A COMPARATIVE STUDY OF SOME GREEDY PURSUIT ALGORITHMS FOR SPARSE APPROXIMATION

Gagan Rath[1] *and Arabinda Sahoo*[2]

[1]INRIA, Centre Rennes - Bretagne Atlantique
Campus Universitaire de Beaulieu
35042 Rennes, France

[2]Dept. of Electronics and Telecommunication Engg.
Silicon Institute of Technology
Bhubaneswar, India

ABSTRACT

Solving an under-determined system of equations for the sparsest solution has attracted considerable attention in recent years. Among the two well known approaches, the greedy algorithms like matching pursuits (MP) are simpler to implement and can produce satisfactory results under certain conditions. In this paper, we compare several greedy algorithms in terms of the sparsity of the solution vector and the approximation accuracy. We present two new greedy algorithms based on the recently proposed complementary matching pursuit (CMP) and the sensing dictionary framework, and compare them with the classical MP, CMP, and the sensing dictionary approach. It is shown that in the noise-free case, the complementary matching pursuit algorithm performs the best among these algorithms.

1. INTRODUCTION

In many areas of signal processing, such as compression, denoising, compressed sensing, error decoding, etc., one often encounters the following problem: Given an underdetermined system of equations

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where the matrix A is of dimension $K \times N$, $K < N$, what is the solution vector $\mathbf{x}$ that has the minimum number of non-zero elements? In other words, how can we represent the given vector $\mathbf{b}$ as a linear sum of the fewest columns of A? The solution to this problem is important for various reasons. For example, in the case of compression, the vector $\mathbf{b}$ represents a signal to be compressed and the columns of A represent a redundant set of basis signals. The solution to the above problem yields the fewest coefficients to represent the signal under consideration thus leading to maximal compression efficiency.

Representing a given signal as a linear sum of a few signals from a signal set is commonly known as sparse approximation in the signal processing literature. The signals which represent the columns of the matrix A are commonly called atoms and the set of atoms is called a dictionary. Since the signal set is larger than necessary (i.e., $N > K$) and the signals span $\mathbb{R}^K$, the signals in the dictionary are said to form a redundant basis. The sparse representation problem thus can be posed as

$$\min\{\|\mathbf{x}\|_0 \ : \ A\mathbf{x} = \mathbf{b}\}, \tag{2}$$

where the L_0 norm denotes the number of non-zero elements. The sparse approximation problem, which allows some approximation error, can be posed as:

$$\min\{\|\mathbf{x}\|_0 \ : \ \|A\mathbf{x} - \mathbf{b}\|_p \leq \delta\}, \tag{3}$$

for some $\delta > 0$. The norm p is usually 2, but could be 1 or ∞ as well.

The exact solutions to above problems can be found through combinatorial optimization methods. Since such methods require high computational complexity, most of the research activity in this area is centered on finding approximate solutions with tractable complexity. There are basically two approaches to arrive at a sub-optimal solution. One is a greedy approach which approximates the signal vector through a sequence of incremental approximations by selecting atoms suitably. Such approaches are known as matching pursuits (MP) [1, 2]. The other is known as the basis pursuit (BP), which relaxes the L_0 norm condition by L_1 norm and solves the problem through linear programming [3, 4, 5]. BP algorithms can produce more accurate solutions than the matching pursuit algorithms but require higher complexity. Recently, some other iterative algorithms such as the regularized orthogonal matching pursuit (ROMP) [6], the compressive sampling matching pursuit (CoSaMP) [7], and the subspace pursuit (SP) [8] have been proposed. These algorithms aim to provide the same guarantee as the BP but with computational complexity akin to orthogonal matching pursuit (OMP) [2]. Gradient pursuit [9] is similar to the matching pursuit but updates the sparse solution vector at each iteration with a directional update computed based on the gradient or the conjugate gradient. Recently Rath and Guillemot [10] proposed the complementary matching pursuit (CMP) which is similar to the matching pursuits, but is performed in the coefficient space rather than the signal space. Through simulation results, they showed that by performing approximations in the coefficient space, the convergence speed and the sparsity of resulting vectors are improved. The atom selection in the CMP can be viewed through the sensing dictionary framework of Schnass and Vandergheynst [11].

