A FORMAL THEORY OF INDUCTIVE INFERENCE, Part 1*†

Ray J. Solomonoff

Visiting Professor, Computer Learning Research Center
Royal Holloway, University of London
Mailing Address: P.O.B. 400404, Cambridge, Ma. 02140, U.S.A.

Information and Control, Volume 7, No. 1, Pp. 1-22, March 1964

1 SUMMARY

In Part I, four ostensibly different theoretical models of induction are presented, in which the problem dealt with is the extrapolation of a very long sequence of symbols—presumably containing all of the information to be used in the induction. Almost all, if not all problems in induction can be put in this form.

Some strong heuristic arguments have been obtained for the equivalence of the last three models. One of these models is equivalent to a Bayes formulation, in which a priori probabilities are assigned to sequences of symbols on the basis of the lengths of inputs to a universal Turing machine that are required to produce the sequence of interest as output.

Though it seems likely, it is not certain whether the first of the four models is equivalent to the other three.

Few rigorous results are presented. Informal investigations are made of the properties of these models. There are discussions of their consistency and meaningfulness, of their degree of independence of the exact nature of the Turing machine used, and of the accuracy of their predictions in comparison to those of other induction methods.

*This research was supported by Air Force Office of Scientific Research Contract No. AF 49(638)-376, Grant No. AF-AFOSR 62-377, and Public Health Service NIH Grant No. GM 11021-01.

†A paper given at the Conference on "Cerebral Systems and Computers" held at the California Institute of Technology, February 8-11, 1960 was the subject of Zator Technical Bulletin No. 138. Much of the material of Sections 3.1 to 3.4 first appeared in Zator Technical Bulletins 138 and 139 of November 1960 and January 1961, respectively. Sections 4.1 and 4.2 are more exact presentations of Zator Technical Bulletins 140 and 141 of April 1961 and April 1962, respectively.

In Part II these models are applied to the solution of three problems—prediction of the Bernoulli sequence, extrapolation of a certain kind of Markov chain, and the use of phrase structure grammars for induction.

Though some approximations are used, the first of these problems is treated must rigorously. The result is Laplace's rule of succession.

The solution to the second problem uses less certain approximations, but the properties of the solution that are discussed, are fairly independent of these approximations.

The third application, using phrase structure grammars, is least exact of the three. First a formal solution is presented. Though it appears to have certain deficiencies, it is hoped that presentation of this admittedly inadequate model will suggest acceptable improvements in it. This formal solution is then applied in an approximate way to the determination of the "optimum" phrase structure grammar for a given set of strings. The results that are obtained are plausible, but subject to the uncertainties of the approximation used.

2 INTRODUCTION AND GENERAL DISCUSSION: THE NATURE OF THE PROBLEM

The problem dealt with will be the extrapolation of a long sequence of symbols—these symbols being drawn from some finite alphabet. More specifically, given a long sequence, represented by T, what is the probability that it will be followed by the subsequence represented by a? In the language of Carnap (1950), we want $c(a, T)$, the degree of confirmation of the hypothesis that a will follow, given the evidence that T has just occurred. This corresponds to Carnap's probability$_1$.

The author feels that all problems in inductive inference, whether they involve continuous or discrete data, or both, can be expressed in the form of the extrapolation of a long sequence of symbols. Although many examples have been investigated to lend credence to this hypothesis, this point is not essential to the present paper, which is limited to the problem of extrapolation of sequences of discrete symbols. In all cases being considered, the known sequence of symbols is very long, and contains all of the information that is to be used in the extrapolation.

Several methods will be presented for obtaining formal solutions to this problem. By a formal solution is meant a mathematical equation that in some sense expresses the probability desired as a function of the sequences involved. It will not, in general, be practical to evaluate the probability directly from this equation. In most cases, there is some question as to whether it is even possible *in theory* to perform the indicated evaluation. In all cases, however, the equations will suggest approximations, and the approximations that have been investigated give predictions that seem both qualitatively and quantitatively

reasonable.

The "solutions" that are proposed involve Bayes' Theorem. A priori probabilities are assigned to strings of symbols by examining the manner in which these strings might be produced by a universal Turing machine. Strings with short and/or numerous "descriptions" (a "description" of a string being an input to the machine that yields that string as output) are assigned high a priori probabilities. Strings with long, and/or few descriptions are assigned small a priori probabilities.

Four ostensibly different models of this general nature are presented in Sections 3.1, 3.2, 3.3 and 3.4 respectively.

It should be noted that in these sections, no theorems or rigorous proofs are presented.

Each of the models is described in some detail. Statements are made about various properties of these models. In few cases have any *rigorous* proofs been constructed for these statements, but in all cases, they represent the author's strong opinions— based for the most part on plausibility arguments and numerous specific examples. The text will sometimes give these arguments and examples.

Occasionally, the phrase, "it can be shown that," will introduce a statement for which either a rigorous proof or a *very* convincing "plausibility argument" has been found. In such cases, the demonstration will not be presented.

These models all lead to somewhat different expressions for the probabilities of various possible extrapolations of a given sequence. Although it is not demonstrated in the present text, it can be made very plausible that the last three methods are equivalent. Whether the first method is equivalent to the other three is not, at the present time, certain, though the author is inclined to think that it is.

These alternate formulations of a general theory make it easier to understand the operation and application of the theory in a variety of types of problems.

