DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume **00**, 0000

Inductive Reasoning

MING LI AND PAUL VITÁNYI

Abstract. Our aim is to explain a general theory of inductive reasoning which is close enough to the concerns of language studies. In this set-up, the optimal prediction rate is assigned to the hypothesis considered most likely by a prior-free form of Bayesian inference. In terms of practical applications a most attractive form of this approach is embodied by the so-called minimum description length (MDL) principle. There, the most likely hypothesis is the one which minimizes the sum of the length of the description of the hypothesis and the length of the description of the data relative to the hypothesis. This theory is solidly based on a provably ideal method of inference using Kolmogorov complexity. We give references to several applications. Similar approaches should work for computational learning of features of language.

The genesis of this work is not rooted in traditional approaches to artificial intelligence (AI), but rather in new exciting general learning theories which have developed out from the computational complexity theory [**21, 20**], statistics and descriptional (Kolmogorov) complexity [**15**]. These new theories have received great attention in theoretical computer science and statistics, [**21, 20, 15, 17, 16, 18, 1, 9**]. One the other hand, the design of real learning systems seemed to be dominated by *ad hoc* trial-and-error methods.

It is commonly accepted that all learning involves compression of experimental data in a compact 'theory' , 'hypothesis', or 'model' of the phenomenon under investigation. In [**11, 12**] the authors analysed the theory of such approaches related to shortest effective description length (Kolmogorov complexity). The question arises whether these theoretical insights can be directly applied to real

1991 *Mathematics Subject Classification.* Primary 68S05, 68T05; Secondary 62C10, 62A99.

The first author was supported in part by ONR Grant N00014-85-K-0445 and ARO Grant DAAL03-86-K-0171 at Harvard University, by NSERC operating grant OGP-0036747 at York University, and by NSERC operating grant OGP-046506 at the University of Waterloo. The second author was supported in part by NWO through NFI Project ALADDIN, and by NSERC through International Scientific Exchange Award ISE0125663.

This paper contains material from our [**12, 13**]. The final version of this paper will be submitted for publication elsewhere.

world problems. To show that this can be done, the first author and Qiong Gao, see [**3, 4**], carried out an experiment in on-line learning to recognize isolated characters written in a particular person's handwriting.

Reasoning from 'experience' to 'truth' has been the subject of intricate theories scattered throughout vastly different areas such as philosophy of science, statistics and probability theory, computer science, and psychology. Kolmogorov complexity allows us to study many seemingly unrelated models or principles from a unified view point. These include, [**12**], the maximum likelihood principle, the maximum entropy principle, the minimum description length principle, induction by enumeration , and probably approximately correct (pac) learning. Each of these ideas has had a pronounced influence in its respective field: philosophy of science, statistics, artificial intelligence, and theory of computing.

The Oxford English Dictionary defines **induction** as the process of inferring a general law or principle from the observations of particular instances. This defines precisely what we would like to call *inductive inference* On the other hand, we regard *inductive reasoning* as a more general concept than inductive inference, namely, as a process of re-assigning a probability (or credibility) to a law or proposition from the observation of particular instances. In other words, inductive inference draws conclusions that *accept or reject* a proposition, possibly without total justification, while inductive reasoning only changes the degree of our belief in a proposition. We need also to distinguish inductive reasoning from *deductive reasoning (or inference).* In deductive reasoning one derives the absolute truth or falsehood of a proposition. This may be viewed as a borderline case of inductive reasoning. A celebrated principle for induction is commonly attributed to William of Ockham (1290?–1349?).

> **Occam's razor.** Entities should not be multiplied beyond necessity.

According to Bertrand Russell, the actual phrase used by William of Ockham was: "It is vain to do with more what can be done with fewer." This is generally interpreted as: Among the theories that are consistent with the observed phenomena, one should pick the simplest theory.

But is a simpler theory really better than a more complicated one? What is the proper measure of simplicity? Is $x^{100}+1$ more complicated than $13x^{17}+5x^3+7x+11$? In this context the contemporary philosopher Karl Popper pronounced that the razor is without sense since there is no objective criterion for simplicity. It is the aim of this paper to show that the principle can be given objective contents.

EXAMPLE 0.1. Let us consider a simple example of inferring a finite grammar with one-letter terminals using Occam's razor. Let us measure 'simplicity' by number of rules in the grammar. The sample data are:

> generated terminal strings: 0, 000, 00000, 000000000;
> not generated terminal strings: ϵ, 00, 0000, 000000;

For these data there exist many consistent finite grammars. Let S denote the starting nonterminal symbol of a grammar. A trivial consistent finite grammar is the first one below, while the second grammar is the smallest consistent one.

$$\begin{aligned} S &\rightarrow 0|000|00000|000000000, \\ S &\rightarrow 00S|0 \end{aligned}$$

Intuitively, the trivial grammar just plainly encodes the data. We therefore do not expect that the grammar anticipates future data. On the other hand, the small grammar makes the plausible *inference* that the language generated consists of strings of odd number of 0's. The latter appeals to our intuition as a reasonable inference.

In the learning grammar example, it turns out that one can prove the following. The celebrated 'Occam's Razor Theorem' in [**1**] states that if sufficient data are drawn randomly from any fixed distribution, then with high probability the smallest consistent grammar (or a 'reasonably' small grammar which compresses the observations far enough) will with high probability correctly predict acceptance or rejection of most data which are drawn afterwards from this distribution. See also [**13**].

In contrast to Ockham, Thomas Bayes took a probabilistic view of Nature. Assume we have observational data D.

> **Bayes' Rule.** The probability of hypothesis H being true is proportional to the learner's initial believe in H (the prior probability) multiplied by the conditional probability of D given H.

Bayesian reasoning is mathematically fine but it has a weakness: it assumes knowledge of the *prior probability.* For many practical problems it is unclear how the prior probability should be defined, how it can be found, or whether it exists at all. Take for example properties of the English language. It has been produced by different people from different times and social backgrounds. Can we claim that there is a definite probability involved among different competing hypotheses about a certain aspect of the language? The historic dispute between Bayesians and non-Bayesians is related to such problems.

Essentially combining the ideas of Ockham, Bayes, and modern computability theory, R.J. Solomonoff has successfully invented a 'perfect' induction theory. First, combine Occam's razor principle and modern computability theory to obtain Kolmogorov complexity. With Kolmogorov complexity define a *universal prior* which dominates, up to a multiplicative constant, all computable prior probability distributions. Use this universal prior in Bayes' Rule substituting it for *any* computable prior probability which may actually hold. This results a general theory of inductive inference.

The notion of 'simplicity' has dominated linguistic argumentation for much of its illustrious history (before Solomonoff was even conceived). For example, Grimm's and Verner's laws (about diachronic sound change) are based on simplicity arguments. Chomsky's masters thesis on the morphophonemics of Hebrew used simplicity as its central criterion, and Chomsky and Halle's pathbreaking (1968) "Sound Pattern of English" also invokes an explicit simplicity metric. Halle has written two excellent papers on this topic, [**6, 7**].

1. Bayesian Reasoning

On the one hand, it seems common sense to assume that people learn in the sense that they generalize from observations by learning a 'law' that governs not only the past observations, but will also apply to the observations in the future. In this sense induction should 'add knowledge'.

