Information Theory and Networks

Lecture 21: Kolmogorov Complexity and Probability

Matthew Roughan
<matthew.roughan@adelaide.edu.au>
http://www.maths.adelaide.edu.au/matthew.roughan/Lecture_notes/InformationTheory/

School of Mathematical Sciences,
University of Adelaide

September 18, 2013

Part I

Kolmogorov Complexity and Probability

> Hence it is that we take Delight in a Prospect which is well laid out, and diversified with Fields and Meadows, Woods and Rivers; in those accidental Landskips of Trees, Clouds and Cities, that are sometimes found in the Veins of Marble; in the curious Fret-work of Rocks and Grottos; and, in a Word, in any thing that hath such a Variety or Regularity as may seem the Effect of Design, in what we call the Works of Chance.
>
> *Joseph Addison*

Formal Kolmogorov Complexity

Definition (Kolmogorov Complexity)

The Kolmogorov complexity $K_{\mathcal{U}}(\mathbf{x})$ of a string $\mathbf{x}$ with respect to a universal computer $\mathcal{U}$ is defined as

$$K_{\mathcal{U}}(\mathbf{x}) = \min_{\{\mathbf{p} | \mathcal{U}(\mathbf{p})=\mathbf{x}\}} \ell(\mathbf{p})$$

- So Kolmogorov complexity is about the ultimate bounds of compression
  - its interesting complex sequences are incompressible
- But we know compression and information are related via entropy, so is complexity in the same mix?

Section 1

Kolmogorov Complexity and Entropy

The Kraft Inequality for Kolmogorov Complexity

Theorem

For any computer $\mathcal{U}$

$$\sum_{p\,:\,\mathcal{U}(p) halts} 2^{-\ell(p)} \leq 1$$

Proof.

- If the computer halts, then it does not look for any more input.
- So halting programs are prefix free
- Hence lengths satisfy Kraft inequality (see earlier proofs)

□

Kolmogorov Complexity and Entropy

Theorem

For an IID stochastic process $\{X_i\}$ over a finite alphabet Ω, then

$$E\left[\frac{1}{n}K(X_1, X_2, \ldots, X_n|n)\right] \to H(X)$$

This is an interesting result

- Complexity is a statement about a single sequence
  - not a “random” sequence (its given)
  - complexity is ultimately incomputable
- Entropy is a statement about an ensemble of sequences
  - given such an ensemble, entropy is the (long term) average of the complexities (even though they are incomputable)

Kolmogorov Complexity and Randomness

What is “random” when we are talking about a single sequence?

- more precisely, given a single sequence, could we argue it was random or not?
- we don’t *a priori* know a model
- we don’t have an ensemble to look at

We care because we are all the time generating pseudo-random sequences. Then we use them in things like cryptography which need really random numbers.

Heads.

Heads.

⋮

Heads.

⋮

It must be indicative of something besides the redistribution of wealth.

Heads.

A weaker man might be moved to re-examine his faith, if for nothing else at least in the law of probability...

Heads.

⋮

Rosencrantz And Guildenstern Are Dead,
Tom Stoppard

Kolmogorov Complexity and Randomness

What is “random”?

Theorem

Let $\{X_i\}$ be drawn from a fair Bernoulli process, i.e., with $p = 1/2$, then

$$P\big(K(X_1, X_2, \ldots, X_n|n) < n - k\big) < 2^{-k}$$

So most such random sequences have complexity close to their length. Lets invert that idea to get a definition for randomness in terms of complexity.

Definition (Algorithmically Random)

A sequence $x_1, x_2, \ldots, x_n$ is algorithmically random if

$$K(x_1, x_2, \ldots, x_n|n) \geq n$$

Compressibility and Randomness

Definition (Incompressible)

An infinite string is incompressible if

$$\lim_{n\to\infty} \frac{1}{n} K(x_1, x_2, \ldots, x_n | n) = 1$$

Theorem (Strong Law of Large No.s for Incompressible Sequences)

If a binary string $x_1, x_2, \ldots$ is incompressible, then it satisfies

$$\frac{1}{n}\sum_{i=1}^{n} x_i \to \frac{1}{2}$$

So incompressible sequences look random: they have the same proportion of ones and zeros as a set of fair coin tosses. But we could have done the same thing for any set of substrings, so any statistic of the sequence will satisfy the statistical tests for randomness.

