Computable Randomness and Its Properties

Jason Rute

Department of Mathematical Science
Carnegie Mellon University

Computability Complexity and Randomness
July 2012

Outline

What makes a randomness notion robust?

- It has many characterizations
  (e.g. ML tests, martingales, machines, etc.)
- It comes up in computable analysis
- It generalizes to other probability spaces
- It is preserved under alt. characterizations of the same space
  (e.g. base invariance)
- It is preserved by morphisms (measure preserving maps)
- It is preserved by isomorphisms

Is computable randomness robust?

- It has many characterizations (e.g. ML tests, martingales, machines, etc.) Yes (Many...)
- It comes up in computable analysis Yes (Brattka et al.; etc.)
- It generalizes to other probability spaces Yes (R.)
- It is preserved under alt. characterizations of the same space (e.g. base invariance) Yes (Brattka et al.; R.)
- It is preserved by morphisms (measure preserving maps) No (R.; Bienvenu and Porter)
- It is preserved by isomorphisms Yes (R.)

Goals of this talk

Primary Goal
Give a better understanding of computable randomness.

Secondary Goal
Give a robust definition of computable randomness outside of Cantor space.

Tertiary Goal
Explain the interplay between

- randomness,
- morphisms, and
- sigma-algebras.

Outline

Computable Randomness on $(2^\omega, \text{fair-coin})$

Definition

A computable martingale is a computable function $M : 2^{<\omega} \to \mathbb{R}^+$ such that for all $\sigma \in 2^{<\omega}$,

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

Given $X \in 2^\omega$, M succeeds on $X \Leftrightarrow \limsup_n M(X \upharpoonright n) = \infty$.

$X \in 2^\omega$ is computably random if there is not any computable martingale M which succeeds on X.

Measures on 2^ω.

Definitions

A computable measure on 2^ω can be represented by a computable function $\mu : 2^{<\omega} \to \mathbb{R}^+$ such that

$$\mu(\sigma 0) + \mu(\sigma 1) = \mu(\sigma).$$

By the Carathéodory extension theorem this can be uniquely extended to a Borel measure on 2^ω.

A probability measure is a measure with $\mu(\text{empty string}) = 1$.
The fair-coin measure is defined by $\mu(\sigma) = 2^{-|\sigma|}$.

We write $\mu(\sigma)$ instead of $\mu([\sigma])$.

Measures and martingales are closely related.

Computable randomness on $(2^\omega, \mu)$.

Let μ be a computable probability measure on 2^ω.

Define a computable martingale on $(2^\omega, \mu)$ as
a partial computable $M : 2^{<\omega} \to \mathbb{R}^+$ which satisfies

- (Fairness condition)
  $M(\sigma 0) \cdot \mu(\sigma 0) + M(\sigma 1) \cdot \mu(\sigma 1) = M(\sigma) \cdot \mu(\sigma)$.
- (Impossibility condition) $M(\sigma)$ is defined $\Leftrightarrow$ $\mu(\sigma) > 0$.

In other words, M is a.e. computable.

$X \in 2^\omega$ is μ-computably random if no such computable martingale succeeds on X (and X is not in any μ-null $[\sigma]$).

Small Question
Is the impossibility condition necessary?

Many test notions

All these tests extend to $(2^\omega, \mu)$ and are equivalent

- Martingale test (with impossibility condition)
- Graded ML tests* (Downey, Griffiths, and LaForte)
- Bounded ML tests (Merkle, Mihailović, and Slaman)
- Machine (K-Complexity) tests (Mihailović)
- Process complexity* (Day)
- Integral tests (R.)
- Solovay tests (R.)
- and more...

*I haven't checked yet, but I am sure they do.

Outline

Computable Metric Spaces

A Crash course!

Definition

A triple (X, d, S) is a computable metric space if

- X is a complete metric space with metric $d : X \times X \to \mathbb{R}^+$,
- $S = \{a_i\}_{i \in \mathbb{N}} \subseteq X$ is a countable dense set (the simple points),
- $d(a_i, a_j)$ is computable from i, j.

