Recover Sparse Signals from Under-Sampled Observations

First formal report for ACSP Laboratory

Ilias Chrysovergis

MSc Communications & Signal Processing, EEE Department
Imperial College London
London, United Kingdom
ilias.chrysovergis17@imperial.ac.uk

***Abstract*—The aim of this lab experiment is to become familiar with sparse signal processing. Sparsity of a signal is a condition under which recovery is possible, when fewer samples of a signal than required by the Shannon-Nyquist sampling theorem are available. That sparse signal processing technique is known as compressed or compressed sensing. Firstly, we realize that the least squared method does not provide the optimal solution for recovering a signal from under-sampled observations. Secondly, we implement three different greedy algorithms - the OMP, SP and IHT algorithms- in Matlab to solve the sparse recovery problem. Finally, we compare the success rate of the three greedy algorithms for a variety of values for the cardinality of the support set *S*. We, also, generate 500 independent tests for each value of *S* to extract more reliable conclusions.**

Keywords*—compressed sensing, sparse signal processing, greedy algorithms, orthogonal matching pursuit (OMP), subspace pursuit (SP), iterative hardthresholding (IHT)

I. Introduction

This report aims to provide an introduction into compressed sensing and sparse signal processing. Compressed sensing (also known as compressive sensing, compressive sampling or sparse sampling) is a signal processing technique based on the principal that a signal can be recovered from far fewer samples than required by the Shannon-Nyquist sampling theorem [1].

Compressed sensing proposes a fundamental paradigm shift on how signals are stored and transmitted. According to Fig.1 the traditional approach is to sense (sample) a physical signal to obtain a digital representation of it (digital signal) and, afterwards, compress it to store or transmit it effectively [2].

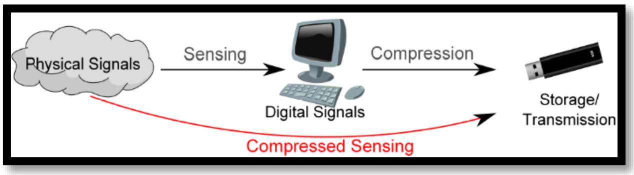

Figure 1: The paradigm shift: Compressed Sensing (Wei Dai, Imperial College London [2])

If someone investigates deeply that long-established approach, they can realize that the sensing system samples so many data that the overall system cannot handle with and, so it seems that during the sampling process redundant information is being measured. Therefore, why should someone sample so many data if they finally do not make use of? [3]

Compressed sensing does not separate the sampling from the compression process. It utilizes the conditions of sparsity and incoherence of the signals in some domain, to reconstruct a signal properly out of a far under-sampled observation of it [1] [2]. This report examines some basic greedy algorithms for sparse signal recovery. Those are the Orthogonal Matching Pursuit (OMP), the Subspace Pursuit (SP) and the Iterative Hardthresholding (IHT) algorithms.

The rest of the report is organized as follows. Section 2 provides the reader with the necessary background, definitions and algorithms used during the experiment. Section 3 presents the procedure that was followed as well as the results of each exercise of the experiment. Finally, section 4 concludes the report.

II. Background

A. *Linear Systems*

A linear system is depicted as follows:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the observation vector, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a given linear transform and $\boldsymbol{x} \in \mathbb{R}^n$ stands for an uknown data vector. It is obvious that when $m < n$ the solution of the linear system is not unique.

Definition: A vector has unit l_2-norm means that the square of the sum of the square of each element of the vector is equal to one.

Therefore, when we say that the columns of the matrix $\boldsymbol{A}$ have unit l_2-norm we mean that:

$$\left|\left|\boldsymbol{A}_{i,:}\right|\right|_2 = \left(\sum_{j=1}^{m} \boldsymbol{A}_{j,i}^2\right)^2 = 1.$$

Definition: An under-determined linear system is a system where $m < n$. Therefore, there are more variables than equations, and thus, many different choices of $\boldsymbol{x}$ may lead to the same $\boldsymbol{y}$. In compressive sensing, by allowing a small number of nonzero coefficients, there may be a unique sparse solution to this system [1] [2]. This statement is going to be examined in this report.

B. *Sparse Vectors*

A vector $\boldsymbol{x} \in \mathbb{R}^n$ is said to be S - sparse if there are S many nonzero entries in $\boldsymbol{x}$.

Definition: The **support set** of $\boldsymbol{x}$ **is:**

$$S = \{i : x_i \neq 0\}$$

Definition: The **sparsity** of $\boldsymbol{x}$ is given by the cardinality (size) of the support set S $= |S|$

Definition: A signal $\boldsymbol{x}$ is called sparse if $S \ll n$.

