A^T(y - A\hat{x}_{i-1}))$$

Algorithm 1 CoSaMP algorithm

sampling matrix A, sparsity s and measurement vector y

Ensure: An s-sparse approximation a of the target signal

$x_0 = 0, v \leftarrow y, k \leftarrow 0$

repeat $k \leftarrow k + 1$

    from signal proxy

    $x \leftarrow A^* v$

    Identify maximum components

    $w \leftarrow supp(y_{2s})$

    Merge support

    $T \leftarrow w \cup \{s^{k-1}\}$

    $b|_T \leftarrow A_T^+ y$

    $b|_{T^c} \leftarrow 0$

    prune approximation

    $s^k = b_s$

    Update current samples

    $v = y - As^k$

until halting criterion is true

where A^T is the transpose of A and $H_s(A)$ is the non linear operator that sets all but the largest (in magnitude) s element of A to zero.

Algorithm 2 Iterative Hard Thresholding (IHT)

Input signal sparsity s, CS matrix A, measurement vector y

Ensure: $\hat{x} = x_i$

$x_0 = 0$

for i in 1, ...stopping criterion is meet do **do**

$x_i = H_s(x_{i-1} + A^T(y - Ax_{i-1}))$

end for

2.4.7 Numerical example of signal recovery. Example

In this section, we give a numerical example of signal recovery using compressed sensing. We consider an artificial sound wave signal x such as:

$$x(t) = \sin(1394\pi t) + \sin(3266\pi t) \tag{2.4.5}$$

The time t is generated in an interval $[0, n]$ with a difference of $1/8$ where n=5000. The signal x is represented by Figure 2.4 (a). As we can see, the signal $x = (x_1, x_2, ..., x_{5000})$ is not sparse. Thus, x is multiplied by the Fourier transform matrix Φ to obtain a sparse signal given by Figure 2.4 (b). We realized that the spectrum is mostly zero except for the two spikes representing the two sine frequencies.

We now sample 10% of the original x and store it in a vector y represented by the following Figure 2.4 (sampled signal).

We want to recovery x from the sample y. In this context, compressed sensing is possible by the fact the signal frequency content is highly sparse.

For that, we first generate the sensing matrix A such that $y = Ax$. Here we use the Gaussian matrix where each entry follow a normal distribution of mean 0 and variance 1. Using the l_1 minimization, we research the sparsest solution $\hat{x} = \min\|x\|_1$: $y = A\Phi x$ where Φ is the Fourier transform matrix. After solving this minimization problem, we obtain $\hat{x}$ which is the recovered signal 2.4 (recovered signal).

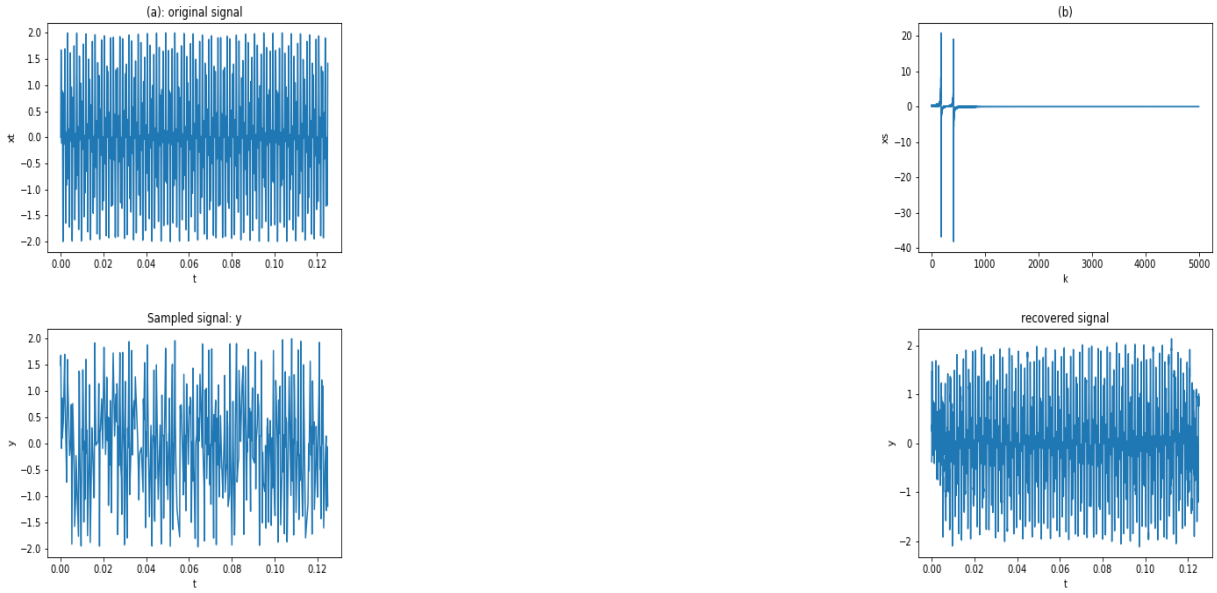


Figure 2.4: Signal recovery

From this figure above, we can remark that $\hat{x}$ is a good approximation of x. Observing figure below with is the mean quadratic error $(x - \hat{x})^2/n$, we can see that in the extremes, the error is high. This can be maybe due to the fact that our sample doesn't respect the periodic boundary condition requirements. In regard to the result above, compressed sensing give a good approximation of signal $x \in \mathbb{R}^n$ from a

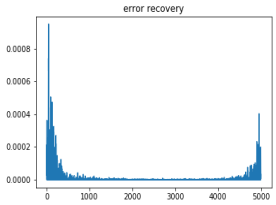


Figure 2.5: error signal recovery

sample $y \in \mathbb{R}^m$ where $m \lll n$.

For applying this method on networked data, we give in the following chapter a small review on complex network.

