"Nothing is always absolutely so"
– T. Sturgeon (Venture S.F., July 57)

Lots of Comments; Corrections

V-131

A PRELIMINARY REPORT ON A GENERAL THEORY OF INDUCTIVE INFERENCE

R. J. Solomonoff

February 4, 1960

ZATOR COMPANY
140½ MOUNT AUBURN STREET, CAMBRIDGE 38, MASS.

A prelim. copy of a prelim. report on a tentative analysis of a hypothetical possibility

Preliminary Copy <- silly! i.e.

In next edition:

1) Mention application to U evaln. —
How proof ? with successful use.

2) Mention ~~[illegible]~~ that
"act of faith" is necy. in all ind. inf. systems — and just where this comes in here

V-131

A PRELIMINARY REPORT ON
A GENERAL THEORY OF INDUCTIVE INFERENCE

R. J. Solomonoff

February 4, 1960

CONTRACT AF 49(638)-376

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
AIR RESEARCH AND DEVELOPMENT COMMAND
UNITED STATES AIR FORCE
WASHINGTON 25, D. C.

ZATOR COMPANY
140½ MOUNT AUBURN STREET, CAMBRIDGE 38, MASS.

TABLE OF CONTENTS

Perhaps desc. 4 ind. inf. methods.
and say which I think are equiv.
— which are most likely true.

1. Min
2. Σ
3. Fixed length, counting.
4. wtd. Pars.
5. The "max redundancy operator"

also discuss. "max redundancy op." as opt. extrapolator.

PREFACE

This memo is an outline of some preliminary work on a completely general theory of inductive inference, for universes containing continuous, discontinuous, numerical and non-numerical objects.

The most important previous attempts to obtain a unified theory have been those of R. A. Fisher and of R. Carnap. It is felt that there is a good possibility that the method outlined here, overcomes some of the serious shortcomings of the methods of Fisher and of Carnap.

The final statement of the present method is Equation (5) of Section 11. The rest of the memo deals with successive approximations leading to Equation (5), and some outlines of applications of Equation (5) to specific problems.

Although the gross approximations used to obtain some of the results of the application of Equation (5) lead the author to have incomplete confidence in them, it is felt that Equation (5) itself is fairly likely to be correct.

The specific inductive inference problem dealt with is the extrapolation of an ordered sequence of discrete symbols. The methods may, however, be used to extrapolate unordered sets of objects. In order to deal with continuous data, any consistent method of converting from continuous to digital symbolism may be used, and then the regular method can be used with the digital symbols.

The method described is used only for the extrapolation of sequences of symbols. If predictions about objects in the real world are desired, one must devise some method of making a correspondence between the symbol sequences and

Somewhat confusing

events in the world. It is believed that using the present extrapolation method on the symbol sequences will result in probability values that correspond to those in the real world, and that the probability values obtained for real-world events in this way will be largely independent of the nature of the correspondence that is devised between the symbols and the events they represent — just as long as the correspondence is not "unreasonably complicated", is used consistently, ~~and it~~ and does not lose too much information.

Also note that the method of presentation of ideas is <u>not</u> the method by which these ideas were discovered. The original ideas were derived from a limited definition of probability, employing ~~limited~~ certain types of stochastic languages. The present more general idea of ~~equ~~ probability was obtained by using a ~~more~~ very general type of stochastic language.

V131

1. INTRODUCTION

We shall be concerned primarily with the problem of extrapolation of a very general time series, whose members may be numbers or non-numerical objects, or mixtures of these. At first, a fairly simple extrapolation formula will be given. Its shortcomings will be discussed, and it will be progressively improved upon, until a final formula that seems to overcome all of these difficulties will be presented.

Consider a very long sequence of symbols – e.g., a passage of English text, or a long mathematical derivation. We shall consider such a sequence of symbols to be "simple" and have high a priori probability, if there exists a very brief description of this sequence – using, of course, some sort of stipulated description method. More exactly, if we use only the symbols 0 and 1 to express our description, we will assign the probability 2^{-N} to a sequence of symbols, if its shortest possible binary description contains N digits.

2. THE CONCEPT OF "BINARY DESCRIPTION".

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 large 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." If S is the shortest such description of T, and S contains N digits, then we will assign to the string, T, the a priori probability, 2^{-N}.

3. THE FIRST APPROXIMATE EQUATION

Let us apply this a priori probability to time series extrapolation. Suppose that T is a string of symbols that constitutes a time series. We want to know the relative probability that the next symbol in the series will the symbol "a" rather than the symbol "b".

Let $T^\frown a$ represent the string of symbols that is T concatenated with the symbol a.

Let $T^\frown b$ similarly defined.

Let S_a be the shortest description of $T^\frown a$, with respect to machine M_1.

Let S_b be the correspondingly minimal description for $T^\frown b$.

Let N_{S_a} be the number of digits in S_a.

Let N_{S_b} be the number of digits in S_b.

