A SPARSE SOLUTION TO THE BOUNDED SUBSET SELECTION PROBLEM: A NETWORK FLOW MODEL APPROACH

Masoud Alghoniemy

Electrical Engineering Department
University of Alexandria,
Alexandria 21544, Egypt

Ahmed H. Tewfik

Electrical Engineering Department
University of Minnesota,
Minneapolis, MN 55455

ABSTRACT

We reformulate the problem of finding the sparsest representation of a given signal using an overcomplete dictionary as a bounded error subset selection problem. Specifically, the reconstructed signal is allowed to differ from the original signal by a bounded error. We argue that this bounded error formulation is natural in many applications, such as coding. Our novel formulation guarantees the sparsest solution to the bounded error subset selection problem by *minimizing* the number of nonzero coefficients in the solution vector. We show that this solution can be computed by finding the minimum cost flow path of an equivalent network. Integer programming is adopted to find the solution.

1. INTRODUCTION

Classical subset selection arises in many signal processing applications. In particular, most signals cannot be represented by a single dictionary. For example, Fourier bases are suitable for signals which are rich in harmonics while wavelet dictionaries can be used for signals that include transients. Hence, overcomplete dictionaries are required to cover the inherent complex signal structure. In the subset selection problem (SSP) it is required to find the best signal representation for a signal vector $\mathbf{b}$ using the overcomplete dictionary represented by the N dimensional vectors spanning the column space of the matrix $\mathbf{A}$. By construction, the number of basis vectors in the dictionary $M \gg N$. Then it is required to find the solution vector $\mathbf{x}$ such that

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

It is also required that the solution be sparse, i.e., the vector $\mathbf{x}$ should have a minimum number of non-zero coefficients. The solution satisfying the previous sparseness requirement is called a sparse solution. It is known that the SSP is NP-complete [1]. In particular, approximate solutions can be found such that

$$\| \mathbf{Ax} - \mathbf{b}\|_p \leq \epsilon \quad (2)$$

for some $\| .\|_p$ and $\epsilon \geq 0$. Several strategies have been developed for solving the SSP. For example, the Method of Frames (MoF) finds the solution that minimizes the l_2 norm which leads to the solution closest to the origin [2]. On the other hand, the solution of the Basis Pursuit (BP) algorithm minimizes the l_1 norm [3]. Mallat *et. al* developed the Matching Pursuit (MP) technique in which the signal is iteratively decorrelated from the basis vector which has maximum correlation with the residual [4]. The Best Orthogonal Basis (BOB) is designed for wavelet and cosine packet dictionaries which finds the solution based on a minimum entropy criterion [5]. The previous techniques do not necessarily find the sparsest solution since the optimization criterion does not address the sparsity issue. The authors in [6] found the sparse solution for certain structured matrices. For a very good discussion on the sparseness constraints the reader is referred to [7].

In this paper we address the sparseness problem directly by finding the solution that *minimizes* the number of nonzero coefficients in the solution vector. This is achieved by reformulating the subset selection problem as a bounded error problem which allows for exact solution, if it exists. In applications such as speech and audio coding, one could have bounds on the error through the masking model. In particular, the equality constraint in (1) could be relaxed according to

$$\mathbf{b}_{min} \leq \mathbf{Ax} \leq \mathbf{b}_{max} \quad (3)$$

Hence, the name *Bounded* Subset Selection Problem (BSSP). Where, $\mathbf{b}_{min} = \mathbf{b} - \vec{\epsilon_1}$ and $\mathbf{b}_{max} = \mathbf{b} + \vec{\epsilon_2}$. With $\vec{\epsilon_1}$ and $\vec{\epsilon_2}$ are error vectors representing the error introduced by the masking model [8]. Then, instead of minimizing the norm of the error as in [2, 3, 4] we bound the error signal by $\vec{\epsilon_1}$ and $\vec{\epsilon_2}$. As stated earlier, bounding the error signal rather than bounding the norm of the error finds application in speech and audio coding, in which the energy of the error does not necessarily represent the quality of the coded signal. In particular, the quality measure is mainly subjective, the temporal structure of the coded signal is more important in determining the signal quality than the energy of the error. [8]. In the following section, a brief overview to the network flow model and integer programming used in solving (3) is given. Simulation results are given in section 3.

