Algorithmic information
Conversation

Péter Gács

Computer Science Department
Boston University

Spring 2021

Randomness

- Somebody shows you a sequence of 100 0's and 1's, and claims he obtained it by tossing a fair coin 100 times. If the sequence looks like

  $$0101010101\ldots0101010101$$

  then you will conclude that he is lying. But:
  What will you base your accusation on?
  This outcome is exactly as probable as any other sequence.
- This question is old, predating the formal foundation of probability by more than 100 years. We arrived at a satisfactory understanding only after the introduction, mainly by Andrei Kolmogorov, of a quantitative measure of disorder and information, description complexity.
- I start with a detour via the notion of information.

Information

- Information is many things for many people. Since Shannon, its main technical meaning is given by the needs of communication technology.
  We want to measure the quantity, leave the content aside. (The content will steal itself back, though.) By Shannon's huge insight, the form is immaterial, since it can be changed by coding. A file storing a picture, or an audio file both consist of 0's and 1's—bits.
- Since a k-bit string can represent 2^k different things, our first definition: information in a variable X is

$$\log_2 |A|$$

  where A is the range of values of the variable X. So, information measures the amount of uncertainty in X.

Entropy

- How much information is in a string of English text of 1000 characters (leaving aside meaning and context)? If I see the part “informa” in a text then I can complete it to “information” myself, most of the time. I will be better off just putting there a sign “…” saying “complete it as usual”. This is the idea behind the compression of material in computers.
- One way to do this is to assign probabilities $p(x_1), p(x_2), \dots, p(x_n)$ to each possible value x_i of the variable X, and say that the amount of surprise in value x is

  $$\log_2 \frac{1}{p(x)}.$$

  When all values are equally probable we get back the old definition.

- The uncertainty in the variable X is then the average amount of surprise:

$$p(x_1)\log\frac{1}{p(x_1)}+\cdots+p(x_n)\log\frac{1}{p(x_n)}.$$

  This is called the entropy of the variable X and is denoted by $H(X)$.
  Shannon showed that, when large, entropy measures reasonably well the average number of bits needed to encode X.
- DVDs are possible only since the amount of entropy in the music and pictures is much less than what you would compute if you just added up the number of pixels (say a million per screenful), multiplied by (say) 30 per second.

Mutual information

The string “pet dog” can also be seen as two strings put together:

$$X = \text{"pet"}, Y = \text{"dog"}.$$

Suprise $\log \frac{1}{p(xy)}$ in seeing a value $XY =$“pet dog” is more than the surprise in “pet” alone, but typically less than the sum of the surprises in X and Y separately (expecting “dog” more than “rock”). The following algebra is interesting:

$$p(xy) = p(x) \cdot p(xy)/p(x),$$
$$\log \frac{1}{p(xy)} = \log \frac{1}{p(x)} + \log \frac{1}{p(xy)/p(x)}.$$

Taking the average:

$$H(XY) = H(X) + H(Y|X),$$

where $H(Y|X)$ is the average remaining surprise in Y after knowing X.

$$I(X:Y) = H(Y) - H(Y|X)$$

is the decrease of uncertainty in Y after X is known: the information that X has about Y. Simple algebra shows

$$I(X:Y) = I(Y:X),$$

so this quantity is also called mutual information. The symmetry is not intuitively obvious—despite the simple algebra. We will return to it.

How to compress in practice

- If X, Y are consecutive frames of a movie, then knowing this, we can use less encoding bits for Y than for X. Probably most objects in the scene are the same, only some parts moved, mostly in one direction and the same speed.
- Similarly, already in the 1930's the telephone company could send hundreds of conversation over a single wire, exploiting that human speech can be greatly compressed.
- In both cases, deeper understanding is involved, about the way the picture or the human speech was created. Telephone researchers developed a sophisticated model of how human voice is formed in the throat and mouth.
- Meaning comes back: understanding the intentions of a speaker narrows the possibilities of completing a sentence that he begun.

- Kolmogorov's first insight was that randomness and information content amount to the same thing, and can be expressed via the notion of incompressibility. The string $0101\dots01$ (1000 times) can be expressed simpler than writing 2000 symbols: we can just say: "write 1000 times 01"!
  This becomes a mathematical concept only after we formalize how to interpret a sentence like "write 1000 times 01".
- Computer science gave a natural candidate for this. Let us fix an interpreter: a computing machine T with binary strings as inputs.

$$C_T(x) = \min_{T(p)=x} |p|.$$

Computers

- It is essential to this mathematical lecture that saying "computer", I have a particular, formal mathematical notion in mind. Still, for now it is sufficient to have an somewhat informal idea; on the level necessary for us now, all computers are similar, and all of us are familiar with some.
- In one important aspect the "computer" used here differs from your laptop: no limitations of storage capacity or computing time. Imagine your laptop connected to an outside disk; whenever that disk fills up during a computation, the machine can ask to switch in another outside disk, and so on to disks $1, 2, 3, \ldots$. It can also ask to switch back to the previous disk.
- "No time limit" means even that the computation may run forever! This is not useful in itself, but we must allow all possible programs, including some about which we don't know whether the computation they start will terminate.

Our computers are universal: For any two computers T, U, you can give a finite program s_{TU} to T, telling it that it should behave as U, "simulate" it with respect to its inputs and outputs.

