Logic and Probability

Probability and Computability

Thomas Icard & Krzysztof Mierzewski

August 12, 2022

Probability theory in a computable setting

Probabilistic algorithms and machine implementations of probabilistic reasoning are limited by what (probabilistic) Turing machines can do.

A framework for understanding the limits and possibilities of what probabilistic computation, and computational probability models, can do in principle.

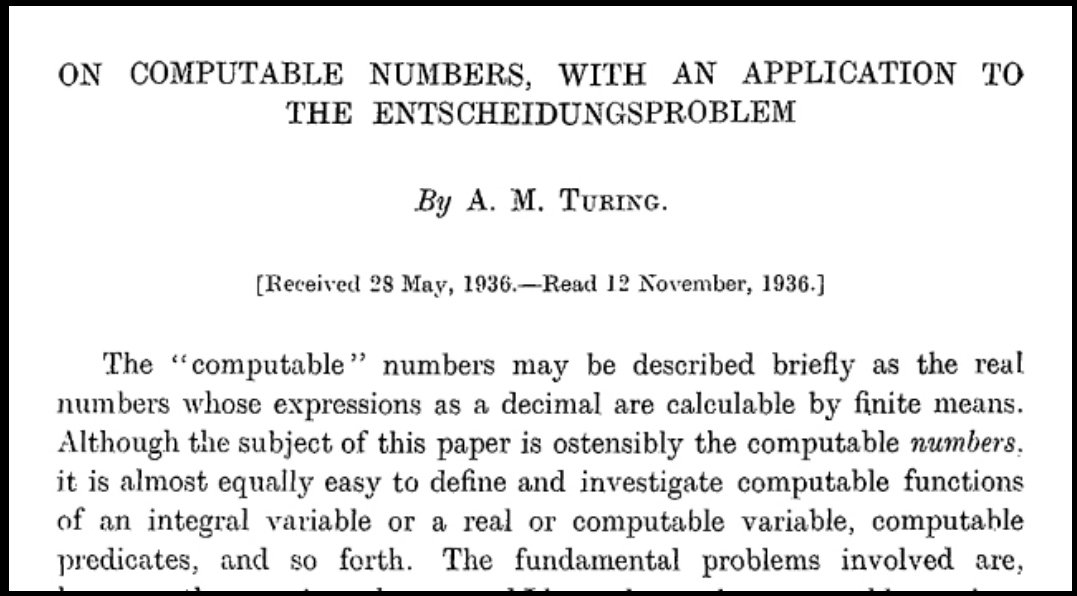
We will see some applications to the theory of Bayesian reasoning and inductive learning performed by computable agents.

Computable Reals

What is a computable real?

Computable Reals

What is a computable real?



ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are,

Proposal: identify a real number with a Turing machine program that computes the digits in its decimal (or binary) expansion, and define computable functions on computable numbers as operating on the corresponding programs.

Computable Reals: WAS TURING WRONG?!

Suppose we have

$$x := 0.232323232323...$$

$$y := 0.767676767676...$$

Compute $x + y$.

Computable Reals: WAS TURING WRONG?!

Suppose we have

$$x := 0.232323232323...$$

$$y := 0.767676767676...$$

Compute $x + y$.

Worry: we want addition to be a computable operation on real numbers, uniformly in the inputs!

Computable Reals

Another proposal:

Computable reals. A real $r \in \mathbb{R}$ is *computable* if and only if both sets $\{q \in \mathbb{Q} \mid q < r\}$ and $\{q \in \mathbb{Q} \mid r < q\}$ are computable. Equivalently:

$$r \text{ is computable} \iff \text{there is a computable sequence of rationals } \{q_n\}_{n\in\mathbb{N}} \text{ such that } |q_n - r| < 2^{-n} \text{ for all } n.$$

- **Examples**: $\sqrt{2}$, π, e,... (almost anything you can think of).

Under this definition, standard arithmetical operations ($+, \times, \ldots$) become computable.

Computable Reals

However:

A real $r \in \mathbb{R}$ if computable in Turing's sense if and only if it is computable in the fast-approximation sense.

So, was Turing's definition "wrong" after all?

(Lesson: when giving a computable account of an object, and of computations on it, it matters what we take as a computable representor for the object.)

Computable Reals

Fix a (prefix-free) universal Turing machine $\mathcal{U}$ that takes binary strings as inputs.