Yet how is it possible to acquire knowledge which is not yet present? If we have a system to deduce a general law from observations, then this law is only part of the knowledge contained in this system and the observations. Then, the law does not represent knowledge over and above what was already present, but it represents in fact only a part of that knowledge. This seeming contradiction is related to the distinction between 'implicit knowledge' and 'explicit (useful) knowledge'. We need to extract the latter from the former, and it may require time and or space to do so. If the resources required are forbiddingly large, then we cannot compute the useful knowledge from the implicit knowledge, even if we have all the information.

As an example consider a book on number theory. Given the axioms and inference rules of number theory, and the statements of the theorems in the book, we can in principle reconstruct all the proofs of the theorems by enumerating all valid proofs of the theory. However, finding the valid proofs is very hard (it took mankind 2000 years). Information which can only be reconstructed from a short description at the expense of great computational effort is called 'logically deep'. The theory of logical depth is due of G. Chaitin and C. Bennett, see for example [**13**]. It possibly gives some insight in the above paradox about the distinction between implicit knowledge and useable knowledge—for example, knowledge the user is aware of. This theory should be developed further, but it is out of the scope of this article.

The calculus of probabilities has come up with an induction principle which estimates the relative likelihood of different possible hypotheses.

Consider the situation in which one has a set of observations (say, sentences in some new language) D, and also a (possibly infinite) set of hypotheses (say, potential grammars): $H_1, H_2 \ldots$ For each hypothesis H_i we would like to assess the probability that H_i is the "correct" hypothesis (that is, the generating grammar), given the observation of D. This quantity, $P(H_i|D)$, can be described and manipulated formally in the following way.

DEFINITION 1.1. *Consider a discrete sample space Ω. Let $D, H_1, H_2, \ldots$ be a countable set of events (subsets) of Ω. The list $\boldsymbol{H} = \{H_1, H_2, \ldots\}$ is called the hypotheses space. The hypotheses H_i are exhaustive (at least one is true). From the definition of conditional probability, that is, $P(A|B) = P(A \cap B)/P(B)$, it is* *easy to derive* ***Bayes' formula*** *(rewrite $P(A \cap B)$ in two different ways):*

$$P(H_i|D) = \frac{P(D|H_i)P(H_i)}{P(D)}. \tag{1.1}$$

If the hypotheses are mutually exclusive ($H_i \cap H_j = \emptyset$ for all i, j), then

$$P(D) = \sum_i P(D|H_i)P(H_i).$$

Despite the fact that Bayes' rule is just a rewriting of the definition of conditional probability and nothing more, its interpretation and applications are most profound and caused much controversy during the past two centuries. In Equation 1.1, the H_i's represent the possible alternative hypotheses concerning the phenomenon we wish to discover. The term D represents the empirically or otherwise known data concerning this phenomenon. The term $P(D)$, the probability of data D, may be considered as a normalizing factor so that $\sum_i P(H_i|D) = 1$. The term $P(H_i)$ is called the *a priori* probability or *initial* probability of hypothesis H_i, that is, it is the probability of H_i being true before we see any data. The term $P(H_i|D)$ is called *a posteriori* or *inferred* probability

The most interesting term is the prior probability $P(H_i)$. In the context of machine learning, $P(H_i)$ is often considered as the learner's *initial degree of belief* in hypothesis H_i. In essence Bayes' rule is a mapping from *a priori* probability $P(H_i)$ to *a posteriori* probability $P(H_i|D)$ determined by data D. In general, the problem is not so much that in the limit the inferred hypothesis would not concentrate on the true hypothesis, but that the inferred probability gives as much information as possible about the possible hypotheses from only a limited number of data. The continuous debate between the Bayesian and non-Bayesian opinions centered on the prior probability. The controversy is caused by the fact that Bayesian theory does not say how to initially derive the prior probabilities for the hypotheses. Rather, Bayes' rule only tells how they are to be *updated.* In the real world problems, the prior proabilities may be unknown, uncomputable, or even conceivably non-existent. This problem would be solved if we can find a *single* probability distribution to use as the prior distribution in each different case, with approximately the same result as if we had used the real distribution. Surprisingly, this turns out to be possible up to some mild restrictions.

1.1. Kolmogorov Complexity. The Kolmogorov complexity, [**10, 22, 11**], of x is simply *the length of the shortest effective binary description of* x. Formally, this is defined as follows. Let $x, y, z \in \mathcal{N}$, where $\mathcal{N}$ denotes the natural numbers and we identify $\mathcal{N}$ and $\{0,1\}^*$ according to the correspondence $(0, \epsilon), (1, 0), (2, 1), (3, 00), (4, 01), \ldots$. Here ϵ denotes the *empty word* '' with no letters. The *length* $l(x)$ of x is the number of bits in the binary string x. For example, $l(010) = 3$ and $l(\epsilon) = 0$.

The emphasis is on binary sequences only for convenience; observations in any alphabet can be so encoded in a way that is 'theory neutral'.

A binary string x is a *proper prefix* of a binary string y if we can write $x = yz$ for $z \neq \epsilon$. A set $\{x, y, \dots\} \subseteq \{0,1\}^*$ is *prefix-free* if for any pair of distinct elements in the set neither is a proper prefix of the other. A prefix-free set is also called a *prefix code*. Each binary string $x = x_1 x_2 \dots x_n$ has a special type of prefix code, called a *self-delimiting code*,

$$\bar{x} = x_1 x_1 x_2 x_2 \dots x_n \neg x_n,$$

where $\neg x_n = 0$ if $x_n = 1$ and $\neg x_n = 1$ otherwise. This code is self-delimiting because we can determine where the code word $\bar{x}$ ends by reading it from left to right without backing up. Using this code we define the standard self-delimiting code for x to be $x' = \overline{l(x)}x$. It is easy to check that $l(\bar{x}) = 2n$ and $l(x') = n + 2\log n$.

Let $T_1, T_2, \dots$ be a standard enumeration of all Turing machines, and let $\phi_1, \phi_2, \dots$ be the enumeration of corresponding functions which are computed by the respective Turing machines. That is, T_i computes ϕ_i. These functions are the *partial recursive* functions or *computable* functions.

Let $\langle \cdot \rangle$ be a standard invertible effective one-one encoding from $\mathcal{N} \times \mathcal{N}$ to prefix-free recursive subset of $\mathcal{N}$. For example, we can set $\langle x, y \rangle = x'y'$. We insist on prefix-freeness and recursiveness because we want a universal Turing machine to be able to read an image under $\langle \cdot \rangle$ from left to right and determine where it ends.

DEFINITION 1.2. *The* prefix Kolmogorov complexity *of x given y (for free) is*

$$K(x|y) = \min_{p,i} \{ l(\langle p, i \rangle) : \phi_i(\langle p, y \rangle) = x, p \in \{0,1\}^*, i \in \mathcal{N} \}.$$

Define $K(x) = K(x|\epsilon)$.

