# A CLASS OF MACHINES WHICH DETERMINE THE STATISTICAL STRUCTURE OF A SEQUENCE OF CHARACTERS

J. D. Foulkes
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

## SUMMARY

The machines described here try to ascertain the statistical structure of a sequence of characters drawn from a finite and known alphabet. They assume that any sequence is the output of a stochastic source of a restricted type, and they attempt to build and improve a model of the source on that basis.

## 1. Introduction

The machines described in this paper were originally formulated in an attempt to model human behavior in experimental situations of the type described by Bush and Mostellar[1] and Estes[2]. In these experiments, a human observer is presented with a sequence of 1's and 0's and is asked at each trial to estimate the probability of occurrence of a 1 on the next trial. It should be emphasized that no claim is being made about how accurately they reflect human behavior. In what follows it will be helpful to think of the machines as replacing a human observer in an out-guessing situation of this sort.

## 2. A Simple Four State Machine

In attempting to guess the next character, X, of a sequence, one could assume it depended solely on the previous N characters, where N is a fixed number. A machine which employs this strategy for N = 2 is shown in Fig. 1(b). It has four "states", labeled 0 through 3 in the diagram, and these are interconnected by directed branches which represent transitions between states. Each state is associated with a particular sequence of two trials, and the 1-0 branches connecting the states are consistent with this hypothesis. Thus every time the machine is in state 01, the two previous trials have been 01. If the next trial results in a 1 the machine proceeds to state 11, if it is an 0, it goes to state 10. Associated with each state are two counters which keep track of the number of times 0 and 1 have occurred when the machine was in that state. These counters are represented diagrammatically in Fig. 1(c). In effect these counters keep a running account of an estimate of the conditional probability of a 1 given the di-gram associated with each state.

In a typical out-guessing experiment the machine would operate as follows. At the start all counters are set to one, and an arbitrary starting state is selected, state 01 say. The counter readings of this state represent the machines estimate of the probability of a 1 on the next trial, 0.5 in this case. The machine is then informed of the outcome of the first trial. If it is a 1, the one's counter of state 01 adds one to its count and the machine goes to state 11, if it is an 0 the zero counter steps on one and the machine proceeds to state 10. At each trial the machine is some state whose counters indicate the machines estimate of a 1 on the next trial. The outcome of that trial decides which counter of the state is processed and what the next state is. There are various methods of converting the machines probability estimates into an output suitable for a particular experimental situation. This will not concern us here, the estimates themselves will be considered the output.

It is interesting to see what the machines performance is in a few specific cases. If the machine is presented with the repetitive sequence 101010 etc., the counters of states 01 and 10 will soon set in such a way that the machine's predictions will be correct. In a similar way the machine will catch on to all repetitive sequences of length three and one of length four. If the machine is presented with a statistical schedule whose structure depends on n-grams of length two or less, the probability estimates given by the machines counters will settle down to steady state values.

Larger machines of this type for $N > 2$ have a state associated with each block $S_N$ of the previous N characters, Fig. 1(a), and are interconnected by lines in a consistent way, see Ref. 3. D. W. Hagelbargers' machine SEER[4] is a machine of this type for N = 3.

## 3. Variable N-gram Machines

An interesting variant of the machines described in the previous section are machines for which N varies. Fig. 2 for example shows a basic diagram machine which has been extended in such a way that for strings of 1's and 0's the machine can count up to four. For human observers, there is a small amount of evidence to suggest that the number of previous inputs which affect a judgment concerning the next input varies and depends on

In: Institute of Radio Engineers (IRE) Western Electronics (Wescon) Convention Record, 1959, Part 4, Automatic Control, Electronic Computers and Information Theory

the detailed nature of the previous inputs. To illustrate the implications of this fact, notice that the machine in Fig. 2 will "catch on" to quite lengthy repetitive patterns of the type j 1's followed by k 0's, but it fails to catch simple patterns which involve more than one change from 1 to 0 in a sequence e.g. 10110.

