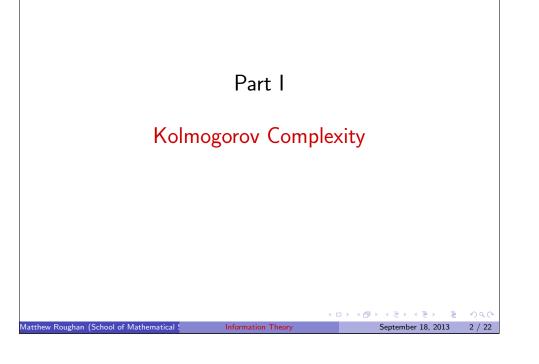
Information Theory and Networks Lecture 20: Kolmogorov Complexity

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

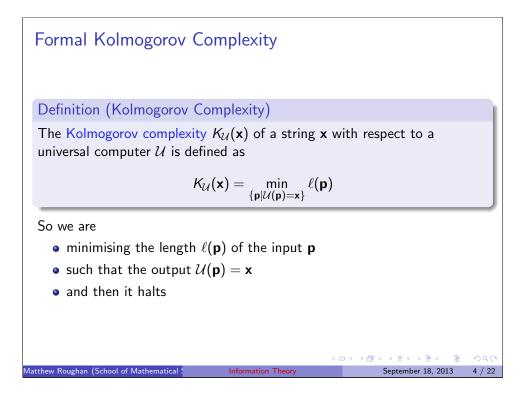
> School of Mathematical Sciences, University of Adelaide

> > September 18, 2013



information. Edward Tuft
Clutter and confusion are failures of design, not attributes of

Information Theory 81- 60- 2012	Cutter and confusion are failows of design, net attributes of information Educate Turk





Theorem

If \mathcal{U} is a universal computer, then for any other computer \mathcal{A}

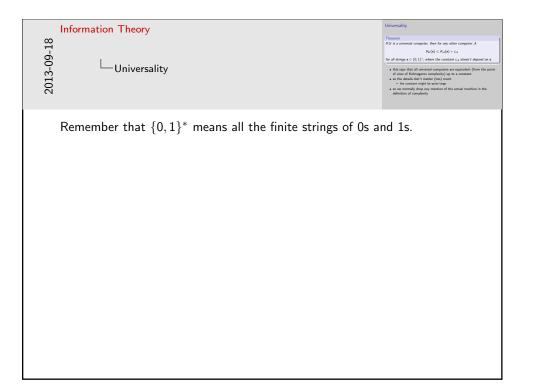
 $K_{\mathcal{U}}(\mathbf{x}) \leq K_{\mathcal{A}}(\mathbf{x}) + c_{\mathcal{A}}$

for all strings $\mathbf{x} \in \{0,1\}^*$, where the constant c_A doesn't depend on \mathbf{x} .

- this says that all universal computers are equivalent (from the point of view of Kolmogorov complexity) up to a constant.
- so the details don't matter (too) much
 - ► the constant might be quite large
- so we normally drop any mention of the actual machine in the definition of complexity

Information Theory

Formal Kolmogorov Information Theory 2013-09-18 $K_{U}(\mathbf{x}) = \min_{\{\mathbf{a}^{2}/(\mathbf{a})=\mathbf{x}\}} \ell\{$ Formal Kolmogorov Complexity gth $\ell(\mathbf{p})$ of the input \mathbf{p} out $\mathcal{U}(\mathbf{p}) = \mathbf{x}$



September 18, 2013

5 / 22

Universality

Proof.

Assume program $\mathbf{p}_{\mathcal{A}}$ for computer \mathcal{A} prints \mathbf{x} , i.e., $\mathcal{A}(\mathbf{p}_{\mathcal{A}}) = \mathbf{x}$.

A \mathcal{U} is a universal computer we can write a simulator for \mathcal{A} in \mathcal{U} , call it $\mathbf{s}_{\mathcal{A}}$.

So the program $\mathbf{s}_{\mathcal{A}} \mathbf{p}_{\mathcal{A}}$, input to \mathcal{U} will simulate the output $\mathcal{A}(\mathbf{p}_{\mathcal{A}})$, i.e., the desired output.