From this, we can define computable points, Σ^0_1 sets, Π^0_1 sets, etc.

Examples.

- $([0,1]^n, |x-y|,$ rational vectors$)$
- $(C[0,1], \|f-g\|_\infty,$ rational piecewise-affine functions$)$

Computable Probability Spaces

A Crash course!

Definition

A pair $(\mathscr{X}, \mu)$ is a computable (Borel) probability space if

- $\mathscr{X} = (X, d, S)$ is a computable metric space,
- μ is a Borel probability measure on X, and
- $\int f \, d\mu$ is uniformly computable for all bounded computable functions $f : \mathscr{X} \to [0, 1]$.

This is equivalent to many other definitions.

Representing a comp. prob. space

Question

How to nicely represent a computable probability space?

Answer

Treat it like Cantor space (2^ω)

...by either...

- ...using isomorphisms to/from Cantor space.
- ...by decompose space into "cells" (similar to cylinder sets $[\sigma]$).

Morphisms and Isomorphisms

A measurable map $T\colon (\mathscr{X},\mu) \to (\mathscr{Y},\nu)$ is a morphism if it is measure preserving, i.e.

$$\mu(T^{-1}(B)) = \nu(B) \quad B \subseteq \mathscr{Y}$$

A measurable map $T\colon (\mathscr{X},\mu) \to (\mathscr{Y},\nu)$ is an isomorphism if T has an inverse morphism $S\colon (\mathscr{Y},\nu) \to (\mathscr{X},\mu)$, i.e.

$$S \circ T = id_X \quad \text{and} \quad T \circ S = id_Y$$

Facts

- All spaces are isomorphic to a Borel measure on Cantor space.
- All atomless spaces are isomorphic to fair-coin measure on 2^ω.

A "computable morphism" is one that is a "computable map".

$3\frac{1}{2}$ Types of "Computable" Maps

Type 1: Total computable maps $f : (\mathscr{X}, \mu) \to \mathscr{Y}$

- Take in fast Cauchy code.
- Return fast Cauchy code always.
- Equiv. input codes give equiv. output codes always.
- Unique and well-defined on all points.
- If $f : \mathscr{X} \to \{0,1\}$, then $f^{-1}(\{1\})$ is decidable.
- Are the effectively continuous maps.
- Example: truth-table computable map.

Type 2: A.e. computable maps $f:(\mathscr{X},\mu)\to\mathscr{Y}$

- Takes in fast Cauchy code.
- Returns fast Cauchy code with probability 1.
- Equiv. input codes give equiv. output codes with probability 1.
- Unique and well-defined on Kurtz randoms.
- If $f:\mathscr{X}\to\{0,1\}$, then $f^{-1}(\{1\})$ is a.e. decidable.
- Are the effectively a.e. continuous maps.
- Example: real $\mapsto$ binary expansion

$3\frac{1}{2}$ Types of "Computable" Maps

Type 3: "Computable with high probability"

Type 3: "Computable with high probability" maps $f : (\mathscr{X}, \mu) \to \mathscr{Y}$

- Takes in fast Cauchy code and some error $\varepsilon > 0$.
- Returns fast Cauchy code with prob. $1 - \varepsilon$ (for each $\varepsilon > 0$).
- Equiv. input codes give equiv. output codes with prob. $1 - \varepsilon$.
- Unique and well-defined on Schnorr randoms.
- If $f : \mathscr{X} \to \{0, 1\}$, then $f^{-1}(\{1\})$ is "decidable with high prob."
- Example: "Schnorr layerwise computable functions" (they are the same).
- Are the effectively measurable pointwise maps (in some sense).

