Information and Computation: Shannon Entropy and Kolmogorov Complexity

Satyadev Nandakumar
Department of Computer Science.
IIT Kanpur

October 19, 2016

Shannon Entropy

Definition

Let X be a random variable taking finitely many values, and P be its probability distribution. The *Shannon Entropy* of X is

$$H(X) = \sum_{i \in X} p(i) \log_2 \frac{1}{p(i)}.$$

Shannon Entropy

Definition

Let X be a random variable taking finitely many values, and P be its probability distribution. The *Shannon Entropy* of X is

$$H(X) = \sum_{i \in X} p(i) \log_2 \frac{1}{p(i)}.$$

This measures the average uncertainty of X in terms of the number of bits.

The Triad

Figure: A. N. Kolmogorov

Figure: Claude Shannon

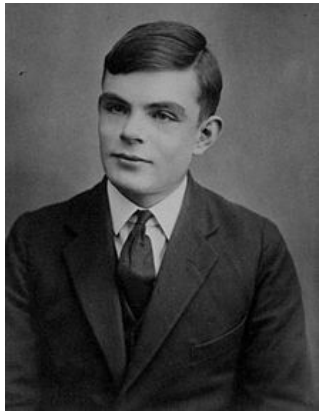

Figure: Alan Turing

Just Electrical Engineering

"Shannon's contribution to *pure mathematics* was denied immediate recognition. I can recall now that even at the International Mathematical Congress, Amsterdam, 1954, my American colleagues in probability seemed rather doubtful about my *allegedly exaggerated interest* in Shannon's work, as they believed it consisted more of techniques than of mathematics itself.

. . . However, Shannon did not provide rigorous mathematical justification of the complicated cases and left it all to his followers. Still his mathematical intuition is amazingly correct."

A. N. Kolmogorov, as quoted in [Shi89].

Kolmogorov and Entropy

Kolmogorov's later work was fundamentally influenced by Shannon's.

1. Foundations: Kolmogorov Complexity - using the theory of algorithms to give a combinatorial interpretation of Shannon Entropy.
2. Analogy: Kolmogorov-Sinai Entropy, *the* only finitely-observable isomorphism-invariant property of dynamical systems.

Three approaches to the definition of entropy

1. Combinatorial
2. Probabilistic
3. Algorithmic

Combinatorial Approach - Ralph Hartley, 1928

To represent an element in a set with N objects, you need $\log_2 N$ bits. So the information content is the logarithm of the size of the population.

Combinatorial Approach - Ralph Hartley, 1928

To represent an element in a set with N objects, you need $\log_2 N$ bits. So the information content is the logarithm of the size of the population.

This leads to a derivation of Shannon Entropy via multinomial coefficients:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \approx 2^{-n\left[\frac{k}{n}\log\frac{k}{n}+\frac{n-k}{n}\log\frac{n-k}{n}\right]}, \tag{1}$$

via Stirling's approximation, for large k and n.

Kolmogorov Complexity - Motivation

Which of the following data looks "more random"? Why?

1. 000000000000000000000000000000000000000000
2. 101111001101011000001011000111100011101O

Kolmogorov Complexity - Motivation

Which of the following data looks "more random"? Why?

1. 00000000000000000000000000000000000000
2. 10111100110101100000101100011110001110 10

The second does not seem to have recognizable patterns.

Shannon: A brief interlude

"A Mind-Reading Machine", by Shannon plays a game of "matching pennies" with a human player, remembers the pattern of play of the player, and uses it to match the player's strategy.

Kolmogorov Complexity- Definition 1

Let U be a universal Turing Machine.

Definition

The (plain) Kolmogorov complexity of a string x is

$$C(x) = \min\{\text{length}(p) \mid U(p) \text{ outputs } x\}.$$

The (plain) conditional Kolmogorov complexity of x given y is

$$C(x) = \min\{\text{length}(\pi) \mid \pi(y) = x\}.$$

Some Examples

The string x=00000000000000000000000000000000 can be compressed as "a string of 32 zeroes".

