Advances in Minimum Description Length: Theory and Applications

MIT: series name??

MIT: series editors name?? TODO, Editor
MIT?? TODO, Associate Editors

Advances in Minimum Description Length: Theory and Applications

edited by
Peter D. Grünwald
In Jae Myung
Mark A. Pitt

The MIT Press
Cambridge, Massachusetts
London, England

Typeset by the authors using LaTeX 2ε
Library of Congress Control No. 2001095750
Printed and bound in the United States of America

Library of Congress Cataloging-in-Publication Data

Advances in Minimum Description Length: Theory and Applications /
edited by Peter D. Grünwald, In Jae Myung, and Mark A. Pitt.
p. cm.
Includes bibliographical references and index.
ISBN TODO (alk. paper)
1. Statistics. 2. Machine learning. 3. Information Theory
I. Grünwald, Peter D. II. Myung, In Jae III. Pitt, Mark TODO.

Contents

1 Algorithmic Statistics and Kolmogorov's Structure Functions

Paul Vitányi
CWI, Kruislaan 413
1098 SJ Amsterdam, The Netherlands
paulv@cwi.nl, http://www.cwi.nl/ paulv

A non-probabilistic foundation for model selection and prediction can be based on Kolmogorov complexity (algorithmic information theory) and Kolmogorov's Structure functions (representing all stochastic properties of the data). A distinguishing feature is the analysis of goodness-of-fit of an individual model for an individual data string. Among other things it presents a new viewpoint on the foundations of 'maximum likelihood' and 'minumum description length'. We provide a leasure introduction to the central notions and results.

"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 $\Phi(k) = \Phi_0 - (k - k_0)$."
[A.N. Kolmogorov [6]]

1.1 Introduction

Naively speaking, Statistics deals with gathering data, ordering and representing data, and using the data to determine the process that causes the data. That this viewpoint is a little too simplistic is immediately clear: suppose that the true cause of a sequence of outcomes of coin flips is a "fair" coin, where both sides come up with equal probability. It is possible that the sequence consists of "heads" only. Suppose that our statistical inference method succeeds in identifying the true cause

(fair coin flips) from these data. Such a method is clearly at fault: from an all-heads sequence a good inference should conclude that the cause is a coin with a heavy bias toward "heads", irrespective of what the true cause is. That is, a good inference method must assume that the data is "typical" for the cause—that is, we don't aim at finding the "true" cause, but we aim at finding a cause for which the data is as "typical" as possible. Such a cause is called a *model* for the data. But what if the data consists of a sequence of precise alternations "head–tail"? This is as unlikely an outcome for a fair coin flip as the all-heads sequence. Yet, within the coin-type models we have no alternative than to choose a fair coin. But we know very well that the true cause must be different. For some data it may not even make sense to ask for a "true cause". This suggests that truth is not our goal; but within given constraints on the model class we try to find the model for which the data is most "typical" in an appropriate sense, the model that best "fits" the data. Considering the available model class as a a magnifying glass, finding the best fitting model for the data corresponds to finding the position of the magnifying glass that best brings the object into focus. In the coin-flipping example, it is possible that the data have no sharply focused model, but within the allowed resolution—ignoring the order of the outcomes but only counting the number of "heads"—we find the best model.

Classically, the setting of statistical inference is as follows: We carry out a probabilistic experiment of which the outcomes are governed by an unknown probability distribution P. Suppose we obtain as outcome the data sample x. Given x, we want to recover the distribution P. For certain reasons we can choose a distribution from a set of acceptable distributions only (which may or may not contain P). Intuitively, our selection criteria are that (i) x should be a "typical" outcome of the distribution selected, and (ii) the selected distribution has a "simple" description. We need to make the meaning of "typical" and "simple" rigorous and balance the requirements (i) and (ii). In probabilistic statistics one analyzes the average-case performance of the selection process.For traditional problems, dealing with frequencies over small sample spaces, this approach is appropriate. But for current novel 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 is 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 outlined strives for the firmer and less contentious ground expressed in finite combinatorics and effective computation.

We embark on a systematic study of model selection where the performance is related to the individual data sample and the individual model selected. It turns out to be more straightforward to investigate models that are finite sets first, and then generalize the results to models that are probability distributions. To simplify matters, and because all discrete data can be binary coded, we consider only data samples that are finite binary strings. Classical statistics has difficulty to express

the notion of an individual "best" model for an individual data sample. But the lucky confluence of information theory, theory of algorithms, and probability, leads to the notion of Kolmogorov complexity—the notion of information content in an individual object, and it allows us to express and analyze the novel notion of the information in one individual object (for example, a model) about another individual object (for example, the data). Development of this theory allows us to precisely formulate and quantify how well a particular model fits a particular piece of data, a matter which formerly was judged impossible.

1.2 Algorithmic Statistics

In 1965 A.N. Kolmogorov [5] combined the theory of computation and a combinatorial approach to information theory into a proposal for an objective and absolute definition of the information contained by an individual finite object, commonly represented by a finite binary string. This is to be contrasted with the average notion of the entropy of a random source as proposed by C. Shannon. The theory of "Kolmogorov complexity" has turned out to be ubiquitously applicable [4]. Continuing this train of thought, as perhaps the last mathematical innovation of an extraordinary scientific career, Kolmogorov in 1974 [6] proposed a refinement, which, in Jorma Rissanen phrasing, "permits extraction of a desired amount of properties from the data, leaving the remainder as something like noise. The properties are modeled by a finite set that includes the data string, which amounts to modeling data by uniform distribution, and the amount of properties is measured by the Kolmogorov complexity of the description of the finite set involved." This proposal can be viewed as one to found statistical theory on finite combinatorial principles independent of probabilistic assumptions, as the relation between the individual data and its explanation (model), expressed by Kolmogorov's structure function. While these notions have been intermittently studied over the years, and have been described in articles and in Cover and Thomas's influential textbook [1], as well as our own [4], they have been previously but poorly understood. Recently, however, the situation has been changed through a sequence of results concerning the "algorithmic" sufficient statistic, and its relation with the corresponding probabilistic notion in [3], and a comprehensive body of results concerning Kolmogorov's so-called "structure function" in [12]. The purpose of this paper is to briefly outline the basic notions involved and the significance of the main results obtained.