3. Review on Complex Network

Every day in our life, we are in relationship with network. Social network, professional, family network, telephone network among others are all types of networks. Thus, the concept of network is applied in many fields such as mathematics, physics, computer science etc. As soon as we hear the word network, we think of a group of interconnected item. In this chapter, we give a definition of network and complex network, define the different type of complex network and give some properties of complex network. **Definition** (network) A network also known as graph G, is a pair (V, E). V is called the vertex set of G, its elements are the vertices of G (also known as nodes). $E \subseteq \{(u, v,), u, v \in V\}$ is a set of edges connecting vertices.

3.1 The different type of graph

There exist different type of graphs such that:

3.1.1 Null graph. A null graph is a graph without edges. In the figure bellow we have an example of null graph with four vertices.

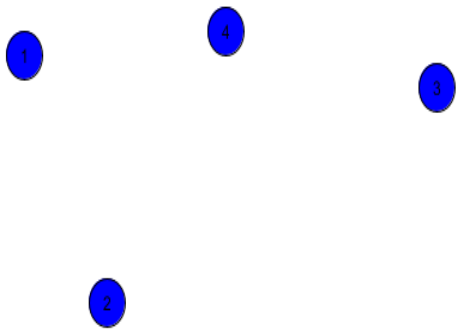


Figure 3.1: Null Graph

3.1.2 Trivial Graph. A graph with only one vertex is call trivial graph.

Figure 3.2: Trivial Graph

3.1.3 Simple graph. A simple graph is a graph with non loop and no parallel edges The graph above

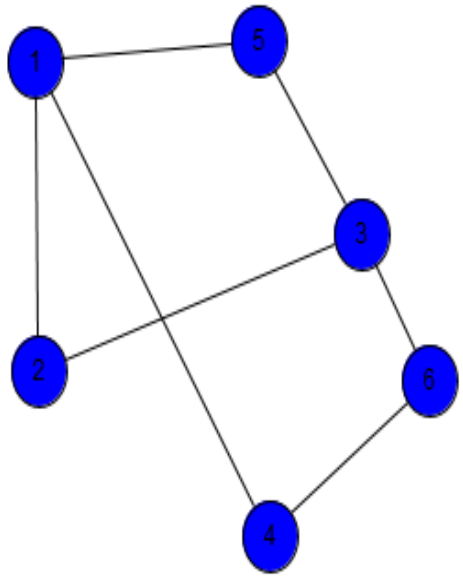


Figure 3.3: simple graph

is a simple graph with six vertices and seven edges.

3.1.4 Non-directed graph. A non-directed graph is a graph without direction, where the edges are bidirectional.In that each edge can be traversed in both directions. This figure shows a simple undirected graph with four nodes and three edges.

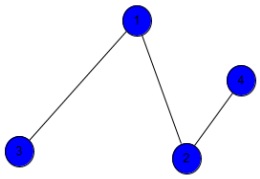


Figure 3.4: non-directed graph Graph

3.1.5 Directed graph. A directed graph is a graph with directed edges. A directed edge is an edge where the endpoints are distinguished, one is the head and one is the tail. In particular, a directed edge is specified as an ordered pair of vertices u, v and is denoted by $u \leftarrow v$. In this case, u is the tail of the edge and v is the head. Bellow a example of a directed graph

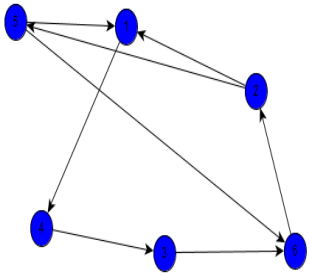


Figure 3.5: directed graph Graph

3.1.6 Weighted graph. A weighted graph is define as a triple G=(V, E, W), where V is the set vertices, E is the set of edges and W is a matrix that carries the edges weights.
The figure bellow is a example of weighted graph. In the figure 3.6, the weight of the edge liking the

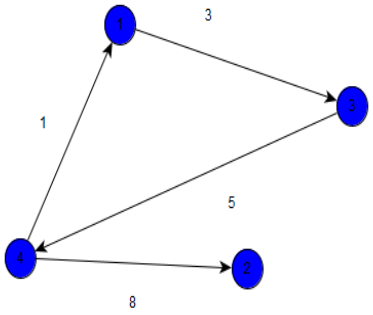


Figure 3.6: weighted graph

vertices 1 and 4 notice by $w_{4,1}$ is 1, the weight of the edge liking the vertices 1 and 4 notice by $w_{4,2}$ is 8, the weight of the edge liking the vertices 3 and 4 notice by $w_{3,4}$ is 5, the weight of the edge liking the vertices 1 and 3 notice by $w_{1,3}$ is 3. A weight of a edge can have various type of meaning in different application. For instance, a weight can represent a distance between two vertices, a traffic flow and so on.

3.2 Connectivity of a graph

3.2.1 adjacent vertices. Two vertices $u \in V$ and $v \in V$ are called adjacent if they share a common edge, in which case the common edge is said to join the two vertices.

3.2.2 Remark. In undirected graphs, if u is adjacent to v, then v must be adjacent to u as well.

3.2.3 Neighbourhood of a vertex. The neighbourhood of a vertex $v \in V$, in a graph G is the set of vertices adjacent to v. The neighbourhood is denoted by $\mathcal{N}(v)$.

