Dynamical systems: unpredictability vs uncomputability

Mathieu Hoyrup

INRIA Nancy (FRANCE)

How to understand unpredictability, randomness?

Several possible answers:

The newtonian physicist

The evolution is deterministic, but:

- sensitivity to initial conditions,
- approximative knowledge of the state of the system.

Mathematical model: deterministic dynamical systems.

The computer scientist

Difficulty, impossibility to compute the evolution of the system.

How to understand unpredictability, randomness?

Theorem

A dynamical system is strongly unpredictable if and only if it has strongly uncomputable trajectories.

Along the talk, we will see:

- What strong chaos is.
- What strongly uncomputable trajectories are.
- Some relations between unpredictability and uncomputability.

Topological entropy

- X is a compact metric space, $T : X \to X$ is a continuous map.
- We fix some small $\epsilon > 0$: observations of the systems will be carried out with precision ϵ.
- Unpredictability: given a set Y of possible initial states, unpredictability arises when several significantly different (i.e. not ϵ-close) trajectories start from Y

Topological entropy

Quantifies the speed of separation of trajectories.
[Adler, Konheim, McAndrew (1965)], [Bowen (1971)]

Topological entropy

Let $\epsilon > 0$ and $n \in \mathbb{N}$.

- Two finite sequences $x_0, x_1, \ldots, x_{n-1}$ and $y_0, y_1, \ldots, y_{n-1}$ are **ϵ-close** if $d(x_i, y_i) < \epsilon$ for every $i \leq n-1$.
- Let $N_Y(n, \epsilon)$ be the minimal number of trajectories of length n such that every trajectory starting from Y is ϵ-close to one of them.

Usually, $N_Y(n, \epsilon)$ grows exponentially fast as $n \to \infty$. The topological entropy is the (maximal) exponential rate:

$$h(Y, T, \epsilon) := \limsup \frac{\log N_Y(n, \epsilon)}{n}$$

and

$$h(Y, T) := \lim_{\epsilon \to 0} h(Y, T, \epsilon) = \sup_{\epsilon > 0} h(Y, T, \epsilon)$$

and

$$h(T) = h(X, T).$$

Topological entropy

A few examples

- The entropy of the shift $\sigma : \Sigma^{\mathbb{N}} \to \Sigma^{\mathbb{N}}$ is $h(\sigma) = \log|\Sigma|$. Indeed, $N_X(n, 2^{-p}) = |\Sigma|^{n+p}$ so $h(X, \sigma, 2^{-p}) = \limsup \frac{(n+p)\log|\Sigma|}{n} = \log|\Sigma|$.
- If the system is not sensitive to initial conditions, then $h(T) = 0$.

  Indeed, the number of distinguishable trajectories of length n is linear in n.
- Having positive entropy is a strong form of chaos.
- Topological entropy is a topological invariant: if two systems are conjugated they have the same entropy.

Topological entropy

A few remarks

- If Y is a singleton, then $h(Y, T) = 0$: it corresponds to the ideal situation when you know exactly the initial state. The evolution is then perfectly predictable (Laplace's demon).
- Hence the unpredictability of a trajectory depends of the observer's knowledge of the initial state.

Topological entropy

A few remarks

- Why $h(\{x\}, T) = 0$?
- Because there is only one trajectory starting from $\{x\}$, as the system is deterministic.
- In other words, one can surround the evolution of the system in a narrow tube.
- But can one do this *effectively*? i.e. generate this tube with a program?

Effective entropy

We define a "constructive version" of topological entropy, which takes this problem of effectivity into account.

Effective entropy

An ϵ**-covering** is a family $\mathbf{E} = (E_n)_{n\in\mathbb{N}}$:

- E_n is a finite set of sequences of length n of representable points,
- every trajectory of length n starting from Y is ϵ-close to a sequence in E_n.