Pick an input string at random according to a fair coin measure. What is the probability that $\mathcal{U}$ will halt?

The halting probability is:

$$\Omega := \sum_{w \in dom(\mathcal{U})} 2^{-|w|}$$

Ω is left-ce. That is, there is a computable enumeration of rationals $\{q_i\}_{i \in \mathbb{N}}$ such that $\lim_{i \to \infty} q_i = \Omega$.

Computable Reals

Fix a (prefix-free) universal Turing machine $\mathcal{U}$ that takes binary strings as inputs.

Pick an input string at random according to a fair coin measure. What is the probability that $\mathcal{U}$ will halt?

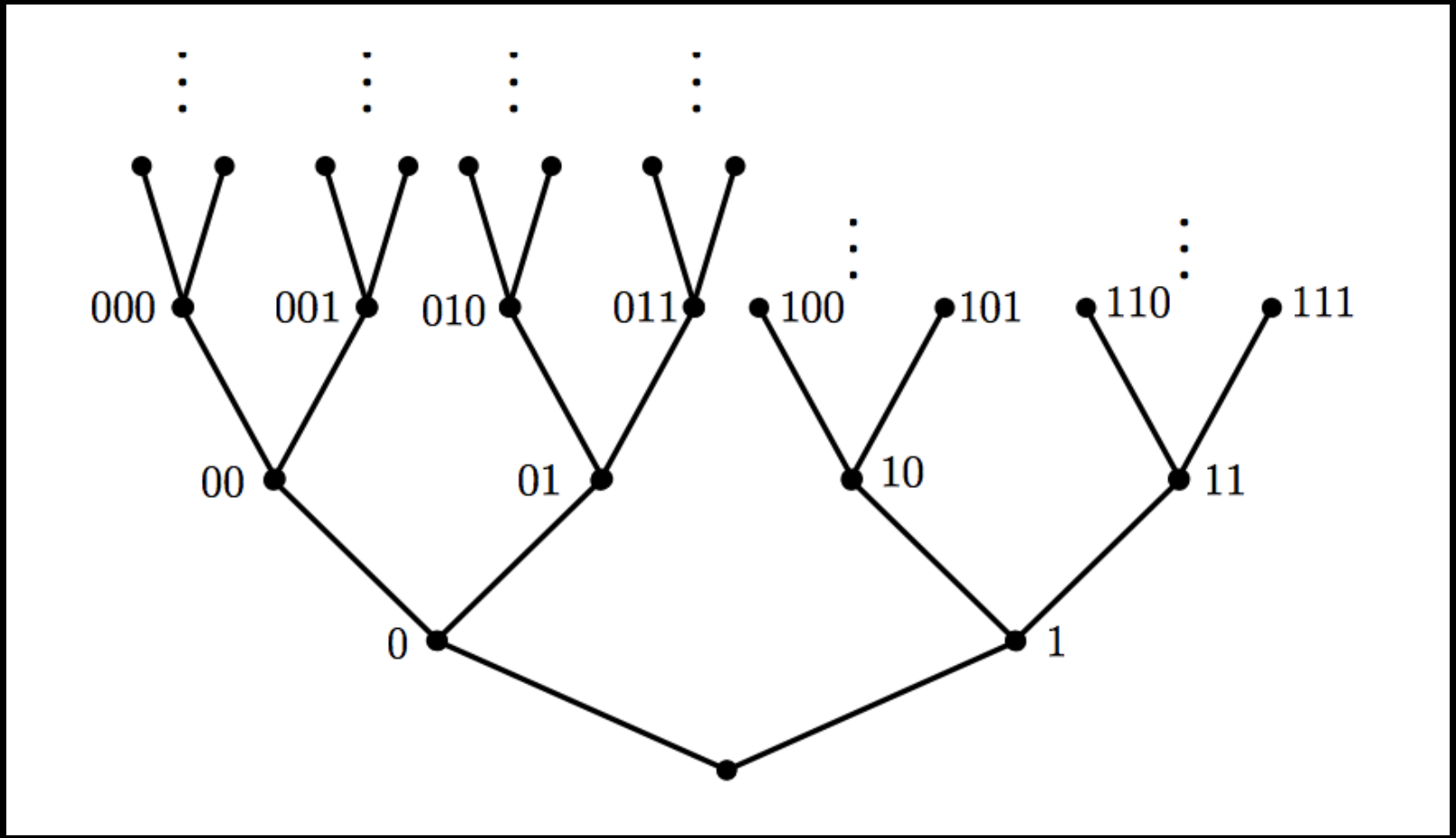
The halting probability is:

$$\Omega := \sum_{w \in dom(\mathcal{U})} 2^{-|w|}$$

Ω is left-ce. That is, there is a computable enumeration of rationals $\{q_i\}_{i \in \mathbb{N}}$ such that $\lim_{i \to \infty} q_i = \Omega$.