It has been observed that for most natural signals a domain, where they are approximately sparse, can be found. For example, an audio signal is typically sparse in the time-frequency domain, whereas an image is typically sparse under wavelet or discrete cosine transform (DCT) [2].

C. *Definitions*

Truncation: Let $S \subset \{1,2,\dots,n\}$ be an index set whose cardinality is S. Let $S^c \equiv \{1,2,\dots n\} \setminus S$ be its complement. For any given vector $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x}_{\boldsymbol{S}} \in \mathbb{R}^S$ denotes the subvector of $\boldsymbol{x}$ indexed by S and $x_{S^c} \in \mathbb{R}^{n-S}$ is the subvector of $\boldsymbol{x}$ indexed by S^c. Similarly, for any given matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{A}_{\boldsymbol{S}} \in \mathbb{R}^{m\times S}$ gives the submatrix of $\boldsymbol{A}$ formed by the columns indexed by S and $\boldsymbol{A}_{\boldsymbol{S}^c} \in \mathbb{R}^{m\times(n-S)}$ represents the submatrix of $\boldsymbol{A}$ consisting by the columns indexed by S^c.

Projection & Residue: Let $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m\times d}$ where $d < m$. Suppose that $\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is invertible. The projection of $\boldsymbol{y}$ onto the subspace spanned by $\boldsymbol{A}$ is defined as

$$\boldsymbol{y}_{\boldsymbol{p}} = proj(\boldsymbol{y},\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{y}$$

where $\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A})^{-\boldsymbol{1}}\boldsymbol{A}^{\boldsymbol{T}}$ denotes the pseudo-inverse of the matrix A. The projection residue is given by

$$\boldsymbol{y}_{\boldsymbol{r}} = resid(\boldsymbol{y},\boldsymbol{A}) = \boldsymbol{y} - \boldsymbol{y}_{\boldsymbol{p}}.$$

Hard Thresholding Function: For given vector $\boldsymbol{x} \in \mathbb{R}^n$ and positive integer $S < n$, the hard thresholding function $H_S(\boldsymbol{x})$ set all but the largest S entries (in magnitude) of $\boldsymbol{x}$ to zero.

D. *Greedy Algorithms*

The three greedy algorithms are presented below:

a) *The Orthogonal Matching Pursuit*

```
Algorithm 1: The Orthogonal Matching Pursuit (OMP)
Input: $S, \boldsymbol{A}, \boldsymbol{y}$.
Output: $\boldsymbol{x}$.
Initialization:
        $\hat{\boldsymbol{x}} = \boldsymbol{0}, S = \emptyset \; and \; \boldsymbol{y_r} = \boldsymbol{y}$
Iteration: $l = 1,2, \dots , S$
    1. $S = S \cup supp\big(H_1(\boldsymbol{A^T y_r})\big)$.
    2. $\boldsymbol{x_s} = \boldsymbol{A_s^{\dagger} y} \; and \; \boldsymbol{x_{S^c}} = \boldsymbol{0}$.
    3. $\boldsymbol{y_r} = \boldsymbol{y} - \boldsymbol{A\hat{x}}$.
```

The Orthogonal Matching Pursuit Algorithm is the easiest for someone to understand. In each iteration the support set is increased by one index value, until the number of iterations is equal to the sparsity of the signal. This index value corresponds to the maximum value of the vector $\boldsymbol{A^T y_r}$. Eventually, the support set will be composed by the indexes of the maximum value of the inner product of the projection residue and the system matrix. [4]

b) *The Subspace Pursuit*

```
Algorithm 2: The Subspace Pursuit (SP)
Input: $S, \boldsymbol{A}, \boldsymbol{y}$.
Output: $\hat{\boldsymbol{x}}$.
Initialization:
    1. $S = supp\big(H_S(\boldsymbol{A^T y})\big)$.
    2. $\boldsymbol{y_r} = resid(\boldsymbol{y}, \boldsymbol{A_S})$.
Iteration: $l = 1,2, \dots$ until the exit criteria are true
    1. $\tilde{S} = S \cup supp\big(H_S(\boldsymbol{A^T y_r})\big)$.
    2. $\boldsymbol{b_{\tilde{S}}} = \boldsymbol{A_{\tilde{S}}^{\dagger} y} \; and \; \boldsymbol{b_{\tilde{S}^c}} = \boldsymbol{0}$.
    3. $S = supp\big(H_S(\boldsymbol{b})\big)$.
    4. $\hat{\boldsymbol{x}} = \boldsymbol{A_S^{\dagger} y} \; and \; \hat{\boldsymbol{x}}_{\boldsymbol{S^c}} = \boldsymbol{0}$.
    5. $\boldsymbol{y_r} = \boldsymbol{y} - \boldsymbol{A\hat{x}}$.
```

