Theories of Learning

Ming Li*
University of Waterloo

Paul M.B. Vitányi†
CWI

Abstract

Can computers learn? Recent research on learning theory suggests that in an appropriate sense the answer to this question is affirmative. We take a peek at some learning theories our predecesors and ourselves have developed in the history of science.[1]

1 Principle of Simplicity

We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances. [Isaac Newton, *Principia*, 1686]

The Oxford English Dictionary defines **induction** as "the process of inferring a general law or principle from the observations of particular instances". This is what we call "learning" in this lecture. In AI one speaks about learning by examples, learning from observation, and, perhaps, learning by analogy. There are other kinds of learning paradigms such as rote learning, learning by being told, and so on [?]. It might be debatable whether the latter methods of acquiring knowledge should be studied as learning or as programming. In this talk, we are interested only in inductive learning, that is, to infer *more* than what the learner was told.

It is useful to realize that learning from examples is precisely what statistics is about. One draws a sample from a population, and tries to induce a general law (distribution) which with some high probability is the correct one. Therefore, it is not surprising that probabilistic and statistical methods are perhaps indispensable in *all* nontrivial theories of learning. This is certainly the case in the development we sketch below.

The history of inductive inference, which is as old as empirical science itself, dates back at least to the Greek philosopher of science Epicurus (342?–270?BC). As human history evolves, man tries to understand and to explain the events that happen around him. This takes the form of different induction methods to formulate scientific (or other) theories from positive and negative experiences. Epicurus' formulated the following maxim.

> **Principle of Multiple Explanations.** If more than one theory is consistent with the data, keep them all.

Epicurus, in his *Letter to Pythocles*, explains this principle as follows. There are cases, especially of events in the heavens such as the risings and settings of heavenly bodies and eclipses, where it is sufficient for our happiness that several explanations be discovered. In these cases, the events "have multiple causes of coming into being and a multiple predication of what exists, in agreement with the perceptions."

He maintains that when several explanations are in agreement with the (heavenly) phenomena, we must keep all of them for two reasons. Firstly, the degree of precision achieved by multiple explanations is sufficient for human happiness. Secondly, it would be unscientific to prefer one explanation to another when both are equally in agreement with the phenomena. This, he claims, would be to "abandon physical inquiry and resort to myth." His follower Lucretius (95–55BC) considered multiple explanations as a stage in scientific progress. According to him, to select one explanation from several equally good ones is not appropriate for the person who would 'proceed step by step [cautiously].'

> "There are also some things for which it is not enough to state a single cause, but several, of which one, however, is the case. Just as if you

*Supported in part by NSERC operating grants OGP-0036747 and OGP-046506. Address: Computer Science Department, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. Email: mli@math.uwaterloo.ca

†Partially supported by NSERC International Scientific Exchange Award ISE0046203, and by NWO through NFI Project ALADDIN under Contract number NF 62-376. Address: Centrum voor Wiskunde en Informatica, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: paulv@cwi.nl

[1] Missing details, proofs, and references can be found in Chapter 5 of [7].

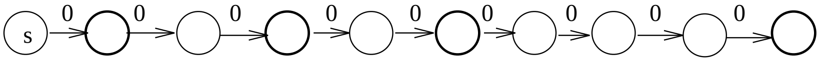

Figure 1: A trivial consistent automaton

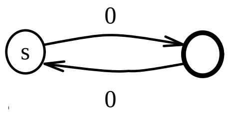

Figure 2: A smallest consistent automaton

were to see the lifeless corpse of a man lying far away, it would be fitting to state all the causes of death in order that the single cause of this death may be stated. For you would not be able to establish conclusively that he died by the sword or of cold or of illness or perhaps by poison, but we know that there is something of this kind that happened to him." [Lucretius]

The second and more sophisticated principle is the celebrated Occam's razor principle commonly attributed to William of Ockham (1290?–1349?). This was formulated about fifteen hundred years after Epicurus. In sharp contrast to the principle of multiple explanations, it states:

> **Occam's razor Principle.** 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 all consistent with the observed phenomena, one should pick the simplest theory'.

EXAMPLE 1.1 Consider the problem of learning a deterministic finite state automaton. Let us measure 'simplicity' by the number of states in the automaton. The sample data are:

Accepted inputs: 0, 000, 00000, 000000000;

Rejected inputs: ϵ, 00, 0000, 000000;

There are infinitely many finite automata which are consistent with these data. Figure 1 shows the trivial automaton, which simply encodes the data. Figure 2 shows the simplest automaton.