Basic Notions of the Theory: We want to describe every individual finite binary sequence x in two parts, one part being the model description (properties, the meaning of the data) and one part being the data-to-model code (the remaining random 'noise'). It is convenient to consider models that are finite sets of finite binary strings, and a contemplated model for x contains x as one of its elements. It turns out that the results are true for more sophisticated models like computable probability density functions. An example of the data-to-model code of x with respect to the model S is the index of x in the lexicographical enumeration of the

elements of S. The description of the model, typically a program that generates the model, is called a 'statistic'. The following properties of how a model S relates to sequence x are crucial:

- *Typicality:* Is x a typical or random element of S?
- *Optimality:* We call S optimal for x if the two-part description of x based on model S is minimal (among all potential two-part descriptions).

A shortest description (or program) for an optimal set S is called an 'algorithmic statistic' for x. The developments in the paper are based on imposing a constraint on the number of bits allowed to describe the model (the amount of desired properties). Let α indicate the maximum number of bits allowed to describe the model. For fixed α, we consider selection of a model for x in three ways, characterized by three different functions:

1. Selection based on the minimum randomness deficiency function (what we shall argue is the *model fitness* estimator) $\beta_x(\alpha)$;

2. Selection based on using Kolmogorov's structure function (what we shall argue is the *maximum likelihood* estimator) $h_x(\alpha)$; and

3. Selection based on the shortest two-part code length (what we shall argue is the) *MDL estimator* $\lambda_x(\alpha)$.

Method 1 is based on the notion of 'typicality' and basically selects the model S for which the data x looks most typical. In a precise mathematical sense, stated below in terms of Kolmogorov complexity, this implies that S is a model of 'best-fit' for x. So method 1 gives us the proper model for x. Unfortunately, it turns out that method 1 is too difficult to apply. But we can obtain our goal in a round-about manner: Method 2 selects a model S—containing data x—that minimizes the data-to-model code length that maximally can occur for a string in S. It is useful to explain the notion of data-to-model code length by example. For data string $x = 00\ldots0$ of length n and the large-cardinality but small-complexity model $\{0,1\}^n$, the data-to-model code length is at about n bits since there are elements in S that require n bits to be singled out. Using the small-cardinality but potentially high-complexity model $\{x\}$, the data-sto-model code length is $O(1)$. This data-to-model code length may be very different from the shortest way to describe x in a model like $\{0,1\}^n$, which is $O(1)$ bits, since x is the lexicographically first element in $\{0,1\}^n$. Method 3 selects the model S such that the total two-part description length, consisting of one part describing S containing x, and the second part describing the maximal data-to-model code of a string in S, is minimized. We will establish, in a mathematically rigorous manner, that the minimax procedure in Methods 2 and 3 result in correct selection according to the criterium of Method 1. The methods are not equivalent, since selection according to Method 1 doesn't imply a correct choice according to the criteria of either Method 2 or Method 3; and Method 3 doesn't imply a correct choice according to the criterium of Method 2.

Outline of the Results: Kolmogorov's structure function, its variations and its relation to model selection, have obtained some notoriety, but no previous comprehension. Before, it has always been questioned why Kolmogorov chose to focus on the mysterious function h_x, rather than on a more evident function denoted as β_x. The main result, in [12], with the beauty of truth, justifies Kolmogorov's intuition. One way to phrase it is this: The structure function determines all stochastic properties of the data in the sense of determining the best-fitting model at every model-complexity level. One easily stated consequence is: For all data x, both method 2 (which below is interpreted as the maximum likelihood estimator) and method 3 (that can be viewed as a minimum description length estimator) select a model that satisfies the best-fit criterion of method 1 (the best-explanation estimator), *in every case* (and not only with high probability). In particular, when the "true" model that generated the data is not in the model class considered, then the ML or MDL estimator still give a model that "best fits" the data, among all the models in the contemplated class. This notion of "best explanation" and "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. A practical consequence is as follows: While the best fit (a model that witnesses $\beta_x(\alpha)$) cannot be computationally monotonically approximated up to any significant precision, we can monotonically minimize the two-part code (find a model witnessing $\lambda_x(\alpha)$), or the one-part code (find a model witnessing $h_x(\alpha)$) and thus monotonically approximate *implicitly* the best fitting model, [12]. But this should be sufficient: we want the best model rather than a number that measures its goodness. We show that—within the obvious constraints—every graph is realized by the structure function of some data. This means that there are data of each conceivable combination of stochastic properties. All these results are not completely precise: they hold up to a logarithmic additive error. They usher in an era of statistical inference that is *always* (almost) best rather than *expected*.

Reach of Results: 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, and in fact other model classes. Since our results hold only within additive logarithmic precision, and the equivalences of the relevant notions and results between the model classes hold up to the same precision, the results hold equally for the more general model classes.

The generality of the results are at the same time a restriction. In classical statistics one is commonly interested in model classes that are partially poorer and partially richer than the ones we consider. For example, the class of Bernoulli processes, or k-state Markov chains, is poorer than the class of computable probability density functions of moderate maximal Kolmogorov complexity α, in that the latter may contain functions that require far more complex computations than the rigid syntax of the former classes allows. Indeed, the class of computable probability density functions of even moderate complexity allows implementation of a function mim-

icking a universal Turing machine computation. On the other hand, even the lowly Bernoulli process can be equipped with a non-computable real bias in $(0,1)$, and hence the generated probability density function over n trials is not a computable function. This incomparability of the here studied algorithmic model classes, and the traditionally studied statistical model classes, means that the current results cannot be directly transplanted to the traditional setting. Indeed, they should be regarded as pristine truths that hold in a platonic world that can be used as guideline to develop analogues in model classes that are of more traditional concern, as in [8]. See also the later Remark 1.9.

1.3 Preliminaries

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*. The *length* $|x|$ of x is the number of bits in the binary string x, not to be confused with the *cardinality* $|S|$ of a finite set S. For example, $|010| = 3$ and $|\epsilon| = 0$, while $|\{0,1\}^n| = 2^n$ and $|\emptyset| = 0$. 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 y is a *proper prefix* of a binary string x 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*. There is a special type of prefix code, the *self-delimiting code*, that has the added property of being *effective* in the sense that there is an algorithm that, starting at the beginning of the code word, scanning from left to right, can determine where the code word ends. A simple example of this is the code that encodes the source word $x = x_1 x_2 \ldots x_n$ by the code word

$$\bar{x} = 1^n 0 x.$$

Using this code we define the standard self-delimiting code for x to be $x' = \overline{|x|}x$. It is easy to check that $|\bar{x}| = 2n+1$ and $|x'| = n + 2\log n + 1$. We can extend this code to pairs of strings: Let $\langle\cdot\rangle$ be a standard invertible effective one-one encoding from $\mathcal{N} \times \mathcal{N}$ to a subset of $\mathcal{N}$. For example, we can set $\langle x, y\rangle = x'y$ or $\langle x, y\rangle = \bar{x}y$. We can iterate this process to define $\langle x, \langle y, z\rangle\rangle$, and so on.

