

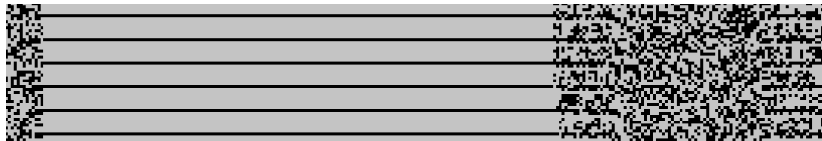
Kolmogorov complexity

Two introductory lectures

George Barmpalias

Chinese Academy of Sciences

UESTC - Chengdu, December 3, 2023



Program-size complexity

Aim: Formalize and quantify the complexity of (finite) data.

How many bits of information do these strings have?

abababababababababababababababababab

4c1j5b2p0cv4w1x8rx2y39umgw5q85s7

The choice of alphabet is not important. Binary is standard.

In unix-based systems (including mac): `xxd -b FILE`



```
00002f9a: 00110011 00101110 00110100 00101101 00110001 00101101 3.4-1-
00002fa0: 00110101 00110000 00101110 00110010 00101101 00110100 50.2-4
00002fa6: 00101110 00110100 00100000 00110001 00110110 00101110 .4 16.
00002fac: 00111000 00101101 00110101 00100000 00110011 00110011 8-5 33
00002fb2: 00101110 00110110 00100000 00110001 00100000 00110101 .6 1 5
00002fb8: 00110000 00101110 00110010 01111010 00100010 00101111 0.2z"/
00002fbe: 00111110 00111100 00101111 01100111 00111110 00001010 ></g>.
00002fc4: 00111100 00101111 01100111 00111110 00001010 00111100 </g>.<
00002fca: 00101111 01110011 01110110 01100111 00111110 00001010 /svg>.
```

Plan for today

- ▶ What is program size complexity
- ▶ What is Algorithmic Information theory
- ▶ Machines, Programs and Numberings
- ▶ Universal machines and Invariance
- ▶ Kolmogorov complexity
- ▶ Quantitative estimates and properties
- ▶ Berry's paradox and undecidability
- ▶ Conditional complexity
- ▶ Conservation of information
- ▶ Self-delimiting programs and complexity

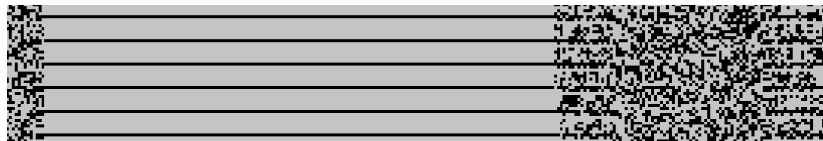
Information theory (Shannon)

- ▶ strings as outcomes of a random process
- ▶ *entropy* with respect to a probability distribution
- ▶ Information content of outcome i with probability p_i is $\log p_i$

Shannon entropy is the expected (average) information per symbol:

$$H := \sum_i p_i \log p_i$$

Entropy measures the amount of surprise for the next symbol.



Pros and Cons

- ▶ Backbone of modern communications and digital encoding
- ▶ information-compression (source-coding theorem)
- ▶ Entropy is the average length of code per word in the signal
- ▶ Error-correcting, communication over noisy channels etc.

Drawbacks - Criticism (Kolmogorov, Solomonoff)

- ▶ depends on the underlying assumed distribution of the data
- ▶ this may be unknown, or even non-existent
- ▶ measures **average** amount of information per letter

Kolmogorov complexity

How many bits of information do these strings have?

ababababababababababababababababab

4c1j5b2p0cv4w1x8rx2y39umgw5q85s7

Definition. The complexity $C(\sigma)$ of string σ is the length of the smallest program that prints σ .

How many bits of information do these strings have?

59265358979323846264338327950288

21001798141066463695774083871573

Which programming language?

Kolmogorov complexity is based on the theory of computation.

The standard abstraction of a computer is the **Turing machine**.

Alternatively, use any *Turing-complete* programming language.

Quiz. Which of the following languages are Turing-complete?

- ▶ python
- ▶ HTML
- ▶ Solidity (Ethereum)
- ▶ Bitcoin scripting language
- ▶ Javascript
- ▶ HTML5

Program-size depends on the language at hand.

Extreme examples

Code golf is a recreational programming competition.

Goal is to achieve the shortest source-code that solves a given problem.

Golfscript for the first 1000 digits of π

```
;''  
6666,-2%{2+.2/@*\ /10.3??2*+}*  
`1000<~\;
```

Whitespace code for the first 1000 digits of π^*

Theory of computation

Program-size depends on the language at hand.

The length of the program depends on the language M .

Facts from computability:

- ▶ there is a program that enumerates all Turing machines
- ▶ there exists a universal Turing machine U
- ▶ U can simulate any Turing machine, with a constant overhead

Invariance

Let $C_M(\sigma)$ be the Kolmogorov complexity with respect to M :

$$C_M(\sigma) := \min\{|\tau| : M(\tau) = \sigma\}$$

Theorem. For every machine M there exists a constant c such that

$$C_U(\sigma) \leq C_M(\sigma) + c$$

Fix a universal machine U and let $C(\sigma) := C_U(\sigma)$.