In this paper, we present two matching pursuit algorithms which are derived from the CMP. In one algorithm, we select atoms as done in the CMP, but the update step is similar to MP. In the other, we select atoms as in MP, but the update step is similar to CMP. We compare their performances with those of MP and CMP. We also compare these results with the sensing dictionary approach of Schnass and Vandergheynst [11] by using the pseudo-inverse of the original dictionary as the sensing dictionary.

2. MATCHING PURSUIT (MP)

Matching pursuit [1] is an iterative greedy algorithm that searches for the sparse representation of a signal through a sequence of mono-atomic approximations. Each iteration of the algorithm consists of two steps: an atom selection step and a residual update step. The atom selection step finds the atom which has the highest correlation with the current residual error, where the correlation is measured as the length of the orthogonal projection. The update step updates the residual error by subtracting the correlated part from it.

Let $\mathbf{a}_i$, $1 \leq i \leq N$, denote the ith column of the dictionary matrix A. Assume that all atoms are normalized with unity magnitude, i.e., $\|\mathbf{a}_i\|_2 = 1$, $\forall i$. At the jth iteration, the algorithm finds

$$\boldsymbol{\alpha}_j = \underset{\{\mathbf{a}_i\}_{1\leq i\leq N}}{\arg\max} \ | < \mathbf{r}_{j-1}, \mathbf{a}_i > |, \qquad \text{[MP: Atom selection]} \tag{4}$$

where $< . >$ denotes the inner-product operation, and $\mathbf{r}_{j-1}$ denotes the residual at the $(j-1)$th iteration with $\mathbf{r}_0 = \mathbf{b}$. The inner product

$< \mathbf{r}_{j-1}, \boldsymbol{\alpha}_j >$ represents the coefficient associated with the selected atom $\boldsymbol{\alpha}_j$. Let us denote it as c_j. The algorithm then updates the residual as

$$\mathbf{r}_j = \mathbf{r}_{j-1} - c_j \boldsymbol{\alpha}_j. \qquad \text{[MP: Residual update]} \tag{5}$$

The approximation at the jth iteration is given as $\mathbf{b}_j = \sum_{k=1}^{j} c_k \boldsymbol{\alpha}_k$.

The algorithm terminates when a halting criterion is satisfied, such as when the norm of the residual falls below a desired approximation error bound, or when the number of distinct atoms in the approximation equals a desired limit.

The matching pursuit algorithm is very simple. But because of the sub-optimality [2], it suffers from slow convergence and poor sparsity result. The orthogonal matching pursuit (OMP) [2, 12] removes this drawback by projecting the signal vector to the subspace spanned by the selected atoms. The atom selection method in OMP remains the same as in MP. Because of the orthogonalization, once an atom is selected, it is never selected again in subsequent iterations. Let A_{j-1} denote the matrix of atoms selected up to the $(j-1)$th iteration. Thus, $A_j \equiv [A_{j-1} \;\; \boldsymbol{\alpha}_j]$, where $\boldsymbol{\alpha}_j$ denotes the atom selected in the jth iteration as in Eqn. 4. The approximation at the jth iteration is given as

$$\mathbf{b}_j = A_j A_j^{\dagger} \mathbf{b} \equiv A_j \mathbf{c}_j, \tag{6}$$

where $\mathbf{c}_j \equiv A_j^{\dagger}\mathbf{b}$ denotes the coefficient vector at the jth iteration, and $A_j^{\dagger}$ denotes the pseudo-inverse of A_j.

In the second step, the algorithm updates the residual as

$$\mathbf{r}_j = \mathbf{b} - A_j \mathbf{c}_j. \qquad \text{[OMP: Residual update]} \tag{7}$$

The algorithm terminates when the halting condition is satisfied.

3. COMPLEMENTARY MATCHING PURSUITS

In [10], Rath and Guillemot have proposed the complementary matching pursuit algorithm. The algorithm is similar to the MP but is performed in the row-space of the dictionary matrix. Consider the original system of equations: $A\mathbf{x} = \mathbf{b}$. Since the atoms make a redundant basis for the K-dimensional vector space $\mathbb{R}^K$, AA^T is an invertible matrix. As a consequence, the exact sparse solution to this system is also an exact sparse solution to the following system and vice versa;

$$A^T(AA^T)^{-1}A\mathbf{x} = \mathbf{x}_2, \tag{8}$$

where $\mathbf{x}_2 \equiv A^T(AA^T)^{-1}\mathbf{b}$ denotes the minimum L_2 norm solution of the original system of equations. Let $\boldsymbol{\phi}_i \equiv A^T(AA^T)^{-1}\mathbf{a}_i$, $1 \le i \le N$, denote the transformed set of atoms. The transformed atoms lie in the row-space of A. If we define a new dictionary matrix Φ whose ith column is $\boldsymbol{\phi}_i$, then the above system of equations can be rewritten as

