An evolutionary approach to the concept of randomness

F. W. M. Stentiford* and D. W. Lewin†

The Von Mises and Kolmogorov definitions of randomness are discussed in terms of the complexity of binary sequences. An evolutionary approach is then described and some results presented.

(Received March 1972)

Problems of feature extraction in pattern recognition can sometimes be related to the problem of defining a measure of structure or lack of structure in a body of data (Jermann, 1970). Although it is intuitively clear what is meant by ‘lack of structure’ or randomness, it is difficult to set down a precise definition. In this paper two already existing approaches to this problem (1, 2) are described and finally the complexity of a binary sequence is discussed using the concept of an evolutionary procedure.

1. Von Mises’s Definition (Von Mises, 1957)
‘An infinite binary sequence possesses the property of randomness if the relative frequency of 1’s (to 0’s) tends to a certain limiting value which remains unchanged by the omission of a certain number of the elements and the construction of a new sequence from those which are left. The formula for omission must leave an infinite number of retained elements and it must not use the attributes of the selected elements.’

This definition of randomness is very close to what is intuitively meant by the word; if it is at all possible to detect structure in a binary sequence then it should also be possible to construct a selection procedure which changes the relative frequencies of the zeros and ones. In other words, if a sequence can be seen to be non-random, then it is certainly non-random according to the definition.

Although the general intent of Von Mises’s definition conforms with what is meant by randomness, the lack of a precise formulation has led to severe criticism (Church, 1940; Wald, 1937; Martin-Löf, 1966; Lovelock, 1966).

2. Kolmogorov definition
Kolmogorov (1965) and Chaitin (1966, 1970) have independently suggested that computing machines be applied to the problem of defining what is meant by a random or patternless finite sequence.

The length n of a binary string a = a₁a₂…aₙ will be denoted by l(a).

Let A be an algorithm transforming a pair of binary strings p, x into a binary string a = A(p, x).

The conditional complexity of a for given x with respect to A is defined as

\[ K_A(a|x) = \begin{cases} \min \{ l(p) \mid A(p, x) = a \}, \\ +\infty \text{ if there does not exist } p \text{ such that } A(p, x) = a \end{cases} \]

With \( p \) can be thought of as a program which when fed into a machine A causes it to compute \( a \) by means of the given data \( x \) (Fig. 1).

![Fig. 1. A computing machine](image)

Inserting the empty string for \( x \) in \( K_A(a|x) \) gives \( K_A(a) \), the complexity of \( a \) with respect to \( A \).

The random or patternless sequences are those having the greatest complexity, or alternatively, those which necessarily require the longest programs when produced by a computing machine. Those sequences that can be obtained by putting into a computer a small program are those that possess a pattern and follow a law. It can be shown that ‘most’ finite binary sequences of length \( n \) require minimal programs of about length \( n \) to generate them. These sequences are considered to be the random sequences (Chaitin, 1970; Solomonooff, 1964).

The Kolmogorov definition provides a conceptually satisfactory solution to the problem. Patterned finite sequences are just those sequences which follow a simple law; unstructured finite sequences follow a complex law, which could possibly incorporate the sequence itself in a ‘table-look up’ scheme.

Kolmogorov has pointed out a disadvantage in his concept of randomness; it does not allow for the ‘difficulty’ of preparing a program which generates the sequence \( a \). Indeed, the theory gives no indication of how the minimal program \( p \) is obtained.

Effective definitions
The definitions discussed above provide concepts which conform with what is intuitively meant by the word ‘randomness’. However, both definitions are not effectively computable; in that they appeal to an external human informer; in the first instance Von Mises requires ‘formulae for omission’, and in the second instance, Kolmogorov requires ‘minimal length programs’.

Effective definitions of an arbitrarily long random sequence are not possible because it has been shown that there will always exist a computable (and therefore non-random) sequence which is labelled as random by the definition (Levin, Minsky and Silver, 1962).

