On Prediction by Data Compression*

Paul Vitányi[1] and Ming Li[2]

[1] CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: paulv@cwi.nl
[2] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. Email: mli@cs.cityu.edu.hk

Abstract. Traditional wisdom has it that the better a theory compresses the learning data concerning some phenomenon under investigation, the better we learn, generalize, and the better the theory predicts unknown data. This belief is vindicated in practice but apparently has not been rigorously proved in a general setting. Making these ideas rigorous involves the length of the shortest effective description of an individual object: its Kolmogorov complexity. In a previous paper we have shown that optimal compression is almost always a best strategy in hypotheses identification (an ideal form of the minimum description length (MDL) principle). Whereas the single best hypothesis does not necessarily give the best prediction, we demonstrate that nonetheless compression is almost always the best strategy in prediction methods in the style of R. Solomonoff.

1 Introduction

Given a body of data concerning some phenomenon under investigation, we want to select the most plausible hypothesis from among all appropriate hypotheses, or predict future data. 'Occam's razor' tells us that, all other things being equal, the simplest explanation is the most likely one. Interpreting 'simplest' as 'having shortest description', the most likely hypothesis is the most compressed one. Traditional wisdom says that improved compression of the learning data samples leads to better generalization properties and better prediction on unseen data. The length of the shortest effective description of some object is its Kolmogorov complexity. The argument says that among all "appropriate" hypotheses the one of least Kolmogorov complexity is the most likely one. In [8] we have rigorously demonstrated that this piece of traditional wisdom is "almost always" valid. This shows that compression is good for hypothesis selection. But is it also good for prediction?

* Ming Li was supported in part by NSERC operating grant OGP-046506, ITRC, and a CGAT grant and the Steacie Fellowship. On sabbatical leave from: Department of Computer Science, University of Waterloo; Email: mli@math.uwaterloo.ca. Paul Vitányi is also affiliated with the University Amsterdam and was supported by NSERC through International Scientific Exchange Award ISE0125663, and by the European Union through NeuroCOLT ESPRIT Working Group Nr. 8556, and by NWO through NFI Project ALADDIN under Contract number NF 62-376.

The best single hypothesis does not necessarily give the best prediction. For example, consider a situation where we are given a coin of unknown bias p of coming up "heads" which is either $p_1 = \frac{1}{3}$ or $p_2 = \frac{2}{3}$. Suppose we have determined that there is probability $\frac{2}{3}$ that $p = p_1$ and probability $\frac{1}{3}$ that $p = p_2$. Then the "best" hypothesis is the most likely one: $p = p_1$ which predicts a next outcome "heads" as having probability $\frac{1}{3}$. Yet the best prediction is that this probability is the expectation of throwing "heads" which is

$$\frac{2}{3}p_1 + \frac{1}{3}p_2 = \frac{4}{9}.$$

Thus, the fact that compression is good for hypothesis identification problems does not imply that compression is good for prediction. We analyse the relation between compression of the data sample and prediction in the very general setting of R. Solomonoff [14, 15]. We explain Solomonoff's prediction method using the universal distribution. We show that this method is not equivalent to the use of shortest descriptions. Nonetheless, we demonstrate that compression of descriptions almost always gives optimal prediction.

1.1 Background and Previous Work

The classical method for induction is Bayes's rule. The problem with applying Bayes's rule is that one requires the prior probabilities of the hypotheses first. Unfortunately, it is often impossible to obtain these. In the unlikely case that we possess the true prior distribution, in practice the data tend to be noisy due to the measuring process or other causes. The latter confuses Bayes's rule into overfitting the hypothesis by adding random features while trying to fit the data.

One way out of the conundrum of *a priori* probabilities is to require prediction or inference of hypotheses to be completely or primarily data driven. For prediction this was achieved using the Kolmogorov complexity based universal distribution, [14, 15], and for hypothesis identification by the minimum description length (MDL or MML) approach, [10, 11, 19, 20]

Ideally, the description lengths involved should be the shortest effective description lengths. (We use 'effective' in the sense of 'Turing computable', [16].) Shortest effective description length is asymptotically unique and objective and known as the *Kolmogorov complexity* of the object being described. Such shortest effective descriptions are 'effective' in the sense that we can compute the described objects from them. Unfortunately, it can be shown, see [6], that one cannot compute the length of a shortest description from the object being described. This obviously impedes actual use. Instead, one needs to consider computable approximations to shortest descriptions, for example by restricting the allowable approximation time. This course is followed in one sense or another in the practical incarnations such as MML and MDL. There one often uses simply the Shannon-Fano code, which assigns prefix code length $-\log P(x)$ to x irrespective of the regularities in x. If $P(x) = 2^{-l(x)}$ for every $x \in \{0,1\}^n$, then the code word length of an all-zero x equals the code word length of a truly irregular x. While the Shannon-Fano code gives an expected code word length close to the entropy, it does not distinguish the regular elements of a probability ensemble from the random ones.