3.2.4 Degree of a vertex. In an undirected graph, the degree of a vertex v is the total number of vertices adjacent to v. The degree of a vertex v is denoted by k_v. We can equivalently define the degree of a vertex as the cardinality of its neighbourhood set and say that, for any vertex v,

$$k_v = |\mathcal{N}(v)| = |\{u : (u, v) \in E\}| = \sum_{u \in V} \chi_{(u,v)\in E}$$

For a directed graph, the degree of vertex is the sum of in-degree and out-degree. where In-degree k_v^{in} is

$$\sum_{u \in V} \chi_{v \in \mathcal{N}(u)} = \sum_{u \in V} \chi_{(u,v)\in E}$$

And out-degree k_v^{out} is

$$\sum_{u \in V} \chi_{u \in \mathcal{N}(v)} = \sum_{u \in V} \chi_{(v,u)\in E}$$

3.2.5 Degree distribution:. A degree distribution p_k is the fraction of nodes with degree k [**?**]. For instance if we have a graph with n nodes and n_k of them have are of degree k, the degree distribution is given by:

$$p_k = \frac{n_k}{n}$$

However, there is some special degree distribution often use in complex network such that: Poisson, Gaussian, exponential and power-law.

3.2.6 Clustering coefficient:. A clustering coefficient is the real number between zero and one with measure the degree which two nodes tends to cluster together. It is equal to zero if there is no cluster and one if the nodes of the network are disjoint.

There exist two version of this measurement: the global clustering and the local clustering coefficient. The global coefficient:

- The global coefficient is the number of closed triplets (or 3 x triangles) over the total number of triplets (both open and closed).

$$C = \frac{number\ of\ closed\ triplets}{number\ of all\ triplets (open\ and\ closed)}$$

- A local coefficient of vertex is given by the proportion of links between the vertices within its neighbourhood divided by the number of links that could possibly exist between them [New03].

  Let N_i the set of neighbourhood of the vertex i and k_i its number of neighbours, then the local coefficient is define by:

$$\frac{\{e_{jk} : v_{jk}, v_{jk} \in N_i, e_{jk} \in E\}}{k_i(k_i - 1)}.$$

3.3 Trees and Forest

3.3.1 Tree. A tree is a connected graph that has no cycles. In a tree, a leaf is a vertex of degree 1. An internal vertex is a vertex of degree at least 2.

3.3.2 Forest. A forest is an undirected graph in which all of its connected components are trees.

Figure 3.7: Tree (b), forest (a) with two trees

3.3.3 Spanning tree. If G is a connected graph, the spanning tree in G is a sub-graph of G which includes every vertex of G and is also a tree graph. The graph (b) of the figure 3.8 is a possible

Figure 3.8: Tree (a), forest (b) spanning tree

spanning of the graph (a). In this transformation process, we have removed the edges (1,3), (1,4) and

(2,4)

3.4 Matrix representation of a Network

Mathematically, a weighted or a no weighted graph G are represented by adjacency matrix A which is constructed from the vertex and edge sets.

3.4.1 Adjacency matrix. Let G=(V, E, W) a weighted graph then the adjacency matrix is define as bellow:

- the number of the vertices $|V|$ serves to stablish the dimension of the adjacency matrix which is $|V|\times\ |V|$.
- The edge set contributes to defining the entry values of the adjacency matrix in the following manner: the i, j-th entry of A is denoted as $A_{ij} = a_{ij} = W_{ij}$ where W_{ij} is the weight of the edge linking i to j. Formally, $\forall(i, j) \in E : a_{ij} \neq 0$ and $\forall(i, j) \notin E : a_{ij} = 0$.

The adjacency matrix takes the following matrix form:

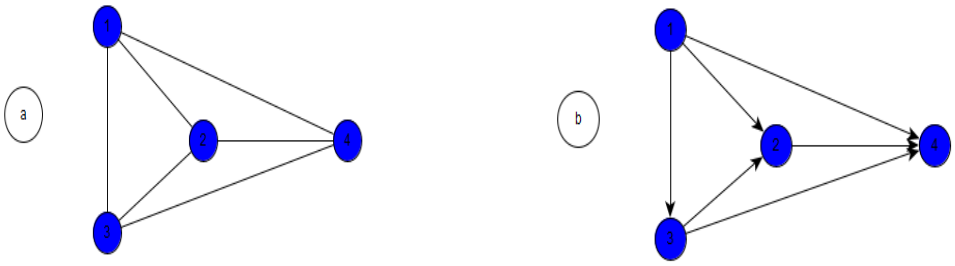
$$\begin{bmatrix} a_{11} & a_{11} & a_{12} & a_{13} & \cdots a_{1|V|} \\ a_{21} & a_{21} & a_{22} & a_{23} & \cdots a_{2|V|} \\ a_{31} & a_{32} & a_{33} & a_{34} & \cdots a_{3|V|} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{|v|1} & a_{|v|2} & a_{|v|3} & a_{|v|4} & \cdots a_{|V||V|} \end{bmatrix}$$

3.4.2 Remark. If a graph G is no weighted then $a_{ij} \in \{0, 1\}$.

3.4.3 Remark. If a graph G is undirected then the adjacency matrix A is symmetric This implies that if $a_{ij} = 0$,$a_{ji} = 0$

3.4.4 Remark. Contrasting to the previous remark, directed graphs may not have symmetric adjacency matrices, as j can be a neighbour of i and the converse may not hold.

The undirected graph show in graph 3.9 has the following adjacency matrix:



3.4.5 Example.

Figure 3.9: (a) undirected graph, (b) directed graph introduce for evaluating their adjacency matrices

$$A = A^T = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

in which the A^T represent the transpose matrix. And the directed graph has the following adjacency matrix:

$$\text{A=} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3.4.6 Degree matrix.

3.4.7 Incidence matrix. Let consider a graph G =(V, E) a graph where $V = \{v_1, ..., v_n\}$, $E = \{e_1, ..., e_n\}$ with $e_i = (u_i, v_i)$. So the incidence matrix is define

Let us consider an arbitrary orientation of every edge in the network, we label each edge v_i, v_j in a way that v_i is the positive end and v_j is the negative end. So the incidence is define as:

$$\begin{cases} 1 & if\ node\ v_i\ is\ the\ positive\ end\ of\ the\ edge\ e_j \\ -1 & if\ node\ v_i\ is\ the\ negative\ end\ of\ the\ edge\ e_j \\ 0 & otherwise. \end{cases}$$

3.4.8 Distance matrix.

3.4.9 Laplacian matrix of a graph. Let G be a graph. The Laplacian matrix of G, denoted L(G), is defined by L(G) $= \Delta(G) - A(G)$, where $A(G)$ is the adjacency matrix of G and $\Delta(G)$ is the diagonal matrix whose (i,i) entry is equal to the degree of the ith vertex of G. The Laplacian matrix of a graph carries the same information as the adjacency matrix obviously, but has different useful and important properties, many relating to its spectrum.

