NUMERICAL ALGORITHMS FOR THE MOORE-PENROSE INVERSE OF A MATRIX: DIRECT METHODS

NOBUO SHINOZAKI, MASAAKI SIBUYA AND KUNIO TANABE

(Received May 31, 1971)

Summary

Direct methods for computing the Moore-Penrose inverse of a matrix are surveyed, classified and tested. It is observed that the algorithms using matrix decompositions or bordered matrices are numerically more stable.

1. Introduction

The purpose of this paper is to survey and classify the published algorithms for obtaining the Moore-Penrose inverse of a possibly rectangular real matrix and to report on tests of their performances using a type of test matrices. The algorithms may be applied for complex matrices but we shall treat only real ones.

Throughout the paper A is an $m\times n$ matrix of rank r which is possibly less than $\min(m, n)$. An $n\times m$ matrix G which satisfies

$$\begin{array}{llll} \text{(i)} & AGA=A & \text{(ii)} & (AG)^T=AG \\ \text{(iii)} & (GA)^T=GA & \text{(iv)} & GAG=G \end{array} \tag{1.1}$$

and is uniquely determined by A is called the Moore-Penrose (generalized) inverse of A and denoted by $A^\dagger$. Any matrix which satisfies (i) is denoted by A^-, that which satisfies (i) and (ii) by A_l^- and that which satisfies (i) and (iii) by A_m^-.

We shall denote the transpose of A by A^T, its range space by $\mathcal{R}(A)$, the orthogonal projection on $\mathcal{R}(A)$ by Π_A, and the orthogonal complement of a subspace S by $S^\perp$.

The role of $A^\dagger$ in the theory of linear statistical model is well known. For the minimization problem of $\|Ax-b\|$, $x=A^\dagger b$ gives the solution of the minimum norm. The solution is said to be numerically more stable among least squares solutions A_l^-b, but this point is not well formulated yet so far as the authors know.

2. A classification and preliminaries

Direct methods for computing $A^\dagger$ can be classified as follows.

A. Methods based on matrix decompositions.
   A1. Matrix decomposition into a product of two matrices of full rank.
   A2. Singular value decomposition.
B. Methods using bordered matrices.
C. Others.
   C1. Greville's recursive method.
   C2. Methods based on the formula $A^\dagger=(A^TA)_m^-A^T$.
   C3. Pyle's gradient projection method.

Discussions in Sections 3–6 follow in this order.

We state some simple facts as preliminaries.

PROPOSITION 2.1. *If P and Q are orthogonal matrices, then*

$$(PAQ)^\dagger=Q^TA^\dagger P^T .$$

PROPOSITION 2.2. $(A^T)^\dagger=(A^\dagger)^T$.

PROPOSITION 2.3. *Let A be a matrix defined in Section* 1 *and $A=BC$ where B and C are $m\times r$ and $r\times n$ matrices respectively. Then,*

$$A^\dagger=C^\dagger B^\dagger .$$

PROPOSITION 2.4. *If B and C are of full rank as above,*

$$B^\dagger=(B^TB)^{-1}B^T \qquad \text{and} \qquad C^\dagger=C^T(CC^T)^{-1} .$$

PROPOSITION 2.5. *If O's are zero matrices of suitable size*

$$[A, O]^\dagger=\begin{bmatrix} A^\dagger \\ O \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} A \\ O \end{bmatrix}^\dagger=[A^\dagger, O] .$$

3. Decomposition into matrices of full rank

In order to decompose a matrix A into a product of the form in Proposition 2.3, the following three algorithms may be used. Pivoting is necessary in these algorithms, however we shall not discuss its rule and assume that the rows and columns of A are permuted so that the algorithms proceed straightforward. The permutations are justified by Proposition 2.1.

Procedure .1. Gaussian elimination (LU decomposition).

By Gaussian elimination A can be represented as

(3.1) $$A=LU\,,$$

where L is lower echelon ($l_{ij}=0$ if $i<j$) and U is upper echelon ($u_{ij}=0$ if $i>j$) both with nonzero "diagonal" elements. When the complete pivoting rule is used, the further decomposition

(3.2) $$A=LD\tilde{U}\,,$$

where $D=\mathrm{diag}\,(u_{ii})$, makes $|l_{ij}|\leqq 1$ and $|\tilde{u}_{ij}|\leqq 1$.

Procedure .2. Householder transformation.

