About Big Matrix Inversion

G. Malaschonok[1], I. Tchaikovsky[2]

[1]*National University of Kyiv-Mohyla Academy, Ukraine*

[2]*Lviv Polytechnic National University, Ukraine*

e-mail: malaschonok@ukma.edu.ua, ihortch@yahoo.com

Abstract. We consider three types of inverse matrices: inverse, pseudoinverse, and generalized inverse. And we discuss algorithms, which are applicable for commutative domains. The research is motivated by modern problems of supercomputing.

Keywords: inverse matrix, pseudoinverse matrix, generalized inverse matrix, commutative domain, supercomputing

1. On the boundary between big data and small data

From the point of view of computational mathematics, the size of the input problem largely determines the accumulation of computational error.

When the calculation error reaches the value of the sought numbers, then a revision of the computational algorithm is required. If algorithms demonstrate such an error starting from a certain matrix size, then this size can be considered a borderline between large and small data.

In paper [1], all matrix algorithms are classified into three classes:
(MA_1) the rational matrix algorithms,
(MA_2) the irrational matrix algorithms, which are expressed in radicals,
(MA_3) the irrational iterative matrix algorithms, which are not expressed in radicals.

For the experiment, the Cholesky algorithm for decomposition of a symmetric positive definite matrix was chosen, which belongs to class MA_2. To obtain the initial matrix for decomposition, triangular matrices with integer coefficients between 1 and 9 were randomly chosen and multiplied by their transposed matrix. Thus, the correct decomposition result was known in advance. The calculations were performed with double precision and 100 experiments were performed for matrices of the same size. Subtracting the matrix obtained in the decomposition from the exact one, the error was found as the largest number in absolute value.

Table 1 shows the results of such experiments. The first line shows the maximum error per 100 experiments, and the second line shows the average error.

Table 1. The value of the calculation error in the Cholesky decomposition algorithm

Matrix size	4×4	$8x8$	16×16	32×32	64×64
Maximum error value	$2 \cdot 10^{-13}$	10^{-10}	$3 \cdot 10^{-6}$	0.6	142
Average error value	$6 \cdot 10^{-15}$	$4 \cdot 10^{-12}$	$6 \cdot 10^{-8}$	0.01	7.9

As can be seen from the table, the average error reaches the value of the input numbers already for matrices of size 64.

These experiments show that size 64 is the boundary between big data and small data for this triangular decomposition problem.

2. Three types of inverse matrices

We consider three types of inverse matrices: inverse, pseudoinverse, and generalized inverse. We are interested in class 1 algorithms, since for these algorithms it is possible to overcome the problem of accumulating computational error if all calculations are carried out in the commutative domain. So we discuss algorithms,which are applicable for commutative domains.

Definition 1. *Matrix A^{-1} is the inverse of a square nondegenerate matrix A if two equalities*

$$A^{-1}A = AA^{-1} = I \tag{1}$$

are true.

Definition 2. *Matrix $A^{\times}$ is a pseudo inverse for matrix A if two equalities*

$$A^{\times} = A^{\times}AA^{\times},\ A = AA^{\times}A \tag{2}$$

are true.

Definition 3. *Matrix A^{+} is a generalized inverse (generalized inverse Moore-Penrose matrix) for matrix A if four equalities*

$$A^{+} = A^{+}AA^{+},\ A = AA^{+}A,\ (A^{+}A)^{T} = A^{+}A,\ (AA^{+})^{T} = AA^{+} \tag{3}$$

are satisfied.

If matrix A is a square and nondegenerate matrix, then all three types of inverse matrices coincide

$$A^{+} = A^{\times} = A^{-1}.$$

Let a matrix A of size $n \times m$ be decomposed as follows: $A = B \cdot C$, $B \in F^{n \times k}$, $C \in F^{k \times m}$, $rank(A) = rank(B) = rank(C) = k$.

It is easy to check that matrix

$$A^{+} = C^{T}(CC^{T})^{-1}(B^{T}B)^{-1}B^{T}, \tag{4}$$

is the generalized inverse matrix for the matrix A. This idea was first expressed by Vera Kublanovskaya in 1965 [2].

3. Algorithms for commutative domains

The first exact matrix inversion algorithm, in which the inverse of the matrix is represented as the ratio of the adjoint matrix and the determinant, and only n^3 integer operations were used, was presented in 1981 and published in 1983 [3].

The most famous of the previous works is [4]. However, the algorithm proposed in [4] did not allow finding the adjoint matrix or the numerators and denominators of fractions, which are elements of the inverse matrix or the solution of a system of linear equations, due to the presence of many unnecessary factors. Apparently, this fact is not generally known, since the work [4] is still the most cited, and the work [3] is little known to specialists.

With regard to additional permutations, in this issue, algorithm [3] is similar to the standard Gaussian elimination algorithm. As you know, if the Gaussian elimination algorithm is applied to a matrix of the form $[A, I]$, here I is the identity matrix, then it allows you to find the inverse matrix for the matrix A. If at the same time a leading element equal to zero is encountered, then it is necessary to perform permutations and preserve the permutation matrices. At the end of the calculation, these permutation matrices are applied to obtain the desired adjoint matrix for matrix A.

