Solution of sparse underdetermined systems of linear equations*

Alan George

Esmond Ng

Department of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA

CS-82-39

September, 1982

* Research of the first author was supported in part by the Canadian Natural Sciences and Engineering Research Council under Grant A8111. Research of the second author was supported in part by the Canadian Natural Sciences and Engineering Research Council through a postgraduate scholarship.

Solution of sparse underdetermined systems of linear equations*

Alan George

Esmond Ng

Department of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA

CS-82-39

September, 1982

* Research of the first author was supported in part by the Canadian Natural Sciences and Engineering Research Council under Grant A8111. Research of the second author was supported in part by the Canadian Natural Sciences and Engineering Research Council through a postgraduate scholarship.

Solution of sparse underdetermined systems of linear equations*

Alan George

Esmond Ng

Department of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA

CS-82-39

September, 1982

ABSTRACT

In this paper we consider the problem of computing the minimal l_2-solution to a consistent underdetermined linear system $Ax=b$, where A is m by n with $m \leq n$. The method of solution is to reduce A to lower trapezoidal form $[L \;\; O]$ using orthogonal transformations, where L is m by m and lower triangular. The method can be implemented efficiently if the matrix AA^T is sparse. However if A contains some dense columns, AA^T may be unacceptably dense. We present a method for handling these dense columns. The problem of solving a rank-deficient underdetermined system is also considered.

* Research of the first author was supported in part by the Canadian Natural Sciences and Engineering Research Council under Grant A8111. Research of the second author was supported in part by the Canadian Natural Sciences and Engineering Research Council through a postgraduate scholarship.

1. Introduction

Let A be an m by n matrix with $m<n$ and b be an m-vector. Consider the underdetermined system of linear equations

$$Ax = b \quad . \tag{1.1}$$

If A has full row rank, the system is always consistent and has an infinite number of solutions. It can be shown that

$$\bar{x} = A^T(AA^T)^{-1}b \tag{1.2}$$

is the solution which minimizes the l_2-norm (the so-called minimal l_2-solution) [Cline 76]. One way to compute this minimal l_2-solution is as follows. Form the m by m symmetric positive definite matrix $B=AA^T$ and compute its Cholesky decomposition. Then use the Cholesky factor to solve the system $Bw=b$. Finally compute $\bar{x}=A^Tw$. However the condition number(1) of B is the square of that of A. Thus if A is ill-conditioned, the computed solution may be sensitive to rounding errors. Furthermore severe roundoff and/or cancellation may occur when B is computed.

A more stable way of computing the minimal l_2-solution is described in [Saunders 72] and [Paige 73]. In this paper we describe an implementation of this method. The implementation will be efficient if the matrix AA^T is sparse. This condition is usually satisfied when A is sparse. However when A contains some dense columns, the matrix AA^T may be unacceptably dense. One possible way to solve this problem is to withhold these dense columns from the original matrix, thus giving a smaller matrix, say $\bar{A}$, such that $\bar{A}\bar{A}^T$ is sparse. Then the minimal l_2-solution is obtained by a technique which uses the Cholesky factor of $\bar{A}\bar{A}^T$ and the withheld columns.

An outline of this paper is as follows. In Section 2 we describe the method and its implementation. In Section 3 the effect of dense columns will be examined. In Sections 4 and 5 the handling of dense columns in the solution of sparse underdetermined systems is considered. The solution of consistent rank-deficient

(1) The Euclidean norm will be assumed throughout this paper.

underdetermined systems is described in Section 6. The handling of dense rows in sparse least squares problems using similar techniques is considered in Section 7, and some concluding remarks appear in Section 8.

2. A method based on orthogonal reductions

There are a number of stable methods for solving underdetermined systems of linear equations [Cline 76]. One of them is based on an orthogonal decomposition of the coefficient matrix A ([Saunders 72], [Paige 73]). Suppose the m by n matrix A is reduced to lower trapezoidal form using orthogonal transformations. That is,

$$A = \begin{bmatrix} L & O \end{bmatrix} Q \ , \tag{2.1}$$

where Q is an n by n orthogonal matrix and L is an m by m lower triangular matrix. We will call such a decomposition the LQ-decomposition of A. Then the minimal l_2-solution is given by

$$\bar{x} = A^T \left(\begin{bmatrix} L & O \end{bmatrix} Q Q^T \begin{bmatrix} L^T \\ O \end{bmatrix} \right)^{-1} b = A^T (LL^T)^{-1} b \ . \tag{2.2}$$

Note that the condition number of L is the same as that of A since the Euclidean norm is preserved under orthogonal transformations. Thus one would expect that using (2.2), the error in the computed solution should depend on the square of the condition number of A. However Paige showed that if A is not too badly conditioned, the error depends essentially on the condition of A [Paige 73].