That these kinds of models might be valid is suggested by "Occam's razor," one interpretation of which is that the more "simple" or "economical" of several hypotheses is the more likely. Turing machines are then used to explicate the concepts of "simplicity" or "economy"—the most "simple" hypothesis being that with the shortest "description."

Another suggested point of support is the principle of indifference. If all inputs to a Turing machine that are of a given fixed length, are assigned "indifferently equal a priori" likelihoods, then the probability distribution on the output strings is equivalent to that imposed by the third model described, i.e., that of Section 3.3.

Huffman coding gives a third rationale for these models. If we start out with an ensemble of long strings of a certain type, and we know the probability of each of these strings, then Huffman coding will enable us to code these strings, "minimally," so that on the average, the codes for the most probable strings are as short as possible.

More briefly, given the probability distribution on the strings, Huffman tells us how to code them minimally. The presently proposed inductive inference methods can in a sense be regarded as an inversion of Huffman coding, in that we first obtain the minimal code for a string, and from this code, we obtain the probability of that string.

The question of the "validity" of these inductive inference methods is a difficult one. In general, it is impossible to prove that any proposed inductive inference method is "correct." It is possible to show that one is "incorrect" by proving it to be internally inconsistent, or showing that it gives results that are grossly at odds with our intuitive evaluation.

The strongest evidence that we can obtain for the validity of a proposed induction method, is that it yields results that are in accord with intuitive evaluations in many different kinds of situations in which we have strong intuitive ideas,

The internal consistency and meaningfulness of the proposed methods are discussed in the sections following 3.1.1. The author feels that the proposed systems are consistent and meaningful, but at the present time, this feeling is supported only by heuristic reasoning and several nonrigorous demonstrations.

Evidence for the validity of the methods is principally in the form of applications to specific problems. Some of these have been worked out in Part II. Although the results of these applications appear to be strongly in accord with intuitive evaluations of the problems treated, several approximations are used in going from the basic inference methods to the final solutions. For this reason, the "correctness" of these apparent solutions makes a rather imperfect corroboration of the basic inference methods.

The extrapolation problems dealt with in the present paper are a Bernoulli sequence, and two types of sequences generated by progressively more complex formal grammars.

Success has been obtained in applying the basic methods to problems in continuous prediction such as fitting curves to empirical data, and it is expected that the results of this work will appear in forthcoming papers.

2.1 AN EVALUATION OF THE VALIDITY OF THE METHODS PROPOSED

There are four types of evidence presented for the validity of the proposed models.

First, there is the general intuitive basis, involving such things as Occam's razor, the principle of indifference, and the inversion of Huffman codes. This type of evidence is of much less importance than the *second* kind—the application of the methods to specific problems and comparison of the results with intuitive evaluations. The *third* type is given in Section 3.4, in which it is made plausible that for a certain "goodness" criterion, and a very large body of data,

the model is at least as good as any other that may be proposed. Any proposed general inductive inference system must at *least* satisfy this condition.

The *fourth* type of evidence is the discussion of the consistency and meaningfulness of the methods in the sections following 3.1.1.

Of most importance is the second type of evidence, which is presented in Part II. In evaluating a method of induction, we apply it to problems in which we have strong intuitive ideas about what the solutions are, or about certain properties of the solutions. The degree of correlation between our intuitive beliefs and the results obtained through the theory, will largely determine our degree of confidence in the theory. If this correlation is high in many problems of diverse nature in which we have strong intuitive feelings, we will begin to trust the theory in cases in which our intuitive feelings are weaker.

The first application of the present approach is made in Section 4.1 of Part II. Prediction of the next element of a Bernoulli sequence is shown to obtain Laplace's rule of succession. Though this particular result is by no means universally accepted, it is not an unreasonable one.

The second application, in Section 4.2, treats a type of sequence in which there are certain kinds of intersymbol constraints, and the results seem to be very reasonable.

The third application, in Section 4.3, deals with extrapolation through the use of context-free phrase structure grammars. Little work has been done in this field, and we have few intuitive ideas about what the results should be. However, some of the properties of the results are investigated and are found to be intuitively reasonable.

Another application, which is not dealt with in the present paper, is the fitting of curves to empirical data. The results obtained thus far agree with results obtained by classical methods.

3 SEVERAL INDUCTIVE INFERENCE SYSTEMS

Sections 3.1, 3.2, 3.3 and 3.4 will describe in some detail four ostensibly different systems for extrapolating a long sequence of symbols.

It has been made plausible that the last three methods are essentially the same, but it is not certain as to whether the first one is the same as the others.

Section 3 will define more exactly the concepts "description" and "universal machine."

Section 3.1 describes an induction system in which an a priori probability is assigned to a sequence on the basis of a weighted sum of all possible descriptions of that sequence with all possible continuations of it. The weight assigned to a description of length N is 2^{-N}. Several criticisms of this model are discussed—among them, the meaningfulness of the formulation and the degree of depen-

dence upon just what Turing machine was used.

Section 3.2 describes a model of induction that attempts to explain in a uniform way, all of the data that we receive from the universe around us. A new type of universal machine, more like an ordinary digital computer is introduced. It is a kind of 3-tape Turing machine, and some of its properties are discussed. A comparison is made between the induction models being presented, and induction that is based on the explicit formulation of scientific laws.

Section 3.3 is a model in which the a priori probability assigned to a string is proportional to the number of descriptions that it has of a given fixed length. This model may be viewed as a more exact formulation of "The principle of insufficient reason."

Section 3.4 is a model whose predictions are a weighted sum of the predictions of all describable probability evaluation methods. The weights are assigned to a method on the basis of both its past success and its a priori probability.

