Algorithmic “Kolmogorov” Complexity

[for the encyclopedia www.scholarpedia.org]

Marcus Hutter

RSISE @ ANU and SML @ NICTA
Canberra, ACT, 0200, Australia
marcus@hutter1.net   www.hutter1.net

19 January 2008

Abstract

The information content or complexity of an object can be measured by the length of its shortest description. For instance the string ‘01010101010101010101010101010101’ has the short description “16 repetitions of 01”, while ‘11001000011000011101111011101100’ presumably has no simpler description other than writing down the string itself. More formally, the Algorithmic “Kolmogorov” Complexity (AC) of a string x is defined as the length of the shortest program that computes or outputs x, where the program is run on some fixed reference universal computer.

Contents

Keywords

algorithmic information theory, prefix code, prefix Turing machine, Universal Turing machine, Kolmogorov complexity, plain complexity, prefix complexity.

1 Overview

Section 2 introduces the notion of the complexity of an effective code in general and the concept of algorithmic "Kolmogorov" complexity in particular. The section is deliberately vague about the precise underlying Turing machine model, its input and output tape (alphabet) and their interpretation. There are many "minor" variants of algorithmic complexity, based on "minor" variants of the Turing machine model, whether the tape heads are monotone or not, there are blank symbol or not, halting requirement or not, etc. References for some of them can be found in Section 6. The subtle differences are beyond the scope of this article, so only the most popular one, the prefix "Chaitin" complexity, is described in more detail in Section 3, together with its most important properties in Section 4. Section 5 briefly describes other complexities, related concepts, and major variants, and points to other Scholarpedia articles for details.

2 Kolmogorov Complexity

Kolmogorov complexity formalizes the concept of simplicity and/or complexity. Intuitively, a string is simple if it can be described in a few words, like "the string of one million ones", and is complex if there is no such short description, like for a random string whose shortest description is specifying it bit by bit. Typically one is only interested in descriptions or *codes* that are effective in the sense that decoders are algorithms on some computer. According to Church's Thesis, the intuitive notion of 'computability' in its widest sense is captured by the formal notion of 'computable by a Turing machine', and no formal mechanism can define a stronger notion of 'computable.' The function $K(\cdot)$ below, though defined in terms of a particular machine model, is machine-independent up to an additive constant and acquires 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.

More formally, we say that (program) p is a description of string x on Turing machine T if $T(p) = x$. The length of the shortest description is denoted by

$$K_T(x) := \min_p \{\ell(p) : T(p) = x\}$$

where $\ell(p)$ is the length of p measured in bits. This complexity measure depends on T, and one may ask whether there exists a Turing machine which leads to the shortest codes among *all* Turing machines for *all* x. Remarkably, there exists a Turing machine (the universal one, U) which "nearly" has this property: If p is the shortest description of x on $T = T_i$, then $\langle i \rangle p$ is a description of x under U, where $\langle i \rangle$ is some binary (prefix) code of i. Hence

$$K_U(x) \leq K_T(x) + c_{TU}$$

with $c_{TU} = \ell(\langle i \rangle)$, and similarly for other choices of universal Turing machines. The shortest description of x under U is at most a constant number of bits longer than the shortest description under T. The statement and proof of this invariance theorem in Solomonoff (1964), Kolmogorov (1965) and Chaitin (1969) is often regarded as the birth of Algorithmic Information Theory. Furthermore, for each pair of universal Turing machines U' and U'' satisfying the invariance theorem, the complexities coincide up to an additive constant:

$$|K_{U'}(x) - K_{U''}(x)| \leq c_{U'U''}$$

Since $c_{U'U''}$ is essentially a compiler/interpreter constant, it is "small" for "natural" universal Turing machines U' and U''. Therefore, it is customary to write $O(1)$ for terms like $c_{U'U''}$ that only depend on the choice of universal Turing machines, but which are independent of the strings under consideration.

The two bounds above may be termed 'Kolmogorov's Thesis': The intuitive notion of 'shortest effective code' in its widest sense is captured by the formal notion of Kolmogorov complexity, and no formal mechanism can yield an essentially shorter code. Note that the shortest code is one for which there is a general decompressor: The Kolmogorov complexity establishes the ultimate limit on how short a file can be compressed by a general purpose compressor.

