Report of Progect 2

Various Algorithms of CS

Cheng Peng	Baohua Liu	Qing Ding
5060309697	5060309697	5060309697

May 4, 2009

Abstract

In this report we will discuss the basic principles of CS and many algorithms.For some Algorithm we used MatLab to analyze its result and did some comparisons with them

Contents

1 Compressive Sensing:basic ideas and principles

Conventional approaches to sampling signals or images follow Shannons celebrated theorem: the sampling rate must be at least twice the maximum frequency present in the signal (the so-called Nyquist rate). In fact, this principle underlies nearly all signal acquisition protocols used in consumer audio and visual electronics, medical imaging devices, radio receivers, and so on. (For some signals, such as images that are not naturally bandlimited, the sampling rate is dictated not by the Shannon theorem but by the desired temporal or spatial resolution. However, it is common in such systems to use an antialiasing low-pass filter to bandlimit the signal before sampling, and so the Shannon theorem plays an implicit role.) In the field of data conversion, for example, standard analog-to-digital converter (ADC) technology implements the usual quantized Shannon representation: the signal is uniformly sampled at or above the Nyquist rate.

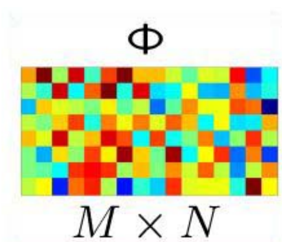
The theory of compressive sampling, also known as compressed sensing or CS, is a novel sensing or sampling paradigm that goes against the common wisdom in data acquisition. CS theory asserts that one can recover certain signals and images from far fewer samples or measurements than traditional methods use. To make this possible, CS relies on two principles: sparsity, which pertains to the signals of interest, and incoherence, which pertains to the sensing modality.

Sparsity expresses the idea that the information rate of a continuous time signal may be much smaller than suggested by its bandwidth, or that a discrete-time signal depends on a number of degrees of freedom which is comparably much smaller than its (finite) length. More precisely, CS exploits the fact that many natural signals are sparse or compressible in the sense that they have concise representations.

By create appropriate converting matrix we can compress the signal which has sparsity traits into a signal much less in the data amount.

for example, we have this image:

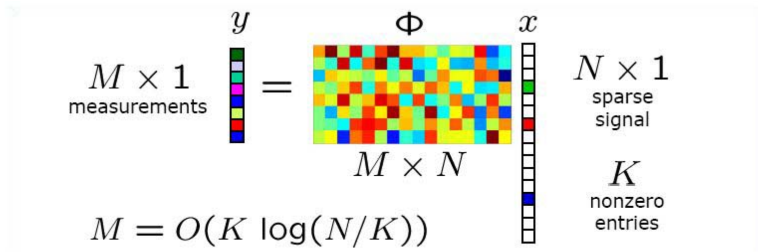
and we know that each line could be represented in an array of 1 line and N column.That is X.
Then we invent a matrix of M-line and N-column,designated as Φ:



Then we can get Y by

$$y = \Phi x$$

The whole plot is like:



Where K is how many non-zero number in X. Here we use the random Matrix and there's theory showing that the result is pretty good, It's enough to recover X from Y.
But there're prerequisites if we want to get a proximately identical signal to X from Y:
1. The signal X must be sparse enough:

$$M \gtrsim K \cdot \log N$$

2. The matrix must be Orthogonal.

Acording to experience,If we have M = 3 to 4, the signal is basically recoverable.
Now here comes the major problems of "How to recover the signal".There is plenty of algorithm doing this thing.

