Learning Probabilistic Prediction Functions

(Extended Abstract)

Alfredo DeSantis[1,3], *George Markowsky*[2], *Mark N. Wegman*[1]

[1] IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598
[2] Department of Computer Science, University of Maine, Orono, ME 04469

Abstract

We are interested in the question of how to learn rules, when those rules make probabilistic statements about the future. In this paper we discuss issues that arise when attempting to determine what a good prediction function is, when those prediction functions make probabilistic assumptions.

Learning has at least two purposes, 1) to enable the learner to make predictions in the future, and 2) to satisfy intellectual curiosity as to the underlying cause of a process. We will give two results related to these distinct goals. In both cases, the inputs are a countable collection of functions which make probabilistic statements about a sequence of events. One of our results shows how to find one of the functions, which generated the sequence, the other result allows us to do as well in terms of predicting events as the best of the collection. In both cases the results are obtained by evaluating a function based on a trade-off between its simplicity and the accuracy of its predictions.

1 Introduction

We are interested in the question of how to learn rules, when those rules make probabilistic statements about the future. An example of such a rule, might be that with each toss of a die, there is a one sixth chance of a one showing up. The results in this paper help give a philosophical and mathematical basis for the scientific method. The scientific method, in turn has resulted, for example in our current belief in Quantum Physics [4].

Before giving our results we review classical philosophical thoughts on the principal of induction. The principal of induction says that things in the future will behave as they have in the past. All human scientific knowledge comes either from laws derived by generalizing events in the past, induction, or by following the consequences of those laws, deduction. Deductive reasoning is mathematical reasoning based on an accepted set of axioms. These axioms are usually arrived at by inductive reasoning. Hume [Hu] asked whether there was any deductive justification for the principal of induction.

A physicist might say that he believes a given law will work in the future because it has always worked in the past. When questioned as to why this is a good justification, he might say that when he has used induction to derive laws in the past, they have continued to work (again in the past) thus he expects that these laws will again continue to work (but this time the assertion is about the future). This amounts to justifying the principal of induction by using the principal of induction.

A justification for the principle of induction first requires a more complete description of how we generalize from past events to a law. William of Ockham suggested a rule which has become known as Occam's razor: "pluralites non est ponenda sine necessitate" or multiplicity should not be posited without necessity [Mo]. This has been paraphrased as the simplest explanation which fits the available facts is the correct one.

Gold [Go] suggested a concrete model for induction that was later improved by Blum and Blum [BlBl]. We will refer to the latter one. They try to find a rule which explains or generates a sequence[5]. A rule which explains the sequences 1, 4, 9, 16, 25 ... is that the i^{th} element of the sequence is i^2. Note that this explanation can also be used to generate the sequence. They start out with a possibly countably infinite list of all explanations for the sequence. All written English forms a countable list; relating their model to Ockham's the earlier explanations in the list are the simpler. They examine each element in the se-

[3] Work done at IBM while on leave from Dipartimento di Informatica ed Applicazioni, Universita' di Salerno, 84100 Salerno, Italy.

[4] Quantum Physics makes probabilistic statements about the future, and suggests that randomness is inherent in nature. Other sciences, which are about the macro world make statements which are precise about an idealized world. For example, two objects of different weight will fall at the same speed. This statement will be subtlety off when actually measured. One object might be buffeted by the wind, or the two objects are not released at the same time, or other measurements are not precise. Thus, the prediction that would be made based on this exact law is that the two will fall approximately at the same time, and the measured difference in time will be gaussianly distributed around zero.

[5] Infering the rule which generates a sequence is no easier than infering a rule that describes experimental results based on the data about the experiment. If the experiments are written out, beforehand, then predicting the sequence models predicting the results. Our dependent prediction functions model what happens if the sequence of experiments change in view of the previous results.

0272-5428/88/0000/0110$01.00 © 1988 IEEE

quence and after this examination, announce their current guess as to the best explanation or generating function. One can "infer" the rule in their sense if there is a method converging on a current explanation. Consider the procedure of always announcing the earliest generator in the list consistent with the element of the sequence seen thus far. This follows Ockham's principle of using the simplest rule. If there is a generator of the sequence there must be an earliest generator. All explanations proceeding this one are wrong and hence there is some element of the sequence they do not explain. Thus after some number of elements have been seen, enough to eliminate all incorrect explanations proceeding the first correct explanations, this learning procedure will converge on the correct explanation. Blum and Blum further consider how to modify this procedure when the list contains functions which are not guaranteed to yield an answer in all cases.

If an explanation is in a list of total functions, the above procedure answer's Hume's question in the sense that this elaborated principle of induction is guaranteed to eventually get the right explanation. This procedure follows our understanding of the scientific method and the meaning of scientific theory. A scientific theory is always viewed as a theory and not a fact. It is always open to refutation, and scientists always keep an open mind. However, until disproof a theory is assumed to be correct.

The situation when judging probabilistic rules is more complex. This follows because a rule which says that "the next element in the sequence has a certain probability of being x and another probability of being y" may never be completely wrong, and hence cannot be eliminated. Occam's Razor will not distinguish between a simple explanation and a more complex one which fits the data better. Rissanen proposed a trade off between the accuracy of an explanation in predicting events and the index of the explanation. We propose a similar function of the index and the accuracy as a more precise version of Occam's razor and show that by choosing the explanation that minimizes the value of this function we can learn probabilistic explanations in the same way that the above procedure learned non-probabilistic explanations.

We assign each function an initial weight based on its index i. These weights sum to one. $6/(\pi i)^2$ might be such an initial weight. After a sequence of elements has been seen, we say the weight of a function is its initial weight times the probability it assigns to that sequence occurring. We call this the validity of a function. The function with the largest weight we will show is in some sense the "best" one. If these weights are normalized to sum 1 they may be used to obtain a prediction which is in some sense a "consensus" of all the other functions. This consensus is obtained by summing each function's prediction times its normalized weight. We show that this consensus can be computed and is almost as good as the best function.

Learning has at least two purposes, 1) to enable the learner to make predictions in the future, and 2) to satisfy intellectual curiosity as to the underlying cause of a process. This latter purpose may be augmented by the ability once the "underlying cause" is known to make predictions about wholly unrelated things. In this paper we discuss issues that arise when attempting to determine what a good prediction function is, when those prediction functions make probabilistic assumptions. We will give two results related to these distinct goals. In both cases, the inputs are a countable collection of functions which make probabilistic statements about a sequence of events. One of our goals is to find one of the functions, which generated the sequence which is obtained by choosing the function with the largest validity. The other result allows us to do as well in terms of predicting events as the best of the collection which is done by the "consensus" function. In both cases we are trying to "learn" a model of the sequence of events based on the given collection of functions.

