Efficient Backward Elimination Algorithm for Sparse Signal Representation Using Overcomplete Dictionaries

Shane F. Cotter, K. Kreutz-Delgado, and B. D. Rao

***Abstract*—A sparse representation of a signal, i.e., a representation using a small number of vectors chosen from a dictionary of vectors, is highly desirable in many applications. Here, we extend the backward elimination sparse representation algorithm presented in [1] to allow for an overcomplete dictionary and develop recursions for its implementation. In the overcomplete case, the representation error cannot be used as a general criterion for elimination of a dictionary vector and other criteria must be considered. Simulation on a test-case dictionary shows that the performance of the proposed algorithm can improve upon that of forward selection methods.**

***Index Terms*—Backward elimination algorithm, overcomplete dictionary, sparse representation, subset selection.**

I. Introduction

In THIS letter, which builds on work presented in [2], we consider the problem of representing a signal, $b \in \mathbb{C}^m$, using elements from a large dictionary of signals and provide a novel extension to the results reported in [1]. The dictionary is denoted by the matrix $A \in \mathbb{C}^{m \times n}$ where each column of A is an element of the dictionary. The dictionaries considered here are overcomplete, as opposed to the undercomplete dictionaries in [1], which means that $m < n$, and in many cases $m \ll n$. In general, an infinite number of different representations of the signal exist due to the existence of a nontrivial nullspace. However, in many applications we seek a sparse representation of the signal [3]. Such representations, which are the focus of this paper, are a linear combination of $r \leq m$ dictionary elements and so there are no more than $\sum_{r=1}^{m} \binom{n}{r}$ possible sparse representations. This is nonetheless still usually a very substantial number of solution possibilities and searching for the best sparse solution from within this set is known to be NP-complete [4]. Consequently, many suboptimal sparse representation algorithms have been proposed.

Suboptimal forward selection methods and methods based on l_1-norm minimization as well as the more general $l_{p\leq 1}$ p-norm-like diversity measure have been explored [5], [6]. Reference [1] has recently proposed an efficient implementation of a backward greedy algorithm which improves computationally on a method first proposed in [7]. The algorithm of [1] proceeds by sequentially eliminating elements from a nonovercomplete dictionary (i.e., $m \geq n$) such that the error in representing the signal, b, using the remaining dictionary elements is minimized in each iteration. The algorithm as presented in [1], [7] is appropriate for a dictionary which is undercomplete and for which the representation error is an increasing function of the representation size. For an overcomplete dictionary, as considered in this work, the representation error will, in general, be zero. Therefore, alternative criteria to the representation error must be used in determining which dictionary elements are to be removed.

Working with a different, more expository, mathematical framework in [2], we independently derived essentially the same recursions for an undercomplete dictionary as in [1]. However, in [2] we also derived recursions appropriate to an initial overcomplete dictionary. These results complement and extend the work of [1], [7] but, in contrast to the derivations of [1], the derivations in [2] necessitated the nonstraightforward application of a *generalized* Sherman-Morrison-Woodbury matrix inversion lemma derived in [8]. By adopting the more elegant methodology of [1], we have found a more succinct derivation than [2] which also clarifies how the resulting algorithm would be implemented in practice. A brief simulation is included which demonstrates the success of this technique. The results of the current paper in combination with those of [1] now provide a complete solution to the backward sequential elimination problem for the cases of overcomplete and undercomplete dictionaries.

Manuscript received April 20, 2001; revised March 27, 2002. This work was supported in part by the National Science Foundation under Grant CCR-9902961. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yingbo Hua.

The authors are with the Electrical and Computer Engineering Department, University of California, San Diego, La Jolla, CA 92093-0407 (e-mail: scotter@ece.ucsd.edu; kreutz@ece.ucsd.edu; brao@ece.ucsd.edu).

Publisher Item Identifier S 1070-9908(02)06041-8.