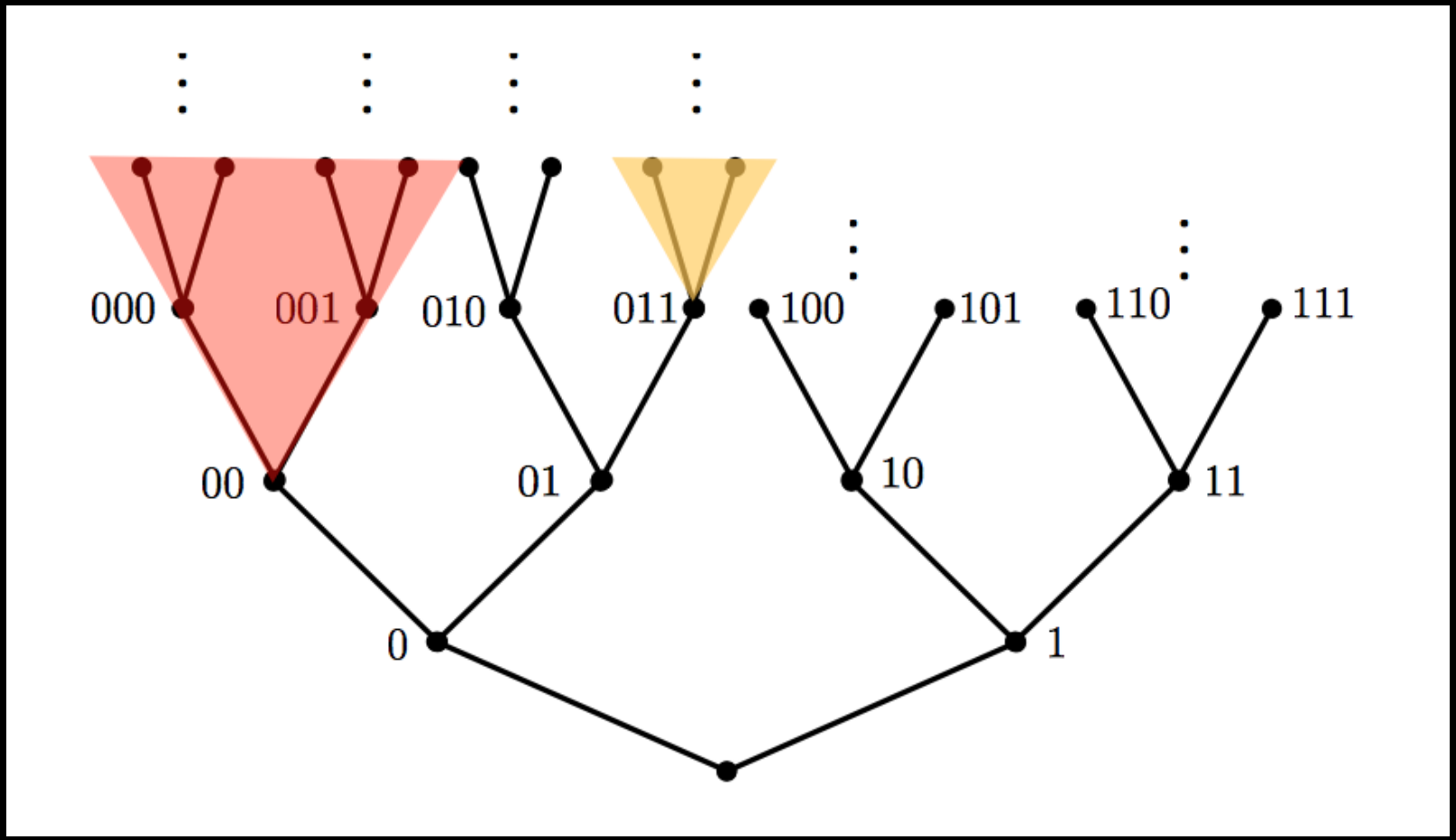
However, Ω is not computable: approximating Ω from the right would allow us to compute the Halting set.