Since the automaton in Figure 1 just plainly encodes the data, we do not expect that the machine anticipates future data. On the other hand, the automaton in Figure 2 makes the plausible *inference* that the language accepted consists of strings of odd number of 0's. It selects the 'simplest' described division of the positive and negative data. The latter appeals to our intuition as a reasonable inference. In fact, pac learning in Section 4 justifies this approach.

> A too simplistic application of Occam's razor may also lead to nonsense as the following story illustrates (by N.R. Hanson). Once upon a time, there was a little girl named Emma. Emma had never eaten a banana, nor had she ever been on a train. One day she had to journey from New York to Pittsburgh by train. To relieve Emma's anxiety, her mother gave her a large bag of bananas. At Emma's first bite of her banana, the train plunged into a tunnel. At the second bite, the train broke into daylight again. At the third bite, Lo! into a tunnel; the fourth bite, La! into daylight again. And so on all the way to Pittsburgh. Emma, being a bright little girl, told her grandpa at the station: "Every odd bite of a banana makes you blind; every even bite puts things right again."

The history of science also contains a many examples where the principle of parsimony is used to find the proper theory. In fact, Isaac Newton formulated this as Rule 1 for doing science (as quoted in the heading of this paper). In a sense, science often proceeds in a way that the theories become more and more baroque, and then are reduced to some simple form by a general over-all principle which combines many previously different looking items.

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

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

Consider a discrete sample space S. Let $D, H_1, H_2, \ldots$ be a countable set of events (subsets) of S. The countable set $\mathbf{H} = \{H_1, H_2, \ldots\}$ is called the *hypothesis space*. The hypotheses H_i are exhaustive (at least one is true): $\bigcup H_i = S$. Bayes' rule is

$$P(H_i|D) = \frac{P(D|H_i)P(H_i)}{P(D)}. \quad (1)$$

If the hypotheses are mutually exclusive ($H_i \bigcap H_j = \emptyset$ for all i, j), then $P(D) = \sum_i P(D|H_i)P(H_i)$. Bayes rule looks perfect but it is not. The prior probably $P(H)$ is not known in general. People have resorted to various more or less arbitrary choices of $P(H)$ such as assuming all hypotheses are equally likely. Clearly, this is not always justified. The elusive goal seemed to be a universal choice of $P(H)$ which is always applicable, whatever the real prior distribution is.

Essentially combining the ideas of Epicurus, Ockham, Bayes, and modern computability theory, R.J. Solomonoff, [9], has successfully invented a 'perfect' theory of induction by using Kolmogorov complexity to give a universal prior distribution for Bayes' rule.

2 Inductive Inference

Let us consider theory formation in science as the process of obtaining a compact description of the past observations together with predictions of future ones. The investigator obtains increasingly longer initial segments of an infinite binary sequence ω by performing more and more experiments on some aspect of nature. To describe the underlying regularity of ω, the investigator tries to formulate a theory that governs ω, on the basis of the outcome of past experiments. Candidate theories (hypotheses) are identified with computer programs that compute binary sequences starting with the observed initial segment.

A concept can be viewed as a probability distribution (rather, measure) μ over a sample space $S = \mathcal{B}^\infty$ of all one-way infinite binary sequences. Each such sequence corresponds to one never-ending sequential history of conjectures and refutations and confirmations. Clearly, if we know μ then we can best predict by Bayes' Rule that

$$\mu(a|x) = \mu(xa)/\mu(x). \tag{2}$$

The final probability $\mu(a|x)$ is the probability of the next symbol being a, given the initial sequence x. Obviously we now only need the prior probability μ to evaluate $\mu(a|x)$. The goal of inductive inference in general is to be able to either (i) predict, or extrapolate, the next element after x, or (ii) to infer an underlying effective process that generated x, and hence to be able to predict the next symbol. In the most general deterministic case such an effective process is a Turing machine, but it can also be a probabilistic Turing machine or, say, a Markov process. The central task of inductive inference is to find a universally valid approximation to μ which is good at estimating the conditional probability that a given segment x will be followed by a segment y.

In general this is impossible. But suppose restrict the class of priors to the recursive semimeasures (see [7] for precise definitions). For the purpose of understanding this lecture, the reader can simply think semimeasures as the computable probability distributions. Under this relatively mild restriction, it turns out that we can use the single universal semimeasure $\mathbf{M}$ as a 'universal prior' (replacing the real prior) for prediction.

