Kolmogorov Complexity

Carleton College, CS 254, Fall 2013, Prof. Joshua R. Davis

based on Sipser, *Introduction to the Theory of Computation*

1. Introduction

Kolmogorov complexity is a theory of lossless data compression. It ponders the existence of compression/decompression schemes, in which long strings of data are compressed into short strings, that can be decompressed back into their longer versions. Kolmogorov complexity is also a theory of information. Intuitively, the length of a compressed string is a measure of how much information the uncompressed string contains. To get an idea of what we mean, consider the uncompressed strings

$$0101010101010101010101010101010101010101010101010101010101010101010101,$$

$$0111101011000010100000101111111111101001101010100000000110111100000101.$$

Both strings have length 70. The first string can be described succinctly: "repeat '01' 35 times". In contrast, the second string does not have any apparent pattern. There is no obvious way to describe it, other than just laying out the whole string. It seems to contain more information than the first string does.

We begin with a minimal assumption: The process of decompressing a compressed string into its uncompressed version should be algorithmic. Therefore, we define a decompression *scheme* to be a deterministic Turing machine F that halts on all inputs. This F takes in an input string w, which we regard as the compressed string, and outputs the corresponding uncompressed string $F(w)$ (as the contents of its tape upon halting). For any x, any string w such that $F(w) = x$ is called a *description* of x. Let $d_F(x)$ be the minimal description of x, according to lexicographic order. The *Kolmogorov complexity* $K_F(x)$ is the length of the minimal description of x.

Throughout this tutorial, all strings are taken over the alphabet $\{0, 1\}$. Whenever we need to represent a non-negative integer m as a string $\langle m \rangle$, we do so in binary. In particular, $|\langle m \rangle| \leq \log_2 m + 1$ for all $m \geq 1$. These choices have no serious effect on the theory.

Theorem 1.1. *Under any scheme F, for any string length m, there are incompressible strings — meaning, strings x such that $K_F(x) \geq |x|$.*

Proof. Let $m \geq 1$ be any positive string length. There are $2^m - 1$ strings of length less than m, which, when fed to F, can produce at most $2^m - 1$ decompressed strings. But there are 2^m strings of length m. Therefore, at least one string of length m is incompressible. □

2. The default scheme

Let's define a particular decompression scheme F. This F takes as input $\langle M, w \rangle$, where M is a Turing machine and w is an input for M, all encoded into bits according to some pre-arranged system. This F runs M on w until it halts (if ever). Then F halts, with its tape containing the same contents as M's final tape.

It is conventional to assume that the encoding $\langle\rangle$ is such that w is given explicitly at the end of $\langle M, w\rangle$. That is, $\langle M, w\rangle$ consists of some encoding of M, followed by some separator mark, followed by w. In particular, $|\langle M, w\rangle| = |\langle M, \rangle| + |w|$.

This scheme will be our default. When we write d and K without any subscript, we mean this scheme.

Example 2.1. *Let M be the Turing machine that, on input w, produces 35 concatenated copies of w on its tape, and then halts. Then $\langle M, 01\rangle$ is a description of the first 70-bit string given above. The length of $\langle M, w\rangle$ is $|\langle M, \rangle| + 2$.*

Our next example expresses the idea that $d(x)$ should never be much longer than x itself, because "here is the string x" should always be a description of x.

Example 2.2. *There exists a constant c such that for all x, $K(x) \leq c + |x|$.*

Proof. Let M be a Turing machine that immediately halts. Let $c = |\langle M, \rangle|$. Then, for any string x, $\langle M, x\rangle$ is a description of x, of length $|\langle M, x\rangle| = |\langle M, \rangle| + |x| = c + |x|$. Thus the minimal description of x can be no longer than $c + |x|$, and $K(x) \leq c + |x|$. □

This next example says that xx should not require much more description than x.

Example 2.3. *There exists a constant c such that for all x, $K(xx) \leq c + |x|$.*

Proof. Let M be a Turing machine that repeats its input twice on its tape and then halts. Let $c = |\langle M, \rangle|$. Then, for any string x, $\langle M, x\rangle$ is a description of xx, of length $c + |x|$. □

Now that we have a taste for how compression and decompression work, let's prove a result that says that our default scheme is about as good as any other.

