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

NOBUO SHINOZAKI, MASAAKI SIBUYA AND KUNIO TANABE

(Received Jan. 10, 1972)

Summary

The iterative methods by Ben-Israel and others for computing the Moore-Penrose inverse of a matrix are examined. Ill conditioned test matrices are inverted by the methods and some difficulties are found out. The iterative methods do not seem superior to direct ones.

1. Introduction

The role of generalized inverse matrices in linear statistical inference and other applications is well recognized as recent publications show. See, for example, three books by Pringle and Rayner [5], Boullion and Odell [2], and Rao and Mitra [6]. Accordingly it is important both practically and theoretically to find good algorithms for computing generalized inverse matrices.

In a previous paper the authors surveyed, classified and tested direct methods for computing the Moore-Penrose inverse of a possibly rectangular real matrix. In this subsequent paper we consider about iterative methods. The Moore-Penrose inverse is not the only one important among generalized inverses of a matrix. It is necessary, however, to define a unique inverse to find algorithm, and the Moore-Penrose inverse is defined in a very natural way for a given matrix.

The motivation of the research was the hope for finding a better algorithm in ill conditioned case. Our experiences show that the iterative methods are not superior numerically to the direct methods.

The iterative methods may be applied to least-squares solution of a system of linear equations, which is not treated here. There exists a good iterative method as proposed by one of the authors [8].

Ben-Israel and others proposed iterative methods for computing the Moore-Penrose inverse of a matrix (see [1] and the papers in its references), and Petryshyn [4] and Zlobec [9] extended the methods to a general one (see also [3]). The purpose of this paper is to summarize these papers and add some new results (Section 2); and to apply the methods to test matrices and point out difficulties in practical computation (Section 3).

Let A be an $m\times n$ complex matrix of rank $r\leqq\min(m, n)$ (we assume $m\geqq n$ without loss of generality), A^* its conjugate transpose, $A^\dagger$ its Moore-Penrose inverse, $\mathcal{R}(A)$ its range space, $\sigma_1(A)\geqq\cdots\geqq\sigma_r(A)>0$ its singular values ($\sigma_1(A)$ is the spectral norm of A), and Π_A an orthogonal projection on $\mathcal{R}(A)$. Let $A=UDV^*$ be a singular value decomposition of A, that is U and V are unitary matrices of dimension m and n respectively, D is $m\times n$ and has zero elements except for that the first r diagonal elements are equal to $\sigma_1(A),\cdots,\sigma_r(A)$. The following known facts will be used often in our discussion:

$$A^\dagger A=\Pi_{A^*}\,,\qquad AA^\dagger=\Pi_A\,,\qquad \Pi_{A^*}A^\dagger=A^\dagger\Pi_A=A^\dagger\,,$$

and $\Pi_A X=X$ for any matrix X such that $\mathcal{R}(X)\subset\mathcal{R}(A)$.

2. Algorithms and theorems

Algorithm 1 (the pth order hyperpower method). Let $p\geqq 2$ be a fixed integer. Compute a sequence of $n\times m$ matrices $\{X(k);\ k=0, 1, 2,\cdots\}$ starting from a matrix $X(0)$ and following the relation

$$(1)\qquad X(k+1)=\sum_{j=0}^{p-1}(I-X(k)A)^j X(k)\ .$$

Remark. Since $(I-XA)X=X(I-AX)$ the expression (1) can also be written as

$$(1')\qquad X(k+1)=\sum_{j=0}^{p-1}X(k)(I-AX(k))^j\ .$$

Notice that we are assuming $m\geqq n$ and the number of arithmetic multiplications for computing one step of (1) is $(p-1)n^3+mn^2$ while for (1′) $(p-1)m^3+nm^2$.

THEOREM 1. *If* $X(0)=A^*WA^*$, *where* W *is an arbitrary* $m\times n$ *matrix, then the following relation holds for the sequence* $\{X(k)\}$ *of Algorithm* 1,

$$(2)\qquad \begin{aligned} A^\dagger-X(k+1)&=A^\dagger(I-AX(k))^p=A^\dagger(A(A^\dagger-X(k))^p\\ &=(I-X(k)A)^pA^\dagger=((A^\dagger-X(k))A)^pA^\dagger\,,\\ &\qquad\qquad k=0, 1, 2,\cdots\ . \end{aligned}$$

Remark. The condition on $X(0)$ is equivalent to $\mathcal{R}(X(0))\subset\mathcal{R}(A^*)$ and $\mathcal{R}(X(0)^*)\subset\mathcal{R}(A)$.

