Algorithmic Randomness

Denis R. Hirschfeldt — University of Chicago

NZIMA / NZMRI Summer Workshop — Napier, New Zealand, January 2009

"How dare we speak of the laws of chance?
Is not chance the antithesis of all law?"

— Joseph Bertrand, Calcul des Probabilités, 1889

Part 1: Three Approaches to Defining Randomness

Computability Theory

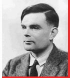
A First Look at Randomness

The Statistician's Approach: Martin-Löf Randomness

The Coder's Approach: Kolmogorov complexity

The Gambler's Approach: Martingales

Part 1: Three Approaches to Defining Randomness



Computability Theory

A First Look at Randomness

The Statistician's Approach: Martin-Löf Randomness

The Coder's Approach: Kolmogorov complexity

The Gambler's Approach: Martingales

Computable Functions

We work with functions $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are countable sets like $\mathbb{N}$, $2^{<\omega}$, $\mathbb{Q}$, $\{0, 1\}$, etc.

Computable Functions

We work with functions $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are countable sets like $\mathbb{N}$, $2^{<\omega}$, $\mathbb{Q}$, $\{0, 1\}$, etc.

We identify $A \subset \mathcal{X}$ with its characteristic function: the function $f : \mathcal{X} \to \{0, 1\}$ s.t. $x \in A$ iff $f(x) = 1$.

Computable Functions

We work with functions $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are countable sets like $\mathbb{N}$, $2^{<\omega}$, $\mathbb{Q}$, $\{0, 1\}$, etc.

We identify $A \subset \mathcal{X}$ with its characteristic function: the function $f : \mathcal{X} \to \{0, 1\}$ s.t. $x \in A$ iff $f(x) = 1$.

A function is computable if its values can be determined by an algorithm.

The notion of algorithm can be formalized using Turing machines.

Computable Functions

We work with functions $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are countable sets like $\mathbb{N}$, $2^{<\omega}$, $\mathbb{Q}$, $\{0, 1\}$, etc.

We identify $A \subset \mathcal{X}$ with its characteristic function: the function $f : \mathcal{X} \to \{0, 1\}$ s.t. $x \in A$ iff $f(x) = 1$.

A function is computable if its values can be determined by an algorithm.

The notion of algorithm can be formalized using Turing machines.

Example: The set of primes is computable.

Computable Functions

We work with functions $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are countable sets like $\mathbb{N}$, $2^{<\omega}$, $\mathbb{Q}$, $\{0, 1\}$, etc.

We identify $A \subset \mathcal{X}$ with its characteristic function: the function $f : \mathcal{X} \to \{0, 1\}$ s.t. $x \in A$ iff $f(x) = 1$.

A function is computable if its values can be determined by an algorithm.

The notion of algorithm can be formalized using Turing machines.

Example: The set of primes is computable.

On input $n > 0$, run through all $1 < m \leqslant \sqrt{n}$.
  For each m, check whether m divides n.
    If some m does, return 0.
    If no m does, return 1.

Uniformly Computable Functions

A sequence of functions $f_0, f_1, \ldots$ is uniformly computable if there is a single algorithm that on input (e, n) returns $f_e(n)$.

Uniformly Computable Functions

A sequence of functions $f_0, f_1, \ldots$ is uniformly computable if there is a single algorithm that on input (e, n) returns $f_e(n)$.

Prop. There is no way to list the computable functions $\mathbb{N} \to \mathbb{N}$ so that they are uniformly computable.

Uniformly Computable Functions

A sequence of functions $f_0, f_1, \ldots$ is uniformly computable if there is a single algorithm that on input (e, n) returns $f_e(n)$.

Prop. There is no way to list the computable functions $\mathbb{N} \to \mathbb{N}$ so that they are uniformly computable.

Pf. Suppose there were such a listing $f_0, f_1, \ldots$ and let $g(n) = f_n(n) + 1$.

Uniformly Computable Functions

A sequence of functions $f_0, f_1, \ldots$ is uniformly computable if there is a single algorithm that on input (e, n) returns $f_e(n)$.

Prop. There is no way to list the computable functions $\mathbb{N} \to \mathbb{N}$ so that they are uniformly computable.

Pf. Suppose there were such a listing $f_0, f_1, \ldots$ and let $g(n) = f_n(n) + 1$.

Then g is computable, so $f_e = g$ for some e.

Uniformly Computable Functions

A sequence of functions $f_0, f_1, \ldots$ is uniformly computable if there is a single algorithm that on input (e, n) returns $f_e(n)$.

Prop. There is no way to list the computable functions $\mathbb{N} \to \mathbb{N}$ so that they are uniformly computable.

Pf. Suppose there were such a listing $f_0, f_1, \ldots$ and let $g(n) = f_n(n) + 1$.

Then g is computable, so $f_e = g$ for some e.

But then $f_e(e) = g(e) = f_e(e) + 1$, a contradiction. □

Partial Computable Functions

An algorithm is just a finite specification in some language, so we do have a nice listing of all algorithms.

Partial Computable Functions

An algorithm is just a finite specification in some language, so we do have a nice listing of all algorithms.

However, not all algorithms halt on all inputs.

Partial Computable Functions

An algorithm is just a finite specification in some language, so we do have a nice listing of all algorithms.

However, not all algorithms halt on all inputs.

A partial function $f : \mathcal{X} \to \mathcal{Y}$ is one whose domain is a (possibly proper) subset of $\mathcal{X}$.

Partial Computable Functions

An algorithm is just a finite specification in some language, so we do have a nice listing of all algorithms.

However, not all algorithms halt on all inputs.

A partial function $f : \mathcal{X} \to \mathcal{Y}$ is one whose domain is a (possibly proper) subset of $\mathcal{X}$.

We write $f(x)\downarrow$ to mean that $f(x)$ is defined, and $f(x)\uparrow$ otherwise.

If $f(x)\downarrow$ for all $x \in \mathcal{X}$, then f is total.

Partial Computable Functions

An algorithm is just a finite specification in some language, so we do have a nice listing of all algorithms.

However, not all algorithms halt on all inputs.

A partial function $f : \mathcal{X} \to \mathcal{Y}$ is one whose domain is a (possibly proper) subset of $\mathcal{X}$.

We write $f(x)\downarrow$ to mean that $f(x)$ is defined, and $f(x)\uparrow$ otherwise.

If $f(x)\downarrow$ for all $x \in \mathcal{X}$, then f is total.

f is a partial computable function if there is an algorithm that on input x outputs $f(x)$ if $f(x)\downarrow$ and does not halt if $f(x)\uparrow$.

Universal Partial Computable Functions

We can list all partial computable functions $\mathbb{N} \to \mathbb{N}$ as $\Phi_0, \Phi_1, \ldots$ so that there is a single algorithm that on input (e, n) outputs $\Phi_e(n)$ if $\Phi_e(n)\downarrow$ and does not halt if $\Phi_e(n)\uparrow$.

Universal Partial Computable Functions

We can list all partial computable functions $\mathbb{N} \to \mathbb{N}$ as $\Phi_0, \Phi_1, \dots$ so that there is a single algorithm that on input (e, n) outputs $\Phi_e(n)$ if $\Phi_e(n)\downarrow$ and does not halt if $\Phi_e(n)\uparrow$.

This algorithm is universal.

Universal Partial Computable Functions

We can list all partial computable functions $\mathbb{N} \to \mathbb{N}$ as $\Phi_0, \Phi_1, \ldots$ so that there is a single algorithm that on input (e, n) outputs $\Phi_e(n)$ if $\Phi_e(n)\downarrow$ and does not halt if $\Phi_e(n)\uparrow$.

This algorithm is universal.

In the context of partial computable functions $2^{<\omega} \to 2^{<\omega}$, we can take a nice listing $\Phi_0, \Phi_1, \ldots$ and define $U(0^e 1\sigma) = \Phi_e(\sigma)$.

Universal Partial Computable Functions

We can list all partial computable functions $\mathbb{N} \to \mathbb{N}$ as $\Phi_0, \Phi_1, \ldots$ so that there is a single algorithm that on input (e, n) outputs $\Phi_e(n)$ if $\Phi_e(n)\downarrow$ and does not halt if $\Phi_e(n)\uparrow$.

This algorithm is universal.

In the context of partial computable functions $2^{<\omega} \to 2^{<\omega}$, we can take a nice listing $\Phi_0, \Phi_1, \ldots$ and define $U(0^e 1\sigma) = \Phi_e(\sigma)$.

U is a universal partial computable function.

Universal Partial Computable Functions

We can list all partial computable functions $\mathbb{N} \to \mathbb{N}$ as $\Phi_0, \Phi_1, \ldots$ so that there is a single algorithm that on input (e, n) outputs $\Phi_e(n)$ if $\Phi_e(n)\downarrow$ and does not halt if $\Phi_e(n)\uparrow$.

This algorithm is universal.

In the context of partial computable functions $2^{<\omega} \to 2^{<\omega}$, we can take a nice listing $\Phi_0, \Phi_1, \ldots$ and define $U(0^e 1\sigma) = \Phi_e(\sigma)$.

U is a universal partial computable function.

The definition of U depends on the choice of listing, but U's basic properties do not.

The Halting Problem

The Halting Problem $\emptyset'$ is $\{(e, n) : \Phi_e(n)\downarrow\}$.

The Halting Problem

The Halting Problem $\emptyset'$ is $\{(e, n) : \Phi_e(n)\downarrow\}$.

Prop. $\emptyset'$ is not computable.

The Halting Problem

The Halting Problem $\emptyset'$ is $\{(e, n) : \Phi_e(n)\downarrow\}$.

Prop. $\emptyset'$ is not computable.

Pf. Suppose it is and define

$$f_e(n) = \begin{cases} \Phi_e(n) & \text{if } \Phi_e(n)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

The Halting Problem

The Halting Problem $\emptyset'$ is $\{(e,n) : \Phi_e(n)\downarrow\}$.

Prop. $\emptyset'$ is not computable.

Pf. Suppose it is and define

$$f_e(n) = \begin{cases} \Phi_e(n) & \text{if } \Phi_e(n)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Then $f_0, f_1, \ldots$ is a uniformly computable listing of all total computable functions, a contradiction. □

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17 47

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17 47 6

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17 47 6 3,413,217

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17 47 6 3,413,217 57 ...

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17 47 6 3,413,217 57 ...

A set is c.e. iff it is the domain of a partial computable function.

A set is c.e. iff it is the range of a partial computable function.

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17 47 6 3,413,217 57 ...

A set is c.e. iff it is the domain of a partial computable function.

A set is c.e. iff it is the range of a partial computable function.

$\emptyset'$ is c.e., as are, e.g., the word problem for a finitely generated group, the set of solvable Diophantine equations, the set of theorems of a computably specified formal system, etc.

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17   47   6   3,413,217   57   . . .

A set is c.e. iff it is the domain of a partial computable function.

A set is c.e. iff it is the range of a partial computable function.

$\emptyset'$ is c.e., as are, e.g., the word problem for a finitely generated group, the set of solvable Diophantine equations, the set of theorems of a computably specified formal system, etc.

A sequence of sets $A_0, A_1, \ldots$ is uniformly c.e. if there is a single algorithm listing all pairs $(e, n) : n \in A_e$.