An evolutionary estimate of relative complexity
Structure is detected in a sequence when it becomes possible to predict terms in the sequence according to some rule. It might seem safe to say that, in general, a sequence is more complex than another if it is more difficult to think up a prediction rule. Although this has the right spirit, it is far too vague to be useful in a rigorous definition.

†Department of Electrical Engineering and Electronics, Brunel University, Kingston Lane, Uxbridge, Middlesex.
‡For a discussion on effective computability see Minsky (1967), Rogers (1967).
Nevertheless, suppose an evolutionary type of process was conceived which had as its goal the correct prediction of binary sequences. It would be possible to get an idea of the complexity of the current binary sequence by monitoring the error rate. It would not be acceptable to use these ideas in an absolute definition of randomness because an evolutionary process itself requires a source of random changes. However, it is possible to give an estimate of the complexity of one sequence relative to another.

These concepts can now be expressed more formally.

**Definition**

Suppose \( s_1, s_2, \ldots, s_N \) is a finite binary sequence then a **predictive evolutionary procedure** \( \Phi \) is a sequence of functions \( \phi_1, \phi_2, \ldots, \phi_M \) \((M < N)\) which generates a finite binary prediction sequence \( s'_1, s'_2, \ldots, s'_r \) where

\[
s'_r = \phi_1(s_1, s_2, \ldots, s_{r-1}) \quad r > 1, i > 1
\]

and \( s'_i = \phi_i \) with \( \phi_1 = 1 \), say.

If \( s'_r \neq s_r \) then \( \phi_{r+1} \) is produced from \( \phi_1 \) by a random change with the constraint that

\[
s_r = \phi_{r+1}(s_1, s_2, \ldots, s_{r-1})
\]

**Definition**

The finite binary sequence \( a_1, a_2, \ldots, a_N \) is more complex (in terms of this definition) with respect to \( \Phi \) than the finite binary sequence \( b_1, b_2, \ldots, b_N \) if the following condition can be satisfied:

Let \( a'_1, a'_2, \ldots, a'_N \) and \( b'_1, b'_2, \ldots, b'_N \) be prediction sequences generated by a predictive evolutionary procedure \( \Phi = \{\phi_1, \phi_2, \ldots, \phi_M\} \) where

\[
a'_r = \phi_1(a_1, a_2, \ldots, a_{r-1})
\]

\[
b'_r = \phi_1(b_1, b_2, \ldots, b_{r-1})
\]

Then there exists a positive integer \( R < N \) such that

\[
g(r, \Phi, a_1, a_2, \ldots, a_N) < g(r, \Phi, b_1, b_2, \ldots, b_N) \quad \text{for all} \quad R < r \leq N
\]

where \( g(r, \Phi, s_1, s_2, \ldots, s_k) \), the predictability score, is the number of correct predictions minus the number of incorrect predictions in the first \( r \) terms of the prediction sequence \( s'_1, s'_2, \ldots, s'_r \).

It must be emphasised that the value of \( N \) is crucial to this definition. \( N \) is the length of the binary sequence under investigation and must be sufficiently large to see a stable trend in \( g(r, \Phi, s_1, s_2, \ldots, s_k) \) as \( r \to N \). \( N \) should certainly be much greater than the length of any periodicity that is known to be present in the binary sequences. In general it is felt that values of \( N \) should be determined empirically, some experimental results are given in the next section which give some indication of the relationship between \( N \) and \( g(N, \Phi, s_1, s_2, \ldots, s_N) \).

The definition enables finite binary sequences to be ordered in terms of their predictability scores. This ordering is by no means an absolute indication of the randomness of a sequence because other attempts at the same ordering would not necessarily give identical results.

We observe that \( \Phi \) is not an effective procedure in the accepted sense (Minsky, 1967; Rogers, 1967). This is because the process employs random changes and the computation is not therefore carried forward deterministically; there is no guarantee that an evolutionary experiment would give the same results if it was carried out at two different times. This means that it would not be possible to design a computable sequence which would be guaranteed to have an estimate of complexity based on an evolutionary procedure.