Using ϵ-coverings, the topological entropy is

$$h(Y, T, \epsilon) = \inf_{\epsilon\text{-covering } \mathbf{E}} \left\{ \limsup \frac{\log |E_n|}{n} \right\}.$$

An **effective** ϵ**-covering** is an ϵ-covering $\mathbf{E}$ such that there is a program which on input n, enumerates E_n (E_n is r.e. given n).

Definition (Effective entropy)

$$h_e(Y, T, \epsilon) := \inf_{\text{effective } \epsilon\text{-covering } \mathbf{E}} \left\{ \limsup \frac{\log |E_n|}{n} \right\}.$$

Effective entropy

- Of course, $h(Y, T, \epsilon) \leq h_e(Y, T, \epsilon)$.
- If x is a computable point, then $h_e(\{x\}, T) = 0$. What happens when x is not computable?

Theorem

$$h(T) = \sup_{x \in X} h_e(\{x\}, T).$$

- $h_e(\{x\}, T)$ can be as large as possible.
- $h_e(\{x\}, T)$ expresses, in some way, the effective unpredictability – or *uncomputability* – of the evolution of the system, when starting from x.

Sequel of the talk

We make it more precise and show what this quantity is exactly.

Kolmogorov complexity

Some background

- Let S be a countable set (elements of S can be identified with integers or finite binary strings).
- If $A : \{0,1\}^{\mathbb{N}} \to S$ is a computable function we define the **Kolmogorov complexity** of an element $x \in S$ **relative to** A as

$$K_A(x) = \min\{|p| : p \in \{0,1\}^*, A(p) = x\}.$$

If there is no p such that $A(p) = x$, $K_A(x) = \infty$.

Theorem (Kolmogorov, 1965)

There is a universal optimal function $U : \{0,1\}^* \to S$. For every computable function A there is a constant c_A such that

$$K_U(x) \leq K_A(x) + c_A \quad \textit{for all } x \in S.$$

(c_A is the length of a code for the function A)

Kolmogorov complexity

Some background

We fix U once for all and define the **Kolmogorov complexity** of x as

$$K(x) := K_U(x).$$

Examples

- $S = \{0,1\}^*$. There is a constant c such that

$$K(w) \leq |w| + c \quad \text{for all } w \in \{0,1\}^*.$$

  (consider the function $A(p) = p$).
- $S = \mathbb{N}$. There is a constant c such that $K(n) \leq \log_2(n) + c$ for all $n \in \mathbb{N}$. (consider the function $A(p) = n$ where p is the binary expansion of n).

The function $x \mapsto K(x)$ is not computable. Instead, it is **upper-computable**, i.e. the set $\{(x,k) : K(x) < k\}$ is r.e.

Kolmogorov complexity

Some background

For $w \in \{0,1\}^*$ we know that $K(w) \overset{+}{<} |w|$. This bound is usually tight:

Lemma (1)

$$|\{w \in \{0,1\}^n : K(w) < p\}| < 2^p.$$

In other words, most strings are complex:

- in $\{0,1\}^n$, the proportion of strings of complexity $< p$ is 2^{p-n}, i.e.,
- one half of the strings have complexity $K(w) \geq |w| - 1$,
- one quarter have complexity $K(w) \geq |w| - 2$,
- and so on...

Kolmogorov complexity

Some background

To get upper bounds on K, one usually uses the following lemma.

Lemma (2)

If $E \subseteq \mathbb{N} \times X$ is a r.e. set such that $E_n := \{x : (n, x) \in E\}$ is finite for every n, then there is a constant c such that for every n and $x \in E_n$,

$$K(x) \leq 2 \log n + \log |E_n| + c.$$

Proof.