It is then made plausible that this "summation" method under certain conditions, is at least as "good" as any of its component methods for a stated "goodness" criterion.

The methods described will all use universal Turing machines or approximations to such machines to detect regularities in the known part of the sequence. These regularities will then be used for extrapolation.

Critical in the concepts of induction considered here is the "description" of a "corpus" or body of data with respect to a given machine.

Suppose that we have a general purpose digital computer M_1, with a very large memory. Later we shall consider Turing machines—essentially computers having infinitely expandable memories.

Any finite string of 0's and 1's is an acceptable input to M_1 . The output of M_1 (when it has an output) will be a (usually different) string of symbols, usually in an alphabet other than the binary. If the input string S to machine M_1, gives output string T, we shall write

$$M_1(S) = T$$

Under these conditions. we shall say that "S is a description of T with respect to machine M_1 ." $M_1(S)$ will be considered to be meaningless if M_1, never stops when it is given the input S.

Next, the concept of "universal machine" will be defined. A "universal machine" is a subclass of universal Turing machines that can simulate any other computing machine in a certain way.

More exactly, suppose M_2 is an arbitrary computing machine, and $M_2,(x)$ is the output of M_2 , for input string x. Then if M_1, is a "universal machine," there exists some string, a (which is a function of M_1 and M_2 , but not of x), such that for any string, x,

$$M_1(a \frown x) = M_2(x)$$

a may be viewed as the "translation instructions" from M_1 to M_2 . Here the notation $a \frown x$ indicates the concatenation of string a and string x.

It is possible to devise a complete theory of inductive inference using Bayes' Theorem, if we are able to assign an a priori probability to every conceivable sequence of symbols. In accord with this approach. it is felt that sequences should be given high a priori probabilities if they have short descriptions and/or many different descriptions. The methods of Sections 3.1, 3.2, 3.3 and 3.4 may be regarded as more exact formulations of these two ideas; they give, in effect, a relative weighting for these two aspects of the descriptions of a sequence.

In general, any regularity in a corpus may be utilized to write a shorter description of that corpus. Remaining regularities in the descriptions can, in turn, be used to write even shorter descriptions, etc.

The simplest example of a description that will be given is in Section 4.1. Here, a Bernoulli sequence is described.

It is seen that any "regularities" (i.e., deviations of the relative frequencies of various symbols from the average), result in shorter and/or more numerous descriptions.

On a direct intuitive level, the high a priori probability assigned to a sequence with a short description corresponds to one possible interpretation of "Occam's Razor." The assignment of high a priori probabilities to sequences with many descriptions corresponds to a feeling that if an occurrence has many possible causes, then it is more likely.

3.1 INDUCTIVE INFERENCE SYSTEM USING ALL POSSIBLE DESCRIPTIONS AND ALL POSSIBLE CONTINUATIONS OF THE CORPUS

The first method to be discussed will assign a probability to any possible continuation of a known finite string of symbols.

Suppose T is a very long sequence of symbols in some known alphabet, A.

A, the "output alphabet," contains just r different symbols.

T is the corpus that we will extrapolate, *and must contain all of the information that we want to use in the induction.*

M_1 is a universal machine with output alphabet A, and a binary input alphabet.

a is a finite sequence of symbols of alphabet A, and is therefore a possible immediate continuation of sequence T. We want to find the probability of this possible continuation.

$P(a, T, M_1)$ might be called "the probability with respect to M_1, that a will follow T."

What we wish to call P is not of very much importance. P will be *defined* by Eq.(1), and we will then investigate the properties of the quantity defined. It will later appear that this quantity has most (if not all) of the qualities desired in

an explication of Carnap's probability$_1$ (Carnap, 1950) and so we have chosen to refer to P as a "probability." The reader may find it distasteful to refer to P as "probability" until it has been proven to have all of the necessary properties—in which case he might for the time being mentally read "probability$_x$" whenever P is referred to.

$C_{n,k}$ is a sequence of n symbols in the *output* alphabet of the universal machine M_1 . There are r different symbols, so there are r^n different sequences of this type. $C_{n,k}$ is the kth such sequence. k may have any value from 1 to r^n.

$TaC_{n,k}$ is the same as $T \frown a \frown C_{n,k}$.

$(S_{TaC_{n,k}})i$ is the ith description of $TaC_{n,k}$ with respect to Machine M_1. The descriptions can be made in order of length, but the exact ordering method is not critical. $N_{\left(S_{TaC_{n,k}}\right)_i}$ is the number of bits in $(S_{TaC_{n,k}})_i$.

$$P(a,T,M_1) \equiv \lim_{\epsilon \to 0} \lim_{n \to \infty} \frac{\sum_{k=1}^{r^n} \sum_{i=1}^{\infty} [(1-\epsilon)/2]^{N_{(S_{TaC_{n,k}})_i}}}{\sum_{k=1}^{r^{n+1}} \sum_{i=1}^{\infty} [(1-\epsilon)/2]^{N_{(S_{TC_{n+1,k}})_i}}} \tag{1}$$

To get some understanding of this rather complex definition we will first consider only the numerator of the right side of Eq. (1). The denominator is a normalization factor; without it, the equation gives something like the relative probability of the continuation a, as compared with other possible continuations. Next, set ϵ to zero, and let n also equal zero. This gives the very approximate expression

$$\sum_{i=1}^{\infty} \left(\frac{1}{2}\right)^{N_{(S_{Ta})_i}} \tag{2}$$

It becomes clear at this point that if the sequence Ta has a short description (i.e., for some i, $N_{(S_{Ta})_i}$ is small), then expression (2) will hold a lot of weight for that description. Furthermore, if there are many such short descriptions of Ta, then expression (2) will be given much additional weight.