The length of this program is

$$\ell(\mathbf{s}_{\mathcal{A}}\,\mathbf{p}_{\mathcal{A}}) = \ell(\mathbf{s}_{\mathcal{A}}) + \ell(\mathbf{p}_{\mathcal{A}})$$

where $\ell(\mathbf{s}_{\mathcal{A}}) = c_{\mathcal{A}}$ is constant with respect to **x**.

The Kolmogorov complexity is the minimum over such programs, and so it becomes an inequality, because there might be a better way to generate the same sequence.

Examples

atthew Roughan (School of Mathematical

• An integer *n* (written in binary) has

 $K(n) \leq 2\log_2 n + c$

To describe n, repeat every bit of the binary expansion of n twice; then end the description with a 01. Example:

Information Theory

- n = 5, which in binary is 101
- ▶ write as 11,00,11,01
- The first *n* digits of π
 - we know a program to generate digits
 - we also need to let it know how many to generate

00	Information Theory	Universality Proof. Assume program $p_{\mathcal{A}}$ for computer \mathcal{A} prints x_i i.e., $\mathcal{A}(p_{\mathcal{A}})=x_i$
2013-09-18	Universality	A 2/ is a strengt comparison of the λ = 1/4, $\lambda_{\rm el}$, Si H $\lambda_{\rm el}$. Si the segment $x_{\rm el}$, and the strength $\lambda_{\rm el}$ (as the strength $\lambda_{\rm el}$), is a the density of the strength $\lambda_{\rm el}$ (as $\lambda_{\rm el}$). The bargeting $\lambda_{\rm el}$ (as $\lambda_{\rm el}$) = $\lambda_{\rm el}$ ($\lambda_{\rm el}$) = $\lambda_{\rm el}$). The Kolmponer complexity is the strength the scheme as the gragment and the size that there might be a better way be generated the same supresson.



Might be more expressive to write $K(n) \leq 2\lceil \log_2 n \rceil + c$, but the extra bit can easily be moved to the constant.

The commas in the example are just for clarity – they wouldn't be in the actual program.

You can actually do a little better using iterated logs:

 $log^*n = \log n + \log \log n + \cdots$

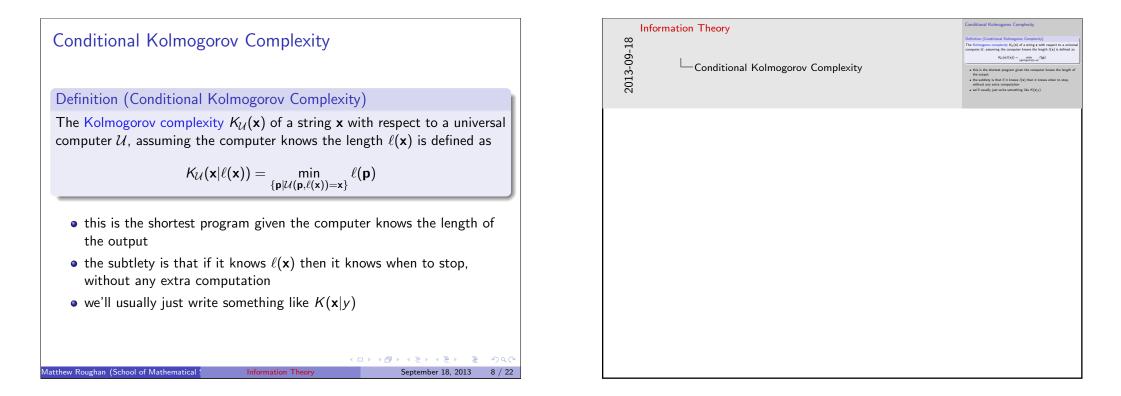
(see [CT91, pp.148-149]).

September 18, 2013

<ロ> (四) (四) (三) (三) (三) (三)

September 18, 2013

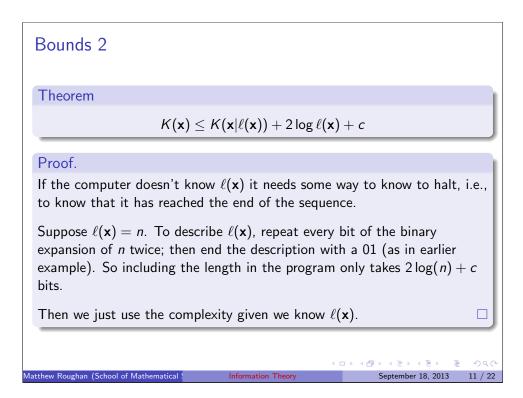
6 / 22



Examples	
 K(00000 ℓ) = c for all ℓ Print ℓ zeros Similar for any simple repeated sequence. K(π₁π₂π_ℓ ℓ) = c for all ℓ We know (short) constant length programs to output the digits of π, given we know how many to output. K(image ℓ) ≤ ℓ/3 + c Use standard compression algorithms, which can probably compress it by about a factor of 3, without any loss. A sequence with n bits and k ones? 	
<ロレイアン ミン ミン マックロン (School of Mathematical : Information Theory September 18, 2013 9 /	