The code of the shortest effective descriptions, with the Kolmogorov complexities as the code word length set, also gives an expected code word length close to the entropy yet compresses the regular objects until all regularity is squeezed out. All shortest effective descriptions are completely random themselves, without any regularity whatsoever. Kolmogorov complexity can be used to develop a theory of (idealized) minimum description length reasoning. In particular, shortest effective descriptions enable us to rigorously analyse the relation between shortest description length reasoning and Bayesianism. This provides a theoretical basis for, and gives confidence in, practical uses of the various forms of minimum description length reasoning mentioned.

In [7, 8] we rigorously derived and justify this Kolmogorov complexity based form of minimum description length, 'Ideal MDL', via the Bayesian approach using a particular prior distribution over the hypotheses (the so-called 'universal distribution'). This leads to a mathematical explanation of correspondences and differences between Ideal MDL and Bayesian reasoning, and in particular it gives some evidence under what conditions the latter is prone to overfitting while the former isn't. Namely, for hypothesis identification Ideal MDL using Kolmogorov complexity can be reduced to the Bayesian approach using the universal prior distribution, provided the minimum description length is reached for those hypotheses with respect to which the data sample is *individually random* in the sense of Martin-Löf, [9]. Under those conditions Ideal MDL, Bayesianism, MDL, and MML, select pretty much the same hypothesis. These conditions hold for almost all combinations of hypothesis and data sample. Consequently, we showed that the hypothesis that compresses the data sample most is almost always the "best" hypothesis.

2 Roots of Kolmogorov Complexity

2.1 A Lacuna of Classical Probability Theory

An adversary claims to have a true random coin and invites us to bet on the outcome. The coin produces a hundred heads in a row. We say that the coin cannot be fair. The adversary, however, appeals to probability theory which says that each sequence of outcomes of a hundred coin flips is equally likely, $1/2^{100}$, and one sequence had to come up.

Probability theory gives us no basis to challenge an outcome *after* it has happened. We could only exclude unfairness in advance by putting a penalty side-bet on an outcome of 100 heads. But what about 1010...? What about an initial segment of the binary expansion of π?

Regular sequence $\Pr(00000000000000000000000000) = \frac{1}{2^{26}}$,
Regular sequence $\Pr(01000110110000010100111001) = \frac{1}{2^{26}}$,
Random sequence $\Pr(10010011011000111011010000) = \frac{1}{2^{26}}$.

The first sequence is regular, but what is the distinction of the second sequence and the third? The third sequence was generated by flipping a quarter. The second sequence is very regular: $0, 1, 00, 01, \ldots$. The third sequence will pass (pseudo-)randomness tests.

In fact, classical probability theory cannot express the notion of *randomness of an individual sequence.* It can only express expectations of properties of outcomes of random processes, that is, the expectations of properties of the total set of sequences under some distribution.

Only relatively recently, this problem has found a satisfactory resolution by combining notions of computability and statistics to express the complexity of a finite object. This complexity is the length of the shortest binary program from which the object can be effectively reconstructed. It may be called the *algorithmic information content* of the object. This quantity turns out to be an attribute of the object alone, and absolute (in the technical sense of being recursively invariant). It is the *Kolmogorov complexity* of the object.

2.2 A Lacuna of Information Theory

Shannon's classical information theory assigns a quantity of information to an ensemble of possible messages. All messages in the ensemble being equally probable, this quantity is the number of bits needed to count all possibilities. This expresses the fact that each message in the ensemble can be communicated using this number of bits. However, it does not say anything about the number of bits needed to convey any individual message in the ensemble. To illustrate this, consider the ensemble consisting of all binary strings of length 9999999999999999.

By Shannon's measure, we require 9999999999999999 bits on the average to encode a string in such an ensemble. However, the string consisting of 9999999999999999 1's can be encoded in about 55 bits by expressing 9999999999999999 in binary and adding the repeated pattern '1'. A requirement for this to work is that we have agreed on an algorithm that decodes the encoded string. We can compress the string still further when we note that 9999999999999999 equals $3^2 \times 1111111111111111$, and that 1111111111111111 consists of 2^4 1's.

Thus, we have discovered an interesting phenomenon: the description of some strings can be compressed considerably, provided they exhibit enough regularity. This observation, of course, is the basis of all systems to express very large numbers and was exploited early on by Archimedes in his treatise *The Sand Reckoner*, in which he proposes a system to name very large numbers:

> "There are some, King Golon, who think that the number of sand is infinite in multitude [... or] that no number has been named which is great enough to exceed its multitude.[...] But I will try to show you, by geometrical proofs, which you will be able to follow, that, of the numbers named by me [...] some exceed not only the mass of sand equal in magnitude to the earth filled up in the way described, but also that of a mass equal in magnitude to the universe."

However, if regularity is lacking, it becomes more cumbersome to express large numbers. For instance, it seems easier to compress the number 'one billion,' than the number 'one billion seven hundred thirty-five million two hundred sixty-eight thousand and three hundred ninety-four,' even though they are of the same order of magnitude.