Definition 1 The *Kolmogorov complexity* of x, denoted as $K(x)$, is the length of shortest self-delimiting binary program for a universal computer (like a universal Turing machine) which cause the computer to compute and print x and then halt. The reader can think of $\mathbf{M}(x)$ as $2^{-K(x)}$ where $\mathbf{M}(y|x)$ is a good estimator for *all* recursive $\mu(y|x)$ [7].

When we predict the continuation of a sequence over $\{0,1\}^*$ randomly drawn from a distribution μ, can we estimate the error we make using $\mathbf{M}$ for prediction instead of the actually recursive distribution μ?

In the following analysis we assume that the one-way infinite binary sequences in $S = \{0,1\}^\infty$ are distributed according to measure μ. If we want to predict the next element (0 or 1) which will appear after an initial sequence $x \in \{0,1\}^*$, then we can do no better than predicting according to $\mu(0|x)$. If instead we predict according to another distribution ρ, then the prediction error can be measured as the difference of the probabilities involved: $|\rho(0|x) - \mu(0|x)|$. Because of the binary choice between 0 and 1 we only have to consider the conditional probability of predicting 0. Instead of the absolute value of the difference we take the square of the difference in the sequel. Instead of looking at the error in the nth prediction after a *particular* initial segment of length $n-1$ we consider the μ-average of the nth prediction error over all initial segments of length $n-1$. Of course, we are interested in the prediction error we make if ρ is set to $\mathbf{M}$.

Definition 2 Let S_n be the μ-expected value of the square of the difference in μ-probability and $\mathbf{M}$-probability of 0 occurring at the nth prediction:

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

We may call S_n the *expected squared error at the nth prediction.*

Theorem 1 *Let μ be a recursive semimeasure. Using the notation above, $\sum_n S_n \le k/2$ with $k = K(\mu)\ln 2$. (Hence, S_n converges to 0 faster than $1/n$.)*

THEOREM 2 *Let μ be a positive recursive measure. If the length of y is fixed and the length of x grows to infinity, then $\mathbf{M}(y|x)/\mu(y|x) \to 1$, with μ-probability one.*

Above, $K(\mu)$ is length of the shortest self-delimiting binary program from which the reference universal computer computes $\mu(x)$ when it is given data x. Instead of using the result in Theorem 1, it is perhaps conceptually easier to use Theorem 2 for the estimation required in Equation 2. To estimate the conditional probability $\mu(y|x)$ that the next segment after x is y we use the conditional prior probability: $\mathbf{M}(y|x) = \mathbf{M}(xy)/\mathbf{M}(x)$. To be able to do so we have to assume that μ in Equation 2 is a recursive measure. If the length of y is fixed, and the length of x grows to infinity, then Theorem 2 states that the conditional *a priori* probability is almost always asymptotically equal to the conditional probability.

The convergence is very fast in some sense. With S_n as in Definition 2, the expected squared error in the nth prediction of 0. For sequences over $\{0,1\}$, this figure of merit accounts for all error in the nth prediction. By Theorem 1, 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.

It is known [7] that $-\log \mathbf{M}(x) = K(x) \pm O(\log K(x))$. This implies that Bayes' Rule using the universal prior distribution gives a formal meaning to Occam's razor principle. The universal prior distribution $\mathbf{M}()$ is not computable, But in the sequel we see that many inference models or principles can be viewed as computable approximations to Solomonoff's inference principle.

3 Recursion-Theoretic Learning

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 in the limit, following E.M. Gold , [3]. 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.

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, taking any positive prior distribution $P(H_k)$, say $P(H_k) = 1/k(k+1)$, and apply Bayes' Rule 1:

$$P(H_k|D) = \frac{P(D|H_k)P(H_k)}{\sum\{P(H_j) : f_j \text{ is consistent with } D\}}. \tag{3}$$

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 (of E.M. Gold) 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.

3.1 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 $\mathbf{M}$.

For this analysis, replace the examples by binary self-delimiting codes [7]: $e = (x, y, 1)$ by $\bar{e} = \bar{x}\bar{y}1$ and $e = (x, y, 0)$ by $\bar{e} = \bar{x}\bar{y}0$. Here, $\bar{x} = 1^{l(x)}0x$, so the machine can see were 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: $P(H_k) = \mathbf{M}\left(\bigcup_{D \in \mathcal{D}} D\right)$. If we assume a recursive distribution on the examples, Solomonoff's maxim says we must predict according to: $\mathbf{M}(e|D)$. By Theorem 1, the expected squared error S_n in the n-th prediction goes to zero faster than $1/n$.

