Gertrude M. Cox

THE ABBREVIATED DOOLITTLE METHOD FOR MATRIX INVERSION

By

Paul Peach

Institute of Statistics
Mimeo. Series #28
For General Distribution

Expository note by

Paul Peach
Institute of Statistics
Raleigh, N. C.

The Abbreviated Doolittle Method for Matrix Inversion

The Abbreviated Doolittle method is described by Paul S. Dwyer in a paper on "The Solution of Simultaneous Equations" (Psychometrika, Vol. 6, No. 2, pp. 101 - 129). In that paper, Dwyer describes a number of methods for solving simultaneous equations and inverting matrices. The Abbreviated Doolittle technique is credited to Horst and Waugh.

In December, 1945, Dwyer published in the "Journal of the American Statistical Association" a method, apparently original with him, which he called the square root method. This method was adopted by Duncan and Kenney in their 1946 monograph "On the Solution of Normal Equations and Related Topics" (Edwards Bros., Ann Arbor, Mich.) as being the most efficient computational method available.

The square root method involves copying fewer entries than the Abbreviated Doolittle; if we include a check column, the square root method saves copying $\frac{1}{2}(n + 1)(n + 2)$ entries. On the debit side, the square root method requires a final division for each answer; this division is avoided by the Doolittle method. In merely solving a set of simultaneous equations only n such final divisions are needed, and though it takes longer to perform a division than to copy a number, yet the overall result is a saving in time for the square root method. But in matrix inversion a division must be performed for each element in the inverse matrix calculated; this means usually $\frac{1}{2}n(n + 1)$ divisions. The time required for all these divisions is much greater than that saved by not having to copy an approximately equal number of entries, and accordingly for matrix inversion the square root method, though having an appearance of great efficiency due to the small number of entries, actually takes longer than the Abbreviated Doolittle. (The computational procedures are otherwise approximately equal for the two methods.)

It is the writer's belief that the Abbreviated Doolittle method is easier to teach to a computer, and is less likely to lead to errors in the forward solution, than the square root method, since reciprocals (used in the Abbreviated Doolittle) are more easily computed than square roots. In the back solution, however, the square root method, through its small number of entries, may be somewhat easier to remember. Once the methods are mastered, these differences become unimportant.

The Abbreviated Doolittle method assumes that there is available a modern calculating machine that will add and subtract products automatically. In its most efficient form, it omits the matrix of 1's and 0's usually shown at the right of the matrix to be inverted. The procedure of solution is shown in Table I; the notation is the present writer's, not Dwyer's. A five-by-five matrix has been chosen for the illustration. The right-hand column (j) is a check column, and each entry is calculated in the same manner as other entries in the same row, save for the first n (here 5) which are merely the sums of rows of the (a) matrix, including of course elements omitted from the table because of symmetry. After the j element has been calculated, it should be compared with the sum of elements in the same row, and should agree within one or two units in the last place.

TABLE I

The first n rows contain the matrix to be inverted.	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	j_1
	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	j_2
			a_{33}	a_{34}	a_{35}	j_3
				a_{44}	a_{45}	j_4
					a_{55}	j_5
Forward solution begins here	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	j_6
	1	B_{12}	B_{13}	B_{14}	B_{15}	j_7
		A_{22}	A_{23}	A_{24}	A_{25}	j_8
		1	B_{23}	B_{24}	B_{25}	j_9
			A_{33}	A_{34}	A_{35}	j_{10}
			1	B_{34}	B_{35}	j_{11}
				A_{44}	A_{45}	j_{12}
				1	B_{45}	j_{13}
Forward solution ends here					A_{55}	j_{14}
The last n rows contain the inverse matrix	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	
		c_{22}	c_{23}	c_{24}	c_{25}	
			c_{33}	c_{34}	c_{35}	
				c_{44}	c_{45}	
					c_{55}	

It is advisable here to calculate the products of the (a) and (c) matrices, which should of course be the unit matrix, allowing for errors of rounding. This calculation serves as a partial check, but its chief usefulness is as the first step in the Hotelling iterative method for improving the accuracy of the solution. It is usually desirable to apply one cycle of iteration. Incidentally, although both the (a) and (c) matrices are symmetrical, their product will not usually be so, and care must be taken to distinguish O_{ij} from O_{ji}.

