Kolmogorov's Structure Functions with an Application to the Foundations of Model Selection

Nikolai Vereshchagin* Paul Vitányi†

Abstract

In 1974 Kolmogorov proposed a non-probabilistic approach to statistics, an individual combinatorial relation between the data and its model. We vindicate, for the first time, the rightness of the original "structure function", proposed by Kolmogorov: minimizing the data-to-model code length (finding the ML estimator or MDL estimator), in a class of contemplated models of prescribed maximal (Kolmogorov) complexity, always *results in a model of best fit (expressed as minimal randomness deficiency). We show that both the structure function and the minimum randomness deficiency function can assume all shapes over their full domain (improving an old result of L.A. Levin and both an old and a recent one of V.V. Vyugin). We determine the (un)computability properties of the various functions and "algorithmic sufficient statistic."*

1 Introduction

As perhaps the last mathematical innovation of an extraordinary scientific career, Kolmogorov [15, 14] proposed to found statistical theory on finite combinatorial principles independent of probabilistic assumptions. The relation between the individual data and its explanation (model) is expressed by Kolmogorov's structure function. This function, its variations and its relation to model selection, have obtained some notoriety but have not before been comprehensively analyzed and understood [18, 2, 22, 5, 12, 19, 23, 9, 11, 8, 3]. It has always been questioned why Kolmogorov chose to focus on the the mysterious function h_x below, rather than on the more evident β_x variant below. Our main result, with the beauty of truth, justifies Kolmogorov's intuition. One easily stated consequence is: For all data, minimizing a two-part code consisting of one part model description and one part data-to-model code (essentially the celebrated MDL code [17]), subject to a given model-complexity constraint, as well as minimizing the one-part code consisting of just the data-to-model code (essentially the maximum likelihood estimator), *in every case* (and not only with high probability) selects a model that is a "best explanation" (within $O(\log |x|)$ accuracy) of the data within the given model-complexity constraint. This notion of "best explanation" ("best fit") is understood in the sense that the data is "most typical" for the selected model in a rigorous mathematical sense that is discussed below: the lack of typicality of x for the model S is measured by the randomness deficiency (3). A practical consequence is as follows: we can *approximate* within $O(\log |x|)$ accuracy the best fitting model. This means the existence of a non-halting algorithm A that given any x, α outputs a finite sequence $p_1, p_2, p_3, \ldots, p_i$ of pairwise different computer programs each of length at most $\alpha + C \log |x|$ such that the last program p_i prints a model S_i which is best in the following sense. There is no program p of length at most α that prints a model S such that the randomness deficiency of x for S is $C \log |x|$ less than that of x for S_i (C is a constant). Note that we are not able to find p_i given x, α, since the algorithm A is non-halting and thus we do not know which program will be output last.

In Kolmogorov's initial proposal, as in this work, models are finite sets of finite binary strings, and the data is one of the strings (all discrete data can be binary encoded). The restriction to finite set models is just a matter of convenience: the main results generalize to the case where the models are arbitrary computable probability density functions, [18], and to the model class consisting of arbitrary total recursive functions, [21], see below.

In probabilistic statistics the goodness of a selection process is measured in terms of expectations over probabilistic ensembles. For current applications, average relations are often irrelevant, since the part of the support of the probability density function that will ever be observed has about zero measure. This may be the case in, for example, complex video and sound analysis. There arises the problem that for individual cases the selection performance may be bad although the performance is good on average, or vice versa. There is also the problem of what probability means, whether it is subjective, objective, or exists at all. Kolmogorov's proposal strives for the firmer and less contentious ground of finite combinatorics and effective computation.

Kolmogorov Complexity and Prefix Codes: Informally, the Kolmogorov complexity, or algorithmic entropy, $K(x)$ of a string x is the length (number of bits) of a shortest binary program (string) to compute x on a fixed reference universal computer (such as a particular universal Turing machine). Intuitively, $K(x)$

*Supported in part by Grant 01-01-01028 from Russian Federation Basic Research Fund. Work done while visiting CWI. Adress: Department of Mathematical Logic and Theory of Algorithms, Faculty of Mechanics and Mathematics, Moscow State University, Leninskie Gory, Moscow, Russia 119992. Email: `ver@mech.math.msu.su`.

†Supported in part by the EU fifth framework project QAIP, IST–1999–11234, the NoE QUIPROCONE IST–1999–29064, the ESF QiT Programmme, and the EU Fourth Framework BRA NeuroCOLT II Working Group EP 27150. Address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: `Paul.Vitanyi@cwi.nl`.

represents the minimal amount of information required to generate x by any effective process, [13]. The conditional Kolmogorov complexity $K(x|y)$ of x relative to y is defined similarly as the length of a shortest program to compute x, if y is furnished as an auxiliary input to the computation. Formally, we consider a special type of Turing machine with a one-way input tape, a separate work tape, and a one-way output tape. Such Turing machines are called *prefix* Turing machines. If a machine T halts with output x after having scanned all of p on the input tape, but not further, then $T(p) = x$ and we call p a *program* for T and a *self-delimiting* code word for x. It is easy to see that $\{p : T(p) = x, x \in \{0,1\}^*\}$ is a *prefix code*. Hence,

$$\sum_x 2^{-K(x)} \leqslant 1. \tag{1}$$

For precise definitions, notation, and results see the text [12]. The functions $K(\cdot)$ and $K(\cdot|\cdot)$, though defined in terms of a particular machine model, are machine-independent up to an additive constant and acquire an asymptotically universal and absolute character through Church's thesis, from the ability of universal machines to simulate one another and execute any effective process. To express the next property, needed later, we introduce some notation: Let $K(x, y)$ denote the Kolmogorov complexity of a string $[x, y]$, encoding the pair (x, y) using some fixed one-to-one correspondence between pairs of strings and strings; let furthermore x^* stand for $[x, K(x)]$, the encoding of the pair $(x, K(x))$. The celebrated *symmetry of information* property is:

$$\begin{aligned} K(x, y) &= K(x) + K(y|x^*) + O(1) \\ &= K(y) + K(x|y^*) + O(1). \end{aligned} \tag{2}$$

Use of additive constants: It is customary in this area to use "additive constant c" or equivalently "additive $O(1)$ term" to mean a constant, accounting for the length of a fixed binary program, independent from every variable or parameter in the expression in which it occurs.

Kolmogorov structure functions: We will prove that there is a close relation between functions describing three, a priori seemingly unrelated, aspects of modeling individual data. We first need a definition. Denote the *complexity of the finite set* S by $K(S)$—the length (number of bits) in the shortest binary program p from which the reference universal prefix machine U computes a listing of the elements of S and then halts. That is, if $S = \{x_1, \ldots, x_n\}$, then $U(p) = [x_1, [x_2, \ldots, [x_{n-1}, x_n] \ldots]]$.

For every finite set $S \subseteq \{0,1\}^*$ containing x we have $K(x|S) \leqslant \log|S| + O(1)$. Indeed, consider the selfdelimiting code of x consisting of its $\lceil \log|S| \rceil$ bit long index of x in the lexicographical ordering of S. This code is called *data-to-model code*. The lack of typicality of x with respect to S is the amount by which $K(x|S)$ falls short of the length of the data-to-model code. The *randomness deficiency* of x in S is defined by

$$\delta(x|S) = \log|S| - K(x|S), \tag{3}$$

for $x \in S$, and ∞ otherwise. The *minimal randomness deficiency* function is

$$\beta_x(\alpha) = \min_S \{\delta(x|S) : S \ni x,\ K(S) \leqslant \alpha\}, \tag{4}$$

where we set $\min \emptyset = \infty$. If $\delta(x|S)$ is small, then x may be considered as a *typical* member of S. This means that S is a "best" model for x—a most likely explanation. There are no simple special properties that single it out from the majority of elements in S. This is not just terminology: If $\delta(x|S)$ is small, then x satisfies *all* properties of low Kolmogorov complexity that hold with high probability for the elements of S. To be precise: Consider strings of length n and let S be a subset of such strings.

(i) If P is a property satisfied by all x with $\delta(x|S) \leqslant \delta(n)$, then P holds with probability at least $1 - 1/2^{\delta(n)}$ for the elements of S.

(ii) Let P be any property that holds with probability at least $1 - 1/2^{\delta(n)}$ for the elements of S. Then, every such P holds simultaneously for every $x \in S$ with $\delta(x|S) \leqslant \delta(n) - K(P|n) - O(1)$.

Kolmogorov [15, 14] proposed the following function: The *Kolmogorov structure* function h_x of given data x is defined by

$$h_x(\alpha) = \min_S \{\log|S| : S \ni x,\ K(S) \leqslant \alpha\}, \tag{5}$$

where $S \ni x$ is a contemplated model for x, and α is a nonnegative integer value bounding the complexity of the contemplated S's. Clearly, the Kolmogorov structure function is non-increasing and reaches $\log|\{x\}| = 0$ for $\alpha = K(x) + c_1$ where c_1 is the number of bits required to change x into $\{x\}$.