PROOF. Evidently $X(k)$'s satisfy the condition on $X(0)$. Here we drop the parameter of $X(k)$ and write X for short. The equivalence of the last four expressions can be shown by noticing the following equalities.

$$\begin{aligned} A^\dagger - X &= A^\dagger - \Pi_{A^*}X = A^\dagger(I-AX) \\ &= A^\dagger - X\Pi_A = (I-XA)A^\dagger , \end{aligned}$$

$$\Pi_A - AX = \Pi_A(I-AX) = (I-AX)\Pi_A ,$$

and

$$\Pi_{A^*} - XA = \Pi_{A^*}(I-XA) = (I-XA)\Pi_{A^*} .$$

Now we prove the first equality by induction with respect to p.

$$\begin{aligned} A^\dagger(I-AX)^2 &= (A^\dagger - X)(I-AX) \\ &= A^\dagger - X - (A^\dagger - X)AX \\ &= A^\dagger - X - (I-XA)X . \end{aligned}$$

If the first equality is valid for $p=q-1$, then

$$\begin{aligned} A^\dagger(I-AX)^q &= \left(A^\dagger - \sum_{j=0}^{q-2}(I-XA)^j X\right)(I-AX) \\ &= A^\dagger - X - \sum_{j=0}^{q-2}(I-XA)^j(I-XA)X \\ &= A^\dagger - \sum_{j=0}^{q-1}(I-XA)^j X . \end{aligned}$$

Remark. The first equality holds for any $X(0)$ such that $\mathcal{R}(X(0)) \subset \mathcal{R}(A^*)$ and $A^\dagger - X(k+1) = (I-X(k)A)^p A^\dagger$ for any $X(0)$ such that $\mathcal{R}(X^*(0)) \subset \mathcal{R}(A)$.

COROLLARY 1. *Under the same condition on* $X(0)$

$$\Pi_A - AX(k+1) = (\Pi_A - AX(k))^p = \Pi_A(I-AX(k))^p , \tag{3}$$

and

$$\Pi_{A^*} - X(k+1)A = (\Pi_{A^*} - X(k)A)^p = (I-X(k)A)^p \Pi_{A^*} . \tag{3'}$$

PROOF. Pre- and post-multiplication of A and (2) leads to (3) and (3′) respectively.

COROLLARY 2. *Under the same condition on* $X(0)$,

$$\begin{aligned} A^\dagger - X(k) &= A^\dagger(I-AX(0))^{p^k} = A^\dagger(\Pi_A - AX(0))^{p^k} \\ &= (I-X(0)A)^{p^k}A^\dagger = (\Pi_{A^*} - X(0)A)^{p^k}A^\dagger . \end{aligned} \tag{4}$$

So that $X(k) \to A^\dagger$, $AX(k) \to \Pi_A$ *and* $X(k)A \to \Pi_{A^*}$ $(k \to \infty)$ *iff* $\sigma_1(\Pi_A -$

$AX(0))=\sigma_1(\Pi_{A^*}-X(0)A)<1$.

PROOF. The equality of the two singular values is shown from the singular value decomposition of A.

COROLLARY 3. *If, in special, $X_0=\omega A^*$, then the convergences in Corollary 2 are valid iff $0<\omega<2/\sigma_1^2(A)$.*

PROOF. $\sigma_i(\Pi_A-\omega AA^*)=\sigma_i(\Pi_{A^*}-\omega A^*A)=|1-\omega\sigma_i^2(A)|$.

COROLLARY 4. *We assume $X(0)=\omega A^*$. When p is even,*

$$\text{trace } X(k)A=\text{trace } AX(k)\uparrow r\ , \qquad \textit{if } 0<\omega<2/\sigma_1^2(A)\ ,$$

and when p is odd,

$$\text{trace } X(k)A=\text{trace } AX(k)\downarrow r\ , \qquad \textit{if } 1/\sigma_r^2(A)<\omega<2/\sigma_1^2(A)\ ,$$

for $k=1, 2,\cdots$.

