Kolmogorov Complexity and NCD

David Clark
University College London

Source material

- Cover and Thomas: Elements of Information Theory, Chapter 7 (1991)
- Li and Vitanyi: An Introduction to Kolmogorov Complexity and its Applications (3rd edition)

Intrinsic Complexity of an Object

algorithmic (descriptive) complexity of an object

The algorithmic complexity of an object is the length of the shortest binary computer program that describes the object

- Defined by Kolmogorov in 1965
- Dispenses with the probability distribution essential to Shannon information
- Essentially a computer independent definition
- Expected length of of the shortest binary computer description of a random variable is approximately equal to its entropy
- Algorithmic complexity is a conceptual precursor to entropy

Motivating Examples

1. 0101010101010101010101010101010101010101010101
2. 0110101000001001111001100110011111111001110111100 1
3. 1101111001110101111101101111101110101101111000101

1. simple. repetitions of 01
2. passes most tests for randomness. $\sqrt{2} - 1$.
3. looks random. Proportion of 1's not near $1/2$. Describe number, k, of 1's in the sequence, order all sequences possessing this number of 1's lexicographically, then give the index of the sequence. Describing the sequence in this way gives a description in roughly $log\, n + n\,\mathcal{H}(\frac{k}{n})$ bits. So less simple, but still simple.

Truly Random Sequences

- Imagine a truly random sequence generated by a random process such as flipping a coin
- 2^n such sequences of length n, all equiprobable
- highly likely such a (truly random) sequence cannot be compressed
- shortest program to produce the (truly random) string is an instruction to print the string

Computer Independence

- can show that this notion of intrinsic complexity is essentially computer independent
- intuition: programs on different machines vary only by a "mimic" constant – for long strings this constant becomes relatively insignificant

Models of Computation

- universal Turing machine – conceptually simplest computer
- Church's thesis - all (sufficiently complex) computational models are equivalent in the sense that they compute the same set of functions
- effectively computable functions

Universal Turing Machine

- machine reads code from right to left only $\Rightarrow$ programs are a prefix-free set of binary strings
- UTM maps a set of finite length binary strings to the set of finite or infinite binary strings
- $f : \{0,1\}^* \rightarrow \{0,1\}^* \cup \{0,1\}^\infty$
- *partial recursive functions*

Definitions and Examples

- Let x be a finite binary string
- Let $\mathcal{U}$ be a universal computer
- let $\ell(x)$ be denote the length of string x

Kolmogorov Complexity

The *Kolmogorov Complexity* $K_{\mathcal{U}}(x)$ of a string x with respect to a universal computer $\mathcal{U}$ is defined as

$$K_{\mathcal{U}}(x) = min_{p\,:\,\mathcal{U}(p)=x}\ \ell(p)$$

Rule of Thumb Upper Bound

- describe a sequence to another person in a manner that leads unambiguously to a computation of that sequence in a finite amount of time
- "Print out the first 1,239,875,981,825,931 bits of the square root of e"
- 73 characters $\times$ 8 bits per character $=$ upper bound of 584 bits on the Kolmogorov complexity of the number
- most numbers of this length have a Kolmogorov Complexity of $1,239,875,981,825,931$ bits

Conditional Complexity

Conditional Kolmogorov Complexity

$$K_{\mathcal{U}}(x|\ell(x)) min_{p\,:\,\mathcal{U}(p,\ell(x))=x}\ \ell(p)$$

- shortest description length if $\mathcal{U}$ has the length of x available to it

Universality

Theorem: Universality of Kolmogorov complexity

If $\mathcal{U}$ is a universal computer, then for any other computer $\mathcal{A}$,

$$K_{\mathcal{U}}(x) \leq K_{\mathcal{A}}(x) + c_{\mathcal{A}}$$

for all strings $x \in \{0,1\}^*$, where $c_{\mathcal{A}}$ does not depend on x

Lower Bound

Lower Bound on Kolmogorov Complexity

The number of strings x with complexity $K(x) < k$ satisfies

$$|\{x \in \{0,1\}^* : K(x) < k\}| < 2^k$$

Proof

- there are not many short programs
- number of programs of length $< k$ is less than $2^k - 1$
  (number of strings of length $< k$) $< 2^k$
- since each program can produce only one output sequence the number of sequences with complexity $< k$ is $< 2^k$

Some Examples

a sequence of n zeros

- assume that the computer knows n
- program: `Print the specified number of zeros`
- complexity: $K(0000\ldots0|n) = c$ for all n

complexity of π

- first n bits of π can be calculated using a simple series expression
- program for this has a small constant length
- complexity: $K(\pi_1\pi_2\pi_3\ldots\pi_n|n) = c$

a fractal drawing

- for different points c on the complex plane calculate the number of iterations of the map $z_{n+1} = z_n + c$ needed for $|z|$ to cross a given threshold
- c is then coloured according to the number of iterations needed
- Kolmogorov complexity is nearly 0

the Mona Lisa

- compress the image
- expect that the compression reduces the size of the representation by a factor of two thirds
- $K(Mona\ Lisa|n) = \frac{n}{3} + c$

Normalised Compression Distance

- Normalised Information Distance (NID)

$$e(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}$$

- $e(x,y) = \max\{K(x|y), K(y|x)\} = K(xy) - min\{K(x), K(y)\}$
- Normalised Compression Distance

$$e_Z(x,y) = \frac{Z(xy) - \min\{Z(x), Z(y)\}}{\max\{Z(x), Z(y)\}}$$

Choosing a Compressor

- factors that influence the NCD value
  - type of string
  - size of strings to compare
  - size of compressor window
- 7zip better on average than other compressors
  - more accurate behaviour for self comparisons
  - lower compressed length on average
- set 7zip to 4GB window – allow comparison of files with 4GB concatenation

NCD(x,x) for 7zip, gzip, winzip, bzip2