Information Theory
Part II. Kolmogorov complexity

Damian Niwiński

University of Warsaw

Winter semester 2020/2021

Disclaimer. Credits to many authors. All errors are mine own.

Which sequence is random ?

0000000000000000000000000000000000000000

0100111001010011100101001110010100111001

1100100100001111110110010100010001000010

Which sequence is random ?

0000000000000000000000000000000000000000

0100111001010011100101001110010100111001

1100100100001111110110010100010001000010

π

Do random sequences exist at all ?
Shannon's information theory measures

$$\text{randomness} \approx H(X)$$
$$\text{dependence} \approx I(X;Y)$$

but in Shannon's own words:
Seldom do more than a few of nature's secrets give way at one time.

Turing machines

For a Turing machine M and $w \in \Sigma^*$ (usually $\{0,1\}^*$),

- $M(w)\downarrow$: machine M **halts** on input w,
- $M(w)\uparrow$: machine M **loops** on input w,
- $M(w) = v$: machine M halts on input w and the **output** is v.

Turing machines

For a Turing machine M and $w \in \Sigma^*$ (usually $\{0,1\}^*$),

- $M(w)\downarrow$: machine M **halts** on input w,
- $M(w)\uparrow$: machine M **loops** on input w,
- $M(w) = v$: machine M halts on input w and the **output** is v.

Encoding of Turing machines

$$M \quad \mapsto \quad \langle M \rangle \text{ code of machine } M,$$

$$v \quad \mapsto \quad M_v \text{ machine of code } v.$$

Proviso: the encoding is prefix-free.

Universal Turing machine

A machine U is **universal** if, for any machine M, and $v \in \{0,1\}^*$,

- if $M(v)\downarrow$ then $U(\langle M\rangle\, v)\downarrow$ and $M(v) = U(\langle M\rangle\, v)$,
- if $M(v)\uparrow$ then $U(\langle M\rangle\, v)\uparrow$,
- for all other inputs w, $U(w)\uparrow$.

$$\begin{aligned} \text{Turing machine} &\approx \text{program}, \\ \text{Universal Turing machine} &\approx \end{aligned}$$

Universal Turing machine

A machine U is **universal** if, for any machine M, and $v \in \{0,1\}^*$,

- if $M(v)\downarrow$ then $U(\langle M \rangle\, v)\downarrow$ and $M(v) = U(\langle M \rangle\, v)$,
- if $M(v)\uparrow$ then $U(\langle M \rangle\, v)\uparrow$,
- for all other inputs w, $U(w)\uparrow$.

Turing machine	$\approx$	program,
Universal Turing machine	$\approx$	**compiler**

Kolmogorov complexity

The **Kolmogorov information complexity** of a word $x \in \{0,1\}^*$

$$C_U(x) \quad = \quad \min\{|v| : U(v) = x\}.$$

Kolmogorov complexity

The **Kolmogorov information complexity** of a word $x \in \{0,1\}^*$

$$C_U(x) \quad = \quad \min\{|v| : U(v) = x\}.$$

Lemma. For an arbitrary Turing machine, let

$$C_M(x) \quad = \quad \min\{|v| : M(v) = x\}.$$

Then there is a constant c_{UM}, such that

$$C_U(x) \quad \leq \quad C_M(x) + c_{UM}.$$

Kolmogorov complexity

The **Kolmogorov information complexity** of a word $x \in \{0,1\}^*$

$$C_U(x) \quad = \quad \min\{|v| : U(v) = x\}.$$

Lemma. For an arbitrary Turing machine, let

$$C_M(x) \quad = \quad \min\{|v| : M(v) = x\}.$$

Then there is a constant c_{UM}, such that

$$C_U(x) \quad \leq \quad C_M(x) + c_{UM}.$$

Proof. If $M(v) = x$ then $U(\langle M \rangle v) = x$. Hence

$$C_U(x) \leq \min\{|\langle M \rangle| + |v| : M(v) = x\} = C_M(x) + \underbrace{|\langle M \rangle|}_{c_{UM}}. \qquad \square$$

Kolmogorov complexity

Def. $C_U(x) = \min\{|v| : U(v) = x\}.$

Lemma. $\forall M \, \exists c_{UM}$ $C_U(x) \leq C_M(x) + c_{UM}.$

Corollaries

Invariance. For any two universal Turing machines U, U',

$$|C_U(x) - C_{U'}(x)| = \mathcal{O}(1).$$

□

Kolmogorov complexity				
Def.		$C_U(x)$	$=$	$\min\{\lvert v\rvert : U(v) = x\}.$
Lemma.	$\forall M\ \exists c_{UM}$	$C_U(x)$	$\leq$	$C_M(x) + c_{UM}.$

Corollaries

Invariance. For any two universal Turing machines U, U',

$$|C_U(x) - C_{U'}(x)| \;=\; \mathcal{O}(1).$$

$\square$

Upper bound.

$$C_U(x) \;=\; |x| + \mathcal{O}(1).$$

Kolmogorov complexity

Def. $C_U(x) = \min\{|v| : U(v) = x\}.$

Lemma. $\forall M\ \exists c_{UM}$ $C_U(x) \leq C_M(x) + c_{UM}.$

Corollaries

Invariance. For any two universal Turing machines U, U',

$$|C_U(x) - C_{U'}(x)| = \mathcal{O}(1).$$

□

Upper bound.

$$C_U(x) = |x| + \mathcal{O}(1).$$

Proof. Take M computing identity. □

Military ordering

$$\begin{array}{r} \varepsilon \\ 0,\ 1 \\ 00,\ 01,\ 10,\ 11 \\ 000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111 \\ \cdots\cdots\cdots \end{array}$$

First by length, then lexicographically.

$$\langle \{0,1\}^*, \sqsubseteq \rangle \quad \approx \quad \langle \mathbb{N}, \leq \rangle.$$

Kolmogorov random words

A word x is **Kolmogorov random** if $C_U(x) \geq |x|$.

Kolmogorov random words

A word x is **Kolmogorov random** if $C_U(x) \geq |x|$.

Such words exist. Let

$$\alpha : x \mapsto v, \text{ where } U(v) = x \text{ and } |v| = C_U(x) \quad \textbf{(1:1)}$$

$$x_n \;=\; \min_{\sqsubseteq}\{x : C_U(x) \geq n\}.$$

Then

$$2^{|x_n|} - 1 \;\leq\; |\alpha(\{z : z \sqsubset x_n\})| \;\subseteq\; \left|\{0,1\}^{<n}\right| \;=\; 2^n - 1.$$

Hence

$$|x_n| \;\leq n \;\leq C_U(x_n).$$

Kolmogorov complexity is uncomputable

Suppose there is an algorithm $x \mapsto C_U(x)$.

