

## SPARSE APPROXIMATE SOLUTIONS TO LINEAR SYSTEMS\*

B. K. NATARAJAN†

**Abstract.** The following problem is considered: given a matrix  $A$  in  $\mathbf{R}^{m \times n}$ , ( $m$  rows and  $n$  columns), a vector  $b$  in  $\mathbf{R}^m$ , and  $\epsilon > 0$ , compute a vector  $x$  satisfying  $\|Ax - b\|_2 \leq \epsilon$  if such exists, such that  $x$  has the fewest number of non-zero entries over all such vectors. It is shown that the problem is NP-hard, but that the well-known greedy heuristic is good in that it computes a solution with at most  $\lceil 18 \text{Opt}(\epsilon/2) \|A^+\|_2^2 \ln(\|b\|_2/\epsilon) \rceil$  non-zero entries, where  $\text{Opt}(\epsilon/2)$  is the optimum number of nonzero entries at error  $\epsilon/2$ ,  $A^+$  is the matrix obtained by normalizing each column of  $A$  with respect to the  $L_2$  norm, and  $A^+$  is its pseudo-inverse.

**Key words.** sparse solutions, linear systems

**AMS subject classification.** 68Q25

**1. Introduction.** The problem of computing a sparse approximate solution to a linear system is a fundamental problem in matrix computation. For matrices over the reals, the problem (variant thereof) has been studied under the name “subset selection” in statistical modeling by Golub and Van Loan (1983). For binary matrices, the problem has been studied as “minimum weight solution” in error corrective coding by Gallager (1968). The related “sparse null-space” problem is of interest in nonlinear optimization; see Coleman and Pothén (1986). Also, the “minimum set cover” problem can be viewed as computing a sparse solution to  $Ax \geq b$ , where all the entries are binary and the inequality is entry-wise; see Garey and Johnson (1979), Johnson (1974).

Our specific motivation for studying the problem is as follows. A widely used technique for the interpolation of irregularly spaced samples in higher dimensions is that of radial basis interpolation; see Hardy (1988). In brief, assign each of the given sample points a coefficient. The value of the interpolant at a new point is the weighted sum of these coefficients, where the weights are the distances of the respective samples from the new point. The coefficients themselves are the solution to the linear system of equations obtained by evaluating the interpolant at each of the given points and equating them to the respective sample values; Micchelli (1986) has shown that the linear system is always nonsingular.

In solving the above linear system, two issues are of interest. (1) The cost of evaluating the interpolant at a general point grows with the number of nonzero coefficients in the solution. Hence, it is desirable to have as few nonzeros as possible, while tolerating some error in the interpolation. (2) By the principle of Occam’s Razor, the interpolant with the fewest nonzero coefficients is most likely to approximate the underlying function that generated the samples. Indeed, in the presence of noise in the data, it is desirable to pick a sparse but approximate interpolant. Recently, learning theoretic justifications of Occam’s Razor in this setting have been established; see Natarajan (1993).

In light of the above, we are interested in obtaining a sparse approximate solution to the linear system specifying the coefficients of the interpolant. Similar arguments can also be made with respect to other interpolation methods, such as polynomial or trigonometric interpolation.

Previously, it was known that the problem is NP-hard if the solution is required to be over the integers; see Garey and Johnson (1979). For matrices over the reals, the heuristic of Golub, Klema, and Stewart (1976) is well known, as discussed in Golub and Van Loan (1983). In this paper, we show that the problem remains NP-hard over the reals. We then show that the obvious and well-known greedy heuristic (Golub and Van Loan (1983)) is provably good in that

\*Received by the editors November 2, 1992; accepted for publication October 29, 1993.

†Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, California 94304.

the number of nonzeros reported is at most  $18 \text{Opt}(\epsilon/2) \|\mathbf{A}^+\|_2^2 \ln(\|b\|_2/\epsilon)$ , where  $\text{Opt}(\epsilon/2)$  is the optimum number of nonzero entries at error  $\epsilon/2$ ,  $\mathbf{A}$  is the matrix obtained by normalizing each column of  $A$  with respect to the  $L_2$  norm, and  $\mathbf{A}^+$  is its pseudo-inverse. The algorithm is essentially a marriage between the greedy set cover algorithm of Johnson (1974) and the QR algorithm for least squares solutions of linear systems of Golub and Van Loan (1983). It is important to note that minimizing the number of zeros is competitive with the stability of the algorithm, i.e., the norm of the computed solution. The heuristic of Golub, Klema, and Stewart (1976) offers stability, but not provably good sparseness, while our algorithm does the converse. An open problem is to obtain an algorithm that allows an explicit tradeoff.