The Subspace Pursuit Algorithm is the more complicated algorithm for someone to understand, but has the best results as investigated in the III.C section. In this algorithm two different exit criteria may be used. The first one is to provide a preset tolerance for the normalized error $\frac{||\boldsymbol{y}-\boldsymbol{A\hat{x}}||_2}{||\boldsymbol{y}||_2}$. In this experiment this exit criterion was used and the tolerance was set equal to 10^{-6}. The second exit criterion is to stop when the algorithm starts to diverge, i.e., when the error in the current iteration is larger than the error in the previous iteration. This second criterion may be better when there is no indication about what value the preset tolerance should be set to. The chosen exit criterion must be combined with a second criterion, which will enforce the iterations to finish if they have exceeded a certain threshold. For this experiment, the threshold used is equal to the sparsity S of the signal, because we did not want to exceed the number of iterations used in the Orthogonal Matching Pursuit algorithm. In this algorithm, the support set is increased by S indexes at the beginning of each iteration. Therefore, the new support set $\tilde{S}$ is composed by $2 * S$ indexes. To form the new support set, the S largest entries of the $\boldsymbol{A_S^{\dagger} y}$ matrix are used. [5]

c) *The Iterative Hardthresholding*

```
Algorithm 3: The Iterative Hardthresholding (IHT)
Input: $S, \boldsymbol{A}, \boldsymbol{y}$.
Output: $\hat{\boldsymbol{x}}$.
Initialization:
        $\hat{\boldsymbol{x}} = \boldsymbol{0}$
Iteration: $l = 1,2, \dots$ until the exit criteria are true
        $\hat{\boldsymbol{x}} = H_S\big(\hat{\boldsymbol{x}} + \boldsymbol{A^T}(\boldsymbol{y} - \boldsymbol{A\hat{x}})\big)$.
```

The Iterative Hardthresholding algorithm is the easiest for someone to implement, as it is only one line in general. The exit criteria are the same as the ones being used in the Subspace Pursuit algorithm. For this experiment the first exit criterion was used, and the preset tolerance was set equal to 10^{-6}. The complementary exit criterion, in case that the first is not met, is examined in the section III.B more thoroughly, because setting it equal to S did not lead to very good results. [6]

III. Procedure & Results

A. *Least Squared Solution*

In the first exercise a random Gaussian matrix $\boldsymbol{A}$ with $m = 128$ and $n = 256$ is generated with unit l_2-norm. Since $m < n$ the system is an underdetermined linear system. Furthermore, the input vector $\boldsymbol{x}$ is a Gaussian random vector. Therefore, knowing the output vector $\boldsymbol{y}$ and assuming that the matrix $\boldsymbol{A}$ is known, an estimation of the initial signal is calculated through the following two equations in Matlab:

1. $\hat{\boldsymbol{x}}_{\boldsymbol{1}} = pinv(\boldsymbol{A}) * \boldsymbol{y}$
2. $\hat{\boldsymbol{x}}_{\boldsymbol{2}} = \boldsymbol{A} \backslash \boldsymbol{y}$

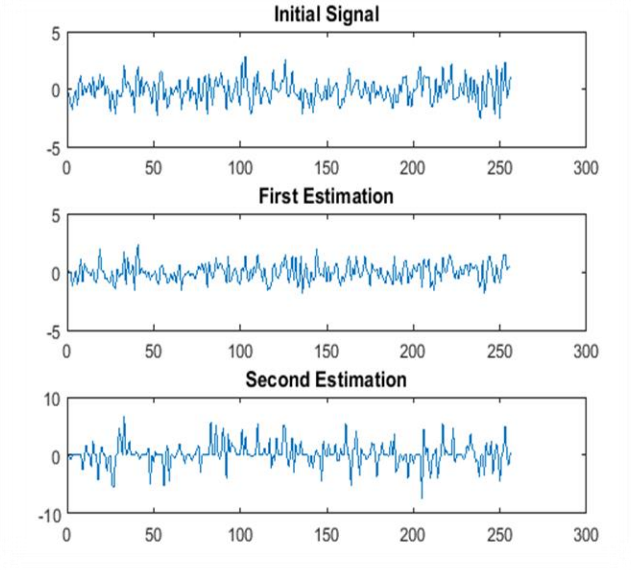

Figure 2: Estimating the Initial Signal

These two solutions use the so called least squared method. The normalized error for the two different estimations is given in the following table.