Computably Enumerable Sets

A set is computably enumerable (c.e.) if it can be listed by an algorithm, but not necessarily in any particular order.

17	47	6	3,413,217	57	...

A set is c.e. iff it is the domain of a partial computable function.

A set is c.e. iff it is the range of a partial computable function.

$\emptyset'$ is c.e., as are, e.g., the word problem for a finitely generated group, the set of solvable Diophantine equations, the set of theorems of a computably specified formal system, etc.

A sequence of sets $A_0, A_1, \ldots$ is uniformly c.e. if there is a single algorithm listing all pairs $(e, n) : n \in A_e$.

There is a uniformly c.e. listing of all c.e. sets.

Part 1: Three Approaches to Defining Randomness

Computability Theory

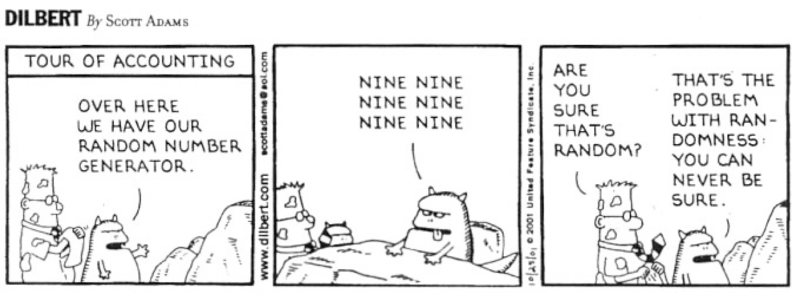
A First Look at Randomness

The Statistician's Approach: Martin-Löf Randomness

The Coder's Approach: Kolmogorov complexity

The Gambler's Approach: Martingales

Intuitive Randomness



Intuitive Randomness

Which of the following binary sequences seem random?

A 000000000000000000000000000000000000000000000000000000000000

B 001101001101001101001101001101001101001101001101001101001101

C 010001101100000101001110010111011100000001001000110100010101

D 001001101101100010001111010100111011001001100000001011010100

E 010101110110111101110010011010110111001101101000011011110111

F 011101111100110110011010010000111111001101100000011011010101

G 000001100010111000100000000101000010110101000000100000000100

H 010100110111101101110101010000010111100000010101110101010001

Intuitive Randomness

Non-randomness: increasingly complex patterns.

A 000000000000000000000000000000000000000000000000000000000000

B 001101001101001101001101001101001101001101001101001101001101

C 010001101100000101001110010111011100000001001000110100010101

D 001001101101100010001111010100111011001001100000001011010100

E 010101110110111101110010011010110111001101101000011011110111

F 011101111100110110011010010000111111001101100000011011010101

G 000001100010111000100000000101000010110101000000100000000100

H 010100110111101101110101010000010111100000010101110101010001

Intuitive Randomness

Randomness: bits coming from atmospheric patterns.

A 000000000000000000000000000000000000000000000000000000000000

B 001101001101001101001101001101001101001101001101001101001101

C 010001101100000101001110010111011100000010010001101000010101

D 001001101101100010001111010100111011001001100000001011010100

E 010101110110111101110010011010110111001101101000011011110111

F 011101111100110110011010010000111111001101100000011011010101

G 000001100010111000100000000101000010110101000000100000000100

H 010100110111101101110101010000010111100000010101110101010001

Intuitive Randomness

Partial Randomness: mixing random and nonrandom sequences.

A 000000000000000000000000000000000000000000000000000000000000

B 001101001101001101001101001101001101001101001101001101001101

C 010001101100000101001110010111011100000001001000110100010101

D 001001101101100010001111010100111011001001100000001011010100

E 010101110110111101110010011010110111001101101000011011110111

F 011101111100110110011010010000111111001101100000011011010101

G 000001100010111000100000000101000010110101000000100000000100

H 010100110111101101110101010000010111100000010101110101010001

Intuitive Randomness

Randomness relative to other measures: biased coins.

A 000000000000000000000000000000000000000000000000000000

B 001101001101001101001101001101001101001101001101001101

C 010001101100000101001110010111011100000001001000110100010101

D 0010011011011000100011110101001110110010011000000001011010100

E 0101011101101111011100100110101101110011011010000110111101111

F 01110111110011011001101001000011111100110110000000110110101010

G 000001100010111000100000000010100001011010100000001000000000100

H 010100110111101101110101010000010111100000001010111010101010001

What Counts as a Nonrandom Pattern?

Consider the following patterns:

1. The sequence α has a 1 in every odd position.

2. Every finite string appears as a segment of α infinitely often.

What Counts as a Nonrandom Pattern?

Consider the following patterns:

1. The sequence α has a 1 in every odd position.
2. Every finite string appears as a segment of α infinitely often.

If α satisfies 1, it is clearly not random.

However, we expect a random α to satisfy 2.

What Counts as a Nonrandom Pattern?

Consider the following patterns:

1. The sequence α has a 1 in every odd position.
2. Every finite string appears as a segment of α infinitely often.

If α satisfies 1, it is clearly not random.

However, we expect a random α to satisfy 2.

Indeed, locally random objects can have highly predictable global structure. For example, the random graph.

What Counts as a Nonrandom Pattern?

Consider the following patterns:

1. The sequence α has a 1 in every odd position.
2. Every finite string appears as a segment of α infinitely often.

If α satisfies 1, it is clearly not random.

However, we expect a random α to satisfy 2.

Indeed, locally random objects can have highly predictable global structure. For example, the random graph.

We need a way to distinguish rare patterns from common patterns.

Three Approaches to Randomness at an Intuitive Level

The statistician's approach: Deal directly with rare patterns using measure theory. Random sequences should not have rare properties.

Three Approaches to Randomness at an Intuitive Level

The statistician's approach: Deal directly with rare patterns using measure theory. Random sequences should not have rare properties.

The coder's approach: Rare patterns can be used to compress information. Random sequences should not be compressible (i.e., easily describable).

Three Approaches to Randomness at an Intuitive Level

The statistician's approach: Deal directly with rare patterns using measure theory. Random sequences should not have rare properties.

The coder's approach: Rare patterns can be used to compress information. Random sequences should not be compressible (i.e., easily describable).

The gambler's approach: A betting strategy can exploit rare patterns. Random sequences should be unpredictable.

Three Approaches to Randomness at an Intuitive Level

The statistician's approach: Deal directly with rare patterns using measure theory. Random sequences should not have rare properties.

The coder's approach: Rare patterns can be used to compress information. Random sequences should not be compressible (i.e., easily describable).

The gambler's approach: A betting strategy can exploit rare patterns. Random sequences should be unpredictable.

We begin by looking at an early attempt to define random sequences, by von Mises.

This attempt predated computability theory.

We will see how each of the three approaches above can be seen as an elaboration on von Mises' flawed attempt.

Von Mises' Approach

For $\alpha \in 2^\omega$, let $R_n(\alpha) = \dfrac{|\{m < n : \alpha(m) = 1\}|}{n}$.

Von Mises' Approach

For $\alpha \in 2^\omega$, let $R_n(\alpha) = \frac{|\{m < n : \alpha(m) = 1\}|}{n}$.

If α is random, we expect it to satisfy the law of large numbers: $\lim_n R_n(\alpha) = \frac{1}{2}$.

Von Mises' Approach

For $\alpha \in 2^\omega$, let $R_n(\alpha) = \dfrac{|\{m < n : \alpha(m) = 1\}|}{n}$.

If α is random, we expect it to satisfy the law of large numbers: $\lim_n R_n(\alpha) = \frac{1}{2}$.

But of course that law is not enough to characterize randomness, since $010101\ldots$, say, satisfies it.

Von Mises' Approach

For $\alpha \in 2^\omega$, let $R_n(\alpha) = \frac{|\{m < n : \alpha(m) = 1\}|}{n}$.

If α is random, we expect it to satisfy the law of large numbers: $\lim_n R_n(\alpha) = \frac{1}{2}$.

But of course that law is not enough to characterize randomness, since $010101\ldots$, say, satisfies it.

Von Mises' basic idea: A gambler should not be able to make any money on a random sequence.

If a gambler can determine a subsequence of α that violates the law of large numbers, then the gambler can make money on α in the long run, so α is not random.

Von Mises' Approach

For $\alpha \in 2^\omega$, let $R_n(\alpha) = \dfrac{|\{m < n : \alpha(m) = 1\}|}{n}$.

If α is random, we expect it to satisfy the law of large numbers: $\lim_n R_n(\alpha) = \frac{1}{2}$.

But of course that law is not enough to characterize randomness, since $010101\ldots$, say, satisfies it.

Von Mises' basic idea: A gambler should not be able to make any money on a random sequence.

If a gambler can determine a subsequence of α that violates the law of large numbers, then the gambler can make money on α in the long run, so α is not random.

Von Mises proposed that this observation could be turned around to characterize randomness.

Von Mises Randomness

A place selection rule is an increasing function $f : \mathbb{N} \to \mathbb{N}$, telling us which bits of a sequence to look at.

Let $R_n^f(\alpha) = \dfrac{|\{m < n : \alpha(f(m)) = 1\}|}{n}$.

Von Mises Randomness

A place selection rule is an increasing function $f : \mathbb{N} \to \mathbb{N}$, telling us which bits of a sequence to look at.

Let $R_n^f(\alpha) = \dfrac{|\{m < n : \alpha(f(m)) = 1\}|}{n}$.

$\alpha \in 2^\omega$ is von Mises random if $\lim_n R_n^f(\alpha) = \frac{1}{2}$ for all acceptable place selection rules.

Here "acceptable" means somehow given by a rule not depending on knowledge of α.

Von Mises Randomness

A place selection rule is an increasing function $f : \mathbb{N} \to \mathbb{N}$, telling us which bits of a sequence to look at.

Let $R_n^f(\alpha) = \dfrac{|\{m < n : \alpha(f(m)) = 1\}|}{n}$.

$\alpha \in 2^\omega$ is von Mises random if $\lim_n R_n^f(\alpha) = \frac{1}{2}$ for all acceptable place selection rules.

Here "acceptable" means somehow given by a rule not depending on knowledge of α.

Let $\mathcal{C}$ be a collection of place selection rules.

α is $\mathcal{C}$-von Mises random if $\lim_n R_n^f(\alpha) = \frac{1}{2}$ for all $f \in \mathcal{C}$.

Von Mises' Approach: The Good News

If acceptable place selection rules have to be finitely specified, then there should be only countably many of them.

Von Mises' Approach: The Good News

If acceptable place selection rules have to be finitely specified, then there should be only countably many of them.

Thm (Wald). Let $\mathcal{C}$ be any countable collection of place selection rules. Then $\mathcal{C}$-von Mises random sequences exist.

Von Mises' Approach: The Good News

If acceptable place selection rules have to be finitely specified, then there should be only countably many of them.

Thm (Wald). Let $\mathcal{C}$ be any countable collection of place selection rules. Then $\mathcal{C}$-von Mises random sequences exist.

Church suggested taking $\mathcal{C}$ to be the *computable* place selection rules.

Von Mises' Approach: The Bad News

Thm (Ville). Let $\mathcal{C}$ be any countable collection of place selection rules. There is a $\mathcal{C}$-von Mises random sequence α s.t. for all n,

$$R_n(\alpha) \geqslant \frac{1}{2}.$$

Such an α is clearly not random.