2.3 Lacuna in Randomness

In the context of the above discussion, random sequences are sequences that cannot be compressed. Now let us compare this with the common notions of mathematical randomness. To measure randomness, criteria have been developed that certify this quality. Yet, in recognition that they do not measure 'true' randomness, we call these criteria 'pseudo' randomness tests. For instance, statistical surveys of initial sequences of decimal digits of π have failed to disclose any significant deviations from randomness. But clearly, this sequence is so regular that it can be described by a simple program to compute it, and this program can be expressed in a few bits.

The notion of randomness of individual objects has a long history which goes back to the initial attempts by von Mises, [17], to formulate the principles of application of the calculus of probabilities to real-world phenomena. Classical probability theory cannot even express the notion of 'randomness of individual objects'. Following almost half a century of unsuccessful attempts, the theory of Kolmogorov complexity, [4], and Martin-Löf tests for randomness, [9], finally succeeded in formally expressing the novel notion of individual randomness in a correct manner, see [6]. Objects which are random in this sense will satisfy *all* effective tests for randomness properties—those which are known and those which are yet unknown alike.

3 Kolmogorov Complexity

The Kolmogorov complexity, [4, 22, 6], 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 $\{x, y, \ldots\} \subseteq \{0,1\}^*$ 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 $x = x_1 x_2 \ldots x_n$ has a special type of prefix code, called a *self-delimiting code*,

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

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 $T_1, T_2, \ldots$ be a standard enumeration of all Turing machines, and let $\phi_1, \phi_2, \ldots$ 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. The Kolmogorov complexity $C(x)$ of x is the length of the shortest binary program from which x is computed. Formally, we define this as follows.

Definition 1. The *Kolmogorov complexity* of x given y (for free on a special input tape) is

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

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

The Kolmogorov complexity is absolute in the sense of being recursively invariant by Church's Thesis and the ability of universal machines to simulate one another, [6]. For technical reasons we also need a variant of complexity, so-called prefix complexity, which associated with Turing machines for which the set of programs resulting in a halting computation is prefix free. We can realize this by equiping the Turing machine with a one-way input tape, a separate work tape, and a one-way output tape. Such Turing machines are called prefix machines since the halting programs for anyone of them form a prefix free set. Taking the universal prefix machine U we can define the prefix complexity analogously with the plain Kolmogorov complexity. If x^* is the first shortest program for x then the set $\{x^* : U(x^*) = x, x \in \{0,1\}^*\}$ is a *prefix code*. That is, each x^* is a code word for some x, and if x^* and y^* are code words for x and y with $x \neq y$ then x^* is not a prefix of x.

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

The nice thing about $K(x)$ is that we can interpret $2^{-K(x)}$ as a probability distribution. Namely, $K(x)$ is the length of a shortest prefix-free program for x. By the fundamental Kraft's inequality, see for example [6], we know that if $l_1, l_2, \ldots$ are the code-word lengths of a prefix code, then $\sum_x 2^{-l_x} \leq 1$. This leads to the notion of universal distribution—a rigorous form of Occam's razor–below.

4 Universal Distribution

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 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 $\sum_{x \in \mathcal{N}} P(x) \leq 1$. (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 $\mathcal{EP}$ of *enumerable* probability distributions on the sample space $\mathcal{N}$ (equivalently, $\{0, 1\}^*$). It is known, [6], that $\mathcal{EP}$ contains an element $\mathbf{m}$ that multiplicatively dominates all elements of $\mathcal{EP}$. That is, for each $P \in \mathcal{EP}$ there is a constant c such that $c\,\mathbf{m}(x) > P(x)$ for all $x \in \mathcal{N}$. We call $\mathbf{m}$ a *universal distribution*.

The family $\mathcal{EP}$ 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 $\mathcal{EP}$ 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's 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, [6], that

Lemma 4.
$$-\log \mathbf{m}(x) = K(x) \pm O(1). \tag{1}$$

That means that $\mathbf{m}$ assigns high probability to simple objects and low probability to complex or random objects. For example, for $x = 00 \ldots 0$ (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 $\mathbf{m}(y) = O(1/2^n)$.

4.1 Example: Betting Against a Crooked Player

Let us apply this to the betting problem on a not-known-to-be false coin we identified in Section 2.1 as a lacuna in probability theory.

Alice, walking down the street, comes across Bob, who is tossing a coin. He is offering odds to all passers-by on whether the next toss will be heads or tails. The pitch is this: he'll pay you two dollars if the next toss is heads; you pay him one dollar if the next toss is tails. Should she take the bet? If Bob is tossing a fair coin, it's a great bet. Probably she'll win money in the long run. After all, she would expect that half Bob's tosses would come up heads and half tails. Giving up only one dollar on each heads toss and getting two for each tails—why in a while she'd be rich!

Of course, to assume that a street hustler is tossing a fair coin is a bit of a stretch, and Alice is no dummy. So she watches for a while, recording how the c oin comes up for other betters, writing down a '1' for 'heads' and a '0' for 'ta ils'. After a while she has written 01010101010101. This doesn't look good. So Alice makes the following offer.