Then the relative probability of a, rather than b, as continuation of the sequence T, will be, with respect to machine M_1,

1) $$2^{-N_{S_a} + N_{S_b}}$$

which is the ratio of the a priori probabilities of $T^\frown a$ and $T^\frown b$.

4. FIRST OBJECTION: THAT EQUATION 1 IS MACHINE DEPENDENT

There are several very serious objections that immediately come to mind. First, it is quite clear that N_{S_a} and N_{S_b} will depend very much upon just what machine is selected — in fact, by properly selecting machines, we can give $N_{S_b} - N_{S_a}$ any value we like.

We will later (in Section ~~8~~ 9) try to make it plausible that if T is a very long sequence of symbols that contains all of the kinds of data that a man is likely to observe in his lifetime, then $N_{S_b} - N_{S_a}$ will be machine independent over a rather large "natural" set of machines.

5. SECOND OBJECTION: THAT THE PROBABILITIES OF EQUATION 1 DO NOT CONVERGE

Another objection is that if we assign a priori probability 2^{-N} to a binary string of length N, then the total a priori probability of all binary strings does not converge – i.e., There are 2 strings for $N = 1$; their individual probabilities are $\frac{1}{2}$ each, their total probability is 1. There are 4 strings for $N = 2$, their total probability is 1 also. Similarly, the total probability of all strings of length N will be 1, for <u>any</u> value of N. Clearly the sum of all these probabilities does not converge.

We can, however, think of the binary descriptions as being formed by a simple Markov process. The digit 0 is produced with probability $\frac{1}{2}$. The digit 1 is also produced with probability $\frac{1}{2}$. Clearly such a Markov chain has no means to terminate. It must be of infinite length.

We can remedy this difficulty in a very natural way by giving the digits 0 and 1, each probability $\frac{1}{2} - \frac{1}{2}\epsilon$, and have the probability of termination of the string be ϵ. Since we will deal only with very long descriptions, ϵ will be very small. Using the ϵ formalism, we find that though the total a priori probability of all sequences, does indeed converge, our prediction probabilities have not changed much. Instead of

$$2^{-N_{S_a} + N_{S_b}}$$

we now write

$$[\tfrac{1}{2}(1-\epsilon)]^{N_{S_a} - N_{S_b}}.$$

Since ϵ is much less than 1,

$$(1-\epsilon)^{N_{S_a} - N_{S_b}} \approx 1$$

and

$$[\tfrac{1}{2}(1-\epsilon)]^{N_{S_a} - N_{S_b}} \quad \text{is very close to} \quad 2^{-N_{S_a} + N_{S_b}}.$$

It is clear that the expected length of a description is about $1/\epsilon$.

6. THIRD OBJECTION: THAT ALL THE PROBABILITY RATIOS OF EQUATION 1 ARE INTEGRAL POWERS OF TWO

Another objection that comes immediately to mind is that $N_{S_b} - N_{S_a}$ must always be an integer, and so the relative probabilities of the two possible continuations of the sequence would have to be integral powers of 2 – certainly this is not a realistic restriction, since, in general, probabilities may have <u>any</u> values between zero and one.

We will overcome this difficulty by three different devices. The first is somewhat ad hoc, and will be discussed immediately. The second will overcome another difficulty in addition to the present one, and will be discussed in Section 11. – p18 (eqs) These two methods do not interfere with one another. A third method is discussed in Section 13. – p20

It will be noted that the present difficulty seems associated with the use of just two symbol types in our description strings. If we used more than two types of symbols (i.e. say N types) there would be even more trouble, since the probability ratios would then be restricted to integral powers of N – an even coarser gradation than integral powers of 2.

An apparently direct source of trouble is that if an integral number of symbol types are used, there is usually some "wastage of bits" in expressing integers. For example, to express the integer 7 in binary notation, we use 3 bits in the sequence 111. However, to express the integer 8, 4 digits are needed, i. e. :1000. It seems unlikely that 8, which is only 14% larger than 7, should require a whole extra bit. Also the numbers 9 through 15 all require only 4 bits.

It isn't clear as to just how this is applicable. i.e. just how are integers used in defns?

Much "bit wastage" can be avoided if we allow a "cost" of just $\log_2 n$ bits for the number n, if n occurs in a context in which the value zero would be meaningless. If zero is meaningful, a cost of $\log_2(n+1)$ should be assigned to the number n.

In the previous paragraph, and in the following example, it will seem as though the means used for representing numbers in descriptions are rather arbitrary. It can, however, be made plausible that the probability ratios obtained using these rather "arbitrary methods" are identical with the ratios obtained using the more intuitively reasonable Equation (5) of Section 11.

7. A SIMPLE EXAMPLE OF INDUCTION

A very simple example is afforded by a sequence of a A's and b B's. The letters A and B occur in arbitrary order. We are then asked "What is the relative likelihood of an A rather than a B following this sequence?"