3.4.10 Theorem. *The Laplacian matrix of a graph G is a positive semi definite matrix.*

3.4.11 Examples of real Networks.

1. Social Networks: A set of people or groups of people with some pattern of contacts or interactions between them. For example friendship network such that Facebook, business between companies, intermarriages beween families, labor markets.

2. Technological network: Technological networks are networks maid by human. It aims to transfer services, resources or commodities such as transport, electricity among others.

4. Biological networks: Biological networks are network in relation with human body, animal and their way survival, chemistry etc.

5. In formation network or knowledge networks: The classic example of an information network is the network of citations between academic papers. Most articles cite previous work in the related topic. These citations form a network in which the vertices are articles and a directed edge from article A to article B indicates that A cites B. The structure of the citation network then reflects the structure of the information stored at its vertices, hence the term information network [New03]. The figure 3.10 give some example real networks.

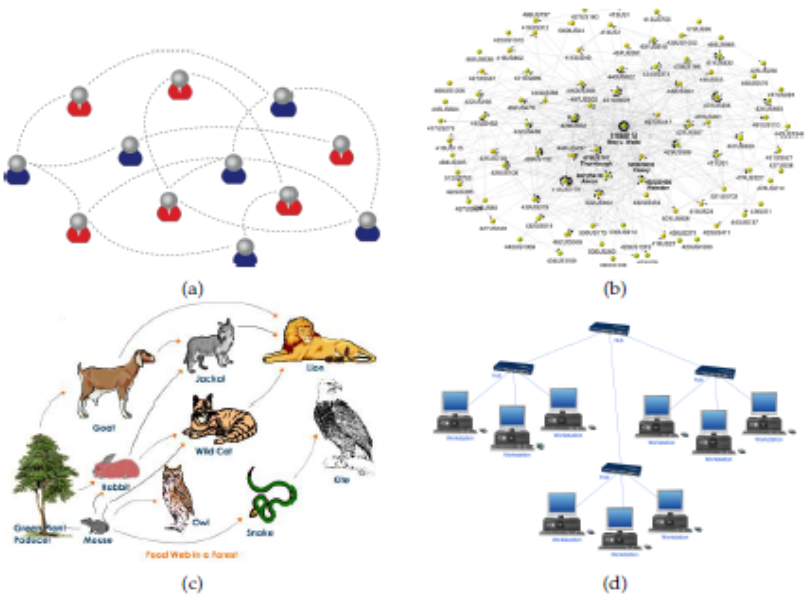


Figure 3.10: Networks in real world: (a) A social network. (b) A citation network. (c) A food web. (d) Computer network.

3.5 Complex Network Models

Different type networks exist and are characterize their topology. For the purpose of studding the topological properties that are linked to real network, several network model have been proposed. Some of these models even have inspired an extensive study due to its features of great interest. As examples of important categories of networks, one can list: random network, random clustered network, wireless sensor network and multi-hop network. In the following subsection, we review these network in detail.

Random Network or random Erdos-Renyi Erdos and Renyni(ref) developed a model that generate a random network with V vertices and E edges. Starting from V vertices completely disconnected (no edges in the network), the network is built from the gradual addition of L edges randomly created with a uniform probability p, in such a way that self-looping is avoided. In random network the most node have the same number of connection(low heterogeneity) and the degree distribution can be either constant or exponential.

Notice that arbitrary edge is present in a random network with a probability p and is absent with a probability p-1. Thus, $\binom{v-1}{k}p^k$ is the probability of k vertices to have exact k other interconnected vertices.

Figure 3.11: An example of random directed and undirected networks with V=30, distribution probability p=0.4

Random clustered network: A random clustered network is a network organize by communities. This communities consist sets of vertices that satisfy the rule: vertices belonging to the same community have many interconnecting edges, while different communities share relatively few edges interconnecting each other.

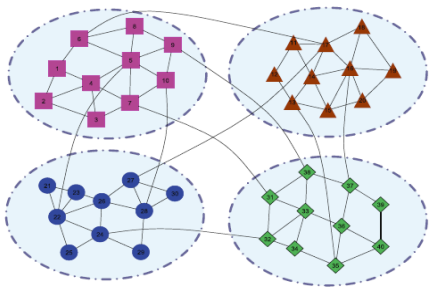


Figure 3.12: Schematic of a random clustered network with four well-defined communities. Each community is distinguished by a unique color or format.

For instance, real network like social media, biological network and worldwide web present this structure of community. In the language of social network, two of your friends will have the greater probability of knowing one another than will two people chosen at random from the population, on account of their common acquaintance with you. There exist many methods of community detection such as:

- The hierarchical method which is a traditional method. It consisted to calculates a weight W_{ij} for every pair i,j of vertices in the network, which represents in some sense how closely connected the vertices are.
- Edge "Betweenness" and community structure. This method was proposed by M. Girvan and

M. E. J. Newman in(ref). This method consist of grouping V initially isolated vertices into M communities. This is done by creating a link between two vertices with a probability p_{in}, if they belong to the same community and with a probability p_{out} if they belong to distinct communities High values of p_{in} and low values of p_{out} refer to networks with well-defined communities, it means there is a high concentration of edges confined within each community and very few edges interconnecting different communities.