Cantor space



The set $2^{\mathbb{N}}$ of infinite binary sequences

Cantor space



Given a sequence $w \in \{0,1\}^*$, let $[\![w]\!] := \{X \in 2^{\mathbb{N}} \mid X \text{ extends } w\}$, called the **cylinder set** of w.

Cantor space

$$(2^{\mathbb{N}}, \mathcal{B})$$

The collection $\mathcal{C}$ of finite unions of cylinders forms a ring.

- $\mathcal{B}$ is the *Borel algebra* generated by the cylinder sets (the smallest σ-algebra extending $\mathcal{C}$).

By Carathéodory's extension theorem, to uniquely specify a (countably additive) measure on $(2^{\mathbb{N}}, \mathcal{B})$, it is enough to have a finitely additive measure on the cylinder sets $[\![w]\!]$.

Computable measures on Cantor space

Any measure on $(2^{\mathbb{N}}, \mathcal{B})$ can be uniquely associated with a pre-measure: a function $m : \{0,1\}^* \to [0,1]$ such that

- $m(\varepsilon) = 1$
- $m(w) = m(w0) + m(w1)$ for every $w \in \{0,1\}^*$.

Setting $\mu([\![w]\!]) := m(w)$, this uniquely extends to a probability measure on $\mathcal{B}$.

A (Borel) measure μ on $(2^{\mathbb{N}}, \mathcal{B})$ is **computable** if and only if the values of $\mu([\![\tau]\!])$ are uniformly computable.

This means: there exists a computable function $f : \mathbb{N} \times \{0,1\}^* \to \mathbb{Q}$ such that for every n, $|f(n,w) - \mu([\![w]\!])| \leq 2^{-n}$.

(i.e. $\mu([\![\tau]\!])$ is a computable real *uniformly in the string* w.)

Computable measures on Cantor space

Examples:

- Uniform measure: $\lambda([\![w]\!]) := 2^{-|w|}$
- Bernoulli measure with computable bias: $B_p([\![w]\!]) = p^{\#w}(1-p)^{|w|-\#w}$ where $p \in (0,1)$ is computable (and $\#w :=$ number of 1's in w).
- Computable mixture of computable measures: $\nu = \alpha\mu_1 + (1-\alpha)\mu_2$

Probabilistic Turing Machines

Random bit tape (read):

1	0	0	1	0	**1**	1	0	...

Work tape (read/write):

0	1	**0**	0	0	1	1	0	...

Output tape (read/write):

1	**1**	0	0	0	0	0	0	...

A measure μ is the output distribution of a PTM T if, for each $w \in \{0,1\}^*$, T outputs w with probability $\mu([\![w]\!])$.

This means: $\mu([\![w]\!])$ is the probability of getting a random bit tape (by tossing a fair coin) that results in output w.

Given a PTM T, write P_T for its output distribution.

Computable measures on Cantor space

Proposition

A probability measure on Cantor space is computable if and only if it is the output distribution of a probabilistic Turing machine that halts almost surely.

$\Leftarrow$: Suppose we run T using a string r as the first bits of the random tape (terminate if the program calls for random bits beyond those in r).

If T halts and outputs w, this tells us that $P_T([\![w]\!]) \geq 2^{-|r|}$ (*any* random tape that extends r will yield the same output).

Dovetail the computation to get a computable enumeration of a prefix-free set of strings $\{r_i\}$ and their corresponding outputs $\{w_i\}$ (such that T outputs w_i when accessing only the bits in r_i). By taking

$$q_n := \sum_{i \leq n, w_i = w} 2^{-|r_i|}$$

we get an approximation of $P_T([\![w]\!])$ from below. Similarly,

$$a_n := 1 - \sum_{i \leq n, w_i \neq w} 2^{-|r_i|}$$

approximates it from above. This process can be done uniformly in w.

$\Rightarrow$: Let α a computable real with a computable approximation $\{q_n\}$. Suppose we want a PTM that halts a.s. and outputs a string w with probability α.

Given a random tape $r = (r_1, ..., r_n, ...) \in 2^{\mathbb{N}}$, perform the following:

Step n Compute $A_n(r) := \sum_{i=1}^{n} r_i \cdot 2^{-n}$.