It is useful to have an example in mind as we describe the results in this paper. A simple example is detecting the probability of a one showing on a die. We also use Markov chains (also known as stochastic automata) as a motivating example, since there are extant results that we can compare our results to. Markov chains are deterministic finite automata, with a set of states, a start state, and transitions, but also have probabilities on the transitions. These probabilities give a prediction about the next character in a sequence. We will assume, so that it is possible to give the machines an index, that the transition probabilities are rational.

Ziv and Lempel [ZiLe1], [ZiLe2] have given an algorithm which asymptotically compresses an infinite length sequence, as well as any Markov chain does. "As well" in this context means that the ratio of their compression to the best Markov[6] chain approaches one. The predictions of our first algorithm can be used to compress the sequence and will be only a fixed number of bits off the optimal. Our algorithm treats the Markov chains as black boxes, and works equally well for other models, for example stochastic DPDA's.

Markov chains may be used to generate a sequence. Given a sequence it may be valuable to determine the generating Markov process. Neither our first algorithm nor Ziv and Lempel's algorithm is able to determine this. All they do is to make predictions which are as good as the generating process would. Rudich in [Ru] attempts to determine the generating process from the input sequence provided that the process is drawn from certain limited kinds of Markov chains. However, he runs into trouble when two chains which have the same underlying transitions, but different start states can be different under some sequences but have the same probability of generating the particular sequence that has been generated (see Figure 1). We solve this problem, by discussing the concept that two processes are distinguishable with respect to the sequence.

A result similar to the one claimed by Rudich is fairly easy to derive given our second algorithm. Our second algorithm examines each element in the sequence, and after examining it chooses one prediction function as its current guess as to the generating function. Provided that all prediction functions are distinguishable with respect to the sequence and that there are only a countably infinite number of these functions, our algorithm is guaranteed to converge on the correct function. To obtain a result similar to Rudich's attempt one basically needs to show that it is possible to determine whether two functions are dis-

[6]Ziv and Lempel actually discuss compression by finite state transducers, but the two are much the same.

tinguishable with respect to the sequence. This is fairly straight forward.

In the next section we give more detail on the theoretical model. Section 3 compares the results we obtain with previous results, and Section 4 describes possible applications. Section 5 gives a formal treatment.

2 Our model

A prediction function is a function from a pair, X and Y to a conditional probability of Y given X, in much the same way that Bayesian analysis is done. Given a prediction[7] function f, and a string $s = s(0)s(1)...s(n)...$ the probability of its prefix of length n, $s_n = s(0)s(1)...s(n-1)$, being generated by f is $\prod_{i=0}^{n-1} f(s_i, s(i))$. Such a function can model a Markov process by taking the X's to be strings. $f(s_i, s(i))$ is computed by running the Markov process on the first i characters, ending in some state, and asking what the probability of taking the transition with $s(i)$ is.

By abuse of notation, when the string is clear we will say that $f(i)$ where i is an integer is the same as $f(s_i, s(i))$.

A prediction function can be said to generate a string $s = s(0)s(1)...s(n)...$ by asking for the probabilities of the various characters in the alphabet when preceded by the empty string, then flipping a fair coin to determine the first character. Then a coin is flipped to determine the second character given the probabilities for the characters following the first character. This process is repeated.

If one is attempting to find a good model of English text, using a Markov process, you are not attempting to find the Markov process which generated the text, since the text was generated by a human, and it is probably not a good idea to insist that humans are the same as Markov processes. Rather you are trying to find a Markov process which does as well as possible in predicting the next character. So, learning for the first purpose we stated, learning for the purpose of making predictions, seems more appropriate here than the second purpose mentioned, learning to find the underlying causes. We will measure the goodness of a predictor by its entropy. The entropy of f on the sequence s_n, is defined as $Ent_{f,n}(s_{n+1}) = -\sum_{i=0}^{n} \ln(f(i)) = -\ln \prod_{i=0}^{n} f(i)$. One of our goals is to minimize entropy.

Let F be a countable collection of prediction functions denoted $f_1, f_2, ...$ We assume that there is some way of obtaining the results of the f_i's (either they are computable or we have an oracle to give their answers). Our first result, Theorem 3, says that we can create a function G' whose results are computable in bounded time (and possibly bounded oracular queries). G' is a prediction function and when computing a prediction for the j^{th} term in the sequence, G' may examine the answers that any of the f_i's give for either the j^{th} term, or any of the preceding terms. The result says that the entropy of G' never exceeds $3 * log(i)$ plus the entropy of f_i. This compares favorably with Ziv-Lempel, who show that if the f_i's are created from Markov processes[6] that their function, h, has the property that $h(j)/f_i(j)$ approaches one or less as j goes to infinity.

[7] In Section 5 we will make a distinction between dependent and oblivious prediction function. We have here the definition of the more general dependent prediction function.

Attempting to find the generating function for a sequence out of a collection of functions is our model of the second purpose of learning, learning to find the underlying cause. However, certain definitional problems occur. Obviously, if two identical functions are in the class and one is the generating function, it will not be possible to determine which of the two functions is the generating one. In fact, if the two functions differ by only a little, it will still be impossible. Consider a sequence of coin flips. We might have one function, the generating function, which says there is a 50-50 chance of a head or a tail. We might have another function which says there is a 75 percent chance of a head on the first flip and a 50-50 chance on the remaining. It will be impossible to say which is the generating function, without additional knowledge, especially if the first flip is a head.

A further complexity arises because two functions may differ on some sequences and not on others. We might have a function which said that on the first flip heads appeared half the time, but if a head appeared first it would then occur 75 percent of the time. If a tail appeared first, then from then on heads and tails would be equally likely. Hence this function can only be distinguished from the 50-50 function when the sequence starts with a head. To solve this problem, we introduce the notion of whether two functions are distinguishable on a given sequence.

Two functions, f and g, are said to be distinguishable on a sequence if $\sum_{j=0}^{n} \sum_{y} (f(j,y) - g(j,y))^2$ goes to infinity as n goes to infinity and where y ranges over the alphabet. Note that this definition takes into account the sequence because that is implicit in the definition of f and g.

We will show that if f_i generates a sequence, and if with regard to that sequence f_i is distinguishable from f_j, then with any given probability and any value there is a point, n, at which $Ent_{f_j,n}(s_{n+1}) - Ent_{f_i,n}(s_{n+1})$ exceeds that value. Our second result follows from this.

Our second result, Theorem 11 says that with probability one, we can find the generating function from a countable collection of distinguishable functions.

