Polynomial Time and Stack Decoding Solutions to Bounded Error Subset Selection

Ahmed H. Tewfik[1] *and Masoud Alghoniemy*[2]
[1]Department of Electrical and Computer Engineering
University of Minnesota, 200 Union St. SE Minneapolis, MN 55455
[2]Dept. of Electrical Engineering, University of Alexandria
Alexandria 21544, Egypt

Abstract

The goal of Bounded Error Subset Selection (BESS) is to find the sparsest representation of an *Nx1* vector *b* using vectors from a dictionary *A* of size *NxM*, such that the approximation is within a distance δ from *b*. Here δ is a user defined approximation threshold. Specifically, the goal is to find the sparsest vector *x* such that $||Ax - b|| \leq \delta$. The BESS is a reformulation of the classical subset selection problem. We describe two enumeration approaches with bounded complexities that find the optimal solution to the BESS problem. In particular, the paper describes the first exhaustive enumeration solution to subset selection type problems with polynomial complexity. Furthermore, it also describes a lower complexity stack decoding approach that finds a solution to the BESS problem with a complexity that is proportional to that of orthogonal matching pursuit. The approaches described here have a markedly better rate-distortion behavior than any of the other known solutions to the subset selection and BESS problems.

1. INTRODUCTION

Sparse signal representation finds application in many signal processing areas such as coding, signal restoration, direction finding, source localization, and linear inverse problems, to name a few. In the subset selection problem (SS), it is required to find the best signal representation for a signal vector *b* using an overcomplete dictionary represented by the *N*-dimensional vectors spanning the column space of the matrix *A*. By construction, the number of basis vectors *M* in the dictionary is such that *N« M.* Thus, it is required to find the sparsest vector x (the vector x with the minimum number of non-zero solution) such that $Ax = b$. It is known that the SS is a NP-hard [1]. Several strategies have been developed for solving the SS problem. In particular, the Method of Frame (MoF) finds the solution which minimizes the 2-norm of the solution vector which is equivalent to minimizing the 2-norm of the reconstruction error. However, the MoF does not address the sparseness issue [2]. The Basis Pursuit (BP) algorithm, which can be solved using linear programming, finds the solution that minimizes the L_1-norm of the solution vector [1]. The BP algorithm produces a reasonably sparse solution due to the properties of solutions to L_1 minimization problems [3]. Matching Pursuit (MP) is an iterative greedy algorithm in which the signal is iteratively decorrelated from the basis vector which has maximum correlation with the residual [5]. A variant of the MP called the Orthogonal Matching Pursuit (OMP) performs an extra step of orthogonalization before each iteration [6]. However, both MP and OMP are greedy algorithms that lack a global optimization criterion. The Best Orthogonal Basis (BOB) uses an entropy measure over orthogonal bases to provide a near-optimal solution. However, as will be seen in the simulation section, BOB fails to find a good representation for some signals when they cannot be represented in terms of the assumed orthogonal structures.

In this paper, we discuss a variation of the SS problem that analyzes a perturbed version of the signal under investigation instead of the signal itself. This is a realistic assumption due to the presence of noise, masking effect, or due to channel distortion. We describe two solutions to the reformulated problem with bounded complexities. We show that exhaustive enumeration can be done for the reformulated problem with polynomial complexity. We also describe a lower complexity solution to the problem and demonstrate its superior performance when compared to any of the known solutions to the SS problem.

.

2. BOUNDED ERROR SUBSET SELECTION

The Bounded Error Subset Selection (BESS) has been introduced by the authors in [8]-[9] as a reformulation of the classical subset selection problem. It has been shown that by introducing a perturbation vector ε to the signal under investigation, *b*, one can obtain a maximally sparse representation of the signal from the overcomplete dictionary *A*. In particular, the goal in BESS is to find the sparsest vector *x* such that $||Ax - b|| \leq \delta$ for a user defined approximation threshold δ.

Two solutions were proposed to the BESS problem in [8] and [9]. In [8], the authors consider the case were the entries of *x* are restricted to be integer. In [9], an approximate solution is derived by converting the BESS problem into a sparse signal representation problem with a positivity constraint that is then solved using ordinary linear programming. In contrast, the solutions that we present here do not convert the problem into a different problem. Further, one solution is exact while the other has lower complexity and better performance than that of [8].

In the remainder of this paper, we will use the L_2 norm to measure approximation errors. Further, with no loss of generality, we will assume that the vector b and all columns a_k of A have been normalized to each have unit L_2 norm.

3. POLYNOMIAL TIME PROCEDURE

The first approach that we present is an exact solution to the BESS problem with complexity that is polynomial in M and $1/\delta$. It is based on the observation that by looking for an approximation to b within δ, we effectively induce a quantization of the N dimensional unit sphere on the surface of which b lives. This affords us the opportunity to implement an exhaustive search procedure with polynomial complexity as we explain below.