For every $S \ni x$ we have

$$K(x) \leqslant K(S) + \log|S| + O(1). \tag{6}$$

Indeed, consider the following *two-part code* for x: the first part is a shortest self-delimiting program p of S and the second part is $\lceil \log|S| \rceil$ bit long index of x in the lexicographical ordering of S. Since S determines $\log|S|$ this code is self-delimiting and we obtain (6) where the constant $O(1)$ is the length of the program to reconstruct x from its two-part code. We thus conclude that $K(x) \leqslant \alpha + h_x(\alpha) + O(1)$, that is, the function $h_x(\alpha)$ never decreases more than a fixed independent constant below the diagonal *sufficiency line* L defined by $L(\alpha) + \alpha = K(x)$, which is a lower bound on $h_x(\alpha)$ and is approached to within a constant distance by the graph of h_x for certain α's (for instance, for $\alpha = K(x) + c_1$). For these α's we thus have $\alpha + h_x(\alpha) = K(x) + O(1)$; a model corresponding to such an α (witness for $h_x(\alpha)$) is called a *sufficient statistic*, and it is *minimal* for the least such α. This suggestive nomenclature is explained in [5, 9].

The length of the minimal two-part code for x consisting of the model cost $K(S)$ and the length of the index of x in S, the complexity of S upper bounded by α, is given by the *MDL function*:

$$\lambda_x(\alpha) = \min_S \{\Lambda(S) : S \ni x,\ K(S) \leqslant \alpha\}, \tag{7}$$

where $\Lambda(S) = \log|S| + K(S) \geqslant K(x) - O(1)$ is the total length of two-part code of x with help of model S. Apart from being convenient for the technical analysis in this work, $\lambda_x(\alpha)$ is the celebrated two-part Minimum Description Length code length (Example 3) with the model-code length restricted to at most α.

EXAMPLE 1. **Lossy Compression:** The function $\beta_x(\alpha)$ is relevant to lossy compression (used, for instance, to compress images). Assume we need to compress x to α bits where $\alpha \ll K(x)$. Of course this implies some loss of information present in x. One way to select redundant information to discard is as follows: Find a set $S \ni x$ with $K(S) \leqslant \alpha$ and with small $\delta(x|S)$, and consider the compressed version S' of S. To reconstruct an x', a decompresser uncompresses S' to S and selects at random an element x' of S. Since $\delta(x|S)$ is small, he will most likely obtain an x' that is effectively indistinguishable from x (use items (i) and (ii) above). Hence, x' serves the purpose of the message x as well as does x itself. Let us look at an example. To transmit a picture of "rain" through a channel with limited capacity α, one can transmit the indication that this is a picture of the rain and the particular drops may be chosen by the receiver at random. In this interpretation, $\beta_x(\alpha)$ indicates how "random" or "typical" x is with respect to the best model at complexity level α—and hence how "indistinguishable" from the original x the randomly reconstructed x' can be expected to be. $\diamond$

Background and Related Work: A.N. Kolmogorov [14, 15] proposed the combinatorial non-probabilistic approach to an individual data-to-model relation, two-part codes separating the *structure* or *meaningful information* of a string from meaningless *accidental* features. There is no written version, apart from a few lines [14] which we reproduce in Example 5, so we have to rely on oral history of witnesses [8, 3, 11], and, says T. Cover [3]: "I remember taking many long hours trying to understand the motivation of Kolmogorov's approach." According to Peter Gács, [8]: "Kolmogorov drew a picture of $h_x(\alpha)$ as a function of α monotonically approaching the diagonal [sufficiency line]. Kolmogorov stated that it was known (proved by L.A. Levin) that in some cases it remained far from the diagonal line till the very end." L.A. Levin [11]: "Kolmogorov told me [about] $h_x(i)$ (or its inverse, I am not sure) and asked how this $h(i)$ could behave. I proved that $i + h(i) + O(\log i)$ is monotone but otherwise arbitrary within $O(\sqrt{i})$ accuracy; it stabilizes on $K(x)$ when i exceeds $I(x : \text{Halting})$. (Actually, this expression for accuracy was Kolmogorov's re-wording, I gave it in less elegant but equivalent terms—$O(p \log i)$ where p is the number of "jumps".) I do not remember Kolmogorov defining $\beta_x(i)$ or suggesting anything like your result. I never published anything on the topic because I do not believe strings x with significant $I(x : \text{Halting})$ could exist in the world." Related work on so-called "non-stochastic objects" (where $h_x(\alpha) + \alpha$ drops to $K(x)$ only for large α) is [18, 22, 19, 20, 21]. In 1987, [22, 23], V.V. V'yugin established that the randomness deficiency function $\beta_x(\alpha)$ can assume all possible shapes, (within the obvious constraints) but only for $\alpha = o(|x|)$. In the survey [4] of Kolmogorov's work in information theory, the authors preferred to mention $\beta_x(\alpha)$, because it by definition optimizes "best fit," rather than $h_x(\alpha)$ of which the usefulness and meaningfulness was mysterious. But Kolmogorov had a seldom erring intuition: we will show that his original proposal h_x in the proper sense incorporates all desirable properties of $\beta_x(\alpha)$, and in fact is superior. In [2, 5, 4] a notion of "algorithmic sufficient statistics", derived from Kolmogorov's structure function, is suggested as the algorithmic approach to the probabilistic notion of sufficient statistic [6, 5] that is central in classical statistics. The paper [9] investigates the algorithmic notion in detail and formally establishes such a relation. The algorithmic (minimal) sufficient statistic is related in [20, 10] to the "minimum description length" principle [17, 1, 25] in statistics and inductive reasoning. Moreover, [9] observed that $\beta_x(\alpha) \leqslant h_x(\alpha) + \alpha - K(x) + O(1)$, establishing a one-sided relation between (4) and (5), and the question was raised whether the converse holds.

This Work: The most fundamental result in this paper is the equality

$$\beta_x(\alpha) = h_x(\alpha) + \alpha - K(x) = \lambda_x(\alpha) - K(x) \tag{8}$$

which holds within logarithmic additive terms in argument and value. Additionally, every set S that witnesses the value $h_x(\alpha)$ (or $\lambda_x(\alpha)$), also witnesses the value $\beta_x(\alpha)$ (but not vice versa). It is easy to see that $h_x(\alpha)$ and $\lambda_x(\alpha)$ are upper semi-computable (Definition 2); but we show that $\beta_x(\alpha)$ is neither upper nor lower semi-computable. A priori there is no reason to suppose that a set that witnesses $h_x(\alpha)$ (or $\lambda_x(\alpha)$) also witnesses $\beta_x(\alpha)$, for *every* α. But the fact that they do, vindicates Kolmogorov's original proposal and establishes h_x's pre-eminence over β_x. The result can be taken as a foundation and justification of common statistical principles in model selection such as maximum likelihood or MDL ([17, 1] and our Examples 3 and 4). The possible (coarse) shapes of the functions λ_x, h_x and β_x are examined in Section 2. Roughly stated: The structure functions λ_x, h_x and β_x can assume all possible shapes over their full domain of definition (up to additive logarithmic precision in both argument and value). This improves and extends V'yugin's result [22, 23] above; it also improves the independent related result of L.A. Levin [11] above; and, applied to "snooping curves" extends a recent result of V'yugin, [24], in Example 2. Computability properties are examined in Section 3, incidentally proving a to our knowledge first natural example, β_x, of a function that is not semi-computable but computable with an oracle for the halting problem. Because of page limitations we omit most proofs and various results concerning realization, fine shape, and details about sufficient statistic(s). The full paper is available from http://xxx.lanl.gov/abs/cs.CC/0204037 .

2 Coarse Structure

Let $\beta_x(\alpha)$ be defined as in (4) and $h_x(\alpha)$ be defined as in (5). Both functions are 0 ($\beta_x(\alpha)$ may be $-O(1)$) for all $\alpha \geqslant K(x)+c_0$ where c_0 is a constant. We *represent* the coarse shape of these functions for different x by functions characteristic of that shape. Informally, g represents f means that the graph of f is contained in a strip of logarithmic (in the length n of x) width centered on the graph of g, Figure 1. For formal statements we rely on the notion in Definition 1. Informally, we obtain the following results (x is of length n and complexity $K(x) = k$):

- Every non-increasing function β represents β_x for some x, and for every x the function β_x is represented by some β, provided $\beta(k) = 0$, $\beta(0) \leqslant n - k$.
- Every function h, with non-increasing $h(\alpha) + \alpha$, represents h_x for some x, and for every x the function h_x is represented by some h as above, provided $h(k) = 0$, $h(0) \leqslant n$ (and by the non-increasing property $h(0) \geqslant k$).
- $h_x(\alpha) + \alpha$ represents $\beta_x(\alpha) + k$, and conversely, for every x.
- For every x and α, every minimal size set $S \ni x$ of complexity at most $\alpha' = \alpha + O(\log n)$, has randomness deficiency $\delta(x|S) \leqslant \beta_x(\alpha) + O(\log n)$.

