Kolmogorov Complexity: An Algorithmic Theory of Information

Merkouris Papamichail[†], Giorgos Roussakis[†]

[†]Computer Science Department
University of Crete

Information Theory,
Fall 2023

Presentation's Outline

Presentation's Outline

An Intuitive Example (1/2)

Consider the sequences:

- $s_1 : 2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots$
- $s_2 : 3, 5, 17, 257, 65537, 4294967297, 18446744073709551617, \ldots$

- What sequence carries the most information?
  - Shannon's Information theory does not regard the *semantics* of a message.
  - A message carries information proportional to the uncertainty of the receiver that observes the message.
  - If a source chooses uniformly between s_1 and s_2, then each sequence is treated the same by information theory.

An Intuitive Example (2/2)

Consider the sequences:

- $s_1 : 2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots$
- $s_2 : 3, 5, 17, 257, 65537, 4294967297, 18446744073709551617, \ldots$

- What sequence carries the most information?
  - s_1 is the sequence of prime numbers, which cannot be represented by a *closed formula*.
  - s_2 is the sequence of Fermat number, which can be given by the formula,

$$F_n = 2^{2^n} + 1. \tag{1}$$

  - Therefore, in a sense s_1 carries more information from s_2.
  - A sender may encode (1) as a message. Then a receiver may reconstruct the whole initial sequence.

Another example

Assume we want to *compress* the image of Figure 1, bellow.

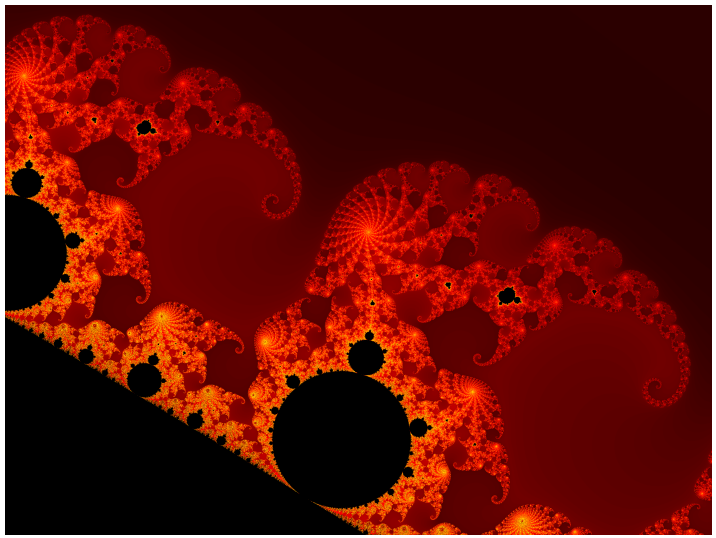

Figure 1: The Mandelbrot set fractal.

- Storing the 24-bit colour of each pixel would require 23MB.
- Storing the definition of Mandelbrot fractal as a computer programme, requires only few kilo-bytes.
- Any computer can reconstruct the image from this programme.

Kolmogorov Complexity

Kolmogorov Complexity:

- Captures the relation between a finite object and the minimum programme that generates it.
- An *algorithmic* approach to information theory.
- Developed Andrey Kolmogorov and (independently) by Ray Solomonoff at 1960s.

Applications:

- Data compression, e.g. the MIDI protocol for music synthesising.
- An algorithmic interpretation of randomness.
- Many theoretical results in combutability theory, theoretical computational reasoning, etc.

Presentation's Outline

Preliminary Notions: Computability Theory

Deciding a description language:

- The language we use to describe an object.
- We could use any programming language, e.g. C, Prolog, Java, etc.
- We will use the abstract formulation of a *Turing machine* (TM).
- Any algorithm can be represented by a TM.
- Any Turing machine M can be *simulated* by the *universal Turing machine* $\mathcal{U}$.

The Universal Turing Machine

The universal machine $\mathcal{U}$ can be given a *binary representation* $\langle M \rangle$ of M simulates M and the halts. When $\mathcal{U}$ simulates M, we will write $\mathcal{U}(\langle M \rangle)$.