Kolmogorov Complexity: For precise definitions, notation, and results see the text [4]. 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, $K(x)$ represents the minimal amount of information required to generate x by any effective process. The conditional Kolmogorov com-

plexity $K(x|y)$ 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. For technical reasons we use a variant of complexity, so-called prefix complexity, which is associated with Turing machines for which the set of programs resulting in a halting computation is prefix free. We realize prefix complexity by considering 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. It is easy to see that $\{p : T(p) = x, x \in \{0,1\}^*\}$ is a *prefix code*. In fact, because the algorithm (in this case a Turing machine T) determines the end of the code word for the source word x (that is, the program p such that $T(p) = x$), this code is in fact *self-delimiting*. Let $T_1, T_2, \ldots$ be a standard enumeration of all prefix Turing machines with a binary input tape, for example the lexicographical length-increasing ordered syntactic prefix Turing machine descriptions, [4], and let $\phi_1, \phi_2, \ldots$ be the enumeration of corresponding functions that are computed by the respective Turing machines (T_i computes ϕ_i). These functions are the *partial recursive* functions or *computable* functions (of effectively prefix-free encoded arguments). The Kolmogorov complexity of x is the length of the shortest binary program from which x is computed.

Definition 1.1 The *prefix Kolmogorov complexity* of x is

$$K(x) = \min_{p,i}\{|\bar{i}| + |p| : T_i(p) = x\}, \tag{1.1}$$

where the minimum is taken over $p \in \{0,1\}^*$ and $i \in \{1, 2, \ldots\}$. For the development of the theory we actually require the Turing machines to use *auxiliary* (also called *conditional*) information, by equipping the machine with a special read-only auxiliary tape containing this information at the outset. Then, the *conditional version* $K(x \mid y)$ of the prefix Kolmogorov complexity of x given y (as auxiliary information) is is defined similarly as before, and the unconditional version is set to $K(x) = K(x \mid \epsilon)$.

One of the main achievements of the theory of computation is that the enumeration $T_1, T_2, \ldots$ contains a machine, say $U = T_u$, that is computationally universal in that it can simulate the computation of every machine in the enumeration when provided with its index. Expressing an index i by the shortest self-delimiting code i^* (if there is more then one such code then the notation is disambiguated in a standard manner that does not need to concern us here) for i usable by U we have: $U(\langle y, i^* p) = T_i(\langle y, p\rangle)$ for all i, p, y. We fix one such machine and designate it as the *reference universal prefix Turing machine*. Using this universal machine it is easy

to show [12]

$$K(x \mid y) = \min_q\{|q| : U(\langle y, q\rangle) = x\} + O(1) \tag{1.2}$$
$$K(x) = \min_q\{|q| : U(q) = x\} + O(1).$$

Remark 1.2 A prominent property of the prefix-freeness of the set of programs for the reference prefix Turing machine U is that we can interpret $2^{-K(x)}$ as a probability distribution. By the fundamental Kraft's inequality, see for example [1; 4], we know that if $l_1, l_2, \ldots$ are the code-word lengths of a prefix code, then $\sum_x 2^{-l_x} \leq 1$. Hence,

$$\sum_x 2^{-K(x)} \leq 1. \tag{1.3}$$

This leads to the notion of universal distribution—a rigorous form of Occam's razor—which implicitly plays an important part in the present exposition. The functions $K(\cdot)$ and $K(\cdot \mid \cdot)$, 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. The Kolmogorov complexity of an individual object was introduced by Kolmogorov [5] as an absolute and objective quantification of the amount of information in it. The information theory of Shannon [9], on the other hand, deals with *average* information *to communicate* objects produced by a *random source*. Since the former theory is much more precise, it is surprising that analogs of theorems in information theory hold for Kolmogorov complexity, be it in somewhat weaker form, see [4].

Precision: 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. In this paper we use the prefix complexity variant of Kolmogorov complexity for convenience. Actually some results are easier to prove for plain complexity. Most results presented here are precise up to an additive logarithmic term, which means that they are valid for plain complexity as well—prefix complexity exceeds plain complexity by at most a logarithmic additve term. Thus, our use of prefix complexity is important for "fine details" only.

Meaningful Information: The information contained in an individual finite object (like a finite binary string) is measured by its Kolmogorov complexity—the length of the shortest binary program that computes the object. Such a shortest program contains no redundancy: every bit is information; but is it meaningful information? If we flip a fair coin to obtain a finite binary string, then with overwhelming probability that string constitutes its own shortest program. However, also with overwhelming probability all the bits in the string are meaningless information, random noise. On the other hand, let an object x be a sequence of

observations of heavenly bodies. Then x can be described by the binary string pd, where p is the description of the laws of gravity, and d the observational parameter setting: we can divide the information in x into meaningful information p and accidental information d. The main task for statistical inference and learning theory is to distil the meaningful information present in the data. The question arises whether it is possible to separate meaningful information from accidental information, and if so, how. The essence of the solution to this problem is revealed when we rewrite (1.1) via (1.2) as follows:

$$\begin{aligned} K(x) &= \min_{p,i}\{|\bar{i}| + |p| : T_i(p) = x\} \\ &= \min_{p,i}\{2|i| + |p| + 1 : T_i(p) = x\}, \\ &= \min_{p,i}\{K(i) + |p| : T_i(p) = x\} + O(1), \end{aligned} \tag{1.4}$$

where the minimum is taken over $p \in \{0,1\}^*$ and $i \in \{1,2,\ldots\}$. In the last step we use first the equality according to (1.2), then that the fixed reference universal prefix Turing machine $U = T_u$ with $|u| = O(1)$, and finally that $U(i^*p) = T_i(p)$ for all i and p. Here i^* denotes the shortest self-delimiting program for i; therefore $|i^*| = K(i)$. The expression (1.4) emphasizes the two-part code nature of Kolmogorov complexity. In the example

$$x = 10101010101010101010101010$$

we can encode x by a small Turing machine printing a specified number of copies of the pattern "01" which computes x from the program "13." This way, $K(x)$ is viewed as the shortest length of a two-part code for x, one part describing a Turing machine T, or *model*, for the *regular* aspects of x, and the second part describing the *irregular* aspects of x in the form of a program p to be interpreted by T. The regular, or "valuable," information in x is constituted by the bits in the "model" while the random or "useless" information of x constitutes the remainder. This leaves open the crucial question: How to choose T and p that together describe x? In general, many combinations of T and p are possible, but we want to find a T that describes the meaningful aspects of x.