To describe the sequence of a A's and b B's, we first note that there are just $(a+b)!/a!\,b!$ different sequences containing just a A's and b B's. A complete description of the sequence would then be given by the string R A B a b k. k tells which of the $(a+b)!/a!\,b!$ different orderings of the symbols A and B actually occurred and

$$1 \le k \le \frac{(a+b)!}{a!\,b!}$$

This formula is obsolete. The Carnap formula (which leads to Laplace's formula), is much better

R tells the computer just what sort of notation is being used. In general, there will be several different symbols of this type.

To compute the bit cost of this description, we would have to know how A, B and R are to be represented in our system. Suppose A costs C_A bits, ~~and~~ B costs C_B bits and R costs C_R bits (C_A, C_B and C_R are all irrelevant to the final probability ratio to be computed).

The numbers a and b cost $\log_2 a$ and $\log_2 b$ bits, respectively.

k will cost $\log_2[(a+b)!/a!\,b!]$ bits.

The cost of k seems a bit arbitrary – should it not be $\log_2 k$?

This looks like "the principal of indifference"

First of all, k differs from a and b, in that k has both upper and lower limits. k is a choice between $(a+b)!/a!\,b!$ alternatives. On the "average," k will have about $\log_2[(a+b)!/a!\,b!]$ bits in its binary representation – but this does not justify using a cost of $\log_2[(a+b)!/a!\,b!]$ bits for k, when k does not have that many digits in its binary representation.

omit underline.

? this arg't. is different from below

Again the true justification to using this bit cost for k is that it results in ? ?
the same probability ratios as the more intuitively reasonable Equation 5 of
Section 10.

The total bit cost obtained for the description R A B a b k is
$C_A + C_B + C_R + \log_2 a + \log_2 b + \log_2 [\,(a+b)!\,/\,a!\,b!\,]$. The resultant
a priori probability is

$$2^{(-C_A - C_B - C_R)} \quad \frac{(a-1)!\,(b-1)!}{(a+b)!} \qquad \cancel{2^{(-C_A - C_B - C_R)}}$$

Let us now consider the same sequence of A's and B's, to which an additional A has been appended. The resultant sequence will have $a+1$ A's and b B's. Its a priori probability is therefore

2)
$$2^{-C_A - C_B - C_R}\, \frac{a!\,(b-1)!}{(a+b+1)!}$$

Appending an A has multiplied the a priori probability of the resultant sequence by a factor of

$$\frac{a}{a+b+1}$$

We may view $-\log_2(a\,/\,(a+b+1))$ as the bit cost of the symbol A, in that particular situation, and we shall call $(a+b+1)\,/\,a$ the "raw cost" of the symbol A in that situation.

Similarly, the a priori probability of the sequence after B has been appended is

3)
$$2^{(-C_A - C_B - C_R)}\, \frac{(a-1)!\,b!}{(a+b+1)!}$$

The bit cost of the appended B is $-\log_2(b\,/\,(a+b+1))$ and the raw cost of B was $(a+b+1)\,/\,b$.

The relative probability of A rather than B following the original sequence of a A's and b B's is the ratio of the a priori probabilities ~~of~~ in expression (2) and expression (3). This is

$$\frac{2^{(-C_A - C_B - C_R)}}{2^{(-C_A - C_B - C_R)}} \; \frac{a!\,(b-1)!\,/\,(a+b+1)!}{(a-1)!\,b!\,/\,(a+b+1)!} = \frac{a}{b}$$

Using the Carnap cost, we get the Laplace $\frac{a+1}{b+1}$

which is approximately what is expected. Note also that

$$\frac{a}{b} = \frac{\text{raw cost of B}}{\text{raw cost of A}}\;,$$

a relationship which ~~is generalizable.~~ continues to be true, when suitably generalized

This simple result, which gives the frequency ratio of 2 kinds of events as an estimate of their probability ratio, is called, in inductive inference circles, "The Straight Rule." An important objection to it is that if $a = 2$ and $b = 0$, then it tells us that we have a probabiltiy of 1 for the next symbol being A. This seems intuitively unreasonable, since we would certainly not be absolutely certain of the next symbol after so short a sequence.

"Laplace's rule" gives the value $(a+1)/(b+1)$.

Carnap (Ref. 1, page 568) gives $(a+k_1)/(b+k_2)$, with the values of k_1 and k_2 dependent upon the exact nature of the properties whose relative frequency one is measuring. If we consider a universe in which very many properties exist, k_1 and k_2 become quite large, and the probability ratio obtained becomes almost independent of empirical data, unless the amount of emirical data is very large.

I should check on this, and see just how Carnap applies his method to get k_1 and $k_2 \neq 1$.

A more detailed analysis reveals that Equation (1) (as modified by the considerations of Section 6) does not give the objectionable ratio a/b, for small values of a ~~and~~ or b. This is true because, under these circumstances, the code R A B a b k is not a minimal code, and it is more economical to write the sequence itself than to use the "R" method.

check on this.