3.2 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 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. In fact, given $\mathbf{m}(x) = 2^{-K(x)}$, there is an algorithm which makes at most $K(n)$ mistakes. For regular n this $K(n)$ can be very low.*

4 Pac-Learning

The model of pac-learning introduced in 1984 by Les Valiant [10] is aimed at describing 'feasible' learning. According to commonly accepted views in the theory of computation, 'feasibility' means that the learning algorithm should run in polynomial time and use a polynomial number of examples. This requirement implies that not all examples can turn up. Hence, it is impossible to infer a concept precisely. This means that we can only hope to learn the concept approximately. But we also have to account for where the examples come from. We could be presented with unrepresentative examples for the concept to be inferred. It seems reasonable to assume that the examples are drawn randomly from a sample space according to a probability distribution. The approximation between the target concept and the learned concept can be expressed in the probability of the set of examples on which the two concepts disagree. In practice, the speed of learning is realized only because learning is just 'approximately correct' with 'high probability'. For example, after a child is given several sample apples, he likely forms an approximately correct concept of apple.

> "Human learning often shows remarkable properties of convergence. In large populations there is a high degree of agreement on the meaning of thousands of words as they relate to everyday situations. This suggests that highly reliable program acquisition may be feasible even in the absence of explicit programming. Hence it is reasonable to insist that our study of learning be disciplined by insistence on fast convergence." [Valiant]

DEFINITION 3 A concept class $\mathcal{C}$ is *pac-learnable* (probably approximately correct learnable) iff there exists a (possibly randomized) learning algorithm A such that, for each $f \in \mathcal{C}$ and $\epsilon > 0$, algorithm A halts in a finite number of steps and examples, and outputs a concept $h \in \mathcal{C}$ which satisfies the following. With probability at least $1 - \epsilon$, $\sum_{f(v) \neq h(v)} P(v) < \epsilon$. A concept class is *polynomially pac-learnable* iff it is pac-learnable and the learning algorithm always halts within time and number of examples $p(l(f), 1/\epsilon)$, for some polynomial p.

DEFINITION 4 Let $\alpha < 1$ be a constant, m be the number of examples, and s be the length (in number of bits) of the smallest concept in C consistent with the examples. An *Occam algorithm* is a polynomial time algorithm which finds a hypothesis $h \in \mathcal{C}$ consistent with the examples and satisfying $K(h) \leq s^c m^\alpha$.

THEOREM 3 [**Occam's Razor Theorem**] *A concept class $\mathcal{C}$ is polynomially pac-learnable if there is an Occam algorithm for it.*

Informally, Occam's Razor Theorem says that any consistent concept of size 'reasonably' shorter than the size of data is an 'approximately' correct concept with high probability. The shorter the conjecture is, the more efficiently it explains the data, hence the more precise the future prediction.

Occam's Razor Theorem is a very useful tool for identifying polynomial time learnable classes in the pac model.

EXAMPLE 4.1 It is known that it is NP-complete to find the minimum k-DNF formula consistent with polynomially many positive and negative data. As an example, we show that the class of k-DNF formulas over n variables is polynomial time pac-learnable.

Consider concepts which are Boolean formulas over variables $x_1, \ldots, x_n$. A k-DNF formula is a DNF formula with at most k literals per monomial.

Step 1 Set $g := \sum_{m \in M_k} m$, where M_k is the set of monomials with at most k literals;

Step 2 Repeat $(2n)^{k+2}$ times:
draw v :=next negative example for f; delete all m from g for which $m(v) = 1$.

Initially, g contains all monomials of at most k literals. There are at most $(2n)^{k+1}$ of them—a slight overestimate. Fix any $\epsilon > 0$. Initially, all positive examples satisfy g. At each step, all monomials in g which imply the drawn negative example are deleted. Clearly, we maintain the invariant $\sum_{f(v)=1 \& g(v)=0} P(v) = 0$. That is, no error on the positive side is introduced. Moreover, rule g stays consistent with the sample. The size of g stays bounded by $(2n)^{k+1}$.