Scale free network: A network is called scale-free if the characteristics of the network are independent of the size of the network, i.e. the number of nodes. That means that when the network grows, the underlying structure remains the same. A scale free network has a power law degree distribution[New03].

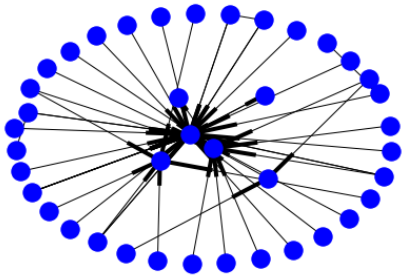


Figure 3.13: Scale free network of parameter with 40 nodes , alpha=0.41, beta=0.54, gamma=0.05, $delta_{in}$=0.2, $delta_{out}$=0

Alpha: Probability for adding a new node connected to an existing node chosen randomly according to the in-degree distribution;

beta: Probability for adding an edge between two existing nodes. One existing node is chosen randomly according the in-degree distribution and the other chosen randomly according to the out-degree distribution;

gamma: Probability for adding a new node connected to an existing node chosen randomly according to the out-degree distribution;

$delta_{in}$: Bias for choosing nodes from in-degree distribution;

$delta_{out}$: Bias for choosing nodes from out-degree distribution.

Watts-Strogatz: Watts-Strogatz network is a random network that produces graphs with small-world properties, including short average path lengths and high clustering [New03]. It was proposed by Duncan J. Watts and Steven Strogatz in 1998. This model starts with a circulant (or ring) network with n nodes

connected to k nearest neighbours. With a fixed probability p, an end to each original link is rewired to a new randomly selected node. The Figure 3.14 gives an example of Watts-Strogatz network.

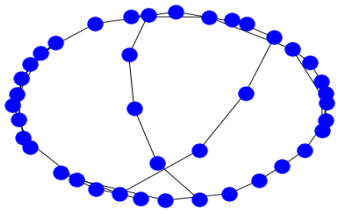


Figure 3.14: Watts-Strogatz,

Wireless Sensor Network: Wireless Sensor Network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind speed and direction, pressure, precision agriculture, etc. The wireless sensor network allow to extract useful information from distributed data and deliver it to a distant destination, called the fusion center (FC). The figure 3.15 bellow is an example of wireless sensor network fusion center. This wireless sensor deliver information about river water contamination.

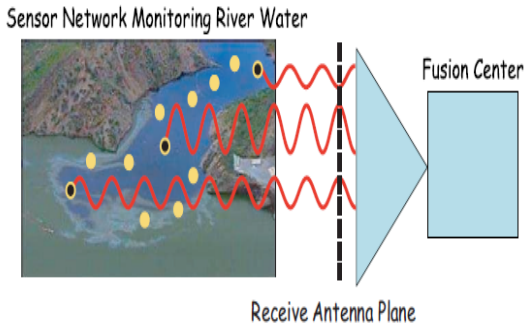


Figure 3.15: wireless sensor monitoring river water

Multi-hop network A multi-hop network is the network with one or more intermediate nodes along the path that receive the data. Multi-hop wireless networks have several benefits: Compared with networks with single wireless links, multi-hop wireless networks can extend the coverage of a network and improve connectivity.

4. Compressed Sensing for Networked data

A impressive quantity of data is generated by our system: telephones, computers, social media etc. In fact, all those systems are linked to internet and every wired or wireless device is capable of generating and disseminating prodigious volume of data. Today it is possible and even frequent to be in front of a system with million of component which are dependent or independent each of other. Managing such quantity of data, despite the existing progress in distributed data processing, remain a big challenge.

In the other hand, compressed sensing is very developed in sparse signal recovering as we show in our chapter two.

In this chapter, we are interested on recovery networked data from small sample. This reconstruction base on the technique of compressed sensing request some conditions specially the sparsity of the data. Thus, in our first section, we show how to sparsify data of a network for instance multi-hop network and wireless sensor network. In our second section, we show the network data compression process in the form $y = Ax$. In the last section we give an example.

4.1 Sparsity of Networked data

As we show in chapter 2, sparsity of the signal is necessary condition to recover a signal from compressed data. If the vector does not exhibit sparsity, we are required to sparsify it by choosing an appropriate representation system in this field typically coined dictionary. Based on some physics fact, it is always possible to find transform basis where the data are sparse. Several method of sparse representation of the data in a specific basis were proposed in [HBRN08]. It consist of Spatial Compression, Graph Wavelets and Diffusion Wavelets. Before we study those methods, we give the definition of some notions that we will use.

4.1.1 Definition. Let G = (V, E) be a finite simple graph with non-isolated vertices for which $|V| = n$, and let A be an adjacency matrix of G. We define the matrix D $\in \mathbb{R}^{n\times n}$ s.t $D_{i,i} = \sum_{i=1}^{n} A_{i,j}$ and $D_{i,j} = 0$ for $i \neq j$

Let M be the column normalized of the adjacency matrix defined as $M = AD^{-1}$.

4.1.2 Definition. (Diffusion matrix()) Let $X \subset \mathbb{R}$ be the set of column stochastic matrices. The diffusion character mapping is defined to be $f : (0, 1) \times X \to \mathbb{R}^{n\times n}$ such that f(α, M)= $(I_n - \alpha M)$. $f(\alpha, M)$ is call diffusion matrix.

4.1.3 Spatial Compression. This method is considered as an image or a signal sparsity. In fact, consider that a wireless sensor which monitor a spatially-varying phenomenon such as temperature, earthquake, moisture etc. The measured data can be see as image or signal.

