Journal of Computer and System Sciences 95 (2018) 98–108

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Algorithmic identification of probabilities is hard

Laurent Bienvenu [a,*], Santiago Figueira [b], Benoit Monin [c], Alexander Shen [a]

[a] LIRMM, CNRS & Université de Montpellier, 161 rue Ada, 34095 Montpellier Cedex 5, France
[b] Universidad de Buenos Aires and CONICET, Pabellón I – Ciudad Universitaria, Buenos Aires, Argentina
[c] LACL, Université Paris 12, 61 avenue du Général de Gaulle, 94010 Créteil Cedex, France

ARTICLE INFO

Article history:
Received 12 April 2017
Received in revised form 28 December 2017
Accepted 10 January 2018
Available online 15 March 2018

Keywords:
Algorithmic learning theory
Algorithmic randomness

ABSTRACT

Reading more and more bits from an infinite binary sequence that is random for a Bernoulli measure with parameter p, we can get better and better approximations of p using the strong law of large numbers. In this paper, we study a similar situation from the viewpoint of inductive inference. Assume that p is a computable real, and we have to eventually guess the program that computes p. We show that this cannot be done computably, and extend this result to more general computable distributions. We also provide a weak positive result showing that looking at a sequence X generated according to some computable probability measure, we can guess a sequence of algorithms that, starting from some point, compute a measure that makes X Martin-Löf random.

1. Introduction

1.1. Inductive inference

The study of learnability of computable sequences is concerned with the following problem. Suppose we have a black box that generates some infinite computable sequence of bits $X = X(0)X(1)X(2), \ldots$ We do not know the program running in the box, and want to guess it by looking at finite prefixes

$$X\restriction n = X(0)\ldots X(n-1)$$

for increasing values of n. There could be different programs that produce the same sequence, and it is enough to guess one of them (since there is no way to distinguish between them by just looking at the output bits). The more bits we see, the more information we have about the sequence. For example, it is hard to say something about a sequence seeing only that its first bit is a 1, but looking at the prefix

11001001000011111101101010100 0

one may observe that this is a prefix of the binary expansion of π, and guess that the machine inside the box does exactly that (though the machine may as well produce the binary expansion of, say, 47627751/15160384).

* Corresponding author.
E-mail address: laurent.bienvenu@computability.fr (L. Bienvenu).

https://doi.org/10.1016/j.jcss.2018.01.002
0022-0000/

The hope is that, as we gain access to more and more bits, we will *eventually* figure out how the sequence X is generated. More precisely, we hope to have a total computable function $\mathfrak{A}$ from strings to integers such that for every computable X, the sequence

$$\mathfrak{A}(X\restriction 1),\ \mathfrak{A}(X\restriction 2),\ \mathfrak{A}(X\restriction 3),\ldots$$

converges to a program (= index of a computable function) that computes X. This is referred to as *identification in the limit*, and can be understood in (at least) two ways. Indeed, assuming that we have a fixed effective enumeration $(\varphi_e)_{e\in\mathbb{N}}$ of partial computable functions from $\mathbb{N}$ to $\{0,1\}$, we can define two kinds of success for an algorithm $\mathfrak{A}$ on a computable sequence X:

- Strong success: the sequence $e_n = \mathfrak{A}(X\restriction n)$ converges to a single value e such that $\varphi_e = X$ (i.e., $\varphi_e(k) = X(k)$ for all k).
- Weak success: the sequence $e_n = \mathfrak{A}(X\restriction n)$ does not necessarily converge, but $\varphi_{e_n} = X$ for all sufficiently large n.

Here we assume that $\mathfrak{A}(X\restriction n)$ is defined for all n or at least for all sufficiently large n.

The strong type of success is often referred to as *explanatory* (EX), see, e.g., Definition VII.5.25 in [7, p. 116]. The second type is referred (see Definition VII.5.44, p. 131 in the same book) as *behaviorally correct* (BC). Note that it is obvious from the definition that strong success implies weak success.

It would be nice to have an algorithm that succeeds on all computable sequences. However, it is impossible even for weak success: for every (total) algorithm $\mathfrak{A}$, there is a computable X such that $\mathfrak{A}$ does not weakly succeed on X. The main obstacle is that certain machines are not total (produce only finitely many bits), and distinguishing total machines from non-total ones cannot be done computably.

However, some *classes* of computable sequences can be learned, i.e., there exists a total algorithm that succeeds on all elements of the class. Consider for example the class of primitive recursive functions. This class can be effectively enumerated, i.e., there is a total computable function f such that $(\varphi_{f(e)})_{e\in\mathbb{N}}$ is exactly the family of primitive recursive functions. Now consider the algorithm $\mathfrak{A}$ such that $\mathfrak{A}(\sigma)$ returns the smallest e such that $\varphi_{f(e)}(i) = \sigma(i)$ for all $i < |\sigma|$ (such an e always exists, since every string is a prefix of a primitive recursive sequence). It is easy to see that if X is primitive recursive, $\mathfrak{A}$ succeeds on X, even in the strong sense (EX).

The theory of learnability of computable sequences (or functions) is precisely about determining which classes of functions can be learned. This depends on the learning model, the type of success, of which there are many variants. We refer to the survey by Zeugman and Zilles [11] and to [7, Chapter VII] for a panorama of the field.

1.2. Learning measures

Recently, Vitányi and Chater [9] proposed to study a related problem. Suppose that instead of a sequence that has been produced by a deterministic machine, we are given a sequence that has been generated by a randomized algorithmic process, i.e., by a Turing machine that has access to a fair coin and produces some output sequence on the one-directional write-only output tape. The output sequence is therefore a random variable defined on the probabilistic space of fair coin tossings. We assume that this machine is *almost total*.[1] This means that the generated sequence is infinite with probability 1.

Looking at the prefix of the sequence, we would like to guess which machine is producing it. For example, for the sequence

$$0001111111100001100000000011111111111111$$

we may guess that it has been generated via the following process: start with 0 and then choose each output bit to be equal to the previous one with probability, say, $4/5$ (so the change happens with probability $1/5$), making all the choices independently.[2]

So what should count as a good guess for some observed sequence? Again, there is no hope to distinguish between two processes that have the same output distribution. So our goal should be to reconstruct the output distribution and not the specific machine.

But even this is too much to ask for. Assume that we have agreed that some machine M with output distribution μ is a plausible explanation for some sequence X. Consider another machine M' that starts by tossing a coin and then (depending on the outcome) either generates an infinite sequence of zeros or simulates M. If X is a plausible output of M, then X is also a plausible output of M', because it may happen (with probability $1/2$) that M' simulates M.

A reasonable formalization of a 'good guess' is provided by the theory of algorithmic randomness. As Chater and Vitányi recall, there is a widely accepted formalization of "plausible outputs" for an almost total probabilistic machine with output distribution μ: the notion of Martin-Löf random sequences with respect to μ. These are the sequences that pass all effective

[1] This requirement may look unnecessary. Still the notion of algorithmic randomness needed for our formalization is well-defined only for computable measures, and machines that are not almost total may not define a computable measure.