A Turing machine T computes a function on the natural numbers. However, we can also consider the computation of real valued functions. For this purpose we consider both the argument of ϕ and the value of ϕ as a pair of natural numbers according to the standard pairing function $\langle \cdot \rangle$. We define a function from $\mathcal{N}$ to the reals $\mathcal{R}$ by a Turing machine T computing a function ϕ as follows. Interprete the computation $\phi(\langle x, t \rangle) = \langle p, q \rangle$ to mean that the quotient p/q is the rational valued tth approxmation of $f(x)$.

DEFINITION 1.3. *A function $f : \mathcal{N} \to \mathcal{R}$ is* enumerable *if there is a Turing machine T computing a total function ϕ such that $\phi(x, t+1) \geq \phi(x, t)$ and $\lim_{t \to \infty} \phi(x, t) = f(x)$. This means that f can be computably approximated from below. If f can also be computably approximated from above then we call f* recursive.

A function $P : \mathcal{N} \to [0,1]$ is a *probability distribution* if $\sum_{x \in \mathcal{N}} \leq 1$. (The inequality is a technical convenience. We can consider the surplus probability to be concentrated on the undefined element $u \notin \mathcal{N}$).

Consider the family $\mathcal{E}P$ of *enumerable* probability distributions on the sample space $\mathcal{N}$ (equivalently, $\{0,1\}^*$). It is known, [**13**], that $\mathcal{E}P$ contains an element $\mathbf{m}$ that multiplicatively dominates all elements of $\mathcal{E}P$. That is, for each $P \in \mathcal{E}P$ there is a constant c such that $c\,\mathbf{m}(x) > P(x)$ for all $x \in \mathcal{N}$.

The family $\mathcal{E}P$ contains all distributions with computable parameters which have a name, or in which we could conceivably be interested, or which have ever been considered. The dominating property means that $\mathbf{m}$ assigns at least as much probability to each object as any other distribution in the family $\mathcal{E}P$ does. In this sense it is a universal *a priori* by accounting for maximal ignorance. It turns out that if the true *a priori* distribution in Bayes Rule is recursive, then using the single distribution $\mathbf{m}$, or its continuous analogue the measure $\mathbf{M}$ on the sample space $\{0,1\}^\infty$ (defined later), is provably as good as using the true *a priori* distribution.

We also know, [**13**], that

$$-\log \mathbf{m}(x) = K(x) \pm O(1). \tag{1.2}$$

That means that $\mathbf{m}$ assigns high probability to simple objects and low probability to complex or random objects. For example, for $x = 00\ldots0$ (n 0's) we have $K(x) = K(n) + O(1) \leq \log n + 2\log\log n + O(1)$ since the program

```
print n_times a '0'
```

prints x. (The additional $2\log\log n$ term is the penalty term for a self-delimiting encoding.) Then, $1/(n\log^2 n) = O(\mathbf{m}(x))$. But if we flip a coin to obtain a string y of n bits, then with overwhelming probability $K(y) \geq n - O(1)$ (because y does not contain effective regularities which allow compression), and hence $\mathbf{m}(y) = O(1/2^n)$. We also know, [**13**], that

$$-\log \mathbf{M}(x) = K(x) \pm O(\log K(x)),$$

Again this means that $\mathbf{M}$ assigns high probability to simple objects and low probability to complex or random objects. For example, for $x = 00\ldots0$ (n 0's) we have $\mathbf{M}(x) \geq 1/(n\log^{O(1)} n)$. But if we flip a coin to obtain a y of n bits, then with overwhelming probability $\mathbf{M}(y) = O(1/2^n)$.

2. Solomonoff's Universal Prior

Consider theory formation in science as the process of obtaining a compact description of the past observations. The investigator observes increasingly larger initial segments of an infinite binary sequence X as the outcome of an infinite sequence of experiments on some aspect of nature. To describe the underlying regularity of X, the investigator tries to formulate a theory that governs X, consistent with past experiments. Candidate theories (hypotheses) are identified with computer programs that compute binary sequences starting with the observed initial segment.

First assume the existence of a probability distribution μ over the continuous sample space $\Omega = \{0,1\}^\infty$. Such a distribution is called a *measure* and is defined by the probabilities of elements belonging to certain subsets of Ω. Denote by $\mu(x)$ the probability of a sequence starting with x, that is, the probability that it is an element of the set of all sequences in Ω that start with x. For $\mu : \{0,1\}^* \rightarrow [0,1]$ to be a measure it must satisfy (i) $\mu(\epsilon) \leq 1$; and (ii) $\mu(x) \geq \mu(x0) + \mu(x1)$. (The inequalities are a technical convenience. We can obtain equalities by concentrating the surplus probabilities on the undefined element $u \notin \{0,1\}$: $\mu(\epsilon) + \mu(u) = 1$ and $\mu(x) = \mu(x0) + \mu(x1) + \mu(xu)$.)

The inference problem can now be formulated as follows. Given a previously observed data string S, predict the next symbol in the sequence, that is, extrapolate the sequence S. In terms of the variables in Equation 1.1, H_a is the hypothesis that the sequence under consideration continues with a. Data D_S consists of the fact that the the sequence starts with initial segment S. Thus, for $P(H_i)$ and $P(D)$ in Formula 1.1 we substitute $\mu(H_a)$ and $\mu(D_S)$, respectively, and obtain

$$\mu(H_a|D_S) = \frac{\mu(D_S|H_a)\mu(H_a)}{\mu(D_S)}.$$

We must have $\mu(D_S|H_a) = 1$ for any a. Hence,

$$\mu(H_a|D_S) = \frac{\mu(H_a)}{\mu(D_S)}. \tag{2.1}$$

Generally, we denote $\mu(H_a|D_S)$ by $\mu(a|S)$. In terms of inductive inference or machine learning, the final probability $\mu(a|S)$ is the probability of the next symbol being a, given the initial sequence S. Obviously we now only need the prior probability μ to evaluate $\mu(a|S)$. The idea is to approximate the unknown proper prior probability μ.

Similar to Definition 1.3 one can define enumerable measures, [**13**]. Just like in the case of probability distributions over a discrete sample space, all measures with computable parameters we may be conceivably interested in are enumerable. The family of enumerable measures is denoted by $\mathcal{E}M$. It can be proved, [**13**], that $\mathcal{E}M$ contains a *universal measure*, denoted by $\mathbf{M}$, such that for all μ in this class there exists a constant c such that $c\,\mathbf{M}(x) \geq \mu(x)$ for all x. We say that $\mathbf{M}$ *dominates* μ. We also call $\mathbf{M}$ the *a priori* probability, since it assigns maximal probability to all hypotheses in absence of any knowledge about them.

Now instead of using Formula 2.1, we estimate the conditional probability $\mu(y|x)$ that the next segment after x is y by the expression

$$\frac{\mathbf{M}(xy)}{\mathbf{M}(x)}. \tag{2.2}$$

Now let μ in Formula 2.1 be an arbitrary computable measure. This case includes all computable sequences. If the length of y is fixed, and the length of x grows to infinity, then it can be shown similar to [**19**], see [**13**], that

$$\frac{\mathbf{M}(y)/\mathbf{M}(x)}{\mu(y)/\mu(x)} \to 1,$$

with μ-probability one. In other words, the conditional *a priori* probability is almost always asymptotically equal to the conditional probability. It has also been shown by Solomonoff that the convergence is very fast and if we use Formula 2.2 instead of the real value Formula 2.1, then our inference is almost as good.