Premultiplying an orthogonal matrix Q to A, we can have

(3.3) $$QA=\begin{bmatrix} U \\ O \end{bmatrix}\,,$$

where U is upper echelon. Denoting the first r rows of Q by Q_r, we have

(3.4) $$A=Q_r^T U$$

and

(3.5) $$A^\dagger=U^\dagger Q_r\,.$$

The orthogonal transformation (3.3) can also be obtained by Givens' method. Since it needs more operations than Householder's method, we leave it out of consideration.

Procedure .3. Gram-Schmidt of orthogonalization.

Let Q be an $m\times r$ matrix whose columns form an orthonormal basis of $\mathcal{R}(A)$. The orthonormalization of the column vectors of A is equivalent to the decomposition

(3.6) $$A=QU\,,$$

where U is an $r\times n$ upper echelon matrix. Then,

(3.7) $$A^\dagger=U^\dagger Q^T\,.$$

By these three procedures the problem is reduced to obtaining the Moore-Penrose inverse of matrices of echelon form with full rank. We shall consider an $r\times n$ upper echelon matrix U.

Procedure .01. Inversion of a non-singular matrix.

From Proposition 2.4 we get $U^\dagger$ by inverting UU^T. An appropriate method for inversion is to get the Cholesky decomposition of $UU^T=FF^T$. Then $(UU^T)^{-1}=(F^{-1})^T F^{-1}$.

Procedure .02. Householder transformation.

If we apply the Householder transformation to U^T, then we have

$$PU^T=\begin{bmatrix} R \\ O \end{bmatrix}, \tag{3.8}$$

where R is upper triangular and non-singular. Thus denoting the first r rows of P by P_r,

$$U^T=P_r^T R \tag{3.9}$$

and

$$U^\dagger=P_r^T(R^{-1})^T . \tag{3.10}$$

Procedure .03. Gram-Schmidt orthogonalization.

Let N be an $n\times r$ matrix whose r columns form an orthonormal basis of $\mathcal{R}(U^T)$. We obtain

$$U^T=NR , \tag{3.11}$$

where R is an $r\times r$ upper triangular non-singular matrix, and

$$U^\dagger=N(R^{-1})^T . \tag{3.12}$$

Using these Procedures .01–.03 and .1–.3 the problem for obtaining $A^\dagger$ is solved. Variations of .01–.03, however, may be considered as follows. We may write

$$U=[S, T]=S[I, G] , \tag{3.13}$$

where S is upper triangular and non-singular and $G=S^{-1}T$. Thus

$$U^\dagger=[I, G]^\dagger S^{-1} . \tag{3.14}$$

The computation of S^{-1} and G is a backward substitution and this additional small work will be compensated by the reduction of operations in computing $[I, G]^\dagger$ rather than $U^\dagger$. Corresponding to Procedures .01–.03 we get the following .04–.06.

Procedure .04. Inversion of non-singular matrix.

We use the formula

$$[I, G]^\dagger=\begin{bmatrix} I \\ G^T \end{bmatrix}(I+GG^T)^{-1} , \tag{3.15}$$

where $(I+GG^T)^{-1}$ is computed from the Cholesky decomposition $I+GG^T=FF^T$ as Procedure .01.

Procedure .05. Householder transformation.

As Procedure .02, using an orthogonal matrix P

$$P\begin{bmatrix} I \\ G^T \end{bmatrix}=\begin{bmatrix} R \\ O \end{bmatrix}, \tag{3.16}$$

where R is an $r\times r$ upper triangular non-singular matrix. So we have

$$[I, G]^\dagger=P^T\begin{bmatrix} (R^{-1})^T \\ O \end{bmatrix}.$$

Denoting the first r rows of P by P_r, the first r columns of P_r by P_{rr}, we get

$$[I, G]^\dagger=P_r^T P_{rr} \tag{3.17}$$

since $P_{rr}^T R=I$ from (3.16).

Procedure .06. Gram-Schmidt orthogonalization.
Similarly to Procedure .03,

$$\begin{bmatrix} I \\ G^T \end{bmatrix}=NR, \tag{3.18}$$

where R is an $r\times r$ upper triangular non-singular matrix. Then we have

$$[I, G]^\dagger=N(R^{-1})^T.$$

Denoting the first r rows of N by N_r we have

$$[I, G]^\dagger=NN_r^T. \tag{3.19}$$

Now comparing Procedures .02 and .03 with Procedures .05 and .06 respectively we see that the difference between them is just the order of decomposition and elimination (backward substitution). Combining Procedures .1–.3 and Procedures .01–.06 we get 18 possible variations, from which we should select some considering number of operations, necessary memory space and accuracy.