The algorithm consumes  $O(mn)$  time per entry reported. Since the entries of the solution are reported in order of their “significance,” the proposed algorithm is a good candidate for the progressive solution of dense linear systems. Computational experiments on using the greedy algorithm for radial basis interpolation are discussed in Carlson and Natarajan (1994).

**2. Hardness.** Let SAS refer to the sparse approximate solution problem stated below.

*Problem.* Given a matrix  $A$  in  $\mathbf{R}^{m \times n}$ , ( $m$  rows and  $n$  columns), a vector  $b$  in  $\mathbf{R}^m$ , and  $\epsilon > 0$ , compute a vector  $x$  satisfying  $\|Ax - b\|_2 \leq \epsilon$  if such exists, such that  $x$  has the fewest number of nonzero entries over all such vectors.

We first establish that SAS is NP-hard on the machine model of the infinite precision RAM; see Preparata and Shamos (1985), for instance.

**THEOREM 1.** SAS is NP-hard.

*Proof.* The proof is by reduction from the problem of “exact cover by 3 sets,” as in the proof of hardness for the problem of “minimum weight solutions to linear systems.” See Garey and Johnson (1979), pp. 221 and 246).

*Exact Cover by 3-sets.*

*Instance.* A set  $S$ , and a collection  $C$  of 3-element subsets of  $S$ .

*Question.* Does  $C$  contain an exact cover for  $S$ , i.e., a sub-collection  $\hat{C}$  of  $C$  such that every element of  $S$  occurs exactly once in  $\hat{C}$ ?

Let us use X3C to refer to the problem of exact cover by 3-sets, and show how to transform an instance of X3C to an instance of SAS. Given is an instance of X3C:  $S = \{s_1, s_2, \dots, s_m\}$ , and  $C = c_1, c_2, \dots, c_n$ . Without loss of generality we can assume that  $m$  is a multiple of 3, since otherwise there is trivially no exact cover. Let  $b$  be the vector  $(1, 1, 1, \dots, 1)$  of  $m$  ones. The matrix  $A$  will have  $n$  column vectors, one for each set in  $C$ . Specifically, for each set  $c \in C$ , the corresponding column vector will have entries  $(z_1, z_2, \dots, z_m)$  where  $z_i = 1$  if  $s_i \in c$  and  $z_i = 0$  otherwise. Pick  $\epsilon = \frac{1}{2}$ .

We show that the constructed instance of SAS has a solution with  $m/3$  or fewer entries if and only if the given instance of X3C has a solution. If the X3C instance has a solution, then consider the vector  $x = (x_1, x_2, \dots, x_n)$ , where  $x_i = 1$  if the  $i$ th set in  $C$  is included in the solution of the X3C instance, and  $x_i = 0$  otherwise. Then,  $Ax = b$  exactly and hence the SAS instance has a solution with exactly  $m/3$  nonzero entries. Conversely, assume the SAS instance has a solution  $x$  with at most  $m/3$  nonzero entries.

Since  $\|Ax - b\|_2 \leq \frac{1}{2}$ , each entry of  $Ax$  must be between  $\frac{1}{2}$  and  $\frac{3}{2}$ . Since each column of  $A$  has only three nonzero entries,  $x$  must have at least  $m/3$  entries. Thus  $x$  has exactly  $m/3$  entries. Now consider the subcollection  $\hat{C}$ , consisting of those sets  $c_i$  such that the  $i$ th entry in  $x$  is nonzero. It is clear that  $\hat{C}$  is an exact cover for  $S$ .  $\square$

**3. The algorithm.** The algorithm we consider is a merging of the greedy set cover algorithm of Johnson (1974) and the QR algorithm for the least-squares problem of Golub and Van Loan (1983). In essence, it is a QR algorithm where the column pivot is chosen greedily with respect to the right-hand side  $b$  of the linear system. Such a modification is well known; see

Golub and Van Loan (1983, p. 168, Exercise 6.4.8). In the interest of simplicity, we will not present the algorithm as a modification of the QR algorithm, but will present it as a two-phase algorithm with a subset selection phase, followed by an explicit solution phase.

