Kolmogorov Complexity 101

BY

陆潇扬

2023年5月9日

1 动机

我们在课堂上学习了对于一个特定随机变量的信息的概念, 即熵. 但是根据我们的常识, 框定一些对象 (比如自然数, 字符串), 其中有一些元素比其他元素显然规则得多, 因而包含更多的信息. 比如 2^{100} 明显比 27382978237492 更加规则. 用我们的知识能解释这种现象吗? 我们要把这个集合做成一个概率空间 (有时这是不可能的), 还是说把对象看成一个随机过程生成的? 一个更自然的想法是, 考虑能输出这个对象的最短程序.

注: 对于文中图灵机等等概念的定义不再赘述. 以及, 文中的 big-O notation $O(f(x))$ 表示绝对值在 $|c\,f(x)|$ 以内的一项, 这和算法的时间复杂度的含义有所不同.

2 基本定理: C 的存在

我们先来试着形式化地定义"最短程序".

定义 1. *对于一个对象的集合 S, 我们给每一个对象编号, 这个单射记为 $n: S \to \mathbb{N}$. 那么对于一个特定的"计算方法"函数 $f: \mathbb{N} \nrightarrow S$ ($\mathbb{N} \to S \cup \{\perp\}$) 而言, 生成 $x \in S$ 的最短程序的长度就是 $C_f(x) = \min_{f(p)=n(x)} l(p)$, 这里 $l(p)$ 是 p 在二进制下的长度. 如果不存在这样的 p, 那么定义 $C_f(x) = +\infty$.*

我们在第一节中设想的通用定义最终肯定不能单单对一个机器而言, 如果考虑程序的精确长度, 我们永远找不到一个最好的"计算方法", 因为每个正整数都有一个"计算方法"可以在空输入下生成它. 因此我们需要允许常数大小的偏差, 也就是说

定义 2. *对于两个"计算方法"函数, 如果*

$$\forall s \in S, |C_f(x) - C_g(x)| \leqslant c,$$

这里 c 是和 x 无关 (但可以和 f, g 有关的常数), 那么称 f 和 g 是等价的.

但是这样我们依然不能得到一个良定义. 如果我们试图在所有 partial function 中找到最好的函数 (在上面等价的意义下), 这是行不通的. 一个论证如下: 令 $S = \mathbb{N}$, 假设我们有一个最优函数 f. 那么由于一定长度的正整数只有有限个, 我们一定能找到一列整数, 使得 $C_f(n_i) \geqslant i$. 我们构造一个新的函数 g, 使得 $g(i) = n_i$. 那么, 对于生成 n_i 的程序, g 只需要 $\log i$ 左右长度的程序, f 却需要 i, 这个差距无法被任何整数 bound 住.

好消息是, 如果我们把允许的计算方法限制在**可计算**的 partial function. 我们真的能找到最优. 这又用到了可计算理论中的常见技巧: 解释器. 为了方便, 我们考虑这样一个更特殊的 universal machine: 对于一个图灵机的枚举 $T_1, \ldots, T_n$, 对应于函数 $\phi_1, \ldots, \phi_n$, 这个 universal machine $\phi_0(\langle n, p \rangle) = \phi_n(p)$, 这里 $\langle n, p \rangle = \underbrace{1 \ldots 1}_{n个1} 0np$. 很清楚, 这样的 T_0 是存在的, 因为我们可以靠 1 的个数来区分 n 和 p, 然后去解释 T_n 在 p 输入上的执行. 这个函数是满足最优性质的: $|C_{\phi_0}(x) - C_{\phi_n}(x)| \leqslant l(n) + 1$. 我们定义 $C(x) = C_{\phi_0}(x)$.

但是这和我们期望的完全是对象的内禀性质的复杂性还有距离, 因为这仍然依赖于图灵机的枚举. 但是可以证明, 对于两个有效枚举定义的 C 和 C', 他们的差距也在一个常数以内. 因此这就是我们想要的定义.

