Algorithmic Randomness

What is a random sequence?

Joachim Giesen, Institute of Computer Science

SMT Workshop: Facets of Probability

References

1. Rod Downey and Denis R. Hirschfeldt. *Algorithmic Randomness.* CACM 2019
2. Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (4th Ed.).* Springer 2019
3. Persi Diaconis and Frederick Brian Skyrms. *Ten Great Ideas about Chance*. Princeton University Press 2017

1. Historical Roots

Pierre-Simon Laplace (1749-1827): What is a random sequence?

We arrange in our thought all possible events in various classes; and we regard as extraordinary those classes which include a very small number. Thus, in the game of heads and tails, if heads comes up a hundred times in a row, then this appears to us extraordinary, because the almost infinite number of combinations that can arise in a hundred throws are divided in regular sequences, or those in which we observe a rule that is easy to grasp, and in irregular sequences, which are incomparably more numerous.

Pierre-Simon Laplace

Émil Borel (1871-1956): Normal numbers

Let x be a sequence of bits. If the bits are the result of fair coin tosses, then we expect by the strong law of large numbers

$$\lim_{n\to\infty} \frac{|\{x_k = 1 \,:\, k < n\}|}{n} = \frac{1}{2}.$$

Émil Borel (1871-1956): Normal numbers

Let x be a sequence of bits. If the bits are the result of fair coin tosses, then we expect by the strong law of large numbers

$$\lim_{n\to\infty} \frac{|\{x_k = 1 \,:\, k < n\}|}{n} = \frac{1}{2}.$$

We can interpret the sequence as a number in $[0, 1]$. It is called **normal** to Base-2 if it satisfies the limit equation. It is called **absolutely normal** if it is normal to every base.

Émil Borel (1871-1956): Normal numbers

Let x be a sequence of bits. If the bits are the result of fair coin tosses, then we expect by the strong law of large numbers

$$\lim_{n\to\infty} \frac{|\{x_k = 1 \ : \ k < n\}|}{n} = \frac{1}{2}.$$

We can interpret the sequence as a number in $[0, 1]$. It is called **normal** to Base-2 if it satisfies the limit equation. It is called **absolutely normal** if it is normal to every base.

We would assume a random number to be absolutely normal.

David Champernowne (1912-2000): Champernowne number

0110111001011101111000100110101011110011011110111110000100011001 0 . . .

David Champernowne (1912-2000): Champernowne number

$$0110111001011101111000100110101011110011011110111110000010001100 10\ldots$$

Is **normal**, but not **random**. If we select only every $f(n)$-th bit from the sequence, where

$$f(n) = 1 + \sum_{k \leqslant n} k\left(2^k - 2^{k-1}\right),$$

then we get

$$1111111111111111111111111111111111111111111111111111111111111111\ldots$$

Richard von Mises (1883-1953): Place-selection functions

Place-selection function: Strictly increasing function $f : \mathbb{N} \to \mathbb{N}$

A sequence can be considered **random** if

$$\lim_{n\to\infty} \frac{|\{x_{f(k)} = 1 \,:\, k < n\}|}{n} = \frac{1}{2}.$$

for all "*possible*" selection functions f.

Alonso Church (1903-1995): Computable selection functions

Question: How to select selection functions?

Alonso Church (1903-1995): Computable selection functions

Question: How to select selection functions?

No sequence can be normal for all selection functions. Therefore, selection functions need to be chosen **independently** from the sequence.

Alonso Church (1903-1995): Computable selection functions

Question: How to select selection functions?

No sequence can be normal for all selection functions. Therefore, selection functions need to be chosen **independently** from the sequence.

Selection functions should be **computable**. So, take all computable selection functions to determine the randomness of a sequence.

Jean Ville (1910-1989): Counterexample

Theorem: *For any* **countable** *set of selection functions there exists sequence x that passes the randomness test for all selection functions from the set, but for every $n \in \mathbb{N}$ there are more zeroes than ones in the sequence up to x_n.*

2. Algorithmic Randomness

Andrey Kolmogorov (1903-1987): Probability

The **probability** that a sequence of coin tosses starts with a specific Bit-string of length n, for instance 101 for $n = 3$, is 2^{-n}.

The uniform (Lebesgue) measure of **all** sequences that start with this prefix is 2^{-n}.

Andrey Kolmogorov (1903-1987): Probability

The **probability** that a sequence of coin tosses starts with a specific Bit-string of length n, for instance 101 for $n = 3$, is 2^{-n}.

The uniform (Lebesgue) measure of **all** sequences that start with this prefix is 2^{-n}.

Important: This notion of probability does not assign any meaning to the randomness of a particular sequence.

Per Martin-Löf (1942): Random sequence

Combines ideas from von Mises, Church, and Kolmogorov.

- Let M be a Turing machine that at level $n \in \mathbb{N}$ enumerates bit finite Bit-strings $\sigma_1^n, \sigma_2^n, \sigma_3^n, \ldots$
- Let T_n be the set of sequences that start with some σ_i^n.
- Let M be such that $T_{n+1} \subseteq T_n$ and the uniform (Lesbesgue) measure of T_n is at most 2^{-n}.