We go on to show how to apply this result to find a Markov process which generates a given sequence. The steps of this proof show how the theorem should be applied. The fundamental steps are 1) to show that given any prefix of the sequence we can recursively enumerate the machines which are distinguishable with probability one. The definition of distinguishable relates to the entire sequence, which is infinite in length and hence can't be examined. However, if two machines have a pair of states with different transition probabilities and we can show that with probability one the machines will get into a pair of states with different transition probabilities infinitely often, then this suffices for our purposes. 2) to show that with probability one the machines will either become equivalent or get into states where they will be distinguishable with probability one. 3) to apply the second result.

3 Comparison with previous work

The field of inductive inference originated in the seminal work of Blum and Blum [BlBl]. In this work they were interested in whether you could tell which machine generated

a given sequence, and what the computational characteristics of the machine needed to be in order to be able to infer the machine which generated the sequence. Their notion was that after seeing each element in the sequence they would say which machine they thought generated the sequence. At some point they will converge on the correct machine. The simplest form of this learning process is very easy to understand, they simply report the machine with the smallest index that is consistent with the data. Eventually, all the machines with lower index than the generating machine will eliminate themselves.

In addition to conceiving this model, they showed that one could find the generating machine for larger computability classes than when predicting the next value in the sequence. This result leads to a different intuition than in our model, where one needs fewer caveats about predicting the next element with machines which probabilistically generate the sequence, than one needs when finding the generating machine.

Valiant [Va] has recently suggested a model which assumes that you are trying to learn a language, and elements are either in the language or not. One is randomly given an element and told that it is in the language. Sometimes you are also told that certain elements are not. This model involves probabilities, but not in the way that ours does. It also discusses the speed at which one converges to a solution, and we have not addressed this important question in the context of finding underlying causes. It may be possible to address it in our model. There have been papers discussing what happens when the oracle, telling you whether an element is in the language or not, occasionally makes an error. This comes closer to our model.

Much of the motivation and intuition for our work comes from the area of data compression. We have already discussed Ziv and Lempel's result. In one sense our results are better and more general. However, Ziv and Lempel's results are computable in realistic time, whereas ours are not. It would be interesting to see if it is possible to specialize our results to finite automata and improve the computational speed.

Rissanen is also a major contributor to the data compression literature. We first found the suggestion that the index can be combined with the entropy of the sequence to choose a good prediction function, although it had earlier been described by Solomonoff [So]. Rissanen's intuition, which we agree with, is that one wants to find the minimum information necessary to express the data that one is given. As with Chaitin-Kolmogorov complexity [Ch], [Ko], we find the smallest machine that can produce the output, where part of the machine contains the data about the output. To encode the machine from an infinite collection requires more than log bits, since it is necessary to say where the encoding of the machine ends, and the data begins. Thus, one needs the length of the encoding, and the length of the length of the encoding, etc... Rissanen suggests using log + log log + log log log + ... This function would suffice for the weighting function of Theorem 3 and of Theorem 9.

The intuition of combining the index with the accuracy of the prediction was one of the main inspirations for this work.

Rissanen also started on the path towards proving results about the power of what he calls *statistical inference*. In particular, he gives a result very similar to Theorem 7, which works with a fixed number of functions. He also extends it so that the functions can have real numbers in them. One of his examples is finding the "right" linear equation to determine an outcome from some measured inputs, and reals are important here. However, since he does not consider countable collections of functions, he gives no theorems which require using the index, he only gives intuition and empirical results. With any fixed collection Theorem 7 shows that the entropy alone suffices to find the correct function. We require the index to be considered in order to prove Theorem 8.

4 Possible applications

The difficulty in applying this work stems from the extreme computational overhead in dovetailing the various machines. However, there are examples where this overhead can be cut down. There are other examples where the intuition gained from this work and that of Rissanen can be applied, even if the results cannot be directly applied.

One example where the results might be applied comes from vision. Here, one is given a collection of pixel values. The sensor's which detect the brightness of each color typically have some noise associated with them, and their values can be assumed to be gaussianly distributed around the "correct" value. We would like to determine the underlying shapes that cause the pixel values. A shape might be flat or curved. If it is flat it is simpler, and hence a description of the scene can be shorter. We can think of each scene as giving probabilities to each pixel value. Hence, we are attempting to find the "correct" generator of probabilities for the pixel values. We can use numerical methods to determine the best curved surface to match the pixel values. One can almost always devise a curve which will match the data better than a flat surface. But there may not be enough data to justify the more complex model, and the results in this paper seem to be similar to some recent advances in vision [FaHe], [FaHePa].

Another example comes from Rissanen's papers. Suppose one wants to come up with a set of linear functions which explain some data. Again, numerical methods may be used to determine the best functions based on one variable, two variables, etc.. One may use the results in this paper to choose amongst them.

An example which does not quite fit the model comes from the work of Ziv and Lempel. They have an adaptive compression scheme which builds an automata that is based on their data. They show that eventually it will approach the best automata. But, this automata has a strange form. There is one state which is the root of the automata. All other states can only be entered from exactly one other state. So the transition diagram resembles a tree, with multiple entering edges only to the root. There is no mechanism for combining two sub-trees. So, the automata is often exponentially bigger than the "best" one. Theoretically this is not a problem (there are other problems with this approach, but we won't focus on them here). In practice the adaptation can be slowed significantly. Our results suggest a method for determining when two subtrees should be combined, and when they should be split

again.

5 Formal treatment.

Let $\mathcal{N}$ denote the *natural numbers*, i.e., the set $\{0, 1, \ldots\}$, $\mathcal{R}$ the *real numbers*, $\mathcal{Y}$ a *fixed countable set*, $\mathcal{S}$ the set of all sequences from $\mathcal{N}$ into $\mathcal{Y}$, i.e., $\mathcal{S} = \{s : \mathcal{N} \to \mathcal{Y}\}$.

Definition 1 *A function $f : \mathcal{N} \times \mathcal{Y} \to \mathcal{R}$ is called an* oblivious prediction function *if $\forall i \epsilon \mathcal{N}, \sum_{y \epsilon \mathcal{Y}} f(i, y) = 1$. The set of all oblivious prediction functions will be denoted by $\mathcal{P}$.*

An example of an oblivious prediction function is the function which says that all flips of a coin have an even chance of being a head or a tail. We call it oblivious since it does not care about the previous flips.

Definition 2 *A function $f : \mathcal{Y}^* \times \mathcal{Y} \to \mathcal{R}$ is called a* dependent prediction function *if $\forall s_i \epsilon \mathcal{Y}^i$, $\sum_{y \epsilon \mathcal{Y}} f(s_i, y) = 1$.*

Dependent prediction functions are used to model functions where, as in sequences generated by Markov process, the probability of the i^{th} element being a particular element depends on previous elements. A dependent prediction function f can be used to generate a sequence. The first element of the sequence is chosen from $y \in \mathcal{Y}$ by flipping a fair coin based on the probabilities $f(s_0, y)$. Note that s_0 is defined as the empty sequence. Once the first element is chosen, s_1 is defined, and second element is based on $f(s_1, y)$. The i^{th} element is based on $f(s_i, y)$.