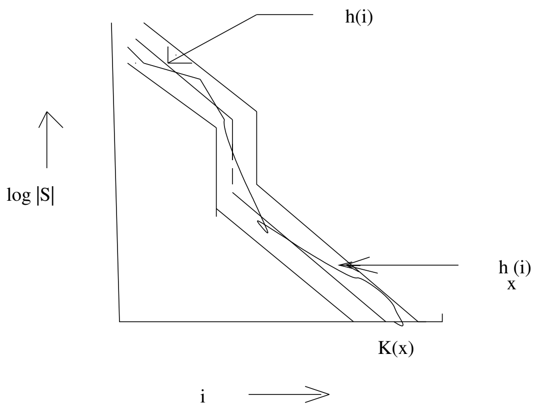

Figure 1. Structure function $h_x(i)$ in strip determined by $h(i)$, that is, $h_x(i) = \mathcal{E}(h(i))$.

To provide precise statements we need a definition.

DEFINITION 1. Let f, g be functions defined on $\{0, 1, \ldots, k\}$ with values in $\mathbb{N} \cup \{\infty\}$. We say that f is $(\varepsilon(i), \delta(i))$-close to g (in symbols: $f = \mathcal{E}(g)$) if

$$f(i) \geqslant \min\{g(j) \colon j \in [\varepsilon(0), k],\ |j - i| \leqslant \varepsilon(i)\} - \delta(i),$$
$$f(i) \leqslant \max\{g(j) \colon j \in [\varepsilon(0), k],\ |j - i| \leqslant \varepsilon(i)\} + \delta(i)$$

for every $i \in [\varepsilon(0), k]$. If $f = \mathcal{E}(g)$ and $g = \mathcal{E}(f)$ we write $f \cong g$.

Here $\varepsilon(i), \delta(i)$ are small values like $O(\log n)$ when we consider data x of length n. Note that this definition is not symmetric and allows $f(i)$ to have arbitrary values for $i \in [0, \varepsilon(0))$. However, it is transitive in the following sense: if f is $(\varepsilon_1(i), \delta_1(i))$-close to g and g is $(\varepsilon_2(i), \delta_2(i))$-close to h then f is $(\varepsilon_1(i) + \varepsilon_2(i), \delta_1(i) + \delta_2(i))$-close to h. If $f = \mathcal{E}(g)$ and g is linear continuous, meaning that $|g(i) - g(j)| \leqslant c|i - j|$ for some constant c, then the difference between $f(i)$ and $g(i)$ is bounded by $c\varepsilon(i) + \delta(i)$ for every $\varepsilon(0) \leqslant i \leqslant k$.

This notion of closeness, if applied unrestricted, is not always meaningful. For example, take as g the function taking value n for all even $i \in [0, k]$ and 0 for all odd $i \in [0, k]$. Then for *every* function f on $[0, k]$ with $f(i) \in [0, n]$ we have $f = \mathcal{E}(g)$ for $\varepsilon = 1$, $\delta = 0$. But if $f = \mathcal{E}(g)$ and g is *non-increasing* then g indeed gives much information about f.

It is instructive to consider the following example. Let $g(i)$ be equal to $2k - i$ for $i = 0, 1, \ldots, \frac{k}{2} - 1$ and to $k - i$ for $i = \frac{k}{2}, \ldots, k$. Let $\varepsilon(i) = \delta(i)$ be constant. Then a function $f = \mathcal{E}(g)$ may take every value for $i \in [0, \delta)$, every value in $[2k - i - 2\delta, 2k - i + 2\delta]$ for $i \in [\delta, \frac{k}{2} - \delta]$, every value in $[k - i - \delta, 2k - i + \delta]$ for $i \in (\frac{k}{2} - \delta, \frac{k}{2} + \delta]$, and every value in $[k - i - 2\delta, k - i + 2\delta]$ for $i \in (\frac{k}{2} + \delta, k]$ (see Figure 1). Thus the point $\frac{k}{2}$ of discontinuity of g gives an interval of size 2δ of large ambiguity of f. Loosely speaking the graph of f can be any line contained in the strip of radius 2δ whose middle line is the graph of g.

The definition of λ_x immediately implies the following properties: $\lambda_x(\alpha)$ is non-increasing, $\lambda_x(\alpha) \geqslant K(x) - O(1)$ for all α.

The next lemma shows that properties of λ_x translate directly into properties of h_x since $h_x(\alpha)$ is always "close" to $\lambda_x(\alpha) - \alpha$.

LEMMA 1. *For every x we have $\lambda_x(\alpha) \leqslant h_x(\alpha) + \alpha \leqslant \lambda_x(\alpha) + K(\alpha) + O(1)$ for all $\alpha \leqslant K(x)$. Hence $\lambda_x(\alpha) \cong h_x(\alpha) + \alpha$ for $\varepsilon = 0$, $\delta = K(\alpha) + O(1)$.*

PROOF. (Sketch.) The inequality $\lambda_x(\alpha) \leqslant h_x(\alpha) + \alpha$ is immediate. So it suffices to prove that $h_x(\alpha) + \alpha \leqslant \lambda_x(\alpha) + K(\alpha) + O(1)$.

Let S witness $\lambda_x(\alpha)$. If $K(S) = \alpha$ then we have $h_x(\alpha) \leqslant \log |S| = \lambda_x(\alpha) - \alpha$. Assume that $K(S) < \alpha$. Partition S into 2^m equal-size parts and select the part S' containing x. Then, $\log |S'| = \log |S| - m$ at the cost of increasing the complexity of S' to

$$K(S') \leqslant K(S) + m + K(m|K(S)) + O(1)$$

(we specify the part S' containing x by its index among all the parts). Choosing

$$m = \alpha - K(S) - K(\alpha|K(S)) - c$$

for appropriate constant c we obtain $K(S') \leqslant \alpha$ and $\log |S'| = \log |S| - m = \lambda(\alpha) - K(S) - m = \lambda(\alpha) - \alpha + K(\alpha|K(S)) + c \leqslant \lambda(\alpha) - \alpha + K(\alpha) + O(1)$. □

The lemma implies that the *same set* witnessing $h_x(\alpha)$ also witnesses $\lambda_x(\alpha)$ up to an additive term of $K(\alpha)$. The converse is only true for the smallest cardinality set witnessing $\lambda_x(\alpha)$. Without this restriction a counter example is: for random $x \in \{0, 1\}^n$ (that is, $K(x) \geqslant n$) the set $S = \{0, 1\}^n$ witnesses $\lambda_x(\frac{n}{2}) = n + O(K(n))$ but does not witness $h_x(\frac{n}{2}) = \frac{n}{2} + O(K(n))$.

The next two theorems state the main results of this work in a precise form. By $K(i, n, \lambda)$ we mean the minimum length of a program that outputs n, i, and computes $\lambda(j)$ given any j in the domain of λ.

THEOREM 1. *(i) For every n and every string x of length n and complexity k there is an integer valued non-increasing function λ defined on $[0, k]$ such that $\lambda(0) \leqslant n$, $\lambda(k) = k$ and $\lambda_x = \mathcal{E}(\lambda)$ for $\varepsilon = \delta = K(n) + O(1)$.*

(ii) Conversely, for every n and non-increasing integer valued function λ whose domain includes $[0, k]$ and such that $\lambda(0) \leqslant n$ and $\lambda(k) = k$, there is x of length n and complexity $k \pm (K(k, n, \lambda) + O(1))$ such that $\lambda_x = \mathcal{E}(\lambda)$ for $\varepsilon = \delta = K(i, n, \lambda) + O(1)$.

PROOF. (i) We first observe that for every x of length n we have $\lambda_x(K(n) + O(1)) \leqslant n + O(1)$, as witnessed by $S = \{0, 1\}^n$. At the other extreme, $\lambda_x(k + O(1)) = k + O(1)$, as witnessed by $S = \{x\}$.

Define $\lambda(i)$ by the equation $\lambda(i) - k = \max\{0, \lambda_x(i + K(n) + O(1)) - k - O(1)\}$. Then $\lambda_x = \mathcal{E}(\lambda)$ with $\varepsilon = \delta = K(n) + O(1)$, and λ satisfies the requirements of Item (i) of the theorem.