- Do we get as many different notions of complexity as there are computing machines T? Kolmogorov's other great insight was that the definition does not depend much on the machine T. (Solomonoff, who invented the notion slightly earlier for another purpose, prediction, also proved this).
  Take another (universal) machine U, then

  $$|C_T(x) - C_U(x)| \leq c_{TU}$$

  for some constant c_{TU}.
- Indeed, for example there is a string s_{TU} instructing machine T to simulate machine U.
  s_{TU} can be quite long, but constant; so asymptotically it does not change much the complexity of long strings x.
- So we drop the subscript and consider $C(x)$ an inherent property of the string x, its description complexity. (Complexity of other kinds of objects can be obtained by first encoding them into strings.)

- $C(x)$ seems a good measure of information content. Also, it belongs to the individual string, does not depend on the size of some range to which x belongs, or on any probability distribution.
- It is quantitatively strongly related to the other two definitions (size of range, probabilities), but I will not cover these relations today.

Examples

- If $|x| = n$ then $C(x) \overset{+}{<} n$.
  Here $a(n) \overset{+}{<} b(n)$ means that there is a constant c such that for all n we have $a(n) \le b(n)$.
  Indeed, our description program can just say "output x".
- For $x = (01)^{n/2}$, we have $C(x) \overset{+}{<} \log_2 n$. Indeed, we can tell our machine in a constant-size program r to write repetitions of 01 $n/2$ times. The program looks like

$$r\beta(n),$$

  where $\beta(n)$ is the binary representation of n; its length is $\log_2 n$.
- So $(01)^{n/2}$ can be compressed to a much shorter string $r\beta(n)$: its description program.
  Note that the interpreting machine T performs the decompression, not the compression.

Most strings have near maximal complexity

- How many strings of length n can have complexity $< n-k$? There are fewer than 2^{n-k} programs of length $< n-k$, so the number of strings whose shortest description has length $< n-k$ is at most

$$2^{n-k} = 2^n/2^k.$$

  This shows that for example the number of strings of length n with complexity $< n-10$ is at most $2^n/2^{10} < 2^n/1000$. So at most every 1000th string can be compressed by 10 bits.
- This insight gives a strong motivation to identify randomness with incompressibility.

Uncomputability

- Description complexity seems a good measure of randomness and information. Too bad, it is not directly useful since it is not computable. This fact is also interesting as it has the simplest proof of uncomputability I know of.
- Here is an old paradox:
  There are some numbers that can be defined with a few words: say, “the first number that begins with 100 9’s”, etc. Clearly, there are only finitely many numbers definable with a sentence shorter than 100 characters. Therefore, there is a first number that cannot be defined by a sentence shorter than 100.
  But—I have just defined it!

- The paradox turns on the insight that the notion of “define” is undefined.
  Let “p defines x” mean for us now $T(p) = x$ using our universal interpreter machine T.
- Assume that the function $C(x)$ is computable; we will arrive at a contradiction.
  So, assume that there is an algorithm that on input x, computes the number $C(x)$. Then there is also an algorithm Q that on input k, outputs the first string $f(k)$ such that $C(x) > k$.

- On our machine T, let q be the length of a program for the above algorithm Q. For some number k, we can write now some program $r(k)$ for T that outputs $f(k)$.
- How long is $r(k)$?
  - q bits for the program of Q,
  - $\log_2 k$ bits for the number k,
  - a constant number p of more bits to tell machine T what to do with this information: this is

$$p + q + \log_2 k.$$

  Since p, q are constant, if k is large then this is less than k.
- We arrived at a contradiction: we output $f(k)$ with a program shorter than k, but $f(k)$ is the first string with $C(x) > k$.

Additivity

The complexity function $C(x)$ behaves in a number of ways similarly to the entropy function $H(X)$ of information theory. The conditional complexity $C(y \mid x)$ of y with respect to x is

$$\min_{T(p,x)=y} |p|.$$

This is conceptually very different from how we defined conditional entropy $H(Y \mid X)$. Still, a nontrivial theorem by Kolmogorov and Levin says

$$C(x) + C(y \mid x) \overset{+}{=} C(x, y) \pm \log_2 C(x, y).$$

Similar to the equation $H(X) + H(Y \mid X) = H(X, Y)$.

- Just as for information theory, we have now

$$I(x : y) \approx I(y : x)$$

  where now $\approx$ hides a logarithmic correction term.
- This symmetry of information relation is counterintuitive: let $x = (p, q)$, where p, q are two prime numbers of 500 digits, and $y = p \cdot q$. It is easy to compute y from x, but is hopeless (at the present state of science) to compute x from y: the known algorithms take exponential time.
- So (in some cases, apparently)
  y noes not help as much computing x as x helps computing y.

I finish with two old quotes:

- *Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.*

  (Pascal)

  Pascal knew that compression can be time-consuming.

- *Thus, at the game of "head or tail", hundred consecutive heads appear to us extraordinary; because dividing the almost infinitely many combinations that may occur in hundred tosses into regular ones, that is those exhibiting some easily discoverable order, and into irregular sequences; the latter are incomparably more numerous.*

  (Laplace)

  In the coin-tossing paradox, Laplace noticed the connection of randomness with irregularity, and intuited (not proved) that the number of all "regular" strings is small. But he refers to "easily noticeable" regularities, while Kolmogorov's notion, focusing on description length, is not. It was this latter notion leading to also a proof.