Comparison of Reconstruction Algorithms for Sparse Signal Recovery from Noisy Measurement

Murat Emre Erkoc[1], Gokcen Ozdemir[2], Nurhan Karaboga[3]
1,2,3- Erciyes University, Electrical and Electronics Engineering Department, Kayseri, Turkey.
Email-1: merkoc@erciyes.edu.tr (Corresponding author)
Email-2: gozdemir@erciyes.edu.tr
Email-3: nurhan_k@erciyes.edu.tr

Received: January 2018 Revised: February 2018 Accepted: February 2018

ABSTRACT:
Compressive sensing is a technique that can reconstruct sparse signals under Nyquist rate. This study is about comparison of widely used sparse signal reconstruction algorithms under noisy measurements. Three algorithms, Orthogonal Matching Pursuit, Compressive Sensing Matching Pursuit and Primal Dual Interior Point method are used to reconstruct sparse signal from noisy measurement and performance results are compared. Firstly, a sparse signal is sampled under Nyquist rate and observation vector is obtained. After that, white Gaussian noise is added to this observation vector. Then, sparse reconstruction algorithms are employed to reconstruct the original signal from noisy measurement. These algorithms are tested for various measurement number and sparsity levels. Test conditions are same for all algorithms. Finally some performance metrics results related to reconstructed signal are obtained. These performance metrics are mean squared error, correlation of the reconstructed signal and original signal, reconstruction time of the algorithms and iteration numbers. According to these metrics, when sparsity level is very smaller than measurement number, Orthogonal Matching Pursuit has better results than others. However, when sparsity level is increased and close to measurement number, Primal Dual Interior Point method has better performance than others in terms of reconstruction a sparse signal from noisy measurement.

KEYWORDS: Compressed sensing, Greedy algorithms, L_1 minimization, Sparse signal reconstruction.

1. INTRODUCTION

Compressive sensing (CS) asserts that it can recover a sparse signal at a lower rate than Nyquist rate [1]-[3]. A signal is called sparse if much of its entries are zero; small numbers of its entries are nonzero. CS theory consists of three main processes: sparse representation, measurement and sparse signal reconstruction [4]. The first process is representing a signal as sparse in its original domain or suitable transform domain. Many natural signals are sparse in its original domain or transform domain like wavelet, Fourier or discrete cosine transform. A signal can be made sparse by using suitable ones of these transformations according to signal properties. In measurement process, signal is sampled at a lower number than signal length. This can be explained in mathematically as in (1).

$$y = \Phi x \quad (1)$$

In (1), $x \in R^N$ represents sparse signal, $\Phi \in R^{MxN}$ measurement matrix and $y \in R^M$ observation matrix. In measurement process, observation matrix is obtained by multiplying measurement matrix and sparse signal. In CS, measurement matrix is needed to be satisfied Restricted Isometry Property (RIP) and incoherence [5], [6]. These two conditions are met with high probability by choosing measurement matrix as random Gaussian matrix [6]. Last process is sparse signal reconstruction. The recovered signal is obtained from observation matrix by using sparse signal reconstruction algorithms. In general these algorithms use the knowledge of observation matrix, measurement matrix and sparsity level. Until this time, many sparse recovery algorithms have been introduced. Widely used sparse signal reconstruction algorithms are Basis Pursuit (BP), Matching Pursuit (MP), Orthogonal Matching Pursuit, (OMP), Compressive Sampling Matching Pursuit (CoSaMP) and Primal Dual Interior Point (PDIP) [7]. Although sparse recovery algorithms can be divided into many different class of algorithms, they can basically be classified into two categories as l_1 minimization and greedy algorithms [8]. These two classes of algorithms are compared in this study. These are OMP, CoSaMP algorithms from greedy algorithms and PDIP method from l_1 minimization. The comparison is especially made under noisy measurements

The rest of the paper is organized as follows: Second part of the study explains PDIP method, OMP and CoSaMP algorithms. Third part includes simulation results and finally, fourth part is conclusion.

2. SPARSE SIGNAL RECOVERY ALGORITHMS

2.1. PDIP method

The original CS problem is l_0 norm optimization. is shown in (2).

$$\min ||z||_0 \ subject\ to \quad \Phi z = \Phi x \qquad (2)$$

However l_0 norm optimization is NP-hard that need exhaustive enumeration. If measurement matrix satisfies RIP and incoherence properties, l_1 norm optimization can be used instead of l_0 norm optimization [3], [9].

$$\mathbf{\min ||z||_1 \ subject\ to \ \ \Phi z = \Phi x} \qquad (3)$$

L_1 norm problem is a convex problem and can be solved by using linear programming method. In this study PDIP algorithm is used as l_1 norm optimization method when comparing with greedy algorithms. Equation (3) can be expressed as linear problem as in (4) [8].