(ii) Fix $\lambda(i)$ satisfying the conditions in the theorem. It suffices to show that there is a string x of length n such that, for every $i \in [0, k]$, we have $\lambda_x(i) \geqslant \lambda(i)$ and $\lambda_x(i + \delta(i)) \leqslant \lambda(i) + \delta(i)$ for $\delta(i) = K(i, n, \lambda) + O(1)$. Then, with $\delta = \delta(k)$, we have $K(x) \leqslant \lambda_x(k + \delta) + O(1) \leqslant \lambda(k) + \delta + O(1) = k + \delta + O(1)$. And the inequality $\lambda_x(k) \geqslant \lambda(k) = k$ implies that $K(x) > k - O(1)$.

CLAIM 1. For every length n, there is a string x of length n such that $\lambda_x(i) \geqslant \lambda(i)$ for every i in the domain of λ.

PROOF. Fix a length n. If $\lambda_x(i) < \lambda(i)$ then x belongs to a set A with $\Lambda(A) < \lambda(i) \leqslant \lambda(0) \leqslant n$. The total number of elements in different such A's is less than $\sum_A 2^{n-K(A)} = 2^n \sum_A 2^{-K(A)} \leqslant 2^n$, where the second inequality follows by (1). □

We prove Item (ii) by demonstrating that the lexicographically first x, as defined in Claim 1, also satisfies $\lambda_x(i+\delta(i)) \leqslant \lambda(i) + \delta(i)$, for $\delta(i) = K(i,n,\lambda) + O(1)$ for all $i \in [0,k]$. It suffices to construct a set $S \ni x$ of cardinality $2^{\lambda(i)-i}$ and of complexity at most $i + \delta(i)$, for every $i \in [0,k]$.

For every fixed $i \in [0,k]$ we can run the following:

Algorithm: Let A be a set variable initially containing all strings of length n, and let S be a set variable initially containing the $2^{\lambda(i)-i}$ first strings of A in lexicographical order. Run all programs of length at most n dovetail style. Every time a program p of some length j halts, $\lambda(j)$ is defined, and p prints a set B of cardinality at most $2^{\lambda(j)-j}$, we remove all the elements of B from A (but not from S); we call a step at which this happens a *j-step*. Every time $S \cap A$ becomes empty at a j-step, we replace the contents of S by the set of the $2^{\lambda(i)-i}$ first strings in lexicographical order of (the current contents of) A. Possibly, the last replacement of S is incomplete because there are less than $2^{\lambda(i)-i}$ elements left in A. It is easy to see that $x \in S \setminus A$ just after the final replacement, and stays there forever after, even though some programs in the dovetailing process may still be running and elements from A may still be eliminated.

CLAIM 2. The contents of the set S is replaced at most 2^{i+1} times.

PROOF. There are two types of replacements that will be treated separately.

Case 1: Replacement of the current contents of S where at some j-step with $j \leqslant i$ *at least one* element was removed from the current contents $S \cap A$. Trivially, the number of this type of replacements is bounded by the number of j-steps with $j < i$, and hence by the number of programs of length less than i, that is, by 2^i.

Case 2: Replacement of the current contents of S where *every one* of the $2^{\lambda(i)-i}$ elements of the current contents of S is removed from A by j-steps with $j \geqslant i$. Let us estimate the number of this type of replacements: Every element x removed at a j-step with $j \geqslant i$ belongs to a set B with $\Lambda(B) \leqslant \lambda(j) \leqslant \lambda(i)$. The overall cumulative number of elements removed from A on j-steps with $j \geqslant i$ is bounded by $\sum_B 2^{\lambda(i)-K(B)} \leqslant 2^{\lambda(i)}$, where the inequality follows by (1). Hence replacements of the second type can happen at most $2^{\lambda(i)-(\lambda(i)-i)} = 2^i$ times. □

By Claim 2, S stabilizes after a certain number of j-steps. That number may be large. However, the number of replacements of S is small. The final set $S \ni x$ has cardinality $2^{\lambda(i)-i}$, and can be specified by the number of replacements resulting in its current contents (as in Claim 2), and by i, n, λ. This shows that $K(S) \leqslant i + K(i,n,\lambda) + O(1)$. □

REMARK 1. The theorem implies that for every function $h(i)$ such that the function $\lambda(i) = h(i) + i$ satisfies the conditions of item (ii) there is an x such that $h_x(i) = \mathcal{E}(h(i))$ with $\varepsilon = \delta = O(K(i,n,h))$.

REMARK 2. The proof of the theorem shows that for every function $\lambda(i)$ satisfying the conditions of item (ii) there is x such that $\lambda_x(i|n,\lambda) = \mathcal{E}(\lambda(i))$ with $\varepsilon = \delta = K(i) + O(1)$ where the conditional structure function $\lambda_x(i|y) = \min_S\{K(S|y) + \log|S| : S \ni x,\ K(S|y) \leqslant i\}$. Consequently, for every function $h(i)$ such that the function $\lambda(i) = h(i) + i$ satisfies the conditions of item (ii) there is an x such that $h_x(i|n,h) = \mathcal{E}(h(i))$ with $\varepsilon = \delta = O(K(i))$ where the conditional structure function $h_x(i|y) = \min_S\{\log|S| : S \ni x,\ K(S|y) \leqslant i\}$.

REMARK 3. In the proof of Item (ii) of the theorem we can consider every finite set U with $|U| \geqslant 2^n$ in place of the set A of all strings of length n. Then we obtain a string $x \in U$ such that $\lambda_x = \mathcal{E}(\lambda)$ with $\varepsilon(i) = \delta(i) = K(i,U,\lambda) + O(1)$.

The following central result of this paper shows that the λ_x (equivalently h_x, by Lemma 1) and β_x can be expressed in one another but for a logarithmic additive error.

THEOREM 2. *For every x of length n and complexity k it holds $\beta_x(\alpha) + k \cong \lambda_x(\alpha)$ for $\varepsilon = \delta = O(\log n)$.*

PROOF. The statement of the theorem easily follows from the following two inequalities that are valid for every x, respectively every x of length n for the second one:

(9) $\quad \beta_x(i) + k \leqslant \lambda_x(i) + O(\log n)$, for every $i \leqslant k$; and

(10) $\quad \lambda_x(i + O(\log n)) \leqslant \beta_x(i) + k + O(\log n)$, for every i satisfying $K(n) + O(1) \leqslant i \leqslant k$.

It is convenient to rewrite the formula defining $\delta(x|A)$ using the symmetry of information (2) as follows:

$$\begin{aligned}(11)\quad \delta(x|A) &= \log|A| + K(A) - K(A|x) - k + O(\log n)\\ &= \Lambda(A) - K(A|x) - k + O(\log n).\end{aligned}$$

Ad (9): This is easy, because for every set $S \ni x$ witnessing $\lambda_x(i)$ we have $\delta(x|S) \leqslant \Lambda(S) - k + O(\log n) = \lambda_x(i) - k + O(\log n)$ and $\beta_x(i) \leqslant \delta(x|S)$.

Ad (10): This is more difficult. By (11) it suffices to prove the following

LEMMA 2. *For every $A \ni x$ there is $S \ni x$ with $K(S) \leqslant K(A) - K(A|x) + O(\log m)$ and $\log|S| = \log|A|$ (where $m = \Lambda(A)$ and $\log i$ means $\lceil \log i \rceil$).*

Indeed for every A witnessing $\beta_x(i)$ the set S will witness $\lambda_x(i + O(\log m)) \leqslant \beta_x(i) + k + O(\log m)$. It remains to note that $m = \log|A| + K(A) = K(x|A) + \beta_x(i) + K(A) \leqslant 3n + O(\log n)$ provided $i \geqslant K(n) + O(1)$.

PROOF. Fix some $A_0 \ni x$ and let $m = \Lambda(A_0)$. Our task is the following: Given $K(A_0), \log|A_0|, K(A_0|x)$, to enumerate a family of at most $2^{K(A_0)-K(A_0|x)+O(\log m)}$ different sets S with $\log|S| = \log|A_0|$ that cover all y's covered by sets A, with $K(A) = K(A_0)$, $K(A|y) = K(A_0|x)$ and $\log|A| = \log|A_0|$. Since the complexity of each enumerated S does not exceed $K(A_0) - K(A_0|x) + O(\log m) + K(K(A_0), \log|A_0|, K(A_0|x)) + O(1) = K(A_0) - K(A_0|x) +$

$O(\log m)$ the lemma will be proved. The proof is by running the following:

Algorithm: Given $K(A_0), \log |A_0|, K(A_0|x)$ we run all programs dovetail style. We maintain auxiliary set-variables C, U, D, all of them initially $\emptyset$. Every time a new program p of length $K(A_0)$ in the dovetailing process halts, with as output a set A with $\log |A| = \log |A_0|$, we execute the following steps:

Step 1: Update $U := U \cup A$.

Step 2: Update $D := \{y \in U \setminus C$: y is covered by at least $t = 2^{K(A_0|x)-\delta}$ different generated A's$\}$, where $\delta = O(\log m)$ will be defined later.