Von Mises' Approach: The Bad News

Thm (Ville). Let $\mathcal{C}$ be any countable collection of place selection rules. There is a $\mathcal{C}$-von Mises random sequence α s.t. for all n,

$$R_n(\alpha) \geqslant \frac{1}{2}.$$

Such an α is clearly not random.

Ville suggested adding another requirement for random sequences, the Law of the Iterated Logarithm.

Von Mises' Approach: The Bad News

Thm (Ville). Let $\mathcal{C}$ be any countable collection of place selection rules. There is a $\mathcal{C}$-von Mises random sequence α s.t. for all n,

$$R_n(\alpha) \geqslant \frac{1}{2}.$$

Such an α is clearly not random.

Ville suggested adding another requirement for random sequences, the Law of the Iterated Logarithm.

But how do we know this added requirement would be enough?

Three Approaches to Improving on von Mises' Idea

The statistician's approach: Define an abstract notion of reasonable statistical test, and define random sequences as those that pass all such tests.

Three Approaches to Improving on von Mises' Idea

The statistician's approach: Define an abstract notion of reasonable statistical test, and define random sequences as those that pass all such tests.

The coder's approach: Define an abstract notion of reasonable description, and define random sequences as those that have no simple descriptions.

Three Approaches to Improving on von Mises' Idea

The statistician's approach: Define an abstract notion of reasonable statistical test, and define random sequences as those that pass all such tests.

The coder's approach: Define an abstract notion of reasonable description, and define random sequences as those that have no simple descriptions.

The gambler's approach: Broaden von Mises' notion of betting, and require random sequences to be immune to every reasonable betting strategy.

Three Approaches to Improving on von Mises' Idea

The statistician's approach: Define an abstract notion of reasonable statistical test, and define random sequences as those that pass all such tests.

The coder's approach: Define an abstract notion of reasonable description, and define random sequences as those that have no simple descriptions.

The gambler's approach: Broaden von Mises' notion of betting, and require random sequences to be immune to every reasonable betting strategy.

Problem: What should count as a statistical test, or a description, or a betting strategy?

Three Approaches to Improving on von Mises' Idea

The statistician's approach: Define an abstract notion of reasonable statistical test, and define random sequences as those that pass all such tests.

The coder's approach: Define an abstract notion of reasonable description, and define random sequences as those that have no simple descriptions.

The gambler's approach: Broaden von Mises' notion of betting, and require random sequences to be immune to every reasonable betting strategy.

Problem: What should count as a statistical test, or a description, or a betting strategy?

Common solution: Use computability theory to define robust classes of tests, description systems, and betting strategies.

Part 1: Three Approaches to Defining Randomness

Computability Theory

A First Look at Randomness

The Statistician's Approach: Martin-Löf Randomness

The Coder's Approach: Kolmogorov complexity

The Gambler's Approach: Martingales

Cantor Space and Effectively Open Sets

We work in Cantor space 2^ω.

For $\sigma \in 2^{<\omega}$, let $[\sigma] = \{\alpha \in 2^\omega : \sigma \prec \alpha\}$.

2^ω is a topological space with basis $\{[\sigma] : \sigma \in 2^{<\omega}\}$.

The uniform measure on 2^ω is given by $\mu([\sigma]) = 2^{-|\sigma|}$.

Cantor Space and Effectively Open Sets

We work in Cantor space 2^ω.

For $\sigma \in 2^{<\omega}$, let $[\sigma] = \{\alpha \in 2^\omega : \sigma \prec \alpha\}$.

2^ω is a topological space with basis $\{[\sigma] : \sigma \in 2^{<\omega}\}$.

The uniform measure on 2^ω is given by $\mu([\sigma]) = 2^{-|\sigma|}$.

For $B \subseteq 2^{<\omega}$, let $[B] = \bigcup_{\sigma \in B} [\sigma]$. Every open set in 2^ω is of this form.

We call B a set of generators for $[B]$.

Cantor Space and Effectively Open Sets

We work in Cantor space 2^ω.

For $\sigma \in 2^{<\omega}$, let $[\sigma] = \{\alpha \in 2^\omega : \sigma \prec \alpha\}$.

2^ω is a topological space with basis $\{[\sigma] : \sigma \in 2^{<\omega}\}$.

The uniform measure on 2^ω is given by $\mu([\sigma]) = 2^{-|\sigma|}$.

For $B \subseteq 2^{<\omega}$, let $[B] = \bigcup_{\sigma \in B} [\sigma]$. Every open set in 2^ω is of this form.

We call B a set of generators for $[B]$.

A Σ^0_1 class is a set of the form $[B]$ for a c.e. $B \subseteq 2^{<\omega}$.

Cantor Space and Effectively Open Sets

We work in Cantor space 2^ω.

For $\sigma \in 2^{<\omega}$, let $[\sigma] = \{\alpha \in 2^\omega : \sigma \prec \alpha\}$.

2^ω is a topological space with basis $\{[\sigma] : \sigma \in 2^{<\omega}\}$.

The uniform measure on 2^ω is given by $\mu([\sigma]) = 2^{-|\sigma|}$.

For $B \subseteq 2^{<\omega}$, let $[B] = \bigcup_{\sigma \in B}[\sigma]$. Every open set in 2^ω is of this form.

We call B a set of generators for $[B]$.

A Σ^0_1 class is a set of the form $[B]$ for a c.e. $B \subseteq 2^{<\omega}$.

Equivalently, a Σ^0_1 class is a set of the form $[B]$ for a computable $B \subseteq 2^{<\omega}$.

Cantor Space and Effectively Open Sets

We work in Cantor space 2^ω.

For $\sigma \in 2^{<\omega}$, let $[\sigma] = \{\alpha \in 2^\omega : \sigma \prec \alpha\}$.

2^ω is a topological space with basis $\{[\sigma] : \sigma \in 2^{<\omega}\}$.

The uniform measure on 2^ω is given by $\mu([\sigma]) = 2^{-|\sigma|}$.

For $B \subseteq 2^{<\omega}$, let $[B] = \bigcup_{\sigma \in B}[\sigma]$. Every open set in 2^ω is of this form.

We call B a set of generators for $[B]$.

A Σ^0_1 class is a set of the form $[B]$ for a c.e. $B \subseteq 2^{<\omega}$.

Equivalently, a Σ^0_1 class is a set of the form $[B]$ for a computable $B \subseteq 2^{<\omega}$.

$\mathcal{C}_0, \mathcal{C}_1, \ldots$ are uniformly Σ^0_1 classes if $\mathcal{C}_n = [B_n]$ for uniformly c.e. $B_0, B_1, \ldots$

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

We can assume without loss of generality that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots$.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

We can assume without loss of generality that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots$.

We call any subset of $\bigcap_n \mathcal{C}_n$ Martin-Löf null.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \dots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

We can assume without loss of generality that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots$.

We call any subset of $\bigcap_n \mathcal{C}_n$ Martin-Löf null.

$\alpha \in 2^\omega$ passes this test if $\alpha \notin \bigcap_n \mathcal{C}_n$.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

We can assume without loss of generality that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots$.

We call any subset of $\bigcap_n \mathcal{C}_n$ Martin-Löf null.

$\alpha \in 2^\omega$ passes this test if $\alpha \notin \bigcap_n \mathcal{C}_n$.

α is Martin-Löf random, or 1-random, if it passes every Martin-Löf test.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

We can assume without loss of generality that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots$.

We call any subset of $\bigcap_n \mathcal{C}_n$ Martin-Löf null.

$\alpha \in 2^\omega$ passes this test if $\alpha \notin \bigcap_n \mathcal{C}_n$.

α is Martin-Löf random, or 1-random, if it passes every Martin-Löf test.

There are countably many ML-tests, each passed by all but measure 0 many sequences, so there are measure 1 many 1-random sequences.

Martin-Löf Randomness

A Martin-Löf test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \dots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

What really matters is that the measures tend effectively to 0.

We can assume without loss of generality that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots$.

We call any subset of $\bigcap_n \mathcal{C}_n$ Martin-Löf null.

$\alpha \in 2^\omega$ passes this test if $\alpha \notin \bigcap_n \mathcal{C}_n$.

α is Martin-Löf random, or 1-random, if it passes every Martin-Löf test.

There are countably many ML-tests, each passed by all but measure 0 many sequences, so there are measure 1 many 1-random sequences.

No computable sequence can be 1-random.

Universal Martin-Löf Tests

We can list all ML-tests as

$$\begin{array}{l} \mathcal{C}^0_0, \mathcal{C}^0_1, \mathcal{C}^0_2 \cdots \\ \mathcal{C}^1_0, \mathcal{C}^1_1, \mathcal{C}^1_2 \cdots \\ \mathcal{C}^2_0, \mathcal{C}^2_1, \mathcal{C}^2_2 \cdots \\ \vdots \end{array}$$

s.t. the whole collection $\{\mathcal{C}^i_n : i, n \in \mathbb{N}\}$ is uniformly Σ^0_1.

Universal Martin-Löf Tests

We can list all ML-tests as

$$
\begin{array}{l}
\mathcal{C}_0^0, \mathcal{C}_1^0, \mathcal{C}_2^0 \cdots \\
\mathcal{C}_0^1, \mathcal{C}_1^1, \mathcal{C}_2^1 \cdots \\
\mathcal{C}_0^2, \mathcal{C}_1^2, \mathcal{C}_2^2 \cdots \\
\vdots
\end{array}
$$

s.t. the whole collection $\{\mathcal{C}_n^i : i, n \in \mathbb{N}\}$ is uniformly Σ_1^0.

Let $\mathcal{U}_n = \bigcup_i \mathcal{C}_{i+n+1}^i$.

Universal Martin-Löf Tests

We can list all ML-tests as

$$\begin{array}{l} \mathcal{C}_0^0, \mathcal{C}_1^0, \mathcal{C}_2^0 \cdots \\ \mathcal{C}_0^1, \mathcal{C}_1^1, \mathcal{C}_2^1 \cdots \\ \mathcal{C}_0^2, \mathcal{C}_1^2, \mathcal{C}_2^2 \cdots \\ \vdots \end{array}$$

s.t. the whole collection $\{\mathcal{C}_n^i : i, n \in \mathbb{N}\}$ is uniformly Σ_1^0.

Let $\mathcal{U}_n = \bigcup_i \mathcal{C}_{i+n+1}^i$.

Then $\mathcal{U}_0, \mathcal{U}_1, \ldots$ is a ML-test, and α is 1-random iff it passes this single test.

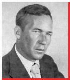
We call $\mathcal{U}_0, \mathcal{U}_1, \ldots$ a universal Martin-Löf test.

Part 1: Three Approaches to Defining Randomness

Computability Theory

A First Look at Randomness

The Statistician's Approach: Martin-Löf Randomness



The Coder's Approach: Kolmogorov complexity

The Gambler's Approach: Martingales

Kolmogorov Complexity

Although 0^{10000} is much longer than
0101011101101111011100100110101101110011011010000110111110111,
it is easier to describe.

Intuitively, the Kolmogorov complexity of an object is its shortest description.

Kolmogorov Complexity

Although 0^{10000} is much longer than 010101110110111101110010011010110111001101101000011011110111, it is easier to describe.

Intuitively, the Kolmogorov complexity of an object is its shortest description.