Kolmogorov Complexity (1/2)

Definition

Let $s \in \{0,1\}^\star$ be a finite sequence. The Kolmogorov complexity of s is the minimum, in length, TM M, such that M prints s and halts. Namely,

$$K(s) = \min\{|\langle M \rangle| \mid \mathcal{U}(\langle M \rangle) = s\}$$

Kolmogorov Complexity (2/2)

Definition

Let $s \in \{0,1\}^\star$ be a finite sequence. The Kolmogorov complexity of s is the minimum, in length, TM M, such that M prints s and halts. Namely,

$$K(s) = \min\{|\langle M \rangle| \,|\, \mathcal{U}(\langle M \rangle) = s\}$$

- A trivial TM P_s which prints s and then stops.
- P_s will have internally s "hard-coded".
- Thus,

$$K(s) \leq |\langle P_s \rangle| = |s| + c.$$

- The constant c depends only from the computational model.

Universality: Decription

- Does $K(s)$ depend on the representational language?

  - We will argue that Kolmogorov complexity is an innate property of an object.
  - Thus $K(s)$ should be *independent* of the computational system.
  - Writing a programme that encodes s in, e.g. C should not make much difference from writing the programme in another language, e.g. Python.
  - Kolmogorov complexity's universality derives from the *Church-Turing thesis*.

Universality: Church-Turing Thesis

Church-Turing Thesis

Any *reasonable* computational model is equivalent to the universal Turing machine $\mathcal{U}$.

- Church-Turing thesis is *not* a theorem.
- Is a well accepted doctrine of computer science.
- It stipulates that there is always a compiler from a programming language $\mathcal{A}$ to the universal Turing machine $\mathcal{U}$ and *vice versa*.
- Any reasonable computational system can be *simulated* by the universal Turing machine $\mathcal{U}$, and vice versa.

Universality: Formal Treatment

Universality Theorem

Let $s \in \{0,1\}^\star$ be a finite binary string. If $\mathcal{U}$ is a universal Turing machine, for any other, deterministic computational model $\mathcal{A}$ there is a constant $c_{\mathcal{A}\to\mathcal{U}}$ such that,

$$K_{\mathcal{U}}(s) \leq K_{\mathcal{A}}(s) + c_{\mathcal{A}\to\mathcal{U}}. \tag{2}$$

- The constant $c_{\mathcal{A}\to\mathcal{U}}$ is the size of a $\mathcal{A}$ to $\mathcal{U}$ compiler.
- Equation (2) corresponds in a process of encoding s in $\mathcal{U}$ given a description of s in $\mathcal{A}$.
- Intuitively we simulate the minimum programme written in $\mathcal{A}$ in $\mathcal{U}$.

Presentation's Outline

Incompressibility (1/2)

- Let $S^n = \{0,1\}^n$ the set of binary string of length at most n. $|S^n| = 2^n$
- Let M^k be the set of TMs with description of length at most k. $|M^k| \leq 2^k$
- Any string $s \in S^n$ compressed by a TM with length shorter than k, will be described by some TM $M \in M^k$.
- Since TMs are *deterministic*, then there is an *injection* from the elements of M^k to a subset of S^n which they compress.
- Since there are at most 2^k TMs of length at most k, there are at most 2^k strings compressible to k bits.

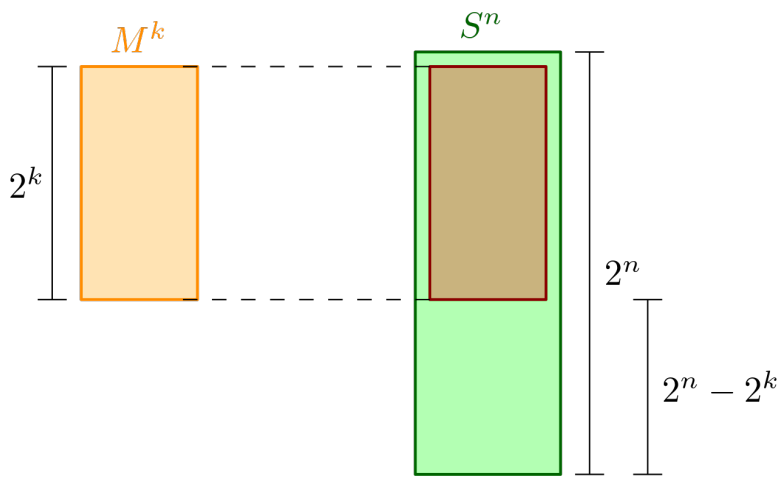

Figure 2: The size of the set of incompressible strings, relative to the size of the set $\{0,1\}^\star$

Incompressibility (2/2)

Incompressible Strings

Let $S^n = \{0,1\}^n$ be the set of strings with length at most n. For every $k \in \mathbb{N}$, there are at least $2^n - 2^k$ strings with $K(s) > k$.

