

Algorithmically Independent Sequences

Cristian Calude¹ Marius Zimand²

¹University of Auckland, New Zealand

²Towson University, USA

DLT, Sept 2008

Basic Question

When are two objects x and y independent?

Informal Idea

x and y are independent if

Information[x | y] = Information[x], and

Information[y | x] = Information[y]

Example: Distributions

- X, Y : distributions on a discrete space.
- $\text{Information}[X]$: Shannon entropy of X , denoted $H(X)$.
- X and Y are independent: $H(X | Y) = H(X)$.

Example: Finite Binary strings

- x, y : finite binary strings.
- $\text{Information}[x]$: Kolmogorov complexity of x , denoted $C(x)$.
- x, y are independent: $C(x | y) \approx C(x)$.

Kolmogorov complexity of a string is the length of its shortest description.

- $\overbrace{0101 \dots 01}^{10^{100}}$ has a short description.
- flipping a coin 10^{100} times:
 $011000101010110010101000101001011 \dots 100$:
 description $\approx 10^{100}$ bits.

Kolmogorov complexity

Plain Kolmogorov complexity

$C(x) = \min\{|p| \mid U(p) = x\}$; $C(x \mid y) = \min\{|p| \mid U(p, y) = x\}$,
where U is a fixed universal Turing machine.

Prefix-free Kolmogorov complexity

$H(x) = \min\{|p| \mid U(p) = x\}$; $H(x \mid y) = \min\{|p| \mid U(p, y) = x\}$,
where U is a fixed prefix-free universal Turing machine.

$$|H(x) - C(x)| \leq 2 \log |x| + O(1).$$

Relativized Kolmogorov complexity

$C^y(x) = \min\{|p| \mid U^y(p) = x\}$ where U is a fixed **oracle** universal Turing machine and y is the oracle. $H^y(x)$ - similar.

Sequence x is random $\Leftrightarrow H(x \upharpoonright n) \geq n - O(1)$.

This Paper: Infinite Binary Sequences

- Sequence: an element of $\{0, 1\}^\infty$.
- The case of sequences has been less studied.
- Literature: Two sequences x and y are independent if each one is random relative to the other one.

$$H^y(x \upharpoonright n) \geq n - O(1) \text{ and } H^x(y \upharpoonright n) \geq n - O(1), \forall n.$$

- Thus independence has been defined only when x and y are random.

OUR GOAL

Define a notion of independence for sequences that

- is not confined to random sequences,
- is natural,
- is simple to use.

Defining Independence

Idea

Sequences x and y are independent if for any prefix x' of x ,
 $\text{Information}[x'] \approx \text{Information}[x' \text{ given } y]$.

Depending on how y is presented, we arrive at two notions of independence.

- **independence**, when y is available in its entirety;
- **finitary-independence**, when y is available through its prefixes.

Two Types of Independence

Definition: Independence

Sequences x and y are independent if for every n ,

$$C^y(x \upharpoonright n) \geq C(x \upharpoonright n) - O(\log n),$$

$$C^x(y \upharpoonright n) \geq C(y \upharpoonright n) - O(\log n).$$

Definition: Finitary-Independence

Sequences x and y are finitary- independent if for every n, m ,

$$C(x \upharpoonright n \mid y \upharpoonright m) \geq C(x \upharpoonright n) - O(\log n + \log m),$$

$$C(y \upharpoonright m \mid x \upharpoonright n) \geq C(y \upharpoonright m) - O(\log n + \log m).$$

Trivial Independence

- DEF: A sequence x is **H-trivial** if $H(x \upharpoonright n) \leq H(n) + O(1)$.
- DEF: A sequence x is **C-logarithmic**, if $C(x \upharpoonright n) = O(\log n)$.
- If x is H-trivial, then x is indep. with any seq. y .
- If x is C-logarithmic, then x is finitary-indep. with any seq. y .
- Why: x is so simple that it does not make a difference whether it is available or not.
- We exclude these trivial instances of (finitary-) indep. from our discussion.

Some Properties

- (x, y) indep $\Rightarrow (x, y)$ finitary-indep. The converse is not true.
- (Finitary-) independence is symmetric.
- Why the additive logarithmic inaccuracy: (Finitary-) independence is robust under type of complexity, and has other technical advantages.
- For any x , most y are indep. with x : $\{y \mid (x, y) \text{ indep.}\}$ has measure 1.
- For any Turing reduction f , x and $f(x)$ are dependent (except for the trivial case).
- **Independence** cannot be generated effectively.
- What about **finitary-independence**?

Generating fin.-indep. sequences: impossibility results

One source case:

We start with one seq. x . Is it possible to produce finitary-independent sequences?