## 4. Janet, a State Splitting Machine

One severe limitation of the above machines is that they are incapable of improving their performance. E.G. if the machine of Fig. 1(b) is presented with the repetitive sequence 0010 it will prescribe a figure of 8 path through the states 10, 00 and 01, see Fig. 3(a). The counters of states 01 and 10 will correctly predict the next character, but the counter of state 00 will oscillate about a 50-50 prediction. The machines predictions can be represented 0??00??00 etc. and it is incapable of improving this performance. Suppose however, that the machine could split state 00 into two states, 100 and 000 as in Fig. 3(b). The figure of 8 is opened into a loop and the resulting machine will catch the pattern.

A machine capable of splitting states would operate in the following way. Each state is associated with an N-gram of characters and it incorporates three counters, see Fig. 4. One counter keeps track of the conditional probability of the next character X given the n-gram of the state $S_N$, i.e. this counter carries an estimate of $P(X/S_N)$. The other two counters perform a similar function for two embryo, (n+1)-gram states which the state embodies, i.e. these counters keep running estimates of $P(X/0S_N)$ and $P(X/1S_N)$. A statistical likelihood test can be used to decide whether these estimates of conditional probabilities agree or not. If they disagree the state splits. The parent state is destroyed and the embryos become parent states embodying embryos. One further feature needs to be built into the machine. It should be able to amalgamate states. If the counters of two states (whose n-grams are identical except in the last place), indicate significantly similar conditional probabilities, they should be amalgamated, i.e. they become embryos' in a single parent state. In this way the machine can correct mistakes due to chance splittings.

A machine of this type is cellular in nature and capable of growing in amoeba-like fashion. It starts with a single state which counts the letter frequencies of the sequence and keeps a watch dog account of the diagram probabilities. If the diagrams indicate statistically different behavior, a split occurs and the machine grows. At any stage of growth the machine is counting those n-grams which its experience has shown to be significant, and it maintains a watch dog interest in the next level of (n+1)-grams. For convenience a machine of this type will be called a Janet machine.[5] Janet of course is not limited to an alphabet of two characters. In principle a Janet machine to handle any sized alphabet can be designed.

## 5. Limitations

It is natural to ask what a machine of this type is good for. In general if it is exposed to a source which emits characters from a finite alphabet, known to the machine, it will split states and expand indefinitely. Only for a very limited class of sources will the machine stabilize its form and size.

The concept of structure for a sequence of characters is difficult to define in a sufficiently general way to cover all cases. Consider the sequencies:

01000110000101 (1)

10101110110010001 (2)

01101001100101l0 (3)

The first is a sample of the output of the stochastic source shown in Fig. 5(a). In the second every fourth symbol is a 1 and the other symbols are 1 and 0 with 0.5 probability.

The third is generated by the following rule: write a symbol and then write its compliment; write the compliments of this pair and the compliments of these four etc. It is difficult to cover cases such as these in a general definition of structure and perhaps it is rather pointless from a practical standpoint. Janet was designed with stochastic sources in mind, i.e. the mathematical models of language employed in Information Theory.[6] In general these models consist of graphs with directed branches and associated with the branches are transitional probabilities and output symbols, Fig. 5(a). A process defined in this way is essentially stochastic on the nodes or states of the graph, and the output symbols are almost artifacts. In order for the process to be stochastic on the output symbols the graph must have the property that if we know the present state and the next output symbol, the next state is always known. A process of this type will be called a Shannon type source, see Ref. 7. Figs. 5(a) and (b) show non-Shannon and Shannon type sources respectively. An n-gram Shannon source will be defined as one in which a unique n-gram of output symbols is associated with every node of the graph. Every time the process enters a particular node, the n symbols associated with that node have just been emitted. Fig. 5(c) shows a simple n-gram source. One further restriction needs to be imposed on the source if the machines described here are to converge. Let $P(X/S_N)$ stand for the conditional probability of the next output symbol X given a block of length N of the previous outputs, where N is not necessarily equal to n. Similarly let $P(X/YS_N)$ stand for the conditional probability of X given the block $S_N$ preceded by the symbol Y. Consider the condition:

If, for some $S_N$ and all X and Y
$P(X/YS_N) = P(X/S_N)$

Then $P(X/S_kS_N) = P(X/S_N)$ for all $S_k$ (1)

Not all Shannon type n-gram sources satisfy condition (1), e.g. consider a source which repetitively emits the pattern 00011011. (This is the sequence of numbers 0123 expressed in binary form.) The letter frequencies, $P(X/S_0)$, are one half and the probabilities $P(X/S_1)$ are also a half. The conditional probabilities $P(X/S_2)$ are not all the same, however, 010 for example never occurs. When presented with this source Janet would never split its initial letter frequency state and yet the pattern clearly comes from an n-gram Shannon source.