3 Prefix Complexity

There are many variants of algorithmic complexity, mainly for technical reasons: The historically first "plain" complexity, the now more important "prefix" complexity, and many others. Most of them coincide within an additive term logarithmic in the length of the involved strings. In this article, K is used for the prefix complexity variant based on the universal prefix Turing machine.

Prefix Turing machine. A prefix Turing machine is defined as a Turing machine with one unidirectional input tape, one unidirectional output tape, and some bidirectional work tapes. Input tapes are read only, output tapes are write only, and unidirectional tapes are those where the head can only move from left to right. All tapes are binary (no blank symbol), work tapes initially filled with zeros. We say T halts on input p with output x, and write $T(p) = x$ if p is to the left of the input head and x is to the left of the output head after T halts. The set of p on which T halts forms a prefix code. Such codes p are called *self-delimiting* programs.

The table of rules of a Turing machine T can be encoded in a canonical way as a binary string, which we denote by $\langle T \rangle$. Hence, the set of Turing machines $\{T_1, T_2, ...\}$ can be effectively enumerated. There are so-called universal Turing machines that can "simulate" all other Turing machines. We define a particular one below, which also allows for side information y.

Universal prefix Turing machine. There exists a universal prefix Turing machine U which simulates prefix Turing machine T_i with input $y'q$ if fed with input $y'i'q$, i.e.

$$U(y'i'q) = T_i(y'q) \quad \forall i,q.$$

where $x' = \langle x \rangle$ is a prefix code of x with $\ell(x') \leq \ell(x) + 2\log \ell(x) + O(1)$.

We call this particular U the *reference* universal Turing machine. Note that for p not of the form $y'i'q$, $U(p)$ does not halt. The price one has to pay for the existence of a universal Turing machine is the undecidability of the halting problem (Turing 1936).

Prefix complexity. The (conditional) prefix Kolmogorov complexity is defined as the shortest program p, for which the universal prefix Turing machine U outputs x (given y):

$$K(x) \;:=\; \min_p\{\ell(p) : U(p) = x\}, \quad K(x|y) \;:=\; \min_p\{\ell(p) : U(y'p) = x\}$$

For general (non-string) objects (like computable functions) one can specify some default coding $\langle\cdot\rangle$ and define $K(\mathit{object}) := K(\langle \mathit{object} \rangle)$, especially for numbers and pairs, e.g. we abbreviate $K(x,y) := K(\langle x,y \rangle) = K(x'y)$.

4 Properties of Prefix Complexity

The most important information-theoretic properties of K are:

(1) Incomputability:
$K : \{0,1\} \to I\!N$ is approximable from above in the limit, but not computable

(2) Upper bounds: $K(x) \;\leq\; \ell(x) + 2\log\, \ell(x), \qquad K(n) \;\leq\; \log n + 2 \log\log\, n$

(3) Kraft's inequality implies: $\sum_x 2^{-K(x)} \;\leq\; 1$,

(4) Lower bounds: $K(x) \;\geq\; \ell(x)$ for 'most' x, $\quad K(n) \to \infty$ for $n \to \infty$

(5) Extra information: $K(x|y) \;\leq\; K(x) \;\leq\; K(x,y)$

(6) Subadditivity: $K(xy) \;\leq\; K(x,y) \;\leq\; K(x) + K(y|x) \;\leq\; K(x) + K(y)$

(7) Symmetry of information: $K(x,y) \;=\; K(x|y,K(y)) + K(y) \;=\; K(y,x)$

(8) Information non-increase:
$K(f(x)) \;\leq\; K(x) + K(f)$ for computable $f : \{0,1\}^* \to \{0,1\}^*$

(9) Coding relative to probability distribution / MDL bound:
$K(x) \;\leq\; -\log P(x) + K(P)$ if $P : \{0,1\}^* \to [0,1]$ is comp. and $\sum_x P(x) \leq 1$

where log is the binary logarithm and all (in)equalities hold within an additive constant.