Theorem 2.4. *For any scheme F there exists a constant c such that $K(x) \leq c + K_F(x)$.*

Proof. Let $c = |\langle F, \rangle|$. Then $\langle F, d_F(x)\rangle$ is a description of x in the default scheme. Its length is $|\langle F, \rangle| + |d_F(x)| = c + K_F(x)$. □

3. Intermediate Results

Henceforth we shall work only with our default scheme. For any $c \geq 0$, we say that a string x is *incompressible* by c if $K(x) > |x| - c$. The notion of incompressibility introduced earlier is incompressibility by 1. This next theorem gets at the idea that $d(x)$, being the minimal description of x, should itself be incompressible.

Theorem 3.1. *There exists a constant c such that for all x, $d(x)$ is incompressible by c.*

Proof. Let N be a Turing machine that, on input $\langle M, w\rangle$, does the following steps.

(1) Run M on w.
(2) If the output of M is not of the form $\langle P, y\rangle$, then reject.
(3) If the output is of the form $\langle P, y\rangle$, then run P on y and halt with that output.

Let $c = |\langle N, \rangle| + 1$. Now suppose, for the sake of contradiction, that x is a string such that $d(x)$ is compressible by c. Then $|d(d(x))| \leq |d(x)| - c$. But $\langle N, d(d(x))\rangle$ is a description of x, and its length is

$$|\langle N, d(d(x))\rangle| = |\langle N, \rangle| + |d(d(x))| \leq (c - 1) + |d(x)| - c = |d(x)| - 1.$$

Therefore $K(x) \leq |d(x)| - 1$, which contradicts the definition of $K(x) = |d(x)|$. □

Recall that this whole theory is founded on a mild assumption: that decompression should be algorithmic. Under that assumption, this next theorem shows that optimal compression cannot be algorithmic. (Perhaps we should place more constraints on decompression, to arrive at a theory in which decompression and optimal compression are both algorithmic?)

Theorem 3.2. *The Kolmogorov complexity is not computable. In other words, there does not exist a Turing machine M that, given any input x, halts with $\langle K(x)\rangle$ on its tape.*

Proof. Suppose, for the sake of contradiction, that such an M exists. Build a decider N that, on input $\langle m\rangle$, outputs some string x satisfying $K(x) \geq m$. (N tries all strings x of length m, using M to compute $K(x)$, until it finds an x such that $K(x) \geq m$. Our first theorem guarantees that such an x will be found.) Now let m be a number large enough that

$$m - \log_2 m - 1 > |\langle N, \rangle|,$$

and let x be the output of N on input $\langle m\rangle$. Then $\langle N, m\rangle$ is a description of x, of length

$$|\langle N, m\rangle| = |\langle N, \rangle| + |\langle m\rangle| < (m - \log_2 m - 1) + (\log_2 m + 1) = m.$$

So $K(x) < m$. On the other hand, $K(x) \geq m$, by the definition of N. This contradiction implies that K is not computable. □

4. Random Strings

In this section, we explain the notion introduced earlier, that a "random" string has no pattern and hence should not be compressible. A *property* of strings over $\{0, 1\}$ is a function $f : \{0, 1\}^* \to \{\text{True}, \text{False}\}$. A property f is said to *hold for almost all strings* if

$$\lim_{n\to\infty} \frac{\#\{x : |x| = n, f(x) = \text{False}\}}{\#\{x : |x| = n\}} = 0.$$

Intuitively, f is True for "typical" strings x and False for "special cases" of x. If you select a string x randomly, then $f(x) = \text{True}$ with high probability. As $n \to \infty$, the probability that a randomly chosen string x of length n will have $f(x) = \text{True}$ goes to 1. Examples of such f include

- "x contains at least 40% 0s and at least 40% 1s."

- "the longest run of 0s in x has length between $0.5\log_2|x|$ and $1.5\log_2|x|$."