The definition of ‘more complex’ is not mathematically satisfactory to a formalist because it relies heavily on the way \( \Phi \) is specified. However, it does provide a practical technique for the investigation of redundancy in binary sequences. In the next section a particular \( \Phi \) is defined and it is shown how the predictability scores relate to certain simple sequences.

**Experimental results**

The particular \( \Phi \) chosen for the experiment was a 32-state Moore machine which was continuously evolved to predict binary input sequences. Thirty-two states were chosen only for programming convenience. Sixteen of the states were associated with the output of a 1 and the remaining sixteen with 0. Initially the interconnections between states were random and an arbitrary start state was chosen. This meant that an arbitrary binary output was given as the first prediction. If the next binary input agreed with this output, no change was made to the machine. On the other hand, if the input disagreed with the prediction, then the last state was changed (randomly) to one which would have given a correct output had the process been repeated.

![Fig. 2. Part of finite-state machine model](image-url)

For example, consider the input \( 110 \ldots \) to the Moore machine a part of which is illustrated by a state diagram in **Fig. 2.** In this diagram each machine state is represented by a circle and the output associated with that state is given within the circle. The input associated with a transition from one state to another is given alongside a directed line (edge) joining those two states. The first output 1 is a correct prediction and control is passed to state \( B \) which gives a 0 output. The second input is a 1 and this means that the output by state \( B \) was in fact wrong. The edge connecting states \( A \) and \( B \) is then randomly altered so that the edge now connects \( A \) to some state \( C \) which would have given the correct output of a 1. The current input 1 is then applied to \( C \) and the next prediction 0 is output by state \( D \).

In the experiment this process was continued and small blocks of the input sequence which occurred frequently were found to correspond to one or more connected groups of states in the evolved machine. If the sequence pattern was very common

---

*This measure of prediction ability was used by Levin, Minsky and Silver (1962). Of course, if evolution is simulated on a computer and a pseudo-random number generator is used, then results can be repeated.*
Table 1

<table>
<thead>
<tr>
<th>SEQUENCE</th>
<th>PREDICTABILITY SCORES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td>101010101</td>
<td>66</td>
</tr>
<tr>
<td>00000000000001111111111</td>
<td>56</td>
</tr>
<tr>
<td>101010101</td>
<td>56</td>
</tr>
<tr>
<td>101010101</td>
<td>56</td>
</tr>
<tr>
<td>101010101</td>
<td>68</td>
</tr>
<tr>
<td>101010101</td>
<td>48</td>
</tr>
<tr>
<td>101010101</td>
<td>50</td>
</tr>
<tr>
<td>101010101</td>
<td>56</td>
</tr>
<tr>
<td>101010101</td>
<td>42</td>
</tr>
<tr>
<td>101010101</td>
<td>52</td>
</tr>
<tr>
<td>101010101</td>
<td>44</td>
</tr>
<tr>
<td>101010101</td>
<td>50</td>
</tr>
<tr>
<td>101010101</td>
<td>42</td>
</tr>
<tr>
<td>101010101</td>
<td>48</td>
</tr>
<tr>
<td>101010101</td>
<td>-2</td>
</tr>
<tr>
<td>101010101</td>
<td>34</td>
</tr>
<tr>
<td>101010101</td>
<td>10</td>
</tr>
<tr>
<td>101010101</td>
<td>20</td>
</tr>
<tr>
<td>101010101</td>
<td>12</td>
</tr>
<tr>
<td>101010101</td>
<td>46</td>
</tr>
<tr>
<td>101010101</td>
<td>16</td>
</tr>
<tr>
<td>101010101</td>
<td>18</td>
</tr>
<tr>
<td>101010101</td>
<td>4</td>
</tr>
<tr>
<td>101010101</td>
<td>4</td>
</tr>
<tr>
<td>101010101</td>
<td>18</td>
</tr>
<tr>
<td>101010101</td>
<td>12</td>
</tr>
<tr>
<td>101010101</td>
<td>20</td>
</tr>
<tr>
<td>101010101</td>
<td>14</td>
</tr>
<tr>
<td>111010101</td>
<td>2</td>
</tr>
<tr>
<td>111010101</td>
<td>-4</td>
</tr>
<tr>
<td>111010101</td>
<td>2</td>
</tr>
</tbody>
</table>