Next apply Theorem 3, with $\alpha = 0$ and some constant $c < 2k$. After a polynomial number of, say $m = (2n)^{k+2}$ examples, and polynomially many steps, we have with probability greater than $1-\epsilon$ that $\sum_{g(v)=0} P^+(v) < \epsilon$ and $\sum_{g(v)=1} P^-(v) < \epsilon$. By Definition 3, therefore, we have polynomially pac-learned k-DNF.

See [1] and the survey of D. Angluin in *STOC'92.*

5 Simple Pac-Learning

It has turned out that many problems are intractable (NP-hard) under the original pac-learning model. Can one obtain a model where more concept classes are polynomial time learnable? Maybe one can just learn a concept under *some* distributions, like the computable ones. And maybe it is too much to ask to be able to learn all finite automata fast (humans cannot either), but surely we ought to be able to learn a sufficiently *simple* finite automaton fast (as humans can) [6].

Certain concepts which are not know to be polynomially pac-learnable in the distribution-free model, can be polynomially *pac*-learned under the uniform distribution by a specialized learning algorithm. However, learning under one distribution may be too restrictive to be useful. It may be useful to investigate polynomial pac-learnability under various classes of distributions. One would like this class to be wide enough to be interesting, yet restrictive enough to make more concept classes polynomially learnable. It is possible to develop a theory of simple pac-learning in the following sense [6].

Discrete sample set. Let Q be a distribution on a discrete sample set. A concept class is polynomially pac-learnable under all distributions which are multiplicatively dominated by Q *provided you draw your sample according to Q in the learning phase* iff the concept class is polynomially pac-learnable under Q. The caveat is that while pac-learning for distribution P we need to draw the sample according to Q. We can develop this theory with $Q = \mathbf{m}$, the universal enumerable distribution $\mathbf{m}(x) = 2^{-K(x)}$. (This bfm is the discrete sample space analog of bfM which is defined for continuous sample spaces.) The remarkable property of the distribution $\mathbf{m}$ is that in a polynomial sample with overwhelming probability all examples of logarithmic complexity (the simple examples) will be represented. Hence, a learning algorithm simply needs to reconstruct a concept approximately when all simple examples are given. This leads to a new type of learning algorithms.

Continuous sample set. Let μ be a semimeasure on a continuous sample set. A concept class is polynomially pac-learnable under all semimeasures which are multiplicatively dominated by μ iff the concept class is pac-learnable under μ. Here we have dropped the polynomial time requirement and also eliminated the need for drawing according to μ in the learning phase. We take for μ the universal enumerable semimeasure $\mathbf{M}$, and we can show that there is a class which is pac-learnable under $\mathbf{M}$ but not pac-learnable under all distributions.

Several concept classes that are not pac learnable have been shown to be simple pac learnable.

6 MDL Principle

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 [8]. The method can be viewed as a computable approximation to the noncomputable approach in Section 2, 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.

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. Bayes' rule is $P(H|D) = P(D|H)P(H)/P(D)$. 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

$$-\log P(H|D) = -\log P(D|H) - \log P(H) + \log P(D). \tag{4}$$

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 same as minimizing $-\log P(D|H) - \log P(H)$. Let us assume that H and D are expressed as natural numbers or finite binary strings. If P is recursive, then $\delta_0(H|P) = \log(\mathbf{m}(H)/P(H))$ is a universal P-test [7]. This notion is part of a sophisticated mathematical theory. For us it is only important that for P-random H the value of $\delta_0(H|P)$ is low. Moreover, the overwhelming majority of H's is P-random in the sense that $\sum_H P(H) 2^{\log \mathbf{m}(H)/P(H)} \leq 1$. The analogous statement holds for $P(D|H)$. For P-random H and $D|H$, we can set: $\log P(H) = \log \mathbf{m}(H) + O(1)$, $\log P(D|H) = \log \mathbf{m}(D|H) + O(1)$. We have [7]: $\log \mathbf{m}(H) = -K(H) \pm O(1)$ and $\log \mathbf{m}(D|H) = -K(D|H) \pm O(1)$, where $K()$ is the prefix complexity. That is, 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 prefix-codes are achieved by constructive versions of the Shannon-Fano code [7].

The term $-\log P(D|H)$ is also known as the *self-information* in information theory and the negative log-likelihood in statistics, 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 by the corresponding $\mathbf{m}$-probabilities, we have [7] $K(H|D) = K(H) + K(D|H) - K(D) + O(1)$. Since $K(H) + K(D|H, K(H)) = K(D) + K(H|D, K(D)) + O(1)$, by symmetry of information [7], 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 $K(H|D)$ is precisely MDL principle.