2 Recovery Algorithm

Now X0 is K-Sparse.$Y = \Psi \times X$
We need to solve this:

$$\min_{f} \ \|f\|_{\ell_1} \quad \text{subject to} \quad \Phi f = y$$

This is a L1 problem. Unarticulately when L1 = 0 the problem is very difficult. But there is mature theory solving the L1=1 problem. It is proved that under most circumstances they can be applied equally.
We have been provided with a Matlab Code package containing l1qc code. They are designed to solve the problem stated above.
The problems fall into two classes: those which can be recast as linear programs (LPs), and those which can be recast as second-order cone programs (SOCPs). The LPs are solved using a generic path-following primal-dual method. The SOCPs are solved with a generic log-barrier algorithm. The implementations follow Chapter 11 of [2]. For maximum computational effi-

ciency, the solvers for each of the seven problems are implemented separately. They all have the same basic structure, however, with the computational bottleneck being the calculation of the Newton step (this is discussed in detail below). The code can be used in either "small scale" mode, where the system is constructed explicitly and solved exactly, or in "large scale" mode, where an iterative matrix-free algorithm such as conjugate gradients (CG) is used to approximately solve the system.

- **Min-ℓ_1 with equality constraints.** The program

$$(P_1) \quad \min \|x\|_1 \quad \text{subject to} \quad Ax = b,$$

  also known as *basis pursuit*, finds the vector with smallest ℓ_1 *norm*

$$\|x\|_1 \ := \ \sum_i |x_i|$$

  that explains the observations b. As the results in [4, 6] show, if a sufficiently sparse x_0 exists such that $Ax_0 = b$, then (P_1) will find it. When x, A, b have real-valued entries, (P_1) can be recast as an LP (this is discussed in detail in [10]).

- **Minimum ℓ_1 error approximation.** Let A be a $M \times N$ matrix with full rank. Given $y \in \mathbb{R}^M$, the program

$$(P_A) \quad \min_x \ \|y - Ax\|_1$$

  finds the vector $x \in \mathbb{R}^N$ such that the *error* $y - Ax$ has minimum ℓ_1 norm (i.e. we are asking that the difference between Ax and y be sparse). This program arises in the context of channel coding [8].

  Suppose we have a channel code that produces a codeword $c = Ax$ for a message x. The message travels over the channel, and has an unknown number of its entries corrupted. The decoder observes $y = c + e$, where e is the corruption. If e is sparse enough, then the decoder can use (P_A) to recover x exactly. Again, (P_A) can be recast as an LP.

- **Min-ℓ_1 with quadratic constraints.** This program finds the vector with minimum ℓ_1 norm that comes close to explaining the observations:

$$(P_2) \quad \min \|x\|_1 \quad \text{subject to} \quad \|Ax - b\|_2 \leq \epsilon,$$

  where ϵ is a user specified parameter. As shown in [5], if a sufficiently sparse x_0 exists such that $b = Ax_0 + e$, for some small error term $\|e\|_2 \leq \epsilon$, then the solution $x_2^\star$ to (P_2) will be close to x_0. That is, $\|x_2^\star - x_0\|_2 \leq C \cdot \epsilon$, where C is a small constant. (P_2) can be recast as a SOCP.

- **Min-ℓ_1 with quadratic constraints.** This program finds the vector with minimum ℓ_1 norm that comes close to explaining the observations:

$$(P_2) \quad \min \|x\|_1 \quad \text{subject to} \quad \|Ax - b\|_2 \leq \epsilon,$$

where ϵ is a user specified parameter. As shown in [5], if a sufficiently sparse x_0 exists such that $b = Ax_0 + e$, for some small error term $\|e\|_2 \leq \epsilon$, then the solution $x_2^\star$ to (P_2) will be close to x_0. That is, $\|x_2^\star - x_0\|_2 \leq C \cdot \epsilon$, where C is a small constant. (P_2) can be recast as a SOCP.

