

ABSOLUTELY NO FREE LUNCHES!

GORDON BELOT

ABSTRACT. This paper is concerned with learners who aim to learn patterns in infinite binary sequences: shown longer and longer initial segments of a binary sequence, they either attempt to predict whether the next bit will be a 0 or will be a 1 or they issue forecast probabilities for these events. Several variants of this problem are considered. In each case, a no-free-lunch result of the following form is established: the problem of learning is a formidably difficult one, in that no matter what method is pursued, failure is incomparably more common than success; and difficult choices must be faced in choosing a method of learning, since no approach dominates all others in its range of success. In the simplest case, the comparison of the set of situations in which a method fails and the set of situations in which it succeeds is a matter of cardinality (countable vs. uncountable); in other cases, it is a topological matter (meagre vs. co-meagre) or a hybrid computational-topological matter (effectively meagre vs. effectively co-meagre).

1. INTRODUCTION

The various *no-free-lunch theorems* of statistical, computational, and formal learning theory offer ways to make precise the basic insight that there can be no optimal general-purpose approach to learning. These theorems come in two main forms. Some show that there are contexts in which certain approaches to learning succeed in each salient situation, but that each such approach has the same expected performance across those possible situations.¹ Results of this kind are *measure-relative*: in order for expectations to be defined, a measure must be imposed on the space of situations that a learner might face—and the results in question only hold relative to some of the measures that one might impose [6]. Results of a second kind are *absolute* in the sense that they do not rely upon the choice of a measure on the space of envisaged situations. Here are descriptions of two paradigmatic results of this kind.

Maybe there exists some kind of universal learner, that is, a learner who has no prior knowledge about a certain task and is ready to be challenged by any task? ... The no-free-lunch theorem states that no such universal learner exists. To be more precise, the theorem states that for binary classification prediction tasks, for every learner there exists a distribution on which it fails. ... In other words, the theorem states that no learner can succeed on all learnable tasks—every learner has tasks on which it fails while other learners succeed.²

Key words and phrases. Induction, Learning, Extrapolation, Forecasting, No-Free-Lunch Theorems.

For helpful comments and discussion, thanks to: an anonymous referee, Josh Hunt, Thomas Icard, Mikayla Kelley, Tom Sterkenburg, and Bas van Fraassen, and Francesca Zaffora Blando.

¹For results of this kind, see [1–3]. For further discussion, see [4] and [5].

²Shalev–Shwartz and Ben–David [7, p. 36].

Let T be any learning machine. . . . [W]e will *defeat* the machine T . That is, we will have constructed a regularity, depending on T , which is beyond the power of T to extrapolate. However . . . it is always possible to build another machine which can extrapolate every regularity that T can extrapolate and also extrapolate the one that T *can't* extrapolate. Thus, there cannot exist a cleverest learning machine: for, for every learning machine T , there exists a machine T' which can learn everything that T can learn and more besides.³

We can think of such results as encapsulating two facts about the predicament of learners situated in certain contexts. (a) They face a daunting problem: no approach they might adopt succeeds across all envisaged situations. (b) Difficult choices must be made: different approaches succeed in different situations, with no approach dominating all others in its range of success.

Here we assemble some more or less elementary results, some already well-known, that combine to give no-free-lunch results of the second, absolute variety, applicable to agents attempting to learn patterns in binary data streams.⁴ Nature presents our agents with one-way infinite binary sequences one bit at a time and after each bit is revealed each agent is asked to make a prediction about the next bit. We will consider five models of learning, differing from one another as to what sort of predictions our agents are required to make or as to the criterion of success. And for each model, we will consider variants in which neither the agent nor Nature is required to follow a computable strategy, in which the agent is required to follow a computable strategy but Nature is not, and in which both the agent and Nature are required to follow computable strategies. For each variant of each model, we establish both elements required for a no-free-lunch result. (i) Difficult choices must be faced in selecting a method of learning: we will show that no approach dominates all of its rivals, either by showing that for each method there is another that succeeds in a disjoint set of situations (*evil twin results*) or by showing that for every method there is another that succeeds in a strictly larger family of situations (*better-but-no-best results*). (ii) We also show our learners face a formidably difficult problem: for each of the problems we consider, there is a sense in which for any method of addressing that problem, the situations in which it fails are incomparably more common than the situations in which it succeeds.⁵

Following some preliminaries in Section 2, we investigate in Section 3 the predicament of learners who must attempt to guess, before each bit is revealed, whether it will be a 0 or a 1.⁶ We will consider two criteria of success for such *next-value* learners: when facing a given data stream they should eventually predict each new bit correctly (NV-learning); or they should predict each new bit correctly, except for a family of errors that has vanishing asymptotic density (weak NV-learning).⁷ In Section 3.1 we will see that for NV-learning of arbitrary sequences, failure is incomparably more common than success in the sense that any method for predicting bits succeeds for a countable family of binary sequences and fails for an

³Putnam [8, pp. 6 f.]. As noted by Case and Smith [9, p. 208]: “This appears to be the earliest result indicating that there may be some difficulty with the mechanization of science.” See also [10, Theorem I.5].

⁴A problem that, according to Li and Vitányi [11, p. 6], “constitutes, perhaps, the central task of inductive reasoning and artificial intelligence.”

⁵So here we go beyond the paradigm results mentioned above, which show only that for each method, there exists a situation in which it fails. For other results of the sort developed here, see [12].

⁶For early investigations of such agents, see the papers of Putnam [8, 13] and Gold [10].

⁷The notion of NV-learning is due to Bärzdīņš [14] (see also Blum and Blum [15]). The notion of weak NV-learning is due to Podnieks and Kinber [16, pp. 80 f.].

uncountable family of binary sequences. In Section 3.2 we will see that for weak NV-learning of arbitrary sequences, any method succeeds for an uncountable set of sequences and fails for an uncountable set of sequences, but the successes are always incomparably less common than the failures in a topological sense, forming a meagre set. In Section 3.3, we restrict attention to computable methods for the next-value learning of computable sequences and find that for any method, the sets of success and failures are equivalent both from the point of view of cardinality and the point of view of topology—but that the successes are nonetheless incomparably less common than the failures in the hybrid topological-computational sense (due to Mehlhorn [17]) that they form an effectively meagre set.⁸ Along the way we will see that the notion of weak NV-learning, while strictly weaker than the notion of NV-learning, is neither weaker nor stronger than two other variants of NV-learning, NV'-learning (due to Bārzdīņš [18]) and NV''-learning (due to Podnieks [16]).

In Section 4 we turn to agents who face a data stream sampled from a probability measure chosen by Nature and who are required to issue forecast probabilities for the next bit's being a 0 or a 1 just before it is revealed.⁹ We consider three criteria of success for agents engaged in such *next chance* prediction: we can ask that for any event, the probabilities that our agents assign to that event converge almost certainly to the true probability as they see larger and larger data sets (strong NC-learning); we can ask that their forecast probabilities for the next bit become arbitrarily accurate, almost certainly, in the limit of large data sets (NC-learning); or we can ask that they meet the last-mentioned standard *modulo* a set of missteps of vanishing asymptotic density (weak NC-learning).¹⁰ For the problem of next-chance learning in the face of a data stream generated by an arbitrary measure, we see in Sections 4.1 and 4.2 that for any of our criteria of success, each method fails for an uncountable set of measures that Nature might have chosen and succeeds for an uncountable set of such measures—but that the former set is always incomparably smaller than the latter, being meagre. In 4.3, we restrict attention to computable strategies for next-chance learning in contexts in which the data stream is generated by a computable measure and find, for each of our three criteria of success, that the set of learnable measures is an effectively meagre subset of the family of computable measures. Section 5 provides a few concluding remarks.

2. PRELIMINARIES

2.1. THE MAIN CHARACTERS

We will be concerned below with a number of topological spaces.

- (i) The space of bits, $\mathcal{B} := \{0, 1\}$, equipped with the discrete topology (so every subset of \mathcal{B} is open).
- (ii) Finite products of \mathcal{B} with itself: for each $n \in \mathbb{N}$, the space of n -bit strings, \mathcal{B}^n , equipped with the discrete topology (we count 0 as a natural number and use \emptyset to denote either the empty string of zero bits that is the sole member of \mathcal{B}^0 or the empty set, depending on context). We will think of elements of \mathcal{B}^n as strings (concatenations of symbols) rather than as n -tuples. For $w \in \mathcal{B}^n$ and $m \leq n$ we write $w(m)$ for the m th bit of w

⁸In the case of NV-learning, this result is due to Fortnow *et al.* [12].

⁹This model of learning appears in Solomonoff [19].

¹⁰These criteria of success were introduced by Blackwell and Dubins [20], Kalai and Lehrer [21], and Lehrer and Smorodinsky [22]. The criteria of success employed in the literature on Solomonoff induction differ in focussing on average or expected performance in the long run—on the relation between those notions and the notion of NC-learning, see [23].

and write $w[m]$ for the m -bit initial segment of w . For $m, n \in \mathbb{N}$ with $m \leq n$, we have the natural projection map $\pi_{nm} : w \in \mathcal{B}^n \mapsto w[m] \in \mathcal{B}^m$.