Furthermore (2.2) is particularly attractive for large sparse systems because:

(1) Even though the method is based on orthogonal transformations, the orthogonal matrix Q which is large and usually tends to be dense is never needed. The orthogonal transformations can be discarded once they have been used.

(2) In [George 80] George and Heath have described an efficient way of computing the QR-decomposition of a sparse rectangular matrix. Thus the LQ-decomposition of A can be obtained efficiently simply by computing the QR-decomposition of A^T. Note that $LL^T = AA^T$. That is, L is the Cholesky factor of the symmetric positive definite matrix AA^T, except possibly for sign differences in some columns. If AA^T is sparse, it is usually possible to reorder

the rows and columns of AA^T such that L is sparse. Furthermore a ***static*** data structure can be set up for L ***before*** any numerical computation begins. See [George 80] and [George 81] for more details. Another advantage of the method is that it only eliminates one column of A (that is, one row of A^T) at a time. We do not have to store the entire matrix A in main storage. Only the lower triangular matrix L and a few vectors are needed in main storage.

We conclude this section by presenting the complete algorithm for solving a sparse underdetermined system of linear equations. It should be emphasized that the matrix AA^T is assumed to be sparse.

Algorithm 1

(1) Compute the LQ-decomposition of A using the method described in [George 80]. That is, $A = \begin{bmatrix} L & O \end{bmatrix} Q$.

(2) Solve $LL^T w = b$.

(3) Compute $\bar{x} = A^T w$.

Note that the method described here can also be used to solve $Ax = b$, where A is square and nonsingular. In particular since (2.2) does not require the orthogonal matrix Q, (2.2) can be used to solve several systems which have the *same* coefficient matrix.

3. Effect of dense columns

The algorithm we described in the previous section assumes that AA^T is sparse. This is usually true if A is sparse. However if A has a few dense columns, AA^T may be unacceptably dense. An example is given below. Consider

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix} .$$

Then

$$AA^T = \begin{bmatrix} 2 & 2 & 3 \\ 2 & 5 & 6 \\ 3 & 6 & 10 \end{bmatrix} , \quad \text{and} \quad L = \begin{bmatrix} \sqrt{2} & & \\ \sqrt{2} & \sqrt{3} & \\ \frac{3}{\sqrt{2}} & \sqrt{3} & \frac{\sqrt{5}}{\sqrt{2}} \end{bmatrix} .$$

The matrix A is sparse except for the last column which is completely full. Notice that AA^T and L are both dense.

In general if A contains some dense columns, both AA^T and L will be dense. One possible way to preserve sparsity and reduce the computational time is to withhold those dense columns from the LQ-decomposition. Let $\bar{A}$ be the matrix containing the sparse columns of A. In the previous example,

$$\bar{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

Then $\bar{A}$ is reduced to lower trapezoidal form. That is,

$$\bar{A} = [\bar{L} \quad O]\bar{Q} \ ,$$

where $\bar{L}$ is a sparse lower triangular matrix. Now the lower triangular matrix $\bar{L}$ and the previously withheld columns are used to find the minimal l_2-solution of the original system $Ax = b$. In the next two sections we will derive two algorithms for handling the dense columns.

4. Partitioned systems (I)

Let A be an m by $(n+p)$ matrix with $m \le n$. Suppose A is partitioned into $[B \quad C]$, where B is m by n and C is m by p. For example, in large sparse problems, B and C will contain the sparse and dense columns of A respectively. We assume that both A and B have full row rank, and p is small. Furthermore we assume that the LQ-decomposition of B is given by

$$B = \begin{bmatrix} L & O \end{bmatrix} Q \ , \tag{4.1}$$

where Q is an n by n orthogonal matrix, and L is an m by m lower triangular matrix.