[2] The probability $4/5$ is not a dyadic rational number, but still can be simulated by an almost total machine using a fair coin.

statistical tests for the measure μ, also known as μ-Martin-Löf tests. (We assume that the reader is familiar with algorithmic randomness and Kolmogorov complexity. The most useful references for our purposes are [4] and [6].) Having this notion in mind, the natural way to extend learning theory to the probabilistic case is as follows:

A class of computable measures $\mathcal{M}$ is learnable if there exists a total algorithm $\mathfrak{A}$ such that for every sequence X that is Martin-Löf random for some measure in $\mathcal{M}$, the sequence

$$\mathfrak{A}(X\upharpoonright 1), \mathfrak{A}(X\upharpoonright 2), \mathfrak{A}(X\upharpoonright 3), \dots$$

identifies in the limit a measure $\mu \in \mathcal{M}$ such that X is Martin-Löf random with respect to μ.

Like in the classical case, there are several ways one can interpret the notion of 'identifying in the limit. We will come back to this after having introduced some basic notation and terminology related to computable measures (for now one may think of a computable measure as an output distribution of an almost total probabilistic machine).

1.3. Background and notation

We denote by 2^ω the set of infinite binary sequences and by $2^{<\omega}$ the set of finite binary sequences (or *strings*). The length of a string σ is denoted by $|\sigma|$. The empty string (string of length 0) is denoted by Λ. For two strings σ, τ we write $\sigma \preceq \tau$ if σ is a prefix of τ. The n-th element of a sequence $X(0)X(1)\dots$ is the value $X(n-1)$ (assuming that the length of X is at least n); the string $X\upharpoonright n = X(0)X(1)\dots X(n-1)$ is the *n-bit prefix of X*. We write $\sigma \preceq X$ if the string σ is a prefix of the infinite sequence X (i.e., $X\upharpoonright|\sigma| = \sigma$). The space 2^ω is endowed with the distance d defined by

$$d(X,Y) = 2^{-\min\{n : X(n)\neq Y(n)\}}.$$

This distance is compatible with the product topology generated by *cylinders*

$$[\sigma] = \{X \in 2^\omega : \sigma \preceq X\}.$$

A cylinder is both open and closed (= *clopen*). Thus, any finite union of cylinders is also clopen. It is easy to see, by compactness, that the converse holds: every clopen subset of 2^ω is a finite union of cylinders. We say that a clopen set C has *granularity at most n* if C is a finite union of some cylinders $[\sigma]$ with $|\sigma| = n$. We denote by Γ_n the family of clopen sets of granularity at most n.

We now give a brief review of the "computable analysis" aspects of the space of probability measures. For a more thorough exposition of the subject, the main reference is [4].

The space of Borel probability measures over 2^ω is denoted by $\mathcal{P}$. In the rest of the paper, when we talk about a 'measure', we mean an element of the space $\mathcal{P}$. This space is equipped with the weak topology, that is, the weakest topology such that for every σ, the application $\mu \mapsto \mu([\sigma])$ is continuous as a function from $\mathcal{P}$ to $\mathbb{R}$. Several classical distances are compatible with this topology; for example, one may use the distance ρ constructed as follows. For $\mu, \nu \in \mathcal{P}$, let $\rho_n(\mu,\nu)$ (for an integer n) be the quantity

$$\rho_n(\mu,\nu) = \max_{C\in\Gamma_n} |\mu(C) - \nu(C)|$$

and then set

$$\rho(\mu,\nu) = \sum_n 2^{-n}\rho_n(\mu,\nu).$$

The *open* (resp. *closed*) *ball $\mathcal{B}$ of center μ and radius r* is the set of measures ν such that $\rho(\mu,\nu) < r$ (resp. $\rho(\mu,\nu) \le r$). In the space of measures, the closure $\overline{\mathcal{B}}$ of the open ball $\mathcal{B}$ of center μ and radius r is the closed ball of center μ and radius r.

The space $\mathcal{P}$ is separable, i.e., has a countable dense set of points. An easily describable one is the set $\mathcal{I}$ consisting of measures $\{\delta_\sigma\}_{\sigma\in 2^{<\omega}}$, where δ_σ is the Dirac measure concentrated on the point $\sigma 0^\omega$, and all rational convex combinations of such measures. Note that every member of $\mathcal{I}$ has a finite description: it suffices to give the list of σ's together with the rational coefficients of the linear combination. Thus one can safely talk about computable functions from/to $\mathcal{I}$.

The set $\mathcal{I}$, together with the distance ρ, make $\mathcal{P}$ a computable metric space [4]. Each point $\mu \in \mathcal{P}$ can be written as the limit of a sequence $(q_1, q_2, \dots)$ of points in $\mathcal{I}$ where $\rho(q_i, q_j) \le 2^{-i}$ for $i < j$. Such a sequence is called a *fast Cauchy name* for μ. We say that a measure μ is *computable* if there is a total computable function $\varphi_e : \mathbb{N} \to \mathcal{I}$ such that $(\varphi_e(n))_{n\in\mathbb{N}}$ is a fast Cauchy name for μ. Such an e is called an *index for μ*.

At this point the way we view measures — as points of the space $\mathcal{P}$ — does not match the presentation of the introduction, where we asked the learning algorithm to guess, on prefixes of input X, a sequence of probabilistic machines M_i such that for almost all i, the machine M_i is almost total and X is a plausible output for M_i. The reason is that in fact there are three ways one can think of measures, which are equivalent for our purposes:

(a) A measure is a point of $\mathcal{P}$.
(b) By Caratheodory's theorem, a measure μ can be identified with the function $\sigma \mapsto \mu([\sigma])$: for every function $f : 2^{<\omega} \to [0,1]$ such that $f(\Lambda) = 1$ and $f(\sigma 0) + f(\sigma 1) = f(\sigma)$ there is a unique measure μ such that $\mu([\sigma]) = f(\sigma)$ for all σ. For example, the uniform measure λ is the unique measure such that $\lambda([\sigma]) = 2^{-|\sigma|}$ for all σ, and the Bernoulli measure β_p of parameter $p \in [0,1]$ is the unique measure satisfying $\beta_p([\sigma 1]) = p \cdot \beta_p([\sigma])$ for all σ.
(c) Consider a Turing functional M, which one might think of as a Turing machine with a read-only input tape, a work tape and a write-only output tape. We say that M is defined on X if M prints an infinite sequence Y on the output tape given X on the input tape. When M is defined on λ-almost every X, where λ is the uniform Lebesgue measure on a Cantor space that corresponds to the fair coin tossings, we say that M is *almost total*. Then the function

$$\mu_M(\sigma) = \lambda\{X : M(X) \succeq \sigma\}$$

defines a measure in the sense of item (b). This measure corresponds to the distribution of a random variable that is the output of M on the sequence of uniform independent random bits.