- (iii) The space of binary strings, $\mathcal{B}^* := \bigcup_{n=0}^{\infty} \mathcal{B}^n$, also equipped with the discrete topology. If v and w are binary strings we write $v.w$ for the string that results from concatenating v and w (in that order) and write $v.w^2$ for the results of concatenating v with w and with w , etc. We write $|w|$ for the number of bits in binary string w .
- (iv) *Cantor space*, \mathcal{C} , the set of all infinite binary sequences equipped with the product topology (we take sequences to be indexed by positive natural numbers). We can characterize this topology as follows: if w is an n -bit string, then we use B_w to denote the set of sequences whose first n bits are given by w ; the set of all such B_w (as w ranges over \mathcal{B}^*) is a basis for the product topology and we call the B_w *basic open sets*. *Illustration*: the set of sequences that have 0 as their second bit is an open set because it is the union of the basic open sets B_{00} and B_{10} . For $\sigma \in \mathcal{C}$ we write $\sigma(m)$ for the m th bit of σ and write $\sigma[m]$ for the m -bit string formed by concatenating the first m bits of σ . For each $n \in \mathbb{N}$ we have the natural projection map $\pi_n : \sigma \in \mathcal{C} \mapsto \sigma[n] \in \mathcal{B}^n$. A sequence of points $\sigma_1, \sigma_2, \sigma_3, \dots$ in Cantor space converges to $\sigma \in \mathcal{C}$ if and only if for each k , there exists an N so that for $n \geq N$, $\sigma_n(k) = \sigma(k)$. We use \mathfrak{B} to denote the σ -algebra of Borel subsets of \mathcal{C} . We use \mathfrak{C} to denote the subspace of \mathcal{C} consisting of computable sequences (i.e., the σ such that the map $k \in \mathbb{N} \mapsto \sigma[k]$ is computable).
- (v) For each $k \in \mathbb{N}$, the space \mathcal{P}_k of Borel probability measures on \mathcal{B}^k . Since $|\mathcal{B}^k| = 2^k$, we can identify any $\mu \in \mathcal{P}_k$ with a 2^k -tuple of real numbers in the closed unit interval that sum to one. We take \mathcal{P}_k to be equipped with the topology that it inherits from being embedded in this way as a closed subset of \mathbb{R}^{2^k} (which we take to be equipped with its standard topology). We call $\mu \in \mathcal{P}^m$ and $\nu \in \mathcal{P}^n$ with $m \leq n$ *consistent* if for each subset A of \mathcal{B}^m we have $\mu(A) = \nu(\pi_{nm}^{-1}(A))$.
- (vi) The space \mathcal{P} of Borel probability measures on \mathcal{C} equipped with the weak topology, which can be characterized as follows.¹¹ For each binary string w and each pair of numbers p and q in the closed unit interval with $p < q$, let

$$S_{w,p,q} := \{\mu \in \mathcal{P} : p < \mu(B_w) < q\}.$$

The set of all such $S_{w,p,q}$ forms a sub-basis for the weak topology on \mathcal{C} : the open sets of the weak topology are arbitrary unions of finite intersections of these sub-basic sets. Under the weak topology, a sequence $\{\mu_k\}$ of measures in \mathcal{P} converges to $\mu \in \mathcal{P}$ if and only if $\lim_{k \rightarrow \infty} \mu_k(B_w) = \mu(B_w)$ for each $w \in \mathcal{B}^*$.¹²

Below, in order to simplify notation, for μ a measure in \mathcal{P} and w a binary string, we will write $\mu(w)$ in place of $\mu(B_w)$. Using this notation, the Carathéodory Extension Theorem tells us that any map $\bar{\nu} : \mathcal{B}^* \rightarrow [0, 1]$ such that $\bar{\nu}(\emptyset) = 1$ and such that

¹¹For the weak topology on spaces of measures on metric spaces, see [24, Chapter II] and [25, §1.2]. For the special case of \mathcal{P} , see [26, Chapter 17] and [27, §2.5]. Since we can specify a Borel probability measure on \mathcal{C} by specifying the weight that it assigns to each binary string, by fixing an enumeration of the binary strings we can identify each $\mu \in \mathcal{P}$ with a sequence of numbers in the closed unit interval that sum to one. In this way we identify \mathcal{P} with a closed subset of the Hilbert cube ($= [0, 1]^\omega$ equipped with the product topology). The weak topology is the topology that \mathcal{P} inherits from this embedding.

¹²Each B_w is a clopen subset of \mathcal{C} and so is a continuity set for any measure in \mathcal{P} . So the Portmanteau Theorem implies that the above condition is necessary for weak convergence. And it is also sufficient, since the B_w form a countable basis for \mathcal{C} closed under finite intersections. See, e.g., [25, Theorems 2.1 and 2.2].

$\bar{\nu}(w) = \bar{\nu}(w.0) + \bar{\nu}(w.1)$ for each $w \in \mathcal{B}^*$ induces a unique $\nu \in \mathcal{P}$ such that $\nu(w) = \bar{\nu}(w)$ for all $w \in \mathcal{B}^*$ (see, e.g., [28, §1.9]).

As usual, we consider $\nu \in \mathcal{P}$ to be computable if and only if there exists a computable $F : \mathcal{B}^* \times \mathbb{N} \rightarrow \mathbb{Q}$ such that $|\nu(w) - F(w, n)| < 2^{-n}$ for all $w \in \mathcal{B}^*$ and $n \in \mathbb{N}$. We use \mathfrak{P} to denote subspace of \mathcal{P} consisting of computable measures.

Remark 2.1. \mathcal{B}^* , \mathcal{C} , \mathcal{P} , each of the \mathcal{B}^k , and each of the \mathcal{P}_k are compact, separable and completely metrizable. In \mathcal{B} , the other \mathcal{B}^n , and in \mathcal{B}^* each point is isolated (i.e., for any point, the singleton set containing that point is open). There are no isolated points in \mathcal{C} , the \mathcal{P}_k ($k > 0$), or \mathcal{P} .

Remark 2.2. \mathcal{C} and the \mathcal{P}_k ($k > 0$) of course have cardinality \mathfrak{c} (the cardinality of the continuum). So does \mathcal{P} : since \mathcal{P} is non-empty, compact, and metrizable there is a continuous map from \mathcal{C} onto \mathcal{P} ; since \mathcal{P} is a non-empty, separable, and completely metrizable space without isolated points, there is an embedding of \mathcal{C} into \mathcal{P} [26, Theorems 4.18 and 6.2].

2.2. THE MEAGRE & THE CO-MEAGRE

We are going to be interested in making comparisons of size for certain subsets of \mathcal{C} and \mathcal{P} . The most straightforward standard of comparison is cardinality: it natural to say that any uncountable set is incomparably larger than any countable set.

Below we will see examples where the set of learnable sequences or measures and the set of unlearnable sequences or measures have the same cardinality—but in which it is intuitively natural to say that the unlearnable sequences or measures are incomparably more common than the learnable sequences or measures.

The intuitive notions of size in play here correspond nicely with the topologists' notions of meagre and co-meagre subsets of a topological space. Recall that a *nowhere dense* subset of a topological space is one whose closure has empty interior—or, equivalently, a subset A of a topological space X is nowhere dense if and only if for any non-empty open set $U \subset X$, there exists a non-empty open set $U^* \subset U$ with $A \cap U^* = \emptyset$. And recall that a *meagre* subset of a topological space is one that can be written as a countable union of nowhere dense sets while a *co-meagre* subset is one that is the complement of a meagre set.

For any topological space X , the class of meagre subsets of X is closed under the operations of taking subsets and taking countable unions. The Baire Category Theorem tells us that in a completely metrizable space, no non-empty open set is meagre. So, in particular, no non-empty completely metrizable space has any subsets that are both meagre and co-meagre.¹³

The results just mentioned motivate the standard practice in topology, analysis, and related mathematical fields of considering the elements of a meagre subset of a completely metrizable space to be extremely rare and the elements of the complement of such a set to be exceedingly common—so that objects that form a co-meagre set are often referred to as being typical. *Illustration:* one says that typical continuous functions on the unit interval are nowhere differentiable because the nowhere differentiable functions form a co-meagre subset of the space of continuous functions under the uniform topology.

Remark 2.3 (The Banach–Mazur Game.). Here is an additional compelling rationale for this practice. Fix a subset S of \mathcal{C} . An infinite two-player game is to be played. In the first round, Player I selects a non-empty binary string v_1 , then Player II selects a non-empty binary string

¹³If $A \subset X$ were both meagre and co-meagre, then so would be its complement. But then X could be written as a union of two meagre sets—which is impossible if no non-empty open subset of X is meagre.

w_1 ; and similarly in each subsequent round, Player I selects a non-empty binary string v_k , then Player II selects a non-empty binary string w_k . Player I wins the game if the infinite binary sequence $v_1.w_1.v_2.w_2.\dots$ is in S , otherwise Player II wins. Intuitively, if Player I has a winning strategy for the Banach–Mazur game for S , then S must be overwhelmingly large as a subset of X , while if Player II has a winning strategy, then S must be nigh ignorably small as a subset of X .

The intuitive notions of small and large subsets appealed to here correspond precisely to the notions of meagre and co-meagre subsets: Player I has a winning strategy if and only if S is co-meagre in some open subset of \mathcal{C} ; Player II has a winning strategy if and only if S is meagre as a subset of \mathcal{C} .¹⁴

3. EXTRAPOLATION

Think of Nature as having chosen a binary sequence, which is now being revealed to a learning agent one bit at a time. After each new bit is presented, the agent attempts to predict what the next value will be on the basis of the data seen so far. The agent succeeds in this task if from a certain point onwards, the predictions made match reality (almost perfectly).

Definition 3.1 (Extrapolators). An *extrapolator* is a function $m : \mathcal{B}^* \rightarrow \mathcal{B}$. We denote the set of extrapolators by \mathcal{E} .

Definition 3.2 (Extrapolating Machines). An *extrapolating machine* is a computable extrapolator—i.e., a computable function $m : w \in \mathcal{B}^* \mapsto m(w) \in \mathcal{B}$. We denote the set of extrapolating machines by \mathfrak{E} .

Definition 3.3 (NV-Learning). Let m be a extrapolator and σ a binary sequence. We say that m *NV-learns* σ (or that σ is *NV-learnable* by m) if there is an N such that for all $n > N$, $m(\sigma[n]) = \sigma(n + 1)$.¹⁵

Definition 3.4 (Weak NV-Learning). We say that $m \in \mathcal{E}$ *weakly NV-learns* $\sigma \in \mathcal{C}$ (or that σ is *weakly NV-learnable* by m) if:

$$\lim_{n \rightarrow \infty} \frac{|\{k \leq n : m(\sigma[k]) = \sigma(k + 1)\}|}{n} = 1.$$

(i.e., incorrect guesses by m have vanishing limiting relative frequency).

Remark 3.1. Weak NV-learning is a special case (corresponding to $r = 1$) of the notion of $NV(r)$ -learning introduced by Podnieks (in collaboration with Kinber) [16, pp. 80 f.] in the computable setting: for $r \in (0, 1]$, we say that $m \in \mathfrak{E}$ *$NV(r)$ -learns* σ if the correct predictions made by m in processing σ have relative frequency at least r .¹⁶ For expository simplicity, we focus on the special case. But the proofs of the propositions below concerning

¹⁴Here we have described a special version of the game adapted to \mathcal{C} . A more general version makes sense in any topological space X and we always have the connection between meagreness and winning strategies for Player II; the connection between co-meagreness and winning strategies for Player I requires some additional hypotheses in the general setting. See [29] and [26, §§8H and 21.C].

¹⁵The notion of NV-learning for extrapolating machines is due to Bärzdiņš [14]; see also Blum and Blum [15]. NV-learning is the subject of an extensive literature—see the canonical surveys on inductive learning, [9, 30–33].

¹⁶The notion of $NV(r)$ -learning appears to have been largely neglected in the subsequent literature (but see [34, 35]). In particular, it is absent from the surveys cited in fn. 15.

weak NV-learning can all be adapted to cover $NV(r)$ learning for any $r \in (0, 1]$ (except for Proposition 3.8, which requires the restriction $r > 1/2$).