TABLE I. The Normalized Error of the two Different Estimations

First Estimation	0.6722
Second Estimation	2.0891

Also, by visualizing the normalized error, it is easy for someone to realize that this method is not very good for an underdetermined system. The two errors are presented below:

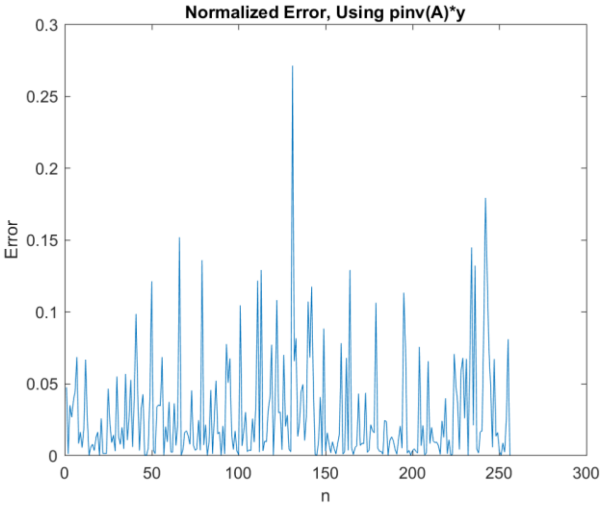

Figure 3:The normalized error by using the first equation

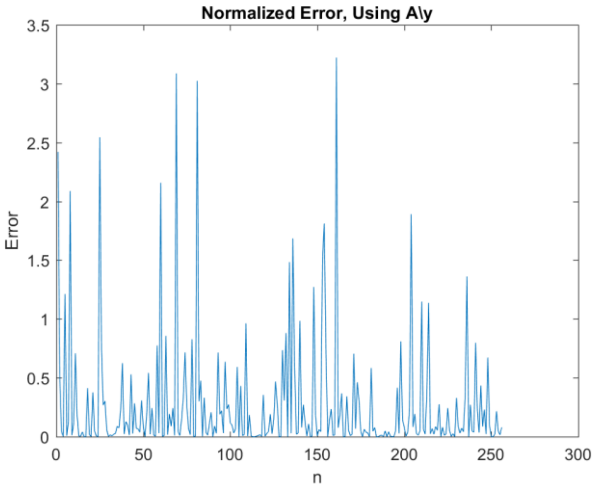

Figure 4:The normalized error by using the second equation

Therefore, for underdetermined systems the least squared method is not able to give the right solution to our problem. This is the reason why sparsity is assumed, since it can lead to the optimal solution for the problem of signal reconstruction. In the next section the sparse solution is examined and the results of three greedy algorithms are presented and discussed.

B. *Sparse Solution*

In the second exercise an S-sparse signal is generated, where $S = 12$. Firstly, a normally distributed signal $\boldsymbol{x}$ of zero mean and standard deviation equal to unity of length $n = 256$ is generated. Then, 12 random samples are selected from the set $\{1,2, \ldots 256\}$ as indexes, to create the S-sparse signal from the generated normally distributed signal. All the other values of $\boldsymbol{x}$ are converted into zeros. The matrix $\boldsymbol{A}$, representing the underdetermined system, is generated as previously and the output of the system $\boldsymbol{y}$ is created.

Afterwards, the three algorithms described at II.D are used to estimate the sparse signal $\boldsymbol{x}$. The implementation of the OMP algorithm is not a difficult task since a for-loop is used for the iteration and there is no need of providing an exit criterion. Furthermore, in each iteration one index is added to the support set, so the implementation is achieved without great effort too. The SP algorithm is a little bit trickier since the support set is constructed by a different way. In addition to this, a while loop is used and the exit criterion includes two different factors. The first one is the normalized error to be less than 10^{-6}, while the second one is to have less than $l = S = 12$ iterations. This criterion is used because the SP algorithm requires less than S iterations to reconstruct the signal most of the times [2]. Therefore, for the cases that the signal cannot be reconstructed, more than S iterations would increase the computations and this would lead to larger complexity. Finally, for the IHT algorithm the same exit criterion is used, but now a maximum of $10 * S$ iterations are allowed since the algorithm did not perform very well with less iterations. The implementation for this algorithm does not require great effort either.