Explanation. All (in)equalities remain valid if K is (further) conditioned under some z, i.e. $K(...) \rightsquigarrow K(...|z)$ and $K(...|y) \rightsquigarrow K(...|y,z)$. Those stated are all valid within an additive constant of size $O(1)$, but there are others that are only

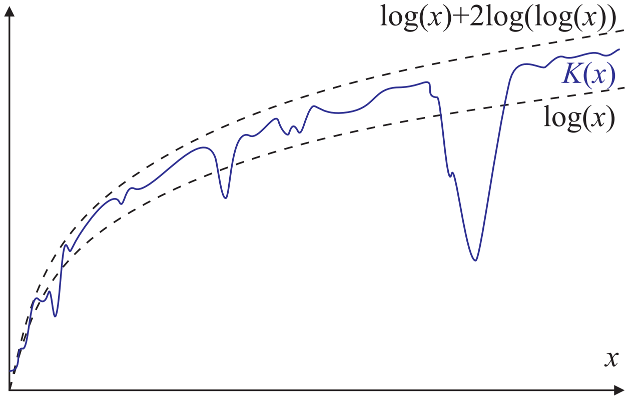

Figure 1: Schematic graph of prefix Kolmogorov complexity $K(x)$ with string x interpreted as integer. $K(x) \geq \log x$ for 'most' x and $K(x) \leq \log x + 2\log\log x + c$ for all x for suitable constant c.

valid to logarithmic accuracy. K has many properties in common with Shannon entropy as it should be, since both measure the information content of a string. Property (2) gives an upper bound on K, and property (3) is Kraft's inequality which implies a lower bound (4) on K valid for 'most' n, where 'most' means that there are only $o(N)$ exceptions for $n \in \{1,...,N\}$ These bounds allow us to draw a schematic graph of K as depicted in Figure 1. Providing side information y can never increase code length, requiring extra information y can never decrease code length (5). Coding x and y separately never helps (6), and transforming x does not increase its information content (8). Property (8) also shows that if x codes some object o, switching from one coding scheme to another by means of a recursive bijection f leaves K unchanged within additive $O(1)$ terms. The first nontrivial result is the symmetry of information (7), which is the analogue of the product rule for probabilities. Property (9) is at the heart of the MDL principle (Rissanen 1989), which approximates $K(x)$ by $-\log P(x) + K(P)$.

Proof ideas. All upper bounds on $K(z)$ are easily proven by devising *some* (effective) code for z of the length of the right-hand side of the inequality and by noting that $K(z)$ is the length of the shortest code among all possible effective codes. For instance, if T_j with $j = O(1)$ is a Turing machine with $T_j(\varepsilon' x') = x$, then $U(\varepsilon' j' x') = x$; hence $K(x) \leq \ell(\varepsilon' j' x') = \ell(x') + O(1) \leq \ell(x) + 2\log\ell(x) + O(1)$, which proves (2). In (9) one uses the Shannon-Fano code based on probability distribution P. Lower bounds are usually proven by counting arguments (easy for (4) and harder for (7)).

5 Other Complexities and Related Concepts

A notion closely related to Kolmogorov complexity is the probability that a universal computer outputs some string x when fed with a program chosen at random. This Algorithmic "Solomonoff" Probability (AP) is key in addressing the old philosophical problem of induction in a formal way (Solomonoff 1964, Hutter 2007).

The major drawback of AC and AP are their incomputability. Time-bounded "Levin" complexity penalizes a slow program by adding the logarithm of its running time to its length. This leads to computable variants of AC and AP, and Universal "Levin" Search (US) that solves all inversion problems in optimal time, apart from a huge multiplicative time constant (Levin 1973).

AC and AP also allow a formal and rigorous definition of randomness of individual strings that does not depend on physical or philosophical intuitions about nondeterminism or likelihood. Roughly, a string is Algorithmically "Martin-Loef" Random (AR) if it is incompressible in the sense that its algorithmic complexity is equal to its length (Martin-Loef 1966).

AC, AP, US, and AR are the core subdisciplines of Algorithmic Information Theory (AIT), but AIT spans into and has applications in many other areas. It serves as the foundation of the Minimum Description Length (MDL) principle, can simplify proofs in computational complexity theory, has been used to define a universal similarity metric between objects, solves the Maxwell demon problem, and many others.

6 History

The general theory of coding and prefix codes can be found in Gallager (1968), and the important Kraft inequality is due to Kraft (1949).

Kolmogorov complexity. A coarse picture of the early history of algorithmic information theory could be drawn as follows: Kolmogorov (1965) and Chaitin (1966) suggested defining the information content of an object as the length of the shortest program computing a representation of it. Solomonoff (1964) independently invented the closely related universal a priori probability distribution. Levin (1970) worked out most of the mathematical details. These papers may be regarded as the invention of (what is now called) Algorithmic Information Theory. The invariance in Section 2 is due to Solomonoff (1964), Kolmogorov (1965) and Chaitin (1969); properties (4) and (9) are due to Levin (1974); the symmetry of information (7) is due to Zvonkin & Levin (1970), Gacs (1974) and Kolmogorov (1983); the other parts are elementary.