$$\Phi\mathbf{x} = \mathbf{x}_2. \tag{9}$$

The CMP algorithm proposed in [10] is equivalent to the MP on this modified system of equations. Similarly, the orthogonal complementary matching pursuit (OCMP) is equivalent to the OMP applied to this modified system of equations. In the modified system, the atoms are no longer normalized. The projection of $\mathbf{x}_2$ on the atom $\boldsymbol{\phi}_k$ is given as

$$y_k = \frac{< \boldsymbol{\phi}_k, \mathbf{x}_2 >}{\|\boldsymbol{\phi}_k\|} = \frac{\mathbf{a}_k^T(AA^T)^{-1}\mathbf{b}}{\sqrt{\mathbf{a}_k^T(AA^T)^{-1}\mathbf{a}_k}}. \tag{10}$$

Thus the weight with respect to $\boldsymbol{\phi}_k$ can be expressed as

$$c_k = \frac{\boldsymbol{\phi}_k^T \mathbf{x}_2}{\|\boldsymbol{\phi}_k\|^2} = \frac{y_k}{\sqrt{\mathbf{a}_k^T(AA^T)^{-1}\mathbf{a}_k}}. \tag{11}$$

The pseudo-codes for CMP and OCMP algorithms are given in **Algorithm 1**. It is to note that, since the weights are obtained from the projection on the new atoms $\boldsymbol{\phi}_i$'s, but not on the original atoms $\mathbf{a}_i$'s, the residual error at any iteration in CMP may not be orthogonal to the immediately selected atom. Similarly, the residual error at any iteration in OCMP may not be orthogonal to all the atoms selected till that iteration.

Algorithm 1

CMP Algorithm:

Input: Dictionary matrix A with atoms $\mathbf{a}_i, i = 1, 2, \ldots, N$, and the signal vector $\mathbf{b}$

Output: Estimation of the solution vector $\mathbf{x}$

Initialization: $\mathbf{r} = \mathbf{b}$, $\mathbf{x} = \mathbf{0}_N$, Δ = diagonal matrix whose ith diagonal entry is equal to $\delta_{ii} = 1/\sqrt{\mathbf{a}_i^T(AA^T)^{-1}\mathbf{a}_i}$.

Repeat

Selection:

$$\begin{aligned} \mathbf{y} &= \Delta A^{\dagger}\mathbf{r} \\ k &= \max_i |\mathbf{y}[i]| \end{aligned}$$

Update:

$$\begin{aligned} c &= \delta_{kk}\mathbf{y}[k] \\ \mathbf{r} &= \mathbf{r} - c\mathbf{a}_k \\ \mathbf{x}[k] &= \mathbf{x}[k] + c \end{aligned}$$

until halting criterion is *true*.

OCMP Algorithm:

Input: Dictionary matrix A with atoms $\mathbf{a}_i$, $i = 1, 2, \ldots, N$, and the signal vector $\mathbf{b}$

Output: Estimation of the solution vector $\mathbf{x}$

Initialization: $\Phi = A^{\dagger}A$, $\mathbf{x}_2 = A^{\dagger}\mathbf{b}$, $\mathbf{r} = \mathbf{b}$, $\mathbf{x} = \mathbf{0}_N$, Δ =diagonal matrix whose ith diagonal entry is equal to $\delta_{ii} = 1/\sqrt{\mathbf{a}_i^T(AA^T)^{-1}\mathbf{a}_i}$, $I = \{\}$.

Repeat

Selection:

$$\begin{aligned} \mathbf{y} &= \Delta A^{\dagger}\mathbf{r} \\ k &= \max_i |\mathbf{y}[i]|; \quad I = I \cup \{k\} \end{aligned}$$

Update:

$$\begin{aligned} \mathbf{c} &= \Phi_I^{\dagger}\mathbf{x}_2 \\ \mathbf{r} &= \mathbf{b} - A_I\mathbf{c} \\ \mathbf{x}[I] &= \mathbf{c} \end{aligned}$$

until halting criterion is *true*.