These approaches are equivalent both in the classical and effective realm, as is well known. The corresponding classes of measures coincide; moreover, one can computably convert an algorithm representing a computable measure according to one of the definition, into other representations. However, depending on the context one characterization may be much easier to handle than the others. And indeed, the techniques of the next section where we will prove our main negative result are of analytic nature, so characterization (a) will be more convenient, while the positive result of the last section has a more 'algorithmic flavor', for which characterization (c) will be better suited.

The *randomness deficiency*[3] function $\mathbf{d}$ is the largest, up to additive constant, function $f : 2^\omega \times \mathcal{P} \to \mathbb{N} \cup \{\infty\}$ such that

- f is lower semi-computable (i.e., $f^{-1}((k,\infty])$ is an effectively open[4] subset of the product space $2^\omega \times \mathcal{P}$, uniformly in k);
- for every $\mu \in \mathcal{P}$, for every integer k, the inequality $\mu\{X : f(X,\mu) > k\} < 2^{-k}$ holds.

We use the usual notation $\mathbf{d}(X \mid \mu)$ instead of $\mathbf{d}(X,\mu)$. We say that X is (uniformly) random relative to measure μ if $\mathbf{d}(X \mid \mu) < \infty$. For computable measures this notion coincides with the classical notion of Martin-Löf randomness.

We end this introduction with a discussion on a concept we will need to state the main theorem of Section 2: orthogonality. Two measures $\mu, \nu \in \mathcal{P}$ are said to be *orthogonal* if there is a Borel set $\mathcal{X} \subseteq 2^\omega$ such that $\mu(\mathcal{X}) = 1$ and $\nu(\mathcal{X}) = 0$ (taking the complement of $\mathcal{X}$, we see that orthogonality is a symmetric relation). This is equivalent to the following condition: for each $\varepsilon > 0$ there is a set $\mathcal{X}_\varepsilon$ such that $\mu(\mathcal{X}_\varepsilon) \geq 1 - \varepsilon$ and $\nu(\mathcal{X}_\varepsilon) < \varepsilon$ (indeed, one can then take $\mathcal{X} = \bigcap_i \bigcup_j \mathcal{X}_{2^{-i-j}}$).

The class of Bernoulli measures provides an easy example of orthogonality: if $p \neq q$, the Bernoulli measures β_p and β_q (see the definition above) are orthogonal (by the law of large numbers, taking for $\mathcal{X}$ the set of sequences with a limit frequency of ones equal to p, we have $\beta_p(\mathcal{X}) = 1$ and $\beta_q(\mathcal{X}) = 0$).

The important fact we need is that when two computable measures μ and ν are orthogonal, they share no random element, i.e., $\mathbf{d}(X \mid \mu)$ and $\mathbf{d}(X \mid \nu)$ cannot both be finite for any X. For a proof of this result, see for example [2].

1.4. Learning models

Most classical learning models for computable sequences can be adapted to our probabilistic setting. For example, the EX and BC models mentioned have the following natural counterparts (we give them the same names, as this should create no confusion).

Definition 1.1. Let $X \in 2^\omega$ and $\mathfrak{A} : 2^{<\omega} \to \mathbb{N}$ a total algorithm. We say that:

- $\mathfrak{A}$ EX-succeeds on X if $\mathfrak{A}(X\restriction n)$ converges to a value e that is an index for a computable measure μ with respect to which X is Martin-Löf random.
- $\mathfrak{A}$ BC-succeeds on X if there exists a computable measure μ such that for almost all n, $\mathfrak{A}(X\restriction n)$ is an index for μ and X is Martin-Löf random with respect to μ.

There are also some natural learning models we can define that are more specific to the probabilistic setting. As we discussed, for a given X that is Martin-Löf random with respect to some computable measure, there are several (actually, infinitely many) computable measures with respect to which X is Martin-Löf random. Thus we could allow the learner to propose different measures at each step and not converge to a specific measure, as long as almost all of them are good

[3] This version of randomness deficiency function is sometimes called "uniform probability-bounded randomness deficiency"; however, we do not use the other versions and call it just "randomness deficiency".
[4] An effectively open set is a union of a computably enumerable set of rational balls (or products of balls, since we consider a product space).

explanations for the observed X. To measure how good an explanation is, we use the randomness deficiency, thus it makes sense to make the distinction between learning with bounded randomness deficiency and with unbounded randomness deficiency.

Definition 1.2. Let $X \in 2^\omega$ and let $\mathfrak{A} : 2^{<\omega} \to \mathbb{N}$ be a total algorithm. We say that:

- $\mathfrak{A}$ BD-succeeds on X if there exists a constant d such that for almost all n, $\mathfrak{A}(X\upharpoonright n)$ is an index for a computable measure with respect to which X is Martin-Löf random, with randomness deficiency at most d.
- $\mathfrak{A}$ UD-succeeds on X if for almost all n, $\mathfrak{A}(X\upharpoonright n)$ is an index for a computable measure with respect to which X is Martin-Löf random.

('BD' and 'UD' stand for 'bounded deficiency' and 'unbounded deficiency'). Our four learning models are by no means an exhaustive list of possibilities. Just like the classical learning theory offers a wide variety of models, one could define a wealth of alternative models (partial learning, team learning, etc.) in our setting. This would take us far beyond the scope of the present paper and we leave this for further investigation.

Let us note in passing that the four learning models we have presented form a hierarchy, namely:

$$\text{EX-success} \Rightarrow \text{BC-success} \Rightarrow \text{BD-success} \Rightarrow \text{UD-success}$$

The fact that EX-success implies BC-success and that BD-success implies UD-success is immediate from the definition. To see that BC-success implies BD-success, recall that in our definition the randomness deficiency depends only on the measure but not on the algorithm that computes it. So if the learning algorithm BC-succeeds on some sequence X, i.e., outputs the same measure (its code) for all sufficiently large prefixes of X, then the deficiency of X with respect to this measure will be a constant and therefore the algorithm BD-succeeds on X.

2. Identifying measures is hard

Now that we have given a precise definition of various learning models for computable probability measures, there are some obvious questions we need to address, the first of which is: For each of the above learning models, is there a single algorithm $\mathfrak{A}$ that succeeds on all sequences X that are random with respect to some computable measure? This measure can be different for different X.) And if not, are there natural classes of measures for which there is an algorithm which succeeds on all X that are random with respect to some measure in this class?

The starting point of this paper was a claim made in a preprint of Vitányi and Chater [8], where it was stated that there exists an algorithm $\mathfrak{A}$ that EX-succeeds on every X that is Martin-Löf random with respect to a Bernoulli measure β_p for some computable p (different for different X). Our results (Theorems 2.2 and 2.3) imply that there is in fact no such algorithm. Vitányi and Chater later corrected this claim and proved the following weaker statement.