For m an extrapolator and σ a binary sequence, we say that according to m , $\sigma(n)$ a *good bit* of σ if $m(\sigma[n-1]) = \sigma(n)$ and corresponds to a *nasty bit* of σ if $m(\sigma[n-1]) \neq \sigma(n)$. To say that m NV-learns σ is to say that according to m , σ eventually consists of nothing but good bits. To say that m weakly NV-learns σ is to say that according to m , although σ may contain infinitely many nasty bits, these have vanishing limiting relative frequency. We will consider each of these two criteria of learning in turn.

Remark 3.2. In the context of computable learners facing computable data streams, one can set the task of identification (rather than extrapolation): require learners to output a natural number after each bit is revealed, aiming to guess the code number of a program that outputs the data stream they are seeing. We won't discuss this approach extensively here, but will occasionally presume familiarity with the literature on identification problems as surveyed in [9, 30–33, 36].

3.1. NV-LEARNING

Officially, the job of an extrapolator is to predict the next bit on the basis of the current data set. But we can also think of an extrapolator m as a means of guessing the entire data sequence on the basis of any initial segment.¹⁷

Definition 3.5. For $m \in \mathcal{E}$ and $w \in \mathcal{B}^n$, we use σ_m^w to denote the sequence defined as follows:

- For $k = 1, \dots, n$, $\sigma_m^w(k) = w(k)$ (i.e., $\sigma_m^w[n] = w$).
- $\sigma_m^w(n+1) = m(w)$;
- $\sigma_m^w(n+\ell) = m(w.\sigma_m^w(n+1).\dots.\sigma_m^w(n+\ell-1))$ ($\ell = 2, 3, \dots$).

We say that m *guesses* σ_m^w on input w .

Note that if $m \in \mathfrak{C}$ then for any $w \in \mathcal{B}^*$, $\sigma_m^w \in \mathfrak{C}$: on any input, an extrapolating machine guesses a computable sequence.

Trivially, there is an equivalence between the sequences NV-learned by an extrapolator and the sequences guessed by it.¹⁸

Proposition 3.1. Extrapolator m NV-learns sequence σ if and only if $\sigma = \sigma_m^w$ for some $w \in \mathcal{B}^*$.

Proof. Suppose that m NV-learns σ . Then there is an n_0 such that for all $n \geq n_0$, $m(\sigma[n]) = \sigma(n+1)$. So m guesses σ on input $w = \sigma[n_0]$. Suppose, on the other hand, there is an n_0 such that m guesses σ on input $w = \sigma[n_0]$. Then m NV-learns σ , since for all $n > n_0$, $m(\sigma[n]) = \sigma(n+1)$. \square

So asking that m eventually correctly predict next bits is equivalent to asking that m eventually be able to answer correctly *all* questions about the data stream.

Proposition 3.2. For any extrapolator m , the sequences NV-learnable by m form a countably infinite set dense in \mathcal{C} while the sequences not NV-learnable by m form a dense subset of \mathcal{C} of cardinality \mathfrak{c} .

¹⁷On this point, see [31, §4.2.1].

¹⁸This is related to the deeper fact that $\mathcal{NV} = \mathcal{PEX}$ [9, Theorem 2.19].

Proof. On the one hand, \mathcal{B}^* is a countable set and the preceding proposition tells us that the map $w \in \mathcal{B}^* \mapsto \sigma_m^w$ has as its range the set of sequences NV-learnable by m . So this set is countable. And since for any $w \in \mathcal{B}^*$, $\sigma_m^w \in B_w$ the set of NV-learnable sequences is dense in \mathcal{C} (and is therefore infinite). On the other hand, each B_w has cardinality \mathfrak{c} but contains only countably many binary sequences NV-learnable by m . \square

Corollary 3.1. The set $\{\sigma \in \mathcal{C} \mid \exists m \in \mathfrak{E} \text{ such that } m \text{ NV-learns } \sigma\}$ is countable.

As usual, we call a sequence $\{\sigma_i\}_{i \in \mathbb{N}}$ of elements of \mathfrak{C} *uniformly computable in i* if there is a computable $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{B}$ such that $f(i, j) = \sigma_i(j)$, for all $i, j \in \mathbb{N}$.

Proposition 3.3. (a) Let $m \in \mathfrak{E}$ and S be a countable subset of \mathcal{C} . Then there is an $m^* \in \mathfrak{E}$ that NV-learns every $\sigma \in S$ as well as everything NV-learned by m . (b) Let $m \in \mathfrak{E}$ and let $S = \{\sigma_i\}_{i \in \mathbb{N}}$ be a family of elements of \mathfrak{C} uniformly computable in i . Then there is an $m^* \in \mathfrak{E}$ that NV-learns every $\sigma \in S$ as well as everything NV-learned by m .

Proof. We present the argument for (b)—essentially the same argument works for (a). Define $\tilde{m} \in \mathfrak{E}$ as follows: on input of $w \in \mathcal{B}^n$, \tilde{m} finds $K = \{k \in \mathbb{N} \mid 1 \leq k \leq n, \sigma_k[n] = w\}$; if $K \neq \emptyset$, then $\tilde{m}(w) = \sigma_\ell(n+1)$, where ℓ is the least element of K ; otherwise, $\tilde{m}(w) = m(w)$. Define m^* as follows: m^* has a counter that keeps tally of how many incorrect prediction have been made in the course of processing a given data stream; in processing input $w \in \mathcal{B}^*$, m^* simulates m if an even number of incorrect predictions have been made and simulates \tilde{m} if an odd number have been made.

Clearly, m^* is an extrapolating machine. Suppose that m^* is shown a data stream σ that it does not NV-learn. Then m^* must make infinitely many incorrect predictions in processing σ . So σ cannot be a sequence NV-learned by m : any such sequence is guessed by m when it sees sufficiently long initial segments. Similarly, σ cannot be any of the σ_k , since each of these is guessed by \tilde{m} when it sees sufficiently long initial segments. \square

Proposition 3.4. Let m be an extrapolator and let $S \subset \mathcal{C}$ be the set of sequences that it NV-learns. Then there is an extrapolator m^\dagger such that the set S^\dagger of sequences that it NV-learns is disjoint from S —and where m^\dagger is in \mathfrak{E} if m is.

Proof. Define m^\dagger by setting $m^\dagger(w) = 1 - m(w)$ for each $w \in \mathcal{B}^*$. \square

So we have both elements required for the sort of no-free-lunch result we seek. The problem of NV-learning is a formidably difficult one: each (computable) extrapolator fails to NV-learn incomparably more sequences than it NV-learns: the set on which it succeeds is countable (and hence meagre), so the set on which it fails is uncountable (indeed, co-meagre). And there are hard choices to be made: for any (computable) extrapolator, there is another that NV-learns sequences that the first cannot NV-learn. There is no optimal method of extrapolation.

3.2. WEAK NV-LEARNING

If an extrapolator NV-learns a sequence, then it also weakly NV-learns it. But the converse is not true.

Example 3.1. Consider the extrapolating machine m_1 that outputs 1 on any input. This machine NV-learns all and only sequences that are eventually all 1's—a countably infinite set. But m_1 weakly NV-learns continuum-many sequences. For, let $\hat{\sigma}$ be an arbitrary binary sequence and let σ be the sequence defined as follows: for $n = 1, 2, \dots$, if $k = 10^n$, then

$\sigma(k) = \hat{\sigma}(n)$; otherwise, $\sigma(k) = 1$. According to m_1 , the nasty bits of σ have vanishing asymptotic density, so m_1 weakly NV-learns σ . And there are continuum-many $\hat{\sigma}$ we could use as input for this construction, each determining a distinct sequence weakly NV-learned by m . Note that there is no input on which m_1 guesses a sequence that contains infinitely many 0's, although it weakly NV-learns uncountably many sequences with this feature. Note also that although m_1 is computable, it weakly NV-learns uncountably many uncomputable sequences and weakly NV-learns sequences of arbitrary Turing degree.

Proposition 3.5. Each extrapolator weakly NV-learns a dense set of sequences of cardinality \mathfrak{c} and fails to weakly NV-learn a dense set of sequences of cardinality \mathfrak{c} .

Proof. Let m be an extrapolator, w an n -bit binary string, and $\hat{\sigma}$ an arbitrary sequence. We construct sequences σ^* and σ^\dagger as follows:

- For $k = 1, \dots, n$, $\sigma^*(k) = \sigma^\dagger(k) = w(k)$.
- For $k = n + 10^\ell$ ($\ell = 1, 2, \dots$), $\sigma^*(k) = \sigma^\dagger(k) = \hat{\sigma}(\ell)$.
- For all other k , $\sigma^*(k) = m(\sigma^*(1).\sigma^*(2).\dots.\sigma^*(k-1))$ and $\sigma^\dagger(k) = 1 - \sigma^*(k)$.

According to m , any nasty (good) bits in σ^* (σ^\dagger) occur with indices of the form $n + 10^\ell$. So m weakly NV-learns σ^* and fails to weakly NV-learn σ^\dagger . By varying w , we obtain weakly NV-learnable and not weakly NV-learnable sequences in each basic open set of \mathcal{C} . And by varying $\hat{\sigma}$ we obtain continuum-many sequences of each type. \square

So for any extrapolator, there are continuum-many sequences that it can weakly NV-learn and continuum-many sequences that it cannot weakly NV-learn. But, intuitively, there is a sense in which it is much more difficult to construct a sequence weakly NV-learnable by a given extrapolator than it is to construct a sequence that is not weakly NV-learnable by that extrapolator. Consider again the extrapolator m_1 that outputs 1 on any input. In order to construct a sequence that this extrapolator weakly NV-learns, you begin with the all 1's sequence, then sprinkle in some 0's, subject to the constraint that the set of indices of the slots containing 0's has vanishing asymptotic density in \mathbb{N} . In order to construct a sequence that this extrapolator *can't* weakly NV-learn, you begin with the all 1's sequence and sprinkle in as many 0's as you like, just being careful to make sure that the set of indices of the slots containing 0's doesn't have vanishing asymptotic density. The latter task, is intuitively, easier: e.g., because there are a lot more densities not equal to zero than equal to zero. This intuition is borne out by the following result.

Proposition 3.6. Let m be any extrapolator. The sequences weakly NV-learnable by m form a meagre subset of \mathcal{C} .

Proof. Let us say that binary string w is *wicked* according to m if at least half of the bits of w are nasty according to m . For each $n \in \mathbb{N}$, let A_n be the set of sequences that do not have at least n initial segments that are wicked according to m .