In the following figures the estimations of the three algorithms and the ground truth signal $\boldsymbol{x}$ are presented and compared.

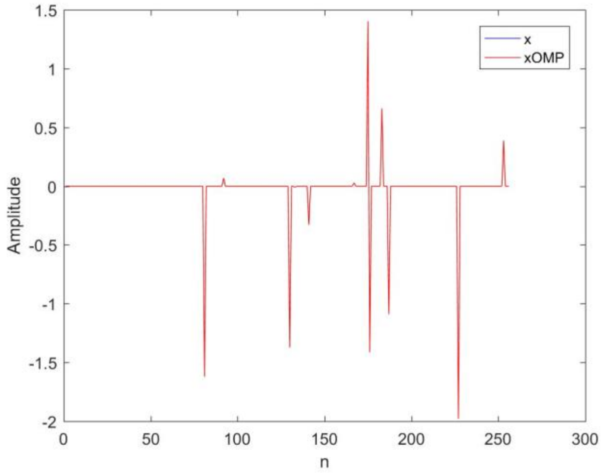

Figure 5:Signal Reconstruction using the OMP algorithm

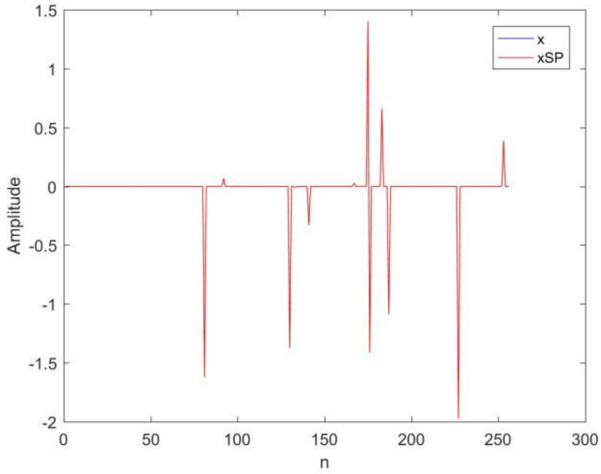

Figure 6: Signal Reconstruction using the SP algorithm

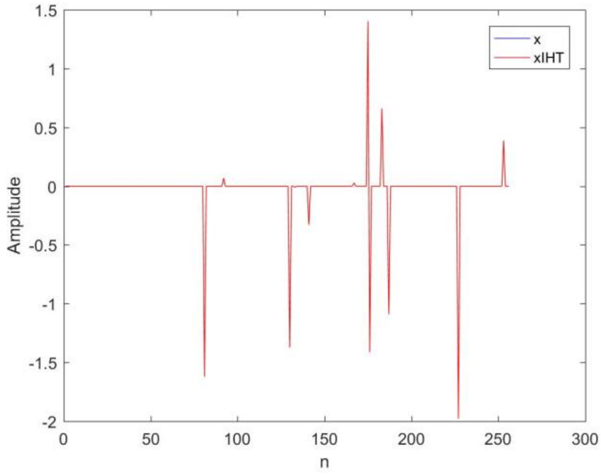

Figure 7: Signal Reconstruction using the IHT algorithm

It is obvious that the reconstructed signal is identical to the initial signal $\boldsymbol{x}$, since the red line covers entirely the blue one for all the three greedy algorithms. The normalized errors for the three algorithms are given in the following table:

TABLE II. THE RECONSTRUCTION NORMALIZED ERROR OF THE THREE GREEDY ALGORITHMS

OMP Error	$2.3026 * 10^{-16}$
SP Error	$2.3026 * 10^{-16}$
IHT Error	$6.8787 * 10^{-5}$

The reconstruction error of the IHT algorithm is much greater than the other two errors. This is one more fact that shows us that the first two algorithms have better performance than the third one. In the following table the normalized error of the reconstruction of the IHT algorithm for different number of maximum iterations is presented. It shows us that the maximum number of iterations must be much greater than S to achieve low reconstruction error. Furthermore, it is obvious that the IHT algorithm can not reach the performance of the other two algorithms.