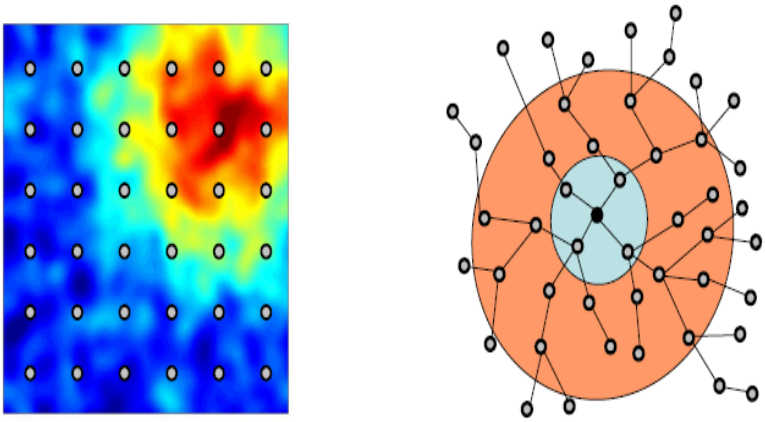


Figure 4.1: (a) is monitored by a network of wireless sensors deployed uniformly over the region. In (b), the graph (Haar) wavelet coefficient at the location of the black node and scale three is given by the difference of the average data values at the nodes in the red and blue regions [HBRN08].

The Figure 4.1 is an illustration of wireless sensor data location. The measured data constitute a sample of the network data.

However, the sparsity of the data considered here as image or signal, can be obtain by using the traditional signal processing such as the discrete Fourier transform or the discrete wavelet transform. For instance, let $x[n]$ represent a signal and Φ the Fourier transform matrix or the transform wavelet matrix then, Φx is a sparse representation of x.

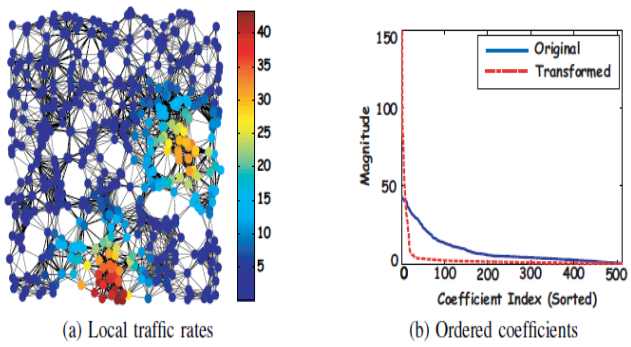
4.1.4 Diffusion Wavelets. Diffusion wavelets have been constructed on graphs in order to allow an efficient multi-scale representation. The diffusion wavelets seek to generalize the construction of the wavelet bases to soften the varieties where there is still a natural notion of translation given by the local coordinate [Zha16]. They give a possibility to create an orthonormal basis for a function on a graph. This basis is obtain by analysing the eigenvectors of a diffusion matrix T generated from the adjacency matrix. The resulting basis vectors are generally localized to neighbourhoods of varying size and may also lead to a sparse representation of data on a graph.

4.1.5 Theorem ([Zha16]). *If each basis ϕ_j of scaling functions that is build is such that the elements*

with large support (large layer index l) are in a subspace spanned by eigenvectors of T corresponding to very small eigenvalues (depending on l), then these basis functions of large support will not be needed to compute the next (dyadic) power of T. This implies that the matrices representing of all the basis transformations will be uniformly sparse.

In this theorem, T is the diffusion operator of a graph with large power of low rank.

The following graph[HBRN08] give an example of sparsification using Diffusion wavelet. In this graph,



(a) Local traffic rates (b) Ordered coefficients

Figure 4.2: Schematic of a random clustered network with four well-defined communities. Each community is distinguished by a unique color or format.

the nodes represent the traffic rate through router in computer network. Observing the graph 4.2 Figure (a), we can notice that there a small number of nodes where the traffic rate is important. And in all the remaining network, the traffic rate is small and can be minimize. It means that the small number of nodes where the traffic level is high can be use to sparsely represent the traffic data in this network.

4.1.6 Graph Wavelets. This method can be seen as a general case since it allow to deal with an arbitrary graph. In fact, the structure of the networks are different and can be very complex. Variation in a network usage, component failure or miss configuration etc, can affect the existing correlation between the traffic level at different nodes. For instance, let us consider the Fourier transform used in the above method. If the frequency of the phenomenon is not constant, it doesn't work well. However, the graph wavelets method adapt the diffusion wavelets transform in an arbitrary graph. Graph wavelet coefficients are then defined by aggregating data at different scales, and computing differences between aggregated data, as shown in Figure 4.1 (b).

4.2 Networked data compression

As we can see in chapter two, the image or signal to recovery is a vector. Thus, to be in the condition to apply the CS method, two techniques of projecting networked data in a vector y of form $y = Ax$ were propose in [HBRN08]. The first technique consider a general multi-hop network such wireless sensor network, wired local area network, weather or agricultural monitoring network or even portion of internet. The second is motivated by many wireless sensor networks where explicit routing information is difficult to obtain and maintain.

4.2.1 multi-hop network. In this first technique, we compute the CS projection of the form $y_i = \sum_{i=1}^{n} A_{i,j}x_j$ in a general multi-hop network. Each y_i as linear combination of $A_{i,j}x_j$, is easily computed and in a efficient and decentralized way. Thus, two steps were proposed in [HBRN08] for computation and distribution of each CS sample $y_i, i = 1, .., k$, $k \ll n$.

1. **Step1:** Each of the n sensors, $j = 1, ..., n$, locally computes the term $A_{i,j}x_jj$ by multiplying its data with the corresponding element of the compressing matrix. The compressing matrix can be generated by letting each node locally generate a realization of $A_{i,j}$ using a pseudo-random number generator seeded with its identifier in a distributed fashion. Here, the integer $j = 1, ..., n$ serve as this identifier. Given the identifiers of the nodes in the network, the requesting node can also easily reconstruct the random vectors $\{A_{i,j}\}_{i=1}^{k}$ for each sensor j = 1, . . . , n.