2. THE PROPOSED APPROACH

In this section, we will formulate the solution strategy for finding the sparsest possible solution to (3). To find the required solution, we will adopt a network flow model approach. We will show that finding the sparse solution is equivalent to finding the minimum cost path between two points in an equivalent network. However, this path is subject to the constraints imposed by the signal to be represented as illustrated below.

2.1. Network Flow Model

Consider a network that has M nodes and there is a flow that starts from the source node, S, and ends at the sink node, T. Each node is connected to the other nodes via arcs. The minimum cost flow problem finds the minimum cost continuous path from S to T. The minimum cost flow network has some constraints which we mention here.

- The flow path should be a *continuous* path. i.e., any path from the source to the sink should be connected and no node can be considered more than once.
- Since the flow starts from S and ends at T, then arcs connected to S should be outgoing from S while arcs connected to T should be terminating at it, i.e., it is not allowed for S to accept flow and at the same time it is not allowed for T to produce any flow.
- Except for S and T, the flow can go from any two nodes in both directions.
- The network is balanced, i.e., except for S and T each node should have zero accumulation rate. This can be achieved by not allowing self-returning arcs at any node which assures that the flow will not accumulate at any node.
- Each arc is weighted with a cost and, if necessary, a weight. The total path cost and weight is the sum of all costs and weights along that path.

Fig. 1 shows a simplified network of 5 nodes that satisfies the previous constraints.

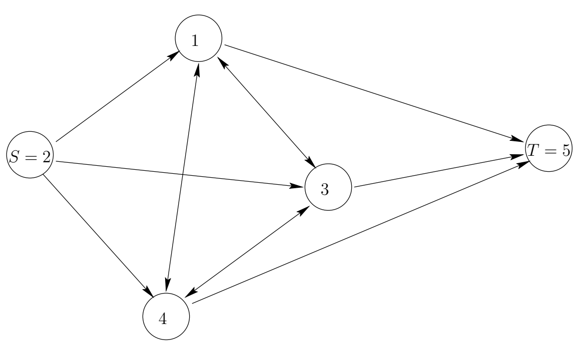

Figure 1: A Network with 5 nodes

The solution to the network flow problem lies in finding a continuous path connecting S to T with a minimum cost. Several approaches are available for the network flow problem solution including dynamic and linear programming [9]. In order to relate the network flow problem to the BSSP, let each node represents a possible candidate in the overcomplete dictionary $\mathbf{A}$. In particular, let node k represents the N-dimensional vector $\mathbf{a}_k$ in the $N \times M$ matrix $\mathbf{A}$, where $M \gg N$ in practical cases. It should be noted that both the source, S, and the sink, T, nodes are represented similarly as they could be part of the solution. This issue is addressed in section 2.2.2. Since we would like to find the minimum cost path in going from the source to the sink subject to certain constraints; then the problem at hand is trying to identify the *least* possible number of vectors $\mathbf{a}_k$ that best fits the signal represented by the vector $\mathbf{b}$. i.e., we are trying to *minimize* the number of vectors that best represent a given signal. As stated earlier, each arc is assigned a certain cost, c_{ij}, and weight, $\mathbf{w}_{ij}$. Where c_{ij} is the cost of going from node i to node j and $\mathbf{w}_{ij}$ is the weight of the arc connecting nodes i and j.

2.1.1. Costs and Weights Assignment

Since each vector, $\mathbf{a}_k$, is a possible candidate in the solution, then the corresponding node should also be a possible candidate in the minimum cost path. Hence, all arcs should have the same cost in the network, i.e.,

$$c_{ij} = 1 \quad \forall \ i, j \tag{4}$$

equation (4) indicates that each vector in the dictionary has the same probability of being selected. It should be noted that $c_{ij} = c_{ji}$.