One useful quantity is the *entropy* defined as follows. We use the natural logarithm ln for ease of computation. Since $\log_2$ is a constant multiple of ln all the results in this paper can be readily expressed in terms of $\log_2$.

Definition 3 *Given an oblivious prediction function, f, and a sequence $s \epsilon \mathcal{S}$ the* f-Entropy of s, *denoted by $Ent_f(s)$, is defined as $-\sum_{i=0}^{\infty} \ln f(i, s(i))$. The symbol $Ent_{f,n}(s)$ is used to represent the quantity $-\sum_{i=0}^{n} \ln f(i, s(i))$.*

5.1 Learning for the purpose of making predictions.

In order to produce a dependent prediction function which is guaranteed not to be much worse than the best in a countable collection of dependent prediction functions, we will give each function in the collection a weight, and average its predictions times that weight to achieve the prediction we will actually use. Since each function will always have positive weight, we will never completely discount any y unless no functions "thinks" it is possible.

We give each function, f_i an initial weight $w_{i,-1}$ with the property that $\sum_{j=1}^{\infty} w_{j,-1} = 1$. For example $w_{i,-1} = 6/(\pi i)^2$.

The function *val* of a dependent prediction function f_i in a countable collection is defined as

$$val(i, n, s) = w_{i,-1} \prod_{k=0}^{n} f_i(s_k, s(k)).$$

We will later show that *val* is related to the validity function of Section 5.2. The weight $w_{i,n}$ of the i^{th} function after the n^{th} term is defined by

$$w_{i,n} = \frac{w_{i,-1} e^{-Ent_{f_i,n}(s)}}{\sum_{j=1}^{\infty} w_{j,-1} e^{-Ent_{f_j,n}(s)}} = \frac{val(i, n, s)}{\sum_{j=1}^{\infty} val(j, n, s)}.$$

If we had infinite compute power we would use the following as the best guess as to the actual probability

$$G(s_n, s(n)) = \sum_{j=1}^{\infty} w_{j,n-1} f_j(s_n, s(n))$$

Note that $\sum_{i=1}^{\infty} w_{i,n} = 1$ thus $\sum_{y \epsilon \mathcal{Y}} G(s_n, y) = 1$. Hence G is a dependent prediction function.

Theorem 1 *Let $\{f_i\}$ be a countable collection of dependent prediction functions. The entropy of G on the first $n+1$ terms of the sequence s, $-\sum_{i=0}^{n} \ln(G(s_i, s(i)))$, is less than or equal to $-\ln(w_{j,-1}) + Ent_{f_j,n}(s)$, for all j.*

Proof. Entropy of G on the first n terms of sequence $s =$

$$-\sum_{i=0}^{n} \ln\Big(G(s_i, s(i))\Big) = -\sum_{i=0}^{n} \ln\left(\sum_{j=1}^{\infty} w_{j,i-1} f_j(s_i, s(i))\right)$$

(by definition of G)

$$= -\sum_{i=0}^{n} \ln\left(\sum_{j=1}^{\infty} \frac{w_{j,-1} \prod_{k=0}^{i-1} f_j(s_k, s(k))}{\sum_{l=1}^{\infty} w_{l,-1} \prod_{k=0}^{i-1} f_l(s_k, s(k))} f_j(s_i, s(i))\right)$$

(by definitions of w)

$$= -\ln\left(\prod_{i=0}^{n} \frac{\sum_{j=1}^{\infty} w_{j,-1} \prod_{k=0}^{i} f_j(s_k, s(k))}{\sum_{j=1}^{\infty} w_{j,-1} \prod_{k=0}^{i-1} f_j(s_k, s(k))}\right)$$

(reorganizing terms)

$$= -\ln\left(\frac{\sum_{j=1}^{\infty} w_{j,-1} \prod_{k=0}^{n} f_j(s_k, s(k))}{\sum_{j=1}^{\infty} w_{j,-1}}\right)$$

(canceling numerator of one term with the denominator of the next)

$$= -\ln\left(\sum_{j=1}^{\infty} w_{j,-1} \prod_{k=0}^{n} f_j(s_k, s(k))\right)$$

(Since the denominator equals 1, by definition)

$$\leq -\ln\left(w_{j,-1} \prod_{k=0}^{n} f_j(s_k, s(k))\right) = -\ln(w_{j,-1}) + Ent_{f_j,n}(s).$$

□

With the same technique we can prove the following:

Theorem 2 *Let $\{f_i\}$ be a countable collection of dependent prediction functions. The entropy of G on the first $n+1$ terms of the sequence s satisfies*

$$-\sum_{i=0}^{n} \ln\Big(G(i, s(i))\Big) \leq \sum_{i=1}^{\infty} w_{i,-1} Ent_{f_i,n}(s)$$

With finite computation we can approximate G, by only considering the most highly weighted functions. If we consider a larger percentage of the weight as the number of terms increases, then we will still converge on the "right" guess.

An example of the kind of theorem we can prove is the following:

Theorem 3 *There is a computable dependent prediction function, G', such that for any predictor, f_j, from a countably infinite collection of dependent prediction functions, provided that $\mathcal{Y}$ is finite, G' will have entropy no greater than $Ent_{f_j} + O(3 \ln j)$.*

Proof. Create a new countable collection of dependent prediction functions by adding a new function to the old collection and renumber the old functions so that the new function has index 1. The new function assigns equal probability to all y's. Let $Ent_{f_1,n}$ be the entropy of this function after $n+1$ steps (this entropy is computable before seeing the y value). Let $w'_{i,n} = 0$ if $i > e^{Ent_{f_1,n}}$ and $(1/i^2)\prod_{k=0}^{n} f_i(k, s(k))$ otherwise. w is computed by normalizing w'. The entropy of the method will be $Ent_{f_1,n}$ until $e^{Ent_{f_1,n}}$ exceeds i. Hence, the method's entropy will exceed that of Theorem 1 by $\ln i$. Theorem 1's function exceeds functions by not more than twice the ln of their index. □

Thus G of Theorem 1 becomes the "consensus" function we promised in the introduction. Theorem 3 shows how to compute G', an approximation to G, which retains G's desirable properties.

5.2 Learning for the purpose of finding underlying causes.

In this section we show how to find the function which generated a sequence from a countable collection of possible generators. If two functions in the list are identical we can not tell which one was the correct generator. Hence, we require that the list only contains distinct functions. Moreover, if two functions are identical except for the probabilities they assign to the first event, we still will not be able to be sure which one generated the sequence. Hence we will introduce the L_2 norm and define the notions of weakly and strongly distinguishable functions.