2.1. Rate of Convergence of Guessing Error. We can quantify how fast Solomonoff's predictions converge to the optimal predictions. Obviously, we cannot do better than predict according to μ. Let S_n be the expected squared error in the nth prediction (with $l(x)$ is the binary length of x):

$$S_n = \sum_{l(x)=n-1} \mu(x) \left(\mu(0|x) - \mathbf{M}(0|x)\right)^2 . \tag{2.3}$$

Since we consider only binary sequences, this figure of merit accounts for all error in the nth prediction. It was shown in [**19**], see also [**13**], that the summed expected error over all predictions is bounded by a constant,

$$\sum_{n=1}^{\infty} S_n < k, \tag{2.4}$$

where k is a constant depending only on μ. (It can be shown, [**13**], that $k = K(\mu)/\ln 2$, where $K(\mu)$ is the length of the shortest program computing μ in a prefix-free programming language, see above.) This means that, using $\mathbf{M}$, the expected prediction error S_n in the nth prediction goes to 0 faster than $1/n$. Used as the prior in Bayes Rule, this proves mathematically that the inferred probability using prior $\mathbf{M}$ converges very fast to the inferred probability using the actual prior μ. The problem with Bayes' Rule has always been the determination of the prior. Using $\mathbf{M}$ universally gets rid of that problem, and is provably perfect.

The point of using Solomonoff's prior is not that we eventually converge to the true hypothesis, but that we do it very fast and make small errors in predictions also in the initial segments. Note that for any prior distribution the inferred probability will *eventially* converge. This can be seen as follows. Suppose we have a bag of coins which are bent in different ways, and hence have different probabilities of coming up heads. Picking a coin from the bag we want to estimate the probability of flipping a head. Initially, before we have experimented with the coin, this probability will be totally determined by the relative frequencies of coins with different probabilities of coming up heads. These relative frequencies consitute the true prior probability distribution over the different hypotheses of the form "the coin has 0.x probability of coming up heads". Using this prior probability in Bayes' rule gives the best predictions. However, whatever prior probability we choose (provided it assigns positive probability to each hypothesis), in the long run of gathering experimental data by flipping the coin the inferred probability in Bayes rule will converge to probability 1 for the correct hypothesis and probability 0 for the incorrect hypotheses, by the law of large numbers. However, using the universal prior we converge almost as fast as possible.

We now come to the punch line: Bayes' rule using the universal prior distribution yields Occam's Razor principle. Namely, if several programs could generate $S0$ then the shortest one is used (for the prior probability), and further if $S0$ has a shorter program than $S1$ then $S0$ is preferred (that is, predict 0 with higher probability than predicting 1 after seeing S). Bayes' rule via the universal prior distribution also gives the so-called indifference principle in case $S0$ and $S1$ have roughly equal length shortest programs which 'explain' $S0$ and $S1$, respectively.

3. Recursion Theory Induction

There are many different ways of formulating concrete inductive inference problems in the real world. We abstract matters as much as possible short of losing significance, following E.M. Gold, [**5**].

We are given an effective enumeration of partial recursive functions $f_1, f_2, \ldots$. Such an enumeration can be the functions computed by Turing machines, but also the functions computed by finite automata. We want to infer a particular function f. To do so, we are presented with a sequence of examples $D = e_1, e_2, \ldots, e_n$, containing elements (possibly with repetitions) of the form

$$e = \begin{cases} (x, y, 0) & \text{if } f(x) \neq y, \\ (x, y, 1) & \text{if } f(x) = y. \end{cases}$$

For $n \to \infty$ we assume that D contains all elements of the displayed form.

3.1. Inference of Hypotheses. Let the different hypotheses H_k be '$f = f_k$'. Since $P(D|H_k)$ is 1 or 0 according to whether D is consistent with f_k or not, take any positive prior distribution $P(H_k)$, say $P(H_k) = 1/k(k+1)$, and apply Bayes' Rule 1.1, to obtain

$$P(H_k|D) = \frac{P(D|H_k)P(H_k)}{\sum\{P(H_j) : f_j \text{ is consistent with } D\}}. \tag{3.1}$$

With increasing n, the denominator term is monotonically nonincreasing. Since all examples eventually appear, the denominator converges to a limit.

For each k, the inferred probability of f_k is monotonically nondecreasing with increasing n, until f_k is inconsistent with a new example, in which case it falls to zero and stays there henceforth. Only the f_k's that are consistent with the sequence of presented examples have positive inferred probability. At each step we infer the f_k with the highest posterior probability. At some point the first copy of f in the sequence will have the highest probability, and will keep it henceforth. This is called *induction by enumeration.* The classical form is to eliminate all functions which are inconsistent with D from left to right in the enumeration of functions, up to the position of the first consistent function. We receive a new example e, set $D := D, e$, and repeat this process. Eventually, the new first function in the enumeration is a copy of f and it doesn't change any more. This deceptively simple idea has generated a large body of sophisticated literature.

> This way one learns more and more about the unknown target function, and approximates it until the correct identification has been achieved. "I wish to construct a precise model for the intuitive notion 'able to speak a language' in order to be able to investigate theoretically how it can be achieved artificially. Since we cannot write down the rules of English which we require one to know before we say he can 'speak English,' an artificial intelligence which is designed to speak English will have to learn its rules from implicit information. That is, its information will consist of examples of the use of English and/or of an informant who can state whether a given usage satisfies certain rules of English, but cannot state these rules explicitly. ... A person does not know when he is speaking a language correctly; there is always the possibility that he will find that his grammar contains an error. But we can guarantee that a child will eventually learn a natural language, even if it will not know when it is correct." [Gold]

How do we use the universal prior probability? Set $P(H_k) = \mathbf{m}(k)$, with $\mathbf{m}(\cdot)$ the universal discrete probability. We have seen, Equation 1.2, that

$$\mathbf{m}(x) = 2^{-K(x)+O(1)},$$

with $K(\cdot)$ the prefix complexity. With this prior, at each stage, $P(\cdot|D)$ will be largest for the simplest consistent hypothesis. In the limit, this will be the case for H_k such that $f_k = f$ and $K(k)$ is minimal. In many enumerations we will find the proper H_k much faster using $\mathbf{m}(\cdot)$ as prior than using $1/k(k+1)$. Sometimes even noncomputably much faster. But since $K(x)$ and hence $\mathbf{m}(x)$ is uncomputable, [**13**], one cannot find $\mathbf{m}$ and hence cannot us it in practice. Therefore, one can only use a computable approximation to $\mathbf{m}$. The function $1/k(k+1)$ is such a computable approximation (a rather trivially simple one).