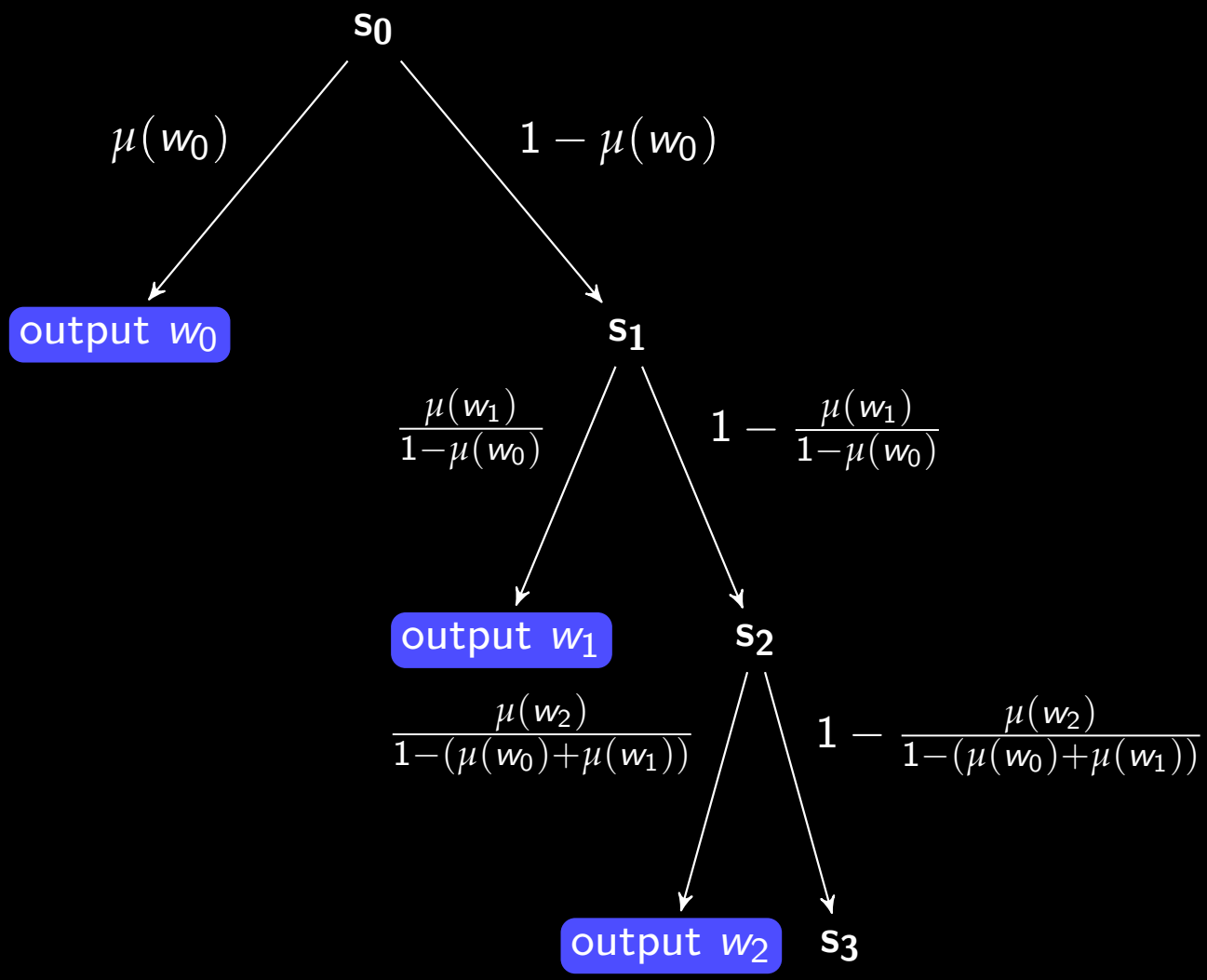
1. if $A_n(r) < q_n - 2^{-n}$: then HALT and OUTPUT w.
2. if $A_n(r) > q_n + 2^{-n}$: then HALT and OUTPUT $w0$.

else: set $\mathbf{n} := \mathbf{n} + \mathbf{1}$ and repeat previous step.

Then $A_\infty := \lim_{n\to\infty} A_n$ is uniformly distributed on the interval. The program returns w if and only if $A_\infty < \alpha$, which happens with probability α.

Note also the program halts with probability one (why?).