According to our classification the published papers are summarized as Table 1. Peters and Wilkinson [14] mentioned of Procedures .1–.3 but not of detailed variations.

Tewarson [17] suggested to compute $(I+GG^T)^{-1}$ by the formula

$$(I+GG^T)^{-1}=(R^TR)^{-1}=N_rN_r^T, \tag{3.20}$$

where R and N_r are defined in (3.18) and (3.19), and compute

$$[I, G]^\dagger=\begin{bmatrix} I \\ G^T \end{bmatrix}N_rN_r^T.$$

This is, however, equivalent to (3.19). Rust, Burrus and Schneeberger [16] published a slight modification of .36.

Table 1

.1	.11	Kublanovskaya (1966)
	.14	Noble (1966)
	.15	Tewarson (1968)
	.16	Tewarson (1967)
.2	.21	Kublanovskaya (1966)
	.25	Tewarson (1969)
.3	.36	Rust, Burrus and Schneeberger (1966)
		Goldstein (1968)

4. Singular value decomposition

Let

$$A=U^T\begin{bmatrix}\sigma_1 & & 0\\ & \ddots & \\ & & \sigma_r\\ 0 & & \end{bmatrix}V. \tag{4.1}$$

be a singular value decomposition of A, where U and V are orthogonal matrices, then

$$A^\dagger=V^T\begin{bmatrix}1/\sigma_1 & & 0\\ & \ddots & \\ & & 1/\sigma_r\\ 0 & & \end{bmatrix}U. \tag{4.2}$$

To obtain a decomposition of (4.1) Golub and others (Golub and Kahan [5], Golub and Businger [6], and Golub and Reinsch [7]) proposed the following algorithms.

By premultiplications and postmultiplications of reflexion matrices A is transformed to bi-diagonal form: (we assume $m\geqq n$ for convenience)

$$PAQ=\begin{bmatrix}J\\ O\end{bmatrix},\qquad J=\begin{bmatrix}\alpha_1 & \beta_1 & 0 & \cdots & 0 & 0\\ 0 & \alpha_2 & \beta_2 & \cdots & 0 & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & 0 & \cdots & \alpha_{n-1} & \beta_{n-1}\\ 0 & 0 & 0 & \cdots & 0 & \alpha_n\end{bmatrix}.$$

The problem is to find a singular value decomposition

$$J=X^T\Sigma Y.$$

Now if write

$$K=\begin{bmatrix}O & J\\ J^T & O\end{bmatrix}$$

and

$$\begin{bmatrix} O & J \\ J^T & O \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\pm\sigma\begin{bmatrix} x \\ y \end{bmatrix},$$

then an eigenvalue σ is a singular value of J, and x and y are column vectors of X and Y respectively. If we permute the rows and columns of K so that the vector $\begin{bmatrix} x \\ y \end{bmatrix}$ is changed to a vector z with components

$$z_{2i}=x_i \quad \text{and} \quad z_{2i-1}=y_i ,$$

then

$$\begin{bmatrix} 0 & \alpha_1 & & & & & \\ \alpha_1 & 0 & \beta_1 & & & & \\ & \beta_1 & 0 & \alpha_2 & & & \\ & & & \ddots & \ddots & & \\ & & & & 0 & \alpha_n \\ & & & & \alpha_n & 0 \end{bmatrix} z=\pm\sigma z .$$

So the problem is reduced to the complete eigenvalue problem of a symmetric tri-diagonal matrix, which can be solved appropriately by the QR method.

5. Bordered matrices

Let V be an $m\times(m-r)$ matrix whose columns form a basis of $\mathcal{R}(A)^{\perp}$ and U be an $n\times(n-r)$ matrix whose columns form a basis of $\mathcal{R}(A^T)^{\perp}$. Define

$$B=\begin{bmatrix} A & V \\ U^T & O \end{bmatrix}, \tag{5.1}$$

which is non-singular, and

$$B^{\dagger}=\begin{bmatrix} A^{\dagger} & (U^T)^{\dagger} \\ V^{\dagger} & O \end{bmatrix}. \tag{5.2}$$

Or considering the first n columns of B,

$$C=\begin{bmatrix} U^T \\ A \end{bmatrix}, \tag{5.3}$$

which has full rank, and

$$C^{\dagger}=[A^{\dagger}, (U^T)^{\dagger}] . \tag{5.4}$$

A method for obtaining V is as follows. U is obtained similarly. If we premultiply a non-singular matrix Q to get the form

$$QA=\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}A=\begin{bmatrix} A_1 \\ O \end{bmatrix},$$

where A_1 is an $r\times m$ upper-echelon matrix, then we may take Q_2^T as V.