3.2. Prediction. Suppose, we want to infer the correct value of $f(x)$ after having seen data D. We can refer to the analysis above and simply predict by

$$P(e|D) = \frac{\sum P(H_k|D,e)}{\sum P(H_k|D)}.$$

But let us use the universal measure $\mathbf{M}$, the continuous version of $\mathbf{m}$ on the sample space $\{0,1\}^\infty$ of one-way infinite binary sequences, [**13**]. For this analysis, replace the examples by binary self-delimiting codes: $e = (x, y, 1)$ by $\bar{e} = \bar{x}\bar{y}1$ and $e = (x, y, 0)$ by $\bar{e} = \bar{x}\bar{y}0$. This way the machine can see where the encoding of x ends without having to look at the next symbol. For convenience, we denote this binary encoding of D also by 'D'. Let $\mathcal{D}$ be the largest set of D's (possibly infinite) such that D is consistent with f_k. Now set

$$P(H_k) = \mathbf{M}\left(\omega : \omega \text{ starts with } D \in \mathcal{D}\right).$$

If we assume a recursive distribution μ on the examples, Solomonoff's maxim says we must predict according to

$$\mathbf{M}(e|D). \tag{3.2}$$

It can be proved, see [**13**], that the expected squared error S_n in the nth prediction, defined as in Equation 2.3 by

$$S_n = \sum_{l(D)=n-1} (\mathbf{M}(0|D) - \mu(0|D))^2,$$

satisfies Equation 2.4. Therefore, S_n goes to zero faster than $1/n$. We hasten to remark that this does not say much about the amount of mistakes in a particular single sequence.

3.3. Mistake Bounds. Consider an effective enumeration $f_1, f_2, \ldots$ of partial recursive functions with values in the set $\{0, 1\}$ only. Each such function f defines an infinite binary sequence $\omega = \omega_1\omega_2 \ldots$ by $\omega_i = f(i)$, for all i. This way, we have an enumeration of infinite sequences ω. These sequences form a binary tree with the root labeled ϵ and each ω is an infinite path starting from the root. We are trying to learn a particular function f, in the form that we predict ω_n from the initial sequence $\omega_1 \ldots \omega_{n-1}$ for all $n \geq 1$. We want to analyze the number of errors we make in this process. If our prediction is wrong (say, we predict a 0 and it should have been a 1), then this counts as 1 mistake.

LEMMA 3.1. *Assume the discussion above and we try to infer $f = f_n$. There is an algorithm which makes less than $2 \log n$ mistakes in all infinitely many predictions.*

PROOF. Define, for each f_i with associated infinite sequence ω^i, a measure μ_i by $\mu_i(\omega^i) = 1$. This implies that also $\mu_i(\omega_1^i \ldots \omega_n^i) = 1$ for all n. Let μ be a semimeasure defined by

$$\mu(x) = \sum_i \frac{1}{i(i+1)} \mu_i(x),$$

for each $x \in \{0, 1\}^*$. (Note that μ is a simple computable approximation to $\mathbf{M}$.) The prediction algorithm is very simple.

If $\mu(0|x) \geq 1/2$, then predict 0, otherwise predict 1.

Suppose that the target $f = f_n$. If there are k mistakes, then the conditional in the algorithm shows that $2^{-k} > \mu(\omega^n)$. (The combined probability of the mistakes is largest if they are concentrated in the first predictions.) By the definition of μ we have $\mu(\omega^n) \geq 1/(n(n+1))$. Together this shows $k < 2\log n$. $\square$

EXAMPLE 3.1. If, in the proof, we put weight $2^{-K(n)}$ on μ_n (instead of $1/(n(n+1)))$, then the number of mistakes is at most $k < K(n)$. Recall that always $K(n) \leq \log n + 2\log\log n$. But for regular n (say, $n = 2^k$) the value $K(n)$ drops to less that $(1+\epsilon)\log\log n$, for all $\epsilon > 0$. Of course, the prediction algorithm becomes noneffective because we cannot compute these weights ($K(\cdot)$ is uncomputable).

LEMMA 3.2. *If the target function is f and we make k errors in the first m predictions, then* $\log\binom{m}{k} + K(m) + O(1) \geq K(f(1)\dots f(m))$.

PROOF. Let A be a prediction algorithm. If k is the number of errors, then we can represent the mistakes by the index j in the ensemble of k mistakes out of m, where

$$j \leq \binom{m}{k}.$$

If we are given A, m, and j, we can reconstruct $f(1)\dots f(m)$. Therefore, $K(A,m,j) \geq K(f(1)\dots f(m))$. Since $K(A) = O(1)$, the lemma is proven. $\square$

EXAMPLE 3.2. Denote $x = f(1)\dots f(m)$. Write $\log\binom{m}{k} + K(m) + O(1)$ as

$$k\log\frac{m}{k} + n\left(1-\frac{k}{m}\right)\log\frac{1}{1-k/n} + O(\log m).$$

- If k/m is small, then this expression is about $k(\log(m/k)+1)+O(\log m)$. This gives approximately
$$k \approx \frac{K(x)}{\log(n/K(x))}.$$
For instance, with $K(x) = \sqrt{m}$ we find $k > 2\sqrt{m}/\log m$.
- If k/m is large, then this expression approximates $mH(k/m)$ (the entropy of a $(k/m, 1-k/m)$ Bernoulli process). For instance, if $k/n = 1/3$, then $nH(1/3) \geq K(x)$.
- Another approximation for k/n small shows $k \geq nH^{-1}(K(x)/n)$. For instance, if $K(x) = m$, then $k \geq n/2$.

3.4. Certification. The following theorem sets limits on the number of examples needed to effectively infer a particular function f. In fact, it does more. It sets a limit to the number of examples we need to *describe* or *certify* a particular function f in any effective way. Let $D = e_1e_2\dots e_n$ be a sequence of examples $e_i = (x_i, y_i, b_i)$ and let $x = x_1x_2\dots x_n$, $y = y_1y_2\dots y_n$, and $b = b_1b_2\dots b_n$. The statement of the lemma must cope with pathological cases such as that x simply spells out f in some programming language.

THEOREM 3.1. *Assume the notation above. Let c be an appropriate constant. If $K(f|x,y) > K(b|x,y) - c$, then we cannot effectively find f.*

PROOF. Otherwise we would be able to compute f, given x, from a program of length significantly shorter than $K(f|x)$, which leads to a contradiction as follows. In [**13**] it is shown that complexity is subadditive: $K(f,b) \leq K(b) + K(f|b) + O(1)$. With extra conditional x, y in all terms,

$$K(f,b|x,y) \leq K(b|x,y) + K(f|b,x,y) + O(1). \tag{3.3}$$

We have assumed that there is an algorithm A which, given D, returns f. That is, describing A in $K(A) = O(1)$ bits, we obtain

$$K(f|x,y,b) = K(f|D) \leq K(A) + O(1) = O(1).$$

Substituting this in Equation 3.3, we obtain $K(f,b|x,y) \leq K(b|x,y) + O(1)$. Since, trivially, $K(f,b|x,y) = K(f|x,y) + O(1)$ the proof is finished. □

4. Minimum Description Length

We can formulate scientific theories in two steps. First, we formulate a set of possible alternative hypotheses, based on scientific observations or other data. Second, we select one hypothesis as the most likely one. Statistics is the mathematics of how to do this. A relatively recent method in statistics was developed by J. Rissanen. The method can be viewed as a computable approximation to the noncomputable approach involving $\mathbf{m}$ or $\mathbf{M}$ and was inspired by it.

> **Minimum description length (MDL) principle**. The best theory to explain a set of data is the one which minimizes the sum of
>
> - the length, in bits, of the description of the theory; and
> - the length, in bits, of data when encoded with the help of the theory.

