Matrix Inversion by a Monte Carlo Method [1]

The following unusual method of inverting a class of matrices was devised by J. VON NEUMANN and S. M. ULAM. Since it appears not to be in print, an exposition may be of interest to readers of $MTAC$. The method is remarkable in that it can be used to invert a class of n-th order matrices (see final paragraph) with only n^2 arithmetic operations in addition to the scanning and discriminating required to play the solitaire game. The method therefore appears best suited to a human computer with a table of random digits and no calculating machine. Moreover, the method lends itself fairly well to obtaining a single element of the inverse matrix without determining the rest of the matrix. The term "Monte Carlo" refers to mathematical sampling procedures used to approximate a theoretical distribution [see $MTAC$, v. 3, p. 546].

Let B be a matrix of order n whose inverse is desired, and let $A = I - B$, where I is the unit matrix. For any matrix M, let $\lambda_r(M)$ denote the r-th proper value of M, and let M_{ij} denote the element of M in the i-th row and j-th column. The present method presupposes that

(1)
$$\max_r |1 - \lambda_r(B)| = \max_r |\lambda_r(A)| < 1.$$

When (1) holds, it is known that

(2)
$$(B^{-1})_{ij} = ([I - A]^{-1})_{ij} = \sum_{k=0}^{\infty} (A^k)_{ij}.$$

The Monte Carlo method to compute $(B^{-1})_{ij}$ is to play a solitaire game $\mathbf{G}_{ij}$ whose expected payment is $(B^{-1})_{ij}$. According to a result of KOLMOGOROFF [2] on the strong law of large numbers, if one plays such a game repeatedly, the average payment for N successive plays will converge to $(B^{-1})_{ij}$ as $N \to \infty$, for almost all sequences of plays. The rules of the game will be expressed in terms of balls in urns, but a computer would probably use a table of random digits.

For $1 \leqq i, j \leqq n$ pick probabilities $p_{ij} \geqq 0$ and corresponding "value factors" v_{ij}, subject only to the conditions that $p_{ij}v_{ij} = a_{ij}$ and $\sum_{j=1}^{n} p_{ij} < 1$. Let the "stop probabilities" p_i be defined by the relations $p_i = 1 - \sum_{j=1}^{n} p_{ij}$. Take n urns. In the i-th urn $U_i (i = 1, 2, \cdots, n)$ put an assortment of $n + 1$ different types of balls. Each ball of the j-th type is marked "j," and will be drawn from U_i with probability $p_{ij}(j = 1, 2, \cdots, n)$. The $(n + 1)$-th type of ball is marked "STOP," and will be drawn from U_i with probability p_i.

For the game $\mathbf{G}_{ij}$ now to be defined, the value of a play is a random variable G_{ij} whose expectation will be proved to be $(B^{-1})_{ij}$. First draw a ball from U_i (all drawings are with replacement). If it is a STOP ball, the payment G_{ij} is 0 (if $i \neq j$) or p_j^{-1} (if $i = j$). Otherwise the ball must carry a mark $i_1(1 \leqq i_1 \leqq n)$, and one is next to draw a ball from U_{i_1}, which in turn tells one whether to stop or draw again. One follows this treasure hunt from urn to urn until a STOP ball is first drawn—say from U_h on the k-th drawing. If $h \neq j$, the payment G_{ij} is 0. If $h = j$, suppose one has

127

arrived at U_j via a route $i = i_0 \to i_1 \to i_2 \to \cdots \to i_{k-1} \to j$, whose abbreviation is ρ. Then the payment G_{ij} is defined to be

$$V_\rho p_j^{-1} = v_{i_0 i_1} v_{i_1 i_2} \cdots v_{i_{k-1} j} p_j^{-1}.$$

THEOREM 1. *If* (1) *holds,* G_{ij} *has the expectation* $(B^{-1})_{ij}$.

Proof. The probability of following the route ρ and then drawing a STOP ball is

$$P_\rho p_j = p_{i_0 i_1} p_{i_1 i_2} \cdots p_{i_{k-1} j} p_j.$$

The expected payment in the game $\mathbf{G}_{ij}$ is

$$E(G_{ij}) = \sum_\rho (P_\rho p_j)(V_\rho p_j^{-1}) = \sum_\rho P_\rho V_\rho,$$

where the sum is extended over all routes ρ from i to j. Since $p_{ij} v_{ij} = a_{ij}$,

$$E(G_{ij}) = \delta_{ij} + \sum_{k=1}^{\infty} \sum_{i_1=1}^{n} \cdots \sum_{i_{k-1}=1}^{n} a_{i_0 i_1} a_{i_1 i_2} \cdots a_{i_{k-1} j}$$

$$= I_{ij} + \sum_{k=1}^{\infty} (A^k)_{ij},$$

where δ_{ij} is the KRONECKER symbol. By (2) the theorem is proved.