Since the solution to the minimum cost flow problem is not unique, it is necessary to assign weights to the corresponding arcs. Assigning weights to each arc guarantees that the subset selection problem solution will satisfy the constraints imposed by the signal to be analyzed. Hence, according to (3), in order to find the sparse solution to the BSSP, the network model should satisfy

$$\mathbf{b}_{min} \leq \sum_{k\in\Omega} \mathbf{a}_k = \sum_{i=S}^{j=T} \mathbf{w}_{ij} \leq \mathbf{b}_{max}. \tag{5}$$

where Ω is the solution set for the basis vectors. Assigning the weights $\mathbf{w}_{ij}$ as follows:

$$\mathbf{w}_{ij} = \begin{cases} 2\mathbf{a}_i - \mathbf{a}_j & if \quad i \neq S, \ j \neq T \\ \mathbf{a}_i - \mathbf{a}_j & if \quad i = S, \ j \neq T \\ 2\mathbf{a}_i + \mathbf{a}_j & if \quad i \neq S, \ j = T \end{cases} \tag{6}$$

guarantees that $\sum_{i=S}^{j=T} \mathbf{w}_{ij} = \sum_{k\in\Omega} \mathbf{a}_k$. It is clear from (6) that $\mathbf{w}_{ij} \neq \mathbf{w}_{ji}$.

2.1.2. Imposing Sparseness

Sparseness is imposed by finding the path that *minimizes* the total number of nodes in going from the source to the sink, hence minimizing the total cost subject to (5). Putting the previous formulation into a mathematical context, then the solution to the BSSP is equivalent to the solution of the following optimization problem:

$$\min_{i,j} \sum_{i=S}^{j=T} c_{ij} \tag{7}$$

$$\textit{subject to} \quad \mathbf{b}_{min} \leq \sum_{i=S}^{j=T} \mathbf{w}_{ij} \leq \mathbf{b}_{max}$$

I what follows we will discuss the integer programming approach in solving (7).

2.2. Integer Programming Solution

To correctly define the variables in the integer program, an Adjacency Matrix (AM) needs to be defined. An adjacency matrix is another form of representing the network which makes it easier to relate variables together. In general, a network of M nodes has an AM of size $M \times M$ and each cell in the matrix represents an arc in the original network and its value is the cost of this arc as in (4). For illustration purposes we will consider the adjacency matrix corresponding to the network in Fig. 1. The network in Fig. 1 has only 5 nodes from which $S = 2$ and $T = 5$ and its AM is represented by

$$\mathbf{AM} = \begin{bmatrix} \alpha & \alpha & 1 & 1 & 1 \\ 1 & \alpha & 1 & 1 & \alpha \\ 1 & \alpha & \alpha & 1 & 1 \\ 1 & \alpha & 1 & \alpha & 1 \\ \alpha & \alpha & \alpha & \alpha & \alpha \end{bmatrix} \tag{8}$$

where α is a large number that represents a high cost. Variable α also represents invalid cells in the sense that its coordinates do not represent a valid arc in the network. For example, α is used in the second column of the AM to represent invalid cells. These cells are invalid since we are not allowed to go to the source, $S = 2$, from any other node as described in section 2.1. The same is true for the fifth row and diagonal cells. It should be noted that the cell AM(S, T) is also invalid to avoid a direct transition from the source to the sink.

Let $y_{ij} \in \{0, 1\}$ be the variable associated with the cell AM(i, j) that captures the solution to the optimization (7), i.e., $y_{ij} = 1$ if the cell AM(i, j) is a valid cell in the flow path and $y_{ij} = 0$ if not. Also, let Γ be the set of valid cells, then one can rewrite the network constraints as follows

$$\sum_{i \in \Gamma} y_{Si} = 1 \tag{9}$$

$$\sum_{i \in \Gamma} y_{iT} = 1 \tag{10}$$

$$\sum_{k \in \Gamma} y_{ki} = \sum_{j \in \Gamma} y_{ij}, \quad i \notin \{S, T\} \tag{11}$$

$$\sum_{i \in \Gamma} y_{ij} \in \{0, 1\}. \tag{12}$$

$$y_{ij} + y_{ji} \in \{0, 1\}. \tag{13}$$

Constraints (9) and (10) ensure that the sequence starts from the source and ends in the sink respectively. On the other hand, (11) takes care of the continuity of the flow by restricting the amount of the flow coming out of any node to be equal to the amount of the flow entering that node, i.e., it captures the continuous path with minimal cost. Constraint (12) ensures that no basis vector is chosen more than once in any solution. Finally constraint (13) is imposed to avoid having two isolated nodes, which guarantees that the captured solution will not be of oscillatory nature.