V131

Let us use the symbol V to denote the identity code, so that if we use the sequence V A B B A as input to machine M_1, its output would be A B B A. Symbolically,

$$M_1 \, (V\ A\ B\ B\ A) \; = \; A\ B\ B\ A$$

or, more generally,

$$M_1 (V^\frown X) \; \equiv \; X$$

for any sequence X.

The cost of coding A B using the "V" method is $C_A + C_B + C_V$.

The cost of coding A B using the "R" method is

$$C_A + C_B + C_R + \log_2 \left(\frac{2!}{(1-1)! \; (1-1)!} \right) \; = \; C_A + C_B + C_R + 1$$

The "V" coding method will be more economical than the "R" coding method in this case if

$$C_V \; < \; C_R + 1 .$$

Probably best to introduce r cost and p cost earlier!

In general, the raw cost of a symbol type (e. g., R or V) will be about equal to the reciprocal of its relative frequency of use in the previous part of the code.

As a result, the V notation will be used here if, in the past, the V notation has been used more than $\frac{1}{2}$ as often as the R notation. If short strings of random symbols have occured quite often in the sequence to be described, then the V notation will be used very often, and will have a low bit cost.

If V has a very low bit cost, then if we want to extrapolate the sequence A A B, the cost of A A B B is $C_V + 2\,C_A + 2\,C_B$. The cost of A A B A is $C_V + 3\,C_A + C_B$. The relative probabilities of A and B following will then be

$$2^{(-C_A + C_B)} \; = \; 2^{C_B} \Big/ 2^{C_A}$$

This will be about equal to the ratio of the frequency of occurrence of the symbol

A and the symbol B in the sequence preceding the subsequence A A B. If A and B have never occurred before, we might obtain the ratio 1, or if the symbols A and B have other structural features, we might obtain some other ratio – corresponding to Carnap's k_1 / k_2. Not clear

However, if the present sequence is quite long, e.g., A B B A B A B A A A B A A, then the R notation is likely to cost less than the V notation, and the computed relative probabilities of A and B following will be independent of their frequencies in the part of the sequence preceding the part under present consideration.

In Section 10, an improved inductive inference method will be described, in which <u>all</u> possible methods of describing a sequence contribute to its probability – rather than just the "minimal method" of description. Using this method, the probability ratio $(a + k_1) / (b + k_2)$ appears to be approximately correct. The values of k_1 and k_2 are, however, not the same as those of Carnap. How did I get this?

8. CODING AND RECODING

The method used in coding a sequence is to first write a code description of it, using any convenient symbols.

This description will sometimes contain the R and V symbols of Section 7, a space symbol, and verious letters and numbers. The numbers are recoded by special methods that take advantage of either the fact that the range of possible values of the number is known, or else that the first digit of the number must be 1, the second digit is more probably a zero than a 1, and so on.

The R, V, and space symbols are recoded using the R notation.

If any regularities are found in the resultant code sequence, it is recoded again in a manner that takes advantage of these regularities. The final "minimal" code for a sequence will contain about an equal number of zeros and ones, and will desplay <u>no</u> "significant" statistical regularities <u>at all</u>.

9. REPLY TO THE FIRST OBJECTION

At this point the reader may note that the original premises have apparently been discarded entirely – that while the original idea was to devise a minimal description for a sequence using an arbitrarily chosen machine, we have instead made a description for a special machine that must be very narrowly specialized to interpret that description!

To answer this criticism it will be necessary to modify the premises a bit. Let us designate by S, the sequence consisting of a A's and b B's. Unless the sequence S is very long, the present methods are not very useful for extrapolating S alone. However, let us define S' to be a very long sequence of symbols containing all the kinds of data that a man is likely to observe in his life time. It would be well if this man had a broad background in the kinds of material that we will be extrapolating, but this is not absolutely necessary.

The present methods will be useful for extrapolating the sequence $S'^{\frown}S$. Note that S' need not have any material bearing directly on the sequences to be extrapolated. The relationship of S' to S will be seen presently.

We shall try to make it plausible that the last few symbols in the minimal description of the sequence $S'^{\frown}S$ will be largely independent of just what computer is to be used ~~to interpret the description,~~ as long as that computer is a "universal machine" – which is a kind of general purpose computer – also, that these last few symbols will probably be R A B a b k, or equivalent symbols having the same bit costs.

First the concept of "universal machine" will be defined. A "universal machine" is a sub-class of universal Turing machines that can simulate any other Turing machine in a certain way.

More exactly, suppose M_2 is an arbitrary Turing 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, α (which is a function of M_1 and M_2, but not of x), such that for any string, x,

$$M_1(\alpha \frown x) = M_2(x).$$

α may be viewed as the "translation instructions" from M_2 to M_1.