Unfortunately, it can be shown that the number of descriptions of length m of Ta (or any other sequence) is at least proportional to 2^m for large enough m. This causes expression 2 to diverge, and so the $1-\epsilon$ factor of Eq. (1) is inserted, giving[1]

$$\lim_{\epsilon \to 0} \sum_{i=1}^{\infty} \left(\frac{1-\epsilon}{2}\right)^{N_{(S_{Ta})_i}} \tag{3}$$

[1]The $\epsilon \to 0$ limit should not be applied to the numerator alone, but to the ratio of numerator and denominator in Eq. (1) [error noted in June 2000].

For nonzero ϵ, the effect is to give negligible weight to descriptions whose lengths are many times greater than $1/\epsilon$.

In the method of Section 3.2, this convergence is obtained in a somewhat different manner. Another way in which Eq. (1) differs from expression (3) is that Eq. (1) considers that the partial sequence Ta might have been the beginning of any number of longer sequences that start with Ta. By including all possible continuations of Ta—i.e., $C_{n,k}$, we give the sequence Ta a larger probability if it is capable of being the beginning of a *longer* sequence that is of high probability. An example is the coding of the sequence $D \equiv abcdabcdabcdabcdab$ which can be dealt with using the methods of Section 4.2.

If described in a direct way, the sequence D has a rather lengthy description. if, however, we first define the subsequence *abcd* to be represented by the intermediate symbol α, we can write D as $B\alpha\alpha\alpha ab$, B being a subsequence that defines α to be *abcd*. it is reasonably likely that the sequence D has the continuation *cdabcd* ... Though the sequence $B\alpha\alpha\alpha ab$ is much shorter than the original D, the description of the sequence Dcd is $B\alpha\alpha\alpha\alpha$, which is even shorter. The sequences like $B\alpha\alpha\alpha ab$ are to be considered as "intermediate codes" for D. The method by which they are represented as a single positive integer—or a sequence of 0's and 1's —is dealt with in Sections 4.1.1 and 4.2.2. It will become clear that the intermediate sequence $B\alpha\alpha\alpha\alpha$, has a far shorter code than $B\alpha\alpha\alpha ab$, since in the latter case, the symbols a and b have not been used much in the intermediate code, and therefore are effectively represented by relatively long expressions in the final code.

It must be stressed that while the above reasoning gives some of the reasons for the choice of Eq. (1) as a reasonable definition of probability, these arguments are meant to be heuristic only. The final decision as to whether Eq. (1) is a good definition or not rests to a rather small extent upon the heuristic reasoning that gave rise to it, and almost entirely upon the results of investigations of the properties of this definition. Investigations of this type are presented in Part II.

The author feels that Eq. (1) is likely to be correct or almost correct, but that the methods of working the problems of Sections 4.1 to 4.3 are *more* likely to be correct than Eq.(1). If Eq. (1) is found to be meaningless, inconsistent, or somehow gives results that are intuitively unreasonable, then Eq. (1) should be modified in ways that do not destroy the validity of the methods used in Sections 4.1 to 4.3.

3.1.1 Criticisms and Questions About Eq. (1)

There are several questions that immediately come to mind when equation 1 is proposed as an explication of probability$_1$.

3.1.1.1. The terms of the summation do not include descriptions that are "meaningless," i.e., inputs to the universal machine M_1, for which the machine does not stop. Turing (1937) has shown that it is impossible to devise a Turing

machine that will always be able to tell, in a finite time, whether an arbitrary string will be "meaningful" for another particular universal Turing machine. This raises the question of whether the right side of Eq. (1) defines anything at all.

3.1.1.2. It has not been shown rigorously that the limit of the right side of Eq. (1) exists, as ϵ, n, and i approach their respective limits.

3.1.1.3. It would seem that the value of $P(a, T, M_1)$ would be critically dependent upon just what universal machine, M_1, was used. Is there any particular M_1 that we can use that is better than any other?

3.1.2 Replies to Criticisms of Eq. (1)

3.1.2.1. It is clear that many of the individual terms of Eq. (1) are not "effectively computable" in the sense of Turing (1937). It is very reasonable to conjecture that the entire right side of Eq. (1) is not "effectively computable." This does *not* mean, however, that we cannot use Eq (1) as the heuristic basis of various approximations. If an approximation to Eq. (1) is found that yields intuitively reasonable probability values and *is* effectively computable, we would probably adopt such an approximation as a better explication of probability$_1$ than Eq. (1).

One approach to such an approximation can be made by first considering as "meaningful" input to the universal machine, only those that complete their output in less than τ operations. With such a limitation, each of the terms of Eq. (1) is "effectively computable." The summations on i and k both become finite summations. If we let τ approach infinity and then let ϵ approach zero we will have some sort of approximation to Eq. (1).

At the present rudimentary state of development of the theory, different approximations to Eq. (1) are used for different types of problems. It is hoped that further work in this field may yield a unified, useful approximation to Eq. (1).

3.1.2.2. Though the existence of the limits designated on the right side of Eq. (1) has not yet been proved, various approximations to this equation have been made which tacitly assume the existence of these limits. Some of these approximations are described in Sections 4.1 to 4.3.