Alice pays Bob \$1 first and proposes that Bob pays her $2^{1000-K(x)}$ dollars. and x is the binary sequence of the 1,000 coin flip results. This is fair since Bob is only expected to pay her

$$\sum_{|x|=1000} 2^{-1000} 2^{1000-K(x)} \leq \$1,$$

by Kraft's inequality. So Bob should be happy to accept the proposal. But if Bob cheats, then, for example, Alice gets $2^{1000-\log 1000}$ dollars for a sequence like 0101010101... !

In the 1 versus 2 dollars scheme, Alice can propose to add this as an extra bonus pay. This way, she is guaranteed to win big: either polynomially increase her money (when Bob does not cheat) or exponentially increase her money (when Bob cheats).

5 Randomness Tests

One can consider those objects as nonrandom in which one can find sufficiently many regularities. In other words, we would like to identify 'incompressibility' with 'randomness'. This is proper if the sequences that are incompressible can be shown to possess the various properties of randomness (stochasticity) known from the theory of probability. That this is possible is the substance of the celebrated theory developed by the Swedish mathematician Per Martin-Löf.

There are many properties known which probability theory attributes to random objects. To give an example, consider sequences of n tosses with a fair coin. Each sequence of n zeros and ones is equiprobable as an outcome: its probability is 2^{-n}. If such a sequence is to be random in the sense of a proposed new definition, then the number of ones in x should be near to $n/2$, the number of occurrences of blocks '00' should be close to $n/4$, and so on.

It is not difficult to show that each such single property separately holds for all incompressible binary strings. But we want to demonstrate that incompressibility implies all conceivable effectively testable properties of randomness (both the known ones and the as yet unknown ones). This way, the various theorems in probability theory about random sequences carry over automatically to incompressible sequences. We do not develop the theory here but refer to the exhaustive treatment in [6] instead. We shall use the properties required in the sequel of this paper.

6 Bayesian Reasoning

Consider a situation in which one has a set of observations of some phenomenon, and also a finite or countably infinite set of hypotheses which are candidates to explain the phenomenon. For example, we are given a coin and we flip it 1000 times. We want to identify the probability that the coin has outcome 'head' in a single coin flip. That is, we want to find the bias of the coin. The set of possible hypotheses is uncountably infinite if we allow each real bias in $[0,1]$, and countably infinite if we allow each rational bias in $[0,1]$.

For each hypothesis H we would like to assess the probability that H is the 'true' hypothesis, given the observation of D. This quantity, $\Pr(H|D)$, can be described and manipulated formally in the following way.

Consider a sample space Ω. Let D denote a sample of outcomes, say experimental data concerning a phenomenon under investigation. Let $H_1, H_2, \ldots$ be an enumeration of countably many hypotheses concerning this phenomenon, say each H_i is a probability distribution over Ω. The list $\mathcal{H} = \{H_1, H_2, \ldots\}$ is called the *hypothesis space*. The hypotheses H_i are exhaustive and mutually exclusive.

For example, say the hypotheses enumerate the possible rational (or computable) biases of the coin. As another possibility there may be only two possible hypotheses: hypothesis H_1 which says the coin has bias 0.2, and hypothesis H_2 which puts the bias at 0.8.

Suppose we have *a priori* a distribution of the probabilities $P(H)$ of the various possible hypotheses in $\mathcal{H}$ which means that $\sum_{H \in \mathcal{H}} P(H) = 1$. Assume furthermore that for all $H \in \mathcal{H}$ we can compute the probability $\Pr(D|H)$ that sample D arises if H is the case. Then we can also compute (or approximate in case the number of hypotheses with nonzero probability is infinite) the probability $\Pr(D)$ that sample D arises at all

$$\Pr(D) = \sum_{H \in \mathcal{H}} \Pr(D|H)P(H).$$

From the definition of conditional probability it is easy to derive **Bayes's formula**[3]

$$\Pr(H|D) = \frac{\Pr(D|H)P(H)}{\Pr(D)}. \tag{2}$$

[3] Some Bayesians prefer replacing $\Pr(D|H)P(H)$ by a joint probability of data and hypotheses together, the prior $P(D,H) = \Pr(D|H)P(H)$.

The prior probability $P(H)$ is often considered as the learner's *initial degree of belief* in hypothesis H. In essence Bayes's rule is a mapping from *a priori* probability $P(H)$ to *a posteriori* probability $\Pr(H|D)$ determined by data D.

Continuing to obtain more and more data, this way the total inferred probability will concentrate more and more on the 'true' hypothesis. We can draw the same conclusion of course, using more examples, by the law of large numbers. In general, the problem is not so much that in the limit the inferred probability 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. Given the prior probability of the hypotheses, it is easy to obtain the inferred probability, and therefore to make informed decisions. However, in general we don't know the prior probabilities. The following MDL approach in some sense replaces an unknown prior probability by a fixed 'universal' probability.