$$\boldsymbol{min_z\ 1_{2n}{}^T\ subject\ to\ A_0 z = y\ and\ z \geq 0} \qquad (4)$$

PDIP method is used to solve (4). PDIP method is basically following. Considering Kuhn tucker conditions at the minimal point $\boldsymbol{z}^*$ of the (4), there exist two vectors $\boldsymbol{v}^* \in \boldsymbol{R^M}$ and $\boldsymbol{\lambda}^* \in \boldsymbol{R^{2N}}$ such that

$$1_{2N}{}^T + A_0{}^T v^* - \lambda^* = 0, \qquad (5)$$

$$\lambda^*{}_i z^*{}_i = 0, i = 1,2, \ldots 2N, \qquad (6)$$

$$A_0\, z^* = y, \qquad (7)$$

$$z^* \geq 0, \quad \lambda^* \geq 0. \qquad (8)$$

The PDIP algorithm solves the Kuhn-Tucker conditions above (5)-(8), by the newton iteration method. [10]. This method is called interior point method because $z[k]$, $v[k]$, $\lambda[k]$ approximated vectors are kept in an interior point of the region defined by (8).

2.2 Greedy Algorithms

Greedy algorithms try to solve a problem in an iterative manner. In CS, greedy algorithms aim to solve (2) by updating the support set iteratively. For each iteration, the columns of measurement matrix that mostly correlated with observation matrix are chosen. After that, the contribution of the support set columns is subtracted from observation matrix. This process continues until correct set of columns are identified.

2.2.1 OMP Algorithm

OMP creates the solution by choosing a mostly correlated column vector of the measurement matrix and calculating new residue at each step. In other words, most strongly correlated column of measurement matrix with observation matrix is chosen firstly. Afterwards, its contribution to observation matrix is subtracted and the same procedure is applied to the residual iteratively. This algorithm finds the solution of sparse reconstruction problem after k number of iterations. The OMP algorithm is explained in detail as shown in Algorithm 1 [11].

Algorithm 1: OMP Algorithm.

Input:

- Φ: MxN-dimensional measurement matrix.
- k: Sparsity level
- y: $Mx1$-dimensional observation vector.

Output:

- x': Estimation of the original signal x.
- Λ_K: An index set of non-zero elements from {1,…,N}.
- Φ_K: Selected columns of measurement matrix.
- a_k: Approximated observation signal y
- $r_k = y - a_k$: Residual vector.

Initialization:

- $r_0 = y$
- $\Lambda_K = \emptyset$
- t=1
- $\Phi_0 = \emptyset$

Iterations:

1. Find the index λ_t which solves the following optimization problem. λ_t is the column of Φ which is mostly correlated with the r_{t-1}

$$\lambda_t = argmax_{j=1..N} |\langle r_{t-1}, \varphi_j \rangle| \qquad (9)$$

2. Augment the index set.

$$\Lambda_K = \Lambda_K \cup \{\lambda_t\} \qquad (10)$$

3. Augment the matrix of selected columns.

$$\Phi_t = [\Phi_{t-1} \quad \varphi_{\lambda_t}] \qquad (11)$$

4. Get a new signal estimation by solving below problem.

$$x_t = argmin_{x'} ||y - \Phi_t x'||_2 \qquad (12)$$

5. Calculate the new approximation of the observation signal.

$$a_t = \Phi_t x_t \qquad (13)$$

6. Calculate the new residual.

$$r_t = y - a_t \qquad (14)$$

7. If $t < k$, increment t and return to Step 1 for a new iteration.

At every new iteration (t), the r_t is orthogonal to columns of the matrix Φ_t. Also, when the residual of the previous iteration ($t-1$) is nonzero, a new column of measurement matrix is chosen and the matrix Φ_t has independent columns. For that reason, the signal estimation x_t, solution to the least squares problem in Step 4, is unique. At the end of the algorithm, the estimation of the signal x has nonzero entries at the components listed in the index set $\Lambda_t = \Lambda_K$, when t=k at last iteration.

2.2.2 CoSaMP Algorithm

Firstly, transpose of the measurement matrix and observation matrix are multiplied in this algorithm. After that, mostly correlated columns of measurement matrix are chosen twice as much as sparsity level and support set is created. And then, the number of elements of support set is decreased from 2k to k by using least squares method. Finally residue vector is calculated, and these iterations continue until residue vector fall below a specified error term. The result of least squares for last iteration is the reconstructed x signal. Detailed steps of CoSaMP algorithm are following in Algorithm 2 [8], [12].

Algorithm 2: CoSaMP Algorithm.

Inputs:

- Φ: MxN -dimensional measurement matrix
- k: Sparsity level of x signal.
- y: $Mx1$ -dimensional observation matrix.

