Monte Carlo Matrix Calculation with Punched Card Machines

FORSYTHE & LIEBLER have described matrix inversion by a Monte Carlo method [*MTAC*, v. 4, p. 127]. While they state that the method is "best suited for a human computer with a table of random digits and no calculating machine," the principle has been adapted for use with punched card machines. The method has been extended to finding powers of a matrix. While an IBM 405 tabulator was used in this work, there is no reason why a newer machine cannot be used.

The class of matrices $B = (b_{ij})$ that may be inverted by this method is limited to those in which

(1) the largest latent root of $A^* = (a_{ij}^*)$ is less than unity, where a_{ij}^* is the absolute value of $a_{ij} = \delta_{ij} - b_{ij}$ and

(2) the sum of the elements in any row of A^* is less than unity

In certain cases these restrictions may be removed.

In this modification of the method of Forsythe and Liebler, the illustrative urns containing numbered balls are replaced by a group of punched cards. Each card is divided into two principal fields: a randomizing field consisting of 30 to 50 random digits and a playing field prepared for each matrix as described below. Each of the n card columns of the playing field of the deck corresponds to one of the urns. Each of the n different digits punched in the card column corresponds to the numbered balls in that urn. An x-punch in a card column corresponds to a "stop ball" in the urns of the original description. The deck of cards is treated as a unit and is so prepared that the probability of a j appearing in the ith card column is p_{ij}. In the case where A has both positive and negative elements, a value of -1 is attached to each draw corresponding to a negative element, $+1$ to all others. The negative value is indicated by a 12 punch appearing in the corresponding card column above the row punch.

In terms of the original reference, n games are played simultaneously to yield one row of the desired inverse B^{-1}. The passage of each card through the machine constitutes one draw in the game.[1] The machine is wired so that the original gambling procedure is followed. Instructed by a digit emitter, the machine reads the card column corresponding to the ith row. The digit punched in that column activates the mechanism which selects one column of the next card for reading. The digit in each column selected causes the reading of the corresponding column in the following card. This procedure is continued until an x is read. The column from which the x was read scores ± 1, all others score zero. As the values are to be multiplied together, the score is -1 if an odd number of negative draws was made, otherwise the score is $+1$. A designated counter keeps count of the total games played. The machine does not stop after each game, but starts anew on the card following the x card, again instructed by the digit emitter. After the entire deck of cards passes through, a total is taken and the machine prints total draws, total games and net score for each of the n

115

games. The actual elements of the inverse matrix are obtained by dividing by the product of the stop probability times the number of games played. If the stop probability can be taken as 0.1, and the number of games as an exact power of 10, the machine will actually list the elements found. By playing successively, starting with $i = 1$ and continuing through $i = n$, the entire inverse will be listed with the elements in their correct positions.

By a simple modification, powers of certain matrices may be found by playing a similar Monte Carlo game. In this case, the stop probabilities are no longer used to terminate each game. Instead, exactly $r + 1$ draws are made and the scoring is performed as before. As all possible routes from i to j consisting of exactly r steps will be covered in a large number of games, this is tantamount to raising the matrix to the rth power. The restriction on the size of the largest latent root no longer applies here, but it is required that $\sum_{j=1}^{n} a_{ij} = 1$. The instruction to score every $(r + 1)$st draw may be given the machine by interspersing control cards by means of a counting collator or by sequencing using interlocked selector relays as described below.

The order of matrices that may be raised to powers or inverted depends on the selector capacity of the machine. For general functions, one single-position selector, one three-position selector and one ten-position selector are required as well as two digit-selectors. For matrices of the nth order, n two-position selectors and n single-position selectors are also needed.