Theorem 7 gives a sufficient condition to ultimately converge on the generator of a sequence from two distinguishable oblivious prediction functions.

So, we can use Theorem 7 to determine which of two coins (with different probabilities of head and tail) was used to generate a sequence.

Theorem 8 generalizes Theorem 7 to allow a countable number of generators to be chosen from. Thus, Theorem 8 will allow us, for example, to converge on a rational number which is the probability that a coin produces a head (assuming that we know that the probability is rational, and thus that there are only a countable number of coins to choose from).

Theorem 11 shows when the ideas of Theorem 8 can be applied in the context of dependent prediction functions. So, for example, they allow us to converge on a Markov process (assuming rational probabilities) which generated a sequence.

Definition 4 *Let f be an oblivious prediction function. Then the L_2* norm *of f, denoted by $\|f\|_2$ is defined as*

$$\sqrt{\sum_{i \in \mathcal{N}} \sum_{y \in \mathcal{Y}} f^2(i,y)}$$

The symbol $\|f\|_{2,n}$ is used to denote the quantity

$$\sqrt{\sum_{i=0}^{n} \sum_{y \in \mathcal{Y}} f^2(i,y)}$$

As usual, a norm can be used to define distances between pairs of elements in $\mathcal{P}$. Thus the L_2 distance *between a pair of elements, f, g in $\mathcal{P}$ is defined as $\|f - g\|_2$.*

We use the norm to enable us to tell whether two prediction functions are distinguishable. We can determine which of two distinguishable functions is the one which generated a sequence. The norm provides us with a way of comparing two functions and, as we will show, determining that if one generated a sequence then it will eventually have much smaller entropy on that sequence.

The fact that $\| \ \|_2$ is a norm follows from the fact that it can be defined in terms of an inner product on $\mathcal{P}$. For the relevant theorems and definitions see Sections 3.1 and 3.2 of the book by Ash [As].

Each oblivious prediction function gives rise to a probability measure on the set of sequences $\mathcal{S}$ in a fairly natural way. To explain this idea we need the following definition.

Definition 5 *For each oblivious prediction function f and natural number i there is a natural probability measure on $\mathcal{Y}$ denoted by $Prob_{f,i}$ and defined as*

$$\forall Y' \subseteq \mathcal{Y}, Prob_{f,i}(Y') = \sum_{y \in Y'} f(i,y).$$

It is clear that each $\text{Prob}_{f,i}$ is a probability measure on $\mathcal{Y}$. From these measures there is a natural way to create a probability measure Prob_f on the set of sequences $\mathcal{S}$, by looking at each position of the sequences and considering the corresponding probabilities. This is described in the following theorem, which is essentially Corollary 2.7.3 of Ash's book [As].

Theorem 4 *Given an oblivious prediction function, f, there exists a* unique *probability measure, $Prob_f$ on $\mathcal{S}$ having the following property: for all finite subsets M of $\mathcal{N}$ and subsets $\{Y_i \subseteq \mathcal{Y} \mid i \in M\}$,*

$$Prob_f(\{s \in \mathcal{S} \mid \forall i \in M, s(i) \in Y_i\}) = \prod_{i \in M} Prob_{f,i}(Y_i).$$

Thus, we can determine the probability of one sequence of events or the probability of a set of sequences of events.

Definition 6 *Given two oblivious prediction functions, f and g, and a sequence $s \in \mathcal{S}$, the* f,g-Entropy of s, *denoted by $Ent_{f,g}(s)$, is defined as*

$$\sum_{i=0}^{\infty} \ln \frac{f(i, s(i))}{g(i, s(i))}.$$

The symbol $Ent_{f,g,n}(s)$ is used to represent the quantity

$$\sum_{i=0}^{n} \ln \frac{f(i,s(i))}{g(i,s(i))}.$$

Note that $\mathrm{Ent}_f(s) = -\ln(Prob_f(s))$.

It is easy to see that $\mathrm{Ent}_{f,g}(s) = \mathrm{Ent}_g(s) - \mathrm{Ent}_f(s)$ whenever the limits make sense. Also, it is always true that $\mathrm{Ent}_{f,g}(s) = -\mathrm{Ent}_{g,f}(s)$ and $\mathrm{Ent}_{f,g,n}(s) = -\mathrm{Ent}_{g,f,n}(s)$. Later in this paper some simple conditions are presented that imply

$$\sum_{i=0}^{\infty} \ln \frac{f(i,s(i))}{g(i,s(i))} = \infty$$

almost everywhere (a.e.) with respect to Prob_f. If this occurs, we will know that f is more likely the generating function for s.

Definitions 7, 8, 10 and 11 define various form of distinguishable collection. All these collections have the property that the "wrong" function will eventually have larger entropy than the "correct" function. Hence, by choosing the function with the smallest entropy, one eventually will converge on the "correct function".

Definition 7 *Two oblivious prediction functions, f and g, are said to be* weakly distinguishable *if given $\epsilon > 0$ and a natural number n, there exists a natural number k_0 such that $\forall k \geq k_0$,*

$$Prob_f(Ent_{f,g,k} < n) < \epsilon \quad and \quad Prob_g(Ent_{g,f,k} < n) < \epsilon.$$

Definition 8 *Two oblivious prediction functions, f and g, are said to be* strongly distinguishable *if $Prob_f(Ent_{f,g} = \infty) = 1$ and $Prob_g(Ent_{g,f} = \infty) = 1$.*

The following theorem shows that, as one would hope, strongly distinguishable is at least as restrictive condition as weakly distinguishable.

Theorem 5 *If two oblivious prediction functions f and g are strongly distinguishable, then they are weakly distinguishable.*

The following definition provides another approach to the concept of strongly distinguishable and is a very useful way to understand that concept.

Definition 9 *Two probability measures, $Prob_1$ and $Prob_2$ on the same measure space X, are said to be* mutually singular *if there exists a subset A of X such that $Prob_1(A) = Prob_2(S - A) = 1$. This is indicated by the notation $Prob_1 \perp Prob_2$.*

Theorem 6 *If f and g are strongly distinguishable oblivious prediction functions, then $Prob_f \perp Prob_g$.*

The definitions of distinguishability for two functions naturally extend to a finite set of oblivious prediction functions.