Type 3.5: Effectively measurable $f\colon (X,\mu) \to Y$

- Doesn't have specific input/output (is equivalence class).
- f^{-1}(effectively measurable) is effectively measurable.
- If $f : X \to \{0,1\}$, then $f^{-1}(\{1\})$ is effectively measurable.
- Are the effectively measurable maps (up to a.e. equivalence)
- Example: "L^1-computable functions"

We will use a.e. computable isomorphisms

- In this talk I will use a.e. computable morphisms and isomorphisms.
- The results also hold for morphisms/isomorphisms that are "computable with high probability" (Schnorr layerwise computable).
- The results also have an interpretation using effectively measurable morphisms. ML/computable/Schnorr randoms become generic filters as in Solovay forcing.

Computable Randomness on $(\mathscr{X}, \mu)$

Theorem (Hoyrup, Rojas)

Let $(\mathscr{X}, \mu)$ be a computable probability space. There always exists a computable measure ν and an a.e. computable isomorphism

$$T\colon (\mathscr{X}, \mu) \cong (2^\omega, \nu)$$

Definition

Let $T\colon (\mathscr{X}, \mu) \cong (2^\omega, \nu)$ be an a.e. computable isomorphism.
$x \in (\mathscr{X}, \mu)$ is computably random (w.r.t. T) if
$T(x)$ is computably random on $(2^\omega, \nu)$.

Theorem (R.)

The above definition does not depend on T.

Preservation of Randomness

Preservation of Randomness

If T is a.e. computable morphism, and x is ML random, then $T(x)$ is ML random.
(Also holds for Schnorr, Kurtz, and more.)

Theorem (R.; Bienvenu, Porter)

Preservation of computable randomness does not hold.

Theorem (R.)

But computable randomness is preserved by a.e. comp. isomorphisms.

Examples

On 2^ω with fair-coin measure. The definition of computable randomness stays the same; take the identity isomorphism.

On $[0,1]$ with Lebesgue measure. $x \in [0,1]$ is computably random iff its binary expansion is. Can use bases other than binary; this shows that computable randomness is base invariant (see Bratkka/Miller/Nies; also see earlier work on poly-time rand).

On $[0,1]^n$ with the Lebesgue measure. $x \in [0,1]^n$ is computably random iff $x = (0.X_1, \dots, 0.X_n)$ and $X_1 \oplus \cdots \oplus X_n$ is computably random in 2^ω.

On $C([0,1])$ with the Wiener measure (Brownian motion). Use Fouché's isomorphism from $(2^\omega, \text{fair-coin})$ to $(C([0,1]), W)$. Then $B \in C([0,1])$ is a computably random Brownian motion if it corresponds to a computable random $X \in 2^\omega$.

Outline

Morphisms as backwards maps

We often think of a morphism in this direction...

$$\begin{array}{rcl} T\colon (\mathscr{X},\mu) & \to & (\mathscr{Y},\nu) \\ x \in \mathscr{X} & \mapsto & T(x) \in \mathscr{Y} \end{array}$$

... but they are better thought of in the other direction.

$$\begin{array}{rcl} T^{-1}\colon \begin{array}{c}\text{Borel } \sigma\text{-algebra} \\ \text{of } (\mathscr{Y},\nu)\end{array} & \to & \begin{array}{c}\text{sub-}\sigma\text{-algebra} \\ \text{of } (\mathscr{X},\mu)\end{array} \\ B \subseteq \mathscr{Y} & \mapsto & T^{-1}(B) \subseteq \mathscr{X} \end{array}$$

Preservation of randomness

We think of preservation of randomness (e.g. for ML randomness) in this direction...

$$\begin{array}{rcl} T\colon (\mathscr{X},\mu) & \to & (\mathscr{Y},\nu) \\ x \in \mathrm{ML}(\mathscr{X}) & \mapsto & T(x) \in \mathrm{ML}(\mathscr{Y}) \end{array}$$

... but it is better thought of in the other direction.

$$T^{-1}\colon (V_n) \quad \mathrm{ML}(\mathscr{Y}) \text{ test} \quad \mapsto \quad \big(T^{-1}(V_n)\big) \quad \mathrm{ML}(\mathscr{X}) \text{ test}$$

This is because

- morphisms preserve measure.
- a.e. decidable morphisms (almost) preserve Σ^0_1 sets.