But what counts as a description?

Berry's Paradox: The smallest natural number that cannot be described in fewer than twenty words.

Kolmogorov Complexity

Although 0^{10000} is much longer than
010101110110111101110010011010110111001101101000011011110111,
it is easier to describe.

Intuitively, the Kolmogorov complexity of an object is its shortest description.

But what counts as a description?

Berry's Paradox: The smallest natural number that cannot be described in fewer than twenty words.

The idea is to think of partial computable functions as systems of descriptions.

Plain Kolmogorov Complexity

Let $f : 2^{<\omega} \to 2^{<\omega}$ be partial computable.

The Kolmogorov complexity of σ relative to f is

$$C_f(\sigma) = \min\{|\tau| : f(\tau) = \sigma\}.$$

Plain Kolmogorov Complexity

Let $f : 2^{<\omega} \to 2^{<\omega}$ be partial computable.

The Kolmogorov complexity of σ relative to f is

$$C_f(\sigma) = \min\{|\tau| : f(\tau) = \sigma\}.$$

$C_f(\sigma)$ depends on f, but there is a "best" choice of f: Let f be a universal partial computable function.

The plain Kolmogorov complexity of σ is $C(\sigma) = C_f(\sigma)$.

Plain Kolmogorov Complexity

Let $f : 2^{<\omega} \to 2^{<\omega}$ be partial computable.

The Kolmogorov complexity of σ relative to f is

$$C_f(\sigma) = \min\{|\tau| : f(\tau) = \sigma\}.$$

$C_f(\sigma)$ depends on f, but there is a "best" choice of f: Let f be a universal partial computable function.

The plain Kolmogorov complexity of σ is $C(\sigma) = C_f(\sigma)$.

For every partial computable g, we have $C(\sigma) \leqslant C_g(\sigma) + O(1)$.

In particular, if f and g are both universal partial computable functions, then $C_f(\sigma) = C_g(\sigma) \pm O(1)$, so the definition of C does not depend on the choice of f, up to an additive constant.

Random Strings

There are 2^n strings of length n, but only $2^n - 1$ descriptions of length $< n$.

So there exist σ s.t. $C(\sigma) \geqslant |\sigma|$.

Random Strings

There are 2^n strings of length n, but only $2^n - 1$ descriptions of length $< n$.

So there exist σ s.t. $C(\sigma) \geqslant |\sigma|$.

Such σ are incompressible, and it makes sense to consider them random.

Random Strings

There are 2^n strings of length n, but only $2^n - 1$ descriptions of length $< n$.

So there exist σ s.t. $C(\sigma) \geqslant |\sigma|$.

Such σ are incompressible, and it makes sense to consider them random.

We might expect every initial segment of a random sequence to be random, and indeed want to characterize randomness of α by

$$C(\alpha \upharpoonright n) \geqslant n - O(1).$$

Random Strings

There are 2^n strings of length n, but only $2^n - 1$ descriptions of length $< n$.

So there exist σ s.t. $C(\sigma) \geqslant |\sigma|$.

Such σ are incompressible, and it makes sense to consider them random.

We might expect every initial segment of a random sequence to be random, and indeed want to characterize randomness of α by

$$C(\alpha \restriction n) \geqslant n - O(1).$$

However:

Thm (Martin-Löf). There is no $\alpha \in 2^\omega$ s.t. $C(\alpha \restriction n) \geqslant n - O(1)$.

A Criticism of Plain Kolmogorov Complexity

The length of a string represents additional information beyond that contained in the bits of the string.

Even $000000\ldots$ has initial segments with moderately high information content: those of the form 0^n where n has high information content.

A Criticism of Plain Kolmogorov Complexity

The length of a string represents additional information beyond that contained in the bits of the string.

Even $000000\ldots$ has initial segments with moderately high information content: those of the form 0^n where n has high information content.

Put another way, to describe binary strings, we use binary strings *plus* termination information.

A Criticism of Plain Kolmogorov Complexity

The length of a string represents additional information beyond that contained in the bits of the string.

Even $000000\ldots$ has initial segments with moderately high information content: those of the form 0^n where n has high information content.

Put another way, to describe binary strings, we use binary strings *plus* termination information.

A partial function $f : 2^{<\omega} \to 2^{<\omega}$ is prefix-free if its domain is an antichain, that is, if $f(\sigma)\downarrow$ and $\sigma \prec \tau$ or $\tau \prec \sigma$, then $f(\tau)\uparrow$.

Using only prefix-free partial computable functions as description systems gets around the above criticism.

Prefix-free Kolmogorov Complexity

List the prefix-free partial computable functions $f_0, f_1, \ldots$ and let

$$U(0^e 1\sigma) = f_e(\sigma).$$

Then U is a universal prefix-free partial computable function

Prefix-free Kolmogorov Complexity

List the prefix-free partial computable functions $f_0, f_1, \ldots$ and let

$$U(0^e 1\sigma) = f_e(\sigma).$$

Then U is a universal prefix-free partial computable function

The prefix-free Kolmogorov complexity of σ is

$$K(\sigma) = C_U(\sigma) = \min\{|\tau| : U(\tau) = \sigma\}.$$

Prefix-free Kolmogorov Complexity

List the prefix-free partial computable functions $f_0, f_1, \dots$ and let

$$U(0^e 1\sigma) = f_e(\sigma).$$

Then U is a universal prefix-free partial computable function

The prefix-free Kolmogorov complexity of σ is

$$K(\sigma) = C_U(\sigma) = \min\{|\tau| : U(\tau) = \sigma\}.$$

As with C, the choice of universal U does not matter up to a constant.

Prefix-free Kolmogorov Complexity

List the prefix-free partial computable functions $f_0, f_1, \ldots$ and let

$$U(0^e 1\sigma) = f_e(\sigma).$$

Then U is a universal prefix-free partial computable function

The prefix-free Kolmogorov complexity of σ is

$$K(\sigma) = C_U(\sigma) = \min\{|\tau| : U(\tau) = \sigma\}.$$

As with C, the choice of universal U does not matter up to a constant.

K is not computable, but it is computably approximable from above, i.e., there is a computable $g : 2^{<\omega} \times \mathbb{N} \to \mathbb{N}$ s.t. $g(\sigma, n) \geqslant g(\sigma, n+1)$ and $\lim_n g(\sigma, n) = K(\sigma)$.

Prefix-Free Sets of Generators

Every open set $\mathcal{C}$ can be written as $[B]$ for some prefix-free $B \subset 2^{<\omega}$.

If $\mathcal{C}$ is a Σ_1^0 class then B can be chosen to be c.e.

Prefix-Free Sets of Generators

Every open set $\mathcal{C}$ can be written as $[B]$ for some prefix-free $B \subset 2^{<\omega}$.

If $\mathcal{C}$ is a Σ_1^0 class then B can be chosen to be c.e.

In any case, $\mu(\mathcal{C}) = \sum_{\sigma \in B} 2^{-|\sigma|}$.

So for any prefix free set B, we have $\sum_{\sigma \in B} 2^{-|\sigma|} \leqslant 1$.

Prefix-Free Sets of Generators

Every open set $\mathcal{C}$ can be written as $[B]$ for some prefix-free $B \subset 2^{<\omega}$.

If $\mathcal{C}$ is a Σ^0_1 class then B can be chosen to be c.e.

In any case, $\mu(\mathcal{C}) = \sum_{\sigma \in B} 2^{-|\sigma|}$.

So for any prefix free set B, we have $\sum_{\sigma \in B} 2^{-|\sigma|} \leqslant 1$.

In particular, for each σ, let σ^* be a minimal length string s.t. $U(\sigma^*) = \sigma$.

Then $\sum_{\sigma} 2^{-K(\sigma)} = \sum_{\sigma} 2^{-|\sigma^*|} \leqslant \sum_{\tau \in \mathrm{dom}\, U} 2^{-|\tau|} \leqslant 1$.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \restriction n) \geqslant n - O(1)$.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \restriction n) \geqslant n - O(1)$.

Proof of the $\Rightarrow$ direction. Suppose that $\forall i \, \exists n \, (K(\alpha \restriction n) < n - i)$.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Rightarrow$ direction. Suppose that $\forall i \, \exists n \, (K(\alpha \upharpoonright n) < n - i)$.

Let $\mathcal{C}_i = \bigcup\{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \restriction n) \geqslant n - O(1)$.

Proof of the $\Rightarrow$ direction. Suppose that $\forall i \, \exists n \, (K(\alpha \restriction n) < n - i)$.

Let $\mathcal{C}_i = \bigcup\{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

$\mathcal{C}_0, \mathcal{C}_1, \ldots$ are uniformly Σ^0_1 classes.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Rightarrow$ direction. Suppose that $\forall i\, \exists n\, (K(\alpha \upharpoonright n) < n - i)$.

Let $\mathcal{C}_i = \bigcup\{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

$\mathcal{C}_0, \mathcal{C}_1, \ldots$ are uniformly Σ^0_1 classes.

Let $\sigma_0, \sigma_1, \ldots$ be a prefix-free set of generators for $\mathcal{C}_i$.

Then $1 \geqslant \sum_j 2^{-K(\sigma_j)} \geqslant \sum_j 2^{-(|\sigma_j| - i)} = 2^i \sum_j 2^{-|\sigma_j|} = 2^i \mu(\mathcal{C}_i)$.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Rightarrow$ direction. Suppose that $\forall i \, \exists n \, (K(\alpha \upharpoonright n) < n - i)$.

Let $\mathcal{C}_i = \bigcup\{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

$\mathcal{C}_0, \mathcal{C}_1, \ldots$ are uniformly Σ^0_1 classes.

Let $\sigma_0, \sigma_1, \ldots$ be a prefix-free set of generators for $\mathcal{C}_i$.

Then $1 \geqslant \sum_j 2^{-K(\sigma_j)} \geqslant \sum_j 2^{-(|\sigma_j| - i)} = 2^i \sum_j 2^{-|\sigma_j|} = 2^i \mu(\mathcal{C}_i)$.

So $\mu(\mathcal{C}_i) \leqslant 2^{-i}$, and hence $\mathcal{C}_0, \mathcal{C}_1, \ldots$ is an ML-test.

1-randomness via Kolmogorov Complexity

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Rightarrow$ direction. Suppose that $\forall i\, \exists n\, (K(\alpha \upharpoonright n) < n - i)$.

Let $\mathcal{C}_i = \bigcup\{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

$\mathcal{C}_0, \mathcal{C}_1, \dots$ are uniformly Σ^0_1 classes.

Let $\sigma_0, \sigma_1, \dots$ be a prefix-free set of generators for $\mathcal{C}_i$.

Then $1 \geqslant \sum_j 2^{-K(\sigma_j)} \geqslant \sum_j 2^{-(|\sigma_j| - i)} = 2^i \sum_j 2^{-|\sigma_j|} = 2^i \mu(\mathcal{C}_i)$.

So $\mu(\mathcal{C}_i) \leqslant 2^{-i}$, and hence $\mathcal{C}_0, \mathcal{C}_1, \dots$ is an ML-test.

Since $\alpha \in \bigcap_i \mathcal{C}_i$, we see that α is not 1-random. □

1-randomness via Kolmogorov Complexity

Proof of the $\Rightarrow$ direction. Suppose that $\forall i \, \exists n \, (K(\alpha \upharpoonright n) < n - i)$.