2.2.1. *Integer Programming*

Transforming the problem into an integer program requires mapping the AM into a one dimensional vector. This is performed by lexicographically ordering the AM into a vector $\mathbf{f} = vec(AM^T)$ of dimensionality $M^2 \times 1$, and transforming y_{ij} into $x_k \in \{0, 1\}$ according to the mapping

$$k = (i - 1)M + j. \tag{14}$$

It should be noted that each element of the vector $\mathbf{f}$ represents the cost of the corresponding arc in the network. Network constraints described in the previous subsection are imposed onto the constraint matrix $\mathbf{B}$ and the corresponding constraint vector $\mathbf{c}$. Similarly, arc weights are represented by the $N \times M^2$ constraint matrix $\mathbf{W}$ by rearranging its column space according to (14). Then, the network flow solution can be found by solving the following binary optimization problem

$$\min_{\mathbf{x}} \ \mathbf{f}^T \mathbf{x} \tag{15}$$

$$subject \ \ to \begin{cases} \mathbf{Bx} = \mathbf{c} \\ \mathbf{Wx} \geq \mathbf{b}_{min} \\ \mathbf{Wx} \leq \mathbf{b}_{max} \\ x_k \in \{0, 1\}. \end{cases}$$

It is clear that (15) follows the standard form of binary integer programming and can be solved using the branch and bound algorithm [10, 11]. According to the network model in Fig. 1, one needs to define the source and the sink. However, the solution set of subset selection problem does not need to start or end with a certain vector. Assume a solution to (15) is found which consists of L bases vectors, then all the $L!$ combinations to these bases will also be a solution to (15). In order to adapt the network model to the subset selection problem, we will proceed using a vector space interpretation as follows

2.2.2. *Vector Space Interpretation*

Since each basis in the dictionary can be represented by a vector in the N-dimensional space, then the dictionary can be represented as an N-dimensional object in the N-dimensional space. For illustration purpose, consider the 3-D case with $M = 8$. The corresponding object is shown in Fig. 2. According to the network model, each node in the object represents a node in the corresponding network and a valid basis in the solution set. The source and the sink can now be chosen as any 2 nodes. However, this would enforce the vectors corresponding to S and T, to be part of the solution. In order to remove this constraint, both the source and the sink can be chosen to be *dummy* nodes, i.e., any two vectors which are not included in the dictionary. A simple, yet effective choice is to let $S = T$ as shown in Fig. 2 where the dummy vector is the mean vector. The system of equations in (15) is modified accordingly

$$\min_{\mathbf{x}} \ \mathbf{f}^T \mathbf{x} \tag{16}$$

$$subject \ \ to \begin{cases} \mathbf{Bx} = \mathbf{c} \\ \mathbf{Wx} \geq \mathbf{b}_{min} + \mathbf{v}_S + \mathbf{v_T} \\ \mathbf{Wx} \leq \mathbf{b}_{max} + \mathbf{v}_S + \mathbf{v_T} \\ x_k \in \{0, 1\}. \end{cases}$$

where $\mathbf{v}_S$ and $\mathbf{v}_T$ are the vectors representing the source and the sink respectively. Here, $\mathbf{v}_S = \mathbf{v}_T$. In this case, the number of the retrieved bases will be increased by two to accommodate for the dummy nodes, namely $\mathbf{v}_S$ and $\mathbf{v}_T$.

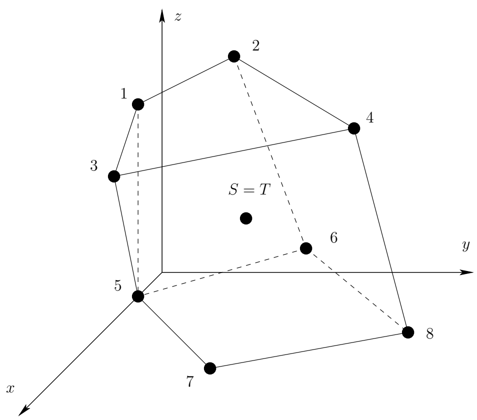

Figure 2: A 3-D object representing an overcomplete dictionary consisting of 8 vectors.