To be able to compute this minimum we need to severely restrict the allowable descriptions. The minimum description length is also called the *stochastic complexity* of the given data.

> With a more complex description of the hypothesis H, it may fit the data better and therefore decreases the misclassified data. If H describes all the data, then it does not allow for measuring errors. A simpler description of H may be penalized by increasing the misclassified data. If H is a trivial hypothesis that contains nothing, then all data are described literally and there is no generalization. The rationale of the method is that a balance in between seems required.

Let us see how we can derive the MDL principle. Recall Bayes' Rule

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}.$$

Here H is an hypothesis, and D is the set of observed data. We must find the hypothesis H such that $P(H|D)$ is maximized. Taking the negative logarithm of both sides of the formula, we obtain

$$-\log P(H|D) = -\log P(D|H) - \log P(H) + \log P(D). \tag{4.1}$$

We can assume $P(D)$ is fixed, since the data D is fixed. This term can be considered as a normalizing factor and is ignored in the following discussion. We are only concerned with maximizing the term $P(H|D)$ or, equivalently, *minimizing* the term $-\log P(H|D)$. This is equivalent to minimizing

$$-\log P(D|H) - \log P(H). \tag{4.2}$$

Let us assume that H and D are expressed as natural numbers or finite binary strings. If P is recursive, then $\kappa_0(H|P) = \log(\mathbf{m}(H)/P(H))$ is a universal sum P-test as defined in [**13**]. Since we are dealing with finite objects, we cannot sharply divide the objects in random ones and nonrandom ones. For finite objects, randomness is necessarily a matter of degree. Namely, it would be absurd if x is random and x with the first nonzero bit set to 0 is nonrandom.

However, for each constant c, we can define $c-P$-random H as those H such that $\kappa_0(H|P) \leq c$. Fix a small constant c, and call the $c-P$-random objects simply P-random. For suitably chosen c, the overwhelming majority of H's is P-random because

$$\sum_H P(H) 2^{\kappa_0(H|P)} \leq 1.$$

The analogous statement holds for $P(D|H)$. Hence, for P-random H and D, we can set

$$\begin{aligned} \log P(H) &= \log \mathbf{m}(H) + O(1), \\ \log P(D|H) &= \log \mathbf{m}(D|H) + O(1). \end{aligned}$$

According to Equation 1.2 (proof in [**13**]),

$$\begin{aligned} \log \mathbf{m}(H) &= -K(H) \pm O(1), \\ \log \mathbf{m}(D|H) &= -K(D|H) \pm O(1), \end{aligned}$$

where $K(\cdot)$ is the prefix complexity. That is, in order to maximize $P(H|D)$ over all hypotheses H, with high P-probability we need to minimize the sum of the minimum lengths of effective self-delimiting programs which compute descriptions of H and $D|H$. Such self-delimiting programs (prefix codes) are achieved by constructive versions of the Shannon-Fano code.

The term $-\log P(D|H)$ is also known as the *self-information* in information theory and the negative log-likelihood in statistics. It can now be regarded as the number of bits it takes to redescribe or encode D with an ideal code relative to H.

If we replace all P-probabilities in Equation 4.1 by the corresponding **m**-probabilities, we obtain in the same way

$$K(H|D) = K(H) + K(D|H) - K(D) + O(1).$$

In [**2**] (see [**13**]) it is proved that

$$K(H) + K(D|H, K(H)) = K(D) + K(H|D, K(D)) + O(1).$$

Since the self-delimiting description of $K(H)$ takes at most $2 \log K(H)$ bits, we have $K(H|D) = K(H,D) - K(D)$ up to a $O(\log K(H))$ additive term. The term $K(D)$ is fixed and doesn't change for different H's. Minimizing the left-hand term $K(H|D)$ can then be interpreted as

> **Alternative formulation MDL principle.** 'Given an hypothesis space $\mathbf{H}$, we want to select the hypothesis H such that the length of the shortest encoding of D together with hypothesis H is minimal'.

In different applications, the hypothesis H can be about different things. For example, decision trees, finite automata, grammars, Boolean formulas, or polynomials. Unfortunately, the function K is not computable, [**13**]. For practical applications (such as in statistics or natural language phenomena), one must settle for easily computable approximations. One way to do this is as follows. First encode both H and $D|H$ by a simply computable bijection as a natural number in $\mathcal{N}$. Assume we have some standard procedure to do this.

Now we make use of a basic property of prefix codes known as the *Kraft Inequality* (see for example any textbook on information theory or [**13**]). Let $I = \{l_1, l_2, \dots\}$ be a set of positive integers such that

$$\sum_{l \in I} 2^{-l} \leq 1. \tag{4.3}$$

Then there exists a prefix code $\{x_1, x_2, \dots\}$ with $l(x_i) = l_i$ for all i. Conversely, if $\{x_1, x_2, \dots\}$ is a prefix code, then its length set satisfies the above inequality.

We consider a simple self-delimiting description of x. For example, let x is encoded by x' as above. This makes $l(x') = \log x + 2 \log \log x$, which is a simple upper approximation of $K(x)$. Since the length of code word sets of prefix codes corresponds with a probability distribution by Kraft's Inequality 4.3, this encoding corresponds with assigning probability $2^{-l(x')}$ to x. In the MDL approach, this is the specific usable approximation to the universal prior. In the literature we find a more precise approximation which, however, has no practical meaning. For convenience, we smooth our encoding as follows.

DEFINITION 4.1. *Let $x \in \mathcal{N}$. The* universal MDL prior *over the natural numbers is $M(x) = 2^{-\log x - 2 \log \log x}$.*

H. Jeffreys has suggested to assign probability $1/x$ to each integer x. But this results in an improper distribution since the harmonic series $\sum 1/x$ diverges.
In the Bayesian interpretation the prior distribution expresses one's prior knowledge about the 'true' value of the parameter. This interpretation may be questionable, since the used prior is usually not generated by repeated random experiments. In Rissanen's view, the parameter is generated by the selection of the class of hypotheses and it has no inherent meaning. It is just one means to describe the properties of the data. The selection of H which minimizes $K(H) + K(D|H)$ (or Rissanen's approximation thereof) allows one to make statements about the data. Since the complexity of the models plays an important part, the parameters must be encoded. To do so, we truncate them to a finite precision and encode them with the prefix code above. Such a code happens to be equivalent to a distribution on the parameters. This may be called the universal MDL prior, but its genesis shows that it expresses no prior knowledge about the true value of the parameter. See [J. Rissanen, *Stochastic Complexity and Statistical Inquiry*, World Scientific, 1989]. Above we have given a validation of MDL from Bayes' Rule, which holds irrespective of the assumed prior, provided it is recursive and the hypotheses and data are random.

EXAMPLE 4.1. In statistical applications, H is some statistical distribution (or model) $H = P(\theta)$ with a list of parameters $\theta = (\theta_1, \ldots, \theta_k)$, where the number k may vary and influence the (descriptional) complexity of θ. (For example, H can be a normal distribution $N(\mu, \sigma)$ described by $\theta = (\mu, \sigma)$.) Each parameter θ_i is truncated to finite precision and encoded with the prefix code above. Under certain general conditions, J. Rissanen has shown that with k parameters and n data (for large n) Equation 4.2 is minimized for hypotheses H with θ encoded by $(k/2) \log n$ bits. This is called the *optimum model cost* since it represents the cost of the hypothesis description at the minimum description length of the total.