Consider the underdetermined system of linear equations $Ax = b$. The minimal l_2-solution is given by (1.2), $\bar{x} = A^T(AA^T)^{-1}b$. Note that

$$AA^T = \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} B^T \\ C^T \end{bmatrix} = BB^T + CC^T = \begin{bmatrix} L & O \end{bmatrix} QQ^T \begin{bmatrix} L^T \\ O \end{bmatrix} + CC^T$$

$$= LL^T + CC^T \quad . \tag{4.2}$$

Thus the minimal l_2-solution is given by

$$\bar{x} = A^T(LL^T + CC^T)^{-1}b \quad . \tag{4.3}$$

Note that both LL^T and $LL^T + CC^T$ are nonsingular.

It is possible to treat (4.3) using the Sherman-Morrison-Woodbury formula which is stated below [Henderson 81].

Lemma 4.1

Let M be an m by m nonsingular matrix, and U and V be m by p matrices. If $M + UV^T$ is nonsingular, then

$$(M + UV^T)^{-1} = M^{-1} - M^{-1}U(I + V^TM^{-1}U)^{-1}V^TM^{-1} \quad .$$

■

Thus applying Lemma 4.1 to (4.3), we have

$$\bar{x} = A^T\{L^{-T}L^{-1} - L^{-T}L^{-1}C(I + C^TL^{-T}L^{-1}C)^{-1}C^TL^{-T}L^{-1}\}b$$

$$= A^T\{L^{-T}L^{-1}b - L^{-T}L^{-1}C(I + C^TL^{-T}L^{-1}C)^{-1}C^TL^{-T}L^{-1}b\} \quad . \tag{4.4}$$

Although this expression appears to be complicated, the computational algorithm resulting from it is quite straight-forward, and is given below.

Algorithm 2

(1) Compute the LQ-decomposition of B. That is, $B = \begin{bmatrix} L & O \end{bmatrix} Q$.

(2) Solve the m by m sparse system $LL^Ty = b$.

(3) Solve p triangular systems $LW = C$.

(4) Form the p by p matrix $D = I + W^TW$.

(5) Solve the p by p dense system $Dz = C^T y$.

(6) Solve the m by m sparse system $LL^T v = Cz$.

(7) Compute $\bar{x} = A^T(y - v)$.

If the method from [George 80] is used to compute the lower triangular matrix L, then Q is discarded and B is stored on secondary storage. Also note that both C and D are small matrices as long as p is small. Thus the amount of main store required is essentially dominated by that required for L. Finally it is easy to see from the algorithm that the cost of the solution process is essentially given by the orthogonal decomposition of B (step 1) and the solution of the m by m triangular systems (steps 2, 3 and 6).

5. Partitioned systems (II)

In Algorithm 2 we assumed that both the matrix A and its sparse submatrix B have full rank. However there are systems in which the sparse portion may be rank-deficient even though the matrix A has full rank. An example is given below. Suppose

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix} .$$

Clearly A has full row rank. Its sparse submatrix is

$$B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} .$$

The rank of B is only 2. Thus BB^T is singular and Algorithm 2 will fail. In this section we show how Algorithm 2 can be modified to handle a rank-deficient sparse submatrix.

Let A be an m by $(n+p)$ matrix with $m \leq n$. Suppose A is partitioned into $[B \quad C]$, where B is m by n and C is m by p. We assume that A has full row rank and B has rank r, where $r < m$. We also assume that both p and $(m-r)$ are small.

Since B is rank-deficient, it is possible to arrange the columns and rows of B so that the LQ-decomposition has the form

$$\begin{bmatrix} L & O \\ S & O \end{bmatrix} Q \ ,$$

where Q is an n by n orthogonal matrix, L is an r by r lower triangular matrix and S is an $(m-r)$ by r matrix. Throughout this section we assume that the columns and rows A have been reordered so that

$$B = \begin{bmatrix} L & O \\ S & O \end{bmatrix} Q \ . \tag{5.1}$$

Consider the underdetermined system of linear equations $Ax = b$. The minimal l_2-solution is given by $\bar{x} = A^T(AA^T)^{-1}b$. Note that

$$AA^T = \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} B^T \\ C^T \end{bmatrix} = BB^T + CC^T = \begin{bmatrix} L & O \\ S & O \end{bmatrix} QQ^T \begin{bmatrix} L^T & S^T \\ O & O \end{bmatrix} + CC^T$$

$$= \begin{bmatrix} L & O \\ S & O \end{bmatrix} \begin{bmatrix} L^T & S^T \\ O & O \end{bmatrix} + CC^T \ . \tag{5.2}$$

Let L_B be an m by m matrix defined by

$$L_B = \begin{bmatrix} L & O \\ S & O \end{bmatrix} \ . \tag{5.3}$$

That is,

$$AA^T = L_B L_B^T + CC^T \ . \tag{5.4}$$

The minimal l_2-solution is therefore given by

$$\bar{x} = A^T(L_B L_B^T + CC^T)^{-1} b \ . \tag{5.5}$$

Note that L_B is singular. Thus Lemma 4.1 cannot be used to handled (5.5) even though $AA^T = L_B L_B^T + CC^T$ is nonsingular.

