No. 249

Sparsity Preserving Computation of Determinants

by

Kazuo Murota

December 1984

Sparsity Preserving Computation of Determinants

Kazuo Murota

Institute of Socio-Economic Planning, University of Tsukuba

Sakura, Ibaraki 305, Japan

December 1984

1. Introduction

The present short note is concerned with sparsity preserving methods for the numerical evaluation of the determinants of large sparse matrices. A standard method for computing the determinant of a matrix would be to decompose the matrix into LU factors and to compute the product of the diagonals of those factors. It seems difficult, however, to take full advantage of the sparsity if we stick to such direct methods. Here we will propose a method that is suitable for computing determinants of sparse matrices by means of parallel computation.

Let $M=(m_{ij})$ be a square matrix of order n and suppose that M is partitioned as

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} , \qquad (1)$$

where A and D are square matrices. Recall the following well-known results.

Lemma 1. $\det(M) = \det(A)\det(D - C A^{-1} B)$, (2)

if A is nonsingular.

Lemma 2 ([1]). Assume M is nonsingular and M^{-1} is partitioned in accordance with (1) as

$$M^{-1} = \begin{pmatrix} E & F \\ G & H \end{pmatrix} . \tag{3}$$

If A is nonsingular, so is H and

$$\det(M) = \det(A) / \det(H). \tag{4}$$

To avoid the complication concerning pivoting, we will assume that M is symmetric positive definite, though the following ideas can easily be adapted to general matrices with appropriate pivoting schemes. The leading principal submatrix of order r of M is denoted by M_r (e.g., $M_n=M$, $M_1=m_{11}$), which is nonsingular if M is positive definite.

2. The Proposed Method

The direct method (the Gaussian elimination) may be regarded as choosing $A=m_{11}$ (a single nonvanishing element of M) in (2) to recursively compute the second factor on the right hand side of (2) by this formula. On the other hand, the method proposed here takes M_{n-1} as the A in (2) and recursively computes the first factor on the right hand side of (2). More specifically, putting $b_r = (m_{1r}, m_{2r}, \ldots, m_{r-1,r})'$ and $c_r = (m_{r1}, m_{r2}, \ldots, m_{r,r-1})$ ($b_r = c_r'$ if M is symmetric), we compute det(M) according to the formula

$$\det(M) = m_{11} \Pi_{r=2}^{n} (m_{rr} - c_r M_{r-1}^{-1} b_r), \tag{5}$$

in which $x_r = M_{r-1}^{-1} b_r$ is evaluated by solving the linear equation

$$M_{r-1} x_r = b_r. \tag{6}$$

It should be noted here that

(i) if M is sparse, so are M_r $(r=1,\ldots,n-1)$ and the equation (6) can possibly be solved to any desired accuracy by some iterative method preserving sparsity,

and that

(ii) each term of (5) can be computed independently in parallel.

In this connection, PCG (=Preconditioned Conjugate Gradient) method or ICCG (=Incomplete Cholesky - Conjugate Gradient) method [2] seems quite useful. Suppose M is decomposed approximately into square-free Cholesky factors as

$$M \simeq L\ D\ L' \tag{7}$$

preserving sparsity. Then the leading principal submatrices of L and D, respectively denoted as L_r and D_r, serve also as the approximate Cholesky factors for M_r:

$$M_r \simeq L_r\ D_r\ L_r' \quad (r=1,2,\ldots,n). \tag{8}$$

Thus, we can use the decomposition (7) to solve (6) for $r=2,\ldots,n$ by ICCG.

The decomposition (8) could also be used to obtain rough estimates of det(M) before solving (6) for all r. Similarly to (5), we can derive

$$\det(M) = \det(M_k)\ \Pi_{r=k+1}^{n}\ (m_{rr} - c_r M_{r-1}^{-1} b_r), \tag{9}$$

in which $\det(M_k)$ may be replaced by $\det(L_k D_k L_k') = \det(D_k)\ (\det(L_k))^2$ and the second factor is computed by solving (6) for $r=k+1,\ldots,n$.

A variant of the method based on Lemma 2 is suggested below. Let e_r be the r-th column vector of the identity matrix of order r, i.e., $e_r = (0,\ldots,0,1)'$.

If we choose $A=M_{n-1}$ in Lemma 2, we have

$$\det(M) = \det(M_{n-1}) \;/\; h_n, \qquad (10)$$

where h_n is the n-th component of $y_n=M^{-1}e_n$. By recursive application of (4) to (10), we obtain

$$\det(M) = 1/ \Pi_{r=1}^{n} h_r \qquad (11)$$

or

$$\det(M) = \det(M_k)/ \Pi_{r=k+1}^{n} h_r, \qquad (12)$$

where h_r is the r-th component of y_r, which is determined by solving

$$M_r y_r = e_r. \qquad (13)$$

As is easily seen, y_r (r=1,...,n) represent the column vectors of the inverse of the U-factor in the LU-decomposition M=LU with L being a unit lower triangular matrix. Note that the L-factor is not computed here.

If we solve (13) by partitioning M_r as in (1) with $A=M_{r-1}$, $B=b_r$, $C=c_r$, $D=m_{rr}$, then we have

$$1/h_r = m_{rr} - c_r\, M_{r-1}^{-1} b_r, \qquad (14)$$

which reveals that (5) and (11) are algebraically equivalent. Thus we may either solve (6) to compute $m_{rr}-c_r\, M_{r-1}^{-1} b_r = m_{rr}-c_r\, x_r$ or work with (13) to obtain h_r directly.

The author thanks Dr. Masaaki Sugihara for valuable comments.

References

[1] A. S. Householder: The Theory of Matrices in Numerical Analysis. Blaisdell, New York, 1964.

[2] J. A. Meijerink and H. A. van der Vorst: An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix. Mathematics of Computation, Vol. 31 (1977), pp.148-162.