The entries in Table I are obtained as follows:

Rows 1 to 5 present the matrix to be inverted. The j's are obtained by summing the elements of each row, not forgetting to supply elements omitted because of symmetry.

Row 6 is the same as Row 1.

Row 7 is obtained by dividing each element of Row 6 by A_{11}. This routine applies throughout the forward solution; the B's are obtained from the A's by dividing by the leading terms A_{ii}.

Row 8 is calculated from Rows 2, 6 and 7:

$$A_{22} = a_{22} - A_{12}B_{12} ,$$

$$A_{23} = a_{23} - A_{12}B_{13} \qquad \underline{\text{or}} \qquad A_{23} = a_{23} - A_{13}B_{12} .$$

(The products $A_{12}B_{13}$ and $A_{13}B_{12}$ may be called diagonal products; they are clearly equal, except for errors of rounding. While it does not matter much which we use, significant figures will be conserved by choosing the pair most nearly equal. For brevity, in the following equations only one of the possible choices of products will be indicated.)

$$A_{24} = a_{24} - A_{12}B_{14} ,$$

$$A_{25} = a_{25} - A_{12}B_{15} ,$$

$$j_8 = j_2 - A_{12}j_7 .$$

Row 9 is obtained from Row 8 by dividing each element by A_{22}, i. e.,

$$B_{23} = A_{23}/A_{22} \text{ , etc.}$$

Row 10 is calculated from Rows 3, 6, 7, 8 and 9, as follows:

$$A_{33} = a_{33} - A_{13}B_{13} - A_{23}B_{23} ,$$

$$A_{34} = a_{34} - A_{13}B_{14} - A_{23}B_{24},$$

$$A_{35} = a_{35} - A_{13}B_{15} - A_{23}B_{25} ,$$

$$j_{10} = j_3 - A_{13}j_7 - A_{23}j_9 .$$

The procedure should now be fairly evident. Equations for the remaining A's and j's are

$$A_{44} = a_{44} - A_{14}B_{14} - A_{24}B_{24} - A_{34}B_{34} ,$$

$$A_{45} = a_{45} - A_{14}B_{15} - A_{24}B_{25} - A_{34}B_{35} ,$$

$$j_{12} = j_4 - A_{14}j_7 - A_{24}j_9 - A_{34}j_{11} ,$$

$$A_{55} = a_{55} - A_{15}B_{15} - A_{25}B_{25} - A_{35}B_{35} - A_{45}B_{45} ,$$

$$j_{14} = j_5 - A_{15}j_7 - A_{25}j_9 - A_{35}j_{11} - A_{45}j_{13} .$$

The forward solution is now complete.

The back solution starts with the computation of c_{55} and builds up the last column of the inverse matrix from the bottom to the top. The element c_{55} is merely the reciprocal of A_{55}. We have

$$c_{55} = 1/A_{55} \; ,$$

$$c_{45} = -B_{45}c_{55}$$

$$c_{35} = -B_{34}c_{45} - B_{35}c_{55} \; ,$$

$$c_{25} = -B_{23}c_{35} - B_{24}c_{45} - B_{25}c_{55} \; ,$$

$$c_{15} = -B_{12}c_{25} - B_{13}c_{35} - B_{14}c_{45} - B_{15}c_{55} \; .$$

At this point it is worth while to calculate

$$a_{15}c_{15} + a_{25}c_{25} + a_{35}c_{35} + a_{45}c_{45} + a_{55}c_{55} \; ,$$

which should be equal to 1, or nearly.