Remark. The last condition on singular values $0.5<\sigma_r^2(A)/\sigma_1^2(A)$ is too restrictive for general application. Trace $X(1)A$ can be greater or less than trace $X(0)A$.

PROOF. It is shown that $(\Pi_{A^*}-\omega A^*A)^p=VK^pV^*$, where K is an $n\times n$ matrix with zero elements except for the first r diagonal elements $1-\omega\sigma_i^2(A)$, $i=1,\cdots,r$. Then

$$\begin{aligned}\text{trace } X(k)A&=r-\text{trace }(\Pi_{A^*}-X(0)A)^{p^k}\\&=r-\sum_{i=1}^{r}(1-\omega\sigma_i^2(A))^{p^k}\ .\end{aligned}$$

Algorithm 2 (the linear method). Compute a sequence of $n\times m$ matrices $\{Y(k);\ k=0, 1, 2,\cdots\}$ starting from a matrix $Y(0)$ and following the relation

$$(5)\qquad Y(k+1)=Y(k)+\alpha(I-Y(k)A)A^*\ .$$

THEOREM 2. *For any $Y(0)$*

$$(6)\qquad A^\dagger-Y(k)=(A^\dagger-Y(0))(I-\alpha AA^*)^k\ , \qquad k=1, 2, 3,\cdots\ .$$

PROOF.

$$\begin{aligned}A^\dagger-Y(k)&=A^\dagger-Y(k-1)-\alpha(A^*-Y(k-1)AA^*)\\&=A^\dagger-\alpha A^\dagger AA^*-Y(k-1)+\alpha Y(k-1)AA^*\\&=(A^\dagger-Y(k-1))(I-\alpha AA^*)\ .\end{aligned}$$

COROLLARY 1. *Assume that $\mathcal{R}(Y^*(0))\subset\mathcal{R}(A)$. Then $Y(k)\to A^\dagger$, $AY(k)\to\Pi_A$ and $Y(k)A\to\Pi_{A^*}$ $(k\to\infty)$ iff $0<\alpha<2/\sigma_1^2(A)$.*

PROOF. It is shown that $(I-\alpha AA^*)^k = U(I-\alpha D^2)^k U^*$. So this sequence of matrices converges to $I-\Pi_A$ iff $-1<1-\alpha\sigma_i^2(A)<1$.

Remark. The dual of Algorithm 2 is defined by the relation

$$Y(k+1)=Y(k)+\alpha A^*(I-AY(k)) \,. \tag{5'}$$

Thus,

$$A^\dagger - Y(k) = (I-\alpha A^*A)^k (A^\dagger - Y(0)) \,, \tag{6'}$$

and for $Y(0)$ such that $\mathcal{R}(Y(0))\subset\mathcal{R}(A^*)$ the convergence statement in Corollary 1 is true. The following Corollary 3 is also true for this dual algorithm.

COROLLARY 2. *If we take* $X(0)=\alpha A^*$ *in Algorithm* 1 *and* $Y(0)=\alpha A^*$ *in Algorithm* 2 *(or its dual algorithm), then*

$$X(k)=Y(p^k-1) \,. \tag{7}$$

PROOF. Both sides are equal to

$$A^\dagger - A^\dagger(I-\alpha AA^*)^{p^k} = A^\dagger - (I-\alpha A^*A)^{p^k} A^\dagger \,.$$

COROLLARY 3. *When* $Y(0)=\omega A^*$ *and* $0<\alpha\leqq 1/\sigma_1^2(A)$ *in Algorithm* 2,

$$\text{trace } AY(k) = \text{trace } Y(k)A \uparrow r \,, \qquad \textit{if } \omega<1/\sigma_1^2(A) \,,$$

and

$$\text{trace } AY(k) = \text{trace } Y(k)A \downarrow r \,, \qquad \textit{if } \omega>1/\sigma_1^2(A) \,.$$