TABLE III. THE RECONSTRUCTION NORMALIZED ERROR OF THE IHT ALGORITHM FOR DIFFERENT NUMBER OF MAXIMUM ITERATIONS

Maximum Iterations	**Normalized Error**
$1 * S$	0.3836
$2 * S$	0.3030
$3 * S$	0.0714
$4 * S$	0.0358
$5 * S$	0.0171
$6 * S$	0.0104
$7 * S$	0.0083
$8 * S$	0.0014
$9 * S$	$4.3051 * 10\hat{}(-4)$
$10 * S$	$2.4215 * 10\hat{}(-4)$

C. Succes Rate Comparison

In the third exercise, the three greedy algorithms are compared using the success rate of recovering the ground truth signal. The success rate is regarded as the normalized error to be less than 10^{-6}. The sparsity S of the signal is varied from 3 to 63 with a step size equal to 3 and 500 independent tests, where both $\boldsymbol{A}$ and $\boldsymbol{x}$ are generated independently and as mentioned in the previous exercises.

In the following figure the three greedy algorithms are compared.

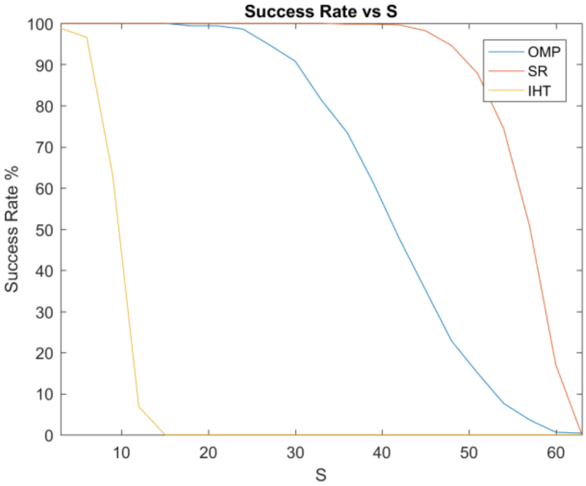

Figure 8:The Success Rate of the three greedy algorithms for different sparsity levels

It can be observed that the SP algorithm is the best algorithm since it has the best success rate no matter the value of S. Additionally, until S becomes equal to 20 the success rate of the OMP algorithm is 100%. This means that when the sparsity of the signal is less or equal to 20, a number that corresponds to almost 10% of the $dimension\ n$, both the first two algorithms give great reconstruction with error almost equal to zero. Afterwards, the success rate of the OMP algorithm is decreasing almost linearly into zero when $S = 60$. The SP algorithm gives great reconstruction (success rate equal to 100%) until the sparsity becomes equal to 45, almost two times better than the OMP algorithm. Then it decreases more rapidly into zero when $S = 63$. Finally, the IHT algorithm is the worst algorithm among the three ones, since it can reconstruct the signal effectively only for very low sparsity levels.

IV. Conclusion

Concludingly, this report investigates the significance of greedy algorithms for sparse signal processing. Firstly, the least squared solution for an underdetermined system is examined and the results show that the reconstruction error is very high. Then, the implications of considering sparse signals are being researched. Three greedy algorithms are used to reconstruct the initial sparse signal from an underdetermined signal. These are the Orthogonal Matching Pursuit (OMP), the Subspace Pursuit (SP) and the Iterative Hardthresholding (IHT) algorithms. The results are very encouraging, since the algorithms achieve very good reconstruction with very low error, especially the OMP and SP algorithms. Finally, the success rate of the three different algorithms is compared and the results show that the SP algorithm performs better and can achieve lower errors for high levels of sparsity than the other algorithms, while the IHT algorithm has the worst performance.

Acknowledgment

The author wishes to thank Dr. Wei Dai, who was the supervisor for the experiment and has taught him the principles of Compressive Sensing and Sparse Signal Recovery in the course “Topics in Large Dimensional Data Processing”, as well as Mr. Yang Lu, demonstrator of the experiment, for his valuable help and comments on his work.

V. References

[1] Wikipedia, Compressed Sensing, [Online]. Available: https://en.wikipedia.org/wiki/Compressed_sensing. [Accessed 16 11 2017].

[2] W. Dai, Topics in Large Dimensional Data Processing (Course Material), London, 2017.

[3] R. G. Baraniuk, Compressive Sensing (Video Lecture), YouTube, [Online]. Available: https://www.youtube.com/watch?v=zytez36XlCU. [Accessed 16 11 2017].

[4] Y. C. Pati, R. Rezaiifar and P. S. Krishnaprasad, *"Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet"*, 1993.

[5] W. Dai and O. Milenkovic, *"Subspace Pursuit for Compressive Sensing Signal Reconstruction"*, 2009.

[6] T. Blumensath and M. E. Davies, *"Iterative Hard Thresholding for Compressed Sensing"*.