As an example, consider a Bernoulli process $(p, 1-p)$ with p close to $1/2$. For such processes $k = 1$. Let the outcome be $x = x_1 x_2 \ldots x_n$. Set $f_x = \sum_{i=1}^n x_i$. For outcome x with $C(x) \geq n - \delta(n)$, the number of 1's can be estimated ([**13**])

$$f_x = n/2 \pm \sqrt{(\delta(n) + c) n \ln 2}.$$

With $\delta(n) = \log n$, the fraction of such x's in $\{0,1\}^n$ goes to 1 with n rises unboundedly. Hence, for the overwhelming number of x's the frequency of 1's will be within

$$2^{-(1/2) \log n - O(R)}, \text{ with } O(R) \ll \log n,$$

of the value $1/2$. That is, to express an estimate to parameter p it suffices to use this precision. This requires at most $(1/2)\log n + O(R)$ bits. It is easy to generalize this example to arbitrary p.

EXAMPLE 4.2. In biological modeling, we often wish to fit a polynomial f of unknown degree to a set of data points

$$D = (x_1, y_1), \dots, (x_n, y_n),$$

such that it can predict future data y given x. Even if the data did come from a polynomial curve of degree, say two, because of measurement errors and noise, we still cannot find a polynomial of degree two fitting all n points exactly. In general, the higher the degree of fitting polynomial, the greater the precision of the fit. For n data points, a polynomial of degree $n-1$ can be made to fit exactly, but probably has no predicting value. The possible hypotheses are $(f, \mathbf{x})$, where f is a polynomial of degree at most $n-1$, and $\mathbf{x} = (x_1, \dots, x_n)$. The vector $\mathbf{x}$ is a standard fixed part of each hypothesis.

Let us apply the MDL principle where we describe all $k-1$-degree polynomials by a vector of k entries, each entry with a precision of d bits. Then, the entire polynomial is described by

$$kd + O(\log kd) \text{ bits.} \tag{4.4}$$

(Remember that we have to describe k, d, and account for self-delimiting encoding of the separate items.) For example, $ax^2 + bx + c$ is described by (a, b, c) and can be encoded by about $3d$ bits.

Consider polynomials f of degree at most $n-1$ which minimize the error

$$\text{error}(f) = \sum_{i=1}^{n} (f(x_i) - y_i)^2. \tag{4.5}$$

This way we find an optimal set of polynomials for each $k = 1, 2, \dots, n$. To apply the MDL principle we must trade the cost of hypothesis H (Equation 4.4) against the cost of describing $D|H$.

To describe measuring errors (noise) in data it is common practice to use the normal distribution. In our case this means that each y_i is the outcome of an independent random variable distributed according to the normal distribution with mean $f(x)$ and variance, say, constant. For each of them we have that the probability of obtaining a measurement y_i, given that $f(x)$ is the true value, that is of the order of $\exp(-(f(x) - y_i)^2)$. Considering this as a value of the universal MDL probability, this is encoded in $s(f(x) - y_i)^2$ bits, where s is a (computable) scaling constant. For all experiments together we find that the total encoding of $D|f, \mathbf{x}$ takes $s \cdot \text{error}(f)$ bits. The MDL principle thus tells us to choose a k-degree function f_k, $k \in \{0, \dots, n-1\}$, which minimizes (ignoring the vanishing $O(\log kd)$ term)

$$kd + s \cdot \text{error}(f_k).$$

EXAMPLE 4.3 (MAXIMUM LIKELIHOOD). The *maximum likelihood* (ML) principle says that for given data D, one should select the hypothesis H that maximizes $P(D|H)$ or, equivalently, minimizes $-\log P(D|H)$. In case of finitely many hypotheses, this is a special case of the MDL principle with the hypotheses distributed uniformly (all have equal probability). The principle has many admirers, is supposedly objective, and is due to R.A. Fisher.

EXAMPLE 4.4 (MAXIMUM ENTROPY). In statistics there are a number of important applications where the ML principle fails, but where the maximum entropy principle has been successful, and conversely.

In order to apply Bayes' Rule, we need to decide what the prior probabilities $p_i = P(H_i)$ are, subject to the constraint $\sum_i p_i = 1$ and certain other constraints provided by empirical data or considerations of symmetry, probabilistic laws, and so on. Usually these constraints are not sufficient to determine the p_i's uniquely. E.T. Jaynes proposed to select the prior by the *maximum entropy (ME) principle.*

The ME principle selects the estimated values $\hat{p}_i$ which maximize the entropy function

$$H(p_1, \ldots, p_k) = -\sum_{i=1}^{k} p_i \ln p_i, \tag{4.6}$$

subject to

$$\sum_{i=1}^{k} p_i = 1 \tag{4.7}$$

and some other constraints. For example, consider a loaded die, $k = 6$. If we do not have any information about the die, then using the principle of indifference, we may assume that $p_i = 1/6$ for $i = 1, \ldots, 6$. This actually coincides with the ME principle, since $H(p_1, \ldots, p_6) = -\sum_{i=1}^{6} p_i \ln p_i$, constrained by Equation 4.7, achieves its maximum $\ln 6 = 1.7917595$ for $p_i = 1/6$ for all i.

Now suppose it has been experimentally observed that the die is biased and the average throw gives 4.5, that is,

$$\sum_{i=1}^{6} i p_i = 4.5. \tag{4.8}$$

Maximizing the expression in Equation 4.6, subject to the constraints in Equations 4.7 and 4.8 gives the estimates

$$\hat{p}_i = e^{-\lambda i} \left(\sum_j e^{-\lambda j}\right)^{-1}, \quad i = 1, \ldots, 6,$$

where $\lambda = -0.37105$. Hence,

$$(\hat{p}_1, \ldots, \hat{p}_6) = (0.0543, 0.0788, 0, 1142, 0.1654, 0.2398, 0.3475).$$

The maximized entropy $H(\hat{p}_1, \ldots, \hat{p}_6)$ equals 1.61358. How dependable is the ME principle? Jaynes has proven an 'entropy concentration theorem' which, for example, implies the following. In an experiment of $n = 1000$ trials, 99.99% of all 6^{1000} possible outcomes satisfying the constraints of Equations 4.8 and 4.7 have entropy

$$1.602 \leq H\left(\frac{n_1}{n}, \ldots, \frac{n_6}{n}\right) \leq 1.614,$$

where n_i is the number of times the value i occurs in the experiment. We show that the Maximum Entropy principle can be considered as a special case of the MDL principle, as follows.