3.1.2.3. it is likely that Eq. (1) is fairly independent of just what universal machine is used, if T is sufficiently long, and contains enough redundance, and M_1 is "fairly good" at expressing the regularities of T. Although a proof is not available, an outline of the heuristic reasoning behind this statement will give clues as to the meanings of the terms used and the degree of validity to be expected of the statement itself. Suppose we have a very long sequence T, containing m symbols, and we have two universal machines, M_1 and M_2. We will try to show that

$$P(a, T, M_1) \approx P(a, T, M_2) \tag{4}$$

Let $N_{(S_T,1)}$ and $N_{(S_T,2)}$ represent the lengths of the shortest codes for T, with respect to M_1 and M_2 respectively.

Let α_1 be M_2's simulation instructions for M_1.

Let α_2 be M_1's simulation instructions for M_2

Then for all strings, x,

$$M_1(x) = M_2(\alpha_1 \frown x)$$

and

$$M_2(x) = M_1(\alpha_2 \frown x)$$

Let N_{α_1} and N_{α_2} be the number of bits in α_1 and α_2 respectively. Then

$$N_{(S_T,1)} \leq N_{(S_T,2)} + N_{\alpha_2} \tag{5}$$

and

$$N_{(S_T,2)} \leq N_{(S_T,1)} + N_{\alpha_1} \tag{6}$$

To explain Eq. (5), note that M_1 can always code anything by using M_2's code, prefixed by α_2. Thus M_1's shortest code cannot be more than N_{α_2} longer than M_2's shortest code for the same sequence. A similar argument holds for Eq. (6).

Suppose that M_1 is basically more efficient in coding T. Then if m, the number of symbols in T, is sufficiently large, it is plausible to hypothesize that M_2's shortest code will indeed be obtained by simulating M_1 and using M_1's shortest code. It is necessary to assume that m is very large, because any slight advantage between M_1's and M_2's coding methods is accentuated in coding a long sequence. One might suppose that for values of m that are not too large, the difference between the code lengths are roughly proportional to m. When m becomes large enough so that the difference is about N_{α_1} , then the difference remains constant, because M_2, is forming its minimal code by simulating M_1

If M_2's shortest codes are all N_{α_1} bits longer than M_1's shortest codes, it is clear that the largest terms of $P(a, T, M_1)$ in Eq. (1) (i.e., the terms due to the shortest codes) will all be $2^{N_{\alpha_1}}$ times as large as the corresponding terms in a corresponding expression for $P(a, T, M_2)$. Since these terms occur in both numerator and denominator, this factor will approximately cancel out with the result that $P(a, T, M_1)$ and $P(a, T, M_2)$ are approximately equal.

Though the weak points in the above heuristic arguments are many, the reasoning is strong to the extent that

1. M_1 is appreciably more efficient than M_2, in coding regularities of the type that occur in T.

2. M_2 is "close" to M_1 in the sense that N_{α_1} is "small".

3. m is sufficiently large so that M_1's shortest code for T is much longer than N_{α_1}.

In using $P(a, T, M_1)$ it would seem best to select an M_1 that is fairly efficient in coding the sequences in which we will be interested. The LISP list processing language in which recursive definitions are easily implemented, or any of the other computer languages that have been devised for convenience in dealing with material in certain large areas of science (Green, 1961) might be used as the basis of simulation of M_1.

3.1.3 A Method of Applying Eq. (1)

In many situations in which induction is to be applied, the sequence T is not constant, but grows in time. The problem is to make many inferences at different times, each based upon the entire sequence up to that time. Instead of having to recode the entire sequence each time to obtain the desired inference, it is possible to summarize the previous work by modifying the machine M_1 suitably and concern one's self only with the coding of the new data. More exactly, suppose that T is our original sequence, and we have made some inferences based on T alone. Later, we are given the subsequence D, which is part of the continuation of T, and are asked to make inferences based on $T \frown D$. Then there always exists a universal machine, M_2, such that

$$P(a, T \frown D, M_1) = P(a, D, M_2)$$

for any conceivable subsequence a. In general, the nature of M_2 will depend upon both T and M_1 M_2 can be viewed as a summary of the inductive data of T, with respect to M_1

Though it has been possible to prove that at least one M_2 exists satisfying the requirements described, the M_2 thereby obtained makes the problem of finding short codes for $D \frown a$ just as difficult as the problem of finding short codes for $T \frown D \frown a$. It is clear that the M_2 obtained in this way does not summarize, in any *useful* way, the information contained in the sequence T. It is felt, however, that if suitable approximations to Eq. (1) are used, it is indeed possible to have M_2 summarize in a useful manner the information contained in T. A trivial example occurs if the only regularity in T is contained in the frequencies of its various symbols. If M_2 contains a listing of the number of occurrences of each of the symbols of T, it will then contain a summary of T in a form that is adequate for most additions of new data on the continuation of T.

It is possible, however, that data that *seems* to summarize the regularities of T, does not do so, in view of new data. For example, if we had "summarized" T by the frequencies of its various symbols, we would not be able to notice the exact repetition of the entire sequence T, if it occurred later.

3.2 SYSTEM IN THE FORM OF A MODEL TO ACCOUNT FOR ALL REGULARITIES IN THE OBSERVED UNIVERSE

Suppose that all of the sensory observations of a human being since his birth were coded in some sort of uniform digital notation and written down as a long sequence of symbols. Then, a model that accounts in an *optimum* manner for the creation of this string, including the interaction of the man with his environment, can be formed by supposing that the string was created as the output of a universal machine of random input.