Definition 10 *A sequence $f_1, f_2, ..., f_N$ of oblivious prediction functions, is said to be* weakly distinguishable *if given $\epsilon > 0$ and a natural number n, there exists a natural number k_0 such that*

$$\forall k \geq k_0, \forall i \neq j, Prob_{f_i}(Ent_{f_i,f_j,k} < n) < \epsilon.$$

Definition 11 *A sequence $f_1, f_2, ..., f_N$ of oblivious prediction functions, is said to be* strongly distinguishable *if $Prob_{f_i}(Ent_{f_i,f_j} = \infty) = 1$ for $i \neq j$.*

Theorem 7 *Let f and g be oblivious prediction functions. If $\|f - g\|_2 = \infty$ then f and g are weakly distinguishable. If $\exists k, j \in \mathcal{N}$ such that $\sum_{n=j}^{\infty} (\|f - g\|_{2,n})^{-k}$ converges, then f and g are strongly distinguishable.*

Corollary 1 *Let f and g be oblivious prediction functions. If a constant $c > 0$ exists such that, for all i but finitely many, $\sum_{y \in \mathcal{Y}} (f(i,y) - g(i,y))^2 \geq c$, then f and g are strongly distinguishable.*

Now we mention a use of Theorem 7 in hypothesis testing. To give evidence that Theorem 7 is a powerful result, we will show how the strong law of large numbers can be derived from it.

Example 1 *Let $coin_1$ and $coin_2$ be two coins with probabilities of an outcome HEAD q and $q + \delta$, respectively, with $\delta > 0$. Suppose we are given an infinite sequence of outcomes generated by using only one of the two coins.*

Define two oblivious prediction functions f and g over the domain $\mathcal{N} \times \{H, T\}$, as $f(i,H) = 1 - f(i,T) = q$ and $g(i,H) = 1 - g(i,T) = q + \delta$, for all natural numbers i. Since $|f(i,H) - g(i,H)|^2 + |f(i,T) - g(i,T)|^2 = 2\delta^2$, by Corollary 1 it follows that f and g are strongly distinguishable, that is $Prob_f(Ent_{f,g} = \infty) = 1$ and $Prob_g(Ent_{g,f} = \infty) = 1$. These two conditions can be also written, respectively, as

$$Prob_f\left(\lim_{k\to\infty}\left(n_k \ln \frac{q(1-q-\delta)}{(1-q)(q+\delta)} + k \ln \frac{1-q}{1-q-\delta}\right) = \infty\right) = 1$$

and

$$Prob_g\left(\lim_{k\to\infty}\left(n_k \ln \frac{(q+\delta)(1-q)}{q(1-q-\delta)} + k \ln \frac{1-q-\delta}{1-q}\right) = \infty\right) = 1$$

where n_k is the number of HEAD in the first k outcomes. In order that the two limits hold the following must be true

$$Prob_f(n_k/k < \alpha(q,\delta), \text{ for all but finitely many } k) = 1$$

and

$$Prob_g(n_k/k > \alpha(q,\delta), \text{ for all but finitely many } k) = 1$$

where

$$\alpha(q,\delta) = \left(\ln \frac{1-q-\delta}{1-q}\right) \Big/ \left(\ln \frac{q(1-q-\delta)}{(1-q)(q+\delta)}\right).$$

Some algebra shows that

$$q < \alpha(q,\delta) < q + \delta$$

and thus we have that, as to be expected, the relative frequency of HEAD is close to the probability of the coin used. Since δ was an arbitrary, but positive constant, we have derived, as a corollary of our Theorem 7, the well known strong law of large numbers, namely that if S_k is the number of successes in the first k trials of a sequence of Bernoulli trials, where q is the probability of success, then, for every $\delta > 0$ with probability one there occur only finitely many of the events

$$\left|\frac{S_k}{k} - q\right| > \delta.$$

Another consequence of Theorem 7 is the following

Corollary 2 *Let $f_1, f_2, ..., f_N$ be oblivious prediction functions. If $\|f_i - f_j\|_2 = \infty$, $1 \leq i < j \leq N$ then $f_1, f_2, ..., f_N$ are weakly distinguishable. If $\exists l, k \in \mathcal{N}$ such that $\sum_{n=l}^{\infty}(\|f_i - f_j\|_{2,n})^{-k}$ converges for $i \neq j$, then $f_1, f_2, ..., f_N$ are strongly distinguishable.*

The above corollary gives us a simple sufficient condition for a finite set of oblivious prediction functions to be weakly or strongly distinguishable. The same conditions are not guaranteed to work for a countable infinite collection of functions $f_1, f_2, ..., f_j, ...$, as the following example shows.

Example 2 *Let f and g be two arbitrary weakly distinguishable oblivious prediction functions with the same domain, and let α_i be the i^{th} rational number (according to an indexing of the rational numbers). Define the functions $h_i = \alpha_i f + (1 - \alpha_i)g$, for each natural number i. The sequence of oblivious prediction functions $h_1, h_2, ..., h_j, ...$ is pairwise weakly distinguishable, that is any two functions are weakly distinguishable, but Corollary 2 cannot be extended to them. More formally we will prove in the final paper that given $\epsilon > 0$ and natural numbers n, k and i, the set of functions h_j such that $Prob_{h_i}(Ent_{h_i,h_j,k} > n) < \epsilon$ is infinite.*

Therefore we cannot use the same criterion suggested by Corollary 2 when dealing with a infinite number of oblivious prediction functions. Nonetheless we make use of Corollary 2 to find the function really used to generate the observed sequence. We will not use the entropy Ent_f as the only tool to distinguish among functions, but we will take also in account the index i of each function. Informally speaking we will give a weight to functions, based on their indexing.

Now we give a definition of distinguishability for an infinite collection of functions. Let s_k denote the k-tuple $(s(0), s(2), ..., s(k-1))$.

Definition 12 *Let $F = \{f_1, f_2, ..., f_i, ...\}$ be a countable collection of oblivious prediction functions. A function $V : \mathcal{N} \times \mathcal{N} \times \mathcal{Y}^* \rightarrow \mathcal{R}^+$ is called a* validity *function for F if, given a natural number k and a k-tuple $s_k \in \mathcal{Y}^k$, it is possible to compute the minimum $\min_{i \in \mathcal{N}} V(i, k, s_k)$. Given a natural number j, the function $V(j, \cdot, \cdot)$ is called the validity of the function f_j in F.*

Definition 13
A countable collection $F = \{f_1, f_2, ..., f_i, ...\}$ of oblivious prediction functions is said to be weakly distinguishable *if a validity function V for F exists such that, given $\epsilon > 0$ and natural numbers n and i, there exists a natural number k_0 such that*

$$\forall k \geq k_0, \forall j \neq i, Prob_{f_i}(V(j, k, s_k) - V(i, k, s_k) < n) < \epsilon$$

Definition 14
A countable collection $F = \{f_1, f_2, ..., f_i, ...\}$ of oblivious prediction functions is said to be strongly distinguishable *if a validity function V for F exists such that, for each $i \in \mathcal{N}$*

$$Prob_{f_i}\left(\lim_{k \to \infty}(\min_{j \neq i} V(j, k, s_k) - V(i, k, s_k)) = \infty\right) = 1.$$

The existence of a validity function for a countable collection F of oblivious prediction functions naturally leads to an algorithm for finding the generating function f_i, provided that the collection F is weakly distinguishable. Indeed after reading the k^{th} letter $s(k)$ we choose the function f_j with the minimum validity in F, i.e. that minimize $V(j, k, s_k)$.