Other complexities and related concepts. There are many variants of "Kolmogorov" complexity. The prefix Kolmogorov complexity K defined here (Levin 1974, Gacs 1974, Chaitin 1975), the earliest form, "plain" Kolmogorov complexity C (Kolmogorov 1965), process complexity (Schnorr 1973), monotone complexity

Km (Levin 1973), and uniform complexity (Loveland 1969), Chaitin's complexity *Kc* (Chaitin 1975), Solomonoff's universal prior $M = 2^{-KM}$ (Solomonoff 1964,1978), extension semimeasure *Mc* (Cover 1974), and some others. They often differ from K only by $O(\log K)$, but otherwise have similar properties. For an introduction to Shannon's (1948) information theory and its relation to Kolmogorov complexity, see Kolmogorov (1965,1983), Zvonkin & Levin (1970) and Cover (1991).

Resource-bounded complexity. The main drawback of all these variants of Kolmogorov complexity is that they are not finitely computable (Kolmogorov 1965, Solomonoff 1964). They may be approximated from above (Kolmogorov 1965, Solomonoff 1964), but no accuracy guarantee can be given, and what is worse, the best upper bound for the runtime until one has reasonable accuracy for $K(n)$ grows faster than any computable function in n. This led to the development of time-bounded complexity/probability that is finitely computable, or more general resource-bounded complexity/probability (e.g. Daley 1973,1977, Feder 1992, Ko 1986, Pintado 1997, Schmidhuber 2002).

7 References

[1] G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13(4):547–569, 1966.

[2] P. Gács. On the symmetry of algorithmic information. *Soviet Mathematics Doklady*, 15:1477–1480, 1974.

[3] R. G. Gallager. *Information Theory and Reliable Communication.* Wiley, New York, 1968.

[4] M. Hutter. On Universal Prediction and Bayesian Confirmation. *Theoretical Computer Science*, 384:1 (2007) 33-48

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

[6] A. N. Kolmogorov. Combinatorial foundations of information theory and the calculus of probabilities. *Russian Mathematical Surveys*, 38(4):27–36, 1983.

[7] L. A. Levin. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problems of Information Transmission*, 10(3):206–210, 1974.

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

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

[10] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.

8 Recommended Reading

For an excellent introduction to Kolmogorov complexity, and a more accurate treatment of its history and detailed references (more than 500), and many applications one should consult the authoritative book of Li and Vitányi (1997).

9 External Links

- Homepage of the author
  - Marcus Hutter: `http://www.hutter1.net/`
- Homepages of
  - Gregory J. Chaitin: `http://www.cs.auckland.ac.nz/CDMTCS/chaitin/`
  - Leonid A. Levin: `http://www.cs.bu.edu/ lnd/`
  - Ray Solomonoff: `http://world.std.com/ rjs/`
  - Cristian Calude: `http://www.cs.auckland.ac.nz/ cristian/`
  - Ming Li: `http://www.cs.uwaterloo.ca/ mli/`
  - Paul Vitanyi: `http://www.cwi.nl/ paulv/`
- Kolmogorov complexity resources: `http://www.hutter1.net/kolmo.htm` (introductions, bibliography, mailing list, researchers, events, ...)
- Andrei Nikolaevich Kolmogorov: `http://kolmogorov.com/` (biography, publications, pictures, interviews, ...)

10 See Also

- Algorithmic “Kolmogorov” Complexity `http://www.scholarpedia.org/article/Algorithmic_Complexity`
- Algorithmic “Solomonoff” Probability `http://www.scholarpedia.org/article/Algorithmic_Probability`
- Universal “Levin” Search `http://www.scholarpedia.org/article/Universal_Search`
- Algorithmic “Martin-Loef” Randomness `http://www.scholarpedia.org/article/Algorithmic_Randomness`

- Recursion Theory
  http://en.wikipedia/wiki/Recursion_Theory
- Applications of AIT
  http://www.scholarpedia.org/article/Applications_of_Algorithmic_Information_Theory