II. Backward Elimination Algorithm for Overcomplete Dictionaries

The problem can be formulated as the linear inverse problem

$$Ax = b \tag{1}$$

where the dictionary is represented by the matrix $A \in \mathbb{C}^{m \times n}$ which is full row rank, $b \in \mathbb{C}^m$ is the target vector to be sparsely represented by a subset of columns of A and $x \in \mathbb{C}^n$ is the solution vector. A solution to this problem is said to be sparse when a significant number of components of x are zero.

A commonly used procedure to obtain a unique solution to the underdetermined system in (1) is to find the minimum 2-norm solution to this problem, i.e.,

$$x^{\dagger} = \arg\min_{x}\{\|x\| \mid Ax = b\} \tag{2}$$

which is evaluated as $x^{\dagger} = A^{\dagger}b$, where $A^{\dagger} = A^H(AA^H)^{-1}$ is the pseudoinverse of A. In this paper, we use $\|\ldots\|$ to denote the 2-norm; other norms will be explicitly subscripted. The solution, $x^\dagger$, lies in the space $\mathcal{R}(A^H)$ but is generally not sparse. Since the equality of (1) must be preserved, the solution is perturbed by utilizing the nullspace of $A, \mathcal{N}(A)$.

The starting point for our algorithm is the calculation of this minimum norm solution and then components of $x^\dagger$ are sequentially removed to give a sparse solution with the remaining components modified if necessary. For example, in the first iteration if the column a_k is removed from A, corresponding to the deletion of the kth component of $x^\dagger$, we denote the remaining matrix by $A_{-k}^{(1)}$. Here, the superscript gives the number of columns deleted from the initial matrix A and k indicates the most recently deleted column. The minimum 2-norm solution is then sought for the modified system

$$A_{-k}^{(1)} x = b \tag{3}$$

and is denoted by x_1, where $x_1 \in \mathbb{C}^{n-1}$. At the lth stage, we have $A^{(l)}$ which is A with l columns deleted. If $\text{rank}(A^{(l)}) = m$, then we still have a full row rank underdetermined system and $x_l \in \mathbb{C}^{n-l}$ can be found which satisfies

$$A^{(l)} x_l = b. \tag{4}$$

This process can be iterated as long as $\text{rank}(A^{(l)})$ continues to be m. However, once $\text{rank}(A^{(l)}) < m$, the system is of full column rank and the equality in (4) may no longer hold, i.e., in general, the representation error, $\|A^{(l)}x - b\| \geq 0, \forall x$. The problem now becomes identical to that considered in [1] and the procedure outlined there may be used to obtain a more sparse solution.

We now consider the recursive computation of the pseudoinverse, as well as the minimum norm solution, once the column a_k has been chosen for deletion. The first iteration is detailed as this allows us to simplify notation by dropping the superscript from $A_{-k}^{(1)}$. We let

$$A = [A_{-k} \;\; a_k]\Pi^H \tag{5}$$

where Π is an appropriate permutation matrix. Then we have

$$AA^H = A_{-k}A_{-k}^H + a_k a_k^H. \tag{6}$$

The pseudoinverse of A_{-k} is obtained as

$$\begin{aligned} A_{-k}^\dagger &= A_{-k}^H (A_{-k}A_{-k}^H)^{-1} \\ &= A_{-k}^H (AA^H - a_k a_k^H)^{-1}. \end{aligned} \tag{7}$$

Using the well-known Sherwood-Morrison-Woodbury formula [9], this becomes

$$A_{-k}^\dagger = A_{-k}^H (AA^H)^{-1} + \frac{A_{-k}^H (AA^H)^{-1} a_k a_k^H (AA^H)^{-1}}{1 - a_k (AA^H)^{-1} a_k}. \tag{8}$$