It was seen that in most cases the percentage variation in scores decreased as the length of input sequence was increased. Table 1 is arranged with the most predictable sequences at the top. It is felt that this ordering is to a certain extent consistent with subjective estimates of the relative complexities of the sequences.

One sequence gave a significantly negative predictability. This might be considered surprising in view of the fact that a purely random sequence would be expected to have a zero predictability score. However, this sequence is a form of pseudo-noise sequence (Golomb, 1967) which has minimal correlation with shifted versions of itself. This means that this type of sequence is the ‘theoretical worst’ in terms of predictions which are based on short patterns discovered earlier in the process. In fact the pseudo-noise sequence displayed a distinctive structure by its inability to be predicted by this evolutionary procedure. It is also significant to point out that pseudo-noise binary sequences satisfy the following three intuitively acceptable criteria for randomness:

(a) A balance of 0 and 1 terms
(b) Two runs of length n for each run of length n + 1
(c) A two-level auto-correlation function.

Simple sequences composed of cycles which were short \( \left( \frac{N}{2} \right) \) compared with the size of the machine, sometimes gave quite variable scores. This was because on occasion the machine was able to lock precisely into the correct cycle and thereby achieve a considerably higher score than the average. The sequences in Table 1 were chosen to be of length ~30 although their internal structure was often quite simple.

Conclusions

This paper has highlighted some of the difficulties to be encountered when accepted definitions of randomness are applied to real binary sequence. It was observed that these definitions were not constructible in the practical sense and therefore conveyed little or no information about particular sequences.

A practical estimate of the relative complexity of binary sequences was then defined. This definition is based on the concept of an evolutionary procedure which is capable of computer implementation. Several sequences were processed using this scheme and their predictability scores obtained. These scores provided a plausible estimate of the relative complexity of the sequences. Further work is necessary on the effect of the machine size on the predictability scores. The very simple \( \Phi \) described in this paper possesses only a few states and is therefore limited in its ability to distinguish structured sequences.

The non-deterministic search technique set out in this paper has already been applied to related areas in the design process. This includes the reduction of finite-state machines (Stentiford and Lewin, 1971) and the design of features for Optical Character Recognition (Stentiford, 1972). It is felt that evolutionary searches will provide useful tools in many areas of information processing where conventional methods have been unsuccessful.

Acknowledgements

One of the authors, F. W. M. Stentiford, wishes to acknowledge the financial support of the Pessey Company Limited and the Science Research Council for an industrial studentship.

References


---

**ACM George E. Forsythe student paper competition**

UNDERGRADUATES AND HIGH SCHOOL STUDENTS: Announcing the 1973 ACM GEORGE E. FORSYTHE STUDENT PAPER COMPETITION and AWARD. An opportunity to submit your original ideas on any topic related to computers and their applications. Best papers will be published in COMMUNICATIONS OF THE ACM and the authors will receive awards.

Anyone who has not received a bachelor’s degree before April 1, 1973 is eligible. Letters of intent should be submitted by June 11, 1973 and manuscripts by September 1, 1973.

For details see the March 1973 COMMUNICATIONS OF THE ACM or write to:

ACM Student Editorial Committee  
Department of Computer and Communication Sciences  
2076 Frieze Building  
The University of Michigan  
Ann Arbor, Michigan 48104  
USA