In order to be able to use Lemma 4.1, we modify the matrix L_B so that it becomes nonsingular. We call such process *rank-promotion*. Define an m by m lower triangular matrix $\bar{L}$ by

$$\bar{L} = \begin{bmatrix} L & O \\ S & I \end{bmatrix} . \tag{5.6}$$

Then

$$\bar{L}\bar{L}^T = \begin{bmatrix} L & O \\ S & I \end{bmatrix} \begin{bmatrix} L^T & S^T \\ O & I \end{bmatrix} = \begin{bmatrix} LL^T & LS^T \\ SL^T & I+SS^T \end{bmatrix} = \begin{bmatrix} LL^T & LS^T \\ SL^T & SS^T \end{bmatrix} + \begin{bmatrix} O & O \\ O & I \end{bmatrix}$$

$$= \begin{bmatrix} L & O \\ S & O \end{bmatrix} \begin{bmatrix} L^T & S^T \\ O & O \end{bmatrix} + \begin{bmatrix} O & O \\ O & I \end{bmatrix} = L_B L_B^T + \begin{bmatrix} O & O \\ O & I \end{bmatrix} . \tag{5.7}$$

Thus we can write AA^T as

$$AA^T = L_B L_B^T + CC^T = \bar{L}\bar{L}^T - \begin{bmatrix} O & O \\ O & I \end{bmatrix} + CC^T . \tag{5.8}$$

Partition C as

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} ,$$

where C_1 is r by p and C_2 is $(m-r)$ by p. Define two m by $(p+m-r)$ matrices U and V by

$$U = \begin{bmatrix} C_1 & O \\ C_2 & -I \end{bmatrix} , \quad \text{and} \quad V = \begin{bmatrix} C_1 & O \\ C_2 & I \end{bmatrix} .$$

Note that

$$UV^T = \begin{bmatrix} C_1 & O \\ C_2 & -I \end{bmatrix} \begin{bmatrix} C_1^T & C_2^T \\ O & I \end{bmatrix} = \begin{bmatrix} C_1C_1^T & C_1C_2^T \\ C_2C_1^T & C_2C_2^T - I \end{bmatrix}$$

$$= \begin{bmatrix} C_1C_1^T & C_1C_2^T \\ C_2C_1^T & C_2C_2^T \end{bmatrix} - \begin{bmatrix} O & O \\ O & I \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \begin{bmatrix} C_1^T & C_2^T \end{bmatrix} - \begin{bmatrix} O & O \\ O & I \end{bmatrix}$$

$$= CC^T - \begin{bmatrix} O & O \\ O & I \end{bmatrix} .$$

Therefore we have

$$AA^T = L_B L_B^T + CC^T = \bar{L}\bar{L}^T + UV^T , \tag{5.9}$$

and the minimal l_2-solution to $Ax=b$ is then given by

$$\bar{x} = A^T(\bar{L}\bar{L}^T + UV^T)^{-1}b \quad . \tag{5.10}$$

The important things to note are that $\bar{L}$ is now nonsingular and the off-diagonal nonzero structure of $\bar{L}$ is identical to that of L_B. Now we can apply Lemma 4.1 to (5.10), whence the minimal l_2-solution is

$$\bar{x} = A^T\{\bar{L}^{-T}\bar{L}^{-1}b - \bar{L}^{-T}\bar{L}^{-1}U(I + V^T\bar{L}^{-T}\bar{L}^{-1}U)^{-1}V^T\bar{L}^{-T}\bar{L}^{-1}b\} \quad . \tag{5.11}$$

We now give the computational algorithm, which is almost identical to Algorithm 2.

Algorithm 3

(1) Compute the LQ-decomposition of B. That is, $B = \begin{bmatrix} L & O \\ S & O \end{bmatrix} Q$. Construct the m by m lower triangular matrix $\bar{L} = \begin{bmatrix} L & O \\ S & I \end{bmatrix}$.

(2) Solve the m by m sparse system $\bar{L}\bar{L}^T y = b$.