Here "random input" means that the input sequence is a Markov chain with the probability of each symbol being a function of any previous symbols in the finite past. The input alphabet may be any finite alphabet.

In the simplest case, the input will be a binary sequence with equal probabilities for zero and one. This situation appears, at first glance, to be identical to Eq. (1), with this equation being used for Bayes' inference, given equal a priori probabilities to all possible input sequences of the same length.

There is, however, an important difference in that the present case deals with infinitely long inputs, while Eq. (1) deals with *finite* inputs only. The meaningfulness of "a legal output" of the machine with infinitely long inputs must then be defined. In Eq. (1), any finite input leading to a nonterminating output is effectively given a priori probability zero. For an infinitely long input, however, the output is often nonterminating.

In the present case, we shall regard an output as "meaningful" if every symbol of the output takes only a finite number of operations to compute it. It is easiest to give this a more rigorous meaning in machines that have separate tapes for infinite input, infinite output and infinitely expandable memory. In such a 3-tape machine we can stipulate that an output symbol, once written, can never be erased, and so we need ask that for each output symbol not more than a finite time elapse before that symbol is written.

There appears to be a difference between the present method and that of Section 3.1, in that the present method only considers part of the future of the sequence to be extrapolated—it does *not* consider its extension into the infinite future, as does Section 3.1.

To compare the method of the present section with that of Eq. (1), first consider the following definitions, which are to be used in the *present section only.*

M_2 is a 3-tape machine with unidirectional output and input tapes.

T is a possible output sequence containing just m symbols.

S is a possible input sequence.

Then we shall say that "S is a code of T (with respect to M_2)," if the subsequence composed of the first m symbols of $M_2(S)$ is identical to T.

We shall say that "S is a minimal code of T" if (a) S is a code of T and (b) if the last symbol of S is removed then the resultant sentence is no longer a

code of T.

Since every minimal code of T is directly representable as a positive integer, these minimal codes can be linearly ordered.

Let $N(T,i)$ be the number of bits in the ith minimal code of T.

Then using the model previously described in the present section, and a simple application of Bayes' theorem, it is found that the probability that the sequence T will be followed by the subsequence a, is

$$P'(a,T,M_2) \equiv \frac{\sum_{i=1}^{\infty} 2^{-N(Ta,i)}}{\sum_{i=1}^{\infty} 2^{-N(T,i)}} \tag{7}$$

The apparent simplicity of Eq. (7) over Eq. (1) is due to two factors: first, Eq. (7) has a very simple automatic device for considering possible continuations of Ta. This device is built into the definition of the "minimal codes of Ta." Second, there are no problems of convergence, since the sums of both numerator or denominator are bounded by unity. This makes the $1-\epsilon$ factor of Eq. (1) unnecessary.

It should be noted that M_2 of Eq. (7) is somewhat different from M_1 of Eq. (1). If $T = M_2(S)$ then if we adjoin more symbols to the right of S, as $S \frown \alpha$, then $S \frown \alpha$ will still be "a code of T," and $M_2(S \frown \alpha)$ will consist of the string $T \frown b$, where b is a finite, infinite or null string.

This condition is *not* true of M_1, which is an unconstrained universal machine. If $M_1(S) = T$, then little, if anything, can be said about $M_1(S \frown \alpha)$. $M_1(S \frown \alpha)$ could be longer or shorter than T; it may even be the null sequence. These properties of M_1 make it difficult to define "a minimal code of T (with respect to M_1)," in the sense that it was defined for M_2.

At the beginning of the present section, it was mentioned that the present model would account for the sequence of interest, in "an optimum manner."

By "optimum manner" it is meant that the model we are discussing is at *least* as good as any other model of the universe in accounting for the sequence in question. Other models may devise mechanistic explanations of the sequence in terms of the known laws of science, or they may devise empirical mechanisms that optimally approximate the behavior and observations of the man within certain limits. Most of the models that we use to explain the universe around us are based upon laws and informal stochastic relations that are the result of induction using much data that we or others have observed. The induction methods used in the present paper are meant to bypass the explicit formulation of scientific laws, and use the data of the past directly to make inductive inferences about specific future events.

It should be noted, then, that if the present model of the universe is to compete with other models of the universe that use scientific laws, then the

sequence used in the present model must contain enough data of the sort that gave rise to the induction of these scientific laws.

The laws of science that have been discovered can be viewed as summaries of large amounts of empirical data about the universe. In the present context, each such law can be transformed into a method of compactly coding the empirical data that gave rise to that law. Instead of including the raw data in our induction sequence, it is possible, using a suitable formalism, to write the laws based on this data into the sequence and obtain similar results in induction. Using the raw data will, however, give predictions that are at least as good, and usually better, than using the summaries of the data. This is because these summaries of the data are almost always imperfect and lose much information through this imperfection.

It may, at this point, seem gratuitous to claim that the proposed model is optimum with respect to all other conceivable models, many of which have not yet been discovered. It would seem to be impossible to compare the present model with the undiscovered models of the future, and thus claim optimality. We will, however, give in Section 3.4 a model of induction, apparently equivalent to the present one, in which all possible induction models are formally considered. The predictions of each possible induction model are used in a weighted sum to obtain predictions that are at least as "good" (in a certain stated sense) as any of the component induction models.