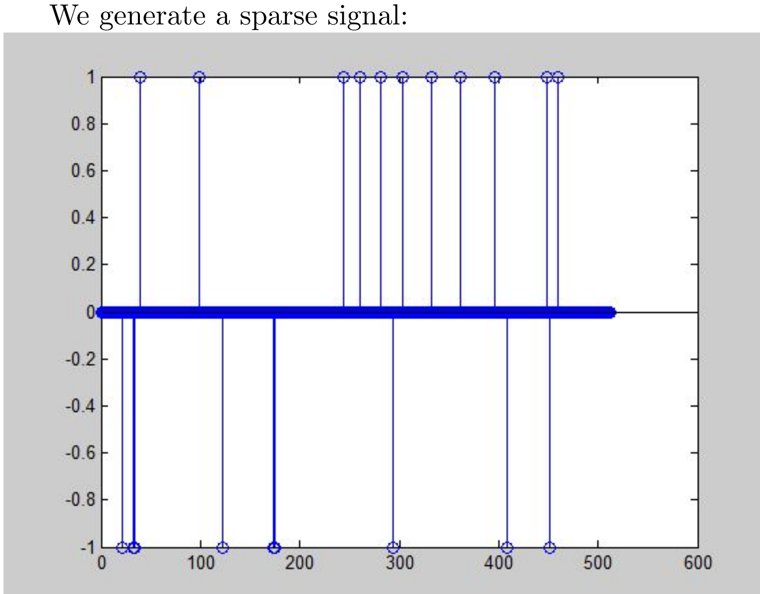
3 Simulation Algorithm:l1eq and Result

We use l1eq algorithm to simulate a compress and recovery process:
In The MatLab M file we wrote the following code to generate a random sparse signal:
N = 512;% number of spikes in the signal
T = 20; % number of observations to make
K = 120;

$$q = randperm(N);$$

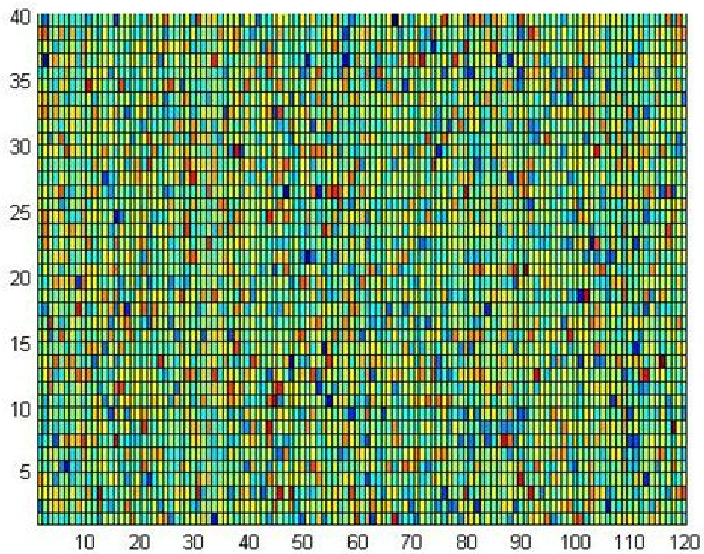
$x(q(1:T)) = sign(randn(T,1));$

We generate a sparse signal:

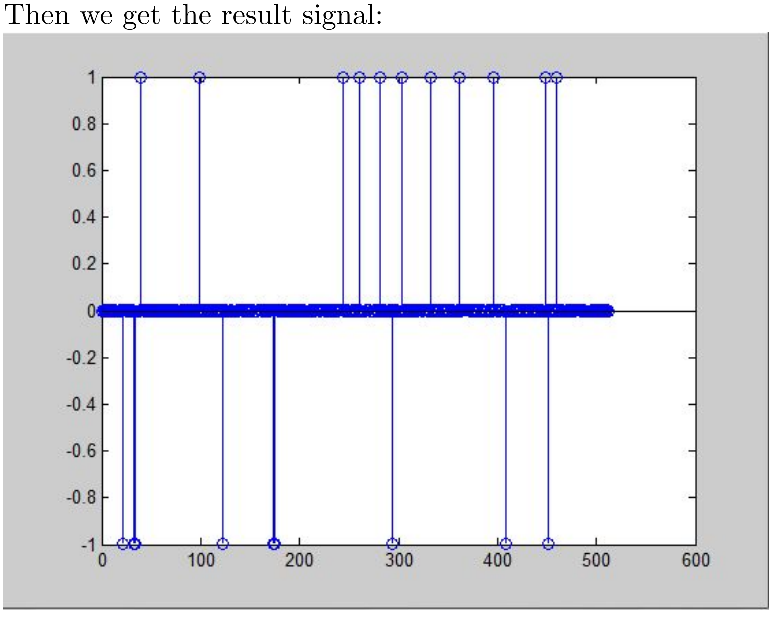


3.1 Gauss Measure Matrix 1

And we have the measure matrix as follows,each cell represent a numerical value ranging from 0 to 1 and they subject to Gauss (0,1) distribution:



Then we get the result signal:

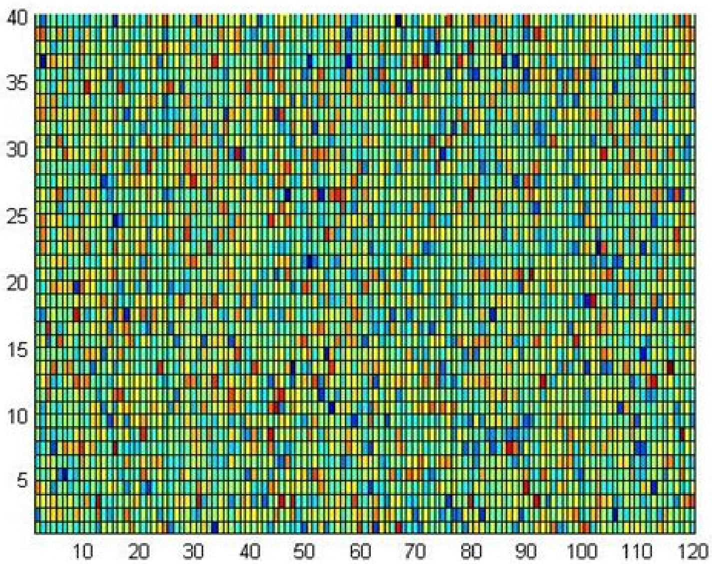


We can see that the result is pretty good,almost every sparse signal has been recovered.

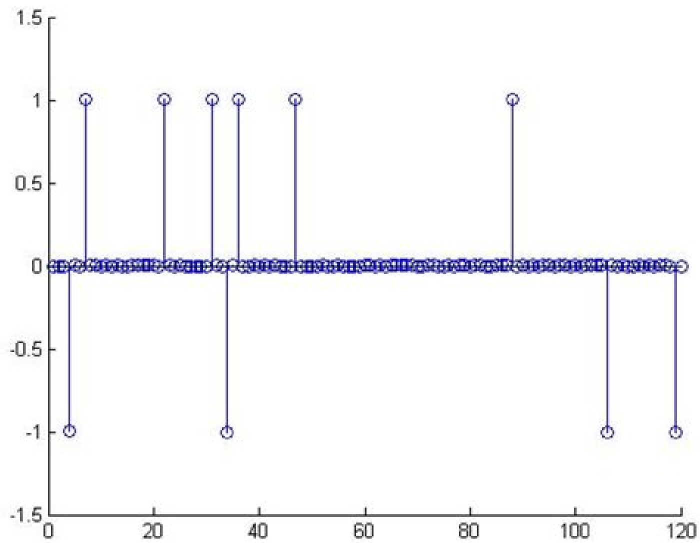
The MSE is 2.8470

3.2 Gauss Measure Matrix 2

Now we use a measure matrix in which value subject to Gauss (0,1/N) distribution:



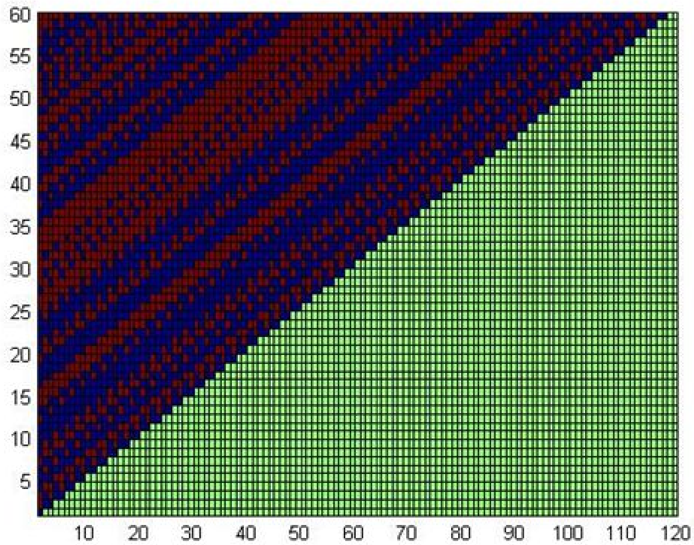
Then we get the result signal:



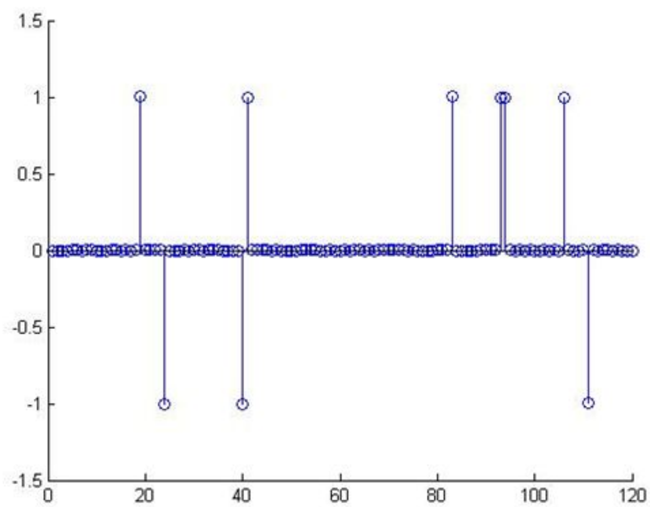
We can see that the result is as good as above, proving the robustness of Lqed algorithm.
The MSE is 5.9171e-005

3.3 PN Measure Matrix

We use PN measure matrix plotted as follows:



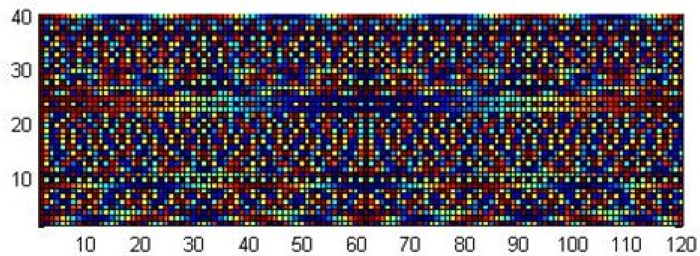
The result signal:

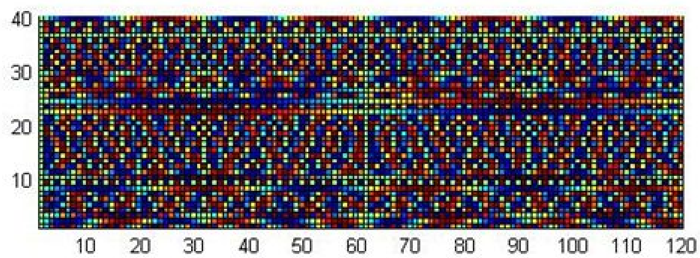


MSE:1.0000

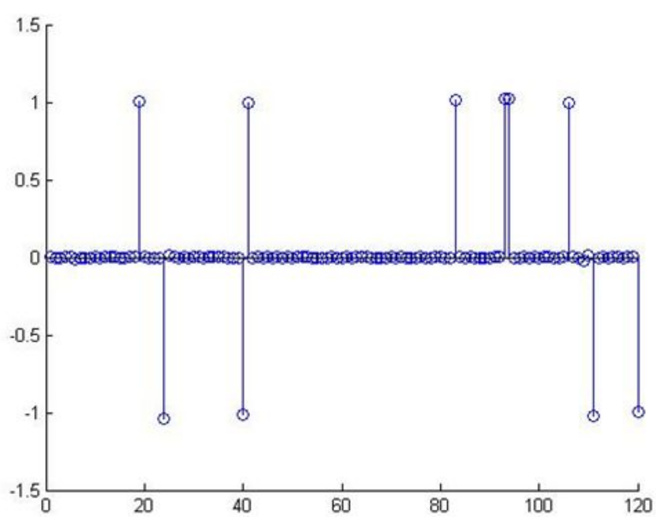
3.4 DFT Measure Matrix

Discrete Fourier Transform(DFT) matrix is the matrix where each cell is filled with Fourier Coefficient:$e^{-2\pi iKn/N}$



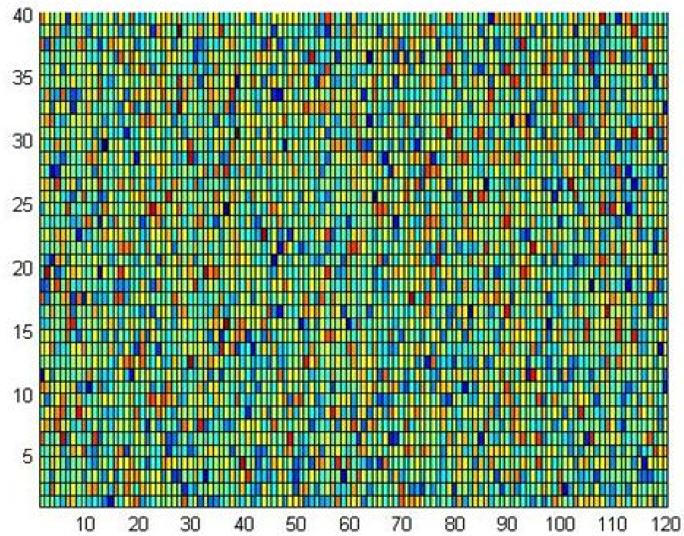


The result signal:

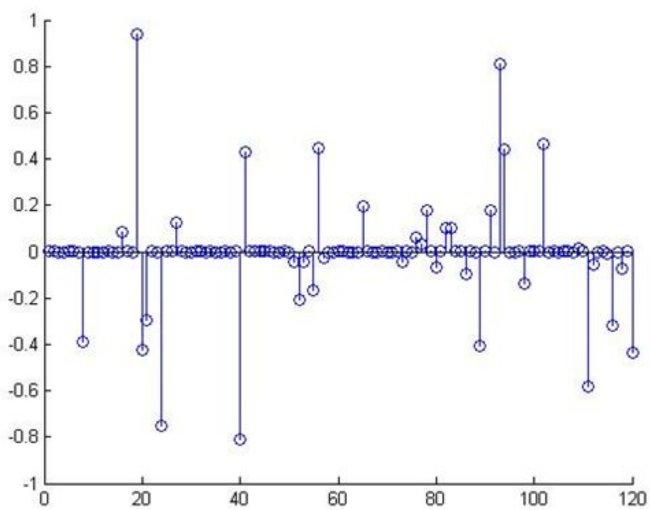


MSE:3.2136

3.5 Bernoulli Measure Matrix



Measure Matrix:



The result signal:
We can see that this Matrix did not do a good job.We guess that may be it's because the Orthogonalization is not fully satisfied.
MSE = 2.1068

4 Summary

In this phase we have familiarized the CS theory,its main problem and the recovery algorithm.Next phase our group would go to the details of recovery algorithm,try to understand how they work.Especially the greedy algorithms,the graphical models and Bayesian approaches.
After that we would take a look into the realistic concise signal models and see some application, to know what can be done by using CS theory.