We claim that each A_n is nowhere dense. To establish this, it suffices to show that for any binary string w , there is another, w^* , depending on n and w , such that w^* extends w and $B_{w^*} \cap A_n = \emptyset$. To this end, let w be a string and let w^* be the result of extending w by $|w|$ bits that are nasty according to m , then tacking on n more nasty bits. Every sequence in B_{w^*} then has at least n initial segments that are wicked according to m .

So $A := \bigcup_{n=1}^{\infty} A_n$ is a meagre subset of \mathcal{C} . And any sequence σ weakly NV-learnable by m must be in A —for otherwise, σ would have the feature that for each k , it contained at least k initial segments wicked according to m , which would mean that the asymptotic density of

nasty bits in σ could not vanish. So the set of sequences weakly NV-learnable by m , being a subset of a meagre set, is meagre. \square

Corollary 3.2. The set $\{\sigma \in \mathcal{C} \mid \exists m \in \mathfrak{E} \text{ such that } m \text{ weakly NV-learns } \sigma\}$ is meagre in \mathcal{C} .¹⁹

So the problem of weakly NV-learning sequences is formidably difficult. And difficult choices must be made in the face of this intractability—there can be no optimal extrapolator for weak NV-learning.

We have the following better-but-no-best result.²⁰

Proposition 3.7. (a) Let $m \in \mathfrak{E}$ and let S be a countable subset of \mathcal{C} . Then there is an $m^* \in \mathfrak{E}$ that NV-learns every $\sigma \in S$ and also weakly NV-learns everything that m does. (b) Let $m \in \mathfrak{E}$ and let $S = \{\sigma_i\}_{i \in \mathbb{N}}$ be a family of elements of \mathfrak{E} that is uniformly computable in i . Then there is an $m^* \in \mathfrak{E}$ that NV-learns every $\sigma \in S$ and also weakly NV-learns everything that m does.

Proof. We present the argument for (b)—essentially the same argument works for (a). Define m^* as follows: on input of $w \in \mathcal{B}^n$, m^* finds $K = \{k \in \mathbb{N} \mid 1 \leq k \leq \log_2 n, \sigma_k[n] = w\}$; if $K \neq \emptyset$, then $m^*(w) = \sigma_\ell(n+1)$, where ℓ is the least element of K ; otherwise, $m^*(w) = m(w)$. Clearly, m^* is an extrapolating machine and NV-learns each σ_k (m^* guesses σ_k whenever shown sufficiently long initial segments). And if $\sigma \in \mathcal{C}$ is weakly NV-learned by m then it is also weakly NV-learned by m^* : in processing the first 2^n bits of σ , m^* can disagree with m at most n times; so the asymptotic density of bits on which m^* and m disagree in processing σ is zero. \square

We also have the usual sort of evil-twin result.

Proposition 3.8. Let m be an extrapolator and let $S \subset \mathcal{C}$ be the set of sequences that it weakly NV-learns. Then there is an extrapolator m^\dagger such that the set S^\dagger of sequences that it weakly NV-learns is disjoint from S —and where m^\dagger is in \mathfrak{E} of m is.

Proof. Define m^\dagger by setting $m^\dagger(w) = 1 - m(w)$ for each $w \in \mathcal{B}^*$. According to either m or m^\dagger , in any sequence that the other weakly NV-learns, the good bits have asymptotic density zero. \square

3.3. EXTRAPOLATION OF COMPUTABLE SEQUENCES

While it is plausible that every method of learning implementable by a natural or artificial learning agent is computable, the data streams that our agents face may or may not be computable.²¹ Still, there are many settings in which we can be confident that our agents face computable data streams. So let us specialize to the setting in which computable extrapolators attempt to (weakly) NV-learn computable sequences and see how the landscape surveyed above is transformed.

As usual, for $m \in \mathfrak{E}$, we denote by $\mathcal{NV}(m)$ the set of computable sequences that are NV-learned by m . We use \mathcal{NV} to denote:

$$\{S \subset \mathfrak{E} \mid \exists m \in \mathfrak{E} \text{ with } S \subseteq \mathcal{NV}(m)\}.$$

¹⁹Remark 3.7 below will show that this strengthens the observation of Jockusch and Schupp [37, p. 438] that the set of coarsely computable sequences is meagre in \mathcal{C} .

²⁰Thanks here to Tom Sterkenburg and to an anonymous referee for helpful suggestions.

²¹Unless, that is, physical reality itself is fundamentally computational in nature—for a range of views of this topic, see the papers collected in [38].

We likewise use $\mathcal{NV}^w(m)$ to denote the set of computable sequences weakly NV-learned by an extrapolating machine m and use \mathcal{NV}^w to denote:

$$\{S \subset \mathfrak{C} \mid \exists m \in \mathfrak{C} \text{ with } S \subseteq \mathcal{NV}^w(m)\}.$$

Proposition 3.9 (Podnieks [16]). \mathcal{NV} is a proper subset of \mathcal{NV}^w .

Proof. Clearly $\mathcal{NV} \subseteq \mathcal{NV}^w$. We give an example of a set in $\mathcal{NV}^w - \mathcal{NV}$.

Consider again the extrapolating machine m_1 of Example 3.1 above that outputs 1 on every input. Let $U = \mathcal{NV}^w(m_1)$, the set of computable binary sequences in which 0's have vanishing asymptotic density. We are going to show that U is not in \mathcal{NV} .

Suppose that there is an extrapolating machine m that NV-learns each sequence in U . Notice that for any $w \in \mathcal{B}^*$, the sequence $w.1^\omega$ is in U —so for sufficiently large $\ell \in \mathbb{N}$, we must have $m(w.1^\ell) = 1$. Let $\sigma \in \mathfrak{C}$ be the sequence of the form $1^{n_1}.0.1^{n_2}.0.1^{n_3}.0\dots$ where each n_j is chosen to be the smallest n larger than 2^j such that $m(1^{n_1}.0.1^{n_2}.0\dots.1^{n_{j-1}}.0.1^n) = 1$. Clearly, m does not NV-learn σ (σ contains infinitely many 0's, each of which m predicts will be a 1). But $\sigma \in U$: since m is computable, so is σ ; and by construction, 0's occur with vanishing asymptotic density in σ . This contradicts our assumption that m NV-learns every $\sigma \in U$. \square

Remark 3.3. We mention two of the most fundamental variations on \mathcal{NV} . A *partial extrapolating machine* is a partial computable function $m : \mathcal{B}^* \rightarrow \mathcal{B}$. Following Bārzdīņš [18], we say that a partial extrapolating machine m *NV'-extrapolates* $\sigma \in \mathfrak{C}$ if: (i) $m(\sigma[k])$ is defined for all $k \in \mathbb{N}$; and (ii) $\exists N \in \mathbb{N}$ such that for all $n > N$, $m(\sigma[n]) = \sigma(n+1)$. We write \mathcal{NV}' for the set of $S \subset \mathfrak{C}$ such that there is a partial extrapolating machine that NV'-extrapolates each $\sigma \in S$.

Following Podnieks [16], we say that a partial extrapolating machine m *NV''-extrapolates* $\sigma \in \mathfrak{C}$ if: $\exists N \in \mathbb{N}$ such that for all $n > N$, $m(\sigma[n])$ is defined and equal to $\sigma(n+1)$. We write \mathcal{NV}'' for the set of $S \subset \mathfrak{C}$ such that there is a partial extrapolating machine that NV''-extrapolates each $\sigma \in S$.

Obviously, $\mathcal{NV} \subseteq \mathcal{NV}' \subseteq \mathcal{NV}''$. In fact, $\mathcal{NV} \subset \mathcal{NV}' \subset \mathcal{NV}''$.²² The proof of Proposition 3.9 above carries over essentially unchanged (except that dove-tailing is required) to show that \mathcal{NV}^w is not contained in \mathcal{NV}'' (let alone in \mathcal{NV}'). We will see below in Remark 3.6 that \mathcal{NV}^w does not contain \mathcal{NV}' (let alone \mathcal{NV}'').

Remark 3.4. Having introduced, for $r \in (0, 1]$, the notion of NV(r) learning (see Remark 3.1 above), Podnieks (in collaboration with Kinber) [16, 34] introduces the classes $\mathcal{NV}(r)$, $\mathcal{NV}'(r)$, and $\mathcal{NV}''(r)$ in the obvious way. Podnieks goes on [16, 34] to establish a number of facts about the containment relations involving these classes. For present purposes, the most notable are: if $r_1 < r_2$, then $\mathcal{NV}(r_2)$ is a proper subset of $\mathcal{NV}(r_1)$; and $\mathfrak{C} \in \mathcal{NV}''(1)$ (from which it follows that \mathcal{NV}'' is properly contained in $\mathcal{NV}''(1)$).

It is also possible to introduce up-to-exceptions-of-density- r variants of standard identification classes. Podnieks [16] generalizes \mathcal{BC} by considering learners who aim to output at each time (modulo permitted exceptions) a code number for a program that generates the data stream they see. Pitt [39] generalizes \mathcal{EX} by considering learners who aim to converge (modulo

²²That $\mathcal{NV} \subset \mathcal{NV}'$ is due to Bārzdīņš [18]; that $\mathcal{NV}' \subset \mathcal{NV}''$ is due to Podnieks [16]. See [9, Corollary 2.29, Corollary 2.31, Theorem 3.1, and Theorem 3.5].

permitted exceptions) to a single such code number.²³ Royer [36, 40] generalizes \mathcal{EX} in a different direction, considering learners who aim to converge to a single code number that (modulo permitted exceptions) outputs the data stream they are seeing. Jain [41] investigates the problem of identification of a function from data streams that are accurate only modulo errors of given asymptotic density.

Via Propositions 3.1 and 3.2, we know that every extrapolating machine NV-learns a countably infinite subset of \mathfrak{C} . But there there can be no best extrapolating machine: Propositions 3.4 and 3.8 tell us each extrapolation machine has an evil twin that (weakly) NV-learns a disjoint set of computable sequences; and Propositions 3.3(b) and 3.7(b) tell us that each extrapolating machine is dominated by another that (weakly) NV-learns everything it can while also NV-learning every member of a uniformly computable family of elements of \mathfrak{C} .

In this setting, what comparative judgements can we make about the sets of computable sequences that a given extrapolating machine (weakly) NV-learns and doesn't (weakly) NV-learn?

Proposition 3.10. For any $m \in \mathfrak{C}$, following are dense subsets of \mathcal{C} :

- (a) the set of computable sequences NV-learnable by m ;
- (b) the set of computable sequences not NV-learnable by m ;
- (c) the set of computable sequences weakly NV-learnable by m ;
- (d) the set of computable sequences not weakly NV-learnable by m .