- **Question A.** Given x , is there a procedure that builds y with (x, y) fin.-indep?
 - Uniform version: procedure works for any x .
 - Non-uniform version: procedure depends on x
- **Question B.** Given x , is there a procedure that builds y, z with (y, z) fin.-indep?
 - Uniform version: procedure works for any x .
 - Non-uniform version: procedure depends on x

- To avoid the trivial case, we need to require that x, y, z are non-C-logarithmic.
- We have results for x, y, z with complexity “just” above that of non-C-logarithmic sequences.
- DEF: x is C-superlogarithmic if for any $c > 0$, $C(x \upharpoonright n) > c \log n$, for almost every n .
- DEF (effective Hausdorff dimension): for a seq. x ,
$$\dim(x) = \liminf \frac{C(x \upharpoonright n)}{n}.$$

- The answer to the uniform versions: NO.
- Question B (uniform): Is there a procedure that on any input x (sufficiently complex to avoid triviality) produces (y, z) fin.-indep.
- TASK for Turing reductions f_1 and f_2 : For every seq. x with $\dim(x) > 0$, the following should hold:
 - $f_1(x), f_2(x)$ both exist;
 - $f_1(x), f_2(x)$ both C-superlogarithmic;
 - $(f_1(x), f_2(x))$ finitary-independent.

Theorem

There are no f_1, f_2 that perform the TASK.

- Proof:

Bienvenu, Doty, Stephan'07. For one source: There is no uniform way to increase the complexity rate from 0.5 to 0.51.

- For any Turing reduction h , any computable, infinite set S , there is x with $\dim(x) = 0.5$ and $C(h(x) \upharpoonright n) < 0.51n$ for infinitely many n in S .

Zimand'08. For two fin. indep. sources: There is a uniform way to increase the complexity rate from very low to almost 1.

- There is a Turing reduction f_{ZIM} and an infinite computable set S s.t. for every seq. x and y with $C(x \upharpoonright n) = \Omega(\log n)$ and $C(y \upharpoonright n) = \Omega(\log n)$, it holds that $C(f_{\text{ZIM}}(x, y) \upharpoonright n) > 0.99n$ for almost every n in S .
- So, if there are f_1, f_2 performing the TASK, $f_{\text{ZIM}}(f_1(x), f_2(x))$ would have prefixes of length n with complexity $\geq 0.99n$ for almost every n in S , which contradicts **BDS'07**.

- Nonuniform versions: Open, but we have some weaker results.
- TASK: For any x with $\dim(x) > 0$, there is a procedure f that on input x produces y, z that are finitary-independent and $\dim(y) > 0, \dim(z) > 0$.

Theorem

There is x for which the TASK cannot be performed.

- Proof uses

Miller'08 (solving a famous question of Reimann & Terwijn).
There is a seq. x with $\dim(x) = 0.5$ and for any Turing reduction f , $\dim(f(x)) \leq 0.5$.

Two or more sources case:

We start with two or more (finitary-) indep. sequences. Is it possible to effectively produce from them more (finitary-) independent sequences?

- There are indep. seq. x, y and a Turing reduction g s.t. x and $g(y)$ are not indep.
- There are fin.-indep. seq. x, y and a Turing reduction g s.t. x and $g(y)$ are not fin.-indep.
- This looks bad for our concepts of independence. One expects independence to be preserved under computable transformations.
- But maybe such phenomena are unavoidable for any notion of independence that applies to nonrandom sequences (which is our objective), because Hausdorff dimension is not preserved under computable transformations either.
- Also, in a weaker form independence is preserved: If x, y are independent and g is a Turing reduction, then $x, g(y)$ are finitary-independent.

- Question: The input consists of a list with n (finitary-) independent sequences. Is it possible to produce a sequence that is independent with any sequence from the input list?
- If the sequences in the input list are random, the answer is easily seen to be YES: Take the bitwise XOR of the sequences in the input list.
- If the sequences in the input list are not random, we don't know the answer.
- For the finite case, we have a positive answer for $n \geq 3$.

Theorem - informal statement

There is a computable function f that on input a triplet of strings (x, y, z) , with x, y, z independent and having linear Kolmogorov complexity produces a string w that is independent with x , with y , and with z .

- The proof uses techniques from the area of randomness extractors.
- **Breaking news:** The result holds also for $n = 2$.

- Using the probabilistic method we show that such a coloring exists and we find one by exhaustive search.
- Let $w = \text{color of cell } (x, y, z)$. We show $K(w|z) \geq m - 8 \log n$.
- Suppose $K(w|z) < m - 8 \log n$.
- Let $A = \{u \in \{0, 1\}^m \mid K(u|z) < m - 8 \log n\}$. $\frac{|A|}{M} \leq \frac{1}{2^{8 \log n}}$.
- Let $B_1 = \{u \in \{0, 1\}^n \mid K(u) \leq K(x)\}$,
 $B_2 = \{u \in \{0, 1\}^n \mid K(u) \leq K(y)\}$
- Look at rectangle $B_1 \times B_2 \times \{z\}$, which has size $\approx 2^{K(x)+K(y)}$.
- Number of cells in the rectangle $B_1 \times B_2 \times \{z\}$ with a color from A is $\leq 2^{K(x)+K(y)-8 \log n + O(1)}$.
- Cell (x, y, z) is in the rectangle and has color from A .
- The set of cells in rectangle with color from A can be enumerated given $z, K(x)$ and $K(y)$.
- $K(xy|z) \leq K(x) + K(y) - 4 \log n + O(1)$, contradicts the independence of (x, y, z) .

Thank you.