The one could also compute

$$n \mapsto x_n = \min_{\sqsubseteq}\{x : C_U(x) \geq n\}.$$

Let M be a Turing machine such that $M(\mathbf{bin}(n)) = x_n$.

Kolmogorov complexity is uncomputable

Suppose there is an algorithm $x \mapsto C_U(x)$.

The one could also compute

$$n \mapsto x_n = \min_{\sqsubseteq}\{x : C_U(x) \geq n\}.$$

Let M be a Turing machine such that $M(\textbf{bin}(n)) = x_n$. Then

$$U(\langle M\rangle\,\textbf{bin}(n)) \quad = \quad x_n$$

$$\underbrace{C_U(x_n)}_{n\leq} \quad \leq \quad |\langle M\rangle| + \log n + 1$$

Kolmogorov complexity is uncomputable

Suppose there is an algorithm $x \mapsto C_U(x)$.

The one could also compute

$$n \mapsto x_n = \min_{\sqsubseteq}\{x : C_U(x) \geq n\}.$$

Let M be a Turing machine such that $M(\mathbf{bin}(n)) = x_n$. Then

$$U(\langle M \rangle\, \mathbf{bin}(n)) \;=\; x_n$$

$$\underbrace{C_U(x_n)}_{n \leq} \;\leq\; |\langle M \rangle| + \log n + 1$$

Impossible for sufficiently large n, **contradiction !**

Prefix-free Kolmogorov complexity

We call Turing machine M **prefix-free** if so is

$$L(M) \quad = \quad \{x : M(x)\downarrow\}.$$

Prefix-free Kolmogorov complexity

We call Turing machine M **prefix-free** if so is

$$L(M) \quad = \quad \{x : M(x)\downarrow\}.$$

The **prefix-free Kolmogorov complexity** of $x \in \{0,1\}^*$

$$\mathbf{K}_U(x) \quad = \quad \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

Prefix-free Kolmogorov complexity

We call Turing machine M **prefix-free** if so is

$$L(M) \quad = \quad \{x : M(x)\downarrow\}.$$

The **prefix-free Kolmogorov complexity** of $x \in \{0,1\}^*$

$$\mathbf{K}_U(x) \quad = \quad \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

A machine U is **prefix-free universal** if it is prefix-free and, for any prefix-free machine M, and $v \in \{0,1\}^*$,

- if $M(v)\downarrow$ then $U(\langle M \rangle\, v)\downarrow$ and $M(v) = U(\langle M \rangle\, v)$,
- if $M(v)\uparrow$ then $U(\langle M \rangle\, v)\uparrow$,
- for inputs w not in the form $\langle M \rangle\, v$, for some M (not necessarily prefix-free), $U(w)\uparrow$.

Prefix-free Kolmogorov complexity

We call Turing machine M **prefix-free** if so is

$$L(M) \quad = \quad \{x : M(x)\downarrow\}.$$

The **prefix-free Kolmogorov complexity** of $x \in \{0,1\}^*$

$$\mathbf{K}_U(x) \quad = \quad \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

A machine U is **prefix-free universal** if it is prefix-free and, for any prefix-free machine M, and $v \in \{0,1\}^*$,

- if $M(v)\downarrow$ then $U(\langle M\rangle\, v)\downarrow$ and $M(v) = U(\langle M\rangle\, v)$,
- if $M(v)\uparrow$ then $U(\langle M\rangle\, v)\uparrow$,
- for inputs w not in the form $\langle M\rangle\, v$, for some M (not necessarily prefix-free), $U(w)\uparrow$.

Do such machines exist ?

Prefix-free Kolmogorov complexity

Lemma. There is an algorithm

arbitrary machine $M \quad \mapsto \quad$ **prefix-free** machine M' such that

- $L(M') \subseteq L(M)$,

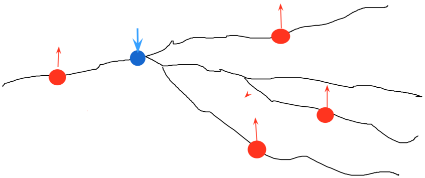
Prefix-free Kolmogorov complexity

Lemma. There is an algorithm

arbitrary machine $M \mapsto$ **prefix-free** machine M' such that

- $L(M') \subseteq L(M)$,
- if $M(x)\downarrow$ and, for all y such that $y < x$ or $x < y$, $M(y)\uparrow$, then

$$M'(x) = M(x).$$



Thus, if M is originally prefix-free then $L(M') = L(M)$.

Proof. $M \mapsto M'$

1. Input (for M'): $= x = x_1 \dots x_k$.
2. $A := \varepsilon$ (* A will run over prefixes of x *).
3. For all words $w_i = \varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots$ in the **zigzag manner** simulate M on Aw.

   Specifically: in the i-th phase, make the next step of the computation of M on $Aw_0, Aw_1, \dots, Aw_{i-1}$, and the first step on Aw_i.

   If $M(Aw_i)\downarrow$, **goto** 4.

4. **if** $w_i = \varepsilon$ **then**

   if $A = x$ **then** **ACCEPT** (* Output $M(Aw_i) = M(x)$ *)

   else **REJECT**

   else

   if $A = x_1 \dots x_\ell$, $\ell < k$ **then** $A := x_1 \dots x_\ell\, x_{\ell+1}$; **goto** 3

   else (* $w_i > \varepsilon \wedge A = x$ *) **REJECT**

□

Prefix-free universal Turing machine

Recall: U is **prefix-free universal** if, for M **prefix-free**

- if $M(v)\downarrow$ then $U(\langle M\rangle\, v)\downarrow$ and $M(v) = U(\langle M\rangle\, v)$,
- if $M(v)\uparrow$ then $U(\langle M\rangle\, v)\uparrow$,
- for all other inputs w, $U(w)\uparrow$.

Claim. The machine obtained from an ordinary universal U by the construction $U \mapsto U'$ of the Lemma is prefix-free universal.

Indeed, if $M(v)\downarrow$ then, for all y such that $y < \langle M\rangle\, v$ or $\langle M\rangle\, v < y$, it holds $U(y)\uparrow$.

Halting problem

Theorem (Turing). The halting problem for the universal Turing machine is undecidable.

Corollary. The halting problem for the **prefix-free** universal Turing machine is undecidable.