(3) Solve $(p+m-r)$ triangular systems $\bar{L}W_1 = U$ and solve $(p+m-r)$ triangular systems $\bar{L}W_2 = V$.

(4) Form the $(p+m-r)$ by $(p+m-r)$ matrix $D = I + W_2^T W_1$.

(5) Solve the $(p+m-r)$ by $(p+m-r)$ dense system $Dz = V^T y$.

(6) Solve the m by m sparse system $\bar{L}\bar{L}^T v = Uz$.

(7) Compute $\bar{x} = A^T(y-v)$.

Note that we only need to compute one of W_1 and W_2. Denote the m by $(m-r)$ matrix $\begin{bmatrix} O \\ I \end{bmatrix}$ by J. Then we have $U = \begin{bmatrix} C & -J \end{bmatrix}$ and $V = \begin{bmatrix} C & J \end{bmatrix}$. Thus

$$W_1 = \bar{L}^{-1}U = \begin{bmatrix} \bar{L}^{-1}C & -\bar{L}^{-1}J \end{bmatrix} \quad , \qquad \text{and} \qquad W_2 = \bar{L}^{-1}V = \begin{bmatrix} \bar{L}^{-1}C & \bar{L}^{-1}J \end{bmatrix} \quad .$$

That is, W_1 and W_2 are the same except that the last $(m-r)$ columns have different signs.

By the same argument, it is easy to see that $W_2^T W_1$ is the same as $W_2^T W_2$ except that the last $(m-r)$ columns have different signs.

Furthermore it is not necessary to reorder the columns and rows of A so as to reduce B to lower trapezoidal form (5.1). Suppose $\hat{L}$ is the lower triangular matrix obtained after applying orthogonal transformations to B. All we need is to identify the columns of $\hat{L}$ which are null and insert ones on the diagonal positions. More details on this technique can be found in [Heath 82].

The final problem to be solved is to determine the rank of B; that is, identifying the null columns of $\hat{L}$. Since finite precision arithmetic is used, it is unlikely that one could find exactly $(m-r)$ null columns. Instead, one is likely to find $(m-r)$ columns which have relatively small elements on the diagonal. Probably the best way to identify the rank is to use a singular value decomposition, but this is a very expensive computation for large sparse problems. A cheap way which is heuristic but usually reliable is to compare the diagonal elements of $\hat{L}$ against some small tolerance. Any diagonal elements which are less than this tolerance in magnitude will be regarded as numerically zero, and the number of such diagonal elements will be assumed to be $(m-r)$. See [Heath 82] for more details on this procedure.

6. Rank-deficient underdetermined systems of linear equations

In this section we show how Algorithm 1 can be adapted to solve a rank-deficient underdetermined system of linear equations. Let A be an m by n matrix with $m<n$. Assume A has rank r, where $r<m$. Suppose the LQ-decomposition of A is given by

$$A = \begin{bmatrix} L & O \\ S & O \end{bmatrix} Q \ , \tag{6.1}$$

where Q is an n by n orthogonal matrix, L is an r by r lower triangular matrix and S is an $(m-r)$ by r matrix. We now show that the minimal l_2-solution to the underdetermined system $Ax=b$ has a form similar to (2.2) provided that the system is consistent.

Let b be partitioned into $\begin{bmatrix} c \\ d \end{bmatrix}$ where c and d are vectors of length r and $(m-r)$ respectively.

First note that the system of linear equations is the same as

$$\begin{bmatrix} L & O \\ S & O \end{bmatrix} Qx = \begin{bmatrix} c \\ d \end{bmatrix} . \tag{6.2}$$

Let Qx be partitioned into $\begin{bmatrix} u \\ v \end{bmatrix}$, where u and v are vectors of length r and $(n-r)$ respectively. Then

$$\begin{bmatrix} L & O \\ S & O \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix} ,$$

or

$$\begin{bmatrix} L \\ S \end{bmatrix} u = \begin{bmatrix} c \\ d \end{bmatrix} . \tag{6.3}$$