Straightforward adaptations of the proofs of Propositions 3.2 and 3.5 yield that (a) and (d) are dense. And (b) and (c) are super-sets of (d) and of (a), respectively. \square

It follows that the set of computable sequences (weakly) NV-learned by an extrapolating machine m and the set of computable sequences not (weakly) NV-learned by an extrapolating machine m are both countably infinite subsets of \mathcal{C} —so we have parity at the level of cardinality. A classical result implies that we also have parity at the level of topology.

Proposition 3.11 (Sierpiński). Any two countable dense subsets of \mathcal{C} are homeomorphic.

Proof. See, e.g., [42, Chapter 17]. \square

So we can say: for any computable method of extrapolating computable sequences, failure and success are equally common—and difficult choices must be made in selecting a computable method of extrapolation, since no method dominates all its rivals in its range of success.

But, intuitively, we ought to be able to say something stronger. After all, Bärzdiņš [18] showed that if S is a set of computable functions, then following are equivalent: (i) each member of S is NV-learnable; (ii) S is a subclass of a computably enumerable set of computable functions; (iii) S is a subclass of an abstract complexity class.²⁴ So only very special subsets of \mathfrak{C} are NV-learnable—which means that generic subsets should not be in \mathcal{NV} .

²³Both these notions turn out to be closely related by identification by probabilistic learners to identification by teams of learners [33, 39]. We do not have the same close association between the corresponding notions in the extrapolation case

²⁴See also Blum and Blum [15, p. 127], who attribute the complexity-theoretic condition independently to Adleman. As Blum and Blum remark, this result shows “in essence, that the extrapolable sequences are the ones that can be computed rapidly.”

Indeed, there is a natural hybrid computational-topological notion of that underwrites the conclusion that failure is incomparably more common than success for computable extrapolation of computable sequences. Mehlhorn [17] introduced the important notion of an effectively meagre subset of the set of computable functions. We specialize this apparatus to \mathfrak{C} .

By way of motivation, note that in any topological space X with basis of open sets \mathcal{W} , a subset A is nowhere dense if and only if for every non-empty $U \in \mathcal{W}$ there is a non-empty $U^* \in \mathcal{W}$ with $U^* \subset U$ such that $A \cap U^* = \emptyset$. So a subset $A \subset \mathfrak{C}$ is nowhere dense if and only if there is a function $f : \mathcal{B}^* \rightarrow \mathcal{B}^*$ such that for each binary string w : (i) $f(w)$ extends w ; and (ii) $A \cap B_{f(w)} = \emptyset$. And $A \subset \mathfrak{C}$ is meagre if and only if there is a function $F : \mathbb{N} \times \mathcal{B}^* \rightarrow \mathcal{B}^*$ such that: (i) for each $n \in \mathbb{N}$ there is an $A_n \subset \mathfrak{C}$ such that $f_n = F(n, \cdot)$ is a witness to the fact that A_n is nowhere dense in \mathfrak{C} ; and (ii) $A = \bigcup_{n \in \mathbb{N}} A_n$.

Definition 3.6 (Mehlhorn [17]). Let A be a subset of \mathfrak{C} and let $f : \mathcal{B}^* \rightarrow \mathcal{B}^*$ be a computable function. Then A is *effectively nowhere dense via f* if for each $w \in \mathcal{B}^*$:

- i) $f(w)$ extends w ;
- ii) $A \cap B_{f(w)} = \emptyset$.

Definition 3.7 (Mehlhorn [17]). A subset A of \mathfrak{C} is *effectively meagre* if there is a computable function $F : \mathbb{N} \times \mathcal{B}^* \rightarrow \mathcal{B}^*$ such that:

- i) for each $n \in \mathbb{N}$, there is an $A_n \subset \mathfrak{C}$ such that A_n is effectively nowhere dense via $f_n = F(n, \cdot)$;
- ii) $A \subset \bigcup_{n \in \mathbb{N}} A_n$.

The complement in \mathfrak{C} of an effectively meagre subset of \mathfrak{C} is called *effectively co-meagre*.

Proposition 3.12 (Mehlhorn [17]). The family of effectively meagre subsets of \mathfrak{C} is closed under the following operations:

- a) taking subsets;
- b) taking finite unions;
- c) taking effective unions.

Proof. The first claim is immediate from the definition and the second follows from the third. So suppose that that M is a subset of \mathfrak{C} such that there exist a computable $H : \mathbb{N} \times \mathbb{N} \times \mathcal{B}^*$ and a decomposition $M = \bigcup N_i$, such that for each $k \in \mathbb{N}$, $H(k, \cdot, \cdot)$ is a witness to the fact that N_k is effectively meagre. There exists, then, for each $i \in \mathbb{N}$, a decomposition $N_i = \bigcup N_{ij}$ such that each N_{ij} is effectively nowhere dense in virtue of $H(i, j, \cdot)$. Fix a computable bijection $\pi : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ and let p_1 and p_2 be the computable components of the inverse of π (so that $\pi(p_1(k), p_2(k)) = k$ for all $k \in \mathbb{N}$). Set $M_k := N_{p_1(k), p_2(k)}$ and for each $w \in \mathcal{B}^*$, set $F(k, t) := H(p_1(k), p_2(k), w)$. Then $F : \mathbb{N} \times \mathcal{B}^*$ is computable, $M = \bigcup M_k$, and each M_k is effectively nowhere dense in virtue of $F(k, \cdot)$. So M is effectively meagre. \square

Crucially, the set of effectively meagre subsets of \mathfrak{C} is not closed under arbitrary countable unions due to an effective analog of the Baire Category Theorem.

Proposition 3.13 (Mehlhorn [17]). Let w be a binary string. Then $B_w \cap \mathfrak{C}$ is not effectively meagre.

Proof. Let $M = \bigcup M_k$ be an effectively meagre set with witness $F : \mathbb{N} \times \mathcal{B}^*$. We construct strings w_0, w_1, \dots inductively: $w_0 := w$; and $w_{k+1} = F(k, w_k).0$. By construction, each w_k

is a proper initial segment of w_{k+1} . Let $\sigma = \lim_{n \rightarrow \infty} w_n$. Then $\sigma \in B_w \cap \mathfrak{C}$. But for each k , $\sigma \notin M_k$ (since σ begins with w_{k+1}), so $\sigma \notin M$. \square

In light of these results, it is natural to think of elements of effectively meagre subsets of \mathfrak{C} as being incomparably less common than elements of effectively co-meagre subsets of \mathfrak{C} , even when the meagre and co-meagre sets in question are both dense as subsets of \mathcal{C} .

Remark 3.5. A further reason (due to Lisagor [43]) for this standard practice: a subset S of \mathfrak{C} is effectively meagre if and only if when the Banach–Mazur game (described in Remark 2.3 above) is played for S , Player II has a winning strategy that is computable. Another reason (due, again, to Mehlhorn [17]): each abstract complexity class is effectively meagre as a subset of the family of computable functions.

Example 3.2 (Self-Describing Sequences). Fix an enumeration M_1, M_2, \dots of the Turing machines, with associated acceptable programming system ϕ_1, ϕ_2, \dots (so that ϕ_k is the partial computable function computed by M_k). Following Blum and Blum [15], we call a sequence $\sigma \in \mathfrak{C}$ *self-describing* if σ has an initial segment of the form $1^k 0$ and is computed by M_k . As Blum and Blum note, the set S_1 of self-describing sequences is non-trivial: it follows from the Recursion Theorem that each computable binary sequence is a finite variant of a self-describing sequence—so there are arbitrarily complex sequences in S_1 .

Fortnow *et al.* [12] observe that S_1 is not effectively meagre. For, consider any computable strategy $\beta : \mathcal{B}^n \rightarrow \mathcal{B}^n$ that Player II could use to play the Banach–Mazur game for S_1 . For each $k \in \mathbb{N}$, let α_k be the following strategy that Player I might adopt: on the first turn, play $1^k 0$; on all subsequent turns, play 0. The assumption that Player I plays strategy α_k and Player II plays strategy β determines a unique sequence $\sigma_k \in \mathcal{C}$. The map $F : (k, \ell) \in \mathbb{N}^2 \mapsto \sigma_k(\ell) \in \mathcal{B}^*$ is computable. So by the Recursion Theorem, there is a $k_0 \in \mathbb{N}$ such that σ_{k_0} is computed by M_{k_0} . That is: there exists a strategy (namely, α_{k_0}) via which Player I can defeat β . So S_1 is not in effectively meagre.

Proposition 3.14. Let m be an extrapolating machine. $\mathcal{NV}^w(m)$ (the set of computable sequences weakly NV-learnable by m) is an effectively meagre subset of \mathfrak{C} .

Proof. A straightforward adaptation of the proof of Proposition 3.6, appealing to the fact that when m is computable, the map $(n, w) \mapsto w^* \notin A_n$ used there is computable. \square

*Corollary 3.3 (Fortnow *et al.* [12]).* Let m be an extrapolating machine. $\mathcal{NV}(m)$ (the set of computable sequences NV-learnable by m) is an effectively meagre subset of \mathfrak{C} .

So there is a natural sense in which, for any computable extrapolator m , among computable sequences, those (weakly) NV-learnable by m are incomparably less common than those not (weakly) NV-learnable by m . The problem of (weakly) NV-learning computable sequences is formidably difficult.

Corollary 3.4. No extrapolating machine can (weakly) NV-learn each self-describing sequence.

Remark 3.6. It is illuminating to situate these results with respect to a couple of results that Fortnow *et al.* [12] establish concerning the identification problem for binary sequences.

- i) They show via an effective Banach–Mazur argument, that any $S \in \mathcal{PEX}$ is effectively meagre.

ii) They observe that the set of self-describing functions (see Example 3.2 above) is in \mathcal{EX}_0 .²⁵

So any identification class that contains \mathcal{EX}_0 has members that are not effectively meagre.

They remark: “Since virtually every inference class is either a subset of \mathcal{PEX} or a superset of \mathcal{EX}_0 the results here settle virtually all open questions that could be raised” [12, p. 145]. Contact can be made with the present approach by recalling that $\mathcal{PEX} = \mathcal{NV}$ and that $\mathcal{EX}_0 \subset \mathcal{NV}'$.²⁶ So the first result of Fortnow *et al.* [12] noted above is our Corollary 3.3: the set of sequences NV-learnable by an extrapolating machine is effectively meagre. And since every set in \mathcal{NV}^w is effectively meagre, neither \mathcal{NV}' nor \mathcal{NV}'' is a subset of \mathcal{NV}^w . In \mathcal{NV}^w we have an example of a natural inference class that is neither a subset of \mathcal{PEX} nor a superset of \mathcal{EX}_0 .