In words, Algorithm Greedy below takes as input  $A$ ,  $b$ , and  $\epsilon$ . The algorithm first normalizes each column  $a_i$  of  $A$ . Then, at each iteration of the selection phase, the algorithm greedily picks that column of  $A$  that is closest in angle to the vector  $b$ . Then,  $b$  and the column vectors of  $A$  are projected onto the subspace orthogonal to the chosen column. This procedure is repeated until  $\|b\|_2 \leq \epsilon$ . In the solution phase, the algorithm solves the linear system problem  $Bx = b^{(0)} - b^{(r)}$  where  $B$  is the matrix consisting of those columns of  $A$  that were chosen in the selection phase,  $b^{(0)}$  is  $b$  at the start of the selection of phase, and  $b^{(r)}$  is  $b$  at the end of the selection of phase.

### Algorithm Greedy

#### input:

matrix  $A$ , column vector  $b$ ,  $\epsilon > 0$ .

#### Subset Selection Phase:

Normalize each column of  $A$  to obtain  $\mathbf{A}$ .

$r \leftarrow 0$ ;  $\tau \leftarrow \emptyset$ ;

$A^{(0)} \leftarrow \mathbf{A}$ ;  $b^{(0)} \leftarrow b$ ;

**while**  $\|b^{(r)}\|_2 > \epsilon$  **do**

**if**  $A^{(r)T} b^{(r)} = 0$  **then**

        no solution exists;

**else**

        choose  $k \in \{1, 2, \dots, n\} - \tau$  such that column  $a_k^{(r)}$  of  $A^{(r)}$  is closest to  $b^{(r)}$ , i.e.,  $|a_k^{(r)T} b^{(r)}|$  is maximum;

        project  $b^{(r)}$  onto the subspace orthogonal to  $a_k^{(r)}$ , i.e.,

$b^{(r+1)} \leftarrow b^{(r)} - (a_k^{(r)T} b^{(r)}) a_k^{(r)}$ ;

$\tau \leftarrow \tau \cup \{k\}$ ;

**for**  $j \in \{1, 2, \dots, n\} - \tau$  **do**

        project  $a_j^{(r)}$  onto the subspace orthogonal to  $a_k^{(r)}$ , i.e.,

$a_j^{(r+1)} \leftarrow a_j^{(r)} - (a_k^{(r)T} a_j^{(r)}) a_k^{(r)}$ ;

        normalize  $a_j^{(r+1)}$ , i.e.,

$a_j^{(r+1)} \leftarrow a_j^{(r+1)} / \|a_j^{(r+1)}\|_2$ ;

**end**

**end**

$r \leftarrow r + 1$ ;

**end**

#### Solution Phase:

Solve  $Bx = b^{(0)} - b^{(r)}$  where  $B$  is the matrix of columns of  $A$  with indices in  $\tau$ .

**end**

**THEOREM 2.** *The number of vectors selected by Algorithm Greedy, and hence the number of nonzero entries in the computed solution, is at most*

$$(1) \quad \left\lceil 18 \text{Opt}(\epsilon/2) \|A^+\|_2^2 \ln \left( \frac{\|b\|_2}{\epsilon} \right) \right\rceil,$$

where  $\text{Opt}(\epsilon/2)$  denotes the fewest number of nonzero entries over all solutions that satisfy  $\|Ax - b\|_2 \leq \epsilon/2$ .