Writing $A^\dagger$ as

$$\begin{aligned} A^\dagger &= \Pi \begin{bmatrix} A_{-k}^H \\ a_k^H \end{bmatrix} (AA^H)^{-1} \\ &= \Pi \begin{bmatrix} G_k^H \\ g_k^H \end{bmatrix} \end{aligned} \tag{9}$$

it is easily seen that the matrices G_k and g_k are obtained from a permutation of $A^\dagger$ and then we obtain

$$\begin{aligned} A_{-k}^\dagger &= G_k^H + \left(\frac{1}{1 - g_k^H a_k}\right) G_k^H a_k g_k^H \\ &= G_k^H \left(I + \frac{1}{1 - g_k^H a_k} a_k g_k^H\right). \end{aligned} \tag{10}$$

Corresponding to the deletion of the column, a_k, the solution vector is updated as

$$\begin{aligned} x_1^{(k)} &= A_{-k}^\dagger b = G_k^H \left(I + \frac{1}{1 - g_k^H a_k} a_k g_k^H\right) b \\ &= G_k^H b + \frac{1}{1 - g_k^H a_k} G_k^H a_k (g_k^H b) \end{aligned} \tag{11}$$

where $g_k^H b$ is the component in $x_0 = A^\dagger b$ associated with the deleted column a_k and $G_k^H b \in \mathbb{C}^{n-1}$ is obtained as a permutation of the other $(n-1)$ components from x_0. Note that the computation of $G_k^H a_k$ is common to both (10) and (11). Letting A_{-k} become the new dictionary A, the same iteration may be repeated to remove another vector from the dictionary.

In [1], the choice of which column to delete from A was based on minimizing the representation error, i.e., $k = \arg\min_j \|b - A_{-j} x_1^{(j)}\|$. However, this error will, in general, be zero for all values of j as the dictionary considered here is overcomplete. Therefore, we must consider other criteria in choosing the column to delete from A to form A_{-k} and some possibilities are outlined in the following section.

III. Criteria for Column Deletion

In each iteration of the algorithm, the solution is calculated using (11). As previously mentioned, the perturbation of the solution lies in $\mathcal{N}(A)$, i.e.,

$$\delta x_{l+1} = x'_{l+1} - x'_l \in \mathcal{N}(A), \quad l = 0, 1, 2, \ldots \tag{12}$$

where $x'_l, x'_{l+1} \in \mathbb{C}^n$, with the entries corresponding to the remaining columns obtained from (11) and zeros inserted for the deleted dictionary vectors; x'_0 is initialized as $x_0 = A^\dagger b$. Each of the vectors, $x'_l, l = 0, 1, \ldots, (n-m)$, will satisfy the equality given in (1). The change in the solution vector from its initialization as x_0 is

$$\Delta x_{l+1} = x'_{l+1} - x_0 = \sum_{j=1}^{l+1} \delta x_j \in \mathcal{N}(A), \quad l = 0, 1, 2, \ldots \tag{13}$$

and the magnitude of the solution in each iteration is obtained as

$$\|x_{l+1}\|^2 = \|x_0\|^2 + \|\Delta x_{l+1}\|^2.$$

The selection of the column for deletion from A may be made based on the magnitude of $\|\delta x_{l+1}\|$ or $\|\Delta x_{l+1}\|$ among many other possible choices. However, it was found in [2] that the best results were obtained by deleting the column in each iteration which resulted in a minimization of the $l_{p\leq 1}$ p-norm-like diversity measure of the solution. This diversity measure is known

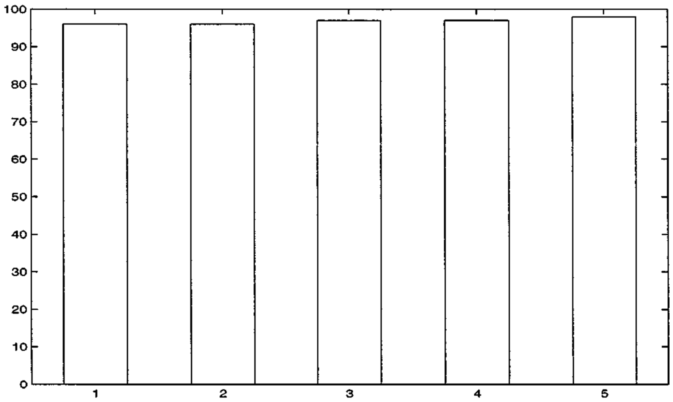

Fig. 1. Comparison of forward selection algorithms and the backward elimination algorithm. Three p-norms are used as different criteria for the backward elimination algorithm and the columns represent respectively MMP, ORMP, $p = 0.8, 0.9, 1.0$.