That is, $Lu = c$ and $Su = d$. Thus, if the system is consistent, we have

$$SL^{-1}c = d \ . \tag{6.4}$$

Now let w be the solution to the r by r system $LL^T w = c$. Then

$$\bar{x} = A^T \begin{bmatrix} w \\ 0 \end{bmatrix} \tag{6.5}$$

is a solution to the system $Ax = b$, since

$$A\bar{x} = AA^T \begin{bmatrix} w \\ 0 \end{bmatrix} = \begin{bmatrix} L & O \\ S & O \end{bmatrix} QQ^T \begin{bmatrix} L^T & S^T \\ O & O \end{bmatrix} \begin{bmatrix} w \\ 0 \end{bmatrix} = \begin{bmatrix} L & O \\ S & O \end{bmatrix} \begin{bmatrix} L^T w \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} LL^T w \\ SL^T w \end{bmatrix} = \begin{bmatrix} c \\ SL^{-1}c \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix} = b \ .$$

Furthermore suppose $\hat{x}$ is any solution to $Ax = b$. Let $\delta = \bar{x} - \hat{x}$. Then

$$A\delta = A(\bar{x} - \hat{x}) = A\bar{x} - A\hat{x} = 0 \ . \tag{6.6}$$

Note that

$$\|\hat{x}\|_2^2 = \|\bar{x}-\delta\|_2^2 = \|\bar{x}\|_2^2 + \|\delta\|_2^2 - 2\bar{x}^T\delta = \|\bar{x}\|_2^2 + \|\delta\|_2^2 - 2\left(A^T\begin{bmatrix} w \\ 0 \end{bmatrix}\right)^T \delta$$

$$= \|\bar{x}\|_2^2 + \|\delta\|_2^2 - 2[\,w^T\ 0\,]A\delta = \|\bar{x}\|_2^2 + \|\delta\|_2^2 \geq \|\bar{x}\|_2^2 \ .$$

Thus $\bar{x}$ is in fact the minimal l_2-solution.

However in terms of implementation, the method may be inefficient since we have to identify the matrix L from the LQ-decomposition. For large sparse problems, the data structure for storing the lower trapezoidal form is usually complicated. Thus it may be difficult and expensive to extract L from the data structure. To solve this problem, we can use the technique described in Section 5: replace the m by m lower triangular matrix $\begin{bmatrix} L & O \\ S & O \end{bmatrix}$ by $\bar{L} = \begin{bmatrix} L & O \\ S & I \end{bmatrix}$. Then instead of solving $LL^Tw=c$, we do the following. First solve

$$\bar{L}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} L & O \\ S & I \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix} \ . \tag{6.7}$$

Note that $y_1=L^{-1}c$ and $y_2=d-Sy_1=d-SL^{-1}c$. Because the system is assumed to be consistent, $y_2=0$. Next solve

$$\bar{L}^T\begin{bmatrix} w \\ f \end{bmatrix} = \begin{bmatrix} L^T & S^T \\ O & I \end{bmatrix}\begin{bmatrix} w \\ f \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ 0 \end{bmatrix} \ . \tag{6.8}$$

Now $f=0$ and $w=L^{-T}y_1=L^{-T}L^{-1}c$. Finally the minimal l_2-solution is given by

$$\bar{x} = A^T\begin{bmatrix} w \\ f \end{bmatrix} = A^T\begin{bmatrix} w \\ 0 \end{bmatrix} \ . \tag{6.9}$$

Algorithm 4

(1) Compute the LQ-decomposition of A. That is, $A = \begin{bmatrix} L & O \\ S & O \end{bmatrix}Q$. Construct the m by m lower triangular matrix $\bar{L} = \begin{bmatrix} L & O \\ S & I \end{bmatrix}$.

(2) Solve the m by m sparse system $\overline{L}\overline{L}^T z = b$.

(3) Compute $\bar{x} = A^T z$.

7. Use of similar techniques for sparse least squares problems

The techniques we have used for handling dense columns can also be used to handle dense rows in the solution of large sparse least squares problems. Let A be an m by n matrix with $m \ge n$. Consider the problem

$$\min_x \| Ax - b \|_2 \quad . \tag{7.1}$$

It is well known that if A has full column rank, then the unique least squares solution is given by the solution to the normal equations

$$\bar{x} = (A^T A)^{-1} A^T b \quad . \tag{7.2}$$

Note that the form of $\bar{x}$ is similar to that of the minimal l_2-solution to underdetermined systems (see (1.2)).

Suppose the QR-decomposition of A is given by

$$A = Q \begin{bmatrix} R \\ O \end{bmatrix} \quad , \tag{7.3}$$

where Q is an m by m orthogonal matrix and R is an n by n upper triangular matrix. Then the least squares solution is given by

$$\bar{x} = \left\{ \begin{bmatrix} R^T & O \end{bmatrix} Q^T Q \begin{bmatrix} R \\ O \end{bmatrix} \right\}^{-1} A^T b = (R^T R)^{-1} A^T b \quad . \tag{7.4}$$

This leads to the following algorithm for solving least squares problems.