	mation Theory	Examples
2013-09-18	Examples	$\label{eq:rescaled} \begin{array}{l} \mathcal{R}(\mathrm{SDES})_{-}(0)(\gamma-4\mathrm{fm}\mathrm{d}\mathrm{d}\mathrm{d}\mathrm{stream}\mathrm{d}\mathrm{stream}\mathrm{d}\mathrm{stream}\mathrm{d}\mathrm{stream}\mathrm{d}\mathrm{stream}\mathrm{d}\mathrm{stream}\mathrm{d}\mathrm{stream}\mathrm{stream}\mathrm{d}\mathrm{stream}stream$

Bounds 1			
Theorem			
	$K(\mathbf{x} \ell(\mathbf{x})) \leq \ell(\mathbf{x}) + c$		
Proof.			
Intuitively, we just write a print the following No bits are needed for ℓ as	ℓ -bit sequence x_1x_2	$\ldots x_{\ell}$	
		1 -	৩৫৫
Matthew Roughan (School of Mathematical	Information Theory	September 18, 2013	10 / 22



Information Theory	Bounds 1 Theorem K(wi(ks)) ≤ f(k) + c Proof. Intolody, we just write a program that upper Prior the Allibring of this expenses a No

	Information Theory	Bounds 2
2013-09-18	Bounds 2	$\label{eq:second} \begin{split} \hline R(s) & \leq R(s(t)(s) + 2 \log t(s) + c \\ \hline Post \\ \hline Post \\ \hline B the compare Asser's boson's (t) it much scores way the boson to halt, i.e., it is boson that the score as the score of the samples. The base the score is the$
	Note we can't efficiently include x in the input because thi	م سميراط خمادم

Note we can't efficiently include x in the input, because this would take $\ell(x)$ bits, and defeat the whole point of trying to find a shorter program to write x

You can actually do a little better using iterated logs:

 $\log^* n = \log n + \log \log n + \cdots$

(see [CT91, pp.148-149]).

Bounds 3

Theorem

The number of strings with complexity $K(\mathbf{x}) < k$ satisfies

$$\left| \left\{ \mathbf{x} \in \{0,1\}^* \mid K(\mathbf{x}) < k
ight\} \right| < 2^k$$

Proof.

List all of the (binary) programs i, and we get 2^i .

Add up all the programs shorter than k and we get

$$\sum_{i=0}^{k-1} 2^i = 2^k - 1 < 2^k$$

Since each program can produce only one output sequence, the number of sequences with complexity < k is $< 2^k$.

Information Theory

Theorem

$$\frac{1}{n+1}2^{nH(k/n)} \le \binom{n}{k} \le 2^{nH(k/n)}$$

Proof.

latthew Rough

Stirling's approximation

atthew Roughan (School of Mathematical :

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)$$

Combinations

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$\sim \sqrt{\frac{n}{2\pi k(n-k)}} \left(\frac{n}{k}\right)^k \left(\frac{n}{n-k}\right)^{n-k}$$

$$\square$$
In (School of Mathematical)

n

Information Theor	Bounds 3 Therean The number of strongs with complexity $K[x] < k$ satisfies $ \{x \in \{0,1\}^n \mid K[x] < k\} < 2^n$ Recall List of the bioary) programs <i>i</i> , and we get 2 ⁿ . Add up all the programs bottler that has the one get $\sum_{i=1}^{n-1} x^i = x^i - 1 < 2^n$. Since such programs <i>c</i> , and positive only an example to the number of sequences with complexity < k < 2 ⁿ .



http://en.wikipedia.org/wiki/Stirling's_approximation or see Feller [Fel71, Chapter VII.2].