Some Examples

The string x=00000000000000000000000000000000 can be compressed as "a string of 32 zeroes". 32 itself can be compressed in binary as 1 followed by 5 zeroes. So $C(x) \leq 3$ (approximately!).

Some Examples

The string x=00000000000000000000000000000000 can be compressed as "a string of 32 zeroes". 32 itself can be compressed in binary as 1 followed by 5 zeroes. So $C(x) \leq 3$ (approximately!).

But a string $y = 10111100110101100000101100011110001110 10$ produced by a random coin toss cannot have any shorter description than

print 10111100110101100000101100011110001110 10.

So $C(y) \approx |y|$.

Incompressible strings

Definition

A string x is *incompressible* if $C(x) > |x|$.

Lemma

Most strings are incompressible.

Kolmogorov Complexity is Uncomputable

Theorem

C is uncomputable.

The most popular proof involves Berry's Paradox : "the smallest number that cannot be expressed in less than 20 words".

Universal Compressors

" Also in those years I began my interest in universal source coding. Shmuel Winograd (who was visiting Israel from IBM) and I ran a seminar on Kolmogorov complexity. I found this concept very unsatisfying. The presence of the large constant in the complexity measure makes it impossible to calculate the Kolmogorov complexity for a specific non-trivial sequence. It was the search for a better complexity measure that began my work on universal data compression."

- Jacob Ziv, "A Conversation with Jacob Ziv" 1997

Some Information-Theoretic Inequalities

Theorem

(Shannon Inequality)

$$H(X \mid Y) \leq H(X)$$

Proof is by the convexity of $x \log x$ for $x \in [0, 1]$, setting $0 \log 0 = 0$.

Some Information-Theoretic Inequalities - II

Theorem

$C(x \mid y) \leq C(x) + O(1)$.

Proof.

Let ξ be a shortest 0 argument program which outputs x.

Some Information-Theoretic Inequalities - II

Theorem

$C(x \mid y) \leq C(x) + O(1)$.

Proof.

Let ξ be a shortest 0 argument program which outputs x.

Construct a 1-argument program π

1. Input w // *ignore input*
2. Output $U(\xi)$.

Some Information-Theoretic Inequalities - II

Theorem

$C(x \mid y) \leq C(x) + O(1)$.

Proof.

Let ξ be a shortest 0 argument program which outputs x.

Construct a 1-argument program π

1. Input w // *ignore input*

2. Output $U(\xi)$.

Then $|\pi| \leq |\xi| + O(1)$. □

Some Information Theoretic Inequalities - III

Theorem

(Data Processing Inequality) If $X \to Y \to Z$ forms a Markov Chain, then

$$I(X;Y) \geq I(X;Z).$$

Corollary

$$I(X;Y) \geq I(X;g(Y))$$

Some Information Theoretic Inequalities IV

Theorem

Let x be an arbitrary string, and f be a total computable function on strings. Then $C(f(x)) \leq C(x) + O(1)$.

Proof.

Let ξ be the shortest program for x, and ϕ be the shortest program to compute f. Then consider the program $\phi \circ \xi$. Its length proves the inequality. $\square$

The Goal

Clearly, $C(x)$ is a notion of information content.

The Goal

Clearly, $C(x)$ is a notion of information content.

We want to claim that $C(x)$ is approximately the same as $H(x)$, when x is viewed as a binary random vector. (What is the distribution of x?)

Subadditivity and Chain Rule

For any two random variables X and Y

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

For any two strings x and y

$$C(x, y) \leq C(x \mid y) + C(y) + 2 \log C(y) + O(1).$$

Some Curious Properties

1. C is non-monotone in the length of the string.
2. C is not additive:

$$C(x, y) \not\leq C(x) + C(y \mid x) + O(1).$$

The last property ends our hope of treating C as H!