The following theorem gives a sufficient condition for the existence of a validity function for a countable collection of oblivious prediction functions.

Theorem 8 *Let $F = \{f_1, f_2, ..., f_i, ...\}$ be a countable collection of oblivious prediction functions. If $\|f_i - f_j\|_2 = \infty$, for $i \neq j$, then F is weakly distinguishable. If for any i, j, such that $0 \leq i < j < \infty$, $k, l \in \mathcal{N}$ exist such that $\sum_{n=l}^{\infty}(\|f_i - f_j\|_{2,n})^{-k}$ converges, then F is strongly distinguishable.*

Theorem 9 *$-\ln val(i, k, s)$ suffices as a validity function to obtain the result claimed in Theorem 8.*

Validity functions must be constructed carefully. At one point, we thought that $\log i + Ent_{f_i,n}(s)$ would be a validity function, but we were able to construct a counterexample.

These theorems are thus the analogue of Theorem 7 but for a countable collection of oblivious prediction functions. The validity function plays the same role of the entropy function, assuring the existence of a natural algorithm for finding the generating function.

Thus it accomplishes our purpose. From the output of an arbitrary but unknown oblivious prediction function we can "learn" a model of the sequence of events based on the given collection of functions. In other words, if we are given a countable number of explanations (modelled by probabilistic functions), and experimental results (the sequence output of an arbitrary but unknown of this functions), it is possible to "learn" the correct explanation for the experimental results (there exists an algorithm that will find the function used to generate the sequence).

5.2.1 Dependent Prediction Functions.

As for oblivious prediction functions, each dependent prediction function gives rise to a probability measure on the set of sequences $\mathcal{S}$ in a fairly natural way. For this aim we need the following definitions.

Let σ be a finite string and denote by $\sigma(k)$ its k^{th} symbol and by σ_j its prefix of length j, i.e. $\sigma_j = \sigma(0)\sigma(1)...\sigma(j-1)$.

Definition 15 *For each dependent prediction function f and natural number i there is a natural probability measure on $\mathcal{Y}^i$ denoted by $Prob_{f,i}$ and defined as*

$$\forall \sigma \in \mathcal{Y}^i, \quad Prob_{f,i}(\sigma) = \prod_{j=0}^{i-1} f(\sigma_j, \sigma(j)).$$

Each $\text{Prob}_{f,i}$ is a probability measure on $\mathcal{Y}^i$. From these measures there is a natural way to create a probability measure Prob_f on the set of sequences $\mathcal{S}$, by looking at each position of the sequences and considering the corresponding probabilities, much as is done with oblivious prediction functions. This is described in the following theorem.

Theorem 10 *Given a dependent prediction function, f, there exists a* unique *probability measure, $Prob_f$ on $\mathcal{S}$ having the following property: for all finite subsets M of $\mathcal{N}$ and subsets $\{Y_i \subseteq \mathcal{Y} \mid i \epsilon M\}$,*

$$Prob_f(\{s \epsilon \mathcal{S} \mid \forall i \epsilon M, s(i) \epsilon Y_i\}) = \sum_{\{\sigma \mid \forall i \epsilon M, \sigma(i) \epsilon Y_i\}} Prob_{f,m}(\sigma).$$

where $m = \max\{i \in M\}$.

There is a close relationship between dependent prediction functions and oblivious prediction functions. From a dependent prediction function it is possible to derive an oblivious prediction function, based on a fixed infinite sequence.

Definition 16 *An oblivious prediction function f' is said to be* s-derived *from a dependent prediction function f and a sequence $s \in \mathcal{S}$ if $f'(i,y) = f(s_i, y)$, for all $i \in \mathcal{N}$ and $y \in \mathcal{Y}$.*

There is a natural way to extend the notions of distinguishability from oblivious prediction functions to dependent prediction functions, using the above concept of s-derivation.

Definition 17
A countable collection $F = \{f_1, f_2, ..., f_i, ...\}$ of dependent prediction functions is said to be weakly (strongly) s-distinguishable *if the collection of s-derived prediction functions $F' = \{f'_1, f'_2, ..., f'_i, ...\}$ is weakly (strongly) distinguishable.*

Notice that the notion of distinguishability for dependent prediction functions is based on a fixed sequence s. It can be the case that two dependent prediction functions are s-distinguishable only for some sequences s, as the following examples show.

Example 3 *Let $\mathcal{Y} = \{a,b\}$. And let f' and g' be two weakly (strongly) distinguishable oblivious prediction functions, with $f'(0,a) = f'(0,b) = g'(0,a) = g'(0,b) = 1/2$. Define two dependent prediction functions g and f as follows, for any $i > 0$, $s_i \in \mathcal{S}$ and $y \in \mathcal{Y}$,*

$$f(s_i, y) = f'(i,y)$$
$$g(s_i, y) = \begin{cases} f'(i,y), & \text{if } s(0) = a \\ g'(i,y), & \text{if } s(0) = b. \end{cases}$$

It is easily seen that f and g are weakly (strongly) s-distinguishable if and only if the first digit of s is b.

A more natural and intriguing example is the following

Example 4 *Consider the two Markov chains of Figure 1, each with 4 states, $state_0$, $state_1$, $state_2$, and $state_3$. They output one of the three symbols of the alphabet $\mathcal{Y} = \{a,b,c\}$, whenever they make a transition.*

Essentially they are the same chain but with a different initial state. If the first output symbol is b, then the two sources reach the same state after the first transition: hence they will have the same behavior from this point on. But if the first symbol is c, and the second is b, then the two sources will never reach a point in which they will have the same behavior.

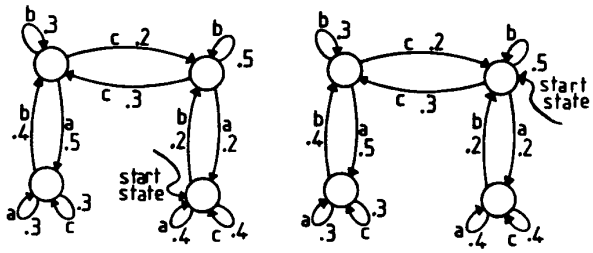

Figure 1

Each Markov source defines a dependent prediction function in a natural way. The two dependent prediction functions are strongly s-distinguishable for any sequence s that begins with cb. But they are not strongly s-distinguishable for any sequence s that begins with b.