This can be done uniformly in (a computable approximation $\{q_n\}$ of) α.


s_0
$\mu(w_0)$
$1-\mu(w_0)$
output w_0
s_1
$\frac{\mu(w_1)}{1-\mu(w_0)}$
$1-\frac{\mu(w_1)}{1-\mu(w_0)}$
output w_1
s_2
$\frac{\mu(w_2)}{1-(\mu(w_0)+\mu(w_1))}$
$1-\frac{\mu(w_2)}{1-(\mu(w_0)+\mu(w_1))}$
output w_2
s_3

Computable measures: the continuous case

What about continuous distributions over more general spaces (standard Borel spaces: $\mathbb{R}^n$, spaces of probability measures,...)?

General idea: a computable object in a space is one that has a computable 'name': it can be computably approximated by a family of 'simple' objects.

- The 'simple' (or *ideal*) points need to be computably enumerable;
- We need a notion of distance on the space to capture approximations;
- The distances between the ideal points must be all uniformly computable;
- There needs to be enough 'simple' points to approximate all distances.

This gives rise to the notion of a *computable metric (Polish) space*.

Computable metric spaces

Computable metric space. A *computable metric space* is a triple (S, d, D), where $d : S \times S \to \mathbb{R}$ is a metric on the set S satisfying

(1) (S, d) is a complete metric space

(2) $D = \{s_i\}_{i \in \mathbb{N}}$ is an enumeration of a dense subset of S, called *ideal points*.

(3) the real numbers $d(s_i, s_j)$ are computable, uniformly in i and j.

Let $B(s_i, q_j)$ denote the ball of radius q_j centered at s_i. The collection

$$\mathcal{B}_S := \{B(s_i, q_j) \mid s_i \in D, q_i \in \mathbb{Q}, q_i > 0\}$$

is the set of *ideal* balls in S. Fix one canonical enumeration $\{B_j\}_{j \in \mathbb{N}}$ of the ideal balls.

Computable functions and random variables

Let (S, δ, D) and (T, δ_T, D_T) be computable metric spaces, the latter with the corresponding enumeration $\{B_i\}_{i\in\omega}$ of the ideal open balls in $\mathcal{B}_T$. A function $f : S \to T$ is said to be computable on $R \subseteq S$ when there is a computable sequence $\{U_n\}_{n\in\omega}$ of c.e. open sets $U_n \subseteq S$ such that

$$\forall n,\ f^{-1}[B_i] \cap R = U_n \cap R.$$

Such a sequence $\{U_n\}_{n\in\omega}$ is a *witness* to the computability of f.

Let S be a computable metric space. A random variable $X : 2^\omega \to S$ in S is a *computable random variable* when X is computable on some measure one subset.

The space of Borel measures over a computable metric space

The space $\mathcal{M}_1(S)$ of probability measures.
Let (S, δ, D_S) a computable metric space. We define the following metric space over $\mathcal{M}_1(S)$:

(1) The set D_P of ideal points consists of rational-valued probability measures that are concentrated on a finite subset of D_S.

(2) the metric is the *Prokhorov metric* given by

$$\delta_P(\mu, \nu) := \inf\{\epsilon > 0 \,|\, \forall A \in \mathcal{B}(S),\, \mu(A) \leq \nu(A^\epsilon) + \epsilon\},$$

where

$$A^\epsilon := \{p \in S \,|\, \exists q \in A, \delta_S(p, q) < \epsilon\} = \bigcup_{p \in A} B_\epsilon(p).$$

Computable measures: the general case

The set of ideal points is defined as follows: we have $\nu \in D_P$ if and only if $\nu = \sum_{i=1}^{k} q_i \delta_{t_i}$, for some rationals $q_i \geq 0$ such that $\sum_{i=1}^{k} q_i = 1$ and some points $t_i \in D_S$, where δ_{t_i} denotes the Dirac delta measure on t_i.

- If S is a computable metric space, so is $(\mathcal{M}_1(S), \delta_P, D_P)$.
- Under the Prokhorov metric, D_P is indeed a dense set in $\mathcal{M}_1(S)$.

Computable probability measure.
We say that $\mu \in \mathcal{M}_1(S)$ is a computable (Borel) probability measure on S when μ is a computable *point* in $\mathcal{M}_1(S)$ as a computable metric space.

Computable measures

Under this general sense of a computable measure, we can show:

A measure μ is computable if and only if $\mu(B_i)$ is a lower-semi computable real, uniformly in an enumeration of the B_i's.