Take Two: Prefix-free Kolmogorov Complexity

Let $\pi_0, \pi_1, \ldots,$ form a prefix-free encoding $\mathcal{P}$ of Turing machines.

Definition

The prefix-free complexity of a string x given a string y is

$$K(x \mid y) = \min\{|\pi_i| \mid \pi_i \text{ outputs } x\}.$$

Kraft's inequality and Universal Semimeasure

Since $\mathcal{P}$ is a prefix-free set, it obeys *Kraft's inequality*:

$$\sum_{n \in \mathbb{N}} \frac{1}{|\pi_i|} < 1.$$

Kraft's inequality and Universal Semimeasure

Since $\mathcal{P}$ is a prefix-free set, it obeys *Kraft's inequality*:

$$\sum_{n\in\mathbb{N}} \frac{1}{|\pi_i|} < 1.$$

Hence

$$m(x) = \sum_{n\in\mathbb{N}, R(\pi_n)=x} \frac{1}{2^{|\pi_i|}}$$

can be viewed as a (semi)measure on the set of strings (almost a probability measure on strings.)

Kraft's inequality and Universal Semimeasure

Since $\mathcal{P}$ is a prefix-free set, it obeys *Kraft's inequality*:

$$\sum_{n \in \mathbb{N}} \frac{1}{|\pi_i|} < 1.$$

Hence

$$m(x) = \sum_{n \in \mathbb{N}, R(\pi_n) = x} \frac{1}{2^{|\pi_i|}}$$

can be viewed as a (semi)measure on the set of strings (almost a probability measure on strings.)

This is called the *universal semimeasure*.

A way of thinking about the universal semimeasure

Toss a fair coin repeatedly until you produce a string in $\mathcal{P}$.

A way of thinking about the universal semimeasure

Toss a fair coin repeatedly until you produce a string in $\mathcal{P}$.

What is the probability that the produced string π *is a program for* x? This is $m(x)$.

Not: toss a coin repeatedly until you produce x itself.

K and H

A landmark result:

Theorem

(Levin's coding theorem)

$$K(x) = -\log m(x) + O(1).$$

i.e. K is H when the underlying probability on strings is m!!

K and H

A landmark result:

Theorem

(Levin's coding theorem)

$$K(x) = -\log m(x) + O(1).$$

i.e. K is H when the underlying probability on strings is m!!

Leonid Levin's paraphrase: "If there are a lot of long programs producing a string, then there is a short program for that string."

Symmetry of Information (sort of)

Lemma

Let x and y be arbitrary strings. Then

$$K(x) + K(y \mid x, K(x)) \quad = K(x, y) + O(1).$$

The proof establishes

$$K(y \mid x, K(x)) \leq K(x, y) - K(x).$$

Some Open Areas

Definition

(Yeung and Zhang 98) An information-theoretic inequality is said to be of *non-Shannon type* if it cannot be derived as a linear combination of inequalites of the form $I(X;Y) \geq 0$.

Leung and Yeung showed that there are non-Shannon type inequalities involving 4 or more random variables.

Some Open Areas

Definition

(Yeung and Zhang 98) An information-theoretic inequality is said to be of *non-Shannon type* if it cannot be derived as a linear combination of inequalites of the form $I(X;Y) \geq 0$.

Leung and Yeung showed that there are non-Shannon type inequalities involving 4 or more random variables.

A good *theory* of non-Shannon-type inequalities is lacking in algorithmic information theory.

Suggested Reading

1. A. N. Kolmogorov, "Logical Basis for information theory and probability theory", IEEE Transactions on Information Theory, (14) pages 662-664, 1968.
2. William Poundstone, "Fortune's Formula", Hill and Wang, 2006.
3. C. E. Shannon, "A Mind-Reading (?) Machine".


A. N. Shiryaev.
A. N. Kolmogorov: Life and Creative Activities.
The Annals of Probability, 17(3):866–944, 1989.