Output:

- $x[t+1]$: Recovered x signal.

Initial Values:

- $x[0] = 0$
- $r[0] = y$, Residual vector.
- $\Lambda = \emptyset$, Support set.
- t=0, Number of iteration.

Algorithm Steps:

1- Choose the support set twice as much as sparsity level, so 2k.

$$\Lambda = \Lambda \cup supp(H_{2k}|\Phi^T \mathrm{r[k]}|) \quad (15)$$

2- Calculate z vector using least square approximation.

$$z = \Phi_\Lambda^\dagger r \quad (16)$$

$$\Phi_\Lambda^\dagger = (\Phi_\Lambda^T \Phi_\Lambda)^{-1} \Phi_\Lambda^T \quad (17)$$

3- Equate zero all z vectors, except biggest k pieces.

$$x[t+1] = H_k(z) \quad (18)$$

4- Update residual vector by using (19) and (20).

$$\Lambda = supp(x[t+1]) \quad (19)$$

$$r[t+1] = y - \Phi \mathrm{x[t+1]} \quad (20)$$

5- $t = t + 1$

6- Exit, if sufficient conditions are met.

Note: H_k represents thresholding operator. This operator equates all elements to zero except k number of elements.

3. SIMULATION RESULTS

In this study, sparse reconstruction performances of three algorithms, OMP, CoSaMP, PDIP, are compared . In this comparison, measurements include 20 dB white Gaussian noise. While performing these experiments, firstly a sparse signal is generated. And then, measurement matrix is generated in such a way that satisfy RIP and incoherence property. After generating measurement matrix, observation matrix is obtained by multiplying sparse signal and measurement matrix. Practical systems always have some noise. In CS, especially noise is occurred while measuring data. For that reason, 20 dB Signal to Noise Ratio (SNR) level noise is added to observation matrix/measurements. Finally, by using three mentioned sparse signal recovery algorithms, the sparse signal is tried to be recovered. And then, algorithms' recovery performances are compared for different test cases. These test cases include different sparsity level and different measurement numbers. The performance metrics are Mean Squared Error (MSE) between recovered signal and original signal, correlation between recovered signal and original signal, reconstruction time and iteration number.

3.1. Test Case 1

In this test case, length of the test signal is 512, number of measurements is 256. Sparse recovery algorithms are tested for different sparsity levels that 16, 32, 64 and 128. The performance metrics of the algorithms for this case are shown in Table I.

Table 1. The results of performance metrics for test case 1.

N=512 M=256	MSE	Corr.	Rec. time (s)	Number of iteration
k=16				
PDIP	1.100×10^{-3}	0.9867	0.1868	18
OMP	4.783×10^{-5}	0.9995	0.0278	16
CoSaMP	1.108×10^{-4}	0.9987	0.0210	6
k=32				
PDIP	2.500×10^{-3}	0.9844	0.1844	19
OMP	5.477×10^{-4}	0.9966	0.0429	32
CoSaMP	6.734×10^{-4}	0.9959	0.0812	12
k=64				
PDIP	5.500×10^{-3}	0.9775	0.2020	19
OMP	1.000×10^{-3}	0.9958	0.1273	64
CoSaMP	2.900×10^{-3}	0.9881	0.3800	21
k=128				
PDIP	4.510×10^{-2}	0.8949	0.2366	19
OMP	5.770×10^{-2}	0.8729	0.4952	128
CoSaMP	9.530×10^{-3}	0.7590	0.7744	20

From Table I, for k=16, OMP has better MSE and correlation values than other two. However, when reconstruction time is considered, CoSaMP is a bit better than OMP. For sparsity level k=32, OMP is the best when MSE, correlation and reconstruction time are considered. However, there is not too much difference between OMP and CoSaMP in terms of MSE and correlation values. When sparsity level is 64, has the best performance metrics. When sparsity level is 128, PDIP is better than other two. This can be explained in such a manner that these three algorithms can't recover a sparse signal even from noiseless measurements for N=512, M=256 and k=128. For that reason this case, k=128, may not reflect exact comparison.

3.2. Test Case 2

In this test case, length of the test signal is 512, number of measurements is 128. Sparse recovery algorithms are tested for different sparsity levels that 16, 32 and 64. The performance metrics of the algorithms for this case are shown in Table II.

Table 2. The results of performance metrics for test case 2.