In [11], the authors have mentioned the use of pseudo-inverse matrix $A^{\dagger}$ as a sensing dictionary. Following their method, in the jth iteration, the atom will be selected as:

$$\mathbf{y} = A^{\dagger}\mathbf{r}_{j-1}. \tag{12}$$

$$k = \arg\max_{1 \le i \le N} |y_i|, \quad \boldsymbol{\alpha}_j = \mathbf{a}_k \qquad \text{[SD: Atom selection]} \tag{13}$$

The update step is the same as in the OMP. Following this sensing dictionary framework, the atom selection method of CMP or OCMP can be interpreted as the application of a sensing dictionary where the sensing dictionary is the weighted pseudo-inverse matrix.

4. NEW GREEDY PURSUIT ALGORITHMS

Recall that pursuit algorithms consist of two steps: atom selection and residual update. These two steps are performed sequentially and thus are dependent. The update step is dependent on the selected atom. Similarly, the selected atom at any iteration is dependent on the residual error resulting from the previous iteration. The the final approximation depends on the joint effects of these two steps.

In MP and OMP, the atom selection and residual update at any iteration are based on the minimization of the residual error. However, this optimization procedure is only based on intuition. Except for the special case where the observation is a scalar multiple of one of the atoms, there is no 100% guarantee that either of these two methods will result in the sparsest solution. The authors of CMP [10] have shown that minimizing the error in the row-space rather than the signal space can improve the sparsity and decrease the approximation error. The row-space optimization results in a selected atom which may not be the closest to the current residual error. Furthermore, the updated residual error may not be orthogonal to the selected atom. The sensing dictionary approach [11] also shows that we can select an atom which is not closest to the current residual error, however, it can lead to sparser approximation, or less approximation error, than the OMP.

Based on the atom selection and the update procedure of the CMP algorithm, we present in the following two new greedy pursuit algorithms. In the first algorithm, we use the CMP atom selection criterion, but update the residual as in MP. In the other, we do the reverse, that is, we use the MP atom selection criterion, but update the residual as in the CMP. We also present the orthogonal extensions of these algorithms. Note that, for practical use, we will consider only the orthogonal versions. The presentation of the basic versions here is only for pedagogic completeness. We discuss the convergence of the algorithms in the sequel.

4.1 CMP(S)-MP/OMP(U): CMP Select, MP/OMP Update

In this algorithm, we select the atom in each iteration using the atom selection criterion in CMP. But we update the residual error as in MP, that is, by subtracting the orthogonal projection on the selected atom from it. In the orthogonal extension, we keep the atom selection process same, but replace the update process by the update process in the OMP. The pseudo-codes for the algorithms are given in **Algorithm 2**. The orthogonal version falls in the sensing dictionary framework of Schnass and Vandergheynst [11] with the weighted pseudo-inverse matrix as the sensing dictionary.

Observe that in the basic version, as in MP, the residual update step makes the new residual orthogonal to the selected atom. However, the atom selection criterion being different, there is a possibility of the same atom being selected in the next iteration. In such a case, the algorithm will enter into an infinite loop. In order that such a situation is avoided, the atom selection should only include the atoms which are not orthogonal to the new residual. Similar comments apply to the orthogonal extension of the algorithm.

4.2 MP(S)-CMP/OCMP(U): MP Select, CMP/OCMP Update

In this algorithm, we select the atom in each iteration as in MP, that is, by finding the closest atom to the current residual error. But we update the residual error as in CMP. In the orthogonal extension, we keep the atom selection process same, but replace the update process by the update process in the OCMP. The pseudo-codes for the algorithms are given in **Algorithm 3**.

It is to note that, in the basic version, since the residual update is the same as in CMP, the updated residual error may not be orthogonal to the immediately selected atom. However, since the atom selection step selects the closet atom, there is a possibility of the same atom being selected again in the next iteration. If this situation arises, the algorithm will enter into an infinite loop. To prevent this situation from happening, we need to apply a similar condition as in the previous algorithm. Since the update follows CMP, we need to derive the condition from the modified system of equations in Eqn. 9. Corresponding to atom $\mathbf{a}_i$, there is a modified atom $\boldsymbol{\phi}_i$. If in the jth iteration $\mathbf{a}_i$ is selected, then the complementary residual error, which is equal to $A^\dagger \mathbf{r}_j$, is orthogonal to $\boldsymbol{\phi}_i$ (refer to [10] for details). Therefore, in the atom selection step in $(j+1)$th iteration, we need to exclude $\mathbf{a}_k$'s if the corresponding $\boldsymbol{\phi}_k$'s are orthogonal to the $A^\dagger \mathbf{r}_j$. Similar comments apply to the orthogonal extension of the algorithm.