For simplicity let us call an n-gram source satisfying condition (1) and H-source. Provided reasonable care is exercised in the choice of the statistical criterion used to split and amalgamate states, Janet will converge on a true estimate of the source if it is a finite H-source. If the source is not an H-type, the machine of course will assume that it is and it will attempt to build a model on this basis. This will probably cause it to expand indefinitely, but at each stage the machine will have, within the limitations of its H-type assumption, and approximation of the source. The utility of this estimate will vary but for the practical cases of language and music it is plausible that an H-type approximation would be very good.

## 6. Experiments

In an effort to study some of the implications of these remarks a simulation routine for the IBM 704 was written by Mrs. C. Lochbaum. The routine employs an $X^2$ test to decide when to split or amalgamate states. Two of the experiments performed will be described. In both of them Janet was presented with 1's and 0's emanating from the source shown in Fig. 6. The ringed numbers in the diagram indicate the transitional probabilities in sixteenths. In the first experiment a confidence level of 10% was set for splitting a state, and 5% for amalgamating two states. Fig. 7 shows the successive developments in Janets growth. It can be seen from this table that Janet grew with slow deliberation into the form of the source and solidly stayed there. The same type of behavior was observed for confidence levels of 20%-10%, 20%-15%, and 20%-18%. In the second experiment a level of 30% was set for splitting and 20% for amalgamating. Fig. 8 shows the results. It can be seen that Janet overshoots the correct form of the source twice before settling down. A future paper will describe other experiments.

## 7. Conclusions

Modern computers probably have sufficient capacity to simulate a Janet machine which handles a large vocabulary and lengthy n-grams. A rough estimate for example shows that the 704 could probably simulate a Janet capable of analyzing the Bach Chorals out to n-grams of length 10. The advantage of using Janet rather than a straight n-gram count is that for a computer with fixed capacity Janet will probably obtain a better n-gram approximation of the source.

## Acknowledgements

I would like to acknowledge the very real help I received in this work from Messrs. J. L. Kelly and V. A. Vyssotsky. Their clear-headed Kabitzing corrected much of my own muddled thinking. I would also like to thank Mrs. C. Lochbaum for programming Janet and herself.

## References

1. R. R. Bush and F. Mostellor, Stochastic Model Learning, John Wiley and Sons 1955.

2. W. K. Estes, Of Models and Men, American Psychologist, October 1957.

3. N. G. DeBraijn, A Combinatorial Problem, Proc. Koninklijke Nederlandsche Akademie Van Wetenschappen, 1946.

4. D. W. Hagelbarger, SEER, A Sequence Extrapolating Robot, IRE Trans, Electronic Computers, March 1956.

5. This happens to be the name of the Author's daughter.

6. C. E. Shannon A Mathematic Theory of Communication BSTJ July 1948.

7. V. A. Vyssotsky pointed out to the Author that most of the formulae in Shannon's early work are applicable only to sources of this type.

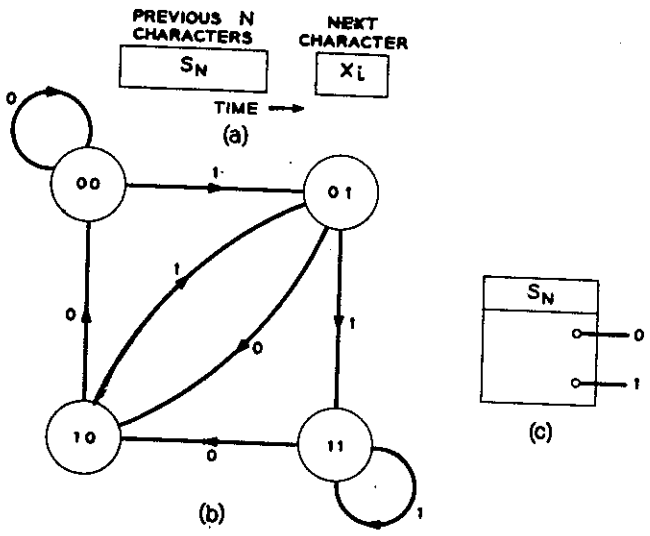
PREVIOUS N CHARACTERS
$S_N$
NEXT CHARACTER
$X_i$
TIME
(a)
00
01
10
11
(b)
$S_N$
0
1
(c)