The advantages inherent in this method are:

1. It may be used with a common machine present in nearly all punched card equipped installations.

2. The actual working procedure is far simpler than for any previously reported matrix inversion method.

3. Approximate inverses may be obtained rapidly. The accuracy of a good approximation to an inverse may be easily improved.[2]

4. Using a machine with large selector capacity,[3] approximate inverses and powers may be obtained for high order matrices. In going from a matrix of the nth order to one of the $(2n)$th order, twice as many cards should go through the machine in twice the number of passes. Thus, as originally stated, the number of operations is proportional to the second instead of the third power of the order, as is otherwise the case.

5. The process of finding all powers from 2 to r is an exceedingly simple one. It should be useful in evaluation of latent roots and vectors.

Example 1—Inversion of a seventh order symmetrical matrix

$$B = \begin{vmatrix} .7 & -.1 & -.1 & -.1 & -.1 & -.1 & -.1 \\ -.1 & .7 & -.1 & -.1 & -.1 & -.1 & -.1 \\ -.1 & -.1 & .7 & -.1 & -.1 & -.1 & -.1 \\ -.1 & -.1 & -.1 & .7 & -.1 & -.1 & -.1 \\ -.1 & -.1 & -.1 & -.1 & .7 & -.1 & -.1 \\ -.1 & -.1 & -.1 & -.1 & -.1 & .7 & -.1 \\ -.1 & -.1 & -.1 & -.1 & -.1 & -.1 & .7 \end{vmatrix}$$

$$A = \begin{vmatrix} .3 & .1 & .1 & .1 & .1 & .1 & .1 \\ .1 & .3 & .1 & .1 & .1 & .1 & .1 \\ .1 & .1 & .3 & .1 & .1 & .1 & .1 \end{vmatrix} \qquad p_j = \begin{vmatrix} .1 \\ .1 \\ .1 \end{vmatrix}$$
$$\text{etc.} \qquad\qquad\qquad\qquad \text{etc.}$$

$$B^{-1} = |\,2.67 \quad 1.33 \quad 1.66 \quad 1.33 \quad 1.55 \quad 1.00 \quad 1.33\,|$$
$$\text{etc.}$$
$$\text{(90 games per element)}$$

$$B^{-1} = |\,2.536 \quad 1.224 \quad 1.288 \quad 1.275 \quad 1.263 \quad 1.263 \quad 1.263\,|$$
$$\text{etc.}$$
$$\text{(784 games per element)}$$

$$B^{-1} = |\,2.500 \quad 1.250 \quad 1.250 \quad 1.250 \quad 1.250 \quad 1.250 \quad 1.250\,|$$
$$\text{etc.}$$
$$\text{(calculated values)}$$

Example 2—A seventh order symmetrical matrix raised to the seventh power.

$$A = \begin{vmatrix} .88 & .02 & .02 & .02 & .02 & .02 & .02 \\ .02 & .88 & .02 & .02 & .02 & .02 & .02 \\ .02 & .02 & .88 & .02 & .02 & .02 & .02 \\ .02 & .02 & .02 & .88 & .02 & .02 & .02 \\ .02 & .02 & .02 & .02 & .88 & .02 & .02 \\ .02 & .02 & .02 & .02 & .02 & .88 & .02 \\ .02 & .02 & .02 & .02 & .02 & .02 & .88 \end{vmatrix}$$

$$A^7 = \begin{vmatrix} .440 & .120 & .088 & .072 & .080 & .088 & .112 \\ .121 & .427 & .096 & .081 & .081 & .096 & .096 \\ .111 & .079 & .428 & .103 & .095 & .087 & .095 \end{vmatrix}$$
$$\text{etc.}$$
$$\text{(125 games per element)}$$

$$A^7 = |\,.4388 \quad .0933 \quad .0954 \quad .0952 \quad .0946 \quad .0946 \quad .0946\,|$$
$$\text{etc.}$$
$$\text{(825 games per element)}$$

$$A^7 = |\,.4410 \quad .0932 \quad .0932 \quad .0932 \quad .0932 \quad .0932 \quad .0932\,|$$
$$\text{etc.}$$
$$\text{(calculated values)}$$