Step 3: This step is executed only if there is $y \in D$ that is covered by at least $2t$ different generated A's. Enumerate as much new disjoint sets S as are needed to cover D: we just chop D into parts of size $2^{\log |A_0|}$ (the last part may be incomplete) and name those parts the new sets S. Every time a new set S is enumerated, update $C := C \cup S$.

CLAIM 3. The string x is an element of some enumerated S, and the number of enumerated S's is at most $2^{K(A_0)-K(A_0|x)+O(\log m)}$.

PROOF. By way of contradiction, assume that x is not an element of the enumerated S's. Then there are less than $2^{K(A_0|x)-\delta+1}$ different generated sets A such that $x \in A$. Every such A therefore satisfies $K(A|x) \leqslant K(A_0|x) - \delta + O(\log m) < K(A_0|x)$ if δ is chosen appropriately. Since A_0 was certainly generated this is a contradiction.

It remains to show that we enumerated at most $2^{K(A_0)-K(A_0|x)+O(\log m)}$ different S's. Step 3 is executed only once per t executions of Step 1, and Step 1 is executed at most $2^{K(A_0)}$ times. Therefore Step 3 is executed at most $2^{K(A_0)}/t = 2^{K(A_0)-K(A_0|x)+\delta}$ times. The number of S's formed from incomplete parts of D's in Step 3 is thus at most $2^{K(A_0)-K(A_0|x)+\delta}$. Let us bound the number of S's formed from complete parts of D's. The total number of elements in different A's generated is at most $2^{K(A_0)+\log |A_0|}$ counting multiplicity. Therefore the number of elements in their union, having multiplicity $2^{K(A_0|x)-\delta}$ or more, is at most $2^{K(A_0)+\log |A_0| - K(A_0|x)+\delta}$. Every S formed from a complete part of a set D in Step 3 accounts for $2^{\log |A_0|}$ of them. Hence the number of S's formed from complete parts of D's is at most $2^{K(A_0)-K(A_0|x)+\delta}$. □□□

COROLLARY 1. *For every x of length n and complexity $k \leqslant n$ there is a non-increasing function β such that $\beta(0) \leqslant n - k$, $\beta(k) = 0$ and $\beta_x = \mathcal{E}(\beta)$ for $\varepsilon, \delta = O(\log n)$. Conversely, for every non-increasing function β such that $\beta(0) \leqslant n - k$, $\beta(k) = 0$ there is x of length n and complexity $k \pm \delta$ such that $\beta_x = \mathcal{E}(\beta)$ for $\varepsilon = \delta = O(\log n) + K(\beta)$.*

PROOF. The first part is more or less immediate. Or use the first part of Theorem 1 and then let $\beta(i) = \lambda(i) - k$. To prove the second part, use the second part of Theorem 2, and the second part of Theorem 1 with $\lambda(i) = \beta(i) + k$. □

REMARK 4. We can see from the proof of Theorem 2 that for every $S \ni x$ of complexity at most $\alpha + O(\log n)$ minimizing $\Lambda(S)$ it holds $\delta(x|S) \leqslant \beta_x(\alpha) + O(\log n)$. Ignoring $O(\log n)$ terms, we see that at every complexity level, every best hypothesis at this level with respect to $\Lambda(S)$ is also a best one with respect to typicality. This explains why it is worth to find shortest two-part descriptions for given data x: this is the single known way to find an $S \ni x$ with respect to which x is as typical as it is possible at that complexity level. Note that the set $\{\langle x, S, \beta\rangle | x \in S,\ \delta(x|S) < \beta\}$ is not enumerable so we are not able to generate such S's directly (Section 3).

The converse is not true: not every hypothesis, consisting of a finite set, witnessing $\beta_x(\alpha)$ also witnesses $\lambda_x(\alpha)$ or $h_x(\alpha)$. For example, let x, y be strings of length $n, n/2$, respectively, with $K(x,y) \geqslant 3n/2$. Let $S_1 = \{0,1\}^n \cup \{y\}$ and let $S_2 = \{0,1\}^n$. Then both S_1, S_2 witness $\beta_x(\frac{n}{2} + O(\log n)) = O(1)$ but $\Lambda(S_1) = \frac{3n}{2} + O(\log n) \gg \lambda_x(\frac{n}{2} + O(\log n)) = n + O(\log n)$ while $\log |S_2| = n \gg h_x(\frac{n}{2} + O(\log n)) = \frac{n}{2} + O(\log n)$.

However, for every α such that $\lambda_x(i)$ decreases when $i \to \alpha$ with $i \leqslant \alpha$, a witness set for $\beta_x(\alpha)$ is also a witness set for $\lambda_x(\alpha)$ and $h_x(\alpha)$. We will call such α *critical* (with respect to x): these are the model complexities at which the two-part MDL code-length decreases, while it is stable in between such critical points. We measure the criticality of α by the function $s_{x,\alpha}(\delta)$ defined as follows: $s_{x,\alpha}(\delta) = \max\{\varepsilon : \lambda_x(\alpha - \varepsilon) \leqslant \lambda_x(\alpha) + \delta\}$.

The function $s_{x,\alpha}(\delta)$ does not decrease and the speed of its growth on small positive δ measures the criticality of α: the slower $s_{x,\alpha}(\delta)$ increases, the more α is critical. For critical α, if $\lambda_x(\alpha_1)$ is close to $\lambda_x(\alpha)$ then α_1 is close to α. More specifically, $\alpha - \alpha_1 \leqslant s_{x,\alpha}(\lambda_x(\alpha_1) - \lambda_x(\alpha))$.

The next theorem shows, for critical α, that for every $S \ni x$ with $K(S) \approx \alpha$ and $\delta(x|S) \approx \beta_x(\alpha)$, we have $\log |S| \approx h_x(\alpha)$ and $\Lambda(S) \approx \lambda_x(\alpha)$.

THEOREM 3. *There is a constant C such that the following holds. Let $x \in S$, and let $\Delta = \delta(x|S) - \beta_x(\alpha) + C \log \Lambda(S)$. Then, $\log |S| \leqslant h_x(\alpha) + s_{x,\alpha}(\Delta) + \Delta$ and $\Lambda(S) \leqslant \lambda_x(\alpha) + s_{x,\alpha}(\Delta) + \Delta + K(S) - \alpha$.*

EXAMPLE 2. **"Snooping Curves" for Logarithmic Loss:** In [24] the notion of a "snooping curve" $L_x(\alpha)$ of x was introduced, expressing the minimal logarithmic loss in predicting the consecutive elements of a given individual string x, in each prediction using the preceding sequence of elements, by the best prediction strategy of complexity at most α. Formally, $L_x(\alpha) = \min_{K(P) \leqslant \alpha} \mathrm{Loss}_P(x)$. The minimum is taken over all prediction strategies P of complexity at most α. A prediction strategy P is a mapping from the set of strings of length less than $|x|$ into the set of rational numbers in the segment $[0,1]$. The value $P(x_1 \dots x_i)$ is regarded as our belief (or probability) that $x_{i+1} = 1$ after we have observed $x_1, \dots, x_i$. If the actual bit x_{i+1} is 1 the strategy suffers the loss $-\log p$ otherwise $-\log(1-p)$. The strategy is a finite object and $K(P)$ may by defined as the complexity of this object, or as the minimum size of a program that identifies $n = |x|$ and given y finds $P(y)$. The notation $\mathrm{Loss}_P(x)$ indicates the total loss of P on x, i.e. the sum of all n losses: $\mathrm{Loss}_P(x) = \sum_{i=0}^{|x|-1} (-\log |P(x_1 \dots x_i) - 1 + x_{i+1}|)$.

A result of [24] describes possible shapes of L_x but only for $\alpha = o(n)$. Here we show that:

LEMMA 3. *$L_x(\alpha \pm O(\log n)) = h_x(\alpha \pm O(\log n))$ for every x and α. Thus, Lemma 1 and Theorem 1 describe also the coarse shape of all possible snooping curves for all $\alpha \leqslant n$.*

PROOF. ($\leqslant$) A given finite set A of binary strings of length n can be identified with the following prediction strategy P: Having read the prefix y of x it outputs $p = |A_{y1}|/|A_y|$ where A_y stands for the number of strings in A having prefix y.

It is easily seen, by induction, that $\mathrm{Loss}_P(y) = \log(|A|/|A_y|)$ for every y. Therefore, $\mathrm{Loss}_P(x) = \log|A|$ for every $x \in A$. Since P corresponds to A in the sense that $K(P|A) = O(1)$, we obtain $L_x(\alpha + O(\log n)) \leqslant h_x(\alpha)$. The term $O(\log n)$ is required, because the initial set of complexity α might contain strings of different lengths while we need to know n to get rid of the strings of lengths different from n.