According to the network model, the solution to the BSSP is a continuous path from the source to the sink passing through the appropriate nodes. For illustration purposes, a possible path is shown in Fig. 3 as the bold line connecting nodes { S, 4, 6, 7, 1, T} together.

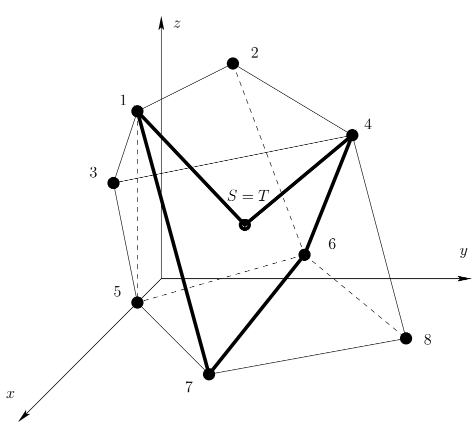

Figure 3: The bold line represents a possible solution to the BSSP

3. SIMULATION

In order to show that the proposed approach finds the sparsest solution to the subset selection problem, we compare the proposed algorithm with the MoF and the BP algorithms. Consider a signal which is represented by the vector $\mathbf{b} = [4.9728\ \ 6.5135\ \ 7.0047\ \ 8.8925\ \ 7.1056\ \ 5.8194\ \ 6.1580\ \ 7.0885\ \ 5.6537\ \ 5.3230]^T$ which can be represented as the sum of the four column vectors $\mathbf{b} = \mathbf{b_1} + \mathbf{b_2} + \mathbf{b_3} + \mathbf{b_4}$ in the set $\Omega = [\mathbf{b_1}\ \mathbf{b_2}\ \mathbf{b_3}\ \mathbf{b_4}]$, not shown here for the sake of space. It is required to find the sparsest solution to the *perturbed* vector $\mathbf{b_p} = \mathbf{b} + \vec{\epsilon}$ where $\vec{\epsilon} = [0.8913\ \ 0.7621\ \ 0.4565\ \ 0.0185\ \ 0.5247\ \ 0.6412\ \ 0.0162\ \ 0.8369\ \ 0.8035\ \ 0.6978]^T$, is a random perturbation. The overcomplete dictionary is formed by $M = 30$ random vectors each of dimension $N = 10$. The vectors of the set Ω were placed at random locations in the dictionary. Both the source and the sink are represented by the mean vector of the dictionary.

The solution was as follows: the MoF, as expected, gave a dense solution, that is, the number of non-zero elements in the solution vector equals to the size of the dictionary, which is 30. While the BP algorithm finds a solution which has 10 non-zero elements. Surprisingly enough, by letting $\vec{\epsilon_1} = \vec{0}$ and $\vec{\epsilon_2} = \vec{\epsilon}$, the BSSP finds the solution vector which has only four non-zero coefficients that corresponds to the four vectors in Ω. Thus, it finds the *exact* and sparsest signal representation even though the input signal was perturbed. Unlike ours, the failure of the BP and the MoF to find the sparsest solution can be understood from the fact that they are not designed to minimize the number of bases in the solution.

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

[4] 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.

[5] 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.

[6] Nafie M., Tewfik A.H., and Ali M., "Deterministic and iterative solutions to subset selection problems", *IEEE Trans. on Signal Processing,* pp. 1591-1601, July 2002.

[7] Bhaskar D. Rao, "Signal Processing with the Sparseness Constraint," *Proc. International Conference on Acoustics, Speech, and Signal Processing.* Vol. 3, pp. 1861-1864, 1998.

[8] Painter T., and Spanias A., "Perceptual Coding of Digital Audio." *Proceedings of the IEEE,* Vol. 88, No. 4, pp. 451-513, April 2000.

[9] R. Rardin, Optimization in Operations Research, Prentice Hall, 1997

[10] L. Wolsey, Integer Programming, John Wiley & Sons Inc., 1998.

[11] C. Floudas, Nonlinear and Mixed-Integer Optimization, Oxford University Press, 1995.