Algorithm 5

(1) Compute the QR-decomposition of A. That is, $A = Q \begin{bmatrix} R \\ O \end{bmatrix}$.

(2) Compute $d = A^T b$.

(3) Solve $R^T R\bar{x} = d$.

Note that the vector d can be computed as either $A^T b$ or $\begin{bmatrix} R & O \end{bmatrix} Q^T b$. In the second alternative, $Q^T b$ can be formed while R is computed. However if several problems which have the same observation matrix A are to be solved, Q is available implicitly only when the first problem is solved. Thus in this case the first alternative may be more useful since d can then be computed using A and b.

When $A^T A$ is sparse, Algorithm 5 can be implemented efficiently by using the algorithm proposed in [George 80] to compute R. However if A contains dense rows, both $A^T A$ and R may be full. This situation is similar to having dense columns in sparse underdetermined systems. In order to reduce the storage required, one can withhold those dense rows in the QR-decomposition. Then the QR-decomposition of the sparse portion and the dense rows are used in a way which is similar to those used in underdetermined systems to find the least squares solution.

We now derive the algorithms for handling dense rows in sparse least squares problems. We assume that A is an $(m+p)$ by n matrix with $m \geq n$. We partition A into

$$A = \begin{bmatrix} B \\ C \end{bmatrix}, \tag{7.5}$$

where B is m by n and C is p by n. Here B and C contain respectively the sparse and dense rows of A. The unique least squares solution is then given by

$$\bar{x} = (A^T A)^{-1} A^T b = (B^T B + C^T C)^{-1} A^T b \quad . \tag{7.6}$$

We first assume that both A and B have full column rank. Suppose the QR-decomposition of B is given by

$$B = Q \begin{bmatrix} R \\ O \end{bmatrix}, \tag{7.7}$$

where Q is an m by m orthogonal matrix and R is an n by n upper triangular matrix. Then (7.6) can be written as

$$\bar{x} = (R^T R + C^T C)^{-1} A^T b \quad . \tag{7.8}$$

Applying Lemma 4.1 yields

$$\bar{x} = \{R^{-1}R^{-T} - R^{-1}R^{-T}C^T(I + CR^{-1}R^{-T}C^T)^{-1}CR^{-1}R^{-T}\}A^T b \quad . \tag{7.9}$$

The resulting computational algorithm is given below.

Algorithm 6

(1) Compute the QR-decomposition of B. That is, $B = Q\begin{bmatrix} R \\ O \end{bmatrix}$.

(2) Compute the vector $d = A^T b$.

(3) Solve the n by n sparse system $R^T R y = d$.

(4) Solve p triangular systems $R^T E = C^T$.

(5) Compute the p by p matrix $D = I + E^T E$.

(6) Solve the p by p dense system $Dz = Cy$.

(7) Solve the n by n sparse system $R^T R v = C^T z$.

(8) Compute the least squares solution $\bar{x} = y - v$.

Now suppose the sparse portion B is rank-deficient, and assume that the rank of B is r, where $r < n$. Let the QR-decomposition of B be

$$B = Q\begin{bmatrix} R & S \\ O & O \end{bmatrix} = QR_B \quad , \tag{7.10}$$

where Q is an m by m orthogonal matrix, R is an r by r upper triangular matrix and S is an r by $(n-r)$ matrix. Then the unique least squares solution which is given by (7.6) becomes

$$\bar{x} = (R_B^T R_B + C^T C)^{-1} A^T b \quad . \tag{7.11}$$

Now $R_B^T R_B$ is singular. In order to be able to use Lemma 4.1, we have to promote the rank of R_B by constructing the n by n upper triangular matrix

$$\bar{R} = \begin{bmatrix} R & S \\ O & I \end{bmatrix} . \tag{7.12}$$

Define the $(p+n-r)$ by n matrices U and V by

$$U = \begin{bmatrix} C \\ -J \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} C \\ J \end{bmatrix} . \tag{7.13}$$

where $J = \begin{bmatrix} O & I \end{bmatrix}$ is an $(n-r)$ by n matrix. Then it can be shown that

$$\overline{R}^T \overline{R} + U^T V = R_B^T R_B + C^T C \ . \tag{7.14}$$

Thus the least squares solution is given by

$$\overline{x} = (\overline{R}^T \overline{R} + U^T V)^{-1} A^T b \ . \tag{7.15}$$

Since $\overline{R}^T \overline{R}$ is now nonsingular, so we can apply Lemma 4.1 to (7.15) and obtain

$$\overline{x} = \{\overline{R}^{-1}\overline{R}^{-T} - \overline{R}^{-1}\overline{R}^{-T} U^T (I + V\overline{R}^{-1}\overline{R}^{-T} U^T)^{-1} V\overline{R}^{-1}\overline{R}^{-T}\} A^T b \ . \tag{7.16}$$

The computational algorithm is given below.