7 Prediction by Compression

Theoretically the idea of predicting time sequences using shortest effective descriptions was first formulated by R. Solomonoff, [14]. He uses Bayes's formula equipped with a fixed 'universal' prior distribution. In accordance with Occam's dictum, it tells us to go for the explanation that compresses the data the most—but not quite as we shall show.

The aim is to *predict* outcomes concerning a phenomenon μ under investigation. In this case we have some prior evidence (prior distribution over the hypotheses, experimental data) and we want to predict future events. This situation can be modelled by considering a sample space S of one-way infinite sequences of basic elements $\mathcal{B}$ defined by $S = \mathcal{B}^\infty$. We assume a prior distribution μ over S with $\mu(x)$ denoting the probability of a sequence starting with x. Here $\mu(\cdot)$ is a *semimeasure*[4] satisfying

$$\begin{aligned} \mu(\epsilon) &\leq 1 \\ \mu(x) &\geq \sum_{a \in \mathcal{B}} \mu(xa). \end{aligned}$$

Given a previously observed data string x, the inference problem is to predict the next symbol in the output sequence, that is, to extrapolate the sequence x. In terms of the variables in formula 2, H_{xy} is the hypothesis that the sequence starts with initial segment xy. Data D_x consists of the fact that the sequence starts with initial segment x. Then, $\Pr(D_x|H_{xy}) = 1$, that is, the data is forced by the hypothesis, or $\Pr(D_z|H_{xy}) = 0$ for z is not a prefix of xy, that is, the hypothesis contradicts the data. For $P(H_{xy})$ and $\Pr(D_x)$ in formula 2 we substitute $\mu(xy)$ and $\mu(x)$, respectively. For $P(H_{xy}|D_x)$ we substitute $\mu(y|x)$. This

[4] Traditional notation is '$\mu(\Gamma_x)$' instead of '$\mu(x)$' where *cylinder* $\Gamma_x = \{\omega \in S :$ ω starts with $x\}$. We use '$\mu(x)$' for convenience. μ is a *measure* if equalities hold.

way the formula is rewritten as

$$\mu(y|x) = \frac{\mu(xy)}{\mu(x)}. \tag{3}$$

The final probability $\mu(y|x)$ is the probability of the next symbol string being y, given the initial string x. Obviously we now only need the prior probability μ to evaluate $\mu(y|x)$. 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 (which makes its brief and single appearance here). 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 we restrict the class of priors μ to the *recursive* semimeasures and restrict the set of basic elements $\mathcal{B}$ to $\{0,1\}$. Under this relatively mild restriction on the admissible semimeasures μ, it turns out that we can use the single universal semimeasure $\mathbf{M}$ as a 'universal prior' (replacing the real prior μ) for prediction. The notion of universal semimeasure $\mathbf{M}$ is a continuous version of $\mathbf{m}$ we saw before, and which is explained in [6]. defined with respect to a special type Turing machine called *monotone* Turing machine. The universal semimeasure $\mathbf{M}$ multiplicatively dominates all enumerable (computable from below) semimeasures. If we flip a fair coin to generate the successive bits on the input tape of the universal reference monotone Turing machine, then the probability that it outputs $x\alpha$ (x followed by something) is $\mathbf{M}(x)$.

It can be shown that the universal distribution *itself* is directly suited for prediction. The universal distribution combines a weighted version of the predictions of all enumerable semimeasures, including the prediction of the semimeasure with the shortest program. It is not a priori clear that the shortest program dominates in all cases—and as we shall see it does not. However, we show that in the overwhelming majority of cases—the typical cases—the shortest program dominates sufficiently to validate the approach that only uses shortest programs for prediction. The properties of $\mathbf{M}(x)$ allow us to demonstrate that a minimum description length procedure is almost always optimal for prediction.

Given a semimeasure on $\{0,1\}^\infty$ and an initial binary string x our goal is to find the most probable extrapolation of x. That is, taking the negative logarithm on both sides of Equation 3, we want to determine y with $l(y) = n$ that minimizes

$$-\log \mu(y|x) = -\log \mu(xy) + \log \mu(x).$$

We assume that μ be a *recursive* semimeasure.

This theory of the *universal semimeasure* $\mathbf{M}$, the analogue in the sample space $\{0,1\}^\infty$ of $\mathbf{m}$ in the sample space $\{0,1\}^*$ equivalent to $\mathcal{N}$, is developed in [6], Chapter 4, and Chapter 5. A celebrated result of Solomonoff, [15], says that

$\mathbf{M}$ is very suitable for prediction. 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 5. *Let μ be a recursive semimeasure. Using the notation above, $\sum_n S_n \leq k/2$ with $k = K(\mu)\ln 2$. (Hence, S_n converges to 0 faster than $1/n$.)*

A proof using Kulback-Leibler divergence is given in [6]. There it is additionally demonstrated that for almost all unbounded x the conditional probability of $\mathbf{M}$ converges to the conditional probability of μ. Note that while the following Theorem *does* imply the convergence of the conditional probabilities similarly to Theorem 5, it *does not* imply the speed of convergence estimate. Conversely, Theorem 5 does not imply the following.