Data and Model: We consider only finite binary data strings x. Our model class consists of Turing machines T that enumerate a finite set, say S, such that on input $p \leq |S|$ we have $T(p) = x$ with x the pth element of T's enumeration of S, and $T(p)$ is a special *undefined* value if $p > |S|$. The "best fitting" model for x is a Turing machine T that reaches the minimum description length in (1.4). Such a machine T embodies the amount of useful information contained in x, and we have divided a shortest program x^* for x into parts $x^* = T^*p$ such that T^* is a shortest self-delimiting program for T. Now suppose we consider only low complexity finite-set models, and under these constraints the shortest two-part description happens to be longer than the shortest one-part description. For example, this can happen if the data is generated by a model that is too complex to be in the contemplated model class. Does the model minimizing the two-part description still capture all

(or as much as possible) meaningful information? Such considerations require study of the relation between the complexity limit on the contemplated model classes, the shortest two-part code length, and the amount of meaningful information captured.

1.4 Algorithmic Sufficient Statistic

In the following we will distinguish between "models" that are finite sets, and the "shortest programs" to compute those models that are finite strings. Such a shortest program is in the proper sense a statistic of the data sample as defined before. In a way this distinction between "model" and "statistic" is artificial, but for now we prefer clarity and unambiguousness in the discussion.

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 $U(p) = \langle x_1, \langle x_2, \ldots, \langle x_{n-1}, x_n \rangle \ldots \rangle \rangle$. Another concept we need is $K(x|S)$, the length of the shortest binary program that computes x from a listing of all the elements of S. We are now about to formulate the central notions 'x is typical for S' and 'S is optimal for x'.

1.4.1 Typical Elements

Consider a string x of length n and prefix complexity $K(x) = k$. We identify the *structure* or *regularity* in x that are to be summarized with a set S of which x is a *random* or *typical* member: given S containing x, the element x cannot be described significantly shorter than by its maximal length index in S, that is, $K(x \mid S) \geq \log |S| + O(1)$. Formally,

Definition 1.3 Let $\beta \geq 0$ be an agreed upon, fixed, constant. A finite binary string x is a *typical* or *random* element of a set S of finite binary strings, if $x \in S$ and

$$K(x \mid S) \geq \log |S| - \beta. \tag{1.5}$$

We will not indicate the dependence on β explicitly, but the constants in all our inequalities ($O(1)$) will be allowed to be functions of this β.

This definition requires a finite S. In fact, since $K(x \mid S) \leq K(x) + O(1)$, it limits the size of S to $O(2^k)$. Note that the notion of typicality is not absolute but depends on fixing the constant implicit in the O-notation.

Example 1.4 Consider the set S of binary strings of length n whose every odd position is 0. Let x be an element of this set in which the subsequence of bits in even positions is an incompressible string. Then x is a typical element of S (or by with some abuse of language we can say S is typical for x). But x is also a typical element of the set $\{x\}$.

1.4.2 Optimal Sets

Let x be a binary data string of length n. For every finite set $S \ni x$, we have $K(x) \leq K(S) + \log |S| + O(1)$, since we can describe x by giving S and the index of x in a standard enumeration of S. Clearly this can be implemented by a Turing machine computing the finite set S and a program p giving the index of x in S. The size of a set containing x measures intuitively the number of properties of x that are represented: The largest set is $\{0,1\}^n$ and represents only one property of x, namely, being of length n. It clearly "underfits" as explanation or model for x. The smallest set containing x is the singleton set $\{x\}$ and represents all conceivable properties of x. It clearly "overfits" as explanation or model for x.

There are two natural measures of suitability of such a set as a model for x. We might prefer either the simplest set, or the smallest set, as corresponding to the most likely structure 'explaining' x. Both the largest set $\{0,1\}^n$ (having low complexity of about $K(n)$) and the singleton set $\{x\}$ (having high complexity of about $K(x)$), while certainly statistics for x, would indeed be considered poor explanations. We would like to balance simplicity of model versus size of model. Both measures relate to the optimality of a two-stage description of x using a finite set S that contains it. Elaborating on the two-part code:

$$\begin{aligned} K(x) \leq K(x,S) &\leq K(S) + K(x \mid S) + O(1) \\ &\leq K(S) + \log |S| + O(1), \end{aligned} \tag{1.6}$$

where only the final substitution of $K(x \mid S)$ by $\log |S| + O(1)$ uses the fact that x is an element of S. The closer the right-hand side of (1.6) gets to the left-hand side, the better the description of x is. This implies a trade-off between meaningful model information, $K(S)$, and meaningless "noise" $\log |S|$. A set S (containing x) for which (1.6) holds with equality

$$K(x) = K(S) + \log |S| + O(1), \tag{1.7}$$

is called *optimal*. A data string x can be typical for a set S without that set S being optimal for x. This is the case precisely when x is typical for S (that is $K(x|S) = \log S + O(1)$) while $K(x,S) > K(x)$.

1.4.3 Sufficient Statistic

A *statistic* of the data $x = x_1 \ldots x_n$ is a function $f(x)$. Essentially, every function will do. For example, $f_1(x) = n$, $f_2(x) = \sum_{i=1}^{n} x_i$, $f_3(x) = n - f_2(x)$, and $f_4(x) = f_2(x)/n$, are statistics. A "sufficient" statistic of the data contains all information in the data about the model. In introducing the notion of sufficiency in classical statistics, Fisher [2] stated: "The statistic chosen should summarize the whole of the relevant information supplied by the sample. This may be called the Criterion of Sufficiency ... In the case of the normal curve of distribution it is evident that the second moment is a sufficient statistic for estimating the standard

deviation." For example, in the Bernoulli model (repeated coin flips with outcomes "0" and "1" according to fixed bias), the statistic f_4 is sufficient. It gives the mean of the outcomes and estimates the bias of the Bernoulli process, which is the only relevant model information. For the classical (probabilistic) theory see, for example, [1]. In [3] an algorithmic theory of sufficient statistic (relating individual data to individual model) was developed and its relation with the probabilistic version established. The algorithmic basics are as follows: Intuitively, a model expresses the essence of the data if the two-part code describing the data consisting of the model and the data-to-model code is as concise as the best one-part description.

Mindful of our distinction between a finite set S and a program that describes S in a required representation format, we call a shortest program for an optimal set with respect to x an *algorithmic sufficient statistic* for x. Furthermore, among optimal sets, there is a direct trade-off between complexity and logsize, which together sum to $K(x) + O(1)$. Equality (1.7) is the algorithmic equivalent dealing with the relation between the individual sufficient statistic and the individual data sample, in contrast to the probabilistic notion in, for example, [1].