Our proof of Theorem 2 is a simplified and generalized variant of that of the greedy set cover algorithm given by Johnson (1974). For each value of  $r$ , at the start of the corresponding iteration of the algorithm, let  $u^{(r)}$  be a vector with the minimum number of nonzero entries such that  $\|A^{(r)}u^{(r)} - b^{(r)}\|_2 \leq \epsilon/2$ . Also, we define  $N^{(r)}$  as the number of nonzero entries in  $u^{(r)}$  and  $q^{(r)} = A^{(r)}u^{(r)}$ . Let  $t$  be the number of iterations the algorithm runs in the selection phase. It is clear that the number of nonzero entries produced by the solution phase of the algorithm is at most  $t$ . Define

$$(2) \quad \rho = 4 \max_{0 \leq r < t} \frac{N^{(r)} \|u^{(r)}\|_2^2}{\|b^{(r)}\|_2^2}.$$

Our proof proceeds in a sequence of lemmas. In the first lemma, we bound  $t$  in terms of  $\rho$ . Lemmas 2 and 3 estimate the right-hand side of the above definition of  $\rho$ . Combining the three lemmas, we will obtain Theorem 2.

For  $z \in \mathbf{R}$ , let  $\lceil z \rceil$  denote the smallest integer greater than or equal to  $z$ .

LEMMA 1. *The number of iterations of the algorithm  $t$  is at most*

$$(3) \quad \left\lceil 2\rho \ln \left( \frac{\|b\|_2}{\epsilon} \right) \right\rceil.$$

*Proof.* Since  $u^{(r)}$  only has  $N^{(r)}$  nonzero entries, we will show that at least one of the columns of  $A^{(r)}$  must be correspondingly close to  $b^{(r)}$ . Specifically, we establish a lower bound on  $\|A^{(r)T}b^{(r)}\|_\infty$ . Now,

$$(4) \quad b^{(r)} = \sum_{i=1}^n u_i^{(r)} a_i^{(r)} + \epsilon \tilde{2},$$

where  $\epsilon \tilde{2}$  stands for the generic error vector of  $L_2$  norm at most  $\epsilon/2$ . Hence

$$(5) \quad b^{(r)T}b^{(r)} = \sum_{i=1}^n u_i^{(r)} b^{(r)T} a_i^{(r)} + b^{(r)T} \epsilon \tilde{2}.$$

$$(6) \quad \leq \left( \max_{i=1}^n |a_i^{(r)T} b^{(r)}| \right) \sum_{i=1}^n |u_i^{(r)}| + \|b^{(r)}\|_2 \epsilon/2.$$

Now,

$$(7) \quad b^{(r)T}b^{(r)} = \|b^{(r)}\|_2^2,$$

$$(8) \quad \max_{i=1}^n |a_i^{(r)T} b^{(r)}| = \|A^{(r)T}b^{(r)}\|_\infty,$$

and

$$(9) \quad \|b^{(r)}\|_2 \geq \epsilon.$$

Also, since  $u^{(r)}$  has  $N^{(r)}$  nonzero entries,

$$(10) \quad \sum_{i=1}^n |u_i^{(r)}| = \|u^{(r)}\|_1 \leq \sqrt{N^{(r)}} \|u^{(r)}\|_2.$$

Hence, we can write (6) as

$$(11) \quad \|b^{(r)}\|_2^2 \leq \|A^{(r)T}b^{(r)}\|_\infty \sqrt{N^{(r)}} \|u^{(r)}\|_2 + 1/2 \|b^{(r)}\|_2^2.$$

Rearranging, we get

$$(12) \quad \|A^{(r)T} b^{(r)}\|_{\infty} \geq \frac{\|b^{(r)}\|_2^2}{2\sqrt{N^{(r)}}\|u^{(r)}\|_2}.$$

Using the above lower bound on  $\|A^{(r)T} b^{(r)}\|_{\infty}$  we now estimate the rate at which  $b^{(r)}$  diminishes at each iteration of the algorithm. Now,

$$(13) \quad b^{(r+1)} = b^{(r)} - \left(a_k^{(r)T} b^{(r)}\right) a_k^{(r)},$$

where  $a_k^{(r)}$  is the column chosen by the algorithm at the  $r$ th iteration. Since  $|a_k^{(r)T} b^{(r)}|$  is a maximum,  $|a_k^{(r)T} b^{(r)}| = \|A^{(r)T} b^{(r)}\|_{\infty}$  and we can rewrite the above as

$$(14) \quad b^{(r+1)} = b^{(r)} - \text{sign}\left(a_k^{(r)T} b^{(r)}\right) \|A^{(r)T} b^{(r)}\|_{\infty} a_k^{(r)},$$

where  $\text{sign}(\cdot)$  is the function that returns the sign of its argument. Since  $b^{(r+1)}$  is orthogonal to  $a_k^{(r)}$ ,

$$(15) \quad \|b^{(r+1)}\|_2^2 = \|b^{(r)}\|_2^2 - \|A^{(r)T} b^{(r)}\|_{\infty}^2.$$