Let us suppose that M_2 is a machine that is able to perform the decoding from the code string R A B a b k, to the sequence S, so that

$$M_2(\mathrm{R\,A\,B\,a\,b\,k}) = S.$$

Suppose that M_1, a universal machine, has some other method of coding the sequence S, so that

$$M_1(D) = S,$$

and that the sequence D is longer (has more bits) than the sequence R A B a b k. Furthermore, let us suppose that the sequence S' contains many subsequences similar to S, in the sense that the same kind of coding method would apply. Let us assume that the R A B a b k method of coding used by M_2 is, on the average, better than that used by M_1, so that on the average, it costs M_1 3 more bits than M_2 to code a sequence like S. If M_2's coding method is in any sense "optimum" (the method described is, indeed, close to optimum), then the assumptions mentioned are ~~realistic.~~ reasonable.

If S' contains 1000 sequences of "type S," then M_1 will take 3000 more bits to code this part of S' than will M_2. Let

we read this section past this point.

$$M_2(E \frown \mathrm{R\,A\,B\,a\,b\,k}) = S' \frown S$$

M

and

$$M_1(F) = S' \frown S$$

be the normal methods of coding for M_1 and M_2.

→ continued on bottom of page 16.

V131

10. A FOURTH OBJECTION: THAT EQUATION (1) CONSIDERS ONLY "MINIMAL" DESCRIPTIONS

Another objection to the method outlined is that Equation (1) uses only the "minimal binary descriptions" of the sequences it analyzes. It would seem that if there are several different methods of describing a sequence, each of these methods should be given <u>some</u> weight in determining the probability of that sequence.

In accordance with this idea, we will modify Equation (1) and write the probability that a, rather than b, will be the continuation of sequence T, will be

$$4) \qquad \lim_{\epsilon \to 0} \frac{\sum_{i=1}^{\infty} \left(\frac{1-\epsilon}{2}\right)^{N_{S_{ai}}}}{\sum_{i=1}^{\infty} \left(\frac{1-\epsilon}{2}\right)^{N_{S_{bi}}}}$$

this part comes after page 17.

this has been patched up for the next edition.

$$M_1(S_{a1}) = M_1(S_{a2}) = M_1(S_{a3}) = \cdots\cdots = M_1(S_{a\infty}) = T^\frown a$$

The S_{ai} are all the descriptions of $T^\frown a$.

Similarly,

$$M_1(S_{b1}) = M_1(S_{b2}) = \cdots\cdots = M_1(S_{b\infty}) = T^\frown b$$

$N_{S_{ai}}$ is the number of digits in S_{ai}.

See page 18 for how to elim. This.

The limit $\epsilon \to 0$ has been incorporated into the equation to overcome the objection in Section 5, that the sum of all the probabilities diverged. In Equation (4) it may not be necessary for ϵ to approach zero. It may be both expedient and adequate to let it take some small value like 0.001.

??

should be $\sim$ (length of coded corpus)$^{-1}$

from the bottom of page 15

Then

$$M_1(\alpha^\frown E^\frown RABabk) = S'^\frown S$$

continued on page 17.

V131

and if the string α contains less than 3003 bits, the code string $\alpha{}^\frown E{}^\frown R\,A\,B\,a\,b\,k$ will be <u>shorter</u> than the code F, so the "minimal" codes for both M_1 and M_2 will terminate in the sequence R A B a b k.

The figure "3003 bits" was arbitrary. In general, α will have a fixed number of bits, but the figure "3003" will be proportional to the length of the sequence $S'{}^\frown S$. As a result, <u>all</u> universal machines will tend to code long sequences ending in S by code sequences ending in R A B a b k , because coding methods of this type will be shortest in the long run.

It will be noted that this latter statement on the similarity of minimal codes for universal machines is not much more than a strong conjecture, with suggestions of how a proof might, under certain circumstances, be constructed.

More exactly, if S' is a very long sequence of a kind containing the kinds of information that a man would normally observe in his lifetime and

S is a short sequence.
M_1 and M_2 are both universal machines.
G_1, G_2, H_1 and H_2 are the shortest strings such that

$$M_1(G_1) = S', \qquad M_2(G_2) = S',$$
$$M_1(H_1) = S'{}^\frown S, \qquad M_2(H_2) = S'{}^\frown S.$$

N_{G_1} is the number of bits in G_1, with similar definitions for N_{G_2} etc.

Then we would like it to be true that

$$N_{H_1} - N_{G_1} = N_{H_2} - N_{G_2},$$

for all fairly short sequences, S, and all pairs of universal machines, M_1 and M_2.

this conjecture is <u>very</u> likely wrong. But it <u>should</u> be <u>almost</u> <u>true</u> for a large, "intuitively likely" sort of ~~univ~~ machines

The truth of this conjecture is a sufficient condition for the probability estimate of Equation (1) to be independent of just what machine was used (providing, of course, that it was a "universal machine").

An objection to this objection! If we derb all seqs of length n, by the code seq: $n\,p\,\alpha$, where α is the der of any seq. that starts out with A, and p is punctuation, then we automatically include the future all — n tells how many symbols to retain of $M(\alpha)$. — I think, heir, that this coding method yields the same result as equ(5).