Example 1.5 It can be shown that the set S of Example 1.4 is also optimal, and so is $\{x\}$. Sets for which x is typical form a much wider class than optimal sets for x: the set $\{x, y\}$ is still typical for x but with most y, it will be too complex to be optimal for x.

For a perhaps less artificial example, consider complexities conditional on the length n of strings. Let y be a random string of length n, let S_y be the set of strings of length n which have 0's exactly where y has, and let x be a random element of S_y. Then x has about 25% 1's, so its complexity is much less than n. The set S_y has x as a typical element, but is too complex to be optimal, since its complexity (even conditional on n) is still n.

An algorithmic sufficient statistic is a sharper individual notion than a probabilistic sufficient statistic. An optimal set S associated with x (the shortest program computing S is the corresponding sufficient statistic associated with x) is chosen such that x is maximally random with respect to it. That is, the information in x is divided in a relevant structure expressed by the set S, and the remaining randomness with respect to that structure, expressed by x's index in S of $\log |S|$ bits. The shortest program for S is itself alone an algorithmic definition of structure, without a probabilistic interpretation.

Optimal sets with the shortest program (or rather that shortest program) is the *algorithmic minimal sufficient statistic* of x. Formally, this is the shortest program that computes a finite set S such that (1.7) holds.

Example 1.6 Sufficient Statistic: Let us look at a coin toss example. Let k be a number in the range $0, 1, \ldots, n$ of complexity $\log n + O(1)$ given n and let x be a string of length n having k ones of complexity $K(x \mid n, k) \geq \log \binom{n}{k}$ given n, k. This x can be viewed as a typical result of tossing a coin with a bias about $p = k/n$. A two-part description of x is given by the number k of 1's in x first, followed by

the index $j \leq \log |S|$ of x in the set S of strings of length n with k 1's. This set is optimal, since $K(x \mid n) = K(x, k \mid n) = K(k \mid n) + K(x \mid k, n) = K(S) + \log |S|$.

Example 1.7 Hierarchy of Sufficient Statistics: In a picture like the "Mona Lisa", to borrow an example from [1], we can be interested in the image depicted, but also in the underlying color pattern or pattern of brush strokes. Each of these aspects suggest that there is a particular "model level" at which there is a sufficient statistic for that aspect. An expert trying to attribute a painting may aim at finding sufficient statistics for many such aspects. This is the intuition we try to capture. Let us now try to obtain a formal version of a situation with many sufficient statistics.

All the information in an object, like a picture, is described by a binary string x of length, say, $n = ml$. Chop x into l substrings x_i $(1 \leq i \leq l)$ of equal length m each. Let k_i denote the number of ones in x_i. Each such substring metaphorically represents a patch of, say, color. The intended color, say "cobalt blue", is indicated by the number of ones in the substring. The actual color depicted may be typical cobalt blue or less typical cobalt blue. The smaller the randomness deficiency of substring x_i in the set of all strings of length m containing precisely k_i ones, the more typical x_i is, the better it achieves a typical cobalt blue color. The metaphorical "image" depicted by x is $\pi(x)$, defined as the string $k_1 k_2 \ldots k_l$ over the alphabet $\{0, 1, \ldots, m\}$, the set of colors available. We can now consider several statistics for x.

Let $X \subseteq \{0, 1, \ldots, m\}^l$ (the set of possible realizations of the target image), and let Y_i for $i = 0, 1, \ldots, m$ be the set of binary strings of length m with i ones (the set of realizations of target color i). Consider the set

$$S = \{y : \pi(y) \in X, y_i \in Y_{k_i} \text{ for all } i = 1, \ldots, l\}$$

One possible application of these ideas are to gouge how good the picture is with respect to the given summarizing set S. Assume that $x \in S$. The set S is then a statistic for x that captures both the colors of the patches and the image, that is, the total picture. If the randomness deficiency $\delta(x \mid S)$ is small, then S is a sufficient statistic of x. This means that x perfectly expresses the meaning aimed for by he image and the true color aimed for in everyone of the color patches.

Another possible application of the theory is to find a good summarization of the meaningful information in a given picture. Let x be a string, and let $S \ni x$ be the set above that has all the randomness deficiencies equal zero. Clearly, S summarizes the relevant information in x since it captures both image and coloring, that is, the total picture. But we can distinguish more sufficient statistics. The set

$$S_1 = \{y : \pi(y) \in X\}$$

is a statistic that captures only the image. It can be sufficient only if all colors used in the picture x are typical (have small randomness deficiency). The set

$$S_2 = \{y : y_i \in Y_{k_i} \text{ for all } i = 1, \ldots, l\}$$

is a statistic that captures the color information in the picture. It can be sufficient only if the image is typical. Finally the set

$$A_i = \{y : y_i \in Y_{k_i}\}$$

is a statistic that captures only the color of patch y_i in the picture. It can be sufficient only if $K(i) \approx 0$ and all the other color applications and the image are typical.

1.5 Structure Functions

We will prove that there is a close relation between functions describing three, a priori seemingly unrelated, aspects of modeling individual data, depicted in Figure 1.1.

1.5.1 Model Fitness

For every finite set $S \subseteq \{0,1\}^*$ containing x we have

$$K(x|S) \leq \log |S| + O(1). \tag{1.8}$$

Indeed, consider the self-delimiting 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

$$\delta(x|S) = \log |S| - K(x|S), \tag{1.9}$$

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

$$\beta_x(\alpha) = \min_S \{\delta(x|S) : S \ni x,\ K(S) \leq \alpha\}, \tag{1.10}$$

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. We therefore like to call $\beta_x(\alpha)$ the *best-fit estimator*. 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 [12]: Consider strings of length n and let S be a subset of such strings. We view a *property* of elements in S as a function $f_P : S \to \{0,1\}$. If $f_P(x) = 1$ then x has the property represented by f_P and if $f_P(x) = 0$ then x does not have the property.

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

(ii) Let f_P be any property that holds with probability at least $1 - 1/2^{\delta(n)}$ for

the elements of S. Then, every such f_P holds simultaneously for every $x \in S$ with $\delta(x|S) \leq \delta(n) - K(f_P|S) - O(1)$.

Example 1.8 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 $K(S) \leq \alpha$ and with small $\delta(x|S)$, and consider a 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 with high probability the randomness deficiency of x' in S is small, 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.