Example 3—A seventh order nonsymmetrical matrix raised to the sixth power.

$$A = \begin{vmatrix} .88 & .02 & .02 & .02 & .02 & -.02 & .02 \\ .02 & .88 & .02 & .02 & .02 & -.02 & .02 \\ .02 & .02 & .88 & .02 & .02 & -.02 & .02 \\ .02 & .02 & .02 & .88 & .02 & -.02 & .02 \\ .02 & .02 & .02 & .02 & .88 & -.02 & .02 \\ .02 & .02 & .02 & .02 & .02 & -.02 & .88 \\ -.02 & -.02 & -.02 & -.02 & -.02 & .88 & -.02 \end{vmatrix}$$

$A^6 = |$ .4683   .0739   .0880   .0704   .0704 $-$.0422 $-$.0523$|$
(568 games per element)

$$A^6 = \begin{vmatrix} .4782 & .0736 & .0736 & .0736 & .0736 & -.0025 & -.0025 \\ .0736 & .4782 & .0736 & .0736 & .0736 & -.0025 & -.0025 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ -.0025 & -.0025 & -.0025 & -.0025 & -.0025 & -.0025 & .4782 \end{vmatrix}$$

(calculated values)

Details of the procedure.—*Card Preparation.*

A. Random numbers may be keypunched from standard tables[4] or pseudo-random numbers may be produced [*MTAC*, v. 5, p. 4].

B. Preparation of a group of cards correctly punched in the playing field may be accomplished as follows.

1. Prepare the needed table of probabilities by listing the absolute values of the matrix with rows and columns transposed. (To prepare cards for matrix inversion, include the stop probabilities as the final row in the table.) The probabilities in each column (including stop probabilities if present) should add to unity.

2. An effective, but tedious, procedure for preparing the playing field for a single card column would be to partition the deck by actual count into n (or $n + 1$ in the case of inversion) groups each containing $p_{ij}\%$ of the card count. For each group the appropriate value of j would be gang punched into the card column representing i. A 12-punch is included when a value of -1 is attached to the probability and an x-punch represents the stop cards.

3. This may also be done by preparing a table of accumulated probabilities, changing this to a table of turning points and collating with cards sorted on a d digit random field when the probabilities are given to d significant figures. The values of j as well as any x or 12 punches are interspersed gang-punched onto the playing field of the random number cards.

Example of Preparation of Table of Turning Points for One Column in the Inversion of a Fourth Order Matrix

j	Column of Probabilities	Accumulated Probabilities	Turning Points	j
1	.323	.323	000	1
2	$-$.076	.399	.323	2*
3	.458	.857	.399	3
4	.109	.966	.857	4
p_j	.034	1.000	.966	x
			Keypunch	

* (also 12 punch)

4. Using either of these procedures, the cards should be sorted on several random fields between the preparation of successive columns of the playing field.

Performing the Calculation

Place the designated wire in position 1 of digit selector 2. Pass entire deck of cards through the tabulator. Press hand key for total. Successively pass cards through with wire in digit selector positions 2 to n.

Machine Wiring

1. *Counters.* Use $n + 2$ counter groups. Read card count into all groups. Add and subtract hubs go to selector F (NX and X, respectively), from C of that selector, through the delayed selectors and the entry selector to "plug to C." Net balance and clear all counters on final total, emitting an asterisk next to negative totals. One counter group is for card count; another counts games played (plug to C through entry selector).

2. *Reading Selectors.* n two-position selectors are used. Each X goes to its corresponding upper brush position in the playing field. C is common and leads to the entry selector. The X in the other position goes to the D–PU of the corresponding delayed selector. The C is again common and leads to a digit of DS 2. (UCI goes to DSC 2, making DS 2 a digit emitter.) The PU of each selector is through the corresponding hub of DS 1.