In the present setting, in which computable extrapolators (i.e., extrapolating machines) attempt to learn computable sequences, we find that generalizing our basic model by allowing merely partially defined extrapolating machines allows us to crash through a size barrier in a way that loosening our criterion of success by allowing infinitely many errors in the sense of weak NV-learning does not—since every set in \mathcal{NV} or in \mathcal{NV}^w is effectively meagre, whereas this is not the case for every set in \mathcal{NV}' or \mathcal{NV}'' . This is the reverse of what we find if we challenge (possibly computable) extrapolators to NV-learn arbitrary sequences. In that setting, in the basic model every learner masters only countably many sequences. And this is unchanged if we countenance merely partially defined learners.²⁷ But if we loosen our criterion of success to weak NV-learnability we crash through a cardinality barrier, as each learner weakly NV-learns uncountably many sequences.

Remark 3.7 (Coarse Computability). Jockusch and Schupp [37, p. 472] remark that “In recent years, there has been a general realization that worst-case complexity measures, such as P , NP , exponential time, and just being computable, often do not give a good overall picture of the difficulty of a problem.” As an example, they observe that although there exist finitely presented groups with unsolvable word problems, in every such group the words expressing the identity have vanishing asymptotic density, when words are enumerated in lexicographic order. So the linear-time algorithm that on the input of any word guesses that that word does not express the identity would make a negligible set of errors if fed all words in lexicographic order. If we demand perfection, then the word problem is impossibly hard—but if we can live with making mistakes a negligible fraction of the time, it is as easy as could be. This motivates Jockusch and Schupp to introduce a generalization of the notion of computability: a sequence is *coarsely computable* if it differs from some computable sequence in a set of bits of vanishing asymptotic density.

Coarse computability implies computable weak NV-learnability: if $\sigma \in \mathcal{C}$ differs from $\sigma^* \in \mathfrak{C}$ only in bits of vanishing asymptotic density, then the extrapolating machine that assumes it is being shown σ^* on any input weakly NV-learns σ .

But weak NV-learnability does not imply coarse computability. Let σ_0 be uncomputable. Construct a sequence σ_1 as follows: begin with two copies of the first bit of σ_0 , followed by four copies of the second bit of σ_0 , \dots followed by 2^k copies of the k th bit of σ_0 , \dots . Suppose that σ_1 is coarsely computable. Then there must be a computable sequence σ_2 that differs

²⁵Consider a learner who is silent until a data set of the form 1^k0 is seen and who from then on assumes that the data stream is being generated by M_k .

²⁶See Case and Smith [9]: that $\mathcal{PEX} = \mathcal{NV}$ is their Theorem 2.19 (attributed to private communications from van Leeuwen and Bärzdiņš); that $\mathcal{EX}_0 \subset \mathcal{NV}'$ is their Theorem 2.28.

²⁷Thanks to an anonymous referee for pointing this out.

from σ_1 only in a set of bits of vanishing asymptotic density. Define a new sequence σ_3 as follows: make the first bit of σ_3 a 0 if at least one of the first two bits of σ_2 is a 0, otherwise make it a 1; make the second bit of σ_3 a 0 if at least two of the next four bits of σ_2 are 0, otherwise make it a 1; \dots ; make the k th bit of σ_3 a 0 if at least 2^{k-1} of the next 2^k bits of σ_2 are 0, otherwise make it a 1; \dots . Since σ_2 is computable (by assumption), so is σ_3 . But σ_3 is a finite variant of σ_0 and so must be uncomputable. So there can be no such σ_2 : σ_1 is not coarsely computable. But σ_1 is weakly NV-learned by the extrapolating machine that predicts the first bit will be a 1 then subsequently predicts that each bit will be the same as the last bit seen.

4. FORECASTING

So far we have set our learners the problem of recognizing which binary sequence is being revealed in the data stream—where such recognition consists in becoming good at predicting future bits. In effect, we have been picturing that in generating new bits, Nature simply consults a lookup table chosen in advance and that the learner’s job is to attempt to guess which of the possible such tables is being used (or, in the case of weak learning, to attempt to come close to guessing the right table, in a certain sense).

We might instead picture a different sort of procedure. Suppose that what Nature has chosen in advance is not a sequence but, rather, a measure $\lambda \in \mathcal{P}$ (i.e., a Borel probability measure on \mathcal{C}) and that the learner’s data stream is randomly sampled from λ . So we now picture Nature as being equipped with a complete set of biased coins and an instruction manual that says which coin to toss to generate the next bit, given the bits that have been generated so far. To mention just some of the tamest possibilities: Nature may have chosen a Bernoulli measure, in which case the instruction will be to use the same coin to generate each new bit; or Nature may have chosen a measure corresponding to a Markov chain, in which case the coin chosen to generate a new bit will depend only on some fixed finite number of immediately preceding bits; or Nature could have chosen a delta-function measure concentrated on a single sequence, in which case only a maximally biased coin will ever be used.

Definition 4.1 (Sources). A *source* is a Borel probability measure on \mathcal{C} .

In what follows, we will think of Nature as having chosen a source $\lambda \in \mathcal{P}$ from which our learner’s data stream is sampled. Recall for $w \in \mathcal{B}^*$, we write $\lambda(w)$ in place of $\lambda(B_w)$. Similarly, for $s = 0, 1$ and $w \in \mathcal{B}^n$, we will write $\lambda(s|w)$ for the conditional probability λ gives for the $(n + 1)$ st bit to be s given that the first n bits were given by w .

How should a learner proceed in the setting where the data stream is given by a probabilistic source? In the setting of Section 3, where we were thinking of new bits as being generated by a deterministic process, we asked learners to choose an extrapolator that would allow them to definitively predict at each stage what the next bit would be, given the data seen so far. That approach would be suboptimal in the present setting: if Nature is using the fair coin measure (the Bernoulli measure of bias .5) to generate the data stream, then (with probability one) no extrapolator will do better (or worse) than random in its predictions of the next bit—but the fact that Nature is using this procedure seems like a paradigm example of the sort of thing that we ought to be able to learn by looking at data. Such learning will be possible if we ask agents to choose a forecasting procedure that allows them to issue a forecast probability before each bit is revealed, rather than choosing an extrapolator that at each stage issues definitive predictions regarding the next bit.

A natural way to encode such a strategy for learning would be via a *confirmation function*: a map $\tilde{\mu} : \mathcal{B} \times \mathcal{B}^* \rightarrow (0, 1)$ with the feature that for all $w \in \mathcal{B}^*$, $\tilde{\mu}(1 | w) + \tilde{\mu}(0 | w) = 1$. In fact, it is more convenient to employ a slightly different representation. Note that any $\tilde{\mu}$ of the above form induces, for each n , a probability measure μ_n on \mathcal{B}^n . Further, for any such $\tilde{\mu}$ and $m \leq n$, μ_m and μ_n are consistent.²⁸ So by the Kolmogorov Consistency Theorem, $\tilde{\mu}$ induces a measure μ on \mathcal{C} (with $\tilde{\mu}$ computable if and only if μ is).²⁹

Example 4.1. Define $\tilde{\mu} : \mathcal{B} \times \mathcal{B}^* \rightarrow (0, 1)$ as follows: if $w \in \mathcal{B}^n$ contains k 1's, then $\tilde{\mu}(1 | w) = \frac{k+1}{n+2}$ and $\tilde{\mu}(0 | w) = \frac{n-k+1}{n+2}$. This map satisfies the condition that for all $w \in \mathcal{B}^*$, $\tilde{\mu}(w, 1) + \tilde{\mu}(w, 0) = 1$. The corresponding measure is the Laplace–Bayes prior (the Lebesgue–uniform mixture of the Bernoulli measures).

Not all measures in \mathcal{P} correspond in this way to such $\tilde{\mu}$: $\mu \in \mathcal{P}$ corresponds to a $\tilde{\mu}$ of the above form if and only if it is a measure of full support (i.e., it assigns positive weight to each open set—or, equivalently, to each basic open set B_w).

Definition 4.2 (Forecasters). A *forecaster* is a Borel probability measure on \mathcal{C} of full support. We denote the family of forecasters by \mathcal{F} .

Definition 4.3 (Forecasting Machines). A *forecasting machine* is a computable Borel probability measure on \mathcal{C} of full support. We denote the family of forecasting machines by \mathfrak{F} .

We are going to distinguish three criteria for successful next-chance learning.³⁰ The most restrictive one, due to Blackwell and Dubins [20], requires that the forecaster eventually offer answers arbitrarily similar to those of the source concerning any (measurable) question that might be asked about the data stream.³¹ The intermediate one, due to Kalai and Lehrer [21], requires that the forecaster's probabilistic predictions concerning the next bit eventually approach the true values arbitrarily closely.³² The least restrictive one, due to Lehrer and Smorodinsky [22], relaxes this last requirement by allowing errors, so long as they eventually become arbitrarily rare.

Definition 4.4 (Strong NC-Learning). We say that forecaster μ *strongly NC-learns* source λ (or that λ is *strongly NC-learnable* by μ) if with λ -probability 1 the data stream $\sigma \in \mathcal{C}$ satisfies:

$$\lim_{n \rightarrow \infty} \sup_{A \in \mathfrak{B}} |\mu(A | \sigma[n]) - \lambda(A | \sigma[n])| = 0$$

(recall that \mathfrak{B} denotes the family of Borel subsets of \mathcal{C}).

Definition 4.5 (NC-Learning). We say that forecaster μ *NC-learns* source λ (or that λ is *NC-learnable* by μ) if with λ -probability 1 the data stream $\sigma \in \mathcal{C}$ satisfies

$$\lim_{n \rightarrow \infty} \mu(s | \sigma[n]) - \lambda(s | \sigma[n]) = 0 \quad s = 0, 1.$$

²⁸See item (v) of Section 2.1 for the relevant notion of consistency.

²⁹For a treatment of the the Kolmogorov Consistency Theorem for the special case of measures on \mathcal{C} , see [44]. For a general treatment, see, e.g., [24, Chapter V].

³⁰Investigation of inductive learning as next-chance learning traces back to Solomonoff [19]. Several criteria of success are prevalent in the literature on Solomonoff induction [11, 45, 46]. But these differ from those considered below in their focus on average or expected performance.

³¹Note that in the deterministic setting of Section 3 above, the distinction between eventually becoming good at answering all questions and eventually becoming good at predicting the next bit collapsed—recall Proposition 3.1 above.

