Lesson 4
Entropy
4-3: Algorithmic Entropy: Kolmogorov Complexity

Jan Reimann

Math 574, Topics in Logic
Penn State, Spring 2014

Information Content of Strings

In the previous lectures, we defined an information measure ***I*** for random variables/probability distributions.

Then we defined entropy as expected information.

Q: Can we define the information content of an ***individual string*** $\sigma \in 2^{<\mathbb{N}}$?

Information Content of Strings

We could try: Put a uniform probability distribution $\boldsymbol{\lambda}$ on $\{0,1\}^n$, and define

$$I(\sigma) = -\log \lambda(\sigma).$$

$\lambda(\sigma) = 2^{-n}$

Problem: all strings have the same probability, hence the same information content.

But we would expect the string

$$0000000000\ldots 0000$$

to have low information content, while the

outcome of a coin toss

has high information content.

Information Content of Strings

Q: Can we find a probability distribution on strings such that

simple strings have high probability, $-\log P$

complex strings have low probability?

We will see later that this indeed possible.

Kolmogorov Complexity

Idea: low information content = high compressibility

Kolmogorov complexity makes this idea rigorous.

σ complex

M_0 long prg

M_1 "σ burned in"

Let M be a Turing machine, $\sigma \in 2^{<\mathbb{N}}$.

$$C_M(\sigma) = \min\{|p| : \underbrace{M(p)}_{\text{code}} = \underline{\sigma}\},$$

where we let $\min \emptyset = \infty$.

M-complexity = length of shortest *M*-description (code)

decoder

Problem: arbitrariness in the choice of *M*. Different machines can assign the same string drastically different complexities.

Solution: Use universal Turing machines.

Kolmogorov Complexity

We define a pairing function for strings as $\langle \sigma, \tau \rangle = 0^{|\sigma|}1\sigma\tau$.

We also identify natural numbers with their binary representation.

Note: $|m| = \log(m)$.

Recall a universal Turing machine U emulates all other TM's:

$$U(\langle e, \sigma \rangle) = M_e(\sigma).$$

Fix any universal TM U and define $C(\sigma) = C_U(\sigma)$.

Kolmogorov Complexity

Kolmogorov, Solomonoff

THM: [Invariance Theorem]
For any TM M there exists a constant c_M such that

$$\forall\sigma\ C(\sigma) \leqslant C_M(\sigma) + c_M.$$

$M = M_e$

- Proof: If p is a shortest M-program for σ, and e is the Gödel number of M, then $\langle e, p\rangle$ is a U-program for σ, and hence

c_M

$$C(\sigma) \leqslant |\langle e, p\rangle| = |0^{|e|}1ep| = 2\log(e) + |p| + 1 = C_M(\sigma) + 2\log(e) + 1.$$

$|e| + 1 + \log(e) + |p|$

" $\log(e)$

$U(\langle e, p\rangle)$

$= M(p)$

Notation: $C(\sigma) \leqslant^+ f(\sigma)$ *means: There exists c s.t.*

$$\forall\sigma\ C(\sigma) \leqslant f(\sigma) + c$$

$\leq f(\sigma) + O(1)$

.

Basic Properties of Kolmogorov Complexity

1. $C(\sigma) \leqslant^+ |\sigma|$.

   Consider the copy machine $M(p) = p$. We have $C_M(\sigma) \leqslant |\sigma|$ and hence by the invariance theorem $C(\sigma) \leqslant^+ |\sigma|$.

   incompressible

2. For any n, there exists a string σ of length n with $C(\sigma) \geqslant |\sigma| = n$.

   A simple counting argument: There are 2^n strings of length n, but only $1 + 2^1 + 2^2 + \cdots + 2^{n-1} = 2^n - 1$ programs of length $< n$.