Represent $x \in E_n$ by n and the index of x in the enumeration of E_n. □

Algorithmic complexity of trajectories

- Let $\epsilon > 0$. Computing a finite trajectory means computing a finite sequence of representable points (rational numbers on $\mathbb{R}$, e.g.) ϵ-close to the actual trajectory.
- We first define the algorithmic complexity of a finite trajectory:

$$\mathcal{K}_n(x, T, \epsilon) = \min\{K(q_0, \ldots, q_{n-1}) : d(q_i, T^i(x)) < \epsilon \text{ for } 0 \leq i < n\}.$$

- Then we consider the growth rate of $\mathcal{K}_n(x, T, \epsilon)$ as $n \to \infty$:

$$\mathcal{K}(x, T, \epsilon) := \limsup \frac{\mathcal{K}_n(x, T, \epsilon)}{n}$$

and

$$\mathcal{K}(x, T) := \lim_{\epsilon \to 0} \mathcal{K}(x, T, \epsilon) = \sup_{\epsilon > 0} \mathcal{K}(x, T, \epsilon).$$

[Brudno, 1983] [Galatolo, 2000]

Algorithmic complexity of trajectories

- Roughly, one needs $n.\mathcal{K}(x, T, \epsilon)$ bits to encode the trajectory of length n starting from x.
- If x is a computable point and T is a computable function then $\mathcal{K}(x, T) = 0$.

  Indeed, $\mathcal{K}_n(x, T, \epsilon) \overset{+}{<} \log(n)$: there is a program which takes n as input and computes the first n iterates of x (up to ϵ).
- If $\mathcal{K}(x, T) > 0$ then the trajectory of x is strongly non-computable: to compute its n first elements, one needs a program of length linear in n.

Relations

Theorem

$$h_e(\{x\}, T) = \mathcal{K}(x, T).$$

Proof of $h_e(\{x\}, T) \leq \mathcal{K}(x, T)$

- Let $\beta > \mathcal{K}(x, T, \epsilon)$: there is n_0 such that for all $n \geq n_0$, $\mathcal{K}_n(x, T, \epsilon) < \beta n$.

Proof of $h_e(\{x\}, T) \leq \mathcal{K}(x, T)$

- Let $\beta > \mathcal{K}(x, T, \epsilon)$: there is n_0 such that for all $n \geq n_0$, $\mathcal{K}_n(x, T, \epsilon) < \beta n$.
- Let
$$E_n := \{(q_0, \dots, q_{n-1}) : K(q_0, \dots, q_{n-1}) < \beta n\}.$$

Proof of $h_e(\{x\}, T) \leq \mathcal{K}(x, T)$

- Let $\beta > \mathcal{K}(x, T, \epsilon)$: there is n_0 such that for all $n \geq n_0$, $\mathcal{K}_n(x, T, \epsilon) < \beta n$.
- Let

$$E_n := \{(q_0, \ldots, q_{n-1}) : K(q_0, \ldots, q_{n-1}) < \beta n\}.$$

- The sequence E_n is an effective ϵ-covering of $\{x\}$, so

$$h_e(\{x\}, T, \epsilon) \leq \limsup \frac{\log |E_n|}{n}.$$

Proof of $h_e(\{x\}, T) \leq \mathcal{K}(x, T)$

- Let $\beta > \mathcal{K}(x, T, \epsilon)$: there is n_0 such that for all $n \geq n_0$, $\mathcal{K}_n(x, T, \epsilon) < \beta n$.
- Let

$$E_n := \{(q_0, \ldots, q_{n-1}) : K(q_0, \ldots, q_{n-1}) < \beta n\}.$$

- The sequence E_n is an effective ϵ-covering of $\{x\}$, so

$$h_e(\{x\}, T, \epsilon) \leq \limsup \frac{\log |E_n|}{n}.$$

- Using a basic property of algorithmic complexity, we get

$$|E_n| \leq 2^{\beta n}.$$

Proof of $h_e(\{x\}, T) \leq \mathcal{K}(x, T)$

- Let $\beta > \mathcal{K}(x, T, \epsilon)$: there is n_0 such that for all $n \geq n_0$, $\mathcal{K}_n(x, T, \epsilon) < \beta n$.
- Let