³²For relations between this criterion of success and those alluded to in fn. 30 above, see [23].

Definition 4.6 (Weak NC-Learning). We say that forecaster μ *weakly NC-learns* source λ (or that λ is *weakly NC-learnable* by μ) if with λ -probability 1, the data stream $\sigma \in \mathcal{C}$ satisfies

$$\lim_{n \in K \rightarrow \infty} \mu(s | \sigma[n]) - \lambda(s | \sigma[n]) = 0 \quad s = 0, 1$$

for some $K \subset \mathbb{N}$ with asymptotic density one.

Remark 4.1 (Weaker Variants of Weak NC-Learning.). In parallel with the definition of NV(r)-learning (see Remark 3.1 above), we could introduce, for each $r \in (0, 1]$ a notion of NC(r)-learning, by altering the definition of weak NC-learning to require K to have asymptotic density at least r . All of the propositions below continue to hold if ‘weak NC-learning’ is replaced by ‘NC(r)-learning’ for any $r \in (0, 1]$ (except Proposition 4.10, which requires the restriction $r > 1/2$).

Example 4.2. If μ is a forecaster, then μ is also a source and it is immediate that μ strongly NC-learns, NC-learns, and weakly NC-learns μ .

Proposition 4.1 (Kalai, Lehrer, and Smorodinsky [21, 22]). For any source λ and any forecaster μ , strong NC-learnability of λ by μ implies NC-learnability (but not conversely) and NC-learnability of λ by μ implies weak NC-learnability (but not conversely).

Proof. Strong NC-learnability implies NC-learnability: in the definition of strong NC-learnability, for each n take A to be the event of the $(n + 1)$ st bit being a 1. To see that the converse is not true, consider the family $\{\lambda_p | p \in (0, 1)\}$ of (non-extreme) Bernoulli measures and let μ be the Laplace–Bayes prior. It is a basic fact about μ that it is statistically consistent for the problem of identifying the bias of a coin from knowledge of outcomes of a sequence of tosses [47]. It follows that the forecaster μ NC-learns each λ_p . But μ does not strongly NC-learn any λ_p : let E_p be the event that the limiting relative frequency of 1’s in the data stream is p ; then $\lambda_p(E_p) = 1$ but $\mu(E_p) = 0$; so for any $w \in \mathcal{B}^*$, $|\mu(E_p | w) - \lambda_p(E_p | w)| = 1$. Clearly, NC-learnability implies weak NC-learnability. To see that the converse is not true, take μ to be the fair coin measure and take λ to be the source that generates bits s_1, s_2, \dots as follows: for $k = 10^m$, s_k is the m th bit in the binary expansion of π ; all other s_j are generated by flipping a fair coin. The forecaster μ weakly NC-learns this λ but does not NC-learn it, since there are large discrepancies between the forecast probabilities and the true probabilities at arbitrarily late times. \square

We are going to see that relative to each of these three criteria, the problem of next-chance learning is formidably difficult and involves hard choices.

4.1. STRONG NC-LEARNING

A famous result and its converse give a necessary and sufficient condition for a source to be strongly NC-learnable by a forecaster.

Proposition 4.2 (Blackwell and Dubins [20]). If source λ is absolutely continuous with respect to forecaster μ (i.e., $\lambda(A) > 0$ implies $\mu(A) > 0$ for all $A \in \mathfrak{B}$), then μ strongly NC-learns λ .

Proposition 4.3 (Lehrer and Smorodinsky [22]). If forecaster μ strongly NC-learns source λ , then λ is absolutely continuous with respect to μ .

Proposition 4.4. Let μ be a forecaster. The sources strongly NC-learnable by μ form a dense subset of \mathcal{P} of cardinality \mathfrak{c} .

Proof. Let w_1, \dots, w_n be binary strings such that \mathcal{C} is a disjoint union of the B_{w_k} . And let $p_1, \dots, p_n \in (0, 1)$ with $\sum_{k=1}^n p_k = 1$. Since the B_{w_k} partition \mathcal{C} , each $w \in \mathcal{B}$ is either one of the w_k , or a proper prefix of some of the w_k , or a proper extension of one of the w_k . We define a map $\bar{\lambda} : \mathcal{B}^* \rightarrow [0, 1]$ as follows:

- (a) If $w = w_k$ for some k , then $\bar{\lambda}(w) = p_k$.
- (b) If w is a prefix of $w_{j_1}, \dots, w_{j_\ell}$, then $\bar{\lambda}(w) = \sum_{k=1}^{\ell} p_{j_k}$.
- (c) If w is of the form $w_k.v$ for some binary string v , then $\bar{\lambda}(w) = p_k \cdot \mu(v | w_k)$.

It is immediate that $\bar{\lambda}(\emptyset) = 1$ and that $\bar{\lambda}(w) = \bar{\lambda}(w.0) + \bar{\lambda}(w.1)$ for each $w \in \mathcal{B}^*$. So by the Carathéodory Extension Theorem, $\bar{\lambda}$ extends uniquely to a measure $\lambda \in \mathcal{C}$ such that $\lambda(w) = \bar{\lambda}(w)$ for each $w \in \mathcal{B}^*$.

The source λ is strongly NC-learnable by μ . For suppose that A is a Borel subset of \mathcal{C} with $\lambda(A) > 0$. Given the law of total probability,

$$\lambda(A) = \sum_{k=1}^n \lambda(A | w_k) \lambda(w_k),$$

there must be some $1 \leq \ell \leq n$ such that $\lambda(A | w_\ell) > 0$. It follows that $\mu(A | w_\ell) > 0$. And since $\mu(w_\ell)$ is also positive (μ being a forecaster) we find that $\mu(A) > 0$ (by the law of total probability, again). So λ is absolutely continuous with respect to μ and Proposition 4.2 tells us that μ strongly NC-learns λ . And since the w_k and the p_k can be chosen arbitrarily, we construct in this way continuum-many such sources in any finite intersection of sub-basic open sets of \mathcal{P} . \square

The next result follows from the stronger Proposition 4.9 below, but we include it here in order to indicate an independent route to establishing it.

Proposition 4.5 (Noguchi [48]). For any $\mu \in \mathcal{F}$, the set $S_\mu \subset \mathcal{P}$ of sources strongly NC-learned by μ is meagre in \mathcal{P} .

Proof. A classical result tells us that for any measure in \mathcal{P} , there is some meagre subset of \mathcal{C} to which it assigns probability 1.³³ And Proposition 1 of [52] tells us that for any meagre subset of \mathcal{C} , the set of probability measures that assign it positive probability is meagre in \mathcal{P} .³⁴ So let A be a meagre subset of \mathcal{C} such that $\mu(A) = 1$ and let $P_A \subset \mathcal{P}$ be the, necessarily meagre, set of measures that assigns A positive probability. By Proposition 4.3, if $\lambda \in S_\mu$, then $\lambda \in P_A$. So S_μ , being a subset of a meagre set, is meagre. \square

Corollary 4.1. The set $\{\lambda \in \mathcal{P} \mid \exists \mu \in \mathfrak{F} \text{ such that } \mu \text{ strongly NC-learns } \lambda\}$ is meagre in \mathcal{P} .

Proposition 4.6. For any $\mu \in \mathcal{F}$, the set J_μ of forecasters that fail to strongly NC-learn any sources strongly NC-learned by μ is co-meagre in \mathcal{P} .

Proof. Let N_μ be the set of $\nu \in \mathcal{P}$ such that there is no $\lambda \in \mathcal{P}$ that is absolutely continuous with respect to both μ and ν . By Propositions 4.2 and 4.3, $J_\mu \subseteq N_\mu \cap \mathcal{F}$. So it suffices to show that N_μ and \mathcal{F} are both co-meagre subsets of \mathcal{P} .

N_μ is co-meagre. Let A and P_A be as in the proof of the preceding proposition. Suppose

³³Szpilrajn [49] shows that any non-atomic Borel probability measure on a separable metric space assigns measure 0 to some co-meagre set. Marczewski (=Szpilrajn) and Sikorski [50] observe that this result implies that in a separable metric space without isolated points, every Borel probability measure assigns probability 0 to some co-meagre set. In fact, the hypothesis of separability can be dropped [51, Corollary 3.7].

³⁴This is a special case of a result of Koumoullis [53].

that $\nu \in \mathcal{P}$ is not in N_μ . So there is a $\lambda \in \mathcal{P}$ absolutely continuous with respect to both μ and ν . So λ must assign the complement of A zero probability (since μ does), which means that ν must assign A positive probability (since λ does)—so λ is in the complement of the co-meagre set \mathcal{P}_A . So the complement of N_μ is meagre, being a subset of the meagre set \mathcal{P}_A . \mathcal{F} is co-meagre. The forecasters form a dense G_δ subset of \mathcal{P} [54, §3.13]. And in any completely metrizable space (such as \mathcal{P}), any dense G_δ subset is co-meagre [55, Theorem 9.2]. \square

Corollary 4.2. For any forecaster, there is another, such that the sets of sources strongly NC-learned by the two forecasters are disjoint.

As usual, we call a sequence $\{\lambda_i\}_{i \in \mathbb{N}}$ of elements of \mathfrak{P} *uniformly computable in i* if there is a computable $F : \mathbb{N} \times \mathcal{B}^* \times \mathbb{N} \rightarrow \mathbb{Q}$ such that $|\lambda_i(w) - F(i, w, n)| \leq 2^{-n}$, for all $i, n \in \mathbb{N}$ and $w \in \mathcal{B}^*$.

Proposition 4.7. (a) Let $\mu \in \mathcal{F}$ and let S be a countable subset of \mathcal{P} . Then there is a $\mu^* \in \mathcal{F}$ that strongly NC-learns every source in S as well as every source strongly NC-learned by μ . (b) Let $\mu \in \mathfrak{F}$ and let $S = \{\lambda_i\}_{i \in \mathbb{N}}$ be a sequence of measures in \mathfrak{P} that is uniformly computable in i . Then there is a $\mu^* \in \mathfrak{F}$ that strongly NC-learns every source in S as well as every source strongly NC-learned by μ .

Proof. For part (a), enumerate the members of S : $\lambda_1, \lambda_2, \dots$ and set

$$\mu^* = \frac{1}{2}\mu + \frac{1}{2} \sum_{k=1}^{\infty} \frac{1}{2^k} \lambda_k.$$

Each λ_k is absolutely continuous with respect to μ^* , so by Proposition 4.2, μ^* strongly NC-learns every source in S . And if ν is a source strongly NC-learned by μ , then by Proposition 4.3, ν must be absolutely continuous with respect to μ and hence also with respect to μ^* —so by Proposition 4.2, μ^* strongly NC-learns ν .