N=512 M=128	MSE	Corr.	Rec. time (s)	Number of iteration
k=16				
PDIP	1.200×10^{-3}	0.9861	0.1379	20
OMP	2.043×10^{-4}	0.9976	0.0284	16
CoSaMP	6.511×10^{-4}	0.9922	0.0342	10
k=32				
PDIP	3.600×10^{-3}	0.9797	0.1251	21
OMP	1.500×10^{-3}	0.9910	0.0423	32
CoSaMP	6.000×10^{-3}	0.9635	0.1761	37
k=64				
PDIP	3.820×10^{-2}	0.8282	0.1235	18
OMP	1.015×10^{-1}	0.6266	0.1139	64
CoSaMP	7.030×10^{-2}	0.6462	0.4064	45

As shown from Table II, when sparsity level is 16, OMP has better performance than other two for all performance metrics. When sparsity level is 32, MSE and correlation values for all algorithms are getting worse w.r.t. previous sparsity level, k=16, because algorithms need to find 32 values instead of 16. Again in this case OMP is the best. Also, PDIP method is better than CoSaMP when sparsity level is 32. When sparsity level is increased to 64, PDIP has better performance metrics values than other two.

3.3. Test Case 3

In this test case, length of the test signal is 512, number of measurements is 64. Sparse recovery algorithms are tested for different sparsity levels that 16 and 32. The performance metrics of the algorithms for this case are shown in Table III.

As shown from Table III, when sparsity level is 16, OMP has better performance than other two for all performance metrics. When sparsity level is 32, the performance orders of the algorithms are following: PDIP, CoSaMP, OMP respectively. However reconstruction time is considered, OMP reconstruct the signal in shorter time.

Table 3. The results of performance metrics for test case 3.

N=512 M=64	MSE	Corr.	Rec. time (s)	Number of iteration
k=16				
PDIP	5.000×10^{-3}	0.9562	0.1175	21
OMP	4.307×10^{-4}	0.9949	0.0267	16
CoSaMP	1.000×10^{-4}	0.9892	0.0484	25
k=32				
PDIP	2.720×10^{-2}	0.8164	0.0987	16
OMP	8.370×10^{-2}	0.5583	0.0394	32
CoSaMP	5.470×10^{-2}	0.5729	0.0774	25

In summary, the algorithms are tested for different sparsity levels and different measurement numbers for proper comparison. It can be said from this study, greedy algorithms are better than PDIP for recovering sparse signal from noisy measurement in general. However, when sparsity level is close to measurement number, PDIP method has better performance metrics of sparse recovery. There exist some other studies that compare these algorithms without noise [13]. In mentioned study, CoSaMP has better performance results with noise free measurements.

4. CONCLUSION

To sum up, CS is a useful method for recovering sparse signal when compared with traditional sampling techniques. In literature, most CS related studies are focused on reconstruction algorithms. Three sparse signal reconstruction algorithms are compared in this study. OMP, CoSaMP and PDIP algorithms are chosen for comparison because they are widely used algorithms in literature. It is understood from this study that in general, OMP has better performance results for recovering sparse signal from noisy measurements.

REFERENCES

[1] D. L. Donoho, **"Compressed sensing,"** *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candès, J. Romberg, and T. Tao, **"Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,"** *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] E. J. Candès, **"The restricted isometry property and its implications for compressed sensing,"** *Comptes Rendus Math.*, vol. 346, no. 9–10, pp. 589–592, 2008.

[4] Y. Arjoune, N. Kaabouch, and H. El Ghazi, **"Compressive sensing: Performance comparison of sparse recovery algorithms,"** in *(Ccwc)*, 2017.

[5] E. J. Candes and M. B. Wakin, **“An Introduction To Compressive Sampling,”** *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[6] R. Baraniuk, **“Compressive Sensing [Lecture Notes],”** *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[7] S. Çelik, M. Başaran, S. Erküçük, and A. Çırpan, **"Seyrek işaret geri oluşturma için sıkıştırılmış algılama tabanlı algoritmaların karşılaştırılması".** *2016 24th Signal Processing and Communication Application Conference*, 16-19 May 2016, Zonguldak, Turkey. 2016

[8] K. Hayashi, M. Nagahara, and T. Tanaka, **“A User’s Guide to Compressed Sensing for Communications Systems,”** *IEICE Trans. Commun.*, Vol. E96.B, No. 3, pp. 685–712, 2013.

[9] E. J. Candes and T. Tao, **“Decoding by Linear Programming,”** 2005.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Vol. 25, No. 3. 2010.

[11] J. a Tropp and A. C. Gilbert, **“Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit,”** *IEEE Trans. Inf. Theory*, Vol. 53, No. 12, pp. 4655–4666, Dec. 2007.

[12] D. Needell and J. A. Tropp, **“CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,”** *Appl. Comput. Harmon. Anal.*, Vol. 26, No. 3, pp. 301–321, 2009.

[13] M. E. Erkoç, N., Karaboğa, **"Comparison of L_1 Minimization and Greedy Algorithms for Recovering Sparse Frequency Domain Signals",** *International Workshop, Mathematical Methods in Engineering, MME-2017*, Ankara, Turkey, 27-29 Apr. 2017, pp.67-67.