Algorithm 7

(1) Assume B has rank $r<n$. Compute the QR-decomposition of B. That is, $B = Q\begin{bmatrix} R & S \\ O & O \end{bmatrix}$. Construct the n by n upper triangular matrix $\overline{R} = \begin{bmatrix} R & S \\ O & I \end{bmatrix}$.

(2) Compute the vector $d = A^T b$.

(3) Solve the n by n sparse system $\overline{R}^T \overline{R} y = d$.

(4) Solve $(p+n-r)$ triangular systems $\overline{R}^T \begin{bmatrix} K_1 & K_2 \end{bmatrix} = \begin{bmatrix} C^T & -J^T \end{bmatrix}$.

(5) Compute the $(p+n-r)$ by $(p+n-r)$ matrix $D = I + \begin{bmatrix} K_1^T \\ -K_2^T \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix}$.

(6) Solve the $(p+n-r)$ by $(p+n-r)$ dense system $Dz = \begin{bmatrix} C \\ -J \end{bmatrix} y$.

(7) Solve the n by n sparse system $\overline{R}^T \overline{R} w = \begin{bmatrix} C^T & J^T \end{bmatrix} z$.

(8) Compute $\overline{x} = y - w$.

Algorithms 5, 6 and 7 are similar to Algorithms 1, 2 and 3 respectively. However, unlike underdetermined systems, the error in the computed least squares solution now depends on the square of the condition number of A. Thus it is important to note that Algorithms 5, 6 and 7 should be used only when A is well conditioned.

8. Concluding remarks

In this paper we have considered a method for solving sparse underdetermined systems of linear equations $Ax=b$. If AA^T is sparse, this method can be implemented efficiently using the technique proposed in [George 80] for computing the QR-decomposition of A^T. When A has dense columns, AA^T, and hence its Cholesky factor L, may have a large amount of fill-in. Techniques have presented for handling these dense columns. Apparently this is the first time updating algorithms have been employed in solving underdetermined systems.

We have also shown how the updating techniques can be applied to handle dense rows in sparse least squares problems. Algorithm 6 is similar to an updating algorithm described in [Plackett 50] but Algorithm 7 appears to be new. Different updating algorithms for sparse least squares problems which use the QR-decomposition of the sparse portion have been proposed in [George 80], [Bjorck 81] and [Heath 82]. These updating algorithms are derived by considering the residual in the least squares problem.

9. References

[Bjorck 81]

A. Bjorck, "A general updating algorithm for constrained linear least squares problems", Report LiTH-MAT-R-81-18, Dept. of Math., Univ. of Linkoping, Linkoping, Sweden.

[Cline 76]

R.E. Cline and R.J. Plemmons, "l_2-solutions to underdetermined linear systems", *SIAM Review*, 18 (1976), pp.92-106.

[George 80]

J.A. George and M.T. Heath, "Solution of sparse linear least squares problems using Givens rotations", *Linear Algebra and Appl.*, 34 (1980), pp.69-83.

[George 81]

J.A. George and J.W.H. Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[Heath 82]

M.T. Heath, "Some extensions of an algorithm for sparse linear least squares problems", *SIAM J. Sci. Stat. Comput.*, 3 (1982), pp.223-237.

[Henderson 81]

H.V. Henderson and S.R. Searle, "On deriving the inverse of a sum of matrices", *SIAM Review*, 23 (1981), pp.53-60.

[Paige 73]

C.C. Paige, "An error analysis of a method for solving matrix equations", *Math. Comp.*, 27 (1973), pp.355-359.

[Plackett 50]

"Some theorems in least squares" *Biometrika* 37 (1950), pp.149-157.

[Saunders 72]

M.A. Saunders, "Large-scale linear programming using the Cholesky factorization", Computer Science Department Report No. CS 252, Stanford University, Stanford, Calif., 1972.