Algorithm 2

Basic version: CMP(S)-MP(U)

Input: Dictionary matrix A with atoms $\mathbf{a}_i, i = 1, 2, \ldots, N$, and the signal vector $\mathbf{b}$

Output: Estimation of the solution vector $\mathbf{x}$

Initialization: $\mathbf{r} = \mathbf{b}$, $\mathbf{x} = \mathbf{0}_N$, $\Delta =$ diagonal matrix whose ith diagonal entry is equal to $\delta_{ii} = 1/\sqrt{\mathbf{a}_i^T (AA^T)^{-1} \mathbf{a}_i}$.

Repeat

Selection:

$$\begin{aligned} \mathbf{y} &= \Delta A^\dagger \mathbf{r} \\ k &= \max_i |\mathbf{y}[i]| \end{aligned}$$

Update:

$$\begin{aligned} c &= \mathbf{a}_k^T \mathbf{r} \\ \mathbf{r} &= \mathbf{r} - c\mathbf{a}_k \\ \mathbf{x}[k] &= \mathbf{x}[k] + c \end{aligned}$$

until halting criterion is *true*.

Orthogonal version: CMP(S)-OMP(U)

Input: Dictionary matrix A with atoms $\mathbf{a}_i$, $i = 1, 2, \ldots, N$, and the signal vector $\mathbf{b}$

Output: Estimation of the solution vector $\mathbf{x}$

Initialization: $\mathbf{r} = \mathbf{b}$, $\mathbf{x} = \mathbf{0}_N$, $\Delta =$diagonal matrix whose ith diagonal entry is equal to $\delta_{ii} = 1/\sqrt{\mathbf{a}_i^T (AA^T)^{-1} \mathbf{a}_i}$, $I = \{\}$.

Repeat

Selection:

$$\begin{aligned} \mathbf{y} &= \Delta A^\dagger \mathbf{r} \\ k &= \max_i |\mathbf{y}[i]|; \quad I = I \cup \{k\} \end{aligned}$$

Update:

$$\begin{aligned} \mathbf{c} &= A_I^\dagger \mathbf{b} \\ \mathbf{r} &= \mathbf{b} - A_I \mathbf{c} \\ \mathbf{x}[I] &= \mathbf{c} \end{aligned}$$

until halting criterion is *true*.

5. CONVERGENCE

Having explained the two algorithms, now the question arises if they will converge. In the first algorithm, since the update is done by subtracting from the current residual the orthogonal projection on the immediately selected atom (basic version), or the orthogonal projection on the subspace spanned by all the selected atoms till the current iteration (orthogonal extension), the residual error magnitude decreases with each new iteration. Therefore, as in MP or OMP, the algorithm will definitely converge.

For the second algorithm, the convergence is not obvious. To prove that the algorithm will converge, let us assume that the atom selected in the jth iteration is $\mathbf{a}_k$. Now referring to Eqn. 10 and Eqn. 11, the weight of the selected atom is given as

$$c_k = \frac{\mathbf{a}_k^T (AA^T)^{-1} \mathbf{r}_{j-1}}{\mathbf{a}_k^T (AA^T)^{-1} \mathbf{a}_k}. \tag{14}$$

Thus the residual error at the end of the jth iteration is given as

$$\mathbf{r}_j = \mathbf{r}_{j-1} - c_k \mathbf{a}_k = (I_K - \frac{\mathbf{a}_k \mathbf{a}_k^T (AA^T)^{-1}}{\mathbf{a}_k^T (AA^T)^{-1} \mathbf{a}_k}) \mathbf{r}_{j-1}, \tag{15}$$

where I_K denotes the identity matrix of order K. Therefore,

$$\|\mathbf{r}_j\|^2 = \mathbf{r}_j^T \mathbf{r}_j = \mathbf{r}_{j-1}^T (I - \frac{\mathbf{a}_k \mathbf{a}_k^T (AA^T)^{-1}}{\mathbf{a}_k^T (AA^T)^{-1} \mathbf{a}_k}) \mathbf{r}_{j-1}. \tag{16}$$