This notion allows us to investigate properties of random strings without really doing any probability theory.

The following purely mathematical lemma shows that we can replace "=" with "≤" in certain parts of the above definition. Sipser uses this fact without proof. You may want to skip the proof on a first reading.

Lemma 4.1. *Let f be a property that holds for almost all strings. Then*

$$\lim_{n\to\infty}\frac{\#\{x:|x|\le n, f(x)=\text{False}\}}{\#\{x:|x|\le n\}}=0.$$

Proof. Let $\epsilon>0$. We wish to show that there exists N such that for all $n\ge N$

$$\frac{\#\{x:|x|\le n, f(x)=\text{False}\}}{\#\{x:|x|\le n\}}<\epsilon.$$

For the sake of brevity, let $L_n=\#\{x:|x|=n, f(x)=\text{False}\}$. Because f holds for almost all strings, there exists an M such that for all $n>M$,

$$\frac{\#\{x:|x|=n, f(x)=\text{False}\}}{\#\{x:|x|=n\}}<\frac{\epsilon}{2}.$$

That is, $L_n<\frac{\epsilon}{2}2^n$ for all $n>M$. Pick N large enough so that

$$\sum_{i=0}^{M}L_i<\frac{\epsilon}{2}\left(2^{N+1}-1\right).$$

Then for all $n\ge N$

$$\begin{aligned}\#\{x:|x|\le n, f(x)=\text{False}\} &= \sum_{i=0}^{M}L_i+\sum_{i=M+1}^{n}L_i\\ &< \sum_{i=0}^{M}L_i+\sum_{i=M+1}^{n}\frac{\epsilon}{2}2^i\\ &< \frac{\epsilon}{2}\left(2^{N+1}-1\right)+\frac{\epsilon}{2}(2^{n+1}-1)\\ &\le \epsilon\left(2^{n+1}-1\right)\\ &= \epsilon\,\#\{|x|\le n\}.\end{aligned}$$

This proves the lemma. □

The following theorem says, roughly, that long incompressible strings have every property that holds for almost all strings. In this sense, they are "random".

Theorem 4.2. *Let f be a computable property that holds for almost all strings. Let $c\ge 1$. Then there exists an N such that $f(x)=$ True for all x such that $|x|\ge N$ and x is incompressible by c.*

Proof. If f is False on only finitely many strings, then f is true for all longer strings, and the theorem is obviously true. Henceforth assume that f is False on infinitely many strings. Denote these strings $s_0, s_1, s_2, \ldots$ in lexicographic order.

For any string x in the sequence $s_0, s_1, s_2, \ldots$, let i_x be its index in the list. That is, i_x is the unique number such that $s_{i_x} = x$. Let M be a Turing machine that on input $\langle i \rangle$ outputs s_i. (How would you design M, using the fact that f is computable?) Then $\langle M, i_x \rangle$ is a description of x.

Fix $c \geq 1$. By the lemma, there exists a large N so that for all $n \geq N$

$$\frac{\#\{x : |x| \leq n, f(x) = \text{False}\}}{\#\{x : |x| \leq n\}} < \frac{1}{2^{c+|\langle M, \rangle|+2}}.$$

Using the fact that $\#\{x : |x| \leq n\} = 2^{n+1} - 1$, we have

$$\#\{x : |x| \leq n, f(x) = \text{False}\} < \frac{2^{n+1}}{2^{c+|\langle M, \rangle|+2}} = 2^{n-c-|\langle M, \rangle|-1}.$$

If x is any string of length $n \geq N$ such that $f(x) = \text{False}$, then

$$i_x < 2^{n-c-|\langle M, \rangle|-1}$$

and

$$|\langle i_x \rangle| \leq n - c - |\langle M, \rangle|.$$

Therefore

$$K(x) \leq |\langle M, i_x \rangle| \leq |\langle M, \rangle| + n - c - |\langle M, \rangle| = n - c.$$

So x is compressible by c. In other words, any x of length at least N that is incompressible by c satisfies $f(x) = \text{True}$. $\square$