($\geqslant$) Conversely, assume that $\mathrm{Loss}_P(x) \leqslant m$. Let $A = \{x \in \{0,1\}^n : \mathrm{Loss}_P(x) \leqslant m\}$. Since $\sum_{|x|=n} 2^{-\mathrm{Loss}_P(x)} = 1$ (proof by induction on n), and $2^{-\mathrm{Loss}_P(x)} \geqslant 2^{-m}$ for every $x \in A$, we can conclude that A has at most 2^m elements. Since $K(A|P) = O(\log m)$, we obtain $h_x(\alpha + O(\log n)) \leqslant L_x(\alpha)$. □

◊

EXAMPLE 3. **Foundations of MDL:** Consider the following algorithm based on the Minimum Description Length principle. Given x, the data to explain, and α, the maximum allowed complexity of explanation, we search for programs p of length at most α that print a finite set $S \ni x$. Such pairs (p, S) are possible explanations. Since the programs use unknown computation time, we can never be certain that we have found all possible explanations. Nonetheless, there is some (unknown) moment at which this will be the case, since the number of possible explanations is finite. The *best explanation* is defined to be the (p, S) for which $\delta(x|S)$ is minimal. But since the function $\delta(x|S)$ is not computable, we cannot find the best explanation through using $\delta(x|S)$ directly. To overcome this problem we use the indirect method of MDL: the (p, S) for which $|p| + \log|S|$ is currently minimum is said to be the currently best hypothesis. Initially, the currently best hypothesis changes from time to time due to the appearance of a better hypothesis. Since no hypothesis is declared best twice, from some moment onwards the explanation $(\tilde{p}, \tilde{S})$ which is declared best does not change anymore. That $\tilde{S}$ witnesses $\lambda_x(\alpha)$. By Theorem 2 we have $\delta(x|S) \leqslant \beta_x(\alpha - O(\log n)) + O(\log n)$. That is, $\tilde{S}$ is only $O(\log n)$ worse than the best hypothesis of complexity at most $\alpha - O(\log n)$. If α is chosen so that β_x makes no jump on the interval $(\alpha - O(\log n), \alpha)$ then $\tilde{S}$ is only a little worse than the best hypothesis of complexity at most α. ◊

EXAMPLE 4. **Foundations of Maximum Likelihood:** The algorithm based on ML principle is similar to the algorithm of the previous example. The only difference is that the currently best (p, S) is the one for which $\log|S|$ is minimal. In this case the limit hypothesis $\tilde{S}$ will witness $h_x(\alpha)$ and we obtain the same corollary: $\delta(x|S) \leqslant \beta_x(\alpha - O(\log n)) + O(\log n)$. ◊

EXAMPLE 5. **"Positive" and "Negative" Individual Randomness:** In [9] the existence of strings was established for which essentially the singleton set consisting of the string itself is a minimal sufficient statistic. While a sufficient statistic of an object yields a two-part code that is as short as the shortest code, restricting the complexity of the allowed statistic may yield two-part codes that are considerably longer than the best one-part code (so the statistic is insufficient). This is what happens for the non-stochastic objects. In fact, for every object there is a complexity bound below which this happens—but if that bound is small (logarithmic) we call the object "stochastic" since it has a simple satisfactory explanation (sufficient statistic). Thus, Kolmogorov in [14] (full text given below) makes the important distinction of an object being random in the "negative" sense by having this bound high (they have high complexity and are not typical elements of a low-complexity model), and an object being random in the "positive, probabilistic" sense by both having this bound small and itself having complexity considerably exceeding this bound (like a string x of length n with $K(x) \geqslant n$, being typical for the set $\{0,1\}^n$ while this set has complexity $K(n) + O(1) = O(\log n)$).

Corollary 1 establishes that for every length n and every complexity $k \leqslant n + K(n) + O(1)$ (the maximal complexity of x of length n) and every $\alpha \in [0, k]$, there are x's of length n and complexity k such that the minimal randomness deficiency $\beta_x(i) \geqslant n - k - O(\log n)$ for every $i \leqslant \alpha - O(\log n)$ and $\beta_x(i) \leqslant O(\log n)$ for every $i > \alpha + O(\log n)$. Therefore, the set of n-length strings of every complexity k can be partitioned in subsets of strings that have a Kolmogorov minimal sufficient statistic of complexity $\Theta(i \log n)$ for $i = 1, \ldots, k/\Theta(\log n)$. For instance, there are n-length non-stochastic strings of almost maximal complexity $n - \sqrt{n}$ having significant $\sqrt{n} - O(\log n)$ randomness deficiency with respect to $\{0,1\}^n$ or, in fact, every other finite set of complexity less than $n - O(\log n)$! Translation by L.A. Levin [11] of Kolmogorov's abstract [14]: "To each constructive object corresponds a function $\Phi_x(k)$ of a natural number k—the log of minimal cardinality of x-containing sets that allow definitions of complexity at most k. If the element x itself allows a simple definition, then the function Φ drops to 1 even for small k. Lacking such definition, the element is "random" in a negative sense. But it is positively "probabilistically random" only when function Φ having taken the value Φ_0 at a relatively small $k = k_0$, then changes approximately as $\Phi(k) = \Phi_0 - (k - k_0)$." ◊

3 Computability Questions

How difficult is it to compute the functions h_x, λ_x, β_x, and the minimal sufficient statistic? To express the properties appropriately we require the notion of functions that are not computable, but can be approximated monotonically by a computable function.

DEFINITION 2. A function $f : \mathcal{N} \to \mathcal{R}$ is *upper semi-computable* if there is a Turing machine T computing a total function ϕ such that $\phi(x, t+1) \leqslant \phi(x, t)$ and $\lim_{t\to\infty} \phi(x, t) = f(x)$. This means that f can be computably approximated from above. If $-f$ is upper semi-computable, then f is lower semi-computable. A function is called *semi-computable* if it is either upper semi-computable or lower semi-computable. If f is both upper semi-computable and lower semi-computable, then we call f *computable* (or recursive if the domain is integer).

Semi-computability gives no speed-of-convergence guaranties: even though the limit value is monotonically approximated we know at no stage in the process how close we are to the limit value.

The functions $h_x(\alpha), \lambda_x(\alpha), \beta_x(\alpha)$ have finite domain for given x and hence can be given as a table—so formally speaking they are computable. But this evades the issue: there is no algorithm that computes these functions for given x and α. Considering them as two-argument functions we show the following (we actually quantify these):

- The functions $h_x(\alpha)$ and $\lambda_x(\alpha)$ are upper semi-computable but not computable.
- Moreover, there is no algorithm that given x^* and α finds $h_x(\alpha)$ or $\lambda_x(\alpha)$.
- The function $\beta_x(\alpha)$ is not upper- or lower semi-computable, but we can compute it given an oracle for the halting problem.
- There is no algorithm that given x and $K(x)$ finds a minimal sufficient statistic for x.

Structure Function: It is easy to see that $h_x(\alpha)$ or $\lambda_x(\alpha)$ are upper semi-computable: run all programs of length up to α dovetailed fashion, check whether a halting program produced a finite set containing x, and replace the previous candidate with the new set if it is smaller.

The next question is: Is the function $\lambda_x(\alpha)$, as the function of two arguments, computable? Of course not, because if this were the case, then we could find, given every large k, a string of complexity at least k. Indeed, we know that there is a string x for which $\lambda_x(k) > k$. Applying the algorithm to all strings in the lexicographical order find first such x. Obviously $K(x) \geqslant k - O(1)$. But it is known that we cannot prove that $K(x) > k$ for sufficiently large k, [12].

Assume now that we are given also $K(x)$. The above argument does not work any more but the statement remains true: $\lambda_x(\alpha)$ is not computable even if the algorithm is given $K(x)$.

Assume first that the algorithm is required to output the correct answer given any approximation to $K(x)$. We show that no algorithm can find λ that is close to $\lambda_x(\alpha)$ for some $0 \ll \alpha \ll K(x)$.

THEOREM 4. *For every constant c there is a constant d such the following holds. There is no algorithm that for infinitely many k, given k and x of length $k + d\log k$ with $|K(x) - k| \leqslant 2\log k$, always finds λ such that there is $2\log k \leqslant \alpha \leqslant k$ with $|\lambda_x(\alpha) - \lambda| < c\log k$.*

What if the algorithm is required to approximate λ_x only if it is given the precise value of $K(x)$? We are able to prove that in this case the algorithm cannot compute $\lambda_x(\alpha)$ too. It is even impossible to approximate the complexity of minimal sufficient statistic. To formulate this result precisely consider the following promise problem:

Input: $x, k = K(x), \alpha \in [\varepsilon, k - \varepsilon]$.
Output:
1, if $\lambda_x(\alpha - \varepsilon) < k + 6\log k$,
0, if $\lambda_x(\alpha + \varepsilon) > k + 3\varepsilon$.

If neither of two above cases occurs the algorithm may output any value or no value at all. We assume that $6\log k \leqslant 3\varepsilon$ hence the algorithm is never required to output 0 and 1 simultaneously.