Rearranging, we get

$$(16) \quad \|b^{(r+1)}\|_2^2 = \left(1 - \frac{\|A^{(r)T} b^{(r)}\|_{\infty}^2}{\|b^{(r)}\|_2^2}\right) \|b^{(r)}\|_2^2.$$

Substituting (12) in (16), we get

$$(17) \quad \|b^{(r+1)}\|_2^2 \leq \left(1 - \frac{\|b^{(r)}\|_2^2}{4N^{(r)}\|u^{(r)}\|_2^2}\right) \|b^{(r)}\|_2^2.$$

Using (2) in (17), we get

$$(18) \quad \|b^{(r+1)}\|_2^2 \leq (1 - 1/\rho) \|b^{(r)}\|_2^2.$$

Hence,

$$(19) \quad \|b^{(t)}\|_2^2 \leq (1 - 1/\rho)^t \|b^{(0)}\|_2^2.$$

The algorithm halts when  $\|b^{(t)}\|_2^2 \leq \epsilon^2$ . For this to happen, it suffices that  $t$  satisfy

$$(20) \quad (1 - 1/\rho)^t \|b^{(0)}\|_2^2 \leq \epsilon^2.$$

Rearranging and taking logarithms, we get

$$(21) \quad t \ln(1 - 1/\rho) \leq \ln\left(\frac{\epsilon^2}{\|b^{(0)}\|_2^2}\right).$$

Incorporating in (21) the following inequality, obtained from a Taylor expansion of the natural logarithm function,

$$(22) \quad \ln(1 - 1/\rho) \leq \frac{-1}{\rho},$$

we get that it suffices if

$$(23) \quad t \geq \rho \ln \left( \frac{\|b^{(0)}\|_2^2}{\epsilon^2} \right) = 2\rho \ln \left( \frac{\|b\|_2}{\epsilon} \right).$$

This completes the proof of the lemma.  $\square$

We will now establish an upper bound on  $\rho$ , by first bounding  $\|u^{(r)}\|_2$  and then incorporating a bound on  $N^{(r)}$ .

**LEMMA 2.** For  $0 \leq r < t$ ,  $\|u^{(r)}\|_2 \leq 3/2\|\mathbf{A}^+\|_2\|b^{(r)}\|_2$ .

*Proof.* Let  $\sigma = \{i \mid u_i^{(r)} \neq 0\}$  be the set of indices of the nonzero entries in  $u^{(r)}$ , and let  $\tau = \{k_1, k_2, \dots, k_r\}$  be the indices of the first  $r$  columns picked by the algorithm. For matrix  $A$  and set of indices  $\mu$ , we use  $\mu(A)$  to denote the set of column vectors  $\{a_i \mid i \in \mu\}$ .

We first show that  $\sigma(A^{(0)}) \cup \tau(A^{(0)})$  is a linearly independent set. Assume the contrary, i.e., some linear combination of vectors from  $\sigma(A^{(0)}) \cup \tau(A^{(0)})$  sum to zero. Not all of the vectors in this combination can be from  $\tau(A^{(0)})$ , since  $\tau(A^{(0)})$  is a linearly independent set by the definition of the algorithm. So it must be that at least one of the vectors in this combination is from  $\sigma(A^{(0)})$ . It follows that this vector from  $\sigma(A^{(0)})$  can be expressed as a linear combination of vectors from  $\sigma(A^{(0)}) \cup \tau(A^{(0)})$ . Since  $\sigma(A^{(r)})$  is the projection of  $\sigma(A^{(0)})$  on the subspace orthogonal to  $\tau(A^{(0)})$ , the above supposition implies that some vector in  $\sigma(A^{(r)})$  can be expressed as a linear combination of vectors in  $\sigma(A^{(r)})$ . But this contradicts the assumption that  $u^{(r)}$  has a minimum number of nonzero entries. Hence  $\sigma(A^{(0)}) \cup \tau(A^{(0)})$  must be linearly independent.