Let $\mathcal{C}_i = \bigcup\{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

$\mathcal{C}_0, \mathcal{C}_i, \ldots$ are uniformly Σ^0_1 classes.

Let $\sigma_0, \sigma_1, \ldots$ be a prefix-free set of generators for $\mathcal{C}_i$.

Then $1 \geqslant \sum_j 2^{-K(\sigma_j)} \geqslant \sum_j 2^{-(|\sigma_j| - i)} = 2^i \sum_j 2^{-|\sigma_j|} = 2^i \mu(\mathcal{C}_i)$.

So $\mu(\mathcal{C}_i) \leqslant 2^{-i}$, and hence $\mathcal{C}_0, \mathcal{C}_i, \ldots$ is an ML-test.

Since $\alpha \in \bigcap_i \mathcal{C}_i$, we see that α is not 1-random. $\square$

1-randomness via Kolmogorov Complexity

Proof of the $\Rightarrow$ direction. Suppose that $\forall i \, \exists n \, (K(\alpha \upharpoonright n) < n - i)$.

Let $\mathcal{C}_i = \bigcup \{[\sigma] : K(\sigma) < |\sigma| - i\}$. Note that $\alpha \in \mathcal{C}_i$ for all i.

$\mathcal{C}_0, \mathcal{C}_i, \ldots$ are uniformly Σ^0_1 classes.

Let $\sigma_0, \sigma_1, \ldots$ be a prefix-free set of generators for $\mathcal{C}_i$.

Then $1 \geqslant \sum_j 2^{-K(\sigma_j)} \geqslant \sum_j 2^{-(|\sigma_j| - i)} = 2^i \sum_j 2^{-|\sigma_j|} = 2^i \mu(\mathcal{C}_i)$.

So $\mu(\mathcal{C}_i) \leqslant 2^{-i}$, and hence $\mathcal{C}_0, \mathcal{C}_i, \ldots$ is an ML-test.

Since $\alpha \in \bigcap_i \mathcal{C}_i$, we see that α is not 1-random. □

In fact, $\mathcal{C}_0, \mathcal{C}_1, \ldots$ is a universal ML-test.

Bespoke Description Systems

For the other direction of Schnorr's Theorem, we need the following result.

KC Thm. Let $\langle n_i, \sigma_i \rangle_{i \in \mathbb{N}}$ be a computable sequence s.t. $\sum_i 2^{-n_i} \leqslant 1$.

There is a prefix-free partial computable f s.t.

$$\forall i \, \exists \tau_i \, (|\tau_i| = n_i \, \wedge \, f(\tau_i) = \sigma_i).$$

Then $C_f(\sigma_i) \leqslant n_i$, whence $K(\sigma_i) \leqslant n_i + O(1)$.

Bespoke Description Systems

For the other direction of Schnorr's Theorem, we need the following result.

KC Thm. Let $\langle n_i, \sigma_i \rangle_{i \in \mathbb{N}}$ be a computable sequence s.t. $\sum_i 2^{-n_i} \leqslant 1$.

There is a prefix-free partial computable f s.t.

$$\forall i \, \exists \tau_i \, (|\tau_i| = n_i \, \wedge \, f(\tau_i) = \sigma_i).$$

Then $C_f(\sigma_i) \leqslant n_i$, whence $K(\sigma_i) \leqslant n_i + O(1)$.

The proof is a little messy, but f is easy to specify:

For each i, let τ_i be the leftmost string of length n_i incomparable with every τ_j for $j < i$, and let $f(\tau_i) = \sigma_i$.

1-randomness via Kolmogorov Complexity Revisited

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Leftarrow$ direction. Let $\mathcal{U}_0, \mathcal{U}_1, \ldots$ be a universal ML-test.

There are uniformly c.e. sets $\{\sigma^0_0, \sigma^0_1, \ldots\}$, $\{\sigma^1_1, \sigma^1_1, \ldots\}$, $\ldots$ s.t. $\{\sigma^i_0, \sigma^i_1, \ldots\}$ is a prefix-free set of generators for $\mathcal{U}_i$.

1-randomness via Kolmogorov Complexity Revisited

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Leftarrow$ direction. Let $\mathcal{U}_0, \mathcal{U}_1, \ldots$ be a universal ML-test.

There are uniformly c.e. sets $\{\sigma_0^0, \sigma_1^0, \ldots\}$, $\{\sigma_1^1, \sigma_1^1, \ldots\}$, ... s.t. $\{\sigma_0^i, \sigma_1^i, \ldots\}$ is a prefix-free set of generators for $\mathcal{U}_i$.

Consider the set of requests $\langle |\sigma_j^{2i+1}| - i, \sigma_j^{2i+1} \rangle$ for all i and j.

1-randomness via Kolmogorov Complexity Revisited

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Leftarrow$ direction. Let $\mathcal{U}_0, \mathcal{U}_1, \ldots$ be a universal ML-test.

There are uniformly c.e. sets $\{\sigma_0^0, \sigma_1^0, \ldots\}$, $\{\sigma_1^1, \sigma_1^1, \ldots\}$, $\ldots$ s.t. $\{\sigma_0^i, \sigma_1^i, \ldots\}$ is a prefix-free set of generators for $\mathcal{U}_i$.

Consider the set of requests $\langle |\sigma_j^{2i+1}| - i, \sigma_j^{2i+1} \rangle$ for all i and j.

$$\sum_{i,j} 2^{-(|\sigma_j^{2i+1}|-i)} = \sum_i 2^i \sum_j 2^{-|\sigma_j^{2i+1}|} = \sum_i 2^i \mu(\mathcal{U}_{2i+1}) \leqslant$$
$$\sum_i 2^i 2^{-(2i+1)} = \sum_i 2^{-(i+1)} = 1.$$

1-randomness via Kolmogorov Complexity Revisited

Thm (Schnorr). $\alpha \in 2^\omega$ is 1-random iff $K(\alpha \upharpoonright n) \geqslant n - O(1)$.

Proof of the $\Leftarrow$ direction. Let $\mathcal{U}_0, \mathcal{U}_1, \ldots$ be a universal ML-test.

There are uniformly c.e. sets $\{\sigma_0^0, \sigma_1^0, \ldots\}$, $\{\sigma_1^1, \sigma_1^1, \ldots\}$, $\ldots$ s.t. $\{\sigma_0^i, \sigma_1^i, \ldots\}$ is a prefix-free set of generators for $\mathcal{U}_i$.

Consider the set of requests $\langle |\sigma_j^{2i+1}| - i, \sigma_j^{2i+1} \rangle$ for all i and j.

$$\textstyle\sum_{i,j} 2^{-(|\sigma_j^{2i+1}|-i)} = \sum_i 2^i \sum_j 2^{-|\sigma_j^{2i+1}|} = \sum_i 2^i \mu(\mathcal{U}_{2i+1}) \leqslant$$
$$\textstyle\sum_i 2^i 2^{-(2i+1)} = \sum_i 2^{-(i+1)} = 1.$$

So by the KC Thm, $K(\sigma_j^{2i+1}) \leqslant |\sigma_j^{2i+1}| - i + O(1)$ for all i and j.

1-randomness via Kolmogorov Complexity Revisited

Proof of the $\Leftarrow$ *direction.* Let $\mathcal{U}_0, \mathcal{U}_1, \ldots$ be a universal ML-test.

There are uniformly c.e. sets $\{\sigma_0^0, \sigma_1^0, \ldots\}$, $\{\sigma_1^1, \sigma_1^1, \ldots\}$, $\ldots$ s.t. $\{\sigma_0^i, \sigma_1^i, \ldots\}$ is a prefix-free set of generators for $\mathcal{U}_i$.

Consider the set of requests $\langle |\sigma_j^{2i+1}| - i, \sigma_j^{2i+1} \rangle$ for all i and j.

$$\sum_{i,j} 2^{-(|\sigma_j^{2i+1}|-i)} = \sum_i 2^i \sum_j 2^{-|\sigma_j^{2i+1}|} = \sum_i 2^i \mu(\mathcal{U}_{2i+1}) \leqslant$$
$$\sum_i 2^i 2^{-(2i+1)} = \sum_i 2^{-(i+1)} = 1.$$

So by the KC Thm, $K(\sigma_j^{2i+1}) \leqslant |\sigma_j^{2i+1}| - i + O(1)$ for all i and j.

Proof of the $\Leftarrow$ direction. Let $\mathcal{U}_0, \mathcal{U}_1, \dots$ be a universal ML-test.

There are uniformly c.e. sets $\{\sigma_0^0, \sigma_1^0, \dots\}$, $\{\sigma_1^1, \sigma_1^1, \dots\}$, $\dots$ s.t. $\{\sigma_0^i, \sigma_1^i, \dots\}$ is a prefix-free set of generators for $\mathcal{U}_i$.

Consider the set of requests $\langle |\sigma_j^{2i+1}| - i, \sigma_j^{2i+1} \rangle$ for all i and j.

$$\sum_{i,j} 2^{-(|\sigma_j^{2i+1}|-i)} = \sum_i 2^i \sum_j 2^{-|\sigma_j^{2i+1}|} = \sum_i 2^i \mu(\mathcal{U}_{2i+1}) \leqslant$$
$$\sum_i 2^i 2^{-(2i+1)} = \sum_i 2^{-(i+1)} = 1.$$

So by the KC Thm, $K(\sigma_j^{2i+1}) \leqslant |\sigma_j^{2i+1}| - i + O(1)$ for all i and j.

If α is not 1-random, then $\alpha \in \mathcal{U}_{2i+1}$ for all i, so $\forall i \, \exists j, n \, (\sigma_j^{2i+1} = \alpha \upharpoonright n)$.

Thus $\forall i \, \exists n \, (K(\alpha \upharpoonright n) \leqslant n - i)$. $\square$

Part 1: Three Approaches to Defining Randomness

Computability Theory

A First Look at Randomness

The Statistician's Approach: Martin-Löf Randomness

The Coder's Approach: Kolmogorov complexity

The Gambler's Approach: Martingales

Martingales and Supermartingales

A martingale is a function $d : 2^{<\omega} \to \mathbb{R}^{\geqslant 0}$ s.t.

$$\frac{d(\sigma 0) + d(\sigma 1)}{2} = d(\sigma).$$

Martingales and Supermartingales

A martingale is a function $d : 2^{<\omega} \to \mathbb{R}^{\geqslant 0}$ s.t.

$$\frac{d(\sigma 0) + d(\sigma 1)}{2} = d(\sigma).$$

The martingale d succeeds on α if $\limsup_n d(\alpha \upharpoonright n) = \infty$.

The success set S_d of d is the set of all sequences on which d succeeds.

Martingales and Supermartingales

A martingale is a function $d : 2^{<\omega} \to \mathbb{R}^{\geqslant 0}$ s.t.

$$\frac{d(\sigma 0) + d(\sigma 1)}{2} = d(\sigma).$$

The martingale d succeeds on α if $\limsup_n d(\alpha \upharpoonright n) = \infty$.

The success set S_d of d is the set of all sequences on which d succeeds.

We can replace d by a closely related martingale $\hat{d}$ s.t. $S_{\hat{d}} = S_d$ and $\liminf_n d(\alpha \upharpoonright n) = \infty$ for all $\alpha \in S_{\hat{d}}$.