THEOREM 5. *There is no algorithm A that for infinitely many n solves this promise problem for $|x| = n$ and $\varepsilon = n/10\log n$.*

COROLLARY 2. *There is no algorithm that given $x, k = K(x)$ finds an integer valued function λ on $[0, k]$ such that $\lambda_x = \mathcal{E}(\lambda)$ for $\varepsilon = \delta = |x|/10\log|x|$.*

Indeed, if there were such algorithm we could solve the above promise problem for all x such that $\varepsilon = |x|/10\log|x| \geqslant 6\log k$ by answering 1 when $\lambda(\alpha) \leqslant k + 2\varepsilon$ and 0 otherwise.

REMARK 5. Let us replace in the above promise problem $K(x)$, the prefix complexity of x, by $C(x)$, the plain complexity of x. For the modified problem we can strengthen the above theorem by allowing $\varepsilon = n/c$ where the constant c depends on the reference computer.

After a discussion of these results, Andrei A. Muchnik suggested, and proved, that if we are also given an α_0 such that $\lambda_x(\alpha_0) \approx K(x)$ but $\lambda_x(\alpha)$ is much bigger than $K(x)$ for α much less than α_0 (which is therefore the complexity of the minimal sufficient statistic), then we can compute λ_x over all of its domain. This result underlines the significance of the information contained in the *minimal* sufficient statistic:

THEOREM 6. *There are a constant $c \geqslant 0$ and an algorithm that given any x, k, α_0 with $K(x) \leqslant k \leqslant \lambda_x(\alpha_0)$ finds a non-increasing function λ defined on $[0, k]$ such that $\lambda_x = \mathcal{E}(\lambda)$ with $\delta = \lambda_x(\alpha_0) - K(x) + O(1)$ and $\varepsilon = \alpha_0 - \alpha_1 + c\log k$ where $\alpha_1 = \min\{\alpha : \lambda_x(\alpha) \leqslant k + c\log k\}$.*

Randomness Deficiency Function: The function $\beta_x(\alpha)$ is computable from x, α given an oracle for the halting problem: run all programs of length $\leqslant \alpha$ dovetailed fashion and find all finite sets S containing x that are produced. With respect to all these sets determine the conditional complexity $K(x|S)$ and hence the randomness deficiency $\delta(x|S)$. Taking the minimum we find $\beta_x(\alpha)$. All these things are possible using information from the halting problem to determine whether a given program will terminate or not. It is also the case that the function $\beta_x(\alpha)$ is upper semi-computable from $x, \alpha, K(x)$ up to a logarithmic error: this follows from the semi-computability of $\lambda_x(\alpha)$ and Theorem 2. More subtle is that β_x is not semi-computable, not even within a large margin of error:

THEOREM 7. *The function $\beta_x(\alpha)$ is*
(i) not lower semi-computable to within precision $|x|/3$; and
(ii) not upper semi-computable to within precision $|x|/\log^4|x|$.

Open question. Is there a non-increasing (with respect to α) upper semi-computable function $f_x(\alpha)$ such that, for all x, $\beta_x(\alpha) = \mathcal{E}(f_x(\alpha))$ for $\varepsilon = \delta = O(\log|x|)$ (or for $\varepsilon = \delta = o(|x|)$)?

4 Conclusion

When we compare statistical hypotheses S_0 and S_1 to explain x we should take into account three parameters: $K(S), K(x|S), \log|S|$. The first parameter is the simplicity of the theory S explaining the data. The difference $\delta(x|S) = \log|S| - K(x|S)$ (the randomness deficiency) shows how typical is the data with respect to S. The sum $\Lambda(S) = K(S) + \log|S|$

reflects how short the total two-part code of the data using theory S is, when we simply use a full-length index of x in S (and we thus ignore non-typicality of x in S). We would like to define that S_0 is not worse than S_1 (as an explanation for x), in symbols: $S_0 \leqslant S_1$, if $K(S_0) \leqslant K(S_1), \delta(x|S_0) \leqslant \delta(x|S_1), \Lambda(S_0) \leqslant \Lambda(S_1)$. Note that this is not equivalent to saying that $K(S_0) \leqslant K(S_1), \delta(x|S_0) \leqslant \delta(x|S_1), \log|S_0| \leqslant \log|S_1|$. (The latter relation implies $S_0 \leqslant S_1$ but not vice versa.) Therefore we are eager, for every x, to define the set A_x of all triples $\langle K(S), \delta(x|S), \Lambda(S)\rangle$ such that $x \in S$ with a component wise order relation on those triples.

Our results (Theorems 1, 2, 3) describe completely (with $O(\log n)$-accuracy) possible shapes of a closely related set B_x consisting of all triples $\langle \alpha, \beta, \lambda\rangle$ such that there is a set $S \ni x$ with $K(S) \leqslant \alpha$, $\delta(x|S) \leqslant \beta$, $\Lambda(S) \leqslant \lambda$. Since the sets A_x and B_x have the same minimal triples, we can informally say that our results describe completely the possible shapes of the set of triples $\langle K(S), \delta(x|S), \Lambda(S)\rangle$ for non-improvable hypotheses S explaining x. Previously, a characterization was obtained by V'yugin [22, 23] for possible shapes of projections of B_x on α, β-coordinates but only for the case when $\alpha = o(|x|)$. Our results describe the possible shapes of the entire set B_x for the full domain of α (with $O(\log n)$-accuracy). Namely, let f be a non-increasing integer valued function such that $f(0) \leqslant n$, $f(k) = k$ and $\tilde{B}_f = \{\langle \alpha, \beta, \lambda\rangle \mid f(\alpha) \leqslant \lambda,\ f(\alpha) \leqslant \beta + k\}$. For every x of length n and complexity k there is f such that

$$\tilde{B}_f + u \subset B_x \subset \tilde{B}_f - u \qquad (12)$$

where $u = \langle c\log n, c\log n, c\log n\rangle$ for some universal constant c. Conversely, for every $k \leqslant n$ and every such f there is x of length n and complexity $k \pm c\log n$ such that (12) holds for $u = \langle c\log K(n,f,k), c\log K(n,f,k), c\log K(n,f,k)\rangle$. Our results imply that the set B_x is not computable given x, k but is computable given x, k and α_0.

The major result of this work is that a finite set that witnesses $\lambda_x(\alpha)$ or $h_x(\alpha)$ (minimizes the log-cardinality or $\Lambda(S)$ of a set containing x of complexity at most α) simultaneously witnesses $\beta_x(\alpha)$ (minimizes the randomness deficiency of x with respect to a set containing it of complexity at most α). We have also addressed the non-computability of λ_x, h_x and β_x.

A Validity for Extended Models

Following Kolmogorov we analyzed a canonical setting where the models are finite sets. As Kolmogorov himself pointed out, this is no real restriction: the finite sets model class is equivalent, up to a logarithmic additive term, to the model class of probability density functions, as studied in [18, 9]. The analysis is valid, up to logarithmic additive terms, also for the model class of total recursive functions, as studied in [21]. The model class of *computable probability density functions* consists of the set of functions $P : \{0,1\}^* \to [0,1]$ with $\sum P(x) = 1$. "Computable" means here that there is a Turing machine T_P that, given x and a positive rational ε, computes $P(x)$ with precision ε. The (prefix-) complexity $K(P)$ of a computable (possibly partial) function P is defined by $K(P) = \min_i\{K(i)$: Turing machine T_i computes $P\}$. A string x is typical for a distribution P if the randomness deficiency $\delta(x|P) = -\log P(x) - K(x|P)$ is small. The conditional complexity $K(x|P)$ is defined as follows. Say that a function A approximates P if $|A(y,\varepsilon) - P(y)| < \varepsilon$ for every y and every positive rational ε. Then $K(x|P)$ is the minimum length of a program that given every function A approximating P as an oracle prints x. Similarly, P is c-optimal for x if $K(P) - \log P(x) \leqslant K(x) + c$. Thus, instead of the data-to-model code length $\log|S|$ for finite set models, we consider the data-to-model code length $-\log P(x)$ (the Shannon-Fano code). The value $-\log P(x)$ measures also how likely is x under hypothesis P and the mapping $x \mapsto P_{\min}$ where $P_{\min}$ minimizes $-\log P(x)$ over P with $K(P) \leqslant \alpha$ is a *maximum likelihood estimator*. Our results thus imply that that maximum likelihood estimator always returns a hypothesis with minimum randomness deficiency.

The model class of *total recursive functions* consists of the set of computable functions $p : \{0,1\}^* \to \{0,1\}^*$ that are defined for every argument. The (prefix-) complexity $K(p)$ of a total recursive function p is defined by $K(p) = \min_i\{K(i)$: Turing machine T_i computes $p\}$. In place of $\log|S|$ for finite set models we consider the data-to-model code length $l_x(p) = \min\{|d| : p(d) = x\}$. A string x is typical for a total recursive function p if the randomness deficiency $\delta(x|p) = l_x(p) - K(x|p)$ is small. The conditional complexity $K(x|p)$ is defined as the minimum length of a program that given p as an oracle prints x. Similarly, p is c-optimal for x if $K(p) + l_x(p) \leqslant K(x) + c$.