2. **Step2:** In this second step, using the randomized gossip algorithm that we will define later, the y_i are redistributed across the network. So, each node exchange information with its neighbours and at the end of the algorithm, the sample y is available at every node in the network. Thus, considering one node of the network, we are able to access to the compressed data $y = Ax$.

In the flowing lines, we show how the gossip algorithm works for a data distribution.

Gossip algorithm for data distribution: The graph 4.3 show one example of gossip iteration, where the color of a node corresponds to its local value. First, the network is initialized to a state where each node has a value $x_i(0), i = 1, ..., n$. Then in an iterative, asynchronous fashion, a random node a is activated and chooses one of its neighbours b at random. The two nodes then gossip they exchange their values $x_a(l) and x_b(l)$, or in the CS setting the values multiplied by pseudo-random compression vector elements, and perform the update $x_a(l+1) = x_b(l+1) \leftarrow x_a(l) + x_b(l)$ while the data at all the other nodes remains unchanged. The bottom panel shows an example network of 100 nodes with (left) random initial values, (middle) after each node has communicated five times with each of its neighbours, and (left) after each node has communicated 50 times with each of its neighbours It can be

shown that for this simple procedure, $x_i(l)$ converges to the average of the initial values, $1/n \sum_{j=1}^{n}$, at every node in the network as l tends to infinity as long as the random choice of neighbours is sufficient to ensure that information will eventually propagate between every pair of nodes [HBRN08].

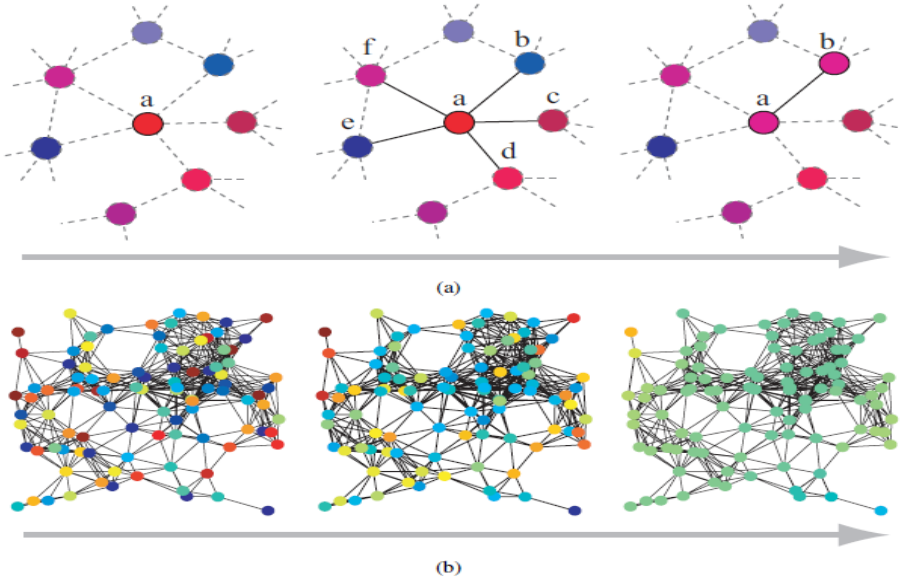


Figure 4.3: Example of gossip algorithm

4.2.2 Wireless sensor network. To determine the projection vector in a wireless sensor network the following two step were define in [HBRN08]:

- **Step 1:** In this first step, each sensor of the network, generate s elements of the random vector $A_{i,j}$, $i = 1, ..., s$. The s elements of $A_{i,j}$ are generated using the network address which constitute a seed of pseudo random number generator.

- **Step 2**: After having the s elements of $A_{i,j}$, each sensor multiply data x_j by the s element. It mean, it generate a vector

$$t_j = (A_1 x_j, ..., A_s x_j)^T \tag{4.2.1}$$

  j= 1,...,n. And this t_j is transmit to the FC over the network using k time slots transmission. After the k transmission by each sensor, the corresponding signal received at the FC is the sum of the s-tuple t_j so we have:

$$y = \sum_{j=1}^{n} t_j + \epsilon = Ax + \epsilon. \tag{4.2.2}$$

  Where ϵ represent the a noise generate by the communication receiver circuitry of the FC.

So, the data at the FC is the sum of the s-tuple t_j given by:

$$y = \sum_{j=1}^{n} t_j + \epsilon = Ax + \epsilon. \tag{4.2.3}$$

Which is the measurement or sample of the data produce by the wireless sensor.

In all the two cases, measurement or sample y is in the form $y = Ax$ or $y = Ax+\epsilon$ for taking account the noise produce during transport communication from the sensor to the FC for a wireless sensor network. However, the data x been already sparsify by the one the who methods define above, the technique of compressed sensing is applicable for the reconstruction of the sparse data x from the sample y. Therefore, the solution of the problem

$$\min_x \|_0 x\| : y = Ax$$

give a exact reconstruction of x. As show in chapter 2, this solution is obtain by using one of the algorithms define in that chapter.

4.3 Example

In this example we consider a network of n=16 sensor. One distinguished sensor observes some positives values while the n-1 remaining observe 0. The objective here is to identify which sensor is different using as few observation as possible. To do this, a random projection observation is made onto vectors of values -1, 1 each with probability 1/2. This constitute a binomial low of parameters n, 1/2. The expectation of the number of hypothesis sensors that are consistent with each particular observation is n/2. If we consider k observations, the anomalous sensor is the intersection of the k hypothesis given by the k observations. For each observation, the number of nodes is divided by 2 until it attains 1 for the last observation. This can be translated by:

observation 1: $\frac{n}{2} \rightarrow$ observation 2: $\frac{n}{2^2} \rightarrow \ldots \rightarrow$ observation h: $\frac{n}{2^h} = 1$.

Since h is the number of observation, according to the last step, about $\log n$ observation are required before the estimate anomalous sensor.