The next step is calculating the diagonal element c_{44}. The routine for the diagonal elements c_{ii} is different for that for the non-diagonal elements. We require the reciprocals of the A_{ii}, and it is frequently convenient to calculate them in the course of the forward solution and write them down. These reciprocals should not, however, be used to compute the B's; it is more efficient to divide each A as it is calculated, as this avoids setting up the A a second time in the machine. We have then

$$c_{44} = 1/A_{44} - B_{45}c_{45}.$$

The calculation of the rest of the fourth column then proceeds:

$$c_{34} = -B_{34}c_{44} - B_{35}c_{45} \; ,$$

$$c_{24} = -B_{23}c_{34} - B_{24}c_{44} - B_{25}c_{45} \; ,$$

$$c_{14} = -B_{12}c_{24} - B_{13}c_{34} - B_{14}c_{44} - B_{15}c_{45} \; .$$

For the diagonal element c_{33} we have

$$c_{33} = 1/A_{33} - B_{34}c_{34} - B_{35}c_{35} \; ,$$

and the rest of the routine should be obvious.

Some computers, in going from the A's to the B's, change the sign of the B elements, each B_{ii} becoming -1 instead of 1. There is no harm in this, but there seems very little point to it either. If the signs of the B's are changed, then all the minus signs in the above formulas become plus. The difference in the resulting solution will be in notation only; the operations in the machine will be identical. Changing the signs of the B's does lead to a somewhat simpler notation if summations are to be represented as usual by the Greek sigma and subscripts; the summations can then be represented for the most part without the use of minus signs.

The following numerical example follows the preceding pattern, except for the omission of four entries in the check column. Instead of calculating all the A's in a row, then all the B's, it is more efficient to calculate each B at once as soon as the corresponding A is known. A little reflection will undoubtedly convince the student that these j's can safely be omitted.

TABLE II -- Numerical Example

	1.	.011466	.047447	.111634	.019635	1.190182
(A)		1.	.686231	.623772	.426464	2.747933
The Matrix to			1.	.648112	.405864	2.787654
be inverted				1.	.428778	2.812296
					1.	2.280741
Forward	1.	.011466	.047447	.111634	.019635	1.190182
Solution	1.	.011466	.047447	.111634	.019635	1.190182
begins here						
		.999869	.685687	.622492	.426239	
		1.	.685777	.622574	.426295	2.734645
			.527520	.215924	.111627	
			1.	.409320	.213504	1.622826
				.511609	.115120	
				1.	.225016	1.225017
Forward solution ends here					.767961	.767961

	1.018579	.095405	-.006543	-.176556	.017672
(C_o)		2.173726	-.994399	-.608763	-.264274
			2.242337	-.764494	-.158082
The approximate				2.020549	-.293004
Inverse Matrix					1.302149

(AC_o)	.99999981	.00000002	-.00000008	-.00000019	-.00000020
	-.00000020	.99999983	.00000091	-.00000064	-.00000016
	-.00000025	-.00000006	1.00000161	-.00000082	-.00000053
	-.00000022	.00000005	.00000076	.99999863	.00000008
	-.00000030	.00000022	-.00000019	.00000019	.99999918

$(2 - AC_o)$	1.00000019	-.00000002	.00000008	.00000019	.00000020
	.00000020	1.00000017	-.00000091	.00000064	.00000016
	.00000025	.00000006	.99999839	.00000082	.00000053
	.00000022	-.00000005	-.00000076	1.00000137	-.00000008
	.00000030	-.00000022	.00000019	-.00000019	1.00000082

(C_1)	1.01857918	.09540500	-.00654286	-.17655600	.01767225
	.09540499	2.17372640	-.99439896	-.60876327	-.26427431
A closer	-.00654286	-.99439896	2.24233485	-.76449383	-.15808105
approximation	-.17655599	-.60876318	-.76449382	2.02055077	-.29300495
to the inverse	.01767224	-.26427433	-.15808103	-.29300495	1.30214997

The Doolittle solution is of course complete as soon as (C_o) is computed. It is convenient to calculate the product (AC_o), though, as implied on Page 2, such computation gives at best an imperfect check on the accuracy of the arithmetical work. However, from the product (AC_o) we can proceed to obtain an improved approximation using the Hotelling iterative method, and the work is indicated above. In practice we should not write down (AC_o) but instead $(2 - AC_o)$ since the only work involved -- subtracting the diagonal elements from 2, and changing signs of non-diagonal elements -- is readily performed mentally.