Proof. Let

$$\alpha : w_1\, w_2 \dots w_n \mapsto w_1\mathbf{0}\, w_2\mathbf{0} \dots w_n\mathbf{0}\,\mathbf{01}.$$

Let $M \mapsto M^\alpha$, such that $M^\alpha(\alpha(w)) \downarrow \iff M(w) \downarrow$.

As M^α is prefix-free, we have

$$\begin{aligned} U(w) \downarrow \quad &\iff \quad U^\alpha(\alpha(w)) \downarrow \\ &\iff \quad U'(\langle U^\alpha \rangle \alpha(w)) \downarrow . \end{aligned}$$

Thus we reduce the halting problem for U to the halting problem for U'. □

Properties of the prefix-free Kolmogorov complexity

$$\mathbf{K}_U(x) \quad = \quad \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

Invariance. For any two prefix-free universal Turing machines U, U',

$$|K_U(x) - K_{U'}(x)| \quad = \quad \mathcal{O}(1).$$

Properties of the prefix-free Kolmogorov complexity

$$\mathbf{K}_U(x) \quad = \quad \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

Invariance. For any two prefix-free universal Turing machines U, U',

$$|K_U(x) - K_{U'}(x)| \quad = \quad \mathcal{O}(1).$$

Uncomputability. The mapping $x \mapsto K_U(x)$ is uncomputable.

Properties of the prefix-free Kolmogorov complexity

$$\mathbf{K}_U(x) \quad = \quad \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

Invariance. For any two prefix-free universal Turing machines U, U',

$$|K_U(x) - K_{U'}(x)| \quad = \quad \mathcal{O}(1).$$

Uncomputability. The mapping $x \mapsto K_U(x)$ is uncomputable.

Upper bound (not optimal).

$$K_U(x) \quad = \quad |x| + \mathcal{O}(\log |x|).$$

Properties of the prefix-free Kolmogorov complexity

$$\mathbf{K}_U(x) \;=\; \min\{|v| : U(v) = x\},$$

for a **prefix-free universal** machine U.

Invariance. For any two prefix-free universal Turing machines U, U',

$$|K_U(x) - K_{U'}(x)| \;=\; \mathcal{O}(1).$$

Uncomputability. The mapping $x \mapsto K_U(x)$ is uncomputable.

Upper bound (not optimal).

$$K_U(x) \;=\; |x| + \mathcal{O}(\log |x|).$$

$$M : a_1\mathbf{0}\, a_2\mathbf{0}\, \ldots a_k\mathbf{0}\,\mathbf{01}\; x \mapsto x,$$

where $a_1 a_2 \ldots a_k$ is the binary representation of the **length** of x.

Chaitin constant

For a prefix-free universal machine U,

$$\Omega = \sum_{U(v)\downarrow} 2^{-|v|}.$$

By Kraft's inequality, $\Omega \leq 1$.

Intuitively: probability that U halts.

Chaitin constant

For a prefix-free universal machine U,

$$\Omega \quad = \quad \sum_{U(v)\downarrow} 2^{-|v|}.$$

By Kraft's inequality, $\Omega \leq 1$.

Intuitively: probability that U halts.

More specifically,

$$\Omega \quad = \quad p\left(U(X \restriction n)\downarrow \ \text{ for some } n\right)$$

viewing $\{0,1\}^\omega \ni X = X_0, X_1, X_2, \ldots$ as a result of an infinite Bernoulli process with $p(X_i = 0) = p(X_i = 1) = \frac{1}{2}$.

Chaitin constant

Theorem.

- There is a Turing machine T with an extra tape containing

$$\Omega \quad = \quad 0.\omega_1\omega_2\omega_3\ldots\ldots$$

  which solves the halting problem for U.

- There is a constant c, such that, for $n \in \mathbb{N}$,

$$K_U(\omega_1\ldots\omega_n) \quad \geq \quad n - c,$$

  i.e., Ω is incompressible.

Proof. If Ω may have two representations

$$\begin{aligned}\Omega &= 0.\omega_1\omega_2\ldots\omega_k 1000\ldots \\ &= \underbrace{0.\omega_1\omega_2\ldots\omega_k 0111\ldots}_{\textbf{choose this one}}\end{aligned}$$

T simulates $U(w)$ and simultaneously, $U(y)$, for all words y in **zigzag manner**, keeping

$$\mathcal{S} = \{y : U(y)\downarrow \text{ so far }\}.$$

- If $U(w)\downarrow$ then T says **YES**.
- If $0.\omega_1\omega_2\ldots\omega_n < \sum_{y\in\mathcal{S}} 2^{-|y|}$, where $n = |w|$, and $w \notin \mathcal{S}$, then T says **NO**.

Consequently, Ω is **irrational**, hence $\Omega < 1$.

Proof of incompressibility. We define a machine R.

For input x, R simulates $U(x)$. Suppose $U(x) = \omega_1\omega_2\ldots\omega_n$.

Next, R simulates $U(y)$, for all words y in zigzag manner, keeping $\mathcal{S}$ as before.

At some moment $0.\,\omega_1\omega_2\ldots\omega_n < \sum_{y\in\mathcal{S}} 2^{-|y|}$.

Proof of incompressibility. We define a machine R.

For input x, R simulates $U(x)$. Suppose $U(x) = \omega_1 \omega_2 \ldots \omega_n$.

Next, R simulates $U(y)$, for all words y in zigzag manner, keeping $\mathcal{S}$ as before.

At some moment $0.\omega_1 \omega_2 \ldots \omega_n < \sum_{y \in \mathcal{S}} 2^{-|y|}$.

Let v be the first such that $v \neq U(y)$, for all $y \in \mathcal{S}$.

Then R **stops** with $R(x) = v$.

Proof of incompressibility. We define a machine R.

For input x, R simulates $U(x)$. Suppose $U(x) = \omega_1\omega_2\ldots\omega_n$.

Next, R simulates $U(y)$, for all words y in zigzag manner, keeping $\mathcal{S}$ as before.

At some moment $0.\omega_1\omega_2\ldots\omega_n < \sum_{y\in\mathcal{S}} 2^{-|y|}$.

Let v be the first such that $v \neq U(y)$, for all $y \in \mathcal{S}$.

Then R **stops** with $R(x) = v$.

Note: $K_U(v) >$

Proof of incompressibility. We define a machine R.

For input x, R simulates $U(x)$. Suppose $U(x) = \omega_1\omega_2\ldots\omega_n$.

Next, R simulates $U(y)$, for all words y in zigzag manner, keeping $\mathcal{S}$ as before.

At some moment $0.\omega_1\omega_2\ldots\omega_n < \sum_{y\in\mathcal{S}} 2^{-|y|}$.

Let v be the first such that $v \neq U(y)$, for all $y \in \mathcal{S}$.