$$E_n := \{(q_0, \ldots, q_{n-1}) : K(q_0, \ldots, q_{n-1}) < \beta n\}.$$

- The sequence E_n is an effective ϵ-covering of $\{x\}$, so

$$h_e(\{x\}, T, \epsilon) \leq \limsup \frac{\log |E_n|}{n}.$$

- Using a basic property of algorithmic complexity, we get

$$|E_n| \leq 2^{\beta n}.$$

- Hence $h_e(\{x\}, T, \epsilon) \leq \beta$ for every $\beta > \mathcal{K}(x, T, \epsilon)$, so we get:

$$h_e(\{x\}, T, \epsilon) \leq \mathcal{K}(x, T, \epsilon).$$

Proof of $\mathcal{K}(x, T) \leq h_e(\{x\}, T)$

- Let $\mathbf{E} = (E_n)_{n\in\mathbb{N}}$ be an effective ϵ-covering of $\{x\}$.

Proof of $\mathcal{K}(x, T) \leq h_e(\{x\}, T)$

- Let $\mathbf{E} = (E_n)_{n\in\mathbb{N}}$ be an effective ϵ-covering of $\{x\}$.
- Using the lemma, for every $(q_0, \ldots, q_{n-1}) \in E_n$, we have

$$K(q_0, \ldots, q_{n-1}) \overset{+}{<} \log |E_n| + 2\log(n).$$

Proof of $\mathcal{K}(x, T) \leq h_e(\{x\}, T)$

- Let $\mathbf{E} = (E_n)_{n \in \mathbb{N}}$ be an effective ϵ-covering of $\{x\}$.
- Using the lemma, for every $(q_0, \dots, q_{n-1}) \in E_n$, we have

$$K(q_0, \dots, q_{n-1}) \overset{+}{<} \log |E_n| + 2\log(n).$$

- As a result,

$$\begin{aligned} \mathcal{K}_n(x, T, \epsilon) &\overset{+}{<} \log |E_n| + 2\log(n) \\ \mathcal{K}(x, T, \epsilon) &\leq \limsup \frac{\log |E_n|}{n}. \end{aligned}$$

Proof of $\mathcal{K}(x, T) \leq h_e(\{x\}, T)$

- Let $\mathbf{E} = (E_n)_{n\in\mathbb{N}}$ be an effective ϵ-covering of $\{x\}$.
- Using the lemma, for every $(q_0, \ldots, q_{n-1}) \in E_n$, we have

$$K(q_0, \ldots, q_{n-1}) \stackrel{+}{<} \log |E_n| + 2\log(n).$$

- As a result,

$$\begin{aligned}\mathcal{K}_n(x, T, \epsilon) &\stackrel{+}{<} \log |E_n| + 2\log(n)\\ \mathcal{K}(x, T, \epsilon) &\leq \limsup \frac{\log |E_n|}{n}.\end{aligned}$$

- As this is true for every effective ϵ-covering E_n, we get:

$$\mathcal{K}(x, T, \epsilon) \leq h_e(\{x\}, T, \epsilon).$$

Relations

As a result,

$$h(T) = \sup_x \mathcal{K}(x, T).$$

In particular, for a computable system (X, T), the following statements are equivalent:

1. The system is strongly unpredictable, i.e. $h(T) > 0$,
2. The system has at least one trajectory which is strongly non-computable, i.e. satisfying $\mathcal{K}(x, T) > 0$.

Relations

As a result,

$$h(T) = \sup_x \mathcal{K}(x, T).$$

In particular, for a computable system (X, T), the following statements are equivalent:

1. The system is strongly unpredictable, i.e. $h(T) > 0$,
2. The system has at least one trajectory which is strongly non-computable, i.e. satisfying $\mathcal{K}(x, T) > 0$.

Thank you