Note that although Stirling's approximation is an asymptotic formula, it's pretty good even for moderate n, and it comes with bounds, so we could do a tighter proof if needed using

$$\sqrt{2\pi n} \left(\frac{n}{e}\right)^n \le n! \le \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \frac{e}{\sqrt{2\pi}}$$

・ロト ・四ト ・ヨト ・ヨト

-2

September 18, 2013 12 / 22

Proof.
And

$$2^{nH(k/n)} = 2^{-k \log_2(k/n) - (n-k) \log_2(1-k/n)} = 2^{k \log_2(n/k)} 2^{(n-k) \log_2(n/(n-k))} = \left(\frac{n}{k}\right)^k \left(\frac{n}{n-k}\right)^{n-k}$$
Also for $k = 1, \dots, n-1$ the term $\sqrt{\frac{n}{2\pi k(n-k)}}$ takes its minimum value for $k = n/2$
 $\sqrt{\frac{n}{2\pi k(n-k)}} = \sqrt{\frac{2}{\pi n}} \ge \frac{1}{n+1}$
and maximum for $k = 1$, so
 $\sqrt{\frac{n}{2\pi k(n-k)}} = \sqrt{\frac{n}{2\pi(n-1)}} \le 1$

Example

Can we compress a sequence of n bits with k ones?

• earlier result was there is no universal compression, so we might guess no

Information Theory

• but the problem is subtlety different

```
Use the following program:
```

- The program has fixed length
- We need to specify
 - k which has range $0, \ldots, n$
 - *i* has conditional range $\binom{n}{k}$

 Information
 Theory
 Pade Aut

 2^{000/00} = $\frac{1}{2}$ · · · · · · 1 the terry $\sqrt{2\pi b(2+1)}$ $= \frac{1}{2}$ · · · · · · · · 1 the terry $\sqrt{2\pi b(2+1)}$ Also for k = 1, ..., k = 1 the terry $\sqrt{2\pi b(2+1)}$ (while a minimum value for k = n/2 $\sqrt{\frac{2\pi b(k-1)}{2\pi b(k-1)}} = \sqrt{\frac{2\pi}{2\pi b(k-1)}}$ (and a minimum value for k = n/2 $\sqrt{\frac{2\pi b(k-1)}{2\pi b(k-1)}} = \sqrt{\frac{2\pi}{2\pi b(k-1)}}$ (b)

Note for k = 0 or k = n cases we can't really use the formulae above. Dealing with them as special cases we use

$$H(0)=H(1)=0$$

So the inequality is

$$\frac{1}{n+1}2^{nH(k/n)} \leq \binom{n}{k} \leq 2^{nH(k/n)}$$
$$\frac{1}{n+1}2^0 \leq \binom{n}{0} \leq 2^0$$
$$\frac{1}{n+1} \leq 1 \leq 2$$

Information Theory	Example Can as compress a sequence of r bits with it own? • serifler much that share is no universal compression, so we might goesn • no • but the problem is subtrive different
5013-09-18 Example	Use the following pergupts: Generator, in Intra-patient events: Of these, print the site of the events: Of these, print the site of the events: The program has find elength = We need to specify We need to specify the

September 18, 2013 15 / 22

<□> <□> <□> <□> <=> <=> <=> <=> <000

Example

Use the following program:

The length of the above is

$$\ell(p) = c + 2\log_2(k) + \log_2\binom{n}{k}$$

Information Theory

(ロ) (四) (E) (E) (E) (E)

September 18, 2013

17 / 22

September 18, 2013 16 / 22

- The program has fixed length c_0 bits
- We need to specify

atthew Roughan (School of Mathematical (

- k which takes $2\log_2(k) + c_1$ bits
- *i* which takes up to $\log_2 \binom{n}{k} + c_2$ bits
 - ***** worst case is that $i = \binom{n}{k}$

Example Theorem The Kolmogorov complexity of a binary string **x** with k ones is bounded by $K(x_1x_2...x_n|n) \le nH\left(\frac{k}{n}\right) + 2\log n + c$ Proof. Use the program from the last example, and note that $k \le n$ and (from result above) $\log_2{\binom{n}{k}} \le nH\left(\frac{k}{n}\right)$

Information Theory

Information Theory 81 60 60 60 60 60 60 60 60 60 60 60 60 60	Example Use the sequence of the set of the sequence of the set o