Consider the same type of problem. Let $\theta = (p_1, \ldots, p_k)$ be the prior probability distribution of a random variable. We perform a sequence of n independent trials. Shannon has observed that the real substance of Formula 4.6 is that we need approximately $nH(\theta)$ bits to record the sequence of n outcomes. Namely, it suffices to state that each outcome appeared $n_1, \ldots, n_k$ times, respectively, and afterwards give the index of which one of the

$$\binom{n}{n_1, \ldots, n_k} = \frac{n!}{n_1! \cdots n_k!} \tag{4.9}$$

possible sequences D of n outcomes actually took place. For this no more than

$$k \log n + \log \binom{n}{n_1, \ldots, n_k} + O(\log \log n) \tag{4.10}$$

bits are needed. The first term corresponds to $-\log P(\theta)$, the second term corresponds to $-\log P(D|\theta)$, and the third term represents the cost of encoding separators between the individual items. Using Stirling's approximation of $n! \sim \sqrt{2\pi n}(n/e)^n$ for the quantity of Equation 4.9, we find that, for large n, Equation 4.10 is approximately

$$n\left(-\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}\right) = nH\left(\frac{n_1}{n}, \ldots, \frac{n_k}{n}\right).$$

Since k and n are fixed, the least upper bound on the minimum description length, for an arbitrary sequence of n outcomes under certain given constraints 4.7 and 4.8, is found by maximizing the term in Equation 4.9 subject to said constraints. This is equivalent to maximizing the entropy function 4.6 under the constraints.

Viewed differently, let S_θ be the set of outcomes with values $(n_1, \ldots, n_k)$, with $n_i = np_i$, corresponding to a distribution $\theta = (p_1, \ldots, p_k)$. Then due to the small number of values (k) in θ relative to the size of the sets, we have

$$\log \sum_{\theta} d(S_\theta) \approx \max_{\theta}\{\log d(S_\theta)\}. \tag{4.11}$$

The left-hand side of Equation 4.11 is is the minimum description; the right-hand side of Equation 4.11 is the maximum entropy.

5. Pointers to Applications of MDL

This approach has been applied to real world learning system design. Some first applications were from learning decision trees [**14**] and in the design of an on-line hand-written character learning system, [**3**]. Relations between pac learning and MDL are explored in [K. Yamanishi, *Machine Learning*, 9(1993), 165-203]. The application of the MDL principle to fitting polynomials, as in Example 4.2, was originally considered by J. Rissanen in [*Ann. Stat.*, 14(1986), 1080-1100] and ['Stochastic complexity and the maximum entropy principle', unpublished]. Decision tree algorithms using MDL principle were also developed by Rissanen and Wax [personal communication with M. Wax, 1988]. Applications of MDL principle to learning on-line handwritten characters can be found in [Q. Gao and M. Li, *11th IJCAI*, 1989, pp. 843-848]; to surface reconstruction problems n computer vision [E.P.D. Pednault *11th IJCAI*, 1989, pp. 1603-1609]; and to protein structure analysis in [H. Mamitsuka and K. Yamanishi, *Proc. 26th Hawaii Int. Conf. Syst. Sciences*, 1993, pp. 659-668]. Applications of the MDL principle range from evolutionary tree reconstruction [P. Cheeseman and R. Kanefsky, Working Notes, *AAAI Spring Symposium Series*, Stanford University, 1990]; inference over DNA sequences [L. Allison, C.S. Wallace, and C.N. Yee, *Int. Symp. Artificial Intelligence and Math.*, January 1990; pattern recognition; smoothing of planar curves [S. Itoh, *IEEE ISIT*, January 1990]; to neural network computing [A.R. Barron, *Nonparametric Functional Estimation and Related Topics*, G. Roussas, Ed., Kluwer, 1991, pp. 561-576]. See also [A. R. Barron and T. M. Cover, *IEEE Trans. Inform. Theory*, IT-37 (1991), 1034-1054 (Correction Sept. 1991)].

Acknowledgement

We thank Les Valiant for many discussions on machine learning and the referees for their thoughtful reviews.

References

1. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. *Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension*, J. Assoc. Comput. Mach., 35:929–965, 1989.
2. P. Gács, *On the symmetry of algorithmic information*, Soviet Math. Dokl., 15: 1477-1480, 1974; Correction, Ibid., 15:1480, 1974.
3. Q. Gao and M. Li. *An application of minimum description length principle to online recognition of handprinted alphanumerals*, In 11th International Joint Conference on Artificial Intelligence, pages 843–848. Morgan Kaufmann Publishers, 1989.
4. Q. Gao, M. Li and P.M.B. Vitányi. *Learning On-Line Handwritten Characters*, In The Minimum Description Length Criterion, (W. Niblack, Ed.), to appear.
5. E.M. Gold, *Language identification in the limit*, Inform. Contr., **10**(1967), 447-474.
6. M. Halle, *On the role of simplicity in linguistic descriptions*, In Proceedings of Symposia in Applied Mathematics, 1961, 89–94 volume XII, Structure of Language and its Mathematical Aspects.
7. M. Halle, *Phonology in generative grammar*, Word, 1962, 18(1–2), 54–72.

8. H. Jeffreys. *Theory of Probability.* Oxford at the Clarendon Press, Oxford, 1961. Third Edition.
9. M. Kearns, M. Li, L. Pitt, and L. Valiant, *On the learnability of boolean formulae*, In Proc. 19th ACM Symp. Theory of Computing, pages 285–295, 1987.
10. A.N. Kolmogorov, *Three approaches to the quantitative definition of information*, Problems Inform. Transmission, 1(1):1–7, 1965.
11. M. Li and P.M.B. Vitányi, *Kolmogorov complexity and its applications*, In J. van Leeuwen, editor, Handbook of Theoretical Computer Science, chapter 4, pages 187–254. Elsevier and MIT Press, 1990.
12. M. Li and P.M.B. Vitányi, *Inductive reasoning and Kolmogorov complexity*, J. Comput. Syst. Sci., 44:343–384, 1992.
13. M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, 1993.
14. J. Quinlan and R. Rivest, *Inferring decision trees using the minimum description length principle*, Inform. Computation, 80:227–248, 1989.
15. J. Rissanen, *Modeling by the shortest data description*, Automatica-J.IFAC, 14:465–471, 1978.
16. J. Rissanen, *Universal coding, information, prediction and estimation*, IEEE Transactions on Information Theory, IT-30:629–636, 1984.
17. J. Rissanen, *Minimum description length principle*, In S. Kotz and N.L. Johnson, editors, Encyclopaedia of Statistical Sciences, Vol. V, pages 523–527. Wiley, New York, 1986.
18. J. Rissanen, *Stochastic complexity*, J. Royal Stat. Soc., series B, 49:223–239, 1987. *Discussion*: pages 252-265.
19. R.J. Solomonoff, *Complexity-based induction systems: comparisons and convergence theorems*, IEEE Trans. Inform. Theory, IT-24:422–432, 1978.
20. L.G. Valiant, *Deductive learning*, Phil. Trans. Royal Soc. Lond., A 312:441–446, 1984.
21. L.G. Valiant, *A theory of the learnable*, Comm. Assoc. Comput. Mach, 27:1134–1142, 1984.
22. A.K. Zvonkin and L.A. Levin, *The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms*, Russian Math. Surveys, 25(6):83–124, 1970.

Computer Science Department, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
E-mail address: mli@math.uwaterloo.ca

CWI and Universiteit van Amsterdam, Amsterdam, The Netherlands
Current address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
E-mail address: paulv@cwi.nl