3.2.1 The concept of stochastic languages (Solomonoff, 1959) suggests another way of looking at the induction model of Section 3.2. A stochastic language is an assignment of probability values to all finite strings of some finite alphabet. Though a specific type of stochastic language is dealt with in Section 4.3, we can characterize the most general possible stochastic language through the use of a 3-tape universal machine, with binary input, and an output in the alphabet of the desired language.

Let D be an arbitrary finite binary sequence, let M_1 be such a 3-tape universal machine, and let R_i be a random infinite binary sequence, with equal probability for zero or one.

$M_1(D \frown R_i)$ will define a probability distribution on all possible output strings. This distribution will then constitute a stochastic language. The string D, can be considered to be a description of this language with respect to M_1

The independence of the induction methods of the present paper upon the exact nature of the Turing machine used can be put in a particularly compact form using the concept. If $[\alpha_j]$ is the set of "translation instructions" from M_1 to all other possible universal machines, M_j, then we may say that the stochastic language defined by $M_1(\alpha_j \frown R_i)$ is fairly independent of α_j for very long sentences, and for α_j within a rather large class.

3.3 SYSTEM USING A UNIVERSAL MACHINE WITH ALL POSSIBLE INPUT STRINGS OF A FIXED LENGTH

Consider a very long string T, of m symbols drawn from an alphabet of r different symbols. We shall first obtain a method for assigning apriori probabilities to all strings longer than T. On this basis we can use Bayes' theorem to obtain a probability distribution for various possible continuations of T. As before, it is desirable that T contain much redundance, and that it contain all of the information that we expect to use, either directly or indirectly, in our induction.

Choose some large number, R, such that

$$2^R \gg r^m \tag{8}$$

In this way, binary strings of length R can be expected to contain more "information" than the string T. In the following development, we shall allow R to approach infinity.

Suppose M to be a universal machine with binary input alphabet, and an output alphabet that is the same as that of T. We shall consider M to be either of the ordinary type, M_1, described in Section 3.1, or the 3-tape type, M_2, described in Section 3.2. In the present case, it has been proved that these two machine types give equivalent results.

Consider all binary strings of length R. Say N_R of them are meaningful inputs to M—i.e., they cause M to stop eventually. Of these N_R meaningful inputs to M, say N_T of them result in outputs whose first m symbols are, respectively, identical to the m symbols of T. Then the a priori probability assigned to T will be

$$N_T/N_R \tag{9}$$

This ratio will become more exact as R approaches infinity, but will usually be good enough if R satisfies Eq. (8).

It can be proved that the present inductive inference model is identical to that of Section 3.2, if M is a machine of either type M_1, or of type M_2.

Although it has not been rigorously proved, it seems likely, at the present time, that the methods of the present section give results identical to those of Section 3.1.

An equation that follows from Eq. (9) that is, however, similar to Eq. (1), is

$$P''(a,T,M_1) \equiv \lim_{\epsilon \to 0} \frac{\sum_{n=1}^{\infty} \sum_{k=1}^{r^n} \sum_{i=1}^{\infty} \left[\frac{(1-\epsilon)}{2}\right]^{N(S_{TaC_{n,k}})_i}}{\sum_{n=1}^{\infty} \sum_{k=1}^{r^{n+1}} \sum_{i=1}^{\infty} \left[\frac{(1-\epsilon)}{2}\right]^{N(S_{TC_{n+1,k}})_i}} \tag{10}$$

A corresponding expression for the probability that the subsequence a (rather than any other subsequence) will follow T, that is based on Eq. (9), is

$$P'''(a,T,M_1) \equiv \lim_{R\to\infty} \frac{N_{Ta}}{N_R} \cdot \left(\frac{N_T}{N_R}\right)^{-1} = \lim_{R\to\infty} \frac{N_{Ta}}{N_T} \tag{11}$$

The formulation of the induction system as a universal machine with input strings of fixed length has an interesting interpretation in terms of "the principle of insufficient reason." If we consider the input sequence to be the "cause" of the observed output sequence, and we consider all input sequences of a given length to be equiprobable (since we have no a priori reason to prefer one rather than any other) then we obtain the present model of induction.

3.4 A SYSTEM EMPLOYING ALL POSSIBLE PROBABILITY EVALUATION METHODS

An inductive inference system will be described that makes probability evaluations by using a weighted mean of the evaluations given by all possible probability evaluation methods. The weight given to any particular evaluation method depends upon two factors. The first factor is the success that method would have had in predicting the now known sequence. The second is the a priori probability of that probability evaluation method. it is approximately measured by the minimum number of bits required to describe that method.

3.4.1 A More Detailed Description of the System

Consider the extrapolation of a long string, T, containing m symbols, drawn from an alphabet, A, containing r different symbols, $b_i (i = 1, 2, \cdots, r)$.

A "probability evaluation method" (which we will henceforth designate as "a PEM") is a method of assigning a priori probability values to any sequence of symbols in A. From these probability assignments it is then possible, using Bayes' theorem, to find the probability of any specified continuation of a known sequence.

A normalized PEM (which we will henceforth designate as a "NPEM") is one in which the sum of the probabilities of all possible continuations of a sequence is equal to the probability of that sequence. More exactly, let $P_1(B)$ be the a priori probability assigned to string B by a certain PEM, Q_i.

If, for all strings, a,

$$\sum_{j=1}^{r} P_i(a \frown b_j) = P_i(a)$$

and

$$\sum_{j=1}^{r} P_i(b_j) = 1$$

then Q_i is a NPEM.