Theorem 6. *Let μ be a positive recursive measure If the length of y is fixed and the length of x grows to infinity, then*

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

with μ-probability one. In infinite sequences ω with prefixes x satisfying the displayed asymptotics are precisely the μ-random sequences.

Proof. We use an approach based on the Submartingale Convergence Theorem, [1] pp. 324-325, which states that the following property holds for each sequence of random variables $\omega_1, \omega_2, \ldots$. If $f(\omega_{1:n})$ is a μ-submartingale, and the μ-expectation $\mathbf{E}|f(\omega_{1:n})| < \infty$, then it follows that $\lim_{n\to\infty} f(\omega_{1:n})$ exists with μ-probability one.

In our case,

$$t(\omega_{1:n}|\mu) = \frac{\mathbf{M}(\omega_{1:n})}{\mu(\omega_{1:n})}$$

is a μ-submartingale, and the μ-expectation $\mathbf{E}t(\omega_{1:n}|\mu) \leq 1$. Therefore, there is a set $A \subseteq \mathcal{B}^\infty$ with $\mu(A) = 1$, such that for each $\omega \in A$ the limit $\lim_{n\to\infty} t(\omega_{1:n}|\mu) < \infty$. These are the μ-random ω's by Corollary 4.8 in [6]. Consequently, for fixed m, for each ω in A, we have

$$\lim_{n\to\infty} \frac{\mathbf{M}(\omega_{1:n+m})/\mu(\omega_{1:n+m})}{\mathbf{M}(\omega_{1:n})/\mu(\omega_{1:n})} = 1,$$

provided the limit of the denominator is not zero. The latter fact is guarantied by the universality of $\mathbf{M}$: for each $x \in \mathcal{B}^*$ we have $\mathbf{M}(x)/\mu(x) \geq 2^{-K(\mu)}$ by Theorem 4.4 and Equation 4.10 in [6].

Example 1. Suppose we are given an infinite decimal sequence ω. The even positions contain the subsequent digits of $\pi = 3.1415\ldots$, and the odd positions contain uniformly distributed, independently drawn random decimal digits. Then, $\mathbf{M}(a|\omega_{1:2i}) \to 1/10$ for $a = 0, 1, \ldots, 9$, while $\mathbf{M}(a|\omega_{1:2i+1}) \to 1$ if a is the ith digit of π, and to 0 otherwise.

There are two possibilities to associate complexities with machines. The first possibility is to take the length of the shortest program, while the second possibility is to take the negative logarithm of the universal probability. In the discrete case, using prefix machines, these turned out to be the same by the Coding Theorem 4. In the continuous case, using monotone machines, it turns out they are different.

Definition 7. The complexity KM is defined as

$$KM(x) = -\log \mathbf{M}(x).$$

In contrast with C and K complexities, in the above definition the greatest prefix-free subset of *all* programs which produce output starting with x on the reference monotone machine U are weighed.

Definition 8. Let U be the reference monotone machine. The complexity Km, called *monotone complexity*, is defined as

$$Km(x) = \min\{l(p) : U(p) = x\omega, \omega \in S_{\mathcal{B}}\}.$$

We omit the Invariance Theorems for KM complexity and Km complexity, stated and proven completely analogous to the Theorems with respect to the C and K varieties. By definition, $KM(x) \leq Km(x)$. In fact, all proper complexities coincide up to a logarithmic additive term. It has been shown that equality does not hold: the difference between $KM(x)$ $(= -\log \mathbf{M}(x))$ and $Km(x)$ is very small, but still rises unboundedly. This contrasts with the equality between $-\log \mathbf{m}(x)$ and $K(x)$ in Theorem 4. Intuitively, this phenomenon is justified by exposing the relation between $\mathbf{M}$ and $\mathbf{m}$.

The Coding Theorem 4 states that $K(x) = -\log \mathbf{m}(x) + O(1)$. L.A. Levin, [5], conjectured that the analogue would hold for the unrestricted continuous version. But it has been shown, [3], that

$$\sup_{x \in \mathcal{B}^*} |KM(x) - Km(x)| = \infty,$$

There it is shown that the exact relation is (for each particular choice of basis $\mathcal{B}$ such as $\mathcal{B} = \mathcal{N}$, the natural numbers, or $\mathcal{B} = \{0, 1\}$)

Lemma 9.

$$KM(x) \leq Km(x) \leq KM(x) + Km(l(x)) + O(1). \tag{4}$$

This shows that the differences between $Km(x)$ and $KM(x)$ must in some sense be very small. The next question to ask is whether the quantities involved are usually different, or whether this is a rare occurrence. In other words, whether for *a priori* almost all infinite sequences x, the difference between Km and KM is bounded by a constant. The following facts have been proven, [3].