3. If $h : 2^{<\mathbb{N}} \to 2^{<\mathbb{N}}$ is computable then $C(h(\sigma)) \leqslant^+ C(\sigma)$.

   Let M be a Turing machine given by $M(\sigma) = h(U(\sigma))$. Then by the invariance theorem,

   p p

   $$C(h(\sigma)) \leqslant^+ C_M(h(\sigma)) \leqslant C(\sigma).$$

   $U(p) = \sigma$

   $M(p) = h(\sigma)$

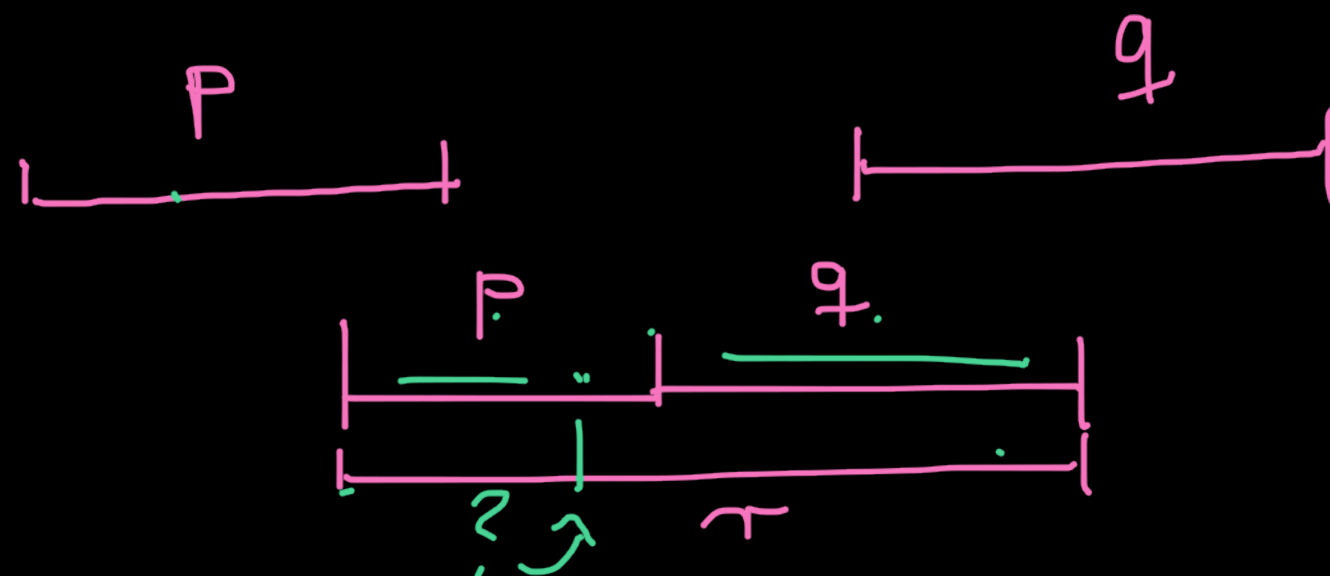
Subadditivity

How robust is C as an information measure?

- Do we have $C(\underline{\sigma, \tau}) \leqslant^{+} C(\sigma) + C(\tau)$?

$$H(X, Y)$$

$$C(\sigma, \tau) = C(\langle \sigma, \tau \rangle)$$



$$M(\tau) = \ldots$$

Failure of Subadditivity

THM: [Martin-Löf]
Suppose k is fixed. For any sufficiently long τ there exists $\sigma \sqsubset \tau$ such that $C(\sigma) < |\sigma| - k$.

Proof:

- Order all finite strings length-lexicographically, i.e.