1.5.2 Maximum Likelihood

Kolmogorov at a conference in Tallinn 1974 (no written version) and in a talk at the Moscow Mathematical Society in the same year of which the abstract [6] is reproduced at the top of this article (the only writing by Kolmogorov about this circle of ideas) proposed the following function: The *Kolmogorov structure* function h_x of given data x is defined by

$$h_x(\alpha) = \min_S \{\log |S| : S \ni x,\ K(S) \leq \alpha\}, \tag{1.11}$$

where $S \ni x$ is a contemplated model for x, and α is a non-negative 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 (1.6), and hence $K(x) \leq \alpha + h_x(\alpha) + O(1)$, 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 a sufficient statistic, and it is *minimal* for the least such α (see above and [1; 3]).

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 [10; 3]. 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,

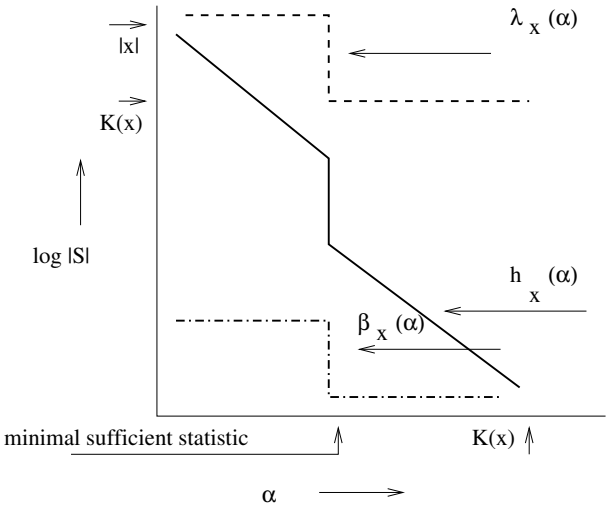

Figure 1.1 Structure functions $h_x(i), \beta_x(\alpha), \lambda_x(\alpha)$, and minimal sufficient statistic.

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 $K(P) = \min_i\{K(i) : \text{Turing machine } T_i \text{ computes } P\}$. A string x is typical for a distribution P if the randomness deficiency $\delta(x \mid P) = -\log P(x) - K(x \mid P)$ is small. The conditional complexity $K(x \mid P)$ is defined as follows. Say that a function A approximates P if $|A(y,\epsilon) - P(y)| < \epsilon$ for every y and every positive rational ϵ. Then $K(x \mid P)$ 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 $K(P) - \log P(x) \leq K(x) + c$. 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 x is under the hypothesis P and the mapping $x \mapsto P_{\min}$ where $P_{\min}$ minimizes $-\log P(x)$ over P with $K(P) \leq \alpha$ is a *maximum likelihood estimator*, see figure 1.2. Our results thus imply that that maximum likelihood estimator always returns a hypothesis with minimum randomness deficiency.

It is easy to show that for every data string x and a contemplated finite set model for it, there is an almost equivalent computable probability density function model. Conversely, for every data string x and a contemplated computable probability density function model for it, there is a finite set model for x that has no worse complexity, randomness deficiency, and worst-case data-to-model code for x, up to additive logarithmic precision. (See [12].)

1.5.3 Minimum Description Length

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

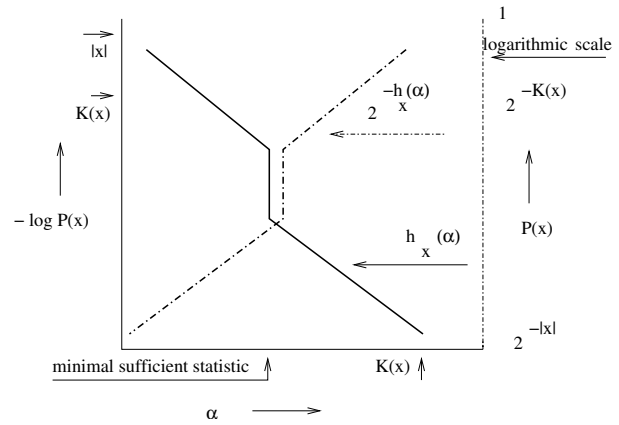

Figure 1.2 Structure function $h_x(i) = \min_P\{-\log P(x) : P(x) > 0,\ K(P) \leq i\}$ with P a computable probability density function, with values according to the left vertical coordinate, and the maximum likelihood estimator $2^{-h_x(i)} = \max\{P(x) : P(x) > 0,\ K(P) \leq i\}$, with values according to the right-hand side vertical coordinate.

given by the *MDL function* or *MDL setimator*:

$$\lambda_x(\alpha) = \min_S\{\Lambda(S) : S \ni x,\ K(S) \leq \alpha\}, \tag{1.12}$$

where $\Lambda(S) = \log|S| + K(S) \geq K(x) - O(1)$ is the total length of two-part code of x with help of model S. Clearly, $\lambda_x(\alpha) \leq h_x(\alpha) + \alpha + O(1)$, but a priori it is still possible that $h_x(\alpha') + \alpha' < h_x(\alpha) + \alpha$ for $\alpha' < \alpha$. In that case $\lambda_x(\alpha) \leq h_x(\alpha') + \alpha' < h_x(\alpha) + \alpha$. However, in [12] it is shown that $\lambda_x(\alpha) = h_x(\alpha) + \alpha + O(\log n)$ for all x of length n. Even so, this doesn't mean that a set S that witnesses $\lambda_x(\alpha)$ in the sense that $x \in S$, $K(S) \leq \alpha$, and $K(S) + \log|S| = \lambda_x(\alpha)$, also witnesses $h_x(\alpha)$. It can be the case that $K(S) \leq \alpha - r$, and $\log|S| = h_x(\alpha) + r$ for arbitrarily large $r \leq n$.

This function $\lambda_x(\alpha)$ is the celebrated two-part Minimum Description Length code length with the model-code length restricted to at most α.

1.6 Overview of Results

The most fundamental result in [12] is the equality

$$\beta_x(\alpha) = h_x(\alpha) + \alpha - K(x) = \lambda_x(\alpha) - K(x) \tag{1.13}$$

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 1.12 below); but we have shown [12] that $\beta_x(\alpha)$ is neither upper nor lower semi-computable (not even within a great tolerance). 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.