Theorem 2.1 *(Vitányi–Chater [9]). Let (p_e) be a partial enumeration of computable reals in $[0,1]$. If $E \subseteq \mathbb{N}$ is c.e. or co-c.e., and for all $e \in E$, p_e is defined, then there exists an algorithm $\mathfrak{A}$ that EX-succeeds on every X that is random with respect to some β_{p_e} for some $e \in E$.*

This result implies, for example, that there is an algorithm that EX-succeeds on all X that are random with respect to some β_q with q a rational number.

We prove that this result cannot be extended to all computable parameters p:

Theorem 2.2. *No algorithm $\mathfrak{A}$ can BD-succeed on every sequence X that is random with respect to some Bernoulli measure β_p for computable p. A fortiori, there is no algorithm $\mathfrak{A}$ can BD-succeeds on every sequence X that is random with respect to some computable measure.*

We will in fact prove a more general theorem, replacing the class of Bernoulli measures by any class of measures having some "reasonable" structural properties, and allowing the learning algorithm to succeed on a fraction of sequences only.

Theorem 2.3. *Let $\mathcal{M}$ be a subset of $\mathcal{P}$ with the following properties*:

- *$\mathcal{M}$ is effectively closed, i.e., its complement is effectively open: one can enumerate a sequence of rational open balls in $\mathcal{P}$ whose union is the complement of $\mathcal{M}$.*
- *$\mathcal{M}$ is computably enumerable, i.e., one can enumerate all rational open balls in $\mathcal{P}$ that intersect $\mathcal{M}$.*
- *for every computable measure ν, and every non-empty open subset of $\mathcal{M}$ (i.e., a non-empty intersection of an open set in $\mathcal{P}$ with $\mathcal{M}$) there is a computable μ in this open subset that is orthogonal to ν.*

Let also δ be a positive number. Then there is no algorithm $\mathfrak{A}$ such that for every computable $\mu \in \mathcal{M}$, the μ-measure of sequences X on which $\mathfrak{A}$ BD-succeeds is at least δ.

The notion of a computably (= recursively) enumerable closed set is standard in computable analysis, see [10, Definition 5.1.1].

Note that the hypotheses on the class $\mathcal{M}$ are not very restrictive: many standard classes of probability measures have these properties. In particular, the class $\{\beta_p : p \in [0,1]\}$ of Bernoulli measures is such a class, which is why. So we get Theorem 2.2 as a corollary: there is no algorithm that can learn all Bernoulli measures (not to speak about all Markov chains). To see that the third condition is true for the class of Bernoulli measures, note that only countably many Bernoulli measures may be non-orthogonal to a given measure μ: the sets L_p of sequences with limit frequency p are disjoint, so only countably many of them may have positive μ-measure. It remains to note that every open non-empty subset of the class of Bernoulli measures has the cardinality of the continuum.

Let us give another example (beyond Bernoulli measures and Markov chains) that satisfies the requirements of Theorem 2.3. In this example, the probability of the next bit to be 1 may depend on many of the previous bits. For every parameter $p \in [0,1]$, consider the measure μ_p associated to the stochastic process that generates a binary sequence bit by bit as follows: the first bit is 1, and the conditional probability of 1 after $\sigma 10^k$ is $p/(k+1)$. One can check that the class $\mathcal{P} = \{\mu_p : p \in [0,1]\}$ satisfies the hypotheses of the theorem (observe that p can easily be reconstructed from the sequence that is random with respect to μ_p).

Note also that these hypotheses are not added just for convenience: although they might not be optimal, they cannot be outright removed. If we do not require the class $\mathcal{M}$ to be effectively closed, compactness, then the class of Bernoulli measures β_p with *rational* parameter p would qualify, but Vitányi and Chater's theorem tells us that there *is* an algorithm that correctly identifies each of the measures in the class with probability 1. The third condition is important, too. Consider the measures β_0 and β_1 concentrated on the sequences $0000\ldots$ and $1111\ldots$ respectively. Then the class $\mathcal{M} = \{p\beta_0 + (1-p)\beta_1 \ : \ p \in [0,1]\}$ is indeed effectively closed and computably enumerable, but it is obvious that there is an algorithm that succeeds with probability 1 for all measures of that class (in the strongest sense: the first bit determines the entire sequence). For the second condition we do not have a counterexample showing that it is really needed, but it is true for all the natural classes (and it is guaranteed to be true if $\mathcal{M}$ has a computable dense sequence).

The rest of this section is devoted to the proof of Theorem 2.3.

Fix a subset $\mathcal{M}$ of $\mathcal{P}$ satisfying the hypotheses of the theorem, and some $\delta > 0$. Assume for the sake of contradiction that there is a total algorithm $\mathfrak{A}$ such that for every computable $\mu \in \mathcal{M}$, the μ-measure of sequences X on which $\mathfrak{A}$ BD-succeeds is at least δ. In the rest of the proof, by "success" we always mean BD-success.

We may assume without loss of generality that our algorithm $\mathfrak{A}$, on an input σ, outputs an integer e which is a code for a partial computable function φ_e from $\mathbb{N}$ to $\mathcal{I}$ (our set of rational points in $\mathcal{P}$, described above) that is defined on the entire $\mathbb{N}$ or at some initial segment of $\mathbb{N}$, and $\rho(\varphi_e(n), \varphi_e(n+1)) < 2^{-n-1}$ when both $\varphi_e(n)$ and $\varphi_e(n+1)$ are defined. When this sequence is total, it converges to a measure μ with computable speed: $\rho(\varphi_e(n), \mu) \leq 2^{-n}$.

This is not guaranteed by the definition of BD-success, but we may "trim" the algorithm by ensuring that indeed the sequence $\varphi_e(0), \varphi_e(1), \ldots$, whether finite or infinite, contains elements of $\mathcal{I}$ and satisfies the distance conditions where defined (by waiting until the conditions are checked; note that we have a strict inequality which will manifest itself at some moment, if true).

Suppose now that for some index e, we do not know whether φ_e is total, but we see that $\varphi_e(n)$ is defined for some n, and $\mathbf{d}(X \,|\, \nu) > d$ holds for some X and for *all* measures ν at distance $\leq 2^{-n}$ of $\varphi_e(n)$. Then we already know, *should φ_e be total and converge to some μ*, that $\mathbf{d}(X \,|\, \mu) > d$. Thus we use the following notation: if $\mathfrak{A}(\sigma)$ returns e, then $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma))$ is the quantity

$$\sup\{d \,|\, \exists n\, \varphi_e(n) \downarrow \text{ and } \mathbf{d}(X \,|\, \nu) > d \text{ for all } \nu \text{ at distance} \leq 2^{-n} \text{ from } \varphi_e(n)\}.$$

The supremum of an empty set (that appears, for example, if $\mathfrak{A}(\sigma)$ is a code of an empty sequence) is considered to be 0. Our function $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma))$ has two key properties which are essential for the rest of our proof:

(a) If $\mathfrak{A}(\sigma) = e$, and φ_e is total and converges to μ, then $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma)) = \mathbf{d}(X \,|\, \mu)$.
(b) $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma))$ is lower semi-computable, uniformly in (X, σ).