... 真的吗? 对于一些应用而言, 另一个只对于前缀图灵机的定义 K 会更加合适. 比如 $K(x,y) := K(\langle x,y\rangle) \leqslant K(x)+K(y)$ (常数偏差意义下), 和熵类似, 但这对 C 是不成立的. 限于篇幅, 这里不介绍 K.

3 简单应用

我们先把 C 推广到条件的情况.

定义 3. 在类似定义 1的条件下, $C_f(x|y) := \min_{f(\langle y,p\rangle)=n(x)} l(p)$.

通过和上一节类似的手法, 也可以证明存在一个最优的机器 $C(x|y)$. 我们还定义

$$C(x,y) := C(\langle x,y\rangle),$$

即得到 x 和 y 的最短程序. 我们通过一些简单的命题看看怎么运用 C 的定义.

命题 4. $C(x) \leqslant l(x)+c, C(x|y) \leqslant C(x)+c$, 其中 c 和 x,y 无关.

证明. 对于第一个命题, 定义一个将输入复制到输出的图灵机 T. 根据定义, ϕ 比这个图灵机优, 因此 $C(x) \leqslant C_T(x)+c = l(x)+c$. 对于后者, 定义一个图灵机 U, 并且 $T(\langle x,y\rangle) = \phi(x)$. 根据定义, $C(x|y) \leqslant C_U(x|y)+c = C(x)+c$. □

4 不可压缩性

简单的计数就可以发现, 有很多字符串是没法用更短的程序来编码的, 因为短程序的个数是有限的. 具体来说,

定理 5. 对于一个元素个数为 m 的集合和任意 y, 其中至少有 $m(1-2^{-c})+1$ 个元素满足 $C(x|y) \geqslant \log m - c$.

证明. 比 $\log m - c$ 要短的程序只有

$$\sum_{i=0}^{\log m - c - 1} 2^i = 2^{\log m - c} - 1 = m\,2^{-c} - 1$$

个. □

这个简单的结论有许多有趣的推论. 下面是之前提到过的 C 与熵的一个不同之处:

命题 6. 对于任意 n, 都有字符串 x,y, 且 $C(x,y) \geqslant C(x)+C(y)+\log n+O(1)$.

证明. 长度之和为 n 的字符串 x,y 共有 $(n+1)\,2^n$ 组. 根据定理 5, 存在这样的 x,y 使得 $C(x,y) \geqslant n+\log n-1$. 但是根据命题 4, $C(x)+C(y) \leqslant l(x)+l(y)+c = n+c$. 因此 $C(x,y) \geqslant C(x)+C(y)+\log n+O(1)$. □

部分的复杂度可以大于整体. 比如, 通过构造特定图灵机的方法可以证明 $C(1^{2^k}) \leqslant \log k + O(1)$ (这个图灵机将输入看作整数 k, 然后写下 2^k 个 1). 但是根据定理 5, 考虑所有长度在 2^k 以内的只包含 1 的字符串的集合, 这个集合中一定有元素满足 $C(x) \geqslant k + O(1)$. 当 k 足够大时, 就会出现部分的复杂度大于整体的情况.

上述现象的原因之一是信息隐藏在了字符串的长度之中, 因此我们可能会想用 $C(x|l(x))$ 来避免这种情况. 但是这依然是行不通的. 考虑形如 $x = n0^{n-l(n)}$ 的字符串, 它的长度是 $l(n)$, 并且 n 是它的前缀. 我们通过构造图灵机可以得出 $C(x|n) = O(1)$. 但是, 假如我们挑出了一个满足 $C(n) \geqslant l(n)$ 的 n (这是可行的), 那么 $C(n|l(n)) \geqslant C(n) - 2\,l(l(n)) + O(1) \geqslant \log n - 2\log\log n + O(1)$, 而 $C(x|l(x)) = O(1)$, 仍然出现了部分的复杂度大于整体的现象.[1]

5 信息论

这一节往后, 可能会忽略等式和不等式中的 $O(1)$.

在多大的程度下, 我们可以把 $C(x)$ 看作 x 包含的信息, 并得出一系列跟熵和互信息类似的结论?