PROOF. Under the assumption on $Y(0)$,

$$Y(k)A = \Pi_{A^*} - (\Pi_{A^*} - \omega A^*A)(\Pi_{A^*} - \alpha A^*A)^k$$

and the second term of the right-hand side is equal to VK_kV^* where K_k is an $n\times n$ matrix with zero elements except for the first r diagonal elements $(1-\omega\sigma_i^2)(1-\alpha\sigma_i^2)^k$, $i=1,\cdots,r$. Thus,

$$\text{trace } Y(k)A = r - \sum_{i=1}^{r}(1-\omega\sigma_i^2)(1-\alpha\sigma_i^2)^k \,, \tag{8}$$

which completes the proof.

Finally, we remark inequalities to determine α and ω:

$$\sigma_1^2(A) \leqq \|A^*A\|_\infty = \max_i \sum_j \left|\sum_k a_{ki}^* a_{kj}\right| \,, \tag{9}$$

and

$$\sigma_1^2(A) \leqq \|A\|_E = \sum_{i,j}|a_{ij}|^2 \,. \tag{9'}$$

Both inequalities are sharp and there is not much difference between them. And for the smallest singular value:

$$\sigma_r^{-2}(A) \leqq \|(A^\dagger)^* A^\dagger\|_\infty \quad \text{and} \quad \sigma_r^{-2}(A) \leqq \|A^\dagger\|_E \,. \tag{10}$$

So we get an estimate of $\sigma_r(A)$ only after obtained a good approximation of $A^\dagger$.

3. Numerical practice

The convergence of our iterative algorithms depends heavily on the distribution of singular values of a given matrix. To examine actual numerical process we need matrices with different conditions. The algorithm used in [4] provides a matrix of given size with a given set of singular values.

Our first observation is the slowness of convergence of the linear method, Algorithm 2. Fig. 1 shows the behavior of trace $Y(k)A$ stated in Theorem 2, Corollary 3 when A is a very well conditioned matrix. The elements of $Y(k)$ converge as slowly as the trace. The proof of Theorem 2, Corollary 3 shows that the convergence of the trace is de-

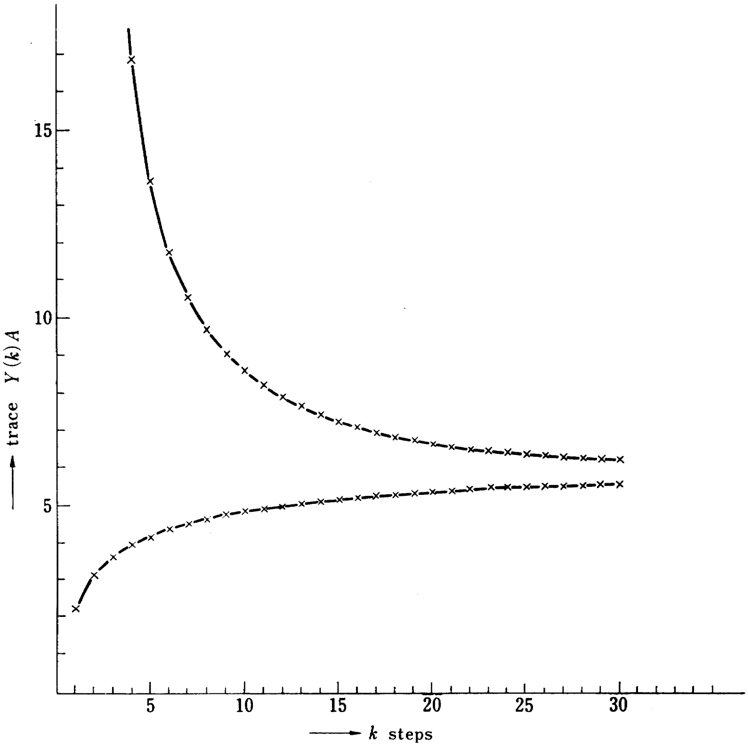

Fig. 1. trace $Y(k)A$ in Theorem 2, Corollary 3. A is 8×8 and has singular values 21.17, 18.33, 15.49, 9.80, 8.00, 4.00.

Table 1.