Then R **stops** with $R(x) = v$.

Note: $K_U(v) > n$.

But $K_U(v) \leq \underbrace{K_R(v)}_{\leq|x|} + c_{UR}$. Hence

$$n < K_U(v) \leq K_R(v) + c_{UR} \leq |x| + c_{UR},$$

for any x, such that $U(x) = \omega_1\omega_2\ldots\omega_n$.

Proof of incompressibility. We define a machine R.

For input x, R simulates $U(x)$. Suppose $U(x) = \omega_1\omega_2\ldots\omega_n$.

Next, R simulates $U(y)$, for all words y in zigzag manner, keeping $\mathcal{S}$ as before.

At some moment $0.\omega_1\omega_2\ldots\omega_n < \sum_{y\in\mathcal{S}} 2^{-|y|}$.

Let v be the first such that $v \neq U(y)$, for all $y \in \mathcal{S}$.

Then R **stops** with $R(x) = v$.

Note: $K_U(v) > n$.

But $K_U(v) \leq \underbrace{K_R(v)}_{\leq|x|} + c_{UR}$. Hence

$$n < K_U(v) \leq K_R(v) + c_{UR} \leq |x| + c_{UR},$$

for any x, such that $U(x) = \omega_1\omega_2\ldots\omega_n$.

Hence

$$n \leq K_U(\omega_1\omega_2\ldots\omega_n) + c_{UR}.$$

Algorithmic probability

Recall

$$\Omega = \sum_{U(v)\downarrow} 2^{-|v|}.$$

How to interpret

$$p_U(y) = \sum_{v:U(v)=y} 2^{-|v|} \quad \textbf{?}$$

Algorithmic probability

Recall

$$\Omega = \sum_{U(v)\downarrow} 2^{-|v|}.$$

How to interpret

$$p_U(y) = \sum_{v:U(v)=y} 2^{-|v|} \quad \textbf{?}$$

Note

$$1 = \overbrace{\sum_y p_U(y)}^{\Omega} + p\,(\,\text{U diverges}\,).$$

$p_U(y) \approx$ probability that a random programe generates y.

Example. Compare $p_U(0^n)$ vs. $p_U(\omega_1 \ldots \omega_n)$.

Algorithmic probability

Recall that, in an optimal encoding $\varphi : S \to \{0,1\}^*$,

$$|\varphi(s)| \quad \approx \quad -\log p(s).$$

We will show

$$K_U(y) \quad \approx \quad -\log p_U(y).$$

Algorithmic probability

Recall that, in an optimal encoding $\varphi : S \to \{0,1\}^*$,

$$|\varphi(s)| \quad \approx \quad -\log p(s).$$

We will show

$$K_U(y) \quad \approx \quad -\log p_U(y).$$

Theorem. There is a constant $c > 0$, such that, for all $y \in \{0,1\}^*$,

$$K_U(y) - c \leq -\log p_U(y) \ \leq \ K_U(y).$$

Algorithmic probability $p_U(y) = \sum_{v:U(v)=y} 2^{-|v|}$.

Theorem. $K_U(y) - c \leq -\log p_U(y) \;\leq\; K_U(y)$.

Proof.

We have $U(x) = y$, for some x, such that $K_U(y) = |x|$, hence

$$\frac{1}{2^{|x|}} \leq p_U(y) \quad \text{and} \quad -\log p_U(y) \leq |x| = K_U(y).$$

Algorithmic probability $p_U(y) = \sum_{v: U(v)=y} 2^{-|v|}$.

Theorem. $K_U(y) - c \leq -\log p_U(y) \ \leq \ K_U(y)$.

Proof.

We have $U(x) = y$, for some x, such that $K_U(y) = |x|$, hence

$$\frac{1}{2^{|x|}} \leq p_U(y) \quad \text{and} \quad -\log p_U(y) \leq |x| = K_U(y).$$

For the other inequality, we will define a prefix-free machine T, such that, forall y, there is w_y, such that $T(w_y) = y$, and

$$|w_y| \ \leq \ -\log p_U(y) + d,$$

which will imply

$$K_U(y) \ \leq \ |\langle T \rangle| + |w_y| \ \leq \ -\log p_U(y) + \underbrace{|\langle T \rangle| + d}_{c}.$$

Proof of $K_U(y) - c \leq -\log p_U(y)$.

A **binary interval** is of the form

$$\left[\underbrace{a_1 \cdot \frac{1}{2} + a_2 \cdot \frac{1}{2^2} + \ldots + a_k \cdot \frac{1}{2^k}}_{L}, L + \frac{1}{2^k}\right),$$

where $a_1, \ldots, a_k \in \{0, 1\}$; $a_k = 1$.

For example,

$$\left[\frac{1}{2}, 1\right), \quad \left[\frac{3}{8}, \frac{1}{2}\right), \quad \left[\underbrace{\frac{1}{4} + \frac{1}{8} + \frac{1}{32}}_{\frac{13}{32}}, \frac{7}{16}\right).$$

Proof of $K_U(y) - c \leq -\log p_U(y)$.

A **binary interval** is of the form

$$\Bigg[\underbrace{a_1 \cdot \frac{1}{2} + a_2 \cdot \frac{1}{2^2} + \ldots + a_k \cdot \frac{1}{2^k}}_{L}, L + \frac{1}{2^k}\Bigg),$$

where $a_1, \ldots, a_k \in \{0, 1\}$; $a_k = 1$.

For example,

$$\left[\frac{1}{2}, 1\right), \quad \left[\frac{3}{8}, \frac{1}{2}\right), \quad \Bigg[\underbrace{\frac{1}{4} + \frac{1}{8} + \frac{1}{32}}_{\frac{13}{32}}, \frac{7}{16}\Bigg).$$

Note: all extensions $0.a_1 \ldots a_k\, \mathbf{v}$ are in $[L, L + \frac{1}{2^k})$.

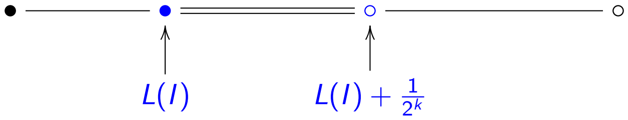
Proof of $K_U(y) - c \leq -\log p_U(y)$.