Bordering was proposed by Hestenes [9] and Germain-Bonne [2].

6. Other methods

6.1. Greville [8] proposed a recursive algorithm, which starting from a column vector computes the Moore-Penrose inverse of a matrix with one more column successively. If

$$A_k=[A_{k-1}, a_k], \tag{6.1}$$

then decomposing

$$a_k=a_k^{(1)}+a_k^{(2)}, \qquad a_k^{(1)}\in\mathcal{R}(A_{k-1}) \qquad \text{and} \qquad a_k^{(2)}\in\mathcal{R}(A_{k-1})^{\perp},$$

we have

$$A_k^{\dagger}=\begin{bmatrix} A_{k-1}^{\dagger}-d_kb_k^T \\ b_k^T \end{bmatrix}, \tag{6.2}$$

where

$$\begin{aligned} &d_k=A_{k-1}^{\dagger}a_k=A_{k-1}^{\dagger}a_k^{(1)}, \\ &b_k^T=\begin{cases} a_k^{(2)\dagger}, & \text{if } a_k^{(2)}\neq 0, \\ (1+d_k^Td_k)^{-1}d_k^TA_{k-1}^{\dagger}, & \text{if } a_k^{(2)}=0. \end{cases} \end{aligned} \tag{6.3}$$

The decomposition of a_k can be computed by the orthogonal projection:

$$a_k^{(1)}=A_{k-1}A_{k-1}^{\dagger}a_k=A_{k-1}d_k, \qquad a_k^{(2)}=a_k-a_k^{(1)}. \tag{6.4}$$

6.2. Ben-Israel and Wersan [1] proposed the use of the formula

$$A^{\dagger}=(A^TA)_m^-A^T. \tag{6.5}$$

Let LU be a decomposition of A^TA by Procedure .1 and

$$L=\begin{bmatrix} R \\ S \end{bmatrix} \tag{6.6}$$

where R is a lower triangular non-singular matrix. We may take as $(A^TA)_m^-$, which is not unique in general,

$$(A^TA)_m^-=U^{\dagger}L^-=U^{\dagger}[R^{-1}, O]. \tag{6.7}$$

Glassey [3] proposed to construct a matrix B whose columns form

an orthogonal basis of $\mathcal{R}(A^T)$ and simultaneously $C^T = B^T A^T$. Thus, substituting

$$(A^T A)^\dagger = B(B^T A^T A B)^{-1} B^T$$

into $(A^T A)_m^-$ in (6.5), we have

$$A^\dagger = B(C^T C)^{-1} C^T \,. \tag{6.8}$$

Penrose ([12], [13]) suggested methods along the same line.

6.3. Pyle [15] called the following method "gradient projection." Let

$$A^T = QU = Q[R, S]$$

be the Gram-Schmidt orthogonalization of Procedure .3, and

$$A_m^- = Q[(R^{-1})^T, O] \,.$$

If $A = PV$ is also the Gram-Schmidt orthogonalization, then

$$\Pi_A = PP^T$$

is the orthogonal projection on $\mathcal{R}(A)$, and

$$A^\dagger = A_m^- \Pi_A \,.$$

7. Test matrices and test results

For numerical examination of the algorithms we have used matrices of the following type. They can be generated by computer for a given set of singular values. Their elements are integers so rounding errors in machine generation can be avoided. Their Moore-Penrose inverse are easily computed. The magnitude of the elements are of the same order so the effect of equilibration can be studied separately.

Let

$$A = UDV$$

where U is an $m \times m$ ($m = 2^k$) Hadamard's orthogonal matrix with ± 1 elements, D is an $m \times n$ matrix with zero elements except for given integer parameters d_{ii} ($i = 1, \cdots, r$), and V is an $n \times n$ row-orthogonal matrix;

$$V = \begin{bmatrix} 1 & 1 & 1 \cdots & 1 & 1 \\ 1 & 1 & 1 \cdots & 1 & -n+1 \\ 1 & 1 & 1 \cdots & -n+2 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & -2 \cdots & 0 & 0 \\ 1 & -1 & 0 \cdots & 0 & 0 \end{bmatrix}.$$

The singular values of A are $d_{11}\sqrt{mn}$, $d_{22}\sqrt{m(n-1)n}$, $d_{33}\sqrt{m(n-2)(n-1)}, \cdots$

Table 2 shows a part of the test results. The computation was done by IBM-7040 WATFOR, with 27 bit mantissa, the accuracy is measured by the number of exact digits: $-\log_{10}$(relative error). The other results showed similar tendency. The elimination, the Gram-Schmidt and the bordering methods seem preferable. In the singular value decomposition method, the QR method gives the eigenvalues with reasonable "absolute" errors, so that the relative error is large in ill-conditioned cases. The algorithms described in Section 6 are more complicated than others, need more operations and consequently accumulate more rounding errors.