3. *Delayed Selectors.* These are picked up by a digit impulse passing into the C of all reading selectors and through the X positions to the D–PU of the corresponding selector. Thus, delayed selector 3 picks up one cycle after reading selector 3 picks up. (This is done so that the machine can store the number of the "winning" digit.) The delayed selectors carry a common "plug to C" on their C hubs: the X hubs lead to the corresponding C positions in selector F.

4. *Entry Selector.* Three-positions picked up by the first card or by an 11 from Digit Selector 1. One position carries a "plug to C" impulse to the counter which records the number of games played and to the C of the delayed selectors. A second is used as a drop-out for the negative impulse counter. The third selects either the output of the reading selector bank-NX or an emitted digit from DS 2 (which is impulsed by UCI)-X to DSC 1.

5. *Negative Impulse Counter.* To distinguish an odd from an even number of 12 impulses, the selector picks up on the first negative impulse, drops out on the second, picks up on the third, etc. An 11 emitted by DS 2 passes through the entry selector (C to NX), through a special drop-out selector (C to NX), through one position of Selector F (C to X) to the X–PU of Selector F. The 12 position in DS 1 is wired through Selector F (C to N) to the D–PU of that selector. The corresponding X position carries the 12 impulse to the special drop-out selector. By this means, Selector F remains picked up after odd 12's, and drops out on all even 12's and as each game starts.

6. *Sequencing Circuit.* By passing an x through one distributor (C to X) to the X–PU of the next distributor, a chain of $r + 1$ distributors may be linked together. If the $(r + 1)$st distributor picks up the entry selector and

starts the chain over again, every $(r + 1)$st draw will be scored. This circuit is used for raising matrices to powers without employing control cards.

ASCHER OPLER

Dow Chemical Co.
Pittsburg, Calif.

[1] This is actually sampling without replacement and therefore does not strictly conform to the MARKOV process originally described. However, if the ratio of the number of punched cards to the order of the matrix is very large, nonreplacement has negligible effect on the remaining transition probabilities until the exhaustion of the deck approaches.

[2] H. HOTELLING, "Some new methods in matrix calculation," *Ann. Math. Stat.* v. 14, 1943, p. 1–34.

[3] The IBM 101 statistical machine should be able to invert matrices up to the 30th order because of its selector capacity.

[4] Random numbers on punched cards are available from the RAND Corp., Santa Monica, Calif.

Computing Logical Truth with the California Digital Computer

1. Introduction. A problem which occasionally arises and in quite surprisingly diverse fields of endeavor is the computation of a truth table for a sentence built up of simple sentences connected by the simple sentential connectives. The problem is to find the truth value of the complete sentence for all combinations of truth values for the component sentences. If the complete sentence is a combination of n two-valued sentences, then it itself is two-valued and 2^n possible conditions must be considered. Thus if the number of different component sentences is at all large, an extremely tedious computation is necessary.

Since the computation itself is performed in a routine manner, the possibility of using a high-speed, automatic computer to carry through the details suggests itself. Indeed, a machine for this purpose (the Kalin-Burkhart Logical-Truth Calculator) has already been built.[1] It seems desirable, however, to investigate the possibility of solving these problems using general-purpose, digital computers as many of these will be in operation within the next few years. With this in mind, a program for truth-table calculation is worked out in the following for the California Digital Computer [MTAC, v. 5, p. 57–61]. The general plan should be applicable to any digital computer.

Most problem-solving programs require three basic routines. First the solution of the problem with a given set of data, second storing the result at the proper location in the memory, and third altering the data as required for the next computation. These will be discussed in subsequent sections.

Since the details of a computation program depend to a considerable extent on the characteristics of the computer for which it is developed, a brief description of the operations needed and available in the California Digital Computer (commonly called Caldic) will be presented. Other operations than those described are also available, but the discussion will not be complicated by describing them.

The program and data are given to the machine in the form of holes punched in a standard teletype tape. Results are printed out of the machine on a similar tape. Numerical data and results are in the form of decimal numbers which are used by the machine in a binary-coded form. The fact