Let us first prove that property (a) holds. Assuming $\mathfrak{A}(\sigma) = e$, and φ_e is total and converging to μ, we know that μ is at distance $\leq 2^{-n}$ of $\varphi_e(n)$ for all n. Thus, in the definition of $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma))$ every member of the set of d on the right-hand side is at most $\mathbf{d}(X \,|\, \mu)$. Thus $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma)) \leq \mathbf{d}(X \,|\, \mu)$. On the other hand, if $\mathbf{d}(X \,|\, \mu) > d$, by lower semicontinuity of the function $\mathbf{d}$, there is n such that $\rho(\nu, \mu) \leq 2^{-n}$ implies $\mathbf{d}(X \,|\, \nu) > d$, and thus $\mathbf{d}(X \,|\, \mathfrak{A}(\sigma)) > d$. Property (a) is proven.

For property (b), we use the fact that the space $\mathcal{P}$ is effectively compact (one can effectively enumerate all covers of $\mathcal{P}$ consisting of a finite union of open rational balls), together with the fact that the infimum of a lower semicomputable function on an effectively compact set is lower semicomputable, uniformly in a code for the effectively compact set (see [4] for both facts). Thus, the predicate "$\mathbf{d}(X \,|\, \nu) > d$ for all ν at distance $\leq 2^{-n}$ from $\varphi_e(n)$" is computably enumerable uniformly

in e, n, X, d (or said otherwise, the set of (e, n, X, d) satisfying this property is an effectively open subset of $\mathbb{N} \times \mathbb{N} \times 2^\omega \times \mathbb{N}$). Property (b) follows.

Thanks to property (a), when $\mathfrak{A}(\sigma) = e$ and φ_e is total and converges to a measure μ, we can safely identify $\mathfrak{A}(\sigma)$ with μ and write $\mathfrak{A}(\sigma) = \mu$. Additionally, we say that "$\mathfrak{A}(\sigma)$ is a measure" when $\mathfrak{A}(\sigma) = \mu$ for some measure μ.

Now, for every pair of integers (N, d), we define the set

$$\textsc{Wrong}(N, d) = \left\{ X \middle| (\exists n \geq N)\ \mathbf{d}(X \mid \mathfrak{A}(X\restriction n)) > d \right\}.$$

This is the set of sequences X on which the algorithm is "visibly wrong" at some prefix of length $n \geq N$, for the deficiency level d.

Note that $\textsc{Wrong}(N, d)$, understood in this way, is effectively open uniformly in (N, d) and is non-increasing in each of its parameters. The intersection of sets $\textsc{Wrong}(N, d)$ for all N and d is some set Wrong; as the name says, the algorithm $\mathfrak{A}$ cannot BD-succeed on any sequence in this set. (Note that other reasons for failure are possible, e.g., $\mathfrak{A}$ may not provide a measure on prefixes of some sequence.)

It is technically convenient to combine the two parameters N and d into one (even they are of different nature) and consider a decreasing sequence of sets $\textsc{Wrong}(N) = \textsc{Wrong}(N, N)$ whose intersection is Wrong.

We also consider a set $\textsc{Succ}(N, d)$ of all sequences X such that $\mathfrak{A}$ BD-succeeds on X at level N with deficiency d, i.e.,

$$\textsc{Succ}(N, d) = \{X\colon\ (\forall n \geq N)\ [\mathfrak{A}(X\restriction n) \text{ is a measure}, \mathbf{d}(X \mid \mathfrak{A}(X\restriction n)) \leq d]\}.$$

The set $\textsc{Succ}(N, d)$ is a closed set. Indeed, it is an intersection of sets indexed by n, so we need to show that each of them is closed. For each n there are finitely many possible prefixes of length n, so the first condition ("$\mathfrak{A}(X\restriction n)$ is a measure") defines a clopen set. The second condition defines an effectively closed subset in each cylinder where $\mathfrak{A}(X\restriction n)$ is a measure. (Note that we do *not* claim that $\textsc{Succ}(N, d)$ is *effectively* closed, since the condition "to be a measure" is only a Π_2-condition.) By definition, the set $\textsc{Succ}(N, d)$ does not intersect the set $\textsc{Wrong}(N, d)$.

The set $\textsc{Succ}(N, d)$ increases as N or d increase; the union of these sets is the set of all X where $\mathfrak{A}$ BD-succeeds; we denote it by Succ. Again we may combine the parameters and consider an increasing sequence of sets $\textsc{Succ}(N) = \textsc{Succ}(N, N)$ whose union in Succ.

All these considerations deal with the space of sequences. Now we switch to the space of measures and the class $\mathcal{M}$. We look what are the measures of sets $\textsc{Wrong}(N)$ according to different $\mu \in \mathcal{M}$. Consider some threshold $x \in [0, 1]$. There are two possible cases:

- there exist some number N, and some non-empty open set $\mathcal{U} \subseteq \mathcal{M}$ such that $\mu(\textsc{Wrong}(N)) \leq x$ for all $\mu \in \mathcal{U}$;
- for every N the set of points $\mu \in \mathcal{M}$ where $\mu(\textsc{Wrong}(N)) > x$ is dense in $\mathcal{M}$.

There is some threshold where we switch from one case to the other, so let use take close values of $p < q$ (i.e., we take the difference $q - p$ to be much smaller than δ from the statement of the theorem; it would be enough to require that $q - p < \delta/10$) such that the first case happens for q and the second one happens for p.

Choose some N and an open ball $\mathcal{B}_0$ that has a non-empty intersection with $\mathcal{M}$ such that $\mu(\textsc{Wrong}(N)) \leq q$ for all $\mu \in \mathcal{B}_0 \cap \mathcal{M}$ (this is possible since the first case happens for q).