In particular, each computable measure on $2^{\mathbb{N}}$ is a computable point in the corresponding metric space of measures over $2^{\mathbb{N}}$.

Another equivalence: a measure μ is computable if and only if the expectation $\mathbb{E}[f]$ of any bounded computable function $f : S \to [0, 1]$ is uniformly computable.

Computable Bayesianism

How much of the theory of Bayesian learning, and which foundational results, survive the passage to the computability-theoretic setting?

Conditioning and continuous measures

A fundamental question for (computational) Bayesian inference:

> Is Bayesian conditioning computable? We want an algorithm that can compute conditional probabilities for computable joint distributions.

Say we have two random variables X and Y such that the joint distribution $P(X, Y)$ is computable, and we want to be able to compute $P(X \mid Y = y)$.

Conditioning and continuous measures

A fundamental question for (computational) Bayesian inference:

> Is Bayesian conditioning computable? We want an algorithm that can compute conditional probabilities for computable joint distributions.

Say we have two random variables X and Y such that the joint distribution $P(X, Y)$ is computable, and we want to be able to compute $P(X \mid Y = y)$.

Theorem (Ackerman, Freer, and Roy 2014)

NOPE.

There is no generic algorithm for Bayesian conditioning.

(However: conditioning *is* computable if we allow some noise!)

Conditioning and Computability

Let C, U and N be independent random variables, where

- U has uniform distribution on $[0, 1]$.
- C is a fair coin (Bernoulli on $\{0, 1\}$ with p=1/2).
- N is a geometric r.v in $\mathbb{N}$: that is, $\mathbf{P}(\mathsf{N} = n) = 1/(2^{n+1})$ for any n.

Fix an computable enumeration $\{r_i\}_{i \in \mathbb{N}}$ of the rationals in $(0, 1)$. Now define the following random variable $\mathsf{X} : 2^{\mathbb{N}} \to [0, 1]$:

$$\mathsf{X} = \begin{cases} \mathsf{U} \text{ if } \mathsf{C} = 1, \\ r_{\mathsf{N}} \text{ otherwise.} \end{cases}$$

Then (every version of) $P[C = 1|X = \cdot]$ is discontinuous everywhere on every P_X measure-1 set: it is uncomputable.

Bayesian convergence to the truth and randomness

A Bayesian agent is interested in estimating the value of a random variable X. As they observe new bits of data, they update their expectation of the value of a random variable.

Theorem (Lévy's Upward Theorem, 1937)

Let $X : 2^{\mathbb{N}} \to \mathbb{R}$ an integrable random variable, relative to $(2^{\mathbb{N}}, \mathcal{B}, \mu)$. Then,

$$\lim_{n \to \infty} \mathbb{E}_\mu[X \mid \mathfrak{F}_n](\omega) = X(\omega)$$

for μ-almost every $\omega \in 2^{\mathbb{N}}$.

In words: with probability one, a Bayesian agent expects their estimates to converge to the correct value.

Bayesian convergence to the truth and randomness

But which *exactly* are the sequences of observations for which convergence to the truth occurs, given a prior?

Suppose you have a *computable* prior μ and are interested in learning the value of some class of effective random variables (e.g. the computable ones, or the lower-semi computable ones, etc.).

Then one can identify exactly the set of data sequences where convergence to the truth occurs: as it happens, these systematically turn out to be the *algorithmically random* sequences! [Huttegger, Walsh, Zaffora Blando, ms.].

Bayesian convergence to the truth and randomness

Algorithmically random: roughly, sequences that are can be characterised in either one of the following (equivalent!) ways:

- *disordered and incompressible*, with no computably identifiable patterns;
- *typical*: not computably distinguishable from most other sequences, satisfying all effectively specifiable probability-one laws;
- *unpredictable*, not gameable by computable gambling strategies.

Other applications of computable probability theory

- Statistics: exchangeability and Bayesian statistical models;
- Solomonoff induction;
- Computable analysis.

Wrapping up

- Probability as a (family of) logic(s)
- Inference: nonmonotonic reasoning and belief dynamics
- Logical languages and measure theory
- Random structures and limit laws
- Probability and computability

Logic and probability theory are intimately and intricately related, with deep connections at multiple levels, both technical and conceptual. Each illuminates the other, and there is much to gain by combining insights from both.

Thank you!