5.1 C 和 H

某种意义下, "H 是 C 的期望".

定理 7. *考虑形如这样的字符串 $x = y_1 y_2 \dots y_m$, 其中 y_i 的长度均为 n. 这对应于一个概率分布 p, 其中 $p_j, 0 \leqslant j < 2^n$ 是 j 出现的频率.*

$$\frac{C(x)}{m} \leqslant H(p) + \varepsilon,$$

其中 $\varepsilon = 2^{n+1}\frac{l(m)}{m} = o(m)$.

证明. 想要得到 x, 我们只要知道每个数字 i 出现的次数 $k_i = p_i m$, 和这些数字的排列. 后者可以简单地用一个 $\left[0, \binom{m}{k_0, \dots, k_{2^n-1}}\right]$ 范围内的索引 t 表示. 因此

$$C(x) \leqslant C(t) + \sum_{i=0}^{2^n-1} 2\,l(k_i) \leqslant 2^{n+1}\,l(m) + l\left(\binom{m}{k_0, \dots, k_{2^n-1}}\right).$$

为了估计第二项,

$$\begin{aligned}
\log\binom{m}{k_0, \dots, k_{2^n-1}} &= \log\frac{m!}{k_0! \cdots k_{2^n-1}!} \quad (\text{因为 } \sum_{i=0}^{2^n-1} k_i = m)\\
&\sim O(1) + \log\frac{\sqrt{m}\,\frac{m^m}{e^m}}{\prod_{i=0}^{2^n-1}\sqrt{k_i}\,\frac{k_i^{k_i}}{e^{k_i}}} \quad (\text{这里不是很严谨, 因为不一定所有 } k \text{ 都在增大})\\
&= O(1) + \frac{1}{2}\log m + m\log m - m - \sum_{i=0}^{2^n-1}\left(\frac{1}{2}\log k_i + k_i\log k_i - k_i\right)\\
&= O(1) + \sum_{i=0}^{2^n-1} -k_i\log\frac{k_i}{m}\\
&= O(1) + m\,H(p).
\end{aligned}$$

1. $C(n) \leqslant C(n|x) + 2\,l(x)$ 是因为可以构造图灵机, 对于输入 $\underbrace{1\dots1}_{l(x)\text{个}1}0xp$, 先分离出 x 和 p, 然后运行 $\phi_0(\langle x, p\rangle)$ (类似我们证明基本定理时用的编码方式).

□

通过证明过程我们发现, $H(p)$ 对应于概率分布, 而 ε 对应于具体的取值.

后话: 前缀复杂度 $K(x)$ 因为是前缀码, 可以看作对象的一个编码. 对任意的 (可计算的) 概率分布 p, 这个编码满足

$$\sum_x p(x)\,K(x) \leqslant \sum_x -p(x)\log p(x) + c_p,$$

其中 c_p 只和 p 有关. 再根据 C 和 K 的关系, 我们可以得到 $C(x)$ 也同样满足这个不等式. 这比上面的定理更有力地说明了 C 和 H 内在的关系.

5.2 I_C

我们当然可以定义 $I_C(x;y) = C(y) - C(y|x)$. 这满足 $I_C(x;x) = C(x)$ 和 $I_C(x;y) \geqslant 0$, 但是可惜的是并不满足 $I_C(x;y) = I_C(y;x)$. 比如, 根据定理 5, 对于任意正整数 n, 都存在一个长度为 n 的字符串 x 满足 $C(x|n) \geqslant n$. 我们进一步选择满足 $C(n) \geqslant l(n)$ 的 n. 那么

$$\begin{aligned} I_C(x;n) &= C(n) - C(n|x)\\ &= C(n)\\ &\geqslant l(n),\\ I_c(n;x) &= C(x) - C(x;n)\\ &\leqslant n - n\\ &= 0. \end{aligned}$$

因此 $I_C(x;y)$ 和 $I_C(y;x)$ 之间的差距至少是 log 级别的. 我们接下来证明这个界是紧的.