Lemma 2.4. *There exists a computable measure $\mu^* \in \mathcal{B}_0 \cap \mathcal{M}$ such that $\mu^*(\textsc{Wrong}) \geq p$.*

Proof. Since the second case happens for p, we can find some $\mu \in \mathcal{B}_0 \cap \mathcal{M}$ such that $\mu(\textsc{Wrong}(0)) > p$. Since $\textsc{Wrong}(0)$ is open, the same is true for some its clopen subset C, i.e., $\mu(C) > p$. Note that $\mu(C)$ is a continuous function of μ for fixed clopen C, so we can find a smaller ball $\mathcal{B}_1 \subseteq \mathcal{B}_0$ intersecting $\mathcal{M}$ such the $\mu(\textsc{Wrong}(0)) \geq \mu(C) > p$ for all $\mu \in \overline{\mathcal{B}}_1 \cap \mathcal{M}$. Then, repeating the same argument, we find an even smaller ball $\mathcal{B}_2 \subseteq \mathcal{B}_1$ intersecting $\mathcal{M}$ such that $\mu(\textsc{Wrong}(1)) > p$ for all $\mu \in \overline{\mathcal{B}}_2 \cap \mathcal{M}$, then some $\mathcal{B}_3 \subseteq \mathcal{B}_2$ such that $\mu(\textsc{Wrong}(2)) > p$ for all $\mu \in \overline{\mathcal{B}}_3 \cap \mathcal{M}$, etc. Using the completeness of the space of measures, consider the intersection point μ^* of all $\mathcal{B}_i$ (we may assume that their radii converge to 0 and that $\overline{\mathcal{B}}_{i+1} \subseteq \mathcal{B}_i$, and this guarantees the existence and the uniqueness of the intersection point). We have $\mu^*(\textsc{Wrong}(i)) > p$ for all i (but $\mu^*(\textsc{Wrong}(N)) \leq q$; the same in true for all subsequent sets $\textsc{Wrong}(i)$ for all $i \geq N$). The continuity property for measure μ^* then guarantees that $\mu^*(\textsc{Wrong}) \geq p$.

Refining this argument, we can get a *computable* measure μ^* with this property. Indeed, we may choose $\mathcal{B}_{i+1}$ in such a way that even the closed ball $\overline{\mathcal{B}}_{i+1}$ of the same radius is contained in $\mathcal{B}_i$; this property is enumerable. "To have a non-empty intersection with $\mathcal{M}$" is also an enumerable property (by assumption), and "$\mu(\textsc{Wrong}(i)) > p$ for all $\mu \in \overline{\mathcal{B}}_{i+1}$" is also an enumerable property (we may assume without loss of generality that p is rational, and $\textsc{Wrong}(i)$ is effectively open uniformly in i). So we can perform a search until $\mathcal{B}_{i+1}$ is found, and the sequence of $\mathcal{B}_i$ is computable, so μ^* is computable. □

Now the argument goes as follows. Since μ^* is computable, the set Succ should have μ^*-probability at least δ by assumption. Success means that (at least) some measures are provided by the learning algorithm $\mathfrak{A}$ for prefixes of sufficiently large length M. There are finitely many possible prefixes, and they correspond to finitely many computable measures

$\mu_1, \dots, \mu_s$. Then we choose a measure μ' orthogonal to all these measures and very close to μ^*. We get the contradiction showing that $\mu'(\text{WRONG}(N))$ is almost $p + \delta$ (or more) and therefore exceeds q which is not possible due to the choice of $\mathcal{B}_0$. To get the δ-increase we use the fact that sequences that are μ'-random cannot be μ_i-random and should therefore have infinite deficiency. Let us now explain this argument in details.

Recall that we have chosen N in such a way that $\mu(\text{WRONG}(N)) \le q$ for all μ sufficiently close to μ^*. On the other hand, $\mu^*(\text{WRONG}(M)) \ge \mu^*(\text{WRONG}) \ge p$ for all M.

Since $\mu^*(\text{SUCC}) \ge \delta$, the continuity property of measures guarantees that $\mu^*(\text{SUCC}(M)) \gtrsim \delta$ for sufficiently large M, where $\gtrsim$ means inequality up to an additive error term that is very small compared to δ (in fact, $\delta/10$ would be small enough; we do not add more than 10 inequalities of this type). Fix some M that is large enough and greater than N from the previous paragraph.

The set $\text{WRONG}(M)$ is open and has μ^*-measure at least p. Therefore, there exist a clopen set $C \subseteq \text{WRONG}(M)$ such that $\mu^*(C) \gtrsim p$. Since the set C is clopen, there exists some $K \ge M$ such that the K-bit prefix determines whether a sequence belongs to C (the granularity of C is at most K).

Now the Cantor space is split into 2^K intervals that correspond to different prefixes of length K. Some of these intervals form the set C (and belong to $\text{WRONG}(M)$ entirely). Among the rest, we distinguish good and bad intervals; good intervals correspond to prefixes for which the learning algorithm $\mathfrak{A}$ produces a measure (whatever this measure is). Let $\mu_1, \dots, \mu_s$ be all measures that are produced by $\mathfrak{A}$ for all good intervals (we have at most 2^K of them).

Note that $\text{SUCC}(M)$ is covered by the good intervals. Indeed, it is disjoint with $\text{WRONG}(M)$ and therefore with C, and also is disjoint with bad intervals by definition (since $K \ge M$, the algorithm $\mathfrak{A}$ should produce a measure when applied to K-bit prefix).

Now consider a measure μ' that is very close to μ and orthogonal to all μ_i. (Our assumption allows us to get a measure very close to μ and orthogonal to a given computable measure; now we have several measure $\mu_1, \dots, \mu_s$ instead of one, but this does not matter since we may consider their average: any measure orthogonal to the average is orthogonal to all μ_i.)

Since the μ^*-measure of $\text{SUCC}(M)$ is almost δ (or more), and it is covered by good intervals, then μ^*-measure of the union of good intervals is also almost δ (or more). The same is true for every measure μ' sufficiently close to μ^* since the union of good intervals is a clopen set.

No μ'-random sequences can be μ_i-random since the measures are orthogonal. This implies infinite deficiency, so all μ'-random sequences in good intervals belong to $\text{WRONG}(M)$. So the μ'-measure of the part of $\text{WRONG}(M)$ outside C is almost δ (or more), and the part of $\text{WRONG}(M)$ inside C has μ'-measure almost p or more (this was true for μ, and μ' is close to μ). Together we get lower bound close to $p + \delta$ for $\mu'(\text{WRONG}(M))$. And this gives us a contradiction, since $\mu'(\text{WRONG}(M)) \le \mu'(\text{WRONG}(N))$, and the latter should be at most q for all μ' close to μ. (Recall that we have chosen $q - p$ much smaller than δ.)

This contradiction finishes the proof of Theorem 2.3.

3. Removing the deficiency boundedness requirement: a positive result

We have established in Theorem 2.2 that, unsurprisingly, there is no total algorithm $\mathfrak{A}$ that BD-succeeds on all sequences X that are random with respect to some computable probability measure. After proving this theorem, the authors conjectured that one could even prove the same result for UD-success. But it turns out that the situation is drastically different for UD-learning: we will show in this section that there *is* a uniform learning algorithm in this model.