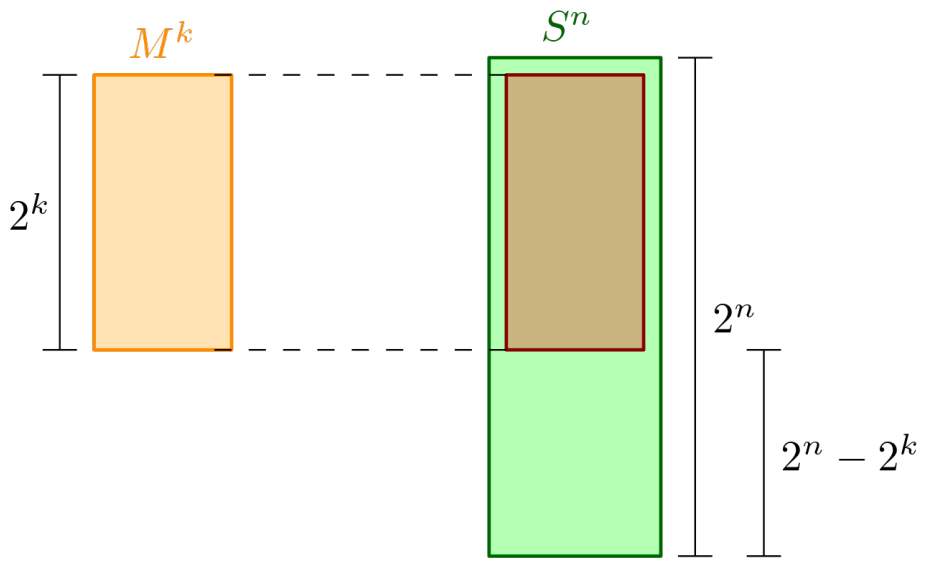

Figure 3: The size of the set of incompressible strings, relative to the size of the set $\{0,1\}^\star$

Incompressibility Results

- Lower Bound $\Omega(n^2)$ for simulating 2 tapes by 1 (open 20 years)
- k heads $>$ k-1 heads for PDAs (open 15 years)
- k one-ways heads can't do string matching (open 13 years)
- 2 heads are better than 2 tapes (open 10 years)
- k tapes are better than k-1 tapes. (open 20 years)

Incompressibility Results

- Lower Bound $\Omega(n^2)$ for simulating 2 tapes by 1 (open 20 years)
- k heads $>$ k-1 heads for PDAs (open 15 years)
- k one-ways heads can't do string matching (open 13 years)
- 2 heads are better than 2 tapes (open 10 years)
- k tapes are better than k-1 tapes. (open 20 years)
- Average case analysis for heapsort (open 30 years)
- Shellsort average case lower bound (open 40 years)

Incompressibility Results

- Lower Bound $\Omega(n^2)$ for simulating 2 tapes by 1 (open 20 years)
- k heads $>$ k-1 heads for PDAs (open 15 years)
- k one-ways heads can't do string matching (open 13 years)
- 2 heads are better than 2 tapes (open 10 years)
- k tapes are better than k-1 tapes. (open 20 years)
- Average case analysis for heapsort (open 30 years)
- Shellsort average case lower bound (open 40 years)
- Simplify old proofs (Hastad Lemma)

Incompressibility Results

- Lower Bound $\Omega(n^2)$ for simulating 2 tapes by 1 (open 20 years)
- k heads $>$ k-1 heads for PDAs (open 15 years)
- k one-ways heads can't do string matching (open 13 years)
- 2 heads are better than 2 tapes (open 10 years)
- k tapes are better than k-1 tapes. (open 20 years)
- Average case analysis for heapsort (open 30 years)
- Shellsort average case lower bound (open 40 years)
- Simplify old proofs (Hastad Lemma)
- Many theorems in combinatorics, formal language/automata, parallel computing, VLSI

Non-computability: the Halting Problem

- A fundamental result of the Theory of Computability is that there are *not computable functions*.
- There are well-defined mathematical functions for which there is *not* an algorithm that computes them.
- One such non-computable function is given by the *Halting Problem*.