Martingales and Supermartingales

A martingale is a function $d : 2^{<\omega} \to \mathbb{R}^{\geqslant 0}$ s.t.

$$\frac{d(\sigma 0) + d(\sigma 1)}{2} = d(\sigma).$$

The martingale d succeeds on α if $\limsup_n d(\alpha \upharpoonright n) = \infty$.

The success set S_d of d is the set of all sequences on which d succeeds.

We can replace d by a closely related martingale $\hat{d}$ s.t. $S_{\hat{d}} = S_d$ and $\liminf_n d(\alpha \upharpoonright n) = \infty$ for all $\alpha \in S_{\hat{d}}$.

A supermartingale is a function $d : 2^{<\omega} \to \mathbb{R}^{\geqslant 0}$ s.t.

$$\frac{d(\sigma 0) + d(\sigma 1)}{2} \leqslant d(\sigma).$$

An Example

Recall that $R_n(\alpha) = \frac{|\{m < n \,:\, \alpha(m) = 1\}|}{n}$.

Suppose that $\liminf_n R_n(\alpha) > \frac{2}{3}$.

An Example

Recall that $R_n(\alpha) = \dfrac{|\{m < n : \alpha(m) = 1\}|}{n}$.

Suppose that $\liminf_n R_n(\alpha) > \frac{2}{3}$.

Let $d(\lambda) = 1$, where λ is the empty sequence.

Given $d(\sigma)$, let $d(\sigma 0) = \frac{d(\sigma)}{2}$ and $d(\sigma 1) = \frac{3d(\sigma)}{2}$.

An Example

Recall that $R_n(\alpha) = \frac{|\{m < n \,:\, \alpha(m) = 1\}|}{n}$.

Suppose that $\liminf_n R_n(\alpha) > \frac{2}{3}$.

Let $d(\lambda) = 1$, where λ is the empty sequence.

Given $d(\sigma)$, let $d(\sigma 0) = \frac{d(\sigma)}{2}$ and $d(\sigma 1) = \frac{3d(\sigma)}{2}$.

Then

$$d(\alpha \upharpoonright n) = \left(\tfrac{1}{2}\right)^{n - nR_n(\alpha)} \left(\tfrac{3}{2}\right)^{nR_n(\alpha)} \geqslant O\left(\left(\tfrac{1}{2}\right)^{\frac{n}{3}} \left(\tfrac{3}{2}\right)^{\frac{2n}{3}}\right) = O\left(\left(\tfrac{1}{2}\tfrac{9}{4}\right)^{\frac{n}{3}}\right) = O\left(\left(\tfrac{9}{8}\right)^{\frac{n}{3}}\right).$$

An Example

Recall that $R_n(\alpha) = \dfrac{|\{m < n \,:\, \alpha(m) = 1\}|}{n}$.

Suppose that $\liminf_n R_n(\alpha) > \frac{2}{3}$.

Let $d(\lambda) = 1$, where λ is the empty sequence.

Given $d(\sigma)$, let $d(\sigma 0) = \frac{d(\sigma)}{2}$ and $d(\sigma 1) = \frac{3d(\sigma)}{2}$.

Then

$$d(\alpha \restriction n) = \left(\tfrac{1}{2}\right)^{n - nR_n(\alpha)} \left(\tfrac{3}{2}\right)^{nR_n(\alpha)} \geqslant O\left(\left(\tfrac{1}{2}\right)^{\frac{n}{3}} \left(\tfrac{3}{2}\right)^{\frac{2n}{3}}\right) = O\left(\left(\tfrac{1}{2}\tfrac{9}{4}\right)^{\frac{n}{3}}\right) = O\left(\left(\tfrac{9}{8}\right)^{\frac{n}{3}}\right).$$

So $\lim_n d(\alpha \restriction n) = \infty$, and hence $\alpha \in S_d$.

Left-c.e. Reals and Functions

A real number x is left-c.e. if it can be computably approximated from below.

That is, there is a computable $f : \mathbb{N} \to \mathbb{Q}$ s.t. $f(n) \leqslant f(n+1)$ and $\lim_n f(n) = x$.

Left-c.e. Reals and Functions

A real number x is left-c.e. if it can be computably approximated from below.

That is, there is a computable $f : \mathbb{N} \to \mathbb{Q}$ s.t. $f(n) \leqslant f(n+1)$ and $\lim_n f(n) = x$.

Equivalently, x is left-c.e. if it is the measure of a Σ^0_1 class.

Equivalently, x is left-c.e. if it is $\sum_{f(\sigma)\downarrow} 2^{-|\sigma|}$ for a prefix-free partial computable f.

Left-c.e. Reals and Functions

A real number x is left-c.e. if it can be computably approximated from below.

That is, there is a computable $f : \mathbb{N} \to \mathbb{Q}$ s.t. $f(n) \leqslant f(n+1)$ and $\lim_n f(n) = x$.

Equivalently, x is left-c.e. if it is the measure of a Σ^0_1 class.

Equivalently, x is left-c.e. if it is $\sum_{f(\sigma)\downarrow} 2^{-|\sigma|}$ for a prefix-free partial computable f.

A function $d : 2^{<\omega} \to \mathbb{R}$ is left-c.e. if there is a computable $f : 2^{<\omega} \times \mathbb{N} \to \mathbb{Q}$ s.t. $f(\sigma, n) \leqslant f(\sigma, n+1)$ and $\lim_n f(\sigma, n) = d(\sigma)$.

In other words, the values $d(\sigma)$ are uniformly left-c.e.

1-randomness via Martingales

Thm (Schnorr). The following are equivalent.

$\alpha \in 2^\omega$ is 1-random.

No left-c.e. martingale succeeds on α.

No left-c.e. supermartingale succeeds on α.

1-randomness via Martingales

Thm (Schnorr). The following are equivalent.

$\alpha \in 2^\omega$ is 1-random.

No left-c.e. martingale succeeds on α.

No left-c.e. supermartingale succeeds on α.

There is a universal left-c.e. martingale, i.e., a left-c.e. martingale u s.t. for every left-c.e. martingale d, we have $S_d \subseteq S_u$.

1-randomness via Martingales

Thm (Schnorr). The following are equivalent.

$\alpha \in 2^\omega$ is 1-random.

No left-c.e. martingale succeeds on α.

No left-c.e. supermartingale succeeds on α.

There is a universal left-c.e. martingale, i.e., a left-c.e. martingale u s.t. for every left-c.e. martingale d, we have $S_d \subseteq S_u$.

Easier to see for supermartingales, because we can nicely list all left-c.e. supermartingales $d_0, d_1, \ldots$ and let

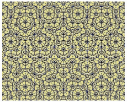
$$u(\sigma) = \sum_n 2^{-n} \frac{d_n(\sigma)}{d_n(\lambda)}.$$

Part 2: Examples, Properties, and Variations

Weakening 1-randomness

A Little More Computability Theory

Strengthening 1-randomness



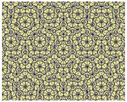
Highly Nonrandom Sequences

Part 2: Examples, Properties, and Variations

Weakening 1-randomness

A Little More Computability Theory

Strengthening 1-randomness



Highly Nonrandom Sequences

Schnorr's Critique

Schnorr pointed out that 1-randomness is a notion of *c.e. randomness*, rather than *computable randomness*.

Schnorr's Critique

Schnorr pointed out that 1-randomness is a notion of *c.e. randomness*, rather than *computable randomness*.

Recall that α is 1-random if no left-c.e. martingale succeeds on α.

A martingale d is computable if the values $d(\sigma)$ are uniformly computable.

α is computably random if no computable martingale succeeds on α.

Schnorr's Critique

Schnorr pointed out that 1-randomness is a notion of *c.e. randomness*, rather than *computable randomness*.

Recall that α is 1-random if no left-c.e. martingale succeeds on α.

A martingale d is computable if the values $d(\sigma)$ are uniformly computable.

α is computably random if no computable martingale succeeds on α.

Schnorr thought that computable randomness is not effective enough.

An order is an unbounded, nondecreasing computable $f : \mathbb{N} \to \mathbb{Q}^+$.

A martingale d succeeds f-fast on α if $d(\alpha \upharpoonright n) \geqslant f(n)$.

Schnorr's Critique

Schnorr pointed out that 1-randomness is a notion of *c.e. randomness*, rather than *computable randomness*.

Recall that α is 1-random if no left-c.e. martingale succeeds on α.

A martingale d is computable if the values $d(\sigma)$ are uniformly computable.

α is computably random if no computable martingale succeeds on α.

Schnorr thought that computable randomness is not effective enough.

An order is an unbounded, nondecreasing computable $f : \mathbb{N} \to \mathbb{Q}^+$.

A martingale d succeeds f-fast on α if $d(\alpha \upharpoonright n) \geqslant f(n)$.

α is Schnorr random if there is no computable martingale d and order f s.t. d succeeds f-fast on α.

Comparing Randomness Notions

Thm (Schnorr). α is Schnorr random iff it passes every ML-test $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. the $\mu(\mathcal{C}_n)$ are uniformly computable.

Comparing Randomness Notions

Thm (Schnorr). α is Schnorr random iff it passes every ML-test $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. the $\mu(\mathcal{C}_n)$ are uniformly computable.

1-randomness implies computable randomness implies Schnorr randomness.

Comparing Randomness Notions

Thm (Schnorr). α is Schnorr random iff it passes every ML-test $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. the $\mu(\mathcal{C}_n)$ are uniformly computable.

1-randomness implies computable randomness implies Schnorr randomness.

Thm (Schnorr). There are computably random sequences that are not 1-random.

Thm (Wang). There are Schnorr random sequences that are not computably random.

Nonmonotonic Randomness

A nonmonotonic betting strategy is one that, given α:

picks a bit n_0 and

bets some fraction p_0 of its initial capital on $\alpha(n_0) = 0$ and $1 - p_0$ of that capital on $\alpha(n_0) = 1$,

Nonmonotonic Randomness

A nonmonotonic betting strategy is one that, given α:

picks a bit n_0 and

bets some fraction p_0 of its initial capital on $\alpha(n_0) = 0$ and $1 - p_0$ of that capital on $\alpha(n_0) = 1$,

then based on the value $\alpha(n_0)$, picks a new bit n_1 and

bets some fraction p_1 of its remaining capital on $\alpha(n_1) = 0$ and $1 - p_1$ of that capital on $\alpha(n_1) = 1$,

Nonmonotonic Randomness

A nonmonotonic betting strategy is one that, given α:

picks a bit n_0 and

bets some fraction p_0 of its initial capital on $\alpha(n_0) = 0$ and $1 - p_0$ of that capital on $\alpha(n_0) = 1$,

then based on the value $\alpha(n_0)$, picks a new bit n_1 and

bets some fraction p_1 of its remaining capital on $\alpha(n_1) = 0$ and $1 - p_1$ of that capital on $\alpha(n_1) = 1$,

and so on.

Nonmonotonic Randomness

A nonmonotonic betting strategy is one that, given α:

picks a bit n_0 and

bets some fraction p_0 of its initial capital on $\alpha(n_0) = 0$ and $1 - p_0$ of that capital on $\alpha(n_0) = 1$,

then based on the value $\alpha(n_0)$, picks a new bit n_1 and

bets some fraction p_1 of its remaining capital on $\alpha(n_1) = 0$ and $1 - p_1$ of that capital on $\alpha(n_1) = 1$,

and so on.