命题 8. $\forall x, y \in \mathbb{N}, C(x,y) = C(x) + C(y|x) + O(\log C(x,y))$.

证明.

- **$\leqslant$.** 构造图灵机, 采用这样的输入: $\underbrace{1\ldots1}_{l(p)}0pq$, 其中 $\phi_0(p) = x, \phi_0(\langle x, q\rangle) = y$. 所以

  $$C(x,y) \leqslant C(x) + C(y|x) + 2\,l(C(x)) \leqslant C(x) + C(y|x) + 2\,l(C(x,y)).$$

  ($C(x) \leqslant C(x,y)$ 可以简单地构造图灵机解决, 这里不赘述.)

- **$\geqslant$.** 即 $\exists c > 0, C(x,y) \geqslant C(x) + C(y|x) - c\log C(x,y)$. 采用反证法, 假设对任意大的 c, c' 都存在 x, y, 使得

  $$C(y|x) > C(x,y) - C(x) + c\log C(x,y) \geqslant C(x,y) - C(x) + c'\,l(C(x,y)).$$

  考虑这样的集合: $A = \{\langle z, w\rangle \mid C(z,w) \leqslant C(x,y)\}$. 这个集合是可以枚举的 (枚举程序并"并行"地运行即可). $A_x = \{z \mid C(x,z) \leqslant C(x,y)\}$ 同样也是可以枚举的. 所以可以构造图灵机, 给定 $C(x,y), x$ 和 y 在 A_x 的枚举中的编号, 计算出 y. 所以

  $$C(y|x) \leqslant l(|A_x|) + 2\,l(C(x,y)) + O(1).$$

  结合假设, 我们有

  $$|A_x| > 2^{C(x,y) - C(x) + (c'-2)\,l(C(x,y)) - O(1)} := 2^t.$$

现在我们来导出矛盾. 给定 $C(x,y)$ 和 t, 我们可以试图从满足 $2^t < |A_u| = |\{w \mid C(z,w) \leqslant C(x,y)\}|$ 的 z 中找出 x. 令 Z 是这样的 z 的集合, 那么 $\{\langle z,w\rangle \mid z \in Z, w \in A_z\} \subseteq A$, 即

$$|A| \geqslant \sum_z |A_z| > |Z| \cdot 2^t.$$

另一方面, 一个程序只有一个输出, 故 $|A| \leqslant 2^{C(x,y)+O(1)}$. 因此 $|Z| < 2^{C(x,y)+O(1)-t}$. 在知道 $C(x,y), t$ 的情况下, 我们可以枚举$|Z|$, 并用 x 在枚举中的序号来计算 x ($\underbrace{1\ldots1}_{l(C(x,y))}\ 0\underbrace{1\ldots1}_{l(t)}0C(x,y)ti$), 故

$$\begin{array}{rcl} C(x) & < & 2\,l\,C(x,y) + 2\,l(t) + l(i) \\ & \leqslant & 2\,l\,C(x,y) + 2\,l(t) + C(x,y) - t + O(1) \\ & = & C(x) + O(1) + (4-c)\,l(C(x,y)) \end{array}$$

当 c 足够大时, $C(x) < C(x)$, 矛盾! □

推论 9. $|I_C(x;y) - I_C(y;x)| = O(\log C(x,y))$.

证明. $C(x,y) = C(x) + C(y|x) + O(\log C(x,y)) = C(y) + C(x|y) + O(\log C(x,y)) = C(y,x)$. 因此

$$C(x) - C(x|y) = C(y) - C(y|x) + O(\log C(x,y)).$$ □

作为后话, $I_K(x;y) = K(y) - K(y|x)$ 是对称的.

6 总结

这只是 Kolmogorov Complexity 最基本的一瞥, 还有许多角度和应用等待探索.

Reference

1. *An Introduction to Kolmogorov Complexity and It's Applications*. Ming Li, Paul Vitányi, 2019.

2. *Elements of Information Theory*. Thomas M. Cover, Joy A. Thomas, 2006.