Table 2. Accuracy of some solutions

$m=n=8$, $r=6$

Case	1	2	3	4
Positive singular values	10^4	10^5	10^7	10^6
	10^3	10^4	10^6	10^5
	10^2	10^3	10^4	10^6
	5×10	10^2	10^3	1
	10	10	10	1
	1	1	1	1
E	4.94	4.00	1.84	1.53
H	3.85	3.18	1.18	0.82
G	4.50	4.07	1.44	1.45
S	3.18	2.77	0.86	1.05
B	4.65	3.83	2.19	1.15
P	2.49	1.21	—	—

E: Elimination (.11)
H: Householder (.25)
G: Gram-Schmidt (.36)
S: Singular value decomposition
B: Bordering
P: Pyle's method

Singular values are approximate values. Accuracy is $-\log_{10}\{\max(|\bar{g}_{ij}-g_{ij}|/|g_{ij}|)\}$, where $\bar{g}_{ij}$ is a computed value of the element g_{ij}.

KEIO UNIVERSITY, FACULTY OF ENGINEERING
IBM JAPAN, SCIENTIFIC CENTER, TOKYO
INSTITUTE OF STATISTICAL MATHEMATICS

REFERENCES

[1] Ben-Israel, A. and Wersan, S. J. (1963). An elimination method for computing the generalized inverse of an arbitrary complex matrix, *Jour. ACM*, **10**, 532-537.

[2] Germain-Bonne, G. (1969). Calcul de pseudo-inverses, *R.F.I.R.O.*, 3e année, NR-2/1969, 3-14.

[3] Glassey, C. R. (1966). An orthogonalization method of computing the generalized inverse of a matrix, *Op. Res. Center, Univ. Calif., Berkeley*, ORC 66-10, (AD633061).

[4] Goldstein, M. J. (1968). Solving systems of linear equations by using the generalized inverse, *USL Report No. 874, US Navy Underwater Sound Laboratory*, (AD667727).

[5] Golub, G. H. and Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix, *SIAM J. Numer. Anal., Ser. B*, **2**, No. 2, 205-224.

[6] Golub, G. H. and Businger, P. (1967). Least squares, singular values and matrix approximations; An ALGOL procedure for computing the singular value decomposition, Technical Report No. CS73, *Stanford University, Computer Science Department*, (AD 662883).

[7] Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solution, *Numer. Math.*, **14**, 403-420.

[8] Greville, T. N. E. (1960). Some applications of the pseudo inverse of a matrix, *SIAM Rev.*, **2**, 15-22.

[9] Hestenes, M. R. (1958). Inversion of matrices by biorthogonalization and related results, *J. SIAM*, **6**, 51-90.

[10] Kublanovskaya, V. N. (1966). Evaluation of a generalized inverse matrix and projector, *USSR Comp. Math. and. Math. Phys.*, 179-188.

[11] Noble, G. (1966). A method for computing the generalized inverse of a matrix, *SIAM J. Numer. Anal.*, **3**, 582-584.

[12] Penrose, R. (1955). A generalized inverse for matrices, *Proc. Camb. Phil. Soc.*, **51**, 406-413.

[13] Penrose, R. (1956). On best approximate solutions of linear matrix equations, *Proc. Camb. Phil. Soc.*, **52**, 17-19.

[14] Peters, G. and Wilkinson, J. H. (1970). The least squares problem and pseudo-inverses, *The Computer Journal*, **13**, 309-316.

[15] Pyle, L. D. (1964). Generalized inverse computations using the gradient projection method, *Jour. ACM*, **11**, 422-428.

[16] Rust, B., Burrus, W. R. and Schneeberger, C. (1966). A simple algorithm for computing the generalized inverse of a matrix, *Comm. ACM*, **9**, 381-387.

[17] Tewarson, R. P. (1967). A direct method for generalized matrix inversion, *SIAM J. Numer. Anal.*, **4**, 499-507.

[18] Tewarson, R. P. (1968). A computational method for evaluating generalized inverses, *The Computer Journal*, **10**, 411-413.

[19] Tewarson, R. P. (1969). On computing generalized inverses, *Computing*, **4**, 139-151.