11. LAST OBJECTION: THAT THE MORE DISTANT FUTURE OF THE SEQUENCE SHOULD BE CONSIDERED

The final objection that we will discuss at any length is that Equation (4) does not consider in any serious way the more distant future of the sequence being extrapolated. Consider, for example, the sequence a b c d a b c d a b c d a b. The next symbol is probably c, and this is so because the sequence a b c d a b c d a b c d a b c d has a particularly simple description, and is therefore very probable.

We take all possible future continuations of the sequence into account in the following ~~further~~ refinement of Equation (4):

NOTE

on Crit Cmi TM8 499.35—501.08)

$$5)\quad \lim_{\epsilon\to 0}\ \cancel{\lim_{n\to\infty}}\ \frac{\sum_{k=1}^{r^n}\sum_{i=1}^{\infty}\left(\frac{1-\epsilon}{2}\right)^{N(S_{Ta\,C_{n,k}})_i}}{\sum_{k=1}^{r^n}\sum_{i=1}^{\infty}\left(\frac{1-\epsilon}{2}\right)^{N(S_{Tb\,C_{n,k}})_i}}$$

This eq. is more or less shown to ~~be false~~ i.e., the limit → indip. of a or b. — I think its rite, how, if you sum on n in both num. and denom. from 1 to ∞ — see crit cmi TM8 499.30

We can elim. the necessity (and ad-hoc'ness) of ϵ, by considering only those descriptions (in both num. and denom) with m bits or less. Then let $m\to\infty$. There may be a way to make n ~~bee~~ unnecy. — By suitable coding ~~or~~ by simply considering all "finite" continuations of Ta (and Tb) that have $\le m$ bits in their dervs.

$C_{n,k}$ is a sequence of n symbols in the output alphabet of the universal machine. 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.

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

$(S_{Ta\,C_{n,k}})_i$ is the ith description of $Ta\,C_{n,k}$ with respect to Machine M_1.

$N(S_{Ta\,C_{n,k}})_i$ is the number of ~~digits~~ bits in $(S_{Ta\,C_{n,k}})_i$.

It can be shown that Equation (5) also eliminates the Third Objection in a very satisfactory way – i.e., the "bit wastage" in both numerator and denominator average out to be the same, and so they cancel. This cancellation does not ordinarily occur in Equation (4).

For simpler expressions for the num. and denom. of 5) that are very probably as correct as 5): For num: the no. of dervs that have exactly m bits, that derb. strings that begin "Ta" ...

see back side of page 17

There are 2^m binary ~~seqs.~~ strings containing just m bits.
A certain number of these strings describe sequences that begin with the subsequence T_a. ~~Let~~ ~~[illegible]~~
$N(m, T_a)$ is the number of such binary strings.

Then th. prob'ty ratio of interest, is

$$\lim_{m \to \infty} \frac{N(m, T_a)}{N(m, T_b)}$$

A poss. trouble with this: th. (imperfect) reasoning that leads one to believe that eq. 5) cancels quantization error (i.e. because of log distrib of lengths of $C_{n,k}$) does not hold here. However, since this method is identical to the ~~[illegible]~~ Σ method for many types of corpus, this criticism, if true, also would apply to the Σ method.

Hvv — Note that ZTB 140 applies to both equ's and th. "finite sequence method" — and ZTB 140 does not have quantization error.

If one uses th. "skew distrib." concept, Consider that a "0" takes T_1 seconds, and a "1" takes T_2 sec. Consider all ~~messages~~ input seqs of length $m \cdot (T_1 + T_2)$ seconds, ($m \gg T_1 + T_2$ and later, $m \to \infty$) ~~that start out with~~ ~~Σ~~ $\to$ they give an output that starts ~~[illegible]~~ with T_a. Th. Σ no. of such seqs. will be th. numerator.
Similarly consider seqs that give T_b as output and write th. Σ no. of them, for th. denominator.

12. AN INTERPRETATION OF EQUATION (5)

Equation (5) has at least one rather simple interpretation. Consider all possible sequences of symbols that could be descriptions of all the things a person might observe in his life. These sequences correspond to the sequences being coded in Equation (5), such as $\mathrm{Ta}\ C_{n,k}$.

Then a "complete model" that "explains" all regularities observed in these sequences is that they were produced by some arbitrary universal machine with a random binary sequence as its input. Equation (5) then enables us to use this model to obtain a priori probabilities to be used in computation of a posteriori probabilities using Bayes' Theorem. Equation (5) finds the probability of a particular sequence by summing the probabilities of all possible ways in which that sequence might have been created.

This particular model of induction is somewhat similar to that of Carnap (Ref. 1, page 562). Carnap restricts his discussions to only the simplest, finite languages, yet he is able to obtain some very reasonable results with this very limited means.

This is deceptive, "similar" — but not because of the "randomness" part!

Here, however, we use the full generality of description methods that are available through Turing machines.

A somewhat more general, ~~[illegible] [illegible]~~ and equally "complete" model may also be obtained, if we allow the input to the Turing machine to be any Markov chain of non-vanishing entropy.

or better: suppose that just before a person dies, he writes down a complete d[illegible] of everything he has experienced since his birth.