PROPOSITION 1. *For every x and every finite set $S \ni x$ there is:*

(a) A computable probability distribution function P with $-\log P(x) = \log|S|$, $\delta(x|P) = \delta(x|S) + O(1)$ and $K(P) = K(S) + O(1)$; and

(b) A total recursive function p such that $l_x(p) \leqslant \log|S|$, $\delta(x|p) \leqslant \delta(x|S) + O(1)$ and $K(p) = K(S) + O(1)$.

PROOF. (a) Define $P(y) = 1/|S|$ for $y \in S$ and 0 otherwise.
(b) If $S = \{x_0, \ldots, x_{m-1}\}$, then define $p(d) = x_{d \bmod m}$. $\square$

PROPOSITION 2. *There are constants c, C, such that for every string x, the following holds:*

(a) For every computable probability distribution function P there is a finite set $S \ni x$ such that $\log|S| < -\log P(x) + 1$, $\delta(x|S) \leqslant \delta(x|P) + K(\lfloor -\log P(x)\rfloor) + C$ and $K(P) \leqslant K(S) + K(\lfloor -\log P(x)\rfloor) + C$; and

(b) For every total recursive function p there is a finite set $S \ni x$ with $\log|S| \leqslant l_x(p)$, $\delta(x|S) \leqslant \delta(x|p) + K(l_x(p)) + c$ and $K(p) \leqslant K(S) + K(l_x(p)) + c$.

PROOF. (a) Let $m = \lfloor -\log P(x)\rfloor$, that is, $2^{-m-1} < P(x) \leqslant 2^{-m}$. Define $S = \{y : P(y) > 2^{-m-1}\}$. Then, $|S| < 2^{m+1} \leqslant 2/P(x)$, which implies the claimed value for $\log|S|$. To list S it suffices to compute all consecutive values of $P(y)$ to sufficient precision until the combined probabilities exceed $1 - 2^{-m-1}$. That is, $K(S) \leqslant K(P) + K(m) + O(1)$. Finally, $\delta(x|S) = \log|S| - K(x|S) < -\log P(x) - K(x|S) + 1 = \delta(x|P) + K(x|P) - K(x|S) + 1 \leqslant \delta(x|P) + K(S|P) + O(1) \leqslant \delta(x|P) + K(m) + O(1)$.

(b) Define $S = \{y : p(d) = y,\ \ |d| = l_x(p)\}$. Then, $\log|S| \leqslant l_x(p)$. To list S it suffices to compute $p(d)$ for every argument of length equal $l_x(p)$. Hence, $K(S) \leqslant K(p) + K(l_x(p)) + O(1)$.

The upper bound for $\delta(x|S)$ is derived just in the same way as in the proof of item (a). $\square$

REMARK 6. How large are the nonconstant additive complexity terms in Proposition 2 for strings x of length n? In item (a), we are commonly only interested in P such that $K(P) \leqslant n + O(\log n)$ and $-\log P(x) \leqslant n + O(1)$. Indeed, for every P there is P' such that $K(P') \leqslant \min\{K(P), n\} + O(\log n)$, $\delta(x|P') \leqslant \delta(x|P) + O(\log n)$, $-\log P'(x) \leqslant \min\{-\log P(x), n\} + 1$. Such P' is defined as follows: If $K(P) > n$ then $P'(x) = 1$ and $P'(y) = 0$ for every $y \neq x$; otherwise $P' = (P + U_n)/2$ where U_n stands for the uniform distribution on $\{0,1\}^n$. Then the additive terms in item (a) are $O(\log n)$. In item (b) we are commonly only interested in p such that $K(p) \leqslant n + O(\log n)$ and $l_x(p) \leqslant n + O(1)$. Indeed, for every p there is p' such that $K(p') \leqslant \min\{K(p), n\} + O(\log n)$, $\delta(x|p') \leqslant \delta(x|p) + O(\log n)$, $l_x(p') \leqslant \min\{l_x(p), n\} + 1$. Such p' is defined as follows: If $K(p) > n$ then p' maps all strings to x; otherwise $p'(0u) = p(u)$ and $p'(1u) = u$. Then the additive terms in item (b) are $O(\log n)$. Thus, in this sense all results in this paper that hold for finite set models extend, up to a logarithmic additive term, to computable probability density function models and to total recursive function models. Since the results in this paper hold only up to additive logarithmic term anyway, this means that all of them equivalently hold for the model class of computable probability density functions, as well as for the model class of total recursive functions.

Acknowledgment: We thank Harry Buhrman, Andrei A. Muchnik, Marcus Hutter, for helpful discussions, and Tom Cover, Péter Gács, and Leonid A. Levin for historical details about Kolmogorov's original proposal of the structure function in the early seventies and about their own unpublished work on the subject. After the first author questioned him whether the result in Theorem 2 was perhaps known before, Muchnik answered negatively and supplied an independent proof of it. He also suggested and proved Theorem 6.

References

[1] A.R. Barron, J. Rissanen, and B. Yu, The minimum description length principle in coding and modeling, *IEEE Trans. Inform. Theory*, IT-44:6(1998), 2743–2760.

[2] T.M. Cover, Kolmogorov complexity, data compression, and inference, pp. 23–33 in: *The Impact of Processing Techniques on Communications*, J.K. Skwirzynski, Ed., Martinus Nijhoff Publishers, 1985.

[3] T.M. Cover, Attending [15], Email to PV, January 24, 2002.

[4] T.M. Cover, P. Gács and R.M. Gray, Kolmogorov's contributions to Information Theory and Algorithmic Complexity, *Ann. Probab.*, 17(1989), 840–865.

[5] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.

[6] R. A. Fisher, On the mathematical foundations of theoretical statistics, *Philosophical Transactions of the Royal Society of London, Ser. A*, 222(1922), 309–368.

[7] P. Gács, On the symmetry of algorithmic information, *Soviet Math. Dokl.*, 15 (1974) 1477–1480. Correction: ibid., 15 (1974) 1480.

[8] P. Gács, Attending [15], Email to PV, January 24, 2002.

[9] P. Gács, J. Tromp, P.M.B. Vitányi. Algorithmic statistics, *IEEE Trans. Inform. Theory*, 47:6(2001), 2443–2463.

[10] Q. Gao, M. Li and P.M.B. Vitányi, Applying MDL to learn best model granularity, *Artificial Intelligence*, 121(2000), 1–29.

[11] L.A. Levin. Emails to PV on February 7,11,20, 2002.

[12] M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications* , Springer-Verlag, New York, 2nd Edition, 1997.

[13] A.N. Kolmogorov, Three approaches to the quantitative definition of information, *Problems Inform. Transmission* 1:1 (1965) 1–7.

[14] A.N. Kolmogorov. Complexity of Algorithms and Objective Definition of Randomness. A talk at Moscow Math. Soc. meeting 4/16/1974. An abstract available in *Uspekhi Mat. Nauk* 29:4(1974),155; translation in Section 5.

[15] A.N. Kolmogorov, Talk at the Information Theory Symposium in Tallinn, Estonia, 1974, according to [8, 3].

[16] P. Martin-Löf, The definition of random sequences, *Inform. Contr.*, 9(1966), 602-619.

[17] J.J. Rissanen, A Universal Prior for Integers and Estimation by Minimum Description Length, *Annals of Statistics*, 11:2(1983), 416–431.

[18] A.Kh. Shen, The concept of (α,β)-stochasticity in the Kolmogorov sense, and its properties, *Soviet Math. Dokl.*, 28:1(1983), 295–299.

[19] A.Kh. Shen, Discussion on Kolmogorov complexity and statistical analysis, *The Computer Journal*, 42:4(1999), 340–342.

[20] P.M.B. Vitányi and M. Li, Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity, *IEEE Trans. Inform. Theory*, IT-46:2(2000), 446–464.

[21] P.M.B. Vitányi, Meaningful information, Proc. ISAAC 2002, LNCS, Springer-Verlag, Berlin, 2002. http://xxx.lanl.gov/abs/cs.CC/0111053

[22] V.V. V'yugin, On the defect of randomness of a finite object with respect to measures with given complexity bounds, *SIAM Theory Probab. Appl.*, 32:3(1987), 508–512.

[23] V.V. V'yugin, Algorithmic complexity and stochastic properties of finite binary sequences, *The Computer Journal*, 42:4(1999), 294–317.

[24] V.V. V'yugin. Does snooping help? *Theoret. Comput. Sci.*, 276:1/2(2002), 407–415.

[25] 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.