Remark 1.9 What we call 'maximum likelihood' in the form of h_x is really 'maximum likelihood' under a complexity constraint α on the models' as in $h_x(\alpha)$. In statistics, it is a well-known fact that maximum likelihood often fails (dramatically overfits) when the models under consideration are of unrestricted complexity (for example, with polynomial regression with Gaussian noise, or with Markov chain model learning, maximum likelihood will always select a model with n parameters, where n is the size of the sample—and thus typically, maximum likelihood will dramatically overfit, whereas for example MDL typically performs well). The equivalent, in our setting, is that allowing models of unconstrained complexity for data x, say complexity $K(x)$, will result in the ML-estimator $h_x(K(x)+O(1)) = 0$—the witness model being the trivial, maximally overfitting, set $\{x\}$. In the MDL case, on the other hand, there may be a long constant interval with the MDL estimator $\lambda_x(\alpha) = K(x)$ ($\alpha \in [\alpha_1, K(x)]$) where the length of the two-part code doesn't decrease anymore. Selecting the least complexity model witnessing this function value we obtain the, very significant, algorithmic *minimal* sufficient statistic. In this sense, MDL augmented with a bias for the least complex explanation, which we may call the 'Occam's Razor MDL', is superior to maximum likelihood and resilient to overfitting. If we don't apply bias in the direction of simple explanations, then MDL may be just as prone to overfitting as is ML. For example, if x is a typical random element of $\{0,1\}^n$, then $\lambda_x(\alpha) = K(x)+O(1)$ for the entire interval $K(n) + O(1) \leq \alpha \leq K(x) + O(1) \approx n$. Choosing the model on the left side, of simplest complexity, of complexity $K(n)$ gives us the best fit with the correct model $\{0,1\}^n$. But choosing a model on the right side, of high complexity, gives us a model $\{x\}$ of complexity $K(x) + O(1)$ that completely overfits the data by modelling all random noise in x (which in fact in this example almost completely consists of random noise).

Thus, it should be emphasized that 'ML = MDL' really only holds if complexities are constrained to a value α (that remains fixed as the sample size grows—note that in the Markov chain example above, the complexity grows linearly with the sample size); it certainly does not hold in an unrestricted sense (not even in the algorithmic setting).

Remark 1.10 In a sense, h_x is more strict than λ_x: A set that witnesses $h_x(\alpha)$ also witnesses $\lambda_x(\alpha)$ but not necessarily vice versa. However, at those complexities α where $\lambda_x(\alpha)$ drops (a little bit of added complexity in the model allows a shorter description), the witness set of λ_x is also a witness set of h_x. But if λ_x stays constant in an interval $[\alpha_1, \alpha_2]$, then we can trade-off complexity of a witness set versus its cardinality, keeping the description length constant. This is of course not possible with h_x where the cardinality of the witness set at complexity α is fixed at $h_x(\alpha)$.

The main result can be taken as a foundation and justification of common statistical principles in model selection such as maximum likelihood or MDL. 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), see [12]. (This establishes the significance of (1.13), since it shows that $\lambda_x(\alpha) \gg K(x)$ is common for x, α pairs—in which case the more or less easy fact that $\beta_x(\alpha) = 0$ for $\lambda_x(\alpha) = K(x)$ is not applicable, and it is a priori unlikely that (1.13) holds: Why should minimizing a set containing x also minimize its randomness deficiency? Surprisingly, it does!) We have exhibited 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.

Example 1.11 "Positive" and "Negative" Individual Randomness: In [3] we showed the existence of strings 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 one part 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). 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 [6] (see the current abstract) 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 $K(x) \geq n$, being typical for the set $\{0,1\}^n$, or the uniform probability distribution over that set, while this set or probability distribution has complexity $K(n) + O(1) = O(\log n)$). We depict the distinction in Figure 1.3. In simple terms: a data string of high Kolmogorov complexity is *positively random* if the simplest satisfactory explanation (sufficient statistic) has low complexity, and it therefore is the typical outcome of a simple random process. Another data string of the same length and the same complexity is *negatively random*, if the simplest satisfactory explanation (sufficient statistic) has high complexity: It can only be the typical outcome of a complex random process.

In [12] it is shown that for every length n and every complexity $k \leq n + K(n) + O(1)$ (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 $\beta_x(i) \geq n - k \pm O(\log n)$ for every $i \leq \alpha \pm O(\log n)$ and $\beta_x(i) \pm O(\log n)$ for every $i > \alpha \pm O(\log n)$. 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 $\sqrt{n} \pm O(\log n)$ randomness deficiency with respect to $\{0,1\}^n$ or, in fact, every other

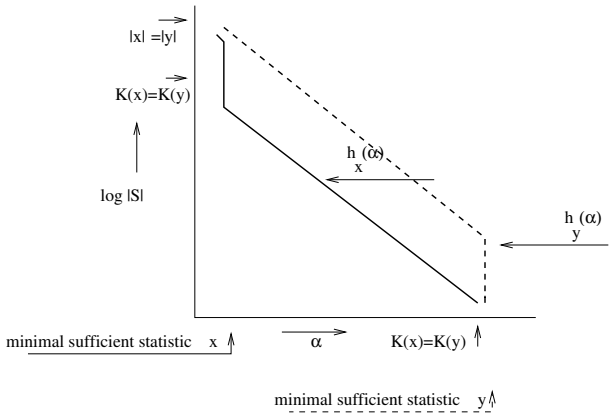

Figure 1.3 Data string x is "positive random" or "stochastic" and data string y is "negative random" or "non-stochastic".

finite set of complexity less than $n - O(\log n)$!

1.7 Relation to MDL

(i) 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. The *best explanation* is defined to be the (p, S) for which $\delta(x|S)$ is minimal. Since the function $\delta(x|S)$ is not computable, we cannot find the best explanation in a finite amount of time. Another reason for this is that the programs use unknown computation time and thus we can never be certain that we have found all possible explanations.

Compare this indirect method with the direct one: after step t of dovetailing select (p, S) for which $\log |S| - K^t(x|S)$ is minimum among all programs p that up to this time have printed a set S containing x, where $K^t(x|S)$ is the approximation of $K^t(x|S)$ obtained after t steps of dovetailing, that is, $K^t(x|S) = \min\{|q| : U \text{ on input } \langle q, S\rangle \text{ prints } x \text{ in at most } t \text{ steps}\}$. Let (q_t, B_t) stand for that model. This time the same hypothesis can be declared best twice. However from some moment onward the explanation (q_t, B_t) which is declared best does not change anymore.

Why do we prefer the indirect method to the direct one? The explanation is that in practice we deal often with t that are much less than the time of stabilization of both L_t and B_t. For small t, the model L_t is better than B_t in the following respect: L_t has some guarantee of goodness, as we know that $\delta(x|L_t) + K(x) \leq |p_t| + \log |L_t| + O(1)$. That is, we know that the sum of deficiency of x in L_t and $K(x)$ is less than some known value. In contrast, the model B_t has no

guarantee of goodness at all: we do not know any upper bound neither for $\delta(x|B_t)$, nor for $\delta(x|B_t) + K(x)$.