13. USE OF A SKEW INPUT DISTRIBUTION TO OVERCOME THE THIRD OBJECTION

The above model for Equation (5) suggests a very natural way to avoid the "bit wastage" inherent in the representation of numbers using any integral radix.

For Equation (5) we used as input to the universal machine, binary sequences in which 0's and 1's were equally probable. In such a situation, the probability of any particular input sequence was always a power of 2. However, suppose that we use the following type of input sequence for the machine:

probability of 0 is $\delta - \frac{1}{2}\epsilon$

probability of 1 is $1 - \delta - \frac{1}{2}\epsilon$

probability of termination of sequence is ϵ

Here again, the "expected length" of a sequence is about $1/\epsilon$. If δ is small, however, we can have very fine gradations of probability available in these sequences – much finer than the integral powers of 2.

It will be noted that the descriptions (i.e., input sequences to the machine) of a given output sequence that are "most probable " are now entirely different from the shortest (and therefore most probable) sequences that were used before for "minimal" descriptions. There exists, however, a translation method, so that it is possible to go from a "shortest" description using equal probabilities for 0 and 1, to a corresponding "most probable" description using the highly skewed distribution.

Using this highly skewed distribution, it is possible to devise sequences that correspond to any integers with arbitrarily little of the "bit wastage" that was evident when an integral radix was used for representation of numbers.

In general, the lengths of sequences of highest probability in the skew distribution that are needed to code a given text will be much longer than the corresponding code sequences using a symmetric distribution.

V131

14. APPLICATION OF THE METHOD TO CURVE FITTING

The application of Equation (4) to numerical extrapolation by means of "curve fitting" has been investigated to some extent. The problem is formulated in the following way: We are given a set of pairs of numbers that correspond to empirically observed data points – e.g., a set of pairs of temperature and pressure readings of a gas. We are then required to extrapolate this data – i.e., given a new temperature reading, to obtain the relative probability of any possible corresponding pressure reading.

An economical ~~A simple~~ method of describing such a set of data points is to give an equation that approximates the data, then give a set of temperatures, then a set of numbers that give the deviations of the empirical pressures from the equation.

We could conceivably try to express the list of temperatures in more compact form, but doing so would not affect the resultant probability ratios.

If the curve fits very well, the cost of the set of deviations will be smaller than for a curve that fits poorly. The cost of describing the equation must also be taken into account, so, in general, a 20-parameter polynomial could give a low-cost set of deviations for 20 empirical points, but the cost of the 20 parameters would be high. There will exist some optimum number of parameters that should be used, such that the total cost of the equation description and the deviation descriptions will be minimal. Here we make use of the fact that it costs less to code small numbers than large ones of the same absolute accuracy.

mention that it costs less to express small nos. than large ones, that the "opt. curve" according to these criteria is indep. of the no. of signif. figs. one wishes to retain throughout the analysis.

If using polynomials for curve fitting has been useful in the past, this method of description will have a low bit cost. Using unusual functions that have few parameters in them, yet are complex to describe and have been used infrequently in the past, will be very expensive to use for extrapolation, so one would tend not to use them unless they gave a very small set of deviations.

These latter notions are certainly what one feels to be true intuitively when one is fitting a curve to empirical data. The present method of analysis seems to put this intuitive idea on a quantitative basis.

An objection might be raised that the curve fitting method described is close to one that assumes a very un-normal distribution of empirical error – certainly a distribution quite different from that which is ordinarily observed.

If, however, in the sequence of data preceding the present problem, there have been many empirical situations in which the deviations had a normal distribution, or if there are enough empirical points in the present problem, then it will be less expensive to describe the deviations <u>as a normal distribution</u> than to simply list them. As a result, we would obtain something close to a mean-square goodness-of-fit criterion – with the added feature of taking the complexity of the curve used into account.

If the empirical data obtained corresponds to a known physical law, then there ~~may~~ will be much previous data to corroborate this law. In such a case, the equation will have been used many times in the past, and will be correspondingly less expensive to use in the present case.

If the physical law used has not been personally empirically verified by the curve fitter, through previous experimentation and he simply read about the law in a book, then the cost of the equation is somewhat more difficult to compute. It ~~will~~ depends, in part, upon the empirical accuracy in the life of the curve-fitter of physical laws that he has read about in books.

15. THE PROBLEM OF CONFLICTING LINES OF EVIDENCE

I think I could now write a much better soln. to this difty! See opp. pg. for a "try"!

An insurance company wants to determine the probability that a man will live over 60 years, and has compiled tables of data to aid in solving this problem.

One day a man ~~comes to~~ drifts into the office of the company and asks to be insured. He is 50 years old, has had pneumonia, and both of his parents died at the age of 95.

The insurance company has tables that tell the probability that a 50-year-old man who has had pneumonia will live to 60. The tables give the probability p_1.

They have tables that tell the probability that a 50-year-old man, both of whose parents lived to be over 90, will live to 60. The tables give the probability p_2.