Elements of the matrix A of Fig. 1

1000903	1000901	1000902	1000894	1000900	998800	998450	998250
− 999701	− 999703	− 999702	− 999694	− 999700	−1001800	− 997950	−1001750
− 999603	− 999601	− 999602	− 999594	− 999600	−1001700	−1002050	− 998250
999801	999803	999802	999794	999800	997700	1001550	1001750
−1000897	−1000899	−1000898	−1000906	−1000900	− 998800	− 998450	− 998250
999699	999697	999698	999706	999700	1001800	997950	−1001750
999597	999599	999598	999606	999600	1001700	1002050	998250
− 999799	− 999797	− 999798	− 999806	− 999800	− 997700	−1001550	−1001750

Singular values of A

8×10^6, $5499.09\ (=300\times\sqrt{6\times 7\times 8})$, $5422.18\ (=350\times\sqrt{5\times 6\times 8})$,
$5291.50\ (=250\times\sqrt{7\times 8\times 8})$, $19.5959\ (=2\times\sqrt{3\times 4\times 8})$, 4

termined by the behavior of $(1-\alpha\sigma_r^2)^k \geqq (1-\sigma_r^2/\sigma_1^2)^k$. So the convergence is tediously slow unless the condition number $\sigma_1(A)/\sigma_r(A)$ is small enough. Theorem 2, Corollary 3 looks nice for determining rank A, but it is practically useless. We shall treat only Algorithm 1 in the following.

Our second observation is the accumulation of rounding errors, which occurs since Algorithm 1, as well as Algorithm 2, is not self-correcting. Put $X(k)=A^{\dagger}+Z$, where Z is an approximation error. Applying Algorithm 1 one step we get

$$\begin{aligned} X(k+1) = {} & A^{\dagger}+Z(I-\Pi_A)+(I-\Pi_{A^*})Z \\ & +(p-2)(I-\Pi_{A^*})Z(I-\Pi_A)+(\text{higer order of } Z)\,. \end{aligned}$$

So that Z does not vanish unless $Z \in \mathcal{R}(A^*)$ and $Z^* \in \mathcal{R}(A)$. Thus, the error in "the subspace orthogonal to $\Pi_{A^*}X\Pi_A$" will accumulate.

A method to cancel the components outside $\Pi_{A^*}X\Pi_A$ is to premultiply $A^*X^*(k)$ and postmultiply $X^*(k)A^*$ to $X(k)$ regarding the multipliers as estimates of Π_{A^*} and Π_A respectively. The multiplications may be needed at the final stage of iteration and they should be done in higer accuracy.

The third observation is the difficulty of terminating the iteration. The trouble occurs when the singular values of A clump into a few groups of different orders. For example, if a matrix of rank 6 has one singular values of order 10^6, three of order 10^3 and two of order 1, then trace $X(k)A$ becomes stationary near 1 at first, after some iterations begins to change and becomes again stationary near 4. The authors failed to find any practical algorithm giving a sequence of numbers which approaches to rank A from above (remember Theorem 1, Corollary 4). A rule, which is sometimes uneconomical, is to let the iteration continue long enough to make trace $X(k)A$ larger than the 'approximate rank.' Let ε be the maximum value such that machine computation of the logical expression $1+\varepsilon=1$ is true. If we disregard singular value such that $\sigma_i/\sigma_1<\delta$ then we need to repeat about k times, where k is such that

$$(1-\delta^2)^{p^k}=\varepsilon\ .$$

For $p=2$ and $\varepsilon=\delta=2^{-\mu}$, k is approximately equal to 2μ which is close to our experiences.