Our result in [12] implies that the indirect method of MDL gives not only some guarantee of goodness but also that, in the limit, that guarantee approaches the value it upper bounds, that is, approaches $\delta(x|L_t) + K(x)$, and $\delta(x|L_t)$ itself is not much greater than $\delta(x|B_t)$ (except for some values of α called "critical" in [12]). That is, in the limit, the method of MDL will yield an explanation that is only a little worse than the best explanation.

(ii) If $S \ni x$ is a smallest set such that $K(S) \leq \alpha$, then S can be converted into a best strategy of complexity at most α, to predict the successive bits of x given the preceding ones, see the "Snooping curve" example in [12]. Interpreting "to explain" as "to be able to predict well", MDL in the sense of sets witnessing $\lambda_x(\alpha)$ gives indeed a good explanations at every complexity level α.

(iii) In statistical applications of MDL [7], MML [14], and related methods, one selects the model in a given model class that minimizes the sum of the model code length and the data-to-model code length; in modern versions one chooses a minimizes the data-to-model code length (ignoring the model code length). In [12] we have shown that these methods are almost equivalent in the case when the model class consists of *all* the models with certain model-complexity constraints. In contrast, ultimate compression of the two-part code, proposed in [13], may be achieved by a model for which the data is not typical, even when the model class consists of *all* the models with certain model-complexity constraints. By ultimate compression of the two-part code we mean minimizing $K(A) + K(x|A)$ over all models in the model class. For instance, let x be a string of length n and complexity about $n/2$ for which $\beta_x(O(\log(n)) = n/4 + O(\log(n))$. Such a string exists by the results in [12]. Moreover, let the model class consists of all finite sets containing x of complexity at most $\alpha = O(\log n)$. Then for the model $A_0 = \{0,1\}^n$ we have $K(A_0) = O(\log n)$ and $K(x|A_0) = n/2 + O(\log n)$ thus the sum $K(A_0) + K(x|A_0) = n/2 + O(\log n)$ is minimal up to a term $O(\log n)$. However, the randomness difficiency of x in A_0 is about $n/2$, which is much bigger than the minimum $\beta_x(O(\log(n)) \approx n/4$. For the model A_1 witnessing $\beta_x(O(\log(n)) \approx n/4$ we also have $K(A_1) = O(\log n)$ and $K(x|A_1) = n/2 + O(\log n)$. However, it has smaller cardinality: $\log |A_1| = 3n/4 + O(\log n)$ and hence smaller randomness deficiency.

The same happens also for other model classes. Consider, for instance, as the model class, the Bernoulli processes with rational bias p for outcome "1" ($0 \leq p \leq 1$) to generate binary strings of length n. Suppose we look for the model minimizing the code length of the model plus data given the model: $K(p|n) + K(x|p,n)$. Let the data be $x = 00\ldots0$. Then the model corresponding to probability $p = \frac{1}{2}$ compresses the data code to $K(x \mid n,p) = O(1)$ bits and $K(p|n) = O(1)$. But we find about the same code length if we take $p' = 0$. Thus we have no basis to distinguish between the two, while obviously the second possibility is preferable. This shows that ultimate compression of the two-part code, here resulting in $K(p|n) + K(x|n,p)$, may yield a model P for which the data has large randomness deficiency ($-\log P(x) - K(x \mid n,p) = n$ for $p = \frac{1}{2}$) and hence is atypical.

However, in the structure functions $h_x(\alpha)$ and $\lambda_x(\alpha)$ the data-to-model code for the model $p = \frac{1}{2}$ is $-\log P(x) = -\log(\frac{1}{2})^n = n$ bits, while $p = 0$ results in $-\log 1^n = 0$ bits. Choosing the shortest data-to-model code results in the minimal randomness deficiency, as in (the generalization to probability distributions of) our main theorem in [12].

(iv) Another question arising in MDL or maximum likelihood (ML) estimation is its performance if the "true" model is not part of the contemplated model class. Given certain data, why would we assume they are generated by probabilistic or deterministic processes? They have arisen by natural processes most likely not conforming to mathematical idealization. Even if we can assume the data arose from a process that can be mathematically formulated, such situations arise if we restrict modeling of data arising from a "complex" source (conventional analogue being data arising from $2k$-parameter sources) by "simple" models (conventional analogue being k-parameter models). Again, our main result in [12] shows that, within the class of models of maximal complexity α, these constraints we still select a simple model for which the data is maximally typical. This is particularly significant for data x if the allowed complexity $\alpha$is significantly below the complexity of the Kolmogorov minimal sufficient statistic, that is, if $h_x(\alpha) + \alpha \gg K(x) + c$. This situation is potentially common, for example if we have a small data sample generated by a complex process. For a data sample that is very large relative to the complexity of the process generating it, this will typically not be the case and the structure function will drop to the sufficiency line early on.

Relation to Maximum Likelihood Estimation: 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: $\delta(x|S) \leq \beta_x(\alpha - O(\log n)) + O(\log n)$.

1.8 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 1.12 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 $\phi(x, t + 1) \leq \phi(x, t)$ 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 or rational).

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 statements):

- The functions $h_x(\alpha)$ and $\lambda_x(\alpha)$ are upper semi-computable but they are not computable up to any reasonable precision.
- 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, not even to any reasonable precision, 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 up to any reasonable precision.

The precise forms of these quite strong non-computability and non-approximability results are given in [12].

Acknowledgement

I thank Peter Gács, John Tromp, and Nikolai Vereshchagin, my co-authors of [3; 12].

Bibliography

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

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

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

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

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

[6] A.N. Kolmogorov. Complexity of Algorithms and Objective Definition of Randomness, *Uspekhi Mat. Nauk* 29:4(1974),155. (Russian abstract talk at Moscow Math. Soc. meeting 4/16/1974. Translation by L.A. Levin in November 2002.)

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

[8] J.J. Rissanen, Kolmogorov's Structure Function for Probability Models, Proc. IEEE Information Theory Workshop, IEEE Press, 2002, 98–99.

[9] C.E. Shannon. The mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.

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

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

[12] N.K. Vereshchagin and P.M.B. Vitányi, Kolmogorov's structure functions and an application to the foundations of model selection, *Proc. 47th IEEE Symp. Found. Comput. Sci.*, 2002, 751–760. Submitted to *IEEE Trans. Inform. Theory.*

[13] 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.

[14] 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.

Index