This concept can be formalized using a nonmonotonic version of martingales.

Nonmonotonic Randomness

A nonmonotonic betting strategy is one that, given α:

picks a bit n_0 and

bets some fraction p_0 of its initial capital on $\alpha(n_0) = 0$ and $1 - p_0$ of that capital on $\alpha(n_0) = 1$,

then based on the value $\alpha(n_0)$, picks a new bit n_1 and

bets some fraction p_1 of its remaining capital on $\alpha(n_1) = 0$ and $1 - p_1$ of that capital on $\alpha(n_1) = 1$,

and so on.

This concept can be formalized using a nonmonotonic version of martingales.

α is nonmonotonically random if no computable nonmonotonic betting strategy makes arbitrarily much money betting on α.

A Fundamental Open Question

Nonmonotonic randomness implies computable randomness.

A Fundamental Open Question

Nonmonotonic randomness implies computable randomness.

Thm (Muchnik, Semenov, and Uspensky). There are computably random sequences that are not nonmonotonically random.

Every 1-random sequence is nonmonotonically random.

A Fundamental Open Question

Nonmonotonic randomness implies computable randomness.

Thm (Muchnik, Semenov, and Uspensky). There are computably random sequences that are not nonmonotonically random.

Every 1-random sequence is nonmonotonically random.

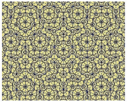
Open Question. Is every nonmonotonically random sequence 1-random?

Part 2: Examples, Properties, and Variations

Weakening 1-randomness

A Little More Computability Theory

Strengthening 1-randomness



Highly Nonrandom Sequences

Relative Computability

f is g-computable if there is an algorithm for computing f using information from g. We write $f \leqslant_{\mathbf{T}} g$.

Relative Computability

f is g-computable if there is an algorithm for computing f using information from g. We write $f \leqslant_{\mathbf{T}} g$.

Example: Recall that $\emptyset' = \{(e, n) : \Phi_e(n)\downarrow\}$.

Let TOT $= \{e : \Phi_e(n)\downarrow \text{ for all } n\}$.

Relative Computability

f is g-computable if there is an algorithm for computing f using information from g. We write $f \leqslant_{\mathrm{T}} g$.

Example: Recall that $\emptyset' = \{(e, n) : \Phi_e(n)\downarrow\}$.

Let TOT $= \{e : \Phi_e(n)\downarrow$ for all $n\}$.

Here is an algorithm showing that $\emptyset' \leqslant_{\mathrm{T}}$ TOT.

On input (e, n), find an i s.t. on any input m,
$\Phi_i(m)$ simulates $\Phi_e(n)$.
[So $\Phi_i(m) = \Phi_e(n)$ if $\Phi_e(n)\downarrow$,
and $\Phi_i(m)\uparrow$ if $\Phi_e(n)\uparrow$.]

Then $(e, n) \in \emptyset'$ iff $i \in$ TOT,
so return 1 if $i \in$ TOT and 0 otherwise.

Relative Computability

f is g-computable if there is an algorithm for computing f using information from g. We write $f \leqslant_{\mathbf{T}} g$.

Example: Recall that $\emptyset' = \{(e, n) : \Phi_e(n)\downarrow\}$.

Let TOT $= \{e : \Phi_e(n)\downarrow$ for all $n\}$.

Here is an algorithm showing that $\emptyset' \leqslant_{\mathbf{T}}$ TOT.

On input (e, n), find an i s.t. on any input m, $\Phi_i(m)$ simulates $\Phi_e(n)$.

[So $\Phi_i(m) = \Phi_e(n)$ if $\Phi_e(n)\downarrow$,
and $\Phi_i(m)\uparrow$ if $\Phi_e(n)\uparrow$.]

Then $(e, n) \in \emptyset'$ iff $i \in$ TOT,
so return 1 if $i \in$ TOT and 0 otherwise.

If $f \leqslant_{\mathbf{T}} g$ and $g \leqslant_{\mathbf{T}} f$, then we say that f and g are Turing equivalent and write $f \equiv_{\mathbf{T}} g$.

Relativization

We can relativize other computability theoretic concepts.

For instance, A is B-c.e. if there is an algorithm for enumerating A using information from B.

Similarly, we can list all the A-partial computable functions $\Phi_0^A, \Phi_1^A, \ldots$ and define the Halting Problem relative to A as $A' = \{(e, n) : \Phi_e^A(n)\downarrow\}$.

Relativization

We can relativize other computability theoretic concepts.

For instance, A is B-c.e. if there is an algorithm for enumerating A using information from B.

Similarly, we can list all the A-partial computable functions $\Phi_0^A, \Phi_1^A, \ldots$ and define the Halting Problem relative to A as $A' = \{(e, n) : \Phi_e^A(n)\downarrow\}$.

A is low if $A' \equiv_{\mathrm{T}} \emptyset'$.

Relativization

We can relativize other computability theoretic concepts.

For instance, A is B-c.e. if there is an algorithm for enumerating A using information from B.

Similarly, we can list all the A-partial computable functions $\Phi_0^A, \Phi_1^A, \ldots$ and define the Halting Problem relative to A as $A' = \{(e, n) : \Phi_e^A(n)\downarrow\}$.

A is low if $A' \equiv_{\mathrm{T}} \emptyset'$.

We can also relativize the notions of ML-test, prefix-free Kolmogorov complexity, and left-c.e. martingale and use these to define a notion of relativized 1-randomness.

Relativization

We can relativize other computability theoretic concepts.

For instance, A is B-c.e. if there is an algorithm for enumerating A using information from B.

Similarly, we can list all the A-partial computable functions $\Phi_0^A, \Phi_1^A, \ldots$ and define the Halting Problem relative to A as $A' = \{(e, n) : \Phi_e^A(n)\downarrow\}$.

A is low if $A' \equiv_{\mathrm{T}} \emptyset'$.

We can also relativize the notions of ML-test, prefix-free Kolmogorov complexity, and left-c.e. martingale and use these to define a notion of relativized 1-randomness.

For example: An A-Martin-Löf Test is a sequence of uniformly Σ_1^A classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\mu(\mathcal{C}_n) \leqslant 2^{-n}$.

α is A-1-random if $\alpha \notin \bigcap_n \mathcal{C}_n$ for every such test.

The Arithmetical Hierarchy

A Σ_1^0 set is one of the form $\{n : \exists x\, R(n, x)\}$ with R a computable predicate.

A Π_1^0 set is one of the form $\{n : \forall x\, R(n, x)\}$ with R a computable predicate.

The Arithmetical Hierarchy

A Σ^0_1 set is one of the form $\{n : \exists x\, R(n, x)\}$ with R a computable predicate.

A Π^0_1 set is one of the form $\{n : \forall x\, R(n, x)\}$ with R a computable predicate.

The Σ^0_1 sets are the c.e. sets, and the Π^0_1 sets are their complements.

The Arithmetical Hierarchy

A Σ^0_1 set is one of the form $\{n : \exists x\, R(n, x)\}$ with R a computable predicate.

A Π^0_1 set is one of the form $\{n : \forall x\, R(n, x)\}$ with R a computable predicate.

The Σ^0_1 sets are the c.e. sets, and the Π^0_1 sets are their complements.

A Σ^0_n set is one of the form $\{n : \exists x\, R(n, x)\}$ with R a Π^0_{n-1} predicate.

A Π^0_n set is one of the form $\{n : \forall x\, R(n, x)\}$ with R a Σ^0_{n-1} predicate.

The Arithmetical Hierarchy

A Σ^0_1 set is one of the form $\{n : \exists x\, R(n, x)\}$ with R a computable predicate.

A Π^0_1 set is one of the form $\{n : \forall x\, R(n, x)\}$ with R a computable predicate.

The Σ^0_1 sets are the c.e. sets, and the Π^0_1 sets are their complements.

A Σ^0_n set is one of the form $\{n : \exists x\, R(n, x)\}$ with R a Π^0_{n-1} predicate.

A Π^0_n set is one of the form $\{n : \forall x\, R(n, x)\}$ with R a Σ^0_{n-1} predicate.

Every c.e. set is $\emptyset'$-computable.

The Arithmetical Hierarchy

A Σ^0_1 set is one of the form $\{n : \exists x\, R(n,x)\}$ with R a computable predicate.

A Π^0_1 set is one of the form $\{n : \forall x\, R(n,x)\}$ with R a computable predicate.

The Σ^0_1 sets are the c.e. sets, and the Π^0_1 sets are their complements.

A Σ^0_n set is one of the form $\{n : \exists x\, R(n,x)\}$ with R a Π^0_{n-1} predicate.

A Π^0_n set is one of the form $\{n : \forall x\, R(n,x)\}$ with R a Σ^0_{n-1} predicate.

Every c.e. set is $\emptyset'$-computable.

Let $\emptyset^{(n)} = (\emptyset^{(n-1)})'$.

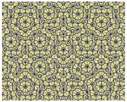
$\emptyset^{(n)}$ is Σ^0_n, and every Σ^0_n set is $\emptyset^{(n)}$-computable.

Part 2: Examples, Properties, and Variations

Weakening 1-randomness

A Little More Computability Theory

Strengthening 1-randomness



Highly Nonrandom Sequences

An Example of a 1-random Sequence

Let U be a universal prefix-free partial computable function.

Let $\Omega = \sum_{\sigma \in \mathrm{dom}\, U} 2^{-|\sigma|}$.

Ω is the halting probability of U.

An Example of a 1-random Sequence

Let U be a universal prefix-free partial computable function.

Let $\Omega = \sum_{\sigma \in \mathrm{dom}\, U} 2^{-|\sigma|}$.

Ω is the halting probability of U.

Ω is a left-c.e. real, and $\Omega \equiv_{\mathbf{T}} \emptyset'$.

Indeed, Ω can be seen as a highly compressed version of $\emptyset'$.

An Example of a 1-random Sequence

Let U be a universal prefix-free partial computable function.

Let $\Omega = \sum_{\sigma \in \mathrm{dom}\, U} 2^{-|\sigma|}$.

Ω is the halting probability of U.

Ω is a left-c.e. real, and $\Omega \equiv_{\mathrm{T}} \emptyset'$.

Indeed, Ω can be seen as a highly compressed version of $\emptyset'$.

Ω is 1-random.

The Kučera-Gács Theorem

Thm (Kučera; Gács). For each α there is a 1-random β s.t. $\alpha \leqslant_{\mathbf{T}} \beta$.

The Kučera-Gács Theorem

Thm (Kučera; Gács). For each α there is a 1-random β s.t. $\alpha \leqslant_{\mathbf{T}} \beta$.

These 1-random sequences, like Ω, are computationally powerful.

In a sense, they are "fake 1-random sequences".

Intuitively, we should not be able to extract information from random sequences, so they should be computationally weak.

The Kučera-Gács Theorem

Thm (Kučera; Gács). For each α there is a 1-random β s.t. $\alpha \leqslant_{\mathbf{T}} \beta$.

These 1-random sequences, like Ω, are computationally powerful.

In a sense, they are “fake 1-random sequences”.

Intuitively, we should not be able to extract information from random sequences, so they should be computationally weak.