FIG. 1

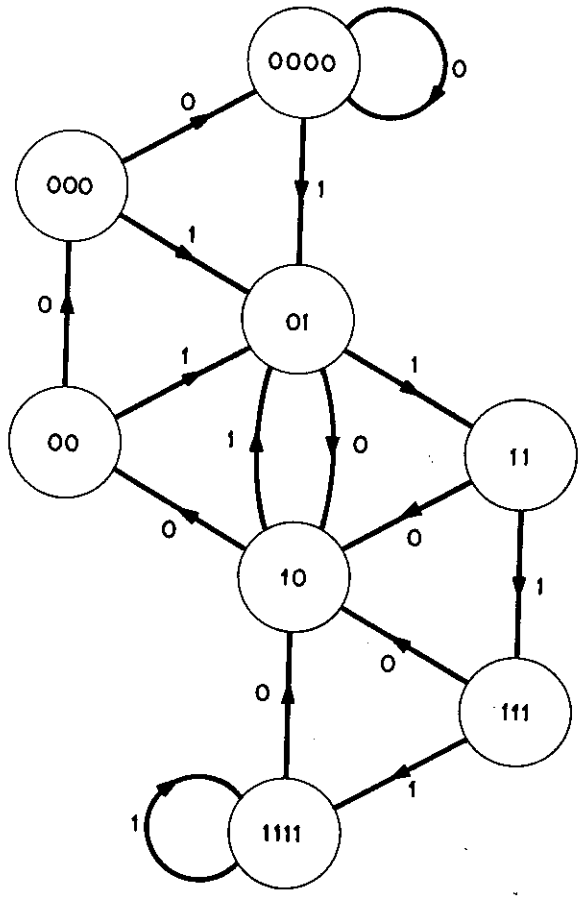
0000
000
01
00
11
10
111
1111

FIG. 2

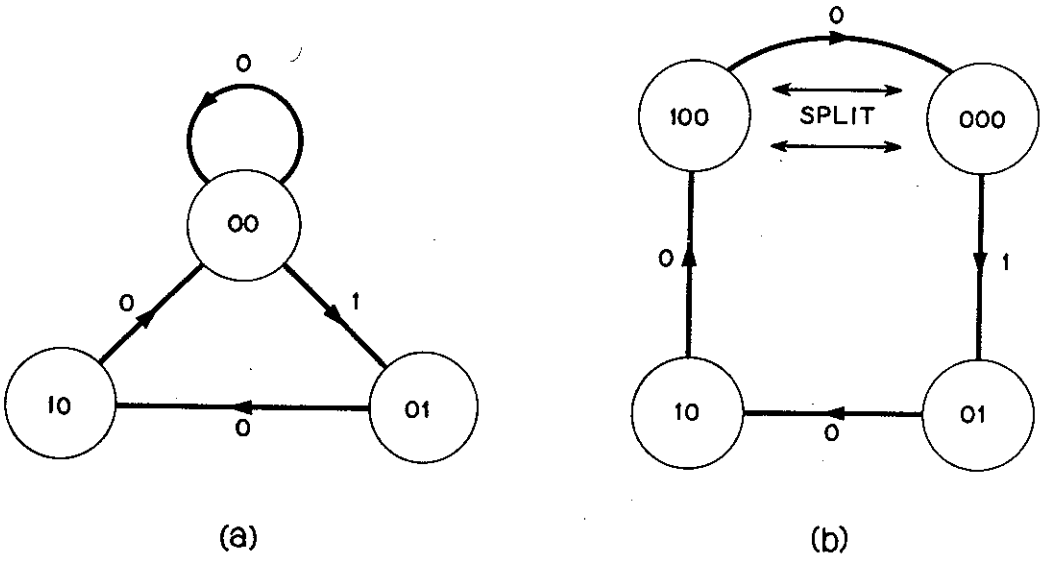
0
00
0
1
10
01
0
(a)
0
100
SPLIT
000
0
1
10
01
0
(b)