Lemma 10. *(i) For random strings $x \in \mathcal{B}^*$ we have $Km(x) - KM(x) = O(1)$.*

(ii) There exists a function $f(n)$ which goes to infinity with $n \to \infty$ such that $Km(x) - KM(x) \geq f(l(x))$, for infinitely many x. If x is a finite binary string ($\mathcal{B} = \{0,1\}$), then we can choose $f(n)$ as the inverse of some version of Ackermann's function

An infinite binary sequence ω is μ-random iff

$$\sup_n \mathbf{M}(\omega_1 \dots \omega_n)/\mu(\omega_1 \dots \omega_n) < \infty,$$

and the set of μ-random sequences has μ-measure one, see [6], Chapter 4. Let ω be a μ-random infinite binary sequence and xy be a finite prefix of ω. For $l(x)$ grows unboundedly with $l(y)$ fixed, we have by Theorem 6

$$\lim_{l(x)\to\infty} \log \mu(y|x) - \log \mathbf{M}(y|x) = 0. \tag{5}$$

Therefore, if x and y satisfy above conditions, then maximizing $\mu(y|x)$ over y means minimizing $-\log \mathbf{M}(y|x)$. It is shown in Lemma 10 that $-\log \mathbf{M}(x)$ is slightly smaller than $Km(x)$, the length of the shortest program for x on the reference universal monotonic machine. For binary programs this difference is very small, Lemma 9, but can be unboundedly in the length of x.

Together this shows the following. Given xy that is a prefix of a (possibly not μ-random) ω, optimal prediction of fixed length extrapolation y from an unboundedly growing prefix x of ω need not necessarily be reached by the shortest programs for xy and x minimizing $Km(xy) - Km(x)$, but is reached by considering the weighted version of all programs for xy and x which is represented by

$$-\log \mathbf{M}(xy) + \log \mathbf{M}(x) = (Km(xy) - g(xy)) - (Km(x) - g(x)).$$

Here $g(x)$ is a function which can rise to in between the inverse of the Ackermann function and $Km(l(x)) \leq \log\log x$—but only in case x is not μ-random.

Therefore, for certain x and y which are *not* μ-random, optimization using the minimum length programs may result in very incorrect predictions. However, for μ-random x we have that $-\log \mathbf{M}(x)$ and $Km(x)$ coincide up to an additional constant independent of x, that is, $g(xy) = g(x) = O(1)$, Lemma 10. Hence, together with Equation 5, we find the following.

Theorem 11. *Let μ be a recursive semimeasure, and let ω be a μ-random infinite binary sequence and xy be a finite prefix of ω. For $l(x)$ grows unboundedly and $l(y)$ fixed,*

$$\lim_{l(x)\to\infty} -\log \mu(y|x) = Km(xy) - Km(x) \pm O(1) < \infty,$$

where $Km(xy)$ and $Km(x)$ grow unboundedly.

By its definition Km is monotone in the sense that always $Km(xy) - Km(x) \geq 0$. The closer this difference is to zero, the better the shortest effective monotone program for x is also a shortest effective monotone program for xy and hence predicts y given x. Therefore, for all large enough μ-random x, predicting by determining y which minimizes the difference of the minimum program lengths of xy and x gives a good prediction. Here y should be preferably large enough to eliminate the influence of the $O(1)$ term.

Corollary 12 Prediction by Data Compression. *Assume the conditions of Theorem 11. With μ-probability going to one as $l(x)$ grows unboundedly, a fixed-length y extrapolation from x maximizes $\mu(y|x)$ iff y can be maximally compressed with respect to x in the sense that it minimizes $Km(xy) - Km(x)$. That is, y is the string that minimizes the length difference between the shortest program that outputs $xy \ldots$ and the shortest program that outputs $x \ldots$.*

7.1 Hypothesis Identification and Compression

We briefly mention the related work on hypothesis identification and compression. The so-called minimum description length principle is an algorithmic paradigm that is widely applied. That is, it is widely applied at least in spirit; to apply it literally may run in computation difficulties since it involves finding an optimum in a exponentially large set of candidates as noted for example in [6]. Yet in some cases one can approximate this optimum, [18, 21]. For the theoretical case where the minimum description lengths involved are the Kolmogorov complexities, we mathematically derived the minimum description length paradigm from first principles, that is, Bayes's rule, [6, 7, 8]. To do so we needed auxiliary notions of *universal distribution* and *Martin-Löf tests* for *randomness of individual objects*.

Before proceeding it is useful to point out that the idea of a two-part code for a body of data D is natural from the perspective of Kolmogorov complexity. If D does not contain any regularities at all, then it consists of purely random data and there is no hypothesis to identify. Assume that the body of data D contains regularities. With help of a description of those regularities (a model) we can describe the data compactly. Assuming that the regularities can be represented in an effective manner (that is, by a Turing machine), we encode the data as a program for that machine. Squeezing all effective regularity out of the data, we end up with a Turing machine representing the meaningful regular information in the data together with a program for that Turing machine representing the remaining meaningless randomness of the data. This is the intuition, which finds its basis in the Definitions 1 and 2. However, it is difficult to find a valid mathematical way to force a sensible division of the information at hand in a meaningful part and a meaningless part. One way to proceed is suggested by the analysis below.