Section 2

Universal Probability

Infinite Monkeys

- Imagine monkeys typing randomly on a keyboard
  - or professors :)
- We know they eventually type Shakepearse's collected works
  - eventually
- Presume they type a random program
  - most programs will be nonsense
  - but a few will execute
- What sort of output should we expect?

Monkey Programs

- What sort of output should we expect?
  - random program $\mathbf{p}$ has probability $P(\mathbf{p}) = 2^{-\ell(\mathbf{p})}$
    - shorter programs are more likely
  - if a short program produces a long output $\mathbf{x}$, then that output must be highly compressible
    - string must have structure
    - it isn't algorithmically random
  - but most strings are close to random (see earlier)
    - so simple strings are more likely than complex strings of the same length

Definition (Universal Probability)

The universal probability of a string $\mathbf{x}$ is

$$P_{\mathcal{U}}(\mathbf{x}) = P\big(\mathcal{U}(p) = \mathbf{x}\big) = \sum_{\mathbf{p}\,:\,\mathcal{U}(\mathbf{p})=\mathbf{x}} 2^{-\ell(\mathbf{p})}$$

Universal Probability

Definition (Universal Probability)

The universal probability of a string **x** is

$$P_{\mathcal{U}}(\mathbf{x}) = P\big(\mathcal{U}(p) = \mathbf{x}\big) = \sum_{\mathbf{p}\,:\,\mathcal{U}(\mathbf{p})=\mathbf{x}} 2^{-\ell(\mathbf{p})}$$

Might be

- probability of the string in nature
  - think of inputs as random in some sense, transformed by some process (nature)
- probability of financial time series
  - random inputs, transformed by market

Implicit belief is that simpler strings are more likely than complicated strings: e.g. Occam's Razor (choose the shortest program that can generate a given string).

Universal Probability

Definition (Universal Probability)

The universal probability of a string **x** is

$$P_{\mathcal{U}}(\mathbf{x}) = P\big(\mathcal{U}(p) = \mathbf{x}\big) = \sum_{\mathbf{p} : \mathcal{U}(\mathbf{p}) = \mathbf{x}} 2^{-\ell(\mathbf{p})}$$

Might be

- probability of the string in nature
  - think of inputs as random in some sense, transformed by some process (nature)
- probability of financial time series
  - random inputs, transformed by market

Implicit belief is that simpler strings are more likely than complicated strings: e.g. Occam's Razor (choose the shortest program that can generate a given string).

Universal Probability and Kolmogorov Complexity

Theorem

There exists a constant c such that for all strings $\mathbf{x}$

$$2^{-K(\mathbf{x})} \leq P_{\mathcal{U}}(\mathbf{x}) \leq c 2^{-K(\mathbf{x})}$$

- So universal probability is essentially determined by complexity.
- The theorem confirms our intuition that simpler sequences are more probable.
- Can also write as

$$K(\mathbf{x}) - c' \leq -\log_2 P_{\mathcal{U}}(\mathbf{x}) \leq K(\mathbf{x})$$

- Remember Shannon code lengths

$$\ell(\mathbf{x}) = \lceil -\log p(\mathbf{x}) \rceil$$

- So we have come around full circle between complexity, compressibility, randomness and probability.

Assignment

1. Argue that the Kolmogorov complexity of a sequence $\mathbf{x}\,\mathbf{y}$ formed by concatenating $\mathbf{x}, \mathbf{y} \in \{0,1\}^*$ satisfies
$$K(\mathbf{x}\,\mathbf{y}) \leq K(\mathbf{x}) + K(\mathbf{y}) + c.$$
Now give an example where the two sequences are complex, but the concatenation is relatively simple.
2. Suppose you have Monkey's typing random 1s and 0s. Give a rough estimate of the probability that the Monkey types:
    1. 0^n,
    2. $\pi_1\pi_2\ldots\pi_n$ (where π_i is ith bit in the binary expansion of π),
    3. A binary representation of the complete works of Shakespeare.

    Now imagine the monkey is typing a random program into a computer. Estimate now the rough probability that the computer outputs
    1. 0^n followed by any arbitrary sequence,
    2. $\pi_1\pi_2\ldots\pi_n$ followed by any arbitrary sequence,
    3. A binary representation of the complete works of Shakespeare followed by any arbitrary sequence.

Further reading I

- Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, John Wiley and Sons, 1991.