FIG. 3

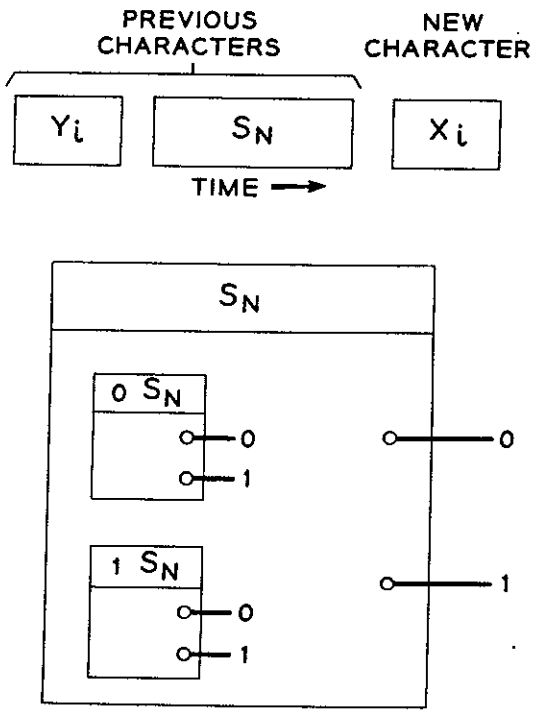
PREVIOUS CHARACTERS
NEW CHARACTER
Yi
SN
Xi
TIME
SN
0 SN
0
1
1 SN
0
1
0
1

FIG. 4

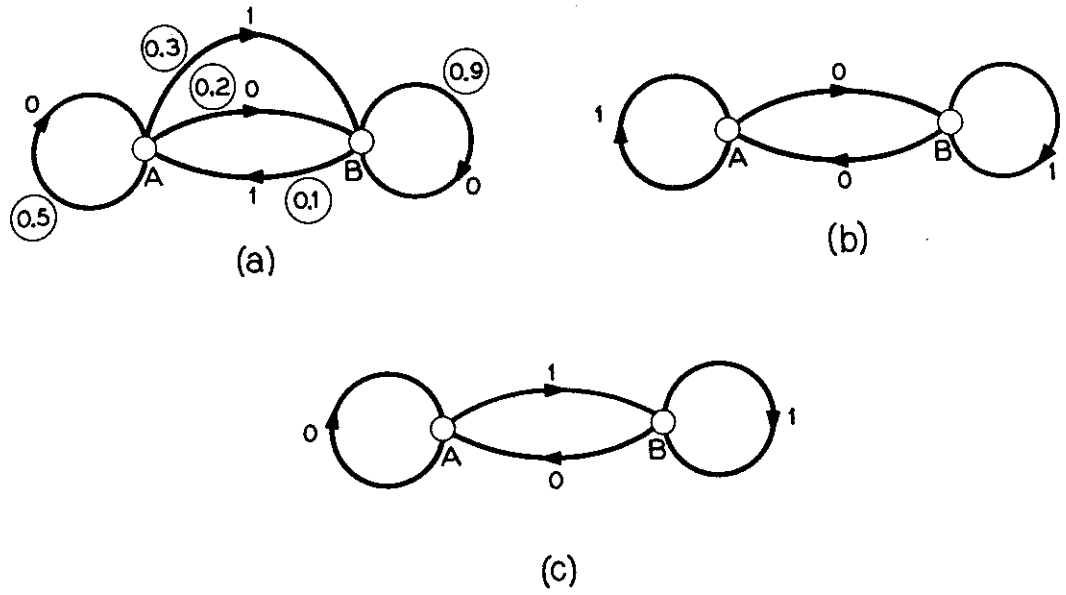
0.3
1
0.2
0
0.9
0
A
B
1
0.1
0
0.5
(a)
0
1
A
B
0
1
(b)
1
0
A
B
0
1
(c)