If the original matrix is a dense integer matrix jf size n and standard algorithms for integers are used, then the total bit complexity will be $\sim n^5$, and if you use the Chinese remainder theorem, then the total bit complexity will be $\sim n^4$.

Just like the Gauss algorithm, this algorithm is efficient for sequential algorithms and for parallel algorithms that are used in computers with shared memory.

However, it is not efficient for modern supercomputers with tens and hundreds of thousands of processors, as finding a pivot and moving rows and columns create delays that can destroy all the advantages of a supercomputer.

AdjDet algorithm

The search for an algorithm effective for a supercomputer continued for a long time. A new block-recursive algorithm *AdjDet* was published in [5] and [6].

The main advantage of the new algorithm was the rejection of permutations of rows and columns of the matrix in favor of local multiplication by the permutation matrix for a separate block.

It should be noted that there are other advantages of this algorithm: less complexity, the ability to calculate the kernel of the matrix operator and the generalized inverse matrix.

Let A be a matrix of rank r. If A is a square matrix of size r, then $A^{-1} = Adj(A)/Det(A)$. If this is not the case, then we can calculate the generalized inverse matrix.

Algorithm *AdjDet* returns permutation matrices P and Q that move the largest nondegenerate minor to the upper left corner. Let A_0 be such nondegenerate minor of size r, $d = \det(A_0)$, located in the upper left corner. Let A_U be a submatrix formed by the first r rows, A_L be a submatrix formed by the first r columns. Then the matrix A is decomposed into the product of three matrices: $A = (1/d)A_L Adj(A_0)A_U$ Hence,

$$A^+ = (1/d)PA_U^*(A_U A_U^*)^{-1}A_0(A_L^*A_L)^{-1}A_L^*Q.$$

Here matrices $A_U A_U^*$ and $A_L^* A_L$ of size $r \times r$ are invertible.

LDU algorithm

Another algorithm for calculating inverse matrices was recently obtained [7]. This is a block-recursive LDU triangular decomposition algorithm: $A = LDU$. Triangular matrices L and U belong to the same commutative domain as matrix A, and the matrix of weighted permutations D has the same rank as the matrix A.

This algorithm computes the inverse matrices M and W to the matrices L and U: $LdM = I$ and $WdU = I$. All these matrices have full rank. Matrix d is a matrix of weighted permutations and it associated with D. All other matrices belong to the commutative domain.

If matrix A is invertible, then its inverse is calculated as follows:

$$A^{-1} = U^{-1}D^+L^{-1} = WdD^+dM = WDM.$$

If matrix A is not invertible, then its psevdo inverse matrix is equal $A^\times = WDM$. We can easily check the fulfillment of equalities (2):

$$A^{\times}AA^{\times} = U^{-1}D^{+}L^{-1}LDUU^{-1}D^{+}L^{-1} = U^{-1}D^{+}L^{-1} = A^{\times}$$

$$AA^{\times}A = LDUU^{-1}D^{+}L^{-1}LDU = LDU = A.$$

However, calculating the generalized inverse matrix requires additional effort.

Let the matrix A be decomposed into the product of three matrices $A = BDC$, that have the same rank r. Where D is the matrix of weighted permutations, C is the matrix of columns corresponding to the matrix D, B is the matrix of strings corresponding to the matrix D. Then we can find the generalized inverse matrix thanks to the basic form:

$$A^{+} = C^{*}(CC^{*})^{-1}D^{+}(B^{*}B)^{-1}B^{*}. \tag{5}$$

We denote by I and J diagonal matrices, whose rank is the same as the rank of the matrix D, and which correspond to nonzero rows and columns of matrix D: $IDJ = D$.

Then we can write $A = (LI)D(JU) = BDC$ with $LI = B$ and $JU = C$ and use the basic form (5), in which matrices CC^{*} and $B^{*}B$ are invertible matrices of rank r.

References

1. *Malaschonok G.* Recursive matrix algorithms, distributed dynamic control, scaling, stability. Proc. of 12th Int. Conf. on Comp. Sci. and Information Technologies (CSIT-2019). September, Yerevan. P. 23–27.

2. *Kublanovskaya V.N.* Evaluation of a generalized inverse matrix and projector. USSR Computational Mathematics and Mathematical Physics. 1966. Vol. 6, No. 2. P. 179–188. (In Russian)

3. *Malashonok G.I.* Solution of a system of linear equations in an integral domain. USSR J. of Comput. Math. and Math. Phys. 1983. Vol. 23, No. 6. P. 1497–1500. arXiv:1711.09452

4. *Bareiss E.N.* Sylvester's identity and multistep integer-preserving Gaussian elimination. Math. Comput. 1968. Vol. 22. P. 565–578.

5. *Malaschonok G.I.* On computation of kernel of operator acting in a module. Tambov University Reports. Natural and Technical Sciences. 2008. Vol. 13, part. 1. P. 129–131. (In Russian)

6. *Malaschonok G. and Ilchenko E.* Recursive matrix algorithms in commutative domain for cluster with distributed memory. 2018 Ivannikov Memorial Workshop (IVMEM), Yerevan, Armenia. P. 40–46. doi: 10.1109/IVMEM.2018.00015. arXiv:1903.04394

7. *Malaschonok G.* LDU-factorization. E-print 2011.04108. 2020. P. 1–16. arXiv:2011.04108