We can now establish a bound on  $\|u^{(r)}\|_2$ . By definition,

$$(24) \quad q^{(r)} = \sum_{i \in \sigma} u_i^{(r)} a_i^{(r)}.$$

For  $i \in \sigma$ , we can write

$$(25) \quad a_i^{(r)} = \frac{a_i^{(r-1)} - v_i^{(r)}}{\sqrt{1 - \|v_i^{(r)}\|_2^2}},$$

where  $v_i^{(r)}$  is a vector in the span of  $\tau(A^{(r-1)})$ . Using this recurrence, we can express  $a_i^{(r-1)}$  in terms of  $a_i^{(r-2)}$  and so on down to  $a_i^{(0)}$ . Since  $\tau(A^{(r)})$  is orthogonal to  $\tau(A^{(r-1)})$ , it follows that  $v_i^{(r)}$  is orthogonal to  $v_i^{(r-1)}$  and hence  $\|v_i^{(r)} + v_i^{(r-1)}\|_2^2 = \|v_i^{(r)}\|_2^2 + \|v_i^{(r-1)}\|_2^2$ . Using this, we can combine the recurrences (25) to obtain a single expression of the form

$$(26) \quad a_i^{(r)} = \frac{a_i^{(0)} - v_i}{\sqrt{1 - \|v_i\|_2^2}},$$

where  $v_i$  is a vector in the span of  $\tau(A^{(0)})$ .

Substituting (26) in (24), we get

$$(27) \quad q^{(r)} = \sum_{i \in \sigma} u_i^{(r)} \frac{a_i^{(0)} - v_i}{\sqrt{1 - \|v_i\|_2^2}}.$$

Noting that  $v_i$  is a vector in the span of  $\tau(A^{(0)})$  and hence can be expressed as a linear combination of the columns of  $\tau(A^{(0)})$ , we can rewrite (27) as

$$(28) \quad q^{(r)} = \sum_{i \in \sigma} \frac{u_i^{(r)}}{\sqrt{1 - \|v_i\|_2^2}} a_i^{(0)} + \sum_{i \in \tau} \delta_i a_i^{(0)},$$

for some  $\delta_i$  in  $\mathbf{R}$ . Let  $Z$  be the matrix with columns  $\sigma(A^{(0)}) \cup \tau(A^{(0)})$ , and  $Z^+$  its pseudo-inverse. Also, let  $w = Z^+ q^{(r)}$ . As established above, the columns of  $Z$  are linearly independent, and hence the entries of  $w$  are uniquely determined as in (28) above. Specifically, for  $i \in \sigma$ ,

$$(29) \quad w_i = \frac{u_i^{(r)}}{\sqrt{1 - \|v_i\|_2^2}},$$

and for  $i \in \tau$ ,  $w_i = \delta_i$ . Since  $\sqrt{1 - \|v_i\|_2^2} < 1$ , we have that for  $i \in \sigma$ ,  $|u_i^{(r)}| \leq |w_i|$ . For  $i \in \tau$ ,  $w_i = \delta_i$ ,  $u_i^{(r)} = 0$ , and hence  $|u_i^{(r)}| \leq |w_i|$ . We therefore have

$$(30) \quad \|u^{(r)}\|_2 \leq \|w\|_2 \leq \|Z^+\|_2 \|q^{(r)}\|_2.$$

Noting that  $q^{(r)} = b^{(r)} + \epsilon \vec{1}/2$ , and that since  $r < t$ ,  $\|b^{(r)}\|_2 > \epsilon$ , we can simplify the above as

$$(31) \quad \|u^{(r)}\|_2 \leq \|Z^+\|_2 \|b^{(r)} + \epsilon \vec{1}/2\|_2 \leq 3/2 \|Z^+\|_2 \|b^{(r)}\|_2.$$