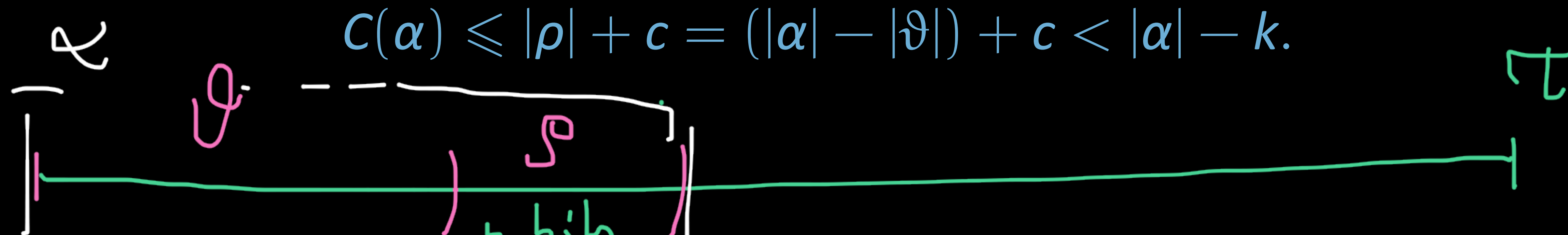
$$\langle\rangle < 0 < 1 < 00 < 01 < 10 < 11 < 000 < 001 < \ldots$$

  and let $n(\sigma)$ be the position of σ in this ordering.

  $\sigma = 00$
  $n(\sigma) = 3$

- Suppose $\vartheta \sqsubset \tau$. Let $n = n(\vartheta)$, and let ρ be the next n bits of τ.
- Put $\alpha = \vartheta \frown \rho$. Then $C(\alpha) \leqslant |\rho| + c$ for some constant c.
- If we choose $|\vartheta| > k + c$, then

$$C(\alpha) \leqslant |\rho| + c = (|\alpha| - |\vartheta|) + c < |\alpha| - k.$$



Failure of Subadditivity

COR: For any d there exists $\tau = \vartheta ^\frown \sigma$ such that

$$C(\tau) = C(\vartheta ^\frown \sigma) > C(\vartheta) + C(\sigma) + d.$$

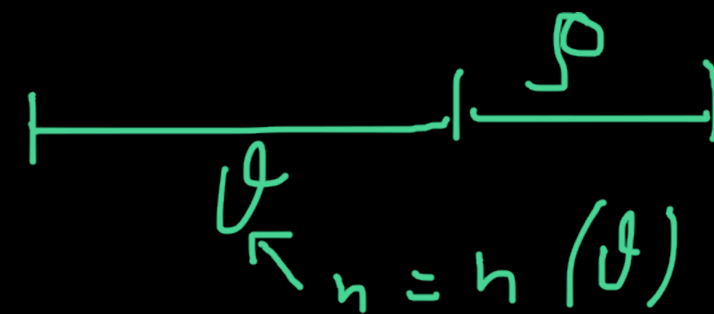
$C(\vartheta, \sigma) \gtrsim C(\vartheta ^\frown \sigma)$

Proof:

- Pick c such that $C(\alpha) \leqslant |\alpha| + c$ for all α.
- Choose a sufficiently long string τ with $C(\tau) \geqslant |\tau|$ and $C(\vartheta) < |\vartheta| - (c + d)$ for some $\vartheta \sqsubset \tau$ (by THM).
- Let σ be such that $\tau = \vartheta ^\frown \sigma$.
- Then

$$C(\vartheta) + C(\sigma) < |\vartheta| - (c + d) + |\sigma| + c = |\tau| - d \leqslant C(\tau) - d.$$

Failure of Subadditivity



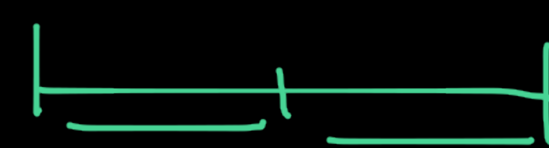
What went wrong?

We exploited that fact that a string not only provides information through its bits, but also through its length.

This fact is not captured by *C*.

Question: Can we alter the definition of complexity to take this into account?

⟶ *Prefix-free complexity*



K