A **binary interval**

$$\Bigg[\underbrace{a_1 \cdot \frac{1}{2} + a_2 \cdot \frac{1}{2^2} + \ldots + a_k \cdot \frac{1}{2^k}}_{L}, L + \frac{1}{2^k}\Bigg),$$

For an interval $I = [a, b)$, let

$$L(I) \quad = \quad \text{the left end of a maximal binary interval } B \subseteq I \text{ (the leftmost one)}$$

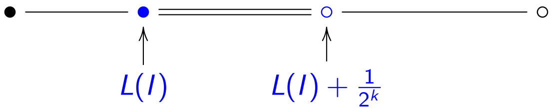


Proof of $K_U(y) - c \leq -\log p_U(y)$.

Lemma. If B is a maximal binary interval contained in a half-open interval I then

$$8 \cdot |B| \quad \geq \quad |I|.$$

Proof of the lemma.



Count how many steps of length $\frac{1}{2^k}$ can we perform from L to the right, and to the left.

Machine T

For an input x, T simulates $U(z)$ in zigzag manner, keeping, $\forall y$

$$
\begin{aligned}
Z_{t,y} &= \{z : U(z) = y \text{ and } U(z)\downarrow \text{ in } \leq t \text{ steps}\} \\
\varphi(t,y) &= \sum_{z \in Z_{t,y}} \frac{1}{2^{|z|}}.
\end{aligned}
$$

For a given t, $\varphi(t,y) > 0$, only for **finitely many** y.

For any y,

$$
\lim_{t\to\infty} \varphi(t,y) = p_U(y).
$$

Machine T

For an input x, T simulates $U(z)$ in zigzag manner, keeping, $\forall y$

$$\begin{aligned} Z_{t,y} &= \{z : U(z) = y \text{ and } U(z)\downarrow \text{ in } \leq t \text{ steps}\} \\ \varphi(t,y) &= \sum_{z \in Z_{t,y}} \frac{1}{2^{|z|}}. \end{aligned}$$

For a given t, $\varphi(t,y) > 0$, only for **finitely many** y.

For any y,

$$\lim_{t\to\infty} \varphi(t,y) = p_U(y).$$

Approximation:

$$\psi(t,y) = \max\left\{\frac{1}{2^k} : \frac{1}{2^k} \leq \varphi(t,y)\right\}.$$

Note:

$$\psi(t,y) > \frac{1}{2}\varphi(t,y).$$

Recall

$$
\begin{aligned}
Z_{t,y} &= \{z : U(z) = y \text{ and } U(z)\downarrow \text{ in } \leq t \text{ steps}\} \\
\varphi(t,y) &= \sum_{z \in Z_{t,y}} \frac{1}{2^{|z|}} \\
\psi(t,y) &= \max\{\frac{1}{2^k} : \frac{1}{2^k} \leq \varphi(t,y)\}.
\end{aligned}
$$

Whenever, for some y, $\psi(t,y)$ incresases, **mark a half-open segment** $I_{t,y}$ with

$$
|I_{t,y}| = \frac{1}{2}\psi(t,y).
$$

Note that since $\forall a > 0$, $\sum_{\frac{1}{2^k} \leq a} \frac{1}{2^k} \leq 2a$,

the total length of all segments does not exceed $\Omega < 1$.

Machine T with input x

Find $L(I_{t,y})$.

If $L(I_{t,y}) = x$ then $T(x)\downarrow$ and $T(x) = y$.

- T is prefix-free because the left-ends of disjoint binary intervals are prefix-free.
- $\forall y \, \exists x \; T(x) = y$.

How is $|x|$ related to $|I_{t,y}| = \frac{1}{2}\psi(t, y)$?

Proof of $K_U(y) - c \leq -\log p_U(y)$.

If $L(I_{t,y}) = x$ then $T(x) = y$.

Hence, the length of the **largest binary interval** $B \subseteq I_{t,y}$ is $\frac{1}{2^{|x|}}$.
By the Lemma ($\mathbf{8} \cdot |B| \geq |I|$),
Therefore

$$\frac{1}{2^{|x|}} \geq \frac{1}{8} \cdot |I_{t,y}|$$

Take t, such that $\varphi(t,y) \geq \frac{1}{2} \cdot p_U(y)$. Since $\psi(t,y) > \frac{1}{2} \cdot \varphi(t,y)$, we have

$$\frac{1}{2^{|x|}} \geq \frac{1}{8}|I_{t,y}| = \frac{1}{16}\psi(t,y) \geq \frac{1}{16} \cdot \frac{1}{4} p_U(y).$$

Hence

$$K_T(y) \leq |x| \leq -\log p_U(y) + 6.$$

Effective tests

A **test** is a mapping $\delta : \{0,1\}^* \to \mathbb{N}$, such that the set

$$\{\langle m, x\rangle : \delta(x) \geq m\}$$

is partially computable, and, for all m, n,

$$\frac{\sharp\{w \in \{0,1\}^n : \delta(w) \geq m\}}{2^n} \quad \leq \quad \frac{1}{2^m}.$$

Effective tests

A **test** is a mapping $\delta : \{0,1\}^* \to \mathbb{N}$, such that the set

$$\{\langle m, x\rangle : \delta(x) \geq m\}$$

is partially computable, and, for all m, n,

$$\frac{\sharp\{w \in \{0,1\}^n : \delta(w) \geq m\}}{2^n} \quad \leq \quad \frac{1}{2^m}.$$

An infinite sequence $u \in \{0,1\}^{\mathbb{N}}$ is **special** with respect to δ, if

$$\limsup_{n\to\infty} \delta(u \upharpoonright n) \quad = \quad \infty.$$

Effective tests

Example. Is it special ?

$$1\,1\,1\,0\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,0\,1\,0\,1\,1\,1\,0\,1\ \ldots$$

Effective tests

Example. Is it special **?**

$$1\,1\,1\,0\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,0\,1\,0\,1\,1\,1\,0\,1\,\ldots$$

$$\mathbf{1}\,1\,\mathbf{1}\,0\,\mathbf{1}\,0\,\mathbf{1}\,1\,\mathbf{1}\,0\,\mathbf{1}\,1\,\mathbf{1}\,0\,\mathbf{1}\,1\,\mathbf{1}\,0\,\mathbf{1}\,0\,\mathbf{1}\,0\,\mathbf{1}\,1\,\mathbf{1}\,0\,\mathbf{1}\,\ldots$$

Define

$$\delta(x) = \max\{i : x_1 = x_3 = \ldots = x_{2i-1} = \mathbf{1}\}.$$

Effective tests

Example. Is it special **?**

$$1\,1\,1\,0\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,0\,1\,0\,1\,1\,1\,0\,1\,\ldots$$