Characterizing Morphisms to 2^ω

Given an a.e. computable morphism $T\colon (\mathscr{X},\mu) \to (2^\omega, \nu)$, we get the following.

$$T^{-1}\colon \{Y \in 2^\omega \mid n\text{th bit of } Y \text{ is } 1\} \mapsto \{x \in \mathscr{X} \mid n\text{th bit of } T(x) \text{ is } 1\}$$

- The family of sets $A_n := \{x \in \mathscr{X} \mid n\text{th bit of } T(x) \text{ is } 1\}$ uniquely determines the morphism T.
- Moreover, any family of a.e. decidable sets $A_n \subseteq \mathscr{X}$ determines an a.e. decidable morphism $T\colon (\mathscr{X},\mu) \to (2^\omega, \mu_T)$ (where μ_T is the distribution of T on 2^ω).

Martingales

Given an a.e. computable morphism $T\colon(\mathscr{X},\mu)\to(2^\omega,\nu)$, we get the following.

$$T^{-1}\colon \quad \text{Martingale } M \quad \mapsto \quad \text{"Martingale" } T^{-1}(M)$$

$T^{-1}(M)$ is a fair betting strategy, but what is it betting on?

- $T^{-1}(M)$ is betting that the nth bit of $T(x)$ is 1 (or 0).
- Conversely, any list of a.e. decidable properties $\{A_n\}$ describes the setup for such a "martingale".
- Examples: non-monotonic martingales & martingale processes.
- It follows that preservation of randomness does not hold for computable randomness. (R.)

Isomorphisms

If $T\colon (\mathscr{X},\mu) \cong (2^\omega,\nu)$ is an a.e. computable isomorphism, then

- The sets $A_n = \{x \in \mathscr{X} \mid n\text{th bit of } T(x) \text{ is } 1\}$ generate the Borel sigma-algebra of $\mathscr{X}$.
- Moreover, $\{A_n\}$ is (almost) a topological basis for $\mathscr{X}$.
- (The converse also holds: If some family $\{A_n\}$ is (almost) a topological basis then the corresponding T is an isomorphism.)
- A "martingale" $T^{-1}(M)$ cannot succeed on computable randoms.
- This gives a martingale test for computable randomness on $(\mathscr{X},\mu)$
- Other tests can be extended as well.
- If follows that T preserves computable randomness. (R.)

Computable Analysis (Advanced!)

In probability theory there is a more general notion of martingale. It is a sequence of L^1-functions (f_n) with the property

$$E[f_{n+1} \mid f_0, \ldots, f_n] = f_n$$

Theorem (R.)

Let (f_n) be an a.e. computable martingale as above. If

- $\sup_n \|f_n\|_{L^1}$ is computable, and
- $\lim_n \sigma(f_0, \ldots, f_n)$ "converges effectively to a sigma-algebra"

Then $f_n(x)$ converges on computable randoms x.

This idea does not work for the Ergodic theorem since the sigma-algebras in the ergodic theorem converge "downward". Hence if the limit $\mathscr{F}$ of the sigma-algebras is "computable", then the expectation $E[f \mid \mathscr{F}]$ is L^1-computable and convergence happens on Schnorr randoms by a theorem of Hoyrup and Rojas.

Summary

- Generalizing randomness notions to other spaces leads to a better understanding of their properties on Cantor space.
- There is a duality between morphisms and representations of a sub-sigma-algebra.
- There is a duality between isomorphisms and representations of a prob. space.
- Isomorphisms and morphisms help to classify "good" and "bad" randomness notions. For example:
  - Nothing between ML randomness and computable randomness is preserved by morphisms.
  - If KL randomness is invariant under isomorphisms, then it is equal to ML randomness.
- Further questions:
  - ~~Is partial computable randomness preserved under isomorphisms?~~ No, computable randomness isn't invariant under permutations (see Merkle).

Thank You!

These slides and the corresponding paper will be available on my webpage:

`math.cmu.edu/~jrute`

Or just Google me, "Jason Rute".