FIG. 5

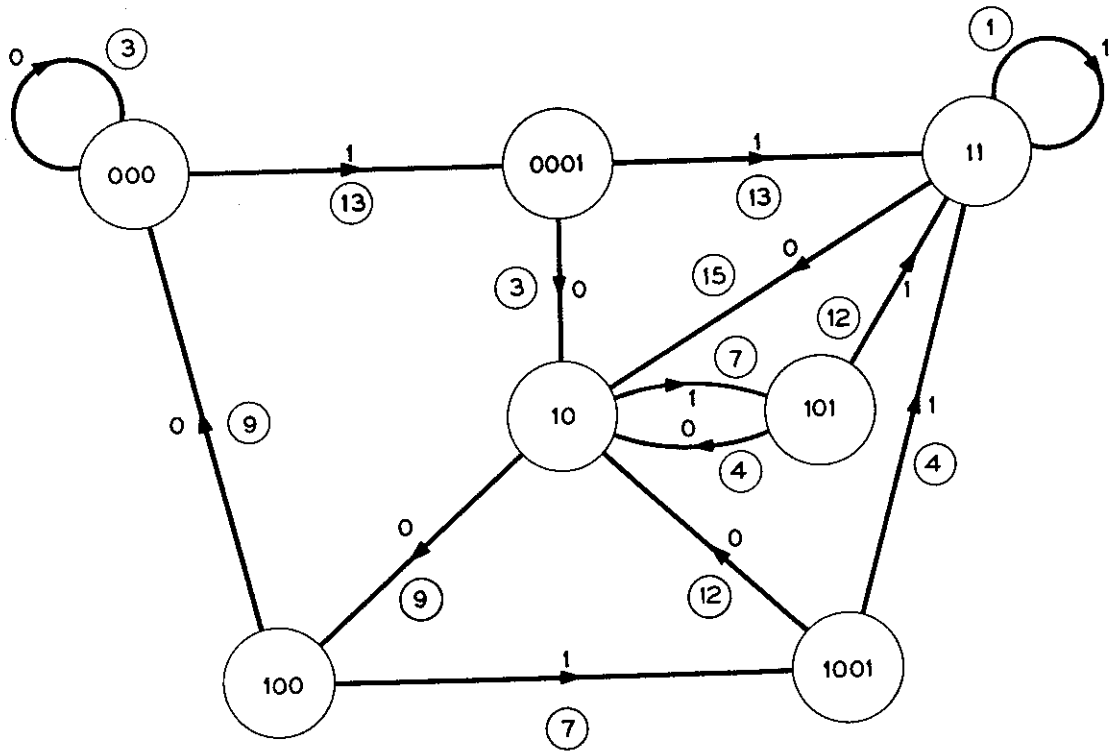
3
0
000
1
13
0001
1
13
1
1
11
3
0
15
0
12
1
7
10
1
0
101
0
9
4
1
4
0
9
0
12
100
1
1001
7

FIG. 6

| NUMBER OF TRIALS | STATES PRESENT |
|---|---|
| Up to 300 | Letter Frequency State |
| 300 to 400 | 0, 1 |
| 400 to 800 | 0, 01, 11 |
| 800 to 1000 | 00, 10, 01, 11 |
| 1000 to 1500 | 000, 100, 10, 01, 11 |
| 1500 to 1700 | 000, 100, 10, 001, 101, 11 |
| 1700 to 3000 | The form of the source. |

Fig. 7

| NUMBER OF TRIALS | STATES PRESENT |
|---|---|
| Up to 100 | Letter Frequency State |
| 100 to 200 | 0, 1 |
| 200 to 400 | 0, 01, 11 |
| 400 to 500 | 00, 01, 10, 11 |
| 500 to 600 | 000, 100, 01, 10, 11 |
| 600 to 900 | 000, 100, 001, 101, 10, 11 |
| 900 to 2500 | 000, 100, 0001, 1001, 101, 10, 11 * |
| 2500 to 2600 | 0000, 1000, 100, 0001, 1001, 101, 10, 11 |
| 2600 to 3000 | 000, 100, 0001, 1001, 101, 10, 11 * |
| 3000 to 4800 | 000, 100, 0001, 1001, 101, 100, 101, 11 |
| 4800 to 6000 | 000, 100, 0001, 1001, 101, 10, 11 * |

Fig. 8

*Indicates correct form of source.