Now let define $\|x\|_0$ the number of non-zero entries in the vector x and $y = Ax$ where A is the matrix of column the vector where the data are projected for each observation it mean the matrix of entries $-1, 1$. Thus, A is a $m \times n$ random Bernoulli matrix of entries -1, 1 where $m = \log_2 n = 4$ and $n = 16$. Therefore, the above problem can be redefine as follow:

$$\min_x \|x\|_0 : \quad y = Ax \tag{4.3.1}$$

Using the compressed sensing method, we can proximate the find the vector with the smallest number of non-zero values.

Conclusion

In this essay, the focus was made on how compressed sensing can be apply for reconstruction of networked data especially wireless sensor and multi hop networks from a small sample. Given the progress of compressed sensing for sparse image/signal recovery, we tried to show the way of creating sparse data in a network by projection in a specific basis. As well as the process of compressing a sample x in the form $y = Ax$ was showed. In others words, our main focus was how to satisfy the necessary and sufficient conditions for applying the technique of compressed sensing for networked data recovery. Thus, the sparsity of the data is obtain by multiplying the data of the network by the transform matrices such as Fourier transform matrix for a wireless sensor network, diffusion matrix and graph wavelets matrix for an arbitrary network. However, this technique is new in the domain of networked data and give a lot of possibility for feature research. In the feature, it can be interesting to use this method in a real network data such as social media with the different algorithms define in chapter 2 and compare their performance.

Acknowledgements

I give thanks to Allah who gave the strength and courage to finish this work. I would like to express my deep gratitude to Dr Bubacarr Bah and Dr Franck Kalala Motumbo, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. It is a big pleasure to work with you. A Big thank to all my family and all the AIMS staff especially to my tutor Mbaye faye. Many Thank to my friends and fellow at AIMS Senegal.

References

[AYSS] Tuncer Can Aysal, Mehmet Ercan Yildiz, Anand D Sarwate, and Anna Scaglione, *Broadcast gossip algorithms for consensus*, IEEE Transactions on Signal processing **57**, no. 7, 2748.

[CDD09] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore, *Compressed sensing and best term approximation*, Journal of the American mathematical society **22** (2009), no. 1, 211–231.

[CRT06] Emmanuel J Candès, Justin Romberg, and Terence Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on information theory **52** (2006), no. 2, 489–509.

[DDEK11] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok, *Introduction to compressed sensing*, preprint **93** (2011), no. 1, 2.

[Don06] David L Donoho, *Compressed sensing*, IEEE Transactions on information theory **52** (2006), no. 4, 1289–1306.

[DOS17] Andjela Draganic, Irena Orovic, and Srdjan Stankovic, *On some common compressive sensing recovery algorithms and applications-review paper*, arXiv preprint arXiv:1705.05216 (2017).

[EK12] Yonina C Eldar and Gitta Kutyniok, *Compressed sensing: theory and applications*, Cambridge University Press, 2012.

[EK15] Ernesto Estrada and Philip A Knight, *A first course in network theory*, Oxford University Press, USA, 2015.

[Est12] Ernesto Estrada, *The structure of complex networks: theory and applications*, Oxford University Press, 2012.

[FR13] Simon Foucart and Holger Rauhut, *A mathematical introduction to compressive sensing*, vol. 1, Birkhäuser Basel, 2013.

[HBRN08] Jarvis Haupt, Waheed U Bajwa, Michael Rabbat, and Robert Nowak, *Compressed sensing for networked data*, IEEE Signal Processing Magazine **25** (2008), no. 2, 92–101.

[LM17] Ming-Jun Lai and Daniel Mckenzie, *A compressive sensing approach to community detection with applications*, arXiv preprint arXiv:1708.09477 (2017).

[New03] Mark EJ Newman, *The structure and function of complex networks*, SIAM review **45** (2003), no. 2, 167–256.

[NT09] Deanna Needell and Joel A Tropp, *Cosamp: Iterative signal recovery from incomplete and inaccurate samples*, Applied and computational harmonic analysis **26** (2009), no. 3, 301–321.

[OY14] Stanley Osher and Wotao Yin, *Sparse recovery via l1 and l1 optimization*, Tech. report, CALIFORNIA UNIV LOS ANGELES DEPT OF MATHEMATICS, 2014.

[PT16a] André Luiz Pilastri and Joao Manuel RS Tavares, *Reconstruction algorithms in compressive sensing: an overview*, 11th edition of the Doctoral Symposium in Informatics Engineering (DSIE¿ 16), 2016.

[PT16b] _____, *Reconstruction algorithms in compressive sensing: an overview*, 11th edition of the Doctoral Symposium in Informatics Engineering (DSIE¿ 16), 2016.

[QBI+13] Saad Qaisar, Rana Muhammad Bilal, Wafa Iqbal, Muqaddas Naureen, and Sungyoung Lee, *Compressive sensing: From theory to applications, a survey*, Journal of Communications and networks **15** (2013), no. 5, 443–456.

[Ste99] Mallat Stephane, *A wavelet tour of signal processing*, The Sparse Way (1999).

[STXM14] Tianlin Shi, Da Tang, Liwen Xu, and Thomas Moscibroda, *Correlated compressive sensing for networked data.*, UAI, 2014, pp. 722–731.

[Wan11] Xiaomin Wang, *Deciding on the type of the degree distribution of a graph (network) from traceroute-like measurements*.

[XM] Tianlin Shi Da Tang Liwen Xu and Thomas Moscibroda, *Correlated compressive sensing for networked data*.

[YLHX15] Penghang Yin, Yifei Lou, Qi He, and Jack Xin, *Minimization of 1-2 for compressed sensing*, SIAM Journal on Scientific Computing **37** (2015), no. 1, A536–A563.

[Zha16] Xi Zhang, *Design of orthogonal graph wavelet filter banks*, Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE, IEEE, 2016, pp. 889–894.