We begin by noting that the sparse signal representation problem and BESS can be solved via exhaustive enumeration Exhaustive enumeration has exponential complexity. For discussion purposes, let us consider the following exhaustive search. At step i, the procedure produces a list P_i of approximations to b using all the subsets of columns of A that were considered up to step $i-1$ after we add to each individually the *ith* column of A. The algorithm is initialized with an empty list P_0 and an empty list of subsets of columns of A used to calculate the approximations. It terminates after M steps.

To reduce the complexity of the algorithm, we proceed as follows. In addition to the lists P_i, we keep track of two additional types of lists. We shall refer to the first type of lists as the approximation subsets lists. An approximation subset list Σ_i is a list of subsets S_n of columns of A that were used to compute corresponding approximations in P_i. We also keep track of the corresponding orthogonalization subsets lists Ω_i. Each subset ω_n in this list initially is the same as the corresponding subset S_n of columns of A that was used to produce the corresponding approximation to b in P_i.

We expand the lists as follows. Suppose that at step k we are dealing with a_k the *kth* column of A. To produce an additional approximation by appending a_k to an approximation subset S_n produced in steps 1 through $k-1$, we add to the approximation corresponding to S_n the projection of b onto the component of a_k that is orthogonal to the orthogonalization subset ω_n corresponding to S_n.

We can reduce the complexity of the exhaustive search approach by using a trimming procedure. To differentiate between the trimmed and un-trimmed lists of approximations, we shall use X_i to refer to the trimmed list of approximation produced in step i. At the end of each step of the exhaustive enumeration algorithm, we trim the list X_i of approximations that we have produced by eliminating from X_i any approximation that is within a distance δ_1 from another approximation that is closer to b or uses a smaller number of columns of A. We also update the orthogonalization subset corresponding to the approximation that we retained by replacing it with the approximation subset corresponding to the approximation that was eliminated. Partial pseudo code for the approach is listed below.

```
Calculate δ1 from δ
M ← |A|
L0={} % Li list of subsets of dictionary vectors ai that were used to
        produce approximations
X0={} % Xi list of approximations bi
for i ← 1 to M
        do {Li, Xi} ← Merge-Lists{(Li-1, Li-1 + ai), (Xi-1, Xi-1 + ai)}
           {Li, Xi} ← Trim(Li, Xi,;δ)
Let b*= A x* be the sparsest approximation to b in A that satisfies ||b-b*|| ≤ ε
return b* and x*
```

The modified exhaustive search approach described can be shown to provide a solution to the BESS problem and have complexity that is polynomial in M for any non zero choice of δ. An outline of the proofs follows. Suppose that for each approximation y_i in the untrimmed list P_i there is an approximation z_i in X_i such that $||z_i-y_i|| \leq \delta_1$. It then follows from the way the orthogonalization and approximation subsets are constructed that if z_i is dropped from X_i, we will nevertheless have $||z_{i+1}-y_{i+1}|| \leq \delta_1$. Hence, for each approximation y_{i+1} in the untrimmed list P_{i+1} there is an approximation z_{i+1} in X_{i+1} that is within δ_1 of y_{i+1}. This will hold true in particular for the optimal approximation $y_* = b$ in P_M. By picking $\delta_1 = \delta/M$ it then follows that the output z_* of the procedure will be a solution to the BESS problem.

The polynomial complexity of the algorithm in M and $1/\delta$ follows from the fact that the algorithm never has to keep track of more than δ^{-N} approximations and terminates after M steps. Note however that the complexity of the procedure is exponential in N.

4. STACK DECODING ALGORITHM

The main drawback of the polynomial time procedure is that it has a large memory requirement. Specifically, in the worst case, it needs to keep track of up to δ^{-N} approximations and their corresponding lists of approximation and orthogonalization vectors. To alleviate this problem, we propose a stack decoding procedure that generalizes orthogonal matching pursuit. As we shall see in the results section, the procedure yields better rate-distortion curves than any of the subset selection procedures that have been reported in the literature so far.

The stack decoding procedure maps the BESS problem onto a tree structure. Each node of the tree represents a particular approximation. The depth of a node indicates how many vectors were used in the approximation. The branches of the tree indicate the vectors that were used to obtain the approximation. In particular, the root of the tree is zero and corresponds to an approximation that uses no column of A. The M children of the root correspond to all possible approximations of b that use a single column of A, with the branch from the root to its child indicating which column of A was used. By expanding each child of the root, we reach the depth 2 nodes. These nodes correspond to approximations of b that use two columns of A. Note that each of the M children of the root has $M-1$ children of its own. The branch from that child to any of its children indicates which column of A was added to get the new approximation. By tracing the path from the root to the any level 2 node, we recover the 2 columns of A that were used to compute the approximation corresponding to the node. The exhaustive solution to the BESS problem can theoretically be obtained by expanding the tree, level by level, up to N levels to get all possible representations of b in terms of subsets of columns of A, or expanding it until an approximation is found with an approximation error less than the desired approximation error bound δ.