Indeed, computing a given noncomputable set is a rare property.

Thm (de Leeuw, Moore, Shannon, and Shapiro; Sacks). If A is not computable then $\mu(\{B : A \leqslant_{\mathbf{T}} B\}) = 0$.

Strengthening 1-randomness

α is *n*-random if it is $\emptyset^{(n-1)}$-1-random.

Strengthening 1-randomness

α is *n-random* if it is $\emptyset^{(n-1)}$-1-random.

Higher order randomness gets us closer to our intuitions about random sequences.

For example, the only c.e. sets computable from a 2-random sequence are the computable ones.

Strengthening 1-randomness

α is *n-random* if it is $\emptyset^{(n-1)}$-1-random.

Higher order randomness gets us closer to our intuitions about random sequences.

For example, the only c.e. sets computable from a 2-random sequence are the computable ones.

There are also interesting notions of randomness strictly between 1-randomness and 2-randomness.

Strengthening 1-randomness

α is *n*-random if it is $\emptyset^{(n-1)}$-1-random.

Higher order randomness gets us closer to our intuitions about random sequences.

For example, the only c.e. sets computable from a 2-random sequence are the computable ones.

There are also interesting notions of randomness strictly between 1-randomness and 2-randomness.

A generalized test is a sequence of uniformly Σ_1^0 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\lim_n \mu(\mathcal{C}_n) = 0$.

Strengthening 1-randomness

α is *n*-random if it is $\emptyset^{(n-1)}$-1-random.

Higher order randomness gets us closer to our intuitions about random sequences.

For example, the only c.e. sets computable from a 2-random sequence are the computable ones.

There are also interesting notions of randomness strictly between 1-randomness and 2-randomness.

A generalized test is a sequence of uniformly Σ^0_1 classes $\mathcal{C}_0, \mathcal{C}_1, \ldots$ s.t. $\lim_n \mu(\mathcal{C}_n) = 0$.

$\alpha \in 2^\omega$ passes this test if $\alpha \notin \bigcap_n \mathcal{C}_n$.

α is weakly 2-random if it passes every generalized test.

n-randomness and Kolmogorov complexity

It is possible to characterize 2-randomness using Kolmogorov complexity.

Thm (Nies, Stephan, and Terwijn; Miller). α is 2-random iff $\exists^\infty n\,(C(\alpha \upharpoonright n) \geqslant n - O(1))$.

There is a similar characterization using prefix-free complexity.

n-randomness and Kolmogorov complexity

It is possible to characterize 2-randomness using Kolmogorov complexity.

Thm (Nies, Stephan, and Terwijn; Miller). α is 2-random iff $\exists^\infty n\,(C(\alpha \upharpoonright n) \geqslant n - O(1))$.

There is a similar characterization using prefix-free complexity.

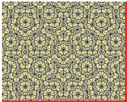
Open Problem. Are there characterizations along these lines for higher levels of randomness?

Part 2: Examples, Properties, and Variations

Weakening 1-randomness

A Little More Computability Theory

Strengthening 1-randomness



Highly Nonrandom Sequences

K-triviality

If α is computable then we can describe $\alpha \upharpoonright n$ by describing n and giving an algorithm for α, which does not depend on n.

So $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ and $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

K-triviality

If α is computable then we can describe $\alpha \upharpoonright n$ by describing n and giving an algorithm for α, which does not depend on n.

So $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ and $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

Thm (Chaitin). If $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ then α is computable.

K-triviality

If α is computable then we can describe $\alpha \upharpoonright n$ by describing n and giving an algorithm for α, which does not depend on n.

So $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ and $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

Thm (Chaitin). If $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ then α is computable.

Thm (Solovay). There is a noncomputable α s.t. $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

K-triviality

If α is computable then we can describe $\alpha \upharpoonright n$ by describing n and giving an algorithm for α, which does not depend on n.

So $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ and $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

Thm (Chaitin). If $C(\alpha \upharpoonright n) \leqslant C(n) + O(1)$ then α is computable.

Thm (Solovay). There is a noncomputable α s.t. $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

We say that α is *K-trivial* if $K(\alpha \upharpoonright n) \leqslant K(n) + O(1)$.

Other Notions of Randomness Theoretic Weakness

Having no derandomization power:
α is low for 1-randomness if every 1-random sequence is α-1-random.

Other Notions of Randomness Theoretic Weakness

Having no derandomization power:
α is low for 1-randomness if every 1-random sequence is α-1-random.

Having no compression power:
α is low for K if $K^{\alpha}(\sigma) = K(\sigma) \pm O(1)$.

Other Notions of Randomness Theoretic Weakness

Having no derandomization power:
α is low for 1-randomness if every 1-random sequence is α-1-random.

Having no compression power:
α is low for K if $K^{\alpha}(\sigma) = K(\sigma) \pm O(1)$.

Thm (de Leeuw, Moore, Shannon, and Shapiro; Sacks). If α is not computable then $\mu(\{\beta : \alpha \leqslant_{\mathbf{T}} \beta\}) = 0$.

Other Notions of Randomness Theoretic Weakness

Having no derandomization power:
α is low for 1-randomness if every 1-random sequence is α-1-random.

Having no compression power:
α is low for K if $K^{\alpha}(\sigma) = K(\sigma) \pm O(1)$.

Thm (de Leeuw, Moore, Shannon, and Shapiro; Sacks). If α is not computable then $\mu(\{\beta : \alpha \leqslant_{\mathbf{T}} \beta\}) = 0$.

By the Kučera-Gács Theorem, $\{\beta : \alpha \leqslant_{\mathbf{T}} \beta\}$ always contains a 1-random sequence, and so is never ML-null.

Other Notions of Randomness Theoretic Weakness

Having no derandomization power:
α is low for 1-randomness if every 1-random sequence is α-1-random.

Having no compression power:
α is low for K if $K^\alpha(\sigma) = K(\sigma) \pm O(1)$.

Thm (de Leeuw, Moore, Shannon, and Shapiro; Sacks). If α is not computable then $\mu(\{\beta : \alpha \leqslant_{\mathrm{T}} \beta\}) = 0$.

By the Kučera-Gács Theorem, $\{\beta : \alpha \leqslant_{\mathrm{T}} \beta\}$ always contains a 1-random sequence, and so is never ML-null.

α is a base for 1-randomness if there is a $\beta \geqslant_{\mathrm{T}} \alpha$ s.t. β is α-1-random (equivalently, if $\{\beta : \alpha \leqslant_{\mathrm{T}} \beta\}$ is not α-ML-null).

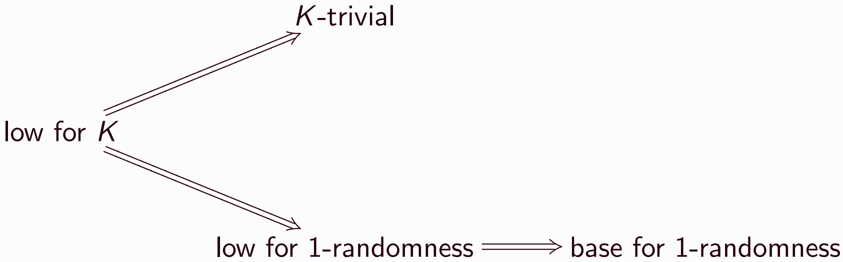
Easy Implications

α is K-trivial if $K(\alpha \restriction n) \leqslant K(n) + O(1)$.

α is low for 1-randomness if every 1-random is α-1-random.

α is low for K if $K^{\alpha}(\sigma) = K(\sigma) \pm O(1)$.

α is a base for 1-randomness if there is a $\beta \geqslant_{\mathbf{T}} \alpha$ s.t. β is α-1-random.



A Remarkable Coincidence

Thm (Nies). A sequence is K-trivial iff it is low for 1-randomness.

A Remarkable Coincidence

Thm (Nies). A sequence is K-trivial iff it is low for 1-randomness.

Thm (Nies and Hirschfeldt). A sequence is K-trivial iff it is low for K.

A Remarkable Coincidence

Thm (Nies). A sequence is K-trivial iff it is low for 1-randomness.

Thm (Nies and Hirschfeldt). A sequence is K-trivial iff it is low for K.

Thm (Hirschfeldt, Nies, and Stephan). A sequence is K-trivial iff it is a base for 1-randomness.

Thus all four notions of randomness theoretic weakness coincide.

How Chaos Resembles Order

Highly random objects can resemble highly patterned ones.

How Chaos Resembles Order

Highly random objects can resemble highly patterned ones.

A musical example.

Excerpt A: >

Excerpt B: >

How Chaos Resembles Order

Highly random objects can resemble highly patterned ones.

A musical example.

Excerpt A: from *Music of Changes* by John Cage

Excerpt B: from *Structures for Two Pianos* by Pierre Boulez

Cage's piece is an example of aleatory music.

Boulez's piece is an example of total serialism.

How Chaos Resembles Order: Mathematical Examples

α is low for Ω if Ω is α-1-random.

How Chaos Resembles Order: Mathematical Examples

α is low for Ω if Ω is α-1-random.

α is weakly low for K if $\exists^\infty \sigma \, (K^\alpha(\sigma) = K(\sigma) \pm O(1))$.

How Chaos Resembles Order: Mathematical Examples

α is low for Ω if Ω is α-1-random.

α is weakly low for K if $\exists^\infty \sigma \, (K^\alpha(\sigma) = K(\sigma) \pm O(1))$.

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

How Chaos Resembles Order: Mathematical Examples

α is low for Ω if Ω is α-1-random.

α is weakly low for K if $\exists^\infty \sigma\,(K^\alpha(\sigma) = K(\sigma) \pm O(1))$.

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

Thm (Miller). Every 3-random sequence is weakly low for K.

How Chaos Resembles Order: Mathematical Examples

α is low for Ω if Ω is α-1-random.

α is weakly low for K if $\exists^\infty \sigma \, (K^\alpha(\sigma) = K(\sigma) \pm O(1))$.

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

Thm (Miller). Every 3-random sequence is weakly low for K.

Open Problem. Give a precise characterization of a notion of "useless information" that explains these and similar phenomena.

Van Lambalgen's Theorem

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

The "if" direction follows from the following key result.

Van Lambalgen's Theorem

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

The "if" direction follows from the following key result.

Thm (van Lambalgen). If α is 1-random and β is α-1-random then α is β-1-random.

Van Lambalgen's Theorem

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

The "if" direction follows from the following key result.

Thm (van Lambalgen). If α is 1-random and β is α-1-random then α is β-1-random.

Proof that every 2-random sequence if low for Ω.

If α is 2-random then it is $\emptyset'$-1-random, and so Ω-1-random.

Thm (Nies, Stephan, and Terwijn). A 1-random sequence is low for Ω iff it is 2-random.

The "if" direction follows from the following key result.

Thm (van Lambalgen). If α is 1-random and β is α-1-random then α is β-1-random.

Proof that every 2-random sequence if low for Ω.

If α is 2-random then it is $\emptyset'$-1-random, and so Ω-1-random.

By van Lambalgen's Theorem, Ω is α-1-random, so α is low for Ω. □

Surveys, Lecture Notes, etc.

homepages.mcs.vuw.ac.nz/~downey/

www.cs.auckland.ac.nz/~nies/

www.math.uchicago.edu/~drh/

www.math.dartmouth.edu/~frg/