Theorem 3.1. *There exists a total algorithm $\mathfrak{A}$ that UD-succeeds on every X that is Martin-Löf random with respect to some computable probability measure.*

Recall that this means that for large enough n, $\mathfrak{A}(X\upharpoonright n)$ is a (code of a) measure with respect to which X is random. However, (*a*) $\mathfrak{A}(X\upharpoonright n)$ may be different for different values of n, and (*b*) the randomness deficiency of X with respect to $\mathfrak{A}(X\upharpoonright n)$ is unbounded.

The proof of this result is inspired by a result of Harrington (reported in [3, Theorem 3.10] or [7, Theorem VII.5.55, p. 139]) which states that there exists an algorithm to learn — in the classical sense — all computable sequences *up to finitely many errors*. More precisely, there is a total algorithm $\mathfrak{A}$ such that for every computable sequence X, for almost all n, $\mathfrak{A}(X\upharpoonright n)$ is a program for an almost everywhere defined function that differs from X only in finitely many places. Indeed, let $\mathfrak{A}(\sigma)$ be the program that, given input m, spends time m searching for the minimal program computing some extension of σ and then runs this program, if found, on m (and returns, say, 0 if no such program is found). Let e be the minimal program that computes X. All smaller programs fail to compute X on some k (by either being undefined or giving a wrong answer). If n is greater than all these k, then none of the smaller programs (than e) would qualify as a candidate for any m, and for large enough m the program e will be approved. (Note that $\mathfrak{A}(\sigma)$ may be a non-total program even if σ is long: we know only that it is defined on large enough values of m.)

Proof. We will use an argument somewhat similar to Harrington's to prove Theorem 3.1. In this section, it is more convenient to consider measures as functions by identifying μ with the function $\sigma \mapsto \mu(\sigma)$ (here and in the rest of the section, $\mu(\sigma)$ is the abbreviation of $\mu([\sigma])$).

It is also more convenient to use an alternative characterization of Martin-Löf randomness, via the Schnorr–Levin theorem, which states that if μ is a computable measure, a sequence X is Martin-Löf random with respect to μ if and only if the prefix complexity of its prefixes is big:

$$(\exists c)(\forall n)\ \mathrm{K}(X\upharpoonright n) > -\log \mu(X\upharpoonright n) - c$$

(see for example [6]). We say that measure μ is *exactly computable* when the function $\sigma \mapsto \mu(\sigma)$ is rational-valued and computable as a function from $2^{<\omega}$ to $\mathbb{Q}$. Of course, not all computable measures are exactly computable, but the following fact holds:

Lemma 3.2. *If $X \in 2^{\omega}$ is random with respect to a computable probability measure μ, it is random with respect to some exactly computable probability measure ν. Moreover, one can suppose that $\nu(\sigma) > 0$ for all strings σ.*

See [5] for a proof (essentially we approximate the given computable measure with enough precision using positive rational numbers).

This lemma is convenient because it is possible to effectively list the family $\mathcal{F}$ of partial computable functions μ from $2^{<\omega}$ to $\mathbb{Q}$ such that

- $\mu(\Lambda) = 1$;
- for every n, either $\mu(\sigma)$ is defined for all strings σ of length n, or is undefined for all strings σ of length n;
- if $\mu(\sigma 0)$ and $\mu(\sigma 1)$ are defined, $\mu(\sigma)$ is defined and is equal to $\mu(\sigma 0) + \mu(\sigma 1)$;
- $\mu(\sigma) > 0$ for all σ on which μ is defined.

Let $(\mu_e)_{e\in\mathbb{N}}$ be an effective enumeration of all the functions in $\mathcal{F}$. It is among these functions that, given a sequence X, we are going to look for the 'best candidate' μ_e such that μ_e is a measure (i.e., is total) and X is random relative to μ_e. Suppose we are given a prefix σ of X. What is a good candidate μ_e for this σ? For this, we use the same approach as algorithmic statistics: a good explanation μ_e for a string σ should (a) be defined on σ, (b) be simple, which is measured by the prefix complexity $\mathrm{K}(e)$, and (c) give σ a small 'local' randomness deficiency, which we measure by the quantity $\mathbf{ld}(e,\sigma) = \max_{\tau\preceq\sigma}[-\log \mu_e(\tau) - \mathrm{K}(\tau)]$, with the convention that $\mathbf{ld}(e,\sigma) = \infty$ when $\mu_e(\tau)$ is undefined for some prefix τ of σ. The Schnorr–Levin theorem mentioned above now can be reformulated as follows: the value

$$d(X \mid \mu_e) = \sup_{\tau\preceq X}[-\log \mu_e(\tau) - \mathrm{K}(\tau)] = \lim_n \mathbf{ld}(e, X\upharpoonright n) = \sup_n \mathbf{ld}(e, X\upharpoonright n)$$

is finite if and only if μ_e is a measure and X is Martin-Löf random with respect to μ_e. In fact, $d(X \mid e)$ is a version of randomness deficiency; for a fixed measure μ_e this quantity is equal to the deficiency $\mathbf{d}$ of the previous section up to logarithmic precision (see, e.g., [1] for details).

Returning to algorithmic statistics, we combine the two quantities into a score function

$$\mathbf{score}(e,\sigma) = \mathrm{K}(e) + \lceil \mathbf{ld}(e,\sigma)\rceil,$$

(as in golf, 'score' is meant in a negative sense: a high $\mathbf{score}(e,\sigma)$ means that μ_e is not a good candidate for being a measure with respect to which σ looks random). Finally, we define the function Best such that $\textsc{Best}(\sigma)$ is the value of e that minimizes $\mathbf{score}(e,\sigma)$ (if there are several, we let $\textsc{Best}(\sigma)$ be the smallest one). That is,

$$\textsc{Best}(\sigma) = \min\{e \mid (\forall e')\ \mathbf{score}(e,\sigma) \leq \mathbf{score}(e',\sigma)\}.$$

The first thing to observe is that Best is computable in the halting set $\mathbf{0}'$. Indeed, to compute $\textsc{Best}(\sigma)$, one can first find e such that $s = \mathbf{score}(e,\sigma) < \infty$ (this can be done computably). Then, using $\mathbf{0}'$, one can find N such that $\mathrm{K}(e) > s$ (and thus $\mathbf{score}(e,\sigma) > s$) for all $e > N$. Finally, take $\textsc{Best}(\sigma)$ to be the number e in $[0,N]$ that minimizes $\mathbf{score}(e,\sigma)$ (again taking the smallest one if there are several), which can be done effectively relative to $\mathbf{0}'$ because **score** is itself computable relative to $\mathbf{0}'$.

The core of the proof of Theorem 3.1 is the following lemma, which is of independent interest. It implies that learning measures in the EX sense, which we showed in the previous section to be impossible, becomes possible if one is given access to oracle $\mathbf{0}'$.