To compute with respect to Q_i the probability that string T will have the continuation a, we can use Bayes' theorem to obtain the value

$$\frac{P_i(T \frown a)}{P_i(T)} \tag{12}$$

If Q_i is a NPEM, we shall define D_i, to be "a binary description of Q_i, with respect to machine M_2," if for all strings, a, $M_2(D_i \frown \Delta \frown a)$ is an infinite string giving the binary expansion of $P_i(a)$. The symbols of D_i are to be drawn from a binary alphabet, and Δ is a special symbol that is used to tell M_2 where D_i ends and a begins.

In order that it be meaningful for M_2 to have an infinite output sequence, we will specify that M_2 be a 3-tape universal machine of the type that was discussed in Section 3.2.

Consider all binary strings of length R. For a given large value of R, a certain fraction of these strings will be binary descriptions with respect to M_2, of the NPEM, Q_i. We will assume (and this assumption can be made plausible) that this fraction approaches a limit, f_i, as R approaches infinity. The inductive inference system that shall be proposed is

$$P''''(a, T, M_2) \equiv \frac{\sum_{i=1}^{\infty} f_i P_i(T \frown a)}{\sum_{i=1}^{\infty} f_i P_i(T)} \tag{13}$$

Here, the summations in numerator and denominator range over all possible NPEM's, Q_i.

Though it is not difficult to show that P'''' defines a NPEM, this NPEM is not "effectively computable" in the sense of Turing (1937) and so it does not include itself in the summation of equation 13.

3.4.2 A Comparison of the Present System with Other PEM's

An important characteristic of Eq. (13) is illustrated, if we write it in the form

$$P''''(a, T, M_2) \equiv \sum_{i=1}^{\infty} \left[\frac{P_i(T \frown a)}{P_i(T)} \cdot \frac{f_i P_i(T)}{\sum_{j=1}^{\infty} f_j P_j(T)} \right] \tag{14}$$

Here, $P_i(T \frown a)/P_i(T)$ is the probability that T will have continuation a, in view of PEM, Q_i. This is the same as expression (12). Equation (14) is then a weighted sum of the probabilities for the continuation a, as given by all possible

PEM's. The factor $f_i P_i(T)$ gives the weight of PEM Q_i, and $1/\sum_{j=1}^{\infty} f_j P_j(T)$ is the normalizing factor for all of the weights.

It would seem, then, that if T is a very long string, P'''' will make an evaluation based largely on the PEM of greatest weight. This is because while the f_i's are independent of T, $P_i(T)$ normally decreases exponentially as T increase in length. Also, if Q_i and Q_j are two different PEM's and Q_i is "better" than Q_j, then usually $P_i(T)/P_j(T)$ increases exponentially as T increases in length. Of greater import, however, $f_i P_i(T)/f_j P_j(T)$, which is the relative weight of Q_i and Q_j, increases to arbitrarily large values for long enough T's. This suggests that for very long T, Eq. (14) gives almost all of the weight to the single "best" PEM.

Here we define "best" using one of the criteria defined by McCarthy (1956), i.e., PEM Q_i is "better" than PEM Q_j with respect to string T, if $P_i(T) > P_j(T)$.

This suggests that for very long T's, P'''' gives at least about as good predictions as any other PEM, and is much better than most of them.

There are some arguments that make it plausible that P'''' is a close approximation to P''' of Eq. (11). If this is so, then it becomes likely that the PEM's of Sections 3.2 and 3.3 are also, for "sufficiently long T's," at least as good as any other PEM.

It should be noted that the arguments used to suggest the superiority of P'''' over other PEM's is similar to that used in Section 3.1.2.3 for the plausibility that P (of Eq. (1)) is largely machine independent for long enough T. Both arguments are, of course, extremely informal, and are meant only to suggest how a proof might possibly be found.

ACKNOWLEDGMENTS

Many of the basic ideas on induction that have been presented are the outgrowth of numerous discussions over many years with Marvin Minsky.

Discussions with Roland Silver and James Slagle have been particularly important in the analysis of the properties of Turing machines. Criticism by Eugene Pendergraft has resulted in a much more readable paper, has done much to clarify the section on stochastic phrase structure grammars.

RECEIVED: May 24, 1962

REFERENCES

Carnap, R. (1950), "Logical Foundations of Probability." Univ. of Chicago Press.

Turing, A. M. (1937), On computable numbers, with an application to the Entsheidungsproblem. *Proc. London Math Soc.* 42, 230–265.

Green, B.F., Jr. (1961), Computer languages for symbol manipulation. *IRE Trans. Electron Comp.* EC-lO, December No. 4. pp. 729–735.

McCarthy, J. (1956), A measure of the value of information. *Proc. Natl. Acad. Sci.* 42, 654–655.

Chomsky, A.N. (1956), "Three Models for the Description of Language," *IRE Transactions on Information Theory*, Vol. IT-2, No. 3, Sept. 1956, pp. 113–124.

Solomonoff, R.J. (1959), "A Progress Report on Machines to Learn to Translate Languages and Retrieve Information," *Advances in Documentation and Library Science*, Vol. III, pt. 2, pp. 941–953. Interscience, New York; AFOSR TN-59-646 (Contract AF 49(638)-376); ZTB-134.

Solomonoff, R.J. (1960), "The mechanization of linguistic learning." *Proc. Second Intern. Cong. Cybernetics, Namur, Belgium, September 1958*, pp. 180-193; AFOSR TN-59-246 (Contract AF 49(638)-376); ASTIA AD No. 212 226; ZTB-125.