A practice oriented theory like MDL, although often lacking in justification, apparently works and is used by practitioners. The MDL principle is very easy to use in some loose sense, but it is hard to justify. A user of the MDL principle does not need to *prove* that the concept class concerned is learnable, rather he needs to choose a concept that can be described shortly and without causing too many errors (and he needs to balance these two things).

In various forms aimed at practical applications the idea of doing induction or data modelling in statistical hypothesis identification or prediction was proposed by C. Wallace and co-authors [19, 20], who formulated the *Minimum Message Length (MML)* principle and J. Rissanen [10, 11] who formulated the *Minimum Description Length (MDL)* principle. Here we abstract away from epistemological and technical differences between MML and MDL, and other variants, and their concessions to reality in the name of feasibility and practicability. We focus only on the following central ideal version involved. Indeed, we do not even care about whether we deal with statistical or deterministic hypotheses. All effectively describable hypotheses are involved.

Definition 13. Given a sample of data, and an effective enumeration of models, *ideal MDL* selects the model which minimizes the sum of

- the length, in bits, of an effective description of the model; and
- the length, in bits, of an effective description of the data when encoded with the help of the model.

Intuitively, 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 number of 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. Similarly to the analysis of prediction above, in [7, 8] we have shown that in almost all cases maximal compression finds the best hypotehsis.

8 Conclusion

The analysis of both hypothesis identification by Ideal MDL and prediction shows that maximally compressed descriptions give good results on the data samples which are random with respect to probabilistic hypotheses. These data samples form the overwhelming majority and occur with probability going to one when the length of the data sample grows unboundedly.

References

1. J.L. Doob, *Stochastic Processes* , Wiley, 1953.
2. P. Gács, On the symmetry of algorithmic information, *Soviet Math. Dokl.*, 15 (1974) 1477-1480. Correction: ibid., 15 (1974) 1480.
3. P. Gács, On the relation between descriptional complexity and algorithmic probability, *Theoret. Comput. Sci.*, 22(1983), 71-93.

4. A.N. Kolmogorov, Three approaches to the quantitative definition of information, *Problems Inform. Transmission* 1:1 (1965) 1-7.
5. L.A. Levin, On the notion of a random sequence, *Soviet Math. Dokl.*, 14(1973), 1413-1416.
6. M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, New York, 1993.
7. M. Li and P.M.B. Vitanyi, Computational Machine Learning in Theory and Praxis. In: 'Computer Science Today', J. van Leeuwen, Ed., Lecture Notes in Computer Science, Vol. 1000, Springer-Verlag, Heidelberg, 1995, 518-535.
8. P.M.B. Vitanyi and M. Li, Ideal MDL and Its Relation To Bayesianism, 'Proc. ISIS: Information, Statistics and Induction in Science', World Scientific, Singapore, 1996, 282-291.
9. P. Martin-Löf, The definition of random sequences, *Inform. Contr.*, 9(1966), 602-619.
10. J.J. Rissanen, Modeling by the shortest data description, *Automatica-J.IFAC* 14 (1978) 465-471.
11. J.J. Rissanen, *Stochastic Complexity and Statistical Inquiry*, World Scientific Publishers, 1989.
12. J.J. Rissanen, Fisher information and stochastic complexity, *IEEE Trans. Inform. Theory*, IT-42:1(1996), 40-47.
13. J. Segen, *Pattern-Directed Signal Analysis*, PhD Thesis, Carnegie-Mellon University, Pittsburgh, 1980.
14. R.J. Solomonoff, A formal theory of inductive inference, Part 1 and Part 2, *Inform. Contr.*, 7(1964), 1-22, 224-254.
15. R.J. Solomonoff, Complexity-based induction systems: comparisons and convergence theorems, *IEEE Trans. Inform. Theory* IT-24 (1978) 422-432.
16. A.M. Turing, On computable numbers with an application to the Entscheidungsproblem, *Proc. London Math. Soc., Ser. 2*, 42(1936), 230-265; Correction, Ibid, 43(1937), 544-546.
17. R. von Mises, Grundlagen der Wahrscheinlichkeitsrechnung, *Mathemat. Zeitsch.*, 5(1919), 52-99.
18. V. Vovk, Minimum description length estimators under the universal coding scheme, in: P. Vitányi (Ed.), *Computational Learning Theory*, Proc. 2nd European Conf. (EuroCOLT '95), Lecture Notes in Artificial Intelligence, Vol. 904, Springer-Verlag, Heidelberg, 1995, pp. 237-251; Learning about the parameter of the Bernoulli model, *J. Comput. System Sci.*, to appear.
19. C.S. Wallace and D.M. Boulton, An information measure for classification, *Computing Journal* 11 (1968) 185-195.
20. C.S. Wallace and P.R. Freeman, Estimation and inference by compact coding, *J. Royal Stat. Soc., Series B*, 49 (1987) 240-251. Discussion: ibid.,252-265.
21. K. Yamanishi, A Randomized Approximation of the MDL for Stochastic Models with Hidden Variables, *Proc. 9th ACM Comput. Learning Conference*, ACM Press, 1996.
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 (1970) 83-124.