To get a manageable algorithm, the stack decoding procedure implements an essentially breadth first tree search. This is to be contrasted with the mainly depth first behavior of the polynomial time exhaustive search procedure. It prunes the tree corresponding to the BESS problem after extending it by one level using three pruning procedures. First, it does not fully expand each node as it increases the depth of the tree. It simply retains for each node the best K_2 of its children, i.e., it retains at most for each node K_2 additional approximations. Next, it implements the pruning approach that is used by the polynomial time search algorithm. Specifically, it eliminates from the tree any approximation that is within a distance δ_1 from another approximation that is closer to b or is at a higher level (lower depth). Finally, it keeps the best K_1 approximations, i.e., it trims the tree and keeps only the best K_1 nodes.

As in all stack algorithms, the cost function that we use for selecting which nodes to trim, can play a major role in the computational complexity of the procedure and its performance. The cost function should measure the likelihood that any given path from the root will be on the optimal solution. Part of that likelihood can be evaluated from the approximation error corresponding to the node at the end of the path at any given step. The likelihood of the remaining path can be estimated as follows. Suppose that we are at node n_* and have performed k tree expansion steps, i.e., the deepest node in the tree is at most at level k. Further, let L be an upper bound on the number of levels that we intend to use ($L<N$). We can add to the energy of the approximation to b corresponding to n_*, the sum of the energy of the projections of b on the closest $L-k$ columns of A to b in the L_2 norm that are not on the path from the root to node n_*. Note that these columns are readily available from the calculations performed by the algorithm to reach node n_*. As the algorithm proceeds, this calculation of the cost function can be refined at the expense of additional memory. This follows from the fact that as the algorithm proceeds, it can keep track of a subset of energies of all approximations with 1, 2, etc. approximations of b in terms of the vectors that have not been used by any of the paths to the best K_1 approximations retained so far.

Note that the complexity of this approach described so far is no more than $K_1 K_2$ that of orthogonal matching pursuit. In our work, we have found that for all values of N, M and δ that we have considered, it is enough to take K_1 and K_2 to be 3 to find the optimal approximation to a given signal within the specified approximation error.

We have also found that we need to add a backwards elimination step to the procedure to find the sparsest possible solution. This is due to the fact that the algorithm may end up with an approximation within the desired error δ that uses more columns of A than is absolutely needed. This behavior is not unlike the behavior seen with normal orthogonal matching pursuit. The backwards elimination step recursively eliminates from the list of columns of A corresponding to the best solution identified by the algorithm one column at a time, as long as the resulting approximation error does not exceed the desired approximation error bound δ.

Finally, note that a variant of the stack decoding is guaranteed to find the optimal solution to BESS at the expense of variable added memory and computational requirements. Specifically, the variant is based on the observation that in the worst case, the rate-distortion curve corresponding to the optimal solution will be linear in the number of vectors a_k retained for approximation. Hence, the cost function it uses at step k is equal to the square root of the square of the approximation error at stage k minus $(L-k)/L$. The procedure then retains all approximations with a cost less than δ as opposed to the best K_1 approximations. The number of approximations it retains will thus change from problem to problem and from iteration to iteration for a given BESS problem.

5. RESULTS

The proposed algorithm was compared to the well-known methods for sparse signal representation, namely, Basis Pursuit, Orthogonal Matching Pursuit, and Best Orthogonal Basis with L_1 entropy. Simulation was performed on different signals and different dictionaries derived from the Atomizer package [10]. For illustration purpose, Figs. 1 and 2 show the Carbon and Doppler signals of [10]. The signals were analyzed using the wavelet packet dictionaries generated by [10]. Figs. 3 and 4 display the corresponding rate-distortion behavior of the various algorithms for the two signals. In Figs. 3 and 4, the proposed stack decoding algorithm and that of [9] are referred to as “Pruned enumeration” and “BESS” respectively. The stack decoding results (“Pruned enumeration”) were obtained by setting $K_1 = K_2 = 3$ and using a cost function equal to the approximation error, i.e., not using any approximation error prediction term in the cost function. Note the dramatically improved behavior of the proposed approach. Note in particular that while OMP fails to represent the Carbon signal properly, the proposed algorithm was able to represent it using fewer coefficients compared to the BOB and BP techniques. Similarly, the proposed algorithm succeeded in sparsely representing the Doppler signal compared to the other techniques as shown in Fig. 2.