This implies that

$$\|\mathbf{r}_{j-1}\|^2 - \|\mathbf{r}_j\|^2 = \mathbf{r}_{j-1}^T \frac{\mathbf{a}_k \mathbf{a}_k^T (AA^T)^{-1}}{\mathbf{a}_k^T (AA^T)^{-1} \mathbf{a}_k} \mathbf{r}_{j-1}. \tag{17}$$

Algorithm 3

Basic version: MP(S)-CMP(U)

Input: Dictionary matrix A with atoms $\mathbf{a}_i, i = 1, 2, \ldots, N$, and the signal vector $\mathbf{b}$

Output: Estimation of the solution vector $\mathbf{x}$

Initialization: $\mathbf{r} = \mathbf{b}$, $\mathbf{x} = \mathbf{0}_N$, Δ = diagonal matrix whose ith diagonal entry is equal to $\delta_{ii} = 1/\sqrt{\mathbf{a}_i^T (AA^T)^{-1} \mathbf{a}_i}$.

Repeat

Selection:

$$\begin{aligned} \mathbf{y} &= A^T \mathbf{r} \\ k &= \max_i |\mathbf{y}[i]| \end{aligned}$$

Update:

$$\begin{aligned} c &= \delta_{kk} \mathbf{y}[k] \\ \mathbf{r} &= \mathbf{r} - c\mathbf{a}_k \\ \mathbf{x}[k] &= \mathbf{x}[k] + c \end{aligned}$$

until halting criterion is *true*.

Orthogonal version: MP(S)-OCMP(U)

Input: Dictionary matrix A with atoms $\mathbf{a}_i$, $i = 1, 2, \ldots, N$, and the signal vector $\mathbf{b}$

Output: Estimation of the solution vector $\mathbf{x}$

Initialization: $\Phi = A^\dagger A$, $\mathbf{x}_2 = A^\dagger \mathbf{b}$, $\mathbf{r} = \mathbf{b}$, $\mathbf{x} = \mathbf{0}_N$, Δ =diagonal matrix whose ith diagonal entry is equal to $\delta_{ii} = 1/\sqrt{\mathbf{a}_i^T (AA^T)^{-1} \mathbf{a}_i}$, $I = \{\}$.

Repeat

Selection:

$$\begin{aligned} \mathbf{y} &= A^T \mathbf{r} \\ k &= \max_i |\mathbf{y}[i]|; \quad I = I \cup \{k\} \end{aligned}$$

Update:

$$\begin{aligned} \mathbf{c} &= \Phi_I^\dagger \mathbf{x}_2 \\ \mathbf{r} &= \mathbf{b} - A_I \mathbf{c} \\ \mathbf{x}[I] &= \mathbf{c} \end{aligned}$$

until halting criterion is *true*.

The expression on the right hand side is a positive semi-definite quadratic form in the components on $\mathbf{r}_{j-1}$. This is because the equivalent matrix (the matrix in the numerator divided by the quadratic form in the denominator) has one eigenvalue 1 and the remaining eigenvalues are all zeros. The expression will be zero only when $\mathbf{r}_{j-1}$ is orthogonal to either $(AA^T)^{-1}\mathbf{a}_k$ or $\mathbf{a}_k$. The first case is ruled out because of the condition in atom selection step (to avoid the infinite loop). The second case is ruled out because of the optimization in the atom selection. As a result, the updated residual error magnitude is less than the original residual error magnitude. This proves that the algorithm will definitely converge.

6. SIMULATION RESULTS

In order to compare the different sparse algorithms, we performed simulations with MATLAB. First, we created a random dictionary of 25 atoms each atom having 16 elements. The dictionary elements were generated using the standard Gaussian random number generator with mean 0 and variance 1. The atoms were then normalized to have unity magnitude.

In the first experiment, we compared the various algorithms in terms of sparsity. Using the dictionary, we created signal vectors from linear combinations of different number of atoms. The number of generating atoms was varied from 1 to 8 and the atoms were randomly selected from the dictionary. The weights of the atoms were randomly generated using a Gaussian random number generator with mean zero and variance 1. For each number of generating atoms, we created 10000 signal vectors, and the algorithms were tested for sparsity by averaging the results over these 10000 signals. We implemented the algorithms without any constraint on the number of atoms in the linear approximation, but giving a residual error bound of 0.001 per component. The plots in Fig. 1 show the average number of nonzero samples in the solution vector versus the number of atoms used to generate the signals. The average was made over 10000 signals generated for the particular number of atoms. We observe that the first algorithm performs better than