Halting Problem

There is no fixed Turning machine H that takes as input the description of a Turing machine $\langle M \rangle$ *decides* whether M stops (*halts*) after a finite number of steps.

Non-computability of Kolmogorov Complexity

- In order to compute Kolmogorov complexity $K(s)$ for a particular string s, we need to calculate the minimal TM M that produces s and then stops.
- We need to check all TMs of length less than $|s|$ if it halts.
- Due to the non-computability of the Halting Problem, the above test is *infeasible*.

Non-computability of Kolmogorov Complexity

- In order to compute Kolmogorov complexity $K(s)$ for a particular string s, we need to calculate the minimal TM M that produces s and then stops.
- We need to check all TMs of length less than $|s|$ if it halts.
- Due to the non-computability of the Halting Problem, the above test is *infeasible*.

Theorem

The function $K : \{0,1\}^\star \to \mathbb{N}$ that gives the Kolmogorov complexity of any finite binary string $s \in \{0,1\}^\star$ is *not* computable.

Universal Probability

Universal probability is a concept derived from Kolmogorov complexity and is associated with a universal Turing machine.

Theorem

$$P_{\mathcal{U}}(x) = \sum_{p:\mathcal{U}(p)=x} 2^{-l(p)} = Pr(\mathcal{U}(p) = x)$$

In other words this is the probability that a program randomly drawn as a sequence of fair coin flips $p_1, p_2, \ldots$ will print out the string x.

Significance

Universal probability plays a crucial role in Solomonoff induction, guiding inductive inference by prioritizing simpler explanations.

Connection to Kolmogorov

What does this have to do with Kolmogorov?

Connection to Kolmogorov

What does this have to do with Kolmogorov?

Theorem

There exists a constant c, independent of x, such that:

$$2^{-K(x)} \leq P_{\mathcal{U}}(x) \leq c2^{-K(x)}$$

for all strings x. Thus, the universal probability of a string x is determined essentially by its Kolmogorov complexity.

Connection to Kolmogorov

What does this have to do with Kolmogorov?

Theorem

There exists a constant c, independent of x, such that:

$$2^{-K(x)} \leq P_{\mathcal{U}}(x) \leq c2^{-K(x)}$$

for all strings x. Thus, the universal probability of a string x is determined essentially by its Kolmogorov complexity.

Importance

This implies that K(x) and $\log \frac{1}{P_{\mathcal{U}(x)}}$ have equal status as universal complexity measures.

Minimum Description Length(1/2)

Suppose we have $X_1, X_2, \ldots, X_n$ drawn i.i.d. from p(x). We don't know p but we know $p \in \mathcal{P}$, where $\mathcal{P}$ is a class of probability functions.
What's the PMF that describes the data?

Theorem

Find the $p \in \mathcal{P}$ that minimizes:

$$L_P(X_1, X_2, \ldots, X_n) = K(p) + log \frac{1}{p(X_1, X_2, \ldots, X_n)}$$

Problems?

Incomputability of Kolmogorov as always and difficulty of computing p

Minimum Description Length(2/2)

Practical MDL

Suppose we are given data D and a model/theory, a hypothesis H. Another definition which is more widely applicable is the following. Select a hypothesis H that minimizes:

$$K(H) + K(D|H)$$

Examples

Imagine probability distribution of heads or tails.
Example of H: encoding "if previous_toss == Heads, predict Tails; if previous_toss == Tails, predict Heads."