6. REFERENCES

[1] B. Natarajan, ”Sparse Approximate Solutions to Linear Systems,” SIAM J. Comp., Vol. 24, pp. 227-234, Apr. 1995.

[2] Daubechies I., ”Time-Frequency Localization operators: A Geometric Phase Space Approach,”*IEEE Trans. on Info. Theory,* Vol. 34, No. 4, pp. 605-612, July 1988.

[3] J. J. Fuchs, B. Delyon, “Minimal L1 norm reconstruction of oversampled signals”. *IEEE Trans. on Info. Theory,* vol 46, No 4, p. 1666-1672, July 2000.

[4] Chen S., and Donoho D., ”Atomic Decomposition By Basis Pursuit,” SIAM J. on Scientific Computing, Vol. 20, No. 1, pp. 33-61, 1998.

[5] Mallat S., and Zhang Z., ”Matching Pursuit with Time-Frequency Dectionaries,” *IEEE Trans. on Signal Processing,* Vol. 41, No. 12, pp. 3397-3415, Dec. 1993.

[6] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, ”Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” *in Proc. 27th Asilomar Conference on Signals, Systems and Computers, A. Singh, ed.*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1993.

[7] Coifman R., and Wickerhauser M., ”Entropy-based Algorithms for Best Basis Selection,” *IEEE Trans. on Info. Theory,* Vol. 38, No. 2, pp.713-718, March 1992.

[8] Masoud Alghoniemy and Ahmed H. Tewfik, ”A Sparse Solution to the Bounded Subset Selection Problem: A Network Flow Model Approach,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.*, Vol. 5, pp. 89-92, May 2004.

[9] Masoud Alghoniemy and Ahmed H. Tewfik, “Reduced Complexity Bounded Error Subset Selection,” *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing.*, Vol. 5, pp. 725-728, March 2005.

[10] http://www-stat.stanford.edu/ atomizer.

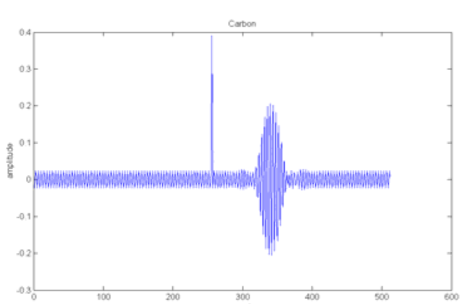

Fig. 1: Carbon signal

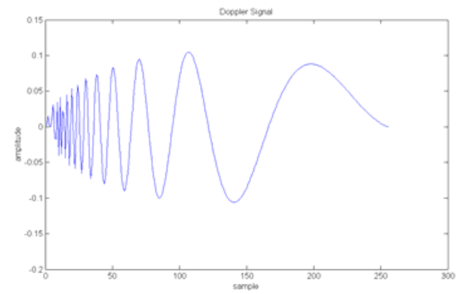

Fig. 2: Doppler signal.

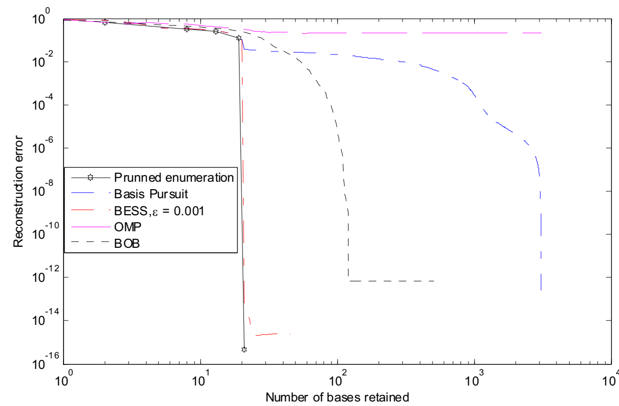

Fig. 3: Rate Distortion curves for Carbon signal.

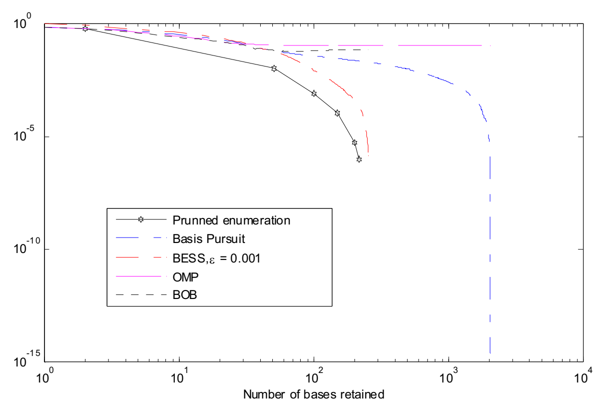

Fig. 4: Rate Distortion curves for Doppler signal.