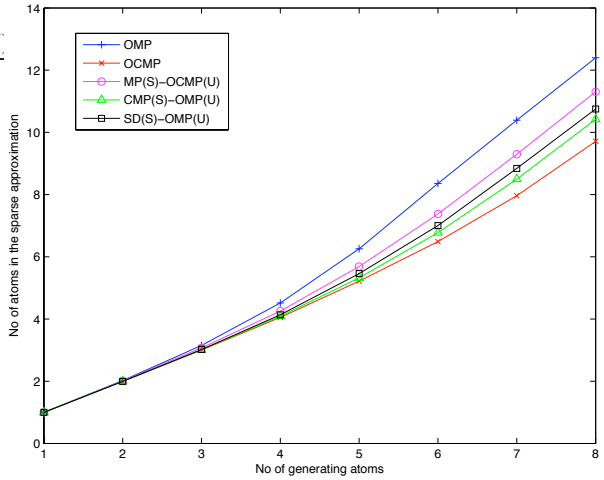

Figure 1: Sparsity of different algorithms. Residual error bound per component is 0.001. SD(S)-OMP(U) is the pseudo-inverse based sensing dictionary approach.

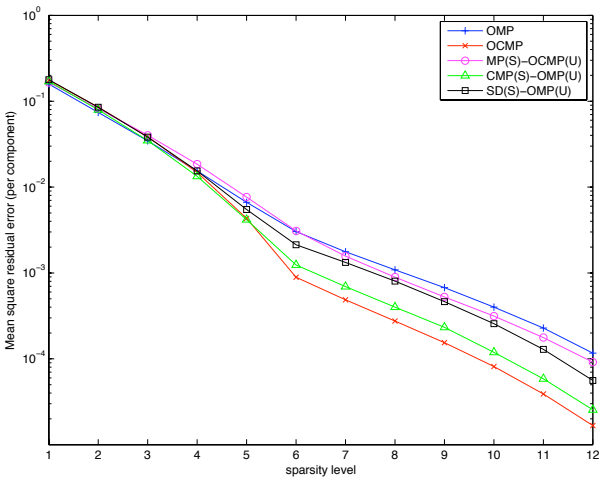

Figure 2: Residual error versus sparsity level. Number of generating atoms is 6.

the second. The performance of the sensing dictionary approach is better than the second algorithm but is worse than the first algorithm. The orthogonal CMP however performs the best among the five algorithms.

In the second experiment, we compared the approximation accuracy of the algorithms as given by the residual error energy at a given sparsity level. We generated 10000 signals with random combinations of 6 atoms as before. For different sparsity levels of the resulting sparse vector, we measured the residual error energy. The plots in Fig. 2 show the mean square error per component averaged over 10000 examples. Notice that, for sparsity level 1, the OMP produces the minimum error. This is obvious since it chooses the nearest atom. However, as the sparsity level is increased, other algorithms show better performance than the OMP. Among the two proposed algorithms, the first algorithm again performs better than the second, and also better than the pseudo-inverse based sensing dictionary approach. However, the complementary matching pursuit still produces the minimum residual error for sparsity levels 6 and higher.

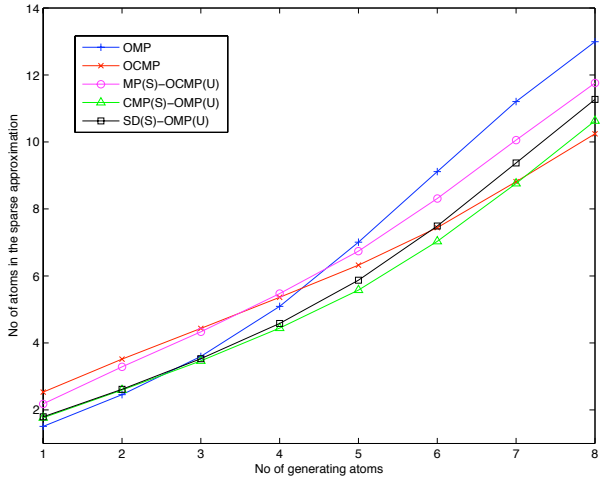

Figure 3: Sparsity of different algorithms with noisy observations. Additive noise is Gaussian with mean 0 and standard deviation 0.001. Residual error bound per component is 0.001.