Kolmogorov complexity is:

invariant of the choice of the universal machine, up to a constant.

Features of Kolmogorov complexity

- ▶ does not require a distribution (model, hypothesis)
- ▶ gives a universal apriori distribution (Bayesian inference)
- ▶ formal framework for artificial general intelligence (Solomonoff)
- ▶ it is incomputable but can only be approximated
- ▶ machine-invariant but huge constants matter in practice

How many bits of information do these strings have?

1637374017199159701353736070558761985836294941139158650322685731114837878

1415926535897932384626433832795028841971693993751058209749445923078164062

Quantitative estimates

- ▶ $C(\sigma) \leq |\sigma| + O(1)$
- ▶ $C(0^n) \leq \log n + O(1)$
- ▶ $C(\sigma\sigma) \leq C(\sigma) + O(1)$

Theorem. If f is computable, $C(f(\sigma)) \leq^+ C(\sigma)$.

Does the converse hold?

Theorem. $|\{\sigma \mid |\sigma| = n \wedge C(\sigma) \leq C(0^n) + d\}| = O(2^d)$.

Incompressibility

Definition. A string σ is *incompressible* if $C(\sigma) \geq |\sigma|$.

Theorem. The set of compressible strings is computably enumerable.

Theorem. For each n there exists an n -bit incompressible string.

Proof. There exist 2^n many strings of length n and

$$1 + \dots + 2^{n-1} = 2^n - 1$$

many strings of length $< n$.

Some n -bit string must fail to have a description of length $< n$.

Incompressible strings are often called *random*.

Can you compute Kolmogorov complexity?

Naive attempt to compute $C(s)$

- ▶ Order the programs in increasing length
- ▶ find the first with output s .

Code:

```
function KolmogorovComplexity(string s)
  for i = 1 to infinity:
    for each string p of length exactly i
      if isValidProgram(p) and evaluate(p) == s
        return i
```

Berry paradox

The smallest positive integer not definable in under sixty letters.

- ▶ There are finitely many sentences of less than sixty letters
- ▶ finitely many numbers can be defined in this way
- ▶ there exists a least m that cannot be define in this way

But the above sentence appears to define m .

Theorem. It is undecidable whether a string is incompressible.

Proof. For a contradiction, assume otherwise.

Program M on input n outputs the lexicographically least incompressible n -bit string σ_n .

Then $C_M(\sigma_n) = \log n + O(1)$. So $C(\sigma_n) \leq \log n + O(1)$.

This contradicts $n \leq C(\sigma_n) + O(1)$. ◀

Conditional complexity

$C(\sigma \mid \tau)$ is the information in σ given τ :

$$C(\sigma \mid \tau) = \min\{|p| \mid U(\tau, p) = \sigma\}$$

The shortest program which, given τ can print σ .

Theorem. If A is finite $\forall \tau \exists \sigma \in A : C(\sigma \mid \tau) \geq \log |A|$.

Theorem. Suppose B is an infinite computably enumerable set of pairs of strings with all projections

$$B_\tau := \{\sigma : (\sigma, \tau) \in B\}$$

finite. Then $\forall \tau \forall \sigma \in B_\tau C(\sigma \mid \tau) \leq^+ \log |B_\tau|$.

Shortest program

Let σ^* denote the shortest program for σ . Then $C(\sigma) = |\sigma^*|$.

Theorem. The following hold:

- ▶ $C(\sigma, C(\sigma)) =^+ C(\sigma^*)$.
- ▶ $C(\sigma^* | \sigma) =^+ C(C(\sigma) | \sigma)$.

Proof. From $\sigma, C(\sigma)$ we can compute σ^*

From σ^* we know $C(\sigma) = |\sigma^*|$ and $\sigma = U(\sigma^*)$. ◀

Theorem. The following hold:

- ▶ $C(\sigma\tau) \leq^+ C(\sigma, \tau) \leq^+ C(\sigma) + C(\tau) + 2 \log C(\sigma)$.
- ▶ $\forall d \exists \sigma, \tau : C(\sigma\tau) > C(\sigma) + C(\tau) + d$.

Conservation of information ?

This is weird... it should not happen.

How can $\sigma\tau$ have more information than $C(\sigma) + C(\tau)$?

Explanation: A program τ for σ

- ▶ carries information in its digits, but also in its length $|\tau|$
- ▶ this can make $C(\sigma)$ smaller than it should be.

By restricting the underlying machines to:

- ▶ Self-delimiting (one-way reading of the the input-tape)
- ▶ or exuivalently, prefix-free

we obtain a refined complexity $K(\sigma)$.

Self-delimiting programs

- ▶ $K(\sigma\tau) \leq^+ K(\sigma) + K(\tau)$
- ▶ $C(\sigma) \leq^+ K(\sigma)$

Halting probability:

$$\Omega := \sum_{U(\tau)\downarrow} 2^{-|\tau|}$$

gives a **probability distribution** on the strings:

$$P(\sigma) = \sum_{U(\tau)\downarrow=\sigma} 2^{-|\tau|}$$

where $\sum_{\sigma} P(\sigma) = \Omega$.

Coding Theorem

$$P(\sigma) = 2^{-K(\sigma)}$$

$$K(\sigma) = -\log P(\sigma)$$