Figure 2 shows the behavior of trace $X(k)A$ in Theorem 1, Corollary 4. The matrix A and its singular values are shown in Table 1. And the values of the (1, 1) elements of $X(k)$ is shown with the true value in Table 2. The approximation of $X(k)$ becomes worse in later steps. Computation was done by IBM-7040 WATFOR with 27 bit man-

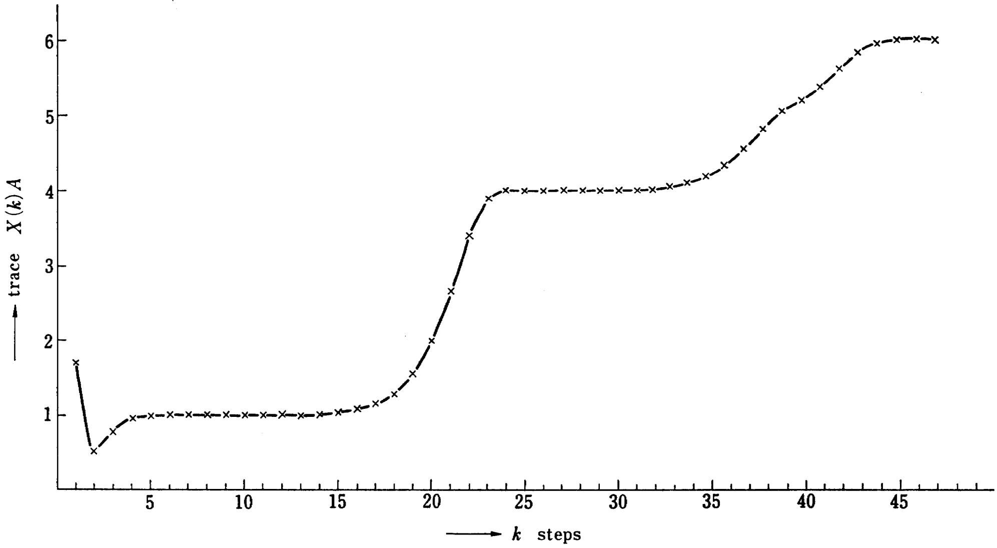

Fig. 2. trace $X(k)A$ in Theorem 1, Corollary 4 for the matrix of Table 1.

Table 2.

k	(1, 1) element of $X(k)$
—	—
42	.05815
43	.06640
44	.06797
45	.06813
46	.06856
47	.06946
—	—
true value	.06774

tissa and all inner products were computed by double precision.

Finally we remark on the choice of p. As discussed by Garnett, Ben-Israel and Yau [3], $p=3$ is the best choice if we consider the convergence rate per algebraic multiplication. When $X(k)$ is close to $A^{\dagger}$, however, the correction term $(I-X(k)A)^2X(k)$ becomes smaller than machine accuracy and the cubic-convergent procedure works actually as if it were quadratic convergent. So, in practice $p=2$ is better than $p=3$. A better procedure is to begin with $p=3$ and change to $p=2$ when the last term of (1) becomes small enough.

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

REFERENCES

[1] Ben-Israel, A and Cohen, D. (1966). On iterative computation of generalized inverses and associated projections, *SIAM Jour. Numer. Anal.*, **3**, 410-419.

[2] Boullion, T. L. and Odell, P. L. (1971). *Generalized Inverse Matrices*, Wiley-Interscience, New York.

[3] Garnett, J. M., III, Ben-Israel, A. and Yau, S. S. (1971). A hyperpower iterative method for computing matrix products involving the generalized inverse, *SIAM Jour. Numer. Anal.*, 8, 104-109.

[4] Petryshyn, W. V. (1967). On generalized inverses and on the uniform convergence of $(1-\beta K)^n$ with application to iterative methods, *Jour. Math. Anal. Appl.*, **18**, 417-439.

[5] Pringle, R. M. and Rayner, A. A. (1972). *Generalized Inverse Matrices with Applications to Statistics*, Charles Griffin, London.

[6] Rao, C. R. and Mitra, S. K. (1971) *Generalized Inverse Matrices and Its Applications*, Wiley, New York.

[7] Shinozaki, N., Sibuya, M. and Tanabe, K. (1972). Numerical algorithms for the Moore-Penrose inverse of a matrix: Direct methods, *Ann. Inst. Statist. Math.*, **24**, 193-203.

[8] Tanabe, K. (1971). An adaptive acceleration of general linear iterative processes for solving systems of linear equations, *Res. Memo. Inst. Statist. Math.*, No. 43.

[9] Zlobec, S. (1967). On computing the generalized inverse of a linear operator, *Glasnik Matematicki*, **2**(22), 265-271.