Thus far we have defined the notion of distinguishability with respect to an infinite sequence. It may be the case, as the above examples show, that all sequences (but a set of measure zero) with a common finite prefix σ always give rise to s-derived distinguishable prediction functions. This leads us to the notion of distinguishability with respect to a finite string σ. Informally, two dependent prediction functions are distinguishable after a certain string σ if they are distinguishable on all the sequences in $\mathcal{S}$ that have σ as prefix, except for a set of measure zero. We will give a formal definition of this notion.

Let σ be a finite string. Denote by H_σ the set of all the sequences $s \in \mathcal{S}$ with σ as prefix, i.e. $H_\sigma = \{s \in \mathcal{S} \mid \forall i < |\sigma|, s(i) = \sigma(i)\}$.

Definition 18 *Two dependent prediction functions, f and g, are said to be* weakly *(strongly)* σ-distinguishable *if the set H of sequences $s \in H_\sigma$ for which they are weakly (strongly) s-distinguishable satisfies*

$$Prob_g(H) = 1 \text{ and } Prob_f(H) = 1$$

Example 5 *We will refer to Examples 3 and 4 to illustrate the notion of σ-distinguishability.*

Consider the Example 3. Then f and g are weakly b-distinguishable, but they are not weakly s-distinguishable for any of the sequences s with a as a prefix.

Now, consider the Example 4. The two dependent prediction functions defined by the Markov sources are strongly cb-distinguishable, but they are not weakly s-distinguishable for any of the sequences s with b as the first symbol.

The following corollary shows that strongly σ-distinguishable is at least as restrictive condition as weakly σ-distinguishable.

Corollary 3 *If two dependent prediction functions f and g are strongly σ-distinguishable, then they are weakly σ-distinguishable.*

Theorem 11 *Let F be a countable collection of dependent prediction functions. Let s be a sequence generated from $f \in F$. If*

1. *for all s_i there is a recursive enumeration of functions g in F such that g and f are weakly s_i-distinguishable.*
2. *with probability one s will separate functions g_i into one of two class, either those that will be listed in the enumeration above or those for which an i_0 exists such that for all $i > i_0$ $g(s_i, y) = f(s_i, y)$.*

then we can find the generating function or one which is not s_k-distinguishable after an initial prefix s_k.

A consequence of Theorem 11 is the following

Corollary 4 *Let F be a countable collection of dependent prediction functions. Let s be a sequence generated from $f \in F$. If for all s_i there is an algorithm that for any two functions in F outputs yes or no, depending if they are weakly s_i-distinguishable, then we can find the generating function or one which is not s_k-distinguishable after an initial prefix s_k.*

We give the next corollary as an illustration of the way in which Theorem 11 may be used.

Corollary 5 *There is an algorithm which, given a sequence generated by any Markov model, with probability one finds a Markov model which gives identical probabilities to the original Markov model after a finite prefix.*

Proof's sketch: First we must establish that two Markov models will ultimately become identical or infinitely often give different probabilities. To do this we consider the automata which is the cross product of the two Markov models. This automata has a state corresponding to each of the pair of states from the two machines. Eliminate all transition which both models give zero probability. Ultimately, this automata, with probability 1, must end up in a strongly connected region with no exits from the region except possibly exits with zero probability. In this region either the transition probabilities given by the two models are the same, or they differ. If there is a transition on which the two machines give different probabilities, then with probability 1 that will be visited infinitely often. Thus, there is some fixed amount by which the two prediction functions will differ infinitely often. By Corollary 1 this is sufficient to prove that they are strongly distinguishable. Now the preconditions of Theorem 11 have been established, and the result follows from it. □

6 Acknowlegments

The authors would like to thank several people who have made important contributions to this work: Renato M. Capocelli, Larry Carter, Alan Cobham, Dexter Kozen, Jean Voldman, and Edwin Wegman.

References

[AnCa] Angluin, D., and Carl, H. S., *Inductive Inference: Theory and Methods*, Computing Surveys, vol. 15, no. 3, September 1983, 237–269.

[As] Ash, R. B., *Measure, Integration and Functional Analysis*, Academic Press, New York, 1972.

[BlBl] Blum, L., and Blum, M., *Toward a Mathematical Theory of Inductive Inference*, Inf. Control 28, 1975, 125–155.

[Ch] Chaitin, G. J., *Algorithmic Information Theory*, IBM J. Res. Develop., 21, July 1977, 350–359.

[FaHe] Faugeras, O. D., and Hebert, M., *A 3-D Recognition and Positioning Algorithm using Geometrical Matching between Primitive Surfaces*, in Proc. 8th Int. Joint Conf. Artificial Intell. (IJCAI-83), Karlsruhe, West Germany, Aug. 1983, 996–1002.

[FaHePa] Faugeras, O. D., Hebert, M., and Pauchor, E., *Segmentation of Range Data into Planar and Quadratic Patches*, in Proc. of Computer Vision & Pattern Recognition, 1983, 8–13.

[Go] Gold, E. M., *Language Identification in the Limit*, Inf. Control, 10, 447–474.

[Ko] Kolmogorov, A. N., *Three Approaches to the Quantitative Definition of Information Transmission*, Problems of Information Transmission, vol. 1, no. 1, 1965, 4–7.

[Hu] Hume, D., *On Human Nature and Understanding*, Macmillan Publishing Company, New York, 1962.

[JuRa] Juang, B. H., and Rabiner, L. R., *A Probabilistic Distance Measure for Hidden Markov Models*, AT&T Technical Journal, vol. 64, no. 2, February 1985, 391–408.

[La] Lamperti, J., *Probability*, W.A. Benjamin, New York, 1966.

[Mo] Moody, Ernest E., *The Logic of William of Occam*, London, 1935.

[R1] Rissanen, J., *Stochastic Complexity and Modeling*, Ann. of Statistics, 1986, 14, 1080-1100.

[R2] Rissanen, J., *Stochastic Complexity*, The Journal of Royal Statistical Society, Series B, No. 3, 1987, 225–265.

[Ru] Rudich, S., *Inferring the Structure of a Markov Chain from Its Output*, 28th Annual Symposium on Foundation of Computer Science, 1985, 321–326.

[So] Solomonoff, R. J., *A Formal Theory of Inductive Inference. Part 1 and 2*, Inf. Control, vol. 7, 1–22, 224–254, 1964.

[Va] Valiant, L. G., *A Theory of the Learnable*, C. ACM, 27, 1134–1142.

[ZiLe1] Ziv, J., and Lempel, A., *A Universal Algorithm for Sequential Data Compression*, IEEE Trans. Inform. Theory, IT-23, 1976, 75–81.

[ZiLe2] Ziv, J., and Lempel, A., *Compression of Individual Sequences via Variable-rate Coding*, IEEE Trans. Inform. Theory, IT-24, 1978, 530–536.