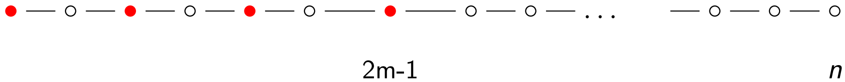
$$\color{red}{1}\,1\,\color{red}{1}\,0\,\color{red}{1}\,0\,\color{red}{1}\,1\,\color{red}{1}\,0\,\color{red}{1}\,1\,\color{red}{1}\,0\,\color{red}{1}\,1\,\color{red}{1}\,0\,\color{red}{1}\,0\,\color{red}{1}\,0\,\color{red}{1}\,1\,\color{red}{1}\,0\,\color{red}{1}\,\ldots$$

Define

$$\delta(x) \quad = \quad \max\{i : x_1 = x_3 = \ldots = x_{2i-1} = \mathbf{1}\}.$$

Then

$$\frac{\sharp\{w \in \{0,1\}^n : \delta(w) \geq m\}}{2^n} = \frac{2^{n-m}}{2^n} = \frac{1}{2^m}$$



Universal test

An infinite sequence $u \in \{0,1\}^{\mathbb{N}}$ is **Martin Löf random** if it is **not special** with respect to any test δ.

Universal test

An infinite sequence $u \in \{0,1\}^{\mathbb{N}}$ is **Martin Löf random** if it is **not special** with respect to any test δ.

One test suffices !

Theorem. There is a test δ^U, such that, for any test δ, and $x \in \{0,1\}^*$,

$$\delta^U(x) \quad \geq \quad \delta(x) - c_\delta,$$

where c_δ is a constant (depending on δ).

Remark. An infinite sequence $u \in \{0,1\}^{\mathbb{N}}$ is **Martin Löf random** if and only if it is **not special** with respect to δ^U.

Indeed, if $\limsup_{n\to\infty} \delta(u \upharpoonright n) = \infty$, for some δ, then

$$\limsup_{n\to\infty} \delta^U(u \upharpoonright n) \geq \limsup_{n\to\infty} \delta(u \upharpoonright n) - c_\delta = \infty.$$

Universal test $\delta^U(x) \geq \delta(x) - c_\delta$

Proof.

Lemma. There exists an effective enumeration of tests $\delta_1, \delta_2, \ldots$

$$\delta^U(x) \overset{def}{=} \max\left(\{\delta_n(x) - n : n \geq 1\} \cup \{0\}\right).$$

For $|x| < n$, $\delta_n(x) \leq |x| < n$, hence max is well-defined.

The universality: $\delta^U(x) \geq \delta_n(x) - n$, by definition.

The effectiveness condition follows from the lemma.

$$\begin{aligned}\sharp\{w \in \{0,1\}^n : \delta^U(w) \geq m\} &\leq \sum_{k=1}^{\infty} \underbrace{\sharp\{w \in \{0,1\}^n : \delta_k(w) \geq m + k\}}_{\leq 2^{n-m-k}} \\ &\leq 2^{n-m} \cdot \underbrace{\sum_{k=1}^{\infty} \frac{1}{2^k}}_{1}.\end{aligned}$$

Kolmogorov complexity vs. Shannon entropy

Let $\mathbf{u} = y_1 y_2 \dots y_m \in \{0,1\}^*$, with $|y_i| = n$.

For $w \in \{0,1\}^n$, define

$$p(w) \quad = \quad \frac{\sharp\{i : w_i = w\}}{m}.$$

Kolmogorov complexity vs. Shannon entropy

Let $\mathbf{u} = y_1 y_2 \ldots y_m \in \{0,1\}^*$, with $|y_i| = n$.

For $w \in \{0,1\}^n$, define

$$p(w) \quad = \quad \frac{\sharp\{i : w_i = w\}}{m}.$$

Note

$$\sum_{w \in \{0,1\}^n} p(w) \quad = \quad 1.$$

Fact. For fixed n,

$$K(\mathbf{u}) \quad \leq \quad m \cdot \left(\sum_{w \in \{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)} \right) + \mathcal{O}(\log m).$$

$$K(\mathbf{u}) \leq m \cdot \left(\sum_{w\in\{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)}\right) + \mathcal{O}(\log m)$$

Proof.

We generate $\mathbf{u} = y_1 y_2 \ldots y_m$ from

- the frequencies $p(w)$, for $w \in \{0,1\}^n$,
- the position $\mathbf{j_u}$ of $\mathbf{u}$ among all the words in $\{0,1\}^{n \cdot m}$ with these frequencies.

$$K(\mathbf{u}) \leq m \cdot \left(\sum_{w\in\{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)} \right) + \mathcal{O}(\log m)$$

Proof.

We generate $\mathbf{u} = y_1 y_2 \dots y_m$ from

- the frequencies $p(w)$, for $w \in \{0,1\}^n$,
- the position $\mathbf{j_u}$ of $\mathbf{u}$ among all the words in $\{0,1\}^{n\cdot m}$ with these frequencies.

Let $\{0,1\}^n \ni w_1, w_2, \dots, w_{2^n}$, in lexicographic order.

$$\begin{aligned} s_i &= \sharp\{j : y_j = w_i\} \text{ (in binary of length } \lfloor \log m \rfloor + 1) \\ X_{\mathbf{u}} &= s_1\, s_2 \dots s_{2^n} \end{aligned}$$

For example $(n = 3, m = 7)$

$$\mathbf{u} \quad = \quad 001\ 110\ 001\ 001\ 111\ 110\ 000$$

$$K(\mathbf{u}) \leq m \cdot \left(\sum_{w\in\{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)}\right) + \mathcal{O}(\log m)$$

Proof.

We generate $\mathbf{u} = y_1 y_2 \dots y_m$ from

- the frequencies $p(w)$, for $w \in \{0,1\}^n$,
- the position $\mathbf{j_u}$ of $\mathbf{u}$ among all the words in $\{0,1\}^{n \cdot m}$ with these frequencies.

Let $\{0,1\}^n \ni w_1, w_2, \dots, w_{2^n}$, in lexicographic order.

$$\begin{aligned} s_i &= \sharp\{j : y_j = w_i\} \text{ (in binary of length } \lfloor \log m \rfloor + 1) \\ X_{\mathbf{u}} &= s_1\, s_2 \dots s_{2^n} \end{aligned}$$

For example $(n = 3, m = 7)$

$$\begin{aligned} \mathbf{u} &= 001\ 110\ 001\ 001\ 111\ 110\ 000 \\ X_{\mathbf{u}} &= \underbrace{001}_{000}\ \underbrace{011}_{001}\ 000\ 000\ 000\ 000\ \underbrace{010}_{110}\ \underbrace{001}_{111}. \end{aligned}$$

Proof of $K(\mathbf{u}) \le m \cdot \left(\sum_{w \in \{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)}\right) + \mathcal{O}(\log m)$

Estimation of $1 \le \mathbf{j_u} \le \left(\frac{m!}{s_1!\, s_2! \ldots s_N!} \right)$, with $N = 2^n$.