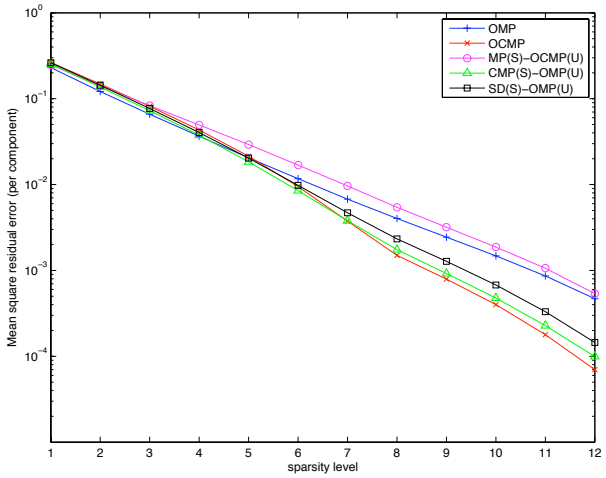

Figure 4: Residual error versus sparsity level with noisy observations. Additive noise is Gaussian with mean 0 and standard deviation 0.001. Number of generating atoms is 6.

In the final experiment, we repeated the above two experiments but with additive Gaussian noise of mean zero and standard deviation 0.001. The average sparsity results for different algorithms are shown in Fig. 3. We notice that the second algorithm and the OCMP perform poorly for small number of generating atoms. But as the number of atoms is increased, the performance of the OCMP improves gradually. The relative performances of the other three algorithms remain more or less similar except for signals generated with one or two atoms. The first algorithm continues to perform better than the pseudo-inverse based sensing dictionary approach for signals with higher number of generating atoms. The sensitiveness of the CMP to noisy observations has been mentioned in [10]. The impact of the additive noise gets less as the number of generating atoms is increased; therefore the performance of the OCMP improves gradually. Fig. 4 shows the residual error energy at different sparsity levels when the number of generating atoms is 6 and the additive noise is Gaussian with mean zero and standard deviation 0.001. We notice that the second algorithm performs worse than the OMP algorithm. The OCMP algorithm produces the minimum residual error if the sparsity is increased beyond 6.

7. CONCLUSIONS

In this paper, we have presented a comparative study of several greedy algorithms. Based on the recently proposed complementary matching pursuits, we have presented two new algorithms where we perform either the atom selection or the residual update as in the CMP. The remaining steps are similar to those in the classical matching pursuit algorithm. We compared these two algorithms with the orthogonal matching pursuit, orthogonal complementary matching pursuit and the pseudo-inverse based sensing dictionary approach. We observed that, for observations obtained from pure linear sums of atoms, the complementary matching pursuit performs the best in terms of sparsity and approximation accuracy. Among the two presented algorithms, the algorithm with atom selection as in CMP has a better performance. In the presence of additive noise, this algorithm shows better sparsity results than even the complementary matching pursuit. Based on these results, we can guess that with practical signals, this algorithm may have a better sparsity than any of the other discussed algorithms. This needs to be verified with practical applications.

REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. SP,* Vol. 41, No. 12, pp. 3397–3415, 1993.

[2] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad,"Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decompositions," *Proc. 27th Asilomar Conf. on Sig., Sys. and Comp.,* 1993.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders,"Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.,* 20, 1, pp. 33–61, 1988.

[4] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Comm. Pure Appl. Math.,* 59, 6, pp. 797–829, 2006.

[5] J. J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise," *IEEE Trans. Inform. Th.,* 51, 10, pp. 3601–3608, 2005.

[6] D. Needell and R. Vershynin, "Signal recovery from inaccurate and incomplete measurements via regularized orthogonal matching pursuit," Preprint Dec. 2007.

[7] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. and Comp. Harm. Anal.,* DOI: 10.1016/j.acha.2008.07.002.

[8] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," Preprint Jan. 2009.

[9] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Trans. on Signal Process.*, Vol. 56, No. 6. pp. 2370–2382, June 2008.

[10] G. Rath and C. Guillemot, "Complementary matching pursuit algorithms for sparse approximation," submitted to *IEEE Trans. on SP*, April 2008.

[11] K. Schnass and P. Vandergheynst, "Dictionary preconditioning for greedy algorithms," *IEEE Trans. SP,* Vol. 56, No. 5, pp. 1994–2002, May 2008.

[12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Th.,* Vol. 53, No. 12, pp. 4655–4666, Dec. 2007.