Presentation's Outline

Kolmogorov Complexity & Shannon Information Theory (1/2)

- Consider a finite set of strings $\mathcal{X}$.
- Let X be a random variable that takes values from $\mathcal{X}$, according to a distribution $P(x)$.
- The expected optimal length of the random variable X is given by,

$$\sum_{x \in \mathcal{X}} P(x)K(x). \tag{3}$$

- In (3) optimality is achieved by optimally encode each individual $x \in \mathcal{X}$, with respect to Kolmogorov complexity.
- Contrarily using Shannon's information theory, the optimal average length is given by the entropy,

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x). \tag{4}$$

- Optimality in (4) is achieved by optimal source coding, with respect to the distribution $P(x)$.
- Can we relate (3) with (4)?

Kolmogorov Complexity & Shannon Information Theory (2/2)

Theorem

Let $P(\cdot)$ be a commutable probability distribution. Then,

$$0 \leq \sum_{x \in \mathcal{X}} P(x)K(x) - H(P) \leq c_P. \tag{5}$$

The constant c_P depends only form the probability distribution P.

Kolmogorov Complexity & Shannon Information Theory (2/2)

Theorem

Let $P(\cdot)$ be a commutable probability distribution. Then,

$$0 \leq \sum_{x \in \mathcal{X}} P(x)K(x) - H(P) \leq c_P. \tag{5}$$

The constant c_P depends only form the probability distribution P.

- Therefore, for every computable distribution $P(\cdot)$, the universal code $D^\star$ whose leangth function is the Kolmogorov compresses on average at least as much as the optimal Shannon-Fano code for $P(\cdot)$.

Presentation's Outline

Conclusions

- In this presentation we briefly discussed Kolmogorov complexity.

  - We defined the notion of Kolmogorov complexity, for measuring the information of an object by the length of its minimum description.
  - We saw that Kolmogorov complexity is *universal*, since depends only of the object.
  - We presented some elementary results, such as:
    - The existence of incompressible strings.
    - The non-computability of Kolmogorov complexity.
    - The notion of universal probability.
    - The fundamental result of Minimum Description Length.
  - We also described the relationship between Kolmogorov complexity and Shnnon entropy.

Uncovered Issues

Kolmogorov complexity is a vast field with many implication to theoretical computer science and artificial intelligence.

- Kolmogorov complexity lays at the heart of *Solomonoff's* theory of inductive inference.
- Alternative proofs of fundamental theorems of computability theory can be derived by Kolmogorov complexity, see Halting problem.
- Provides a way to formalise foundational doctrines of epistemology, such as Occam's razor.

"It seems to me that the most important discovery since Gödel was the discovery by Chaitin, Solomonoff and Kolmogorov of the concept called Algorithmic Probability which is a fundamental new theory of how to make predictions given a collection of experiences and this is a beautiful theory, everybody should learn it, but it's got one problem, that is, that you cannot actually calculate what this theory predicts because it is too hard, it requires an infinite amount of work. However, it should be possible to make practical approximations to the theory that would make better predictions than anything we have today. Everybody should learn all about that and spend the rest of their lives working on it."

– Minsky (2014)

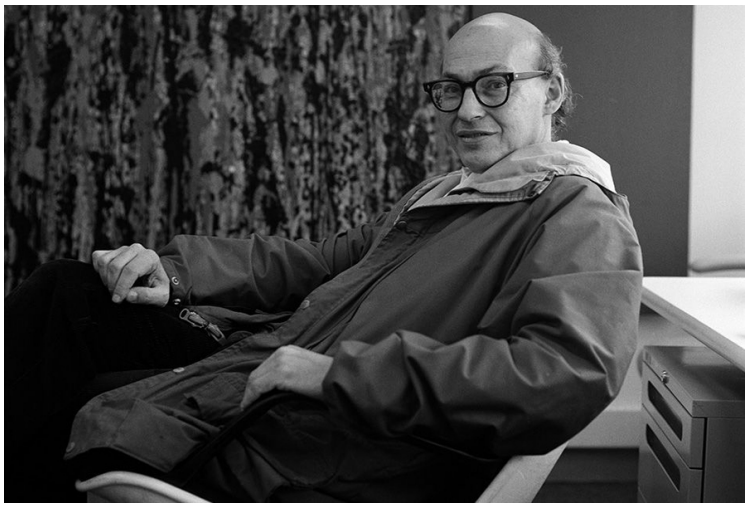

Figure 4: Marvin Minksy (1927 - 2016)