Recall the Stirling formula

$$\log k! \quad = \quad k \log k - k \log e + \mathcal{O}(\log k).$$

$$\begin{aligned}
\log \left(\frac{m!}{s_1!\, s_2! \ldots s_{2^n}!} \right) \quad &= \quad \log m! - \log s_1! - \ldots - \log s_N! \\
&= \quad \underbrace{m}_{s_1 + \ldots + s_N} \log m - (m - s_1 - \ldots - s_N) \cdot \log e \\
&\qquad -s_1 \log s_1 - \ldots - s_N \log s_N + \mathcal{O}(n \cdot \log m) \\
&= \quad -m \cdot \left(\frac{s_1}{m} \log \frac{s_1}{m} + \ldots + \frac{s_N}{m} \log \frac{s_N}{m} \right) + \mathcal{O}(n \log m)
\end{aligned}$$

Proof of $K(\mathbf{u}) \leq m \cdot \left(\sum_{w \in \{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)}\right) + \mathcal{O}(\log m)$

We generate $\mathbf{u} = y_1 y_2 \ldots y_m$ from

$$\langle m, n, s_1\, s_2 \ldots s_{2^n}, \mathbf{j_u} \rangle \qquad \square$$

Proof of $K(\mathbf{u}) \leq m \cdot \left(\sum_{w\in\{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)}\right) + \mathcal{O}(\log m)$

We generate $\mathbf{u} = y_1 y_2 \dots y_m$ from

$$\langle m, n, s_1\, s_2 \dots s_{2^n}, \mathbf{j_u}\rangle \qquad \square$$

Corollary. For $\mathbf{Y} = (Y_1, \dots, Y_m)$, where $Y_i \in \{0,1\}^n$,

$$p(Y_i = w_j) = \frac{s_j}{m}, \quad \text{for } j = 1, \dots, 2^n, \quad Y_1, \dots, Y_m \text{ independent},$$

$$K(\mathbf{u}) \quad \leq \quad H(\mathbf{Y}) + \mathcal{O}(\log m).$$

Proof of $K(\mathbf{u}) \leq m \cdot \left(\sum_{w\in\{0,1\}^n} p(w) \cdot \log \frac{1}{p(w)}\right) + \mathcal{O}(\log m)$

We generate $\mathbf{u} = y_1 y_2 \dots y_m$ from

$$\langle m, n, s_1\, s_2 \dots s_{2^n}, \mathbf{j_u} \rangle \qquad \square$$

Corollary. For $\mathbf{Y} = (Y_1, \dots, Y_m)$, where $Y_i \in \{0,1\}^n$,

$p(Y_i = w_j) = \dfrac{s_j}{m}$, for $j = 1, \dots, 2^n$, $Y_1, \dots, Y_m$ independent,

$$K(\mathbf{u}) \;\; \leq \;\; H(\mathbf{Y}) + \mathcal{O}(\log m).$$

On the other hand,

$$H(\mathbf{Y}) \;\; \leq \;\; \mathbf{E}K(\mathbf{Y}).$$

Applications

The Gödel incompleteness theorem. For any sufficiently reach and consistent theory $\mathcal{T}$, there is a **true** property of natural numbers expressible in $\mathcal{T}$, but **not provable** in $\mathcal{T}$.

Proof by Chaitin (sketch).

Applications

The Gödel incompleteness theorem. For any sufficiently reach and consistent theory $\mathcal{T}$, there is a **true** property of natural numbers expressible in $\mathcal{T}$, but **not provable** in $\mathcal{T}$.

Proof by Chaitin (sketch).

Assume $\mathcal{T}$ can express $U(x) = y$.

$$C(k, n) \quad \equiv \quad k = \min\{m : C_U(m) \geq n\}$$

Suppose that, for all numbers k_n, such that $C(k_n, n)$ is true, $\mathcal{T} \vdash C(k_n, n)$.

Then the following algorithm generates k_n from $bin(n)$.

Examine all proofs in $\mathcal{T}$ until you find a proof of $C(k_n, n)$.

Contradiction ! $\square$

Gambling

The hors race

m	horses
M	gambler's initial wealth
b_i	the fraction invested in horse i $b_1 + \ldots + b_m = 1$
$o_i \cdot b_i \cdot M$	the gain if horse i wins
p_i	the (estimated) **probability** that horse i will win

How to play ?

Gambling

M	gambler's initial wealth
b_i	the fraction invested in horse i
$o_i \cdot b_i \cdot M$	the gain if horse i wins
p_i	the **probability** that horse i will win $p_i = p(X = i)$, where $X \in \{1, \ldots, m\}$.

$$S(X) = o_X \cdot b_X$$

$$\mathbb{E}(\log S(X)) = \sum_{i=1}^{m} p_i \cdot \log o_i \cdot b_i$$

doubling rate

Optimize the gain

Playing n times with the results $X_1, \ldots, X_n$ independent $\sim X$

$$S_n = S(X_1) \cdot \ldots \cdot S(X_n)$$

Then

$$\frac{1}{n} \log S_n = \frac{1}{n} \sum_{i=1}^{n} \log S(X_i) \to \mathbb{E}(\log S(X)) \text{ in probability}$$

Thus

$$\begin{aligned} S_n &\doteq 2^{n \cdot \mathbb{E}(\log S(X))} \\ &= 2^{n \cdot \overbrace{\sum p_i \log o_i b_i}^{\text{max?}}} \end{aligned}$$

Optimize the gain

$$S_n \doteq 2^{n \cdot \sum p_i \log o_i b_i}$$

Thus, to optimize the gain S_n, we have to maximize

$$\sum p_i \log o_i b_i = \sum p_i \log o_i + \sum p_i \log b_i$$

Since $\sum b_i = 1$, then by the **Golden Lemma** *à rebours*,

$$\sum p_i \log b_i \leq -H(X)$$

and is **maximal** for $b_i = p_i$.

So, the best strategy is to invest fraction p_i in horse i.

Fairness

$$\sum p_i \log o_i b_i = \sum p_i \log o_i + \underbrace{\sum p_i \log b_i}_{\leq -H(X)}$$

If the bet is **fair**, i.e.,

$$\sum \frac{1}{o_i} = 1,$$

Fairness

$$\sum p_i \log o_i b_i = \sum p_i \log o_i + \underbrace{\sum p_i \log b_i}_{\leq -H(X)}$$

If the bet is **fair**, i.e.,

$$\sum \frac{1}{o_i} = 1,$$

then $\sum p_i \log o_i \geq H(X)$ and is **minimal** if $o_i = \frac{1}{p_i}$.