Let us calculate the variance σ_{ij}^2 of the payment G_{ij} about its expected value. Let R be the matrix $(a_{ij} v_{ij}) = (p_{ij} v_{ij}^2)$. If $\max_i |\lambda_i(R)| \geqq 1$, the proof below can be modified[3] to show that $\sigma_{ij} = \infty$. Assume $\max_i |\lambda_i(R)| < 1$, and define $(I - R)^{-1} = T$.

THEOREM 2. *If* $\max_i |\lambda_i(R)| < 1$,

$$\sigma_{ij}^2 = T_{ij} p_j^{-1} - (B^{-1})_{ij}^2.$$

Proof. Using the above notation

$$\sigma_{ij}^2 = E\{G_{ij} - (B^{-1})_{ij}\}^2 = E(G_{ij}^2) - (B^{-1})_{ij}^2,$$
$$= \sum_\rho (P_\rho p_j)(V_\rho p_j^{-1})^2 - (B^{-1})_{ij}^2$$
$$= p_j^{-1} \sum_\rho P_\rho V_\rho^2 - (B^{-1})_{ij}^2.$$

If $\max_i |\lambda_i(R)| < 1$, the end of the proof of Theorem 1 can be modified to show that the last sum is T_{ij}. This proves Theorem 2.

One can of course compute one whole row of B^{-1} at once by playing the games $\mathbf{G}_{i\alpha}(\alpha = 1, 2, \cdots, n)$ simultaneously. It may even be practicable to play all the games $\{\mathbf{G}_{ij}\}$ $(i, j = 1, 2, \cdots, n)$ simultaneously, using not only the full route $i_0 \to i_1 \to i_2 \to \cdots \to i_{k-1} \to j$, but also each partial route $i_r \to i_{r+1} \to \cdots \to i_{k-1} \to j$ as a separate play. In any case the arithmetic can be reduced by accumulating the totals of V_ρ for a series of plays and dividing by $N p_j$ at the end, to obtain the average payment after N plays.

If $a_{ij} \geqq 0$ and $\sum_{j=1}^{n} a_{ij} < 1$, one can take $p_{ij} = a_{ij}$ and $v_{ij} \equiv 1$. Then $V_\rho \equiv 1$, and the computer has only to make a series of plays and keep a tally of the frequency with which the STOP ball was drawn in U_j $(j = 1, 2, \cdots, n)$.[4] If N is selected to be a power of 10, the only arithmetic required

to invert B is n divisions of a vector by p_j. When $v_{ij} \equiv 1$, $A = R$ and $T = B^{-1}$, and hence $\sigma_{ij}^2 = p_j^{-1}(B^{-1})_{ij}[1 - p_j(B^{-1})_{ij}]$, corresponding to the variance of the binomial distribution.

GEORGE E. FORSYTHE
RICHARD A. LEIBLER

NBS Institute for Numerical Analysis
Univ. of California, Los Angeles
2901, 18th St.
Washington 9, D. C.

Maximum-Interval Tables

Both the article by HERGET & CLEMENCE [*MTAC*, v. 1, p. 173–176] and the note by MILLER [*MTAC*, v. 1, p. 334] on optimum-interval tables ignore the possibility of a continuously variable interval. It is of some interest to examine the reduction in the number of entries made possible by what might, by analogy, be termed "maximum-interval" tables. Using the principles of optimum-interval tables, with the modifications suggested by Miller, the tabular values of the argument are no longer restricted to terminating decimals so that the interval may be allowed to assume at each point the maximum value consistent with the stated allowable error.

The chief objection to a punched-card table in this form is that all (or nearly all) of the digits in the argument will have to be used in the interpolating factor. In some cases this objection could be overcome by inserting additional cards corresponding to values of the argument terminating in the appropriate number of zeros, or by splitting the whole range into a number of sub-ranges, in each of which the allowable error is varied slightly to make tabular arguments coincide with the end-points; or, of course, by an additional operation of subtraction. Generally, however, the saving in cards is not worth the additional cost of preparation and the resulting complication.

Herget in *The Computation of Orbits* [see *MTAC*, v. 3, p. 418–9] gives an optimum-interval table of $x^{-3/2}$ using quadratic interpolation, with a note "This is the first time such a table has ever been printed for use with a hand calculating machine." A human computer can easily exercise the requisite judgment to use the continuously variable intervals of a maximum-interval table; in the simplest case it will involve nothing more serious than allowing the interpolating factor occasionally to exceed unity in a particular digit. A punched-card machine can only do this at the expense of a separate operation. There may, therefore, be a use for variable interval tables in computation by desk machines.