Lemma 3.3. *Let X be a sequence that is random with respect to some computable probability measure. The sequence of integers $\textsc{Best}(X\upharpoonright n)$ converges to a single value e^* such that μ_{e^*} is a measure, and X is random with respect to μ_{e^*}.*

Proof. Fix such a sequence X. For each e, the sequence $\mathbf{score}(e, X\restriction n)$ is nondecreasing and takes its values in $\mathbb{N}\cup\{\infty\}$, thus converges to some $S(e)\in\mathbb{N}\cup\{\infty\}$. As we have said, the Schnorr–Levin theorem guarantees that $S(e)<\infty$ if and only if μ_e is a measure and X is Martin-Löf random with respect to μ_e. Thus we know that $S(e)<\infty$ for some e by our assumption that X is Martin-Löf random with respect to some computable probability measure.

Let e^* be the index such that $S(e^*)$ is minimal among all $S(e)$ (the smallest one if there are several). For any i such that $K(i)>S(e^*)$, we have for any n:

$$\mathbf{score}(i, X\restriction n) > S(e^*) \geq \mathbf{score}(e^*, X\restriction n)$$

Thus $\textsc{Best}(X\restriction n)\neq i$ for any n. In other words, only the j such that $K(j)\leq S(e^*)$ matter when selecting the best candidate for the sequence X. Those j form a finite set. For all such j, we know that $\mathbf{score}(j, X\restriction n)$ is non-decreasing and eventually reaches its final value. After that, for all sufficiently large n, we have $\textsc{Best}(X\restriction n)=e^*$. □

At this point, we have seen that the function Best does achieve the learning of measures we want, but unfortunately this function is only $\mathbf{0}'$-computable. By Schoenfield's limit lemma, this means that there exists a computable procedure which, given σ, generates a sequence $e_0, e_1, \ldots$ of integers that converges to $e_\infty=\textsc{Best}(\sigma)$. There is in general no way to compute μ_{e_∞} from this sequence. However, what we *can* do is combine all the μ_{e_i} together (being cautious about the fact that some μ_{e_i} may be partial) into a single computable measure ν such that $\nu > c\mu_{e_\infty}$ for some $c>0$, and this, by the Schnorr–Levin theorem, guarantees that everything that is random with respect to μ_{e_∞}, is also ν-random.

More precisely, we have the following lemma.

Lemma 3.4. *Let $f: 2^{<\omega}\to\mathbb{N}$ be a total $\mathbf{0}'$-computable function such that $\mu_{f(\sigma)}$ is a measure for all σ. Then there is a computable function g such that $\mu_{g(\sigma)}$ is a measure for all σ, and $\mu_{g(\sigma)}\geq c_\sigma\mu_{f(\sigma)}$ for some positive c_σ.*

Proof. Consider the following effective procedure. On input σ, we use the Schoenfield limit lemma to effectively get a sequence e_i converging to $e_\infty=f(\sigma)$. Initially all e_i are considered "candidates". We then apply a filtering process that deletes some of these candidates. Recall that the corresponding μ_{e_i} are elements of $\mathcal{F}$. We compute in parallel all $\mu_{e_i}(\tau)$ for all pairs (i,τ) for which e_i is still a candidate. If we find two candidates e_i, e_j and τ such that $\mu_{e_i}(\tau)$ and $\mu_{e_j}(\tau)$ are both defined and not equal different from each other, then we remove e_i and e_j from the list of candidates. This way we ensure, since the sequence converges, that from some point on, for any candidate e_i, the corresponding function μ_{e_i} is equal to μ_{e_∞} on its domain (but μ_{e_i} may be partial). Indeed, each bad candidate (i.e., an e_i such that $e_i\neq e_\infty$) may destroy at most one good candidate, and by assumption almost all candidates are good.

Now we let $\mu_{g(\sigma)}$ to be a computable measure ν constructed in the following way. First, let $\nu(\Lambda)=1$. Then we compute the conditional probabilities $\nu(x0)/\nu(x)$ and $\nu(x1)/\nu(x)$ level by level. When computing them on level N, we use for the computation the conditional probabilities for some candidate that remains alive after N steps of filtering process. (Any of them could be used, for example, we may take the one with smallest computation time. Note the at least one good candidate remains, so we will not wait forever.)

As we have seen, starting from some level only good candidates remain, so the conditional probabilities above this level are the same for $\mu_{f(\sigma)}$ and ν. Since by assumption all values of all measures are positive, this guarantees the required inequality. □

We can now put all pieces together to prove Theorem 3.1. Applying the previous lemma to $f=\textsc{Best}$, we have a computable function g such that for every σ, the measure $\mu_{g(\sigma)}$ dominates, up to a multiplicative constant, the measure $\mu_{\textsc{Best}(\sigma)}$. For every X that is random with respect to some computable measure, we know, by Lemma 3.3, that $\mu_{\textsc{Best}(X\restriction n)}$ is eventually constant and equal to a measure with respect to which X is random. This measure is dominated (up to multiplicative constant) by $\mu_{g(X\restriction n)}$, thus X is also random with respect to $\mu_{g(X\restriction n)}$ (change in the measure increases the deficiency at most by $O(1)$). This finishes the proof. □

Acknowledgments

Laurent Bienvenu and Santiago Figueira acknowledge the support of the Laboratoire International Associé "INFINIS". Laurent Bienvenu and Alexander Shen also acknowledge the support of ANR-15-CE40-0016-01 RaCAF grant.

References

[1] L. Bienvenu, P. Gács, M. Hoyrup, C. Rojas, A. Shen, Algorithmic tests and randomness with respect to a class of measures, Proc. Steklov Inst. Math. 274 (2011) 34–89.
[2] L. Bienvenu, W. Merkle, Constructive equivalence relations for computable probability measures, Ann. Pure Appl. Logic 160 (2009) 238–254.
[3] J. Case, C. Smith, Comparison of identification criteria for machine inductive inference, Theor. Comput. Sci. 25 (1983) 193–220.
[4] P. Gács, Uniform test of algorithmic randomness over a general space, Theor. Comput. Sci. 341 (1–3) (2005) 91–137.
[5] D. Juedes, J. Lutz, Weak completeness in E_1 and E_2, Theor. Comput. Sci. 143 (1) (1995) 149–158.

[6] M. Li, P. Vitányi, An Introduction to Kolmogorov Complexity and Its Applications, 3rd edition, Texts in Computer Science, Springer-Verlag, New York, 2008.
[7] P. Odifreddi, Classical Recursion Theory: Volume II, Elsevier, 1999.
[8] P. Vitányi, N. Chater, Algorithmic identification of probabilities, https://arxiv.org/abs/1311.7385v1, 2013.
[9] P.M. Vitányi, N. Chater, Identification of probabilities, J. Math. Psychol. 76 (Part A) (2017) 13–24.
[10] K. Weihrauch, Computable Analysis, Springer, Berlin, 2000.
[11] T. Zeugmann, S. Zilles, Learning recursive functions: a survey, Theor. Comput. Sci. 397 (2008) 4–56.