But then,

Fairness

$$\sum p_i \log o_i b_i \;=\; \sum p_i \log o_i + \underbrace{\sum p_i \log b_i}_{\leq -H(X)}$$

If the bet is **fair**, i.e.,

$$\sum \frac{1}{o_i} \;=\; 1,$$

then $\sum p_i \log o_i \geq H(X)$ and is **minimal** if $o_i = \frac{1}{p_i}$.

But then, if both gambler and bookie play optimally,

$$\sum p_i \log o_i b_i = H(X) - H(X) = \smile$$

Fairness

$$\sum p_i \log o_i b_i \;=\; \sum p_i \log o_i + \underbrace{\sum p_i \log b_i}_{\leq -H(X)}$$

If the bet is **fair**, i.e.,

$$\sum \frac{1}{o_i} \;=\; 1,$$

then $\sum p_i \log o_i \geq H(X)$ and is **minimal** if $o_i = \frac{1}{p_i}$.

But then, if both gambler and bookie play optimally,

$$\sum p_i \log o_i b_i = H(X) - H(X) = \smiley$$

But, for $o_i = m$,

$$\sum p_i \log o_i b_i \;=\; \log m - H(X).$$

Entropy of English — Shannnon's experiment

Examples from Shannon's original paper and Lucky's book.

Claude Shannon, *A Mathematical Theory of Communication*, **1948**.

The symbols are independent and equiprobable.

XFOML RXKHRJFFJUJ ZLPWCFWKCYJ
FFJEYVKCQSGYD QPAAMKBZAACIBZLHJQD

The symbols are independent. Frequency of letters matches English text.

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI
ALHENHTTPA OOBTTVA NAH BRL

The frequency of pairs of letters matches English text.

ON IE ANTISOUTINYS ARE T INCTORE ST B S DEAMY
ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO
TIZIN ANDY TOBE SEACE CTISBE

Entropy of English — Shannon's experiment

The frequency of triplets of letters matches English text.

IN NO IST LAT WHEY CRATICT FROURE BERS GROCID PONDENOME OF DEMONSTURES OH THE REPTAGIN IS REOGACTIONA OF CRE

The frequency of quadruples of letters matches English text.

Each letter depends previous three letters.

THE GENERATED JOB PROVIDUAL BETTER TRAND THE DISPLAYED CODE, ABOVERY UPONDULTS WELL THE CODERST IN THESTICAL IT DO HOCK BOTHE MERG. (INSTATES CONS ERATION. NEVER ANY OF PUBLE AND TO THEORY. EVENTIAL CALLEGAND TO ELAST BENERATED IN WITH PIES AS IS WITH THE)

Entropy of English

The goal

For an English text $X_1 X_2 X_3 \ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \;\approx\; H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Entropy of English

The goal

For an English text $X_1X_2X_3\ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \approx H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s

Entropy of English

The goal

For an English text $X_1 X_2 X_3 \ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \;\approx\; H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s k

Entropy of English

The goal

For an English text $X_1X_2X_3\ldots\ldots$, estimate

$$H(X_{k+1} \mid X_kX_{k-1}\ldots X_1) \;\approx\; H(X_{k+1} \mid X_kX_{k-1}\ldots\ldots)$$

Guessing game

s k i

Entropy of English

The goal

For an English text $X_1X_2X_3\ldots\ldots$, estimate

$$H(X_{k+1} \mid X_kX_{k-1}\ldots X_1) \approx H(X_{k+1} \mid X_kX_{k-1}\ldots\ldots)$$

Guessing game

s k i t

Entropy of English

The goal

For an English text $X_1 X_2 X_3 \ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \approx H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s k i t o

Entropy of English

The goal

For an English text $X_1X_2X_3\ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \;\approx\; H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s k i t o u

Entropy of English

The goal

For an English text $X_1X_2X_3\ldots\ldots$, estimate

$$H(X_{k+1} \mid X_kX_{k-1}\ldots X_1) \approx H(X_{k+1} \mid X_kX_{k-1}\ldots\ldots)$$

Guessing game

s k i t o u r

Entropy of English

The goal

For an English text $X_1 X_2 X_3 \ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \;\approx\; H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s k i t o u r i

Entropy of English

The goal

For an English text $X_1 X_2 X_3 \ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \quad \approx \quad H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s k i t o u r i n

Entropy of English

The goal

For an English text $X_1 X_2 X_3 \ldots\ldots$, estimate

$$H(X_{k+1} \mid X_k X_{k-1} \ldots X_1) \;\approx\; H(X_{k+1} \mid X_k X_{k-1} \ldots\ldots)$$

Guessing game

s k i t o u r i n g

Estimate the average number of questions needed to correctly identify the next letter.

1.3 [1950]

Entropy of English

By **gambling**. Horses $\approx$ letters (27 including space).

Let $o_i = 27$, for $i = 1, \ldots, 27$.

$$S_n = 27^n \cdot \underbrace{b(X_1, \ldots, X_n)}_{\prod_{i=0}^{n-1} b(X_{i+1}|X_i \ldots X_1)}$$

Entropy of English

By **gambling**. Horses $\approx$ letters (27 including space).

Let $o_i = 27$, for $i = 1, \ldots, 27$.

$$S_n = 27^n \cdot \underbrace{b(X_1, \ldots, X_n)}_{\prod_{i=0}^{n-1} b(X_{i+1}|X_i \ldots X_1)}$$

For example

$$b(t) = \frac{1}{4}$$
$$b(h|t) = \frac{3}{4}$$
$$b(e|th) = \frac{7}{8}$$

$$b(the) = \frac{1}{4} \cdot \frac{3}{4} \cdot \frac{7}{8} = \frac{21}{128}$$

Entropy of English

$$
\begin{aligned}
\frac{1}{n}\mathbb{E}\log S_n(\vec{X}) &= \log 27 + \frac{1}{n}\mathbb{E}\log b(\vec{X}) \\
&= \log 27 + \frac{1}{n}\sum_{\vec{x}} p(\vec{x}) \cdot \log b(\vec{x}) \\
&\leq \log 27 + \frac{1}{n}\sum_{\vec{x}} p(\vec{x}) \cdot \log p(\vec{x}) \\
&= \log 27 - \frac{1}{n}\underbrace{H(X_1, \ldots X_n)}_{\textbf{H(English)}}
\end{aligned}
$$

If the player plays optimally, she approaches the correct value.

1.34 [1978]