Informatio 81 	n Theory –Example	Example Taxors Tax Kolongoro: complexity of a binary string x with k ones is bounded by $K(w_2, \dots, w_d) \le stf(\frac{k}{2}) + 2\log a + c$ Point bush programs from the last example, and notes that $k \le a$ and (from $\log_{10} \left(\frac{k}{2}\right) \le stf\left(\frac{k}{2}\right)$

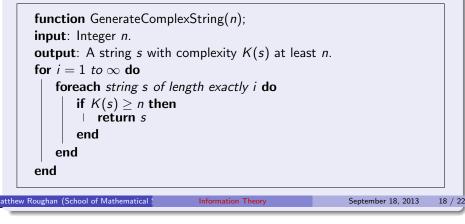
Incomputability

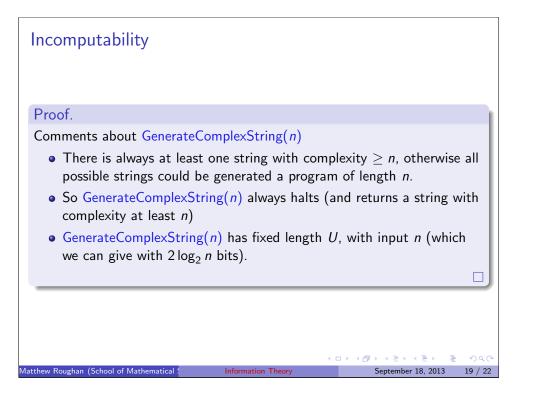
Theorem

The Kolmogorov complexity $K(\mathbf{x})$ is not a computable function (i.e., no program with input \mathbf{x} produces $K(\mathbf{x})$ as output).

Proof.

Imagine such a program exists. Now consider the function





o	Information Theory	Т 7	Fhe The	eon e K	nputability em olmogorov complexity K(x) is not a computal m with input x produces K(x) as output).
7-60-5102	Incomputability	P	ma	oof agin fu in os fo	

Not computable doesn't mean "it's hard", or "We don't know how to compute it", it means there are some values that can never be computed, no matter how smart we are.

Information Theory Incomputability Incomputabi

Incomputability

Proof.

Now define

function GeneratePardoxialString; **output**: A string *s* with complexity K(s) at least n_0 . **return** GenerateComplexString (n_0)

• The length of GeneratePardoxialString is at most

 $U+2\log_2(n_0)+c$

• Since *n* grows faster than $\log_2 n$, there must be a value n_0 such that

$$U + 2\log_2(n_0) + c < n_0$$

But that means there is a function to generate s, whose length is less than n_0 , but the function GenerateComplexString(n_0), created a string s whose complexity was at least n_0 . Hence we have a contradiction.

Information Theory

atthew Roughan (School of Mathematical S

Berry Paradox

The smallest positive integer not definable in under eleven words.

G.G.Berry (1867-1928)

Think about it:

- There are a finite number of words, and hence finite number of sentences with less than eleven words.
- Hence a finite number of positive integers describable, and hence an infinite number that aren't.
- By well ordering property of integers, there is therefore a least such integer.
- But the above description is 10 words, and hence, it is defined with under 11 words.

Information Theory

 $\bullet\,$ Thus it no longer is described by the words, so it isn't \ldots

This leads to Chaitin, Gödel and Escher

21 / 22

September 18, 2013

20 / 22

2013-09-18	Incomputability	Pout Nor de en • Th • Th	$\label{eq:second} \begin{split} & & & & & & & & & & & & & & & & & & $
		n, bet comp	the function conversion (see the second seco

	ation Theory	Berry Paradox The smallest positive integer not definable in under eleven words. <i>G.G.Berry (1867-1628)</i> Think about it
2013-09-18	Berry Paradox	 There are a finite number of seeds, and hence follow number of surrences with the sin-share server. Hence a finite number of possible integrad describubly, and hence an infinite number that area. By well ordering segarity of integras, then is therefore a listed such integrad. By well ordering segarity of integras, then is therefore a listed such integra. But the also discription is 10 words, and hence, it is defined with describe description is the overfit of the listed sector of the number of the listed s

Further reading I



William Feller, *An introduction to probability theory and its applications*, second ed., vol. I, John Wiley and Sons, New York, 1971.

Information Theory

Matthew Roughan (School of Mathematical