They have no tables for 50-year-old-men who have had pneumonia and both of whose parents lived more than 90 years.

How shall the company combine the data from the two tables that it has?

It might be argued that it is impossible – that one must have a table for the coincidence of the three characteristics before one can make a probability estimate. However, every day we are forced to combine evidence of various kinds to make probability estimates, and in many cases the data is inadequate, as in the above problem – yet we make decisions based on such inadequate data. Indeed, it might be argued that there are few decisions that we do make in which we have "adequate data."

A very approximate analysis of this problem was made, using the coding method of probability evaluation. The probability, obtained, that the man will live more than 60 years ~~that was obtained~~ is

6)
$$\frac{p_1 + p_2 + \frac{p(V)}{p(R)}}{(1 - p_1) + (1 - p_2) + \frac{p(V)}{p(R)}}$$

This "soln." seems unreasonable if either p_1 or $p_2 = 1$.

V131

Say we use th. Σ method, and we use th. "correlation" method of coding. One pam, ~~that~~ using ~~th.~~ one correl., will pcost $\left(p_1^{p_1}(1-p_1)^{(1-p_1)}\right)^{N_1}$.

Th. other pam using th. other correl, will pcost

$\left(p_2^{p_2}(1-p_2)^{(1-p_2)}\right)^{N_2}$ (N_1 is no. of cases of p_1).
(N_2 . . . . p_2)

See HR 128 and 129 ff. for a very reasonable Soln. ~~to~~ This problem.

It is clear from this that p_1 and p_2 may not be a "suff statistic" to make v.g. predictions with..

An impt. Q., then is: Gn. some statistical proams., what is th. best way to use them to make predictions?

using HR 129.: p_1 = proby that if he has had pneum he will live to 60
p_2 = " " " . . not had . . . .

s_1 ~~p_3~~ = proby. that a random chosen person has had pneum.

for large samp. sz., we compare

$$p_3(p_1 \ln p_1 + q_1 \ln q_1) + q_3(p_2 \ln p_2 + q_2 \ln q_2)$$

$(q_i \equiv 1-p_i)$ for th. 2 methods of prediction — th. one that is greatest, wins.

Speaking very loosely, $P(V)$ and $P(R)$ are the relative frequencies with which the R coding method and the V coding method of Section 7 have been used in the past. In the present case, $P(V) / p(R)$ is probably much less than 1.

It is characteristic of the present method of induction that most probability values obtained are dependent, to some extent, on sequences of events that are apparently not very closely related to the events whose probabilities are being computed.

It should be noted that the validity of Equation (6) is not very certain, since it was obtained by using some very uncertain assumptions. These uncertain assumptions need not characterize the method and are symptomatic of the author's present inability to always devise good approximation methods for Equation (5).

16. GENERAL REMARKS ON EQUATION (5) AND ITS APPLICATIONS

While Equation (5) is put forth as what is hoped to be an adequate explication of conditional relative probability, the equation itself will not ordinarily be used directly for probability computation – any more than the definition of a Lebesgue integral is used directly in the evaluation of integrals.

Instead, Equation (5) can (and has been) used to obtain theorems about probability from which actual probabilities may be calculated. Among the techniques used are the discovery of coding methods that are simple to use, and nonminimal, yet from which it is possible to obtain the same probability ratios as those given by Equation (5). The apparently ad-hoc number manipulation of Sections 6 and 7 is an example of this, though a proof has not been given here.

Minimal coding techniques do have important direct applications, however. One of these is information retrieval. The minimal coding enables us to discard information that is least relevant to prediction, or to whatever ~~the~~ application ~~of~~ the coded information might ~~be~~ have. Coded information that is most valuable for prediction is also most likely to be correlated with other data, and for this reason, in coding new data, we examine relationships between it and parts of previously coded data that are of most value in prediction. How is this relevant?

Another direct application of minimal coding is in the generalized hill-climbing problem. Here, there is a set of continuous and/or discrete parameters that must be adjusted to maximize the value of a certain evaluation function. Organic evolution is an important example of a hill-climbing problem with discrete parameters. These parameters are the coded sequences that constitute the chromosomes. The evaluation function of such a set of coded sequences is the expected reproduction rate of the resultant organism. Perhaps write more on this.

The method used for hill-climbing in organic evolution of asexual organisms is to make each new set of trial parameters a random change of a few of the parameters of a fairly good organism. This random change corresponds to a mutation.

While there is some reason to believe that the genetic code description of the organism is not a minimal code, it shares with minimal codes the property that a random change of one of the code symbols will yield a code sequence for an organism that has a not-altogether-too-small probability of living and a somewhat smaller probability of being a bit better than his parent.

? Is this what I want?

None of the computing machine simulations of organic evolution have attempted representations or organisms using minimal codes, and it seems like a reasonably good thing to try.

REFERENCE

1. R. Carnap, Logical Foundations of Probability, University of Chicago Press, 1950.