A sequence x **passes** M's test if

$$x \notin \bigcap_{n=1}^{\infty} T_n \qquad [\textit{Constructive nullset}]$$

It is called **Martin-Löf random** if and only if it passes all such tests.

Andrey Kolmogorov (1903-1987): Compressibility

Given a prefix-free, universal Turing machine U the **Kolmogorov complexity** $K(\sigma)$ of a finite Bit-string σ is

$$K(\sigma) = \min\big\{L(\tau) \ : \ U(\tau) = \sigma\big\}.$$

Remark: $K(\sigma)$ depends on the particular U only through an additive constant.

Andrey Kolmogorov (1903-1987): Compressibility

Given a prefix-free, universal Turing machine U the **Kolmogorov complexity** $K(\sigma)$ of a finite Bit-string σ is

$$K(\sigma) = \min\big\{L(\tau) \ : \ U(\tau) = \sigma\big\}.$$

Remark: $K(\sigma)$ depends on the particular U only through an additive constant.

The string σ is called **algorithmically random** if $K(\sigma) \geqslant L(\sigma) - O(1)$.

Andrey Kolmogorov (1903-1987): Compressibility

Given a prefix-free, universal Turing machine U the **Kolmogorov complexity** $K(\sigma)$ of a finite Bit-string σ is

$$K(\sigma) = \min\big\{L(\tau) \ : \ U(\tau) = \sigma\big\}.$$

Remark: $K(\sigma)$ depends on the particular U only through an additive constant.

The string σ is called **algorithmically random** if $K(\sigma) \geqslant L(\sigma) - O(1)$.

Theorem[Schnorr]: A sequence x is Martin-Löf random if and only if

$$K(x_1 \dots x_n) \geqslant n - O(1)$$

for all $n \in \mathbb{N}$.

Claus Peter Peter Schnorr (1943-2025): Martingales

A **martingale** is a function $d : \{0,1\}^\star \to \mathbb{R}_{\geqslant 0}$ that maps Bit-strings to non-negative reals and satisfies

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

A martingale d **succeeds** on a sequence x if $\limsup d(x_1 \ldots x_n) = \infty$.

Theorem: A sequence x is Martin-Löf random if and only if no effectively computable martingale succeeds on x.

Claus Peter Peter Schnorr (1943-2025): Martingales

A **martingale** is a function $d : \{0,1\}^\star \to \mathbb{R}_{\geqslant 0}$ that maps Bit-strings to non-negative reals and satisfies

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

A martingale d **succeeds** on a sequence x if $\limsup d(x_1 \ldots x_n) = \infty$.

Theorem: A sequence x is Martin-Löf random if and only if no effectively computable martingale succeeds on x.

Remark: A martingale is effectively computable if there exists a Turing machine M that on input σ approximates $d(\sigma)$ to arbitrary precision.

Robustness

The definition of algorithmic (Martin-Löf) randomness is **robust** in the sense that three intuitive approaches effectively lead to the same definition:

A random sequence should

- have no rare computable properties [statistics]
- have no regularities that allow for compression [coding]
- be unpredictable [gambling]

3. Solomonoff Induction

Leonid Levin (1948): Universal prior

Let U be a prefix-free, universal Turing machine. The **universal prior** on $\{0,1\}^\star$ is given as

$$p(\sigma) = \sum_{\tau : U(\tau) = \sigma} 2^{-L(\tau)},$$

where τ ranges over all self-delimiting programs that run on U.

Properties:

- $\sum_\sigma p(\sigma) \leqslant 1$
- p is not computable, but only lower-semi-computable
- For every computable probability sequence q, exists a constant $c_q > 0$ such that

$$p(\sigma) \geqslant c_q q(\sigma) \qquad [\textit{Universality}]$$

Ray Solmonoff (1926-2009): Induction

Let p be Levin's universal prior. Given an observed Bit-string σ of length n. Solmonoff induction predicts the probability of the next bit as

$$q(x_{n+1} = 0|\sigma) = \frac{p(\sigma 0)}{p(\sigma)}$$
$$q(x_{n+1} = 1|\sigma) = \frac{p(\sigma 1)}{p(\sigma)}$$

Ray Solmonoff (1926-2009): Induction

Let p be Levin's universal prior. Given an observed Bit-string σ of length n. Solmonoff induction predicts the probability of the next bit as

$$q(x_{n+1} = 0|\sigma) = \frac{p(\sigma 0)}{p(\sigma)}$$
$$q(x_{n+1} = 1|\sigma) = \frac{p(\sigma 1)}{p(\sigma)}$$

Theorem: If the data is generated by a computable probability distribution r, then the Solomonoff predictor q converges to r in the sense that its predictions become arbitrarily close to the true probabilities.

Ray Solmonoff (1926-2009): Induction

Solomonoff induction combines:

1. **Bayes' Theorem**: updating beliefs based on evidence
2. **Occam's Razor**: simpler explanations are more likely
3. **Universal Computation**: all possible computable explanations are considered