7 Maximum Likelihood and Maximum Entropy

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 [2].

It is known In statistics there are a number of important applications where the ML principle fails, but where the maximum entropy principle of E.T. Jaynes [4] has been successful, and vice versa.

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 as follows.

The *maximum entropy (ME) principle* tells us to use the estimated values $\hat{p}_i$ which maximize the entropy function $H(p_1, \ldots, p_k) = -\sum_{i=1}^k p_i \ln p_i$, subject to $\sum_{i=1}^k p_i = 1$ and some other constraints.

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

Maximum Entropy principle can be considered as a special case of the MDL principle. Let S_θ be the set of outcomes with values $(n_1, \ldots, n_k)$, with $n_i = np_i$, corresponding to $\theta = (p_1, \ldots, p_k)$. Then due to the small number of values (k) in θ relative to the size of sets, we have $\log \sum_\theta d(S_\theta) \approx \max_\theta \{\log d(S_\theta)\}$. The lefthand side of this equation is is the minimum description; the righthand side of is the maximum entropy.

8 Thinking, Consciousness, Feeling

We have haphazardly skimmed thousands of years to search for the meanings and methods for inductive learning. Learning by machines may not only be possible (in the sense of recursion-theoretic learning) but also may be *feasible* (in the sense of pac-learning). Of course, this satisfies only some theoretical issues. For practicality we need to translate and adapt to real life issues. This may be simple in principle but it may be very hard in practice. Being in an optimistic mood of representing problems more simple than the probably are, it is seductive to think of the real problem in mimicking human learning process as just a massive engineering problem.

The human cerebral cortex (thinking part of human brain) contains 44×10^9 cells. (The involuntary part of the human brain, the cerebellum, contains even more, 10^{11} cells.) Each cortical cell consists of a cell body forming the root of a tree like structure of which the leaves are called 'synapses'. These synapses touch other cell bodies. Each cell body may receive up to 50,000 synapses (some of which may come from the same source). Once we are able to build parallel computers with such an amount of processors and interconnections, the 50 years old promise of AI has it that we can build real learning robots that resemble human learning abilities can be built, provided they can be simulated by mechanical means at all. The main problem here is that there is currently no knowledge whatsoever about the organizational principle of the brain and how it works at all. The real physiological knowledge available about the human brain and mind is on par with knowledge obtained from a large computer where we cut a random wire and the printer stops. Concluding that the wire is the 'printing center' of the computer (as analogue of what happens in brain physiology) is superficial to the point of being positively false.

We have not touched the subject of neural networks in this paper because we regard the neural networks as just some way of implementing learning—not different from any other computer. The neural network itself is not a new learning theory.

We have avoided talking about computer 'intelligence'. Apparently for a computer to be intelligent, it must be able to 'think' and 'feel'. In order to do all of these, we must have a consensus on a meaningful formal definition of 'consciousness'. Apparently this is the ultimate issue that faces AI, psychology, science, and humanity.

References

[1] M. Anthony and N. Biggs. *Computational Learning Theory.* Cambridge University Press, 1992.

[2] R.A. Fisher. On the mathematical foundations of theoretical statistics. *Philos. Trans. Roy. Soc. London, Ser. A*, 222:309–368, 1925.

[3] E.M. Gold. Language identification in the limit. *Inform. Contr.*, 10:447–474, 1967.

[4] E.T. Jaynes. On the rationale of maximum entropy methods. *Proceedings of the IEEE*, 70:939–952, 1982.

[5] J.G. Kemeny. The use of simplicity in induction. *Philos. Rev.*, 62:391–408, 1953.

[6] M. Li and P.M.B. Vitányi. A theory of learning simple concepts under simple distributions. *SIAM J. Computing*, 20(5):915–935, 1991.

[7] M. Li and P.M.B. Vitányi. *An introduction to Kolmogorov complexity and its applications.* Springer Verlag, 1992. To appear.

[8] J. Rissanen. Modeling by the shortest data description. *Automatica-J.IFAC*, 14:465–471, 1978.

[9] R.J. Solomonoff. A formal theory of inductive inference, part 1 and part 2. *Inform. Contr.*, 7:1–22, 224–254, 1964.

[10] L.G. Valiant. A theory of the learnable. *Comm. ACM*, 27:1134–1142, 1984.