Since the columns of  $Z$  are a linearly independent subset of the columns of  $A^{(0)} = \mathbf{A}$ , the smallest nonzero singular value of  $Z$  is greater than or equal to the smallest nonzero singular value of  $\mathbf{A}$ ; see Golub and Van Loan (1983, p. 286). Recall that  $\mathbf{A}^+$  is the pseudo-inverse of  $\mathbf{A}$ . Since  $\|Z^+\|_2$  and  $\|\mathbf{A}^+\|_2$  are the reciprocals of the smallest nonzero singular values of  $Z$  and  $\mathbf{A}$ , respectively,  $\|Z^+\|_2 \leq \|\mathbf{A}^+\|_2$ , and the proof of the lemma is complete.  $\square$

**LEMMA 3.** *The number of entries in the sparse solution is nonincreasing as the algorithm iterates, i.e.,  $N^{(r+1)} \leq N^{(r)} \leq N^{(0)}$ .*

*Proof.* As before let  $\sigma = \{i \mid u_i^{(r)} \neq 0\}$  be the set of indices of the nonzero entries in  $u^{(r)}$ . By definition,  $|\sigma| = N^{(r)}$ . Suppose that the algorithm selects a vector from  $\sigma$  at the  $r$ th iteration. Then surely  $u^{(r+1)}$  has one fewer nonzero entry than  $u^{(r)}$ , namely the entry corresponding to the selected vector. If the vector selected by the algorithm is not from  $\sigma$ , then  $\sigma$  remains the set of nonzero entries of  $u^{(r+1)}$ . In either case  $N^{(r+1)} \leq N^{(r)}$  and the lemma follows.  $\square$

Noting that  $N^{(0)} = \text{Opt}(\epsilon/2)$  we can use Lemma 3 to rewrite (2) as

$$(32) \quad \rho \leq 4 \max_{0 \leq r < t} \frac{N^{(0)} \|u^{(r)}\|_2^2}{\|b^{(r)}\|_2^2}.$$

Using Lemma 2 to substitute for  $\|u^{(r)}\|_2$  in (32) gives us

$$(33) \quad \rho \leq 9 \max_{0 \leq r < t} N^{(0)} \|\mathbf{A}^{(+)}\|_2^2.$$

Substituting (33) in the statement of Lemma 1, we get that the number of zeros in the computed solution is at most

$$(34) \quad \left\lceil 18 \text{Opt}(\epsilon/2) \|\mathbf{A}^+\|_2^2 \ln \left( \frac{\|b\|_2}{\epsilon} \right) \right\rceil.$$

This concludes the proof of Theorem 1.

**Acknowledgments.** Thanks to C. Bischof, R. Carlson, T. Coleman, J. Gilbert, D. S. Johnson, and J. Ruppert for the discussions, to A. Frieze for helping clear an error in an earlier version of the paper, and to the anonymous referees for their comments.

## REFERENCES

- R. E. CARLSON AND B. K. NATARAJAN (1994), *Sparse approximate multiquadric interpolation*, Comp. and Math. Appl., Vol. 27 (6), pp. 99–108.
- T. COLEMAN AND A. POTHEIN (1986), *The null space problem I. Complexity*, SIAM J. Alg. Disc. Meth., Vol. 7, pp. 527–537.
- R. G. GALLAGER (1968), *Information Theory and Reliable Communication*, John Wiley, New York, NY.
- M. R. GAREY AND D. S. JOHNSON (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York.
- G. H. GOLUB, V. KLEMA, AND, G. W. STEWART (1976), *Rank Degeneracy and Least Squares Problems*, Tech. Report, TR-456, Dept. of Computer Science, University of Maryland, College Park, MD.
- G. H. GOLUB AND C. F. VAN LOAN (1983), *Matrix Computations*, Johns Hopkins Press, Baltimore, MD.
- R. L. HARDY (1990), *Theory and applications of the multi-quadric biharmonic method*, Comput. Math. Appl., Vol. 19, pp. 163–208.
- D. S. JOHNSON (1974), *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., Vol. 9, pp. 256–278.
- C. A. MICHELLI (1986), *Interpolation of scattered data: distance matrices and conditionally positive definite systems*, Constr. Approx., Vol. 2, pp. 11–22.
- B. K. NATARAJAN (1993), *Occam's razor for functions*, Proceedings of the ACM Symposium on Computational Learning Theory, Santa Cruz, CA.
- F. P. PREPARATA AND M. I. SHAMOS (1985), *Computational Geometry, An Introduction*, Springer-Verlag, New York.