to promote sparse solutions [6]. The column a_k to be deleted is therefore obtained as

$$k = \arg\min_j \left\| x_l^{(j)} \right\|_p \quad \text{where}$$
$$\left\| x_l^{(j)} \right\|_p = \sum_{i=1}^{n-l} \left| x_l^{(j)}(i) \right|^p, \quad 0 \le p \le 1. \tag{14}$$

IV. Simulation

The dictionary A is created as a random $m \times n$ matrix A whose entries are Gaussian random variables with mean 0 and variance 1. Each column is then normalized. A sparse solution, x_s, with a specified number of nonzero entries r is created; the indices of these r entries are randomly generated using a uniform distribution, and their amplitudes are Gaussian random variables. The vector b is computed as $b = Ax_s$ so that the solution is known and b is then normalized.

The backward elimination algorithm must then remove $(n - m)$ columns from the dictionary without removing one of the columns used in forming the signal, b. If the algorithm is successful in doing this, the solution is found uniquely using the remaining dictionary vectors. For the purposes of this experiment we chose the dimensions of A as 20×30 and the value of r as 4 (other dimensions and values for r were experimented with and yielded similar results). 100 trials were performed with the matrix A, nonzero entries and amplitudes in x_s randomly created in each trial. A success is where the four vectors used in generating our solution have nonzero weights associated with them in the solution vector, x. Therefore, our measure of success is that of a component detection problem.

In Fig. 1, we plot the success rate for the backward elimination algorithm where the p-norm-like diversity measure, with different values of p, is used as the deletion criterion in each iteration. In addition, we show the success rate for the two forward selection algorithms, MMP and ORMP [5]. It is seen that the backward elimination algorithm performs slightly better than the forward selection techniques in this experiment. Additional simulation results can be found in [2].

V. Conclusion

We have shown how the backward elimination algorithm for determining sparse signal representations presented in [1] can be extended to applications in which we have an overcomplete dictionary. Criteria other than minimizing the representation error must be used in deciding upon the dictionary element to be eliminated in each iteration [2]. A test-case dictionary was used to show the success of the algorithm and how it may improve upon results obtained using forward selection methods. The results of the current paper in combination with those of [1] provide a complete solution to the backward sequential elimination problem for the cases of overcomplete and undercomplete dictionaries.

References

[1] S. Reeves, "An efficient implementation of the backward greedy algorithm for sparse signal reconstruction," *IEEE Signal Processing Lett.*, vol. 6, pp. 266–268, Oct. 1999.
[2] S. F. Cotter, K. Kreutz-Delgado, and B. D. Rao, "Backward sequential elimination for sparse vector subset selection," *Signal Process.*, vol. 81, pp. 1849–1864, Sept. 2001.
[3] B. D. Rao, "Signal processing with the sparseness constraint," in *Proc. ICASSP*, vol. III, Seattle, WA, May 1998, pp. 1861–1864.
[4] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.
[5] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado, "Forward sequential algorithms for best basis selection," *Proc. Inst. Elect. Eng.*, vol. 146, no. 5, pp. 235–244, Oct. 1999.
[6] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Processing*, vol. 47, pp. 187–200, Jan. 1999.
[7] C. Couvreur and Y. Bresler, "On the optimality of the backward greedy algorithm for the subset selection problem," *SIAM J. Matrix Anal. Applicat.*, vol. 21, no. 3, pp. 797–808, 1999.
[8] S. L. Campbell and C. D. Meyer, *Generalized Inverses of Linear Transformations*. New York: Dover, 1991.
[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: John Hopkins Univ. Press, 1989.