For part (b), we can proceed in the same way. The only thing to check is that if $\mu \in \mathfrak{P}$ is computable and $\{\lambda_i\} \subset \mathfrak{P}$ is uniformly computable in i , then the measure μ^* as defined above is also computable. To this end, suppose that $F_0 : \mathcal{B}^* \times \mathbb{N} \rightarrow \mathbb{Q}$ and $F : \mathbb{N} \times \mathcal{B}^* \times \mathbb{N} \rightarrow \mathbb{Q}$ are computable, with $|\mu(w) - F_0(w, n)| \leq 2^{-n}$ and $|\lambda_i(w) - F(i, w, n)| \leq 2^{-n}$, for all $w \in \mathcal{B}^*$ and $i, n \in \mathbb{N}$.

We define $F^* : \mathcal{B}^* \times \mathbb{N} \rightarrow \mathbb{Q}$ as follows:

$$F^*(n, w) := \frac{1}{2}F_0(w, n+1) + \frac{1}{2} \sum_{k=1}^{n+1} \frac{1}{2^k} F(k, w, 2n)$$

Then for any given $w \in \mathcal{B}^*$ and $n \in \mathbb{N}$, we define $\alpha, \beta, \gamma \in \mathbb{R}$:

$$\begin{aligned} \alpha &:= \frac{1}{2} (\mu(w) - F_0(w, n+1)) \\ \beta &:= \sum_{k=1}^{n+1} \frac{1}{2^{k+1}} (\lambda_k(w) - F(k, w, 2n)) \\ \gamma &:= \sum_{k=n+2}^{\infty} \frac{1}{2^{k+1}} \lambda_k(w). \end{aligned}$$

Note that each of $|\alpha|$, $|\beta|$, and $|\gamma|$ is no greater than $2^{-(n+2)}$. In the case of $|\alpha|$, this follows from what we know about F_0 . For $|\beta|$, we have:

$$\begin{aligned}
|\beta| &\leq \sum_{k=1}^{n+1} \frac{1}{2^{k+1}} |\lambda_k(w) - F(k, w, 2n)| \\
&\leq \sum_{k=1}^{n+1} \frac{1}{4} |\lambda_k(w) - F(k, w, 2n)| \\
&\leq \sum_{k=1}^{n+1} \frac{1}{4} 2^{-2n} \\
&= \frac{n+1}{2^n} 2^{-(n+2)},
\end{aligned}$$

and for any $n \geq 1$ we have that $n+1 \leq 2^n$. And since for each k we have $0 \leq \lambda_k(w) \leq 1$, we have that $|\gamma| \leq \sum_{k=n+2}^{\infty} \frac{1}{2^{k+1}}$.

Now, F^* is computable and we have:

$$\begin{aligned}
|\mu^*(w) - F^*(w, n)| &= |\alpha + \beta + \gamma| \\
&\leq |\alpha| + |\beta| + |\gamma| \\
&< 2^{-n}.
\end{aligned}$$

So μ^* is computable. □

Thus we have a no-free-lunch result for strong NC-learning: the set of sources strongly NC-learned by any forecaster is uncountable and dense but meagre; for every (computable) forecaster there is another (computable) forecaster that strongly NC-learns everything it does, plus a further countably infinite set of sources; and for every forecaster there is another that strongly NC-learns a disjoint set of sources (indeed, typical forecasters have this feature).

4.2. NC-LEARNING AND WEAK NC-LEARNING

Proposition 4.5 above tells us that each forecaster strongly NC-learns a dense and uncountable but meagre set of sources. This implies that the sets of sources NC-learned and weakly NC-learned by any forecaster are also dense and uncountable. Strong NC-learning is, intuitively, a much more restrictive notion than NC-learning: being able to accurately answer all questions about the data stream, including questions about the infinite future, is much a much more demanding standard than being able to accurately estimate the chances for the next bit.³⁵ Similarly, NC-learning is, intuitively, a much more restrictive notion than weak NC-learning: we saw in Section 3 above that weakening NV-learning by allowing an infinite number of errors (so long as they were of asymptotic density zero) made a marked difference to the size of the set of sequences that a given extrapolator could learn—any extrapolator NV-learns a countable set of sequences but weakly NV-learns an uncountable set of sequences. So it is not obvious that the set of sources (weakly) NC-learned by a given forecaster should always be meagre.³⁶ Not obvious—but, as we will see, nonetheless true.

³⁵Indeed, there is a sense in which strong NC-learning implies *rapid* NC-learning, and a sense in which the converse implication holds [56, Propositions 2 and 3].

³⁶Noguchi [48, p. 433], after discussing the results cited above in fn. 35, remarks that: “These results lead us to conjecture that, in general, a merged set (of probability measures) [i.e., a set of sources strongly NC-learned by a given forecaster] may be much smaller than a weakly merged set [i.e., a set of sources NC-learned

We begin by introducing a basis \mathcal{W} for the weak topology on \mathcal{P} . First, for each $k \in \mathbb{N}$, we fix a metric on \mathcal{P}_k compatible with its topology: we take the distance between $\lambda, \mu \in \mathcal{P}_k$ to be given by:

$$d(\mu, \lambda) := \max_{w \in \mathcal{B}^k} |\mu(w) - \lambda(w)|.$$

In terms of our identification of \mathcal{P}_k with a closed subset of \mathbb{R}^{2^k} , this is the metric induced by the ℓ^∞ norm on \mathbb{R}^{2^k} . For $\mu \in \mathcal{P}_k$ and $\varepsilon > 0$, we write $B(\mu, \varepsilon)$ for the open metric ball of radius ε centred at μ :

$$B(\mu, \varepsilon) := \{\lambda \in \mathcal{P}_k \mid d(\mu, \lambda) < \varepsilon\}.$$

We call $B(\mu, \varepsilon) \subset \mathcal{P}_k$ *rational* if $\varepsilon \in \mathbb{Q}$ and $\mu(w) \in \mathbb{Q}$ for each $w \in \mathcal{B}^k$.

For each $k \in \mathbb{N}$, let $\Pi_k : \mathcal{P} \rightarrow \mathcal{P}_k$ be the restriction map: for $\mu \in \mathcal{P}$, $\Pi_k(\mu)$ is the measure in \mathcal{P}_k such that $\Pi_k(\mu)(w) = \mu(w)$ for each $w \in \mathcal{B}^k$. We now take \mathcal{W} to comprise the inverse images under the Π_k of the rational open metric balls in the various \mathcal{P}_k :

$$\mathcal{W} := \{W = \Pi_k^{-1}(B(\mu, \varepsilon)) \mid k \in \mathbb{N}, \mu \in \mathcal{P}_k, \mu(w) \in \mathbb{Q} \forall w \in \mathcal{B}^k, \varepsilon > 0, \varepsilon \in \mathbb{Q}\}.$$

Proposition 4.8. \mathcal{W} is a basis for the weak topology on \mathcal{P} .

Proof. It suffices to show: (i) that each $W \in \mathcal{W}$ is open; and (ii) that for any non-empty open set $U \subset \mathcal{P}$ and for any $\nu \in U$, there is a $W \in \mathcal{W}$ with $\nu \in W \subset U$.

(i) Fix $W \in \mathcal{W}$ of the form $W = \Pi_k^{-1}(B(\mu, \varepsilon))$. Let w_1, \dots, w_{2^k} be an enumeration of the k -bit strings. And for each $1 \leq j \leq 2^k$, let $p_j := \max\{0, \mu(w_j) - \varepsilon\}$ and $q_j := \min\{1, \mu(w_j) + \varepsilon\}$. Then we have:

$$\begin{aligned} W &= \left\{ \lambda \in \mathcal{P} \mid \max_{1 \leq j \leq 2^k} |\mu(w_j) - \lambda(w_j)| < \varepsilon \right\} \\ &= \bigcap_{j=1}^{2^k} S_{w_j, p_j, q_j}, \end{aligned}$$

where each S_{w_j, p_j, q_j} is a sub-basic open subset of \mathcal{P} (as in item (vi) of Section 2.1 above). So W is an open subset of \mathcal{P} .

(ii) It suffices to consider a non-empty open set $U \subset \mathcal{P}$ that is a finite intersection of sub-basic open sets. Let $S_{w_1, p_1, q_1}, \dots, S_{w_n, p_n, q_n}$ be arbitrary sub-basic open subsets of \mathcal{P} and suppose that $U := \bigcap_{k=1}^n S_{w_k, p_k, q_k} \neq \emptyset$. Let $N = \max\{|w_1|, \dots, |w_n|\}$ and let $\nu \in U$. Note that each $\Pi_N(S_{w_j, p_j, q_j})$ is an open subset of \mathcal{P}_N : each condition of the form $p_j < \Pi_N(\mu)(w_j) < q_j$ just imposes an inequality on (sums of) differences of coordinate relative to our identification of \mathcal{P}_N with a subset of \mathbb{R}^{2^N} . So we can find a rational open metric ball B contained in $\Pi_N(U)$ with $\Pi_N(\nu) \in B$. Letting $W := \Pi_N^{-1}(B) \in \mathcal{W}$, we have $\nu \in W \subset U$. \square

Proposition 4.9. Let μ be a forecaster. The sources weakly NC-learnable by μ form a meagre subset of \mathcal{P} .

Proof. For any source λ and $k \in \mathbb{N}$, let us say that (μ, λ) considers k *bad* if for each $w \in \mathcal{B}^k$ we have

$$|\mu(s \mid w) - \lambda(s \mid w)| \geq \frac{1}{5} \quad s = 0, 1.$$

by a given forecaster].” He then goes on to observe that each forecaster strongly NC-learns a meagre set of sources—so it is natural to read him as conjecturing that the set of sources NC-learned by a forecaster need not be meagre.

And let us say that (μ, λ) considers k *super-bad* if (μ, λ) considers more than half of the $j \leq k$ to be bad. And, by extension, for any subset $S \subset \mathcal{P}$, let us say that (μ, S) considers $k \in \mathbb{N}$ (super-)bad if (μ, λ) does for each $\lambda \in S$.

For each $n \in \mathbb{N}$, let F_n be the set of $\lambda \in \mathcal{P}$ such that (μ, λ) considers at least n natural numbers to be super-bad. And let A_n be the complement of F_n in \mathcal{P} . Note that μ cannot NC-learn λ if there are infinitely many $k \in \mathbb{N}$ that (μ, λ) considers bad and that μ cannot weakly NC-learn λ if there are infinitely many $k \in \mathbb{N}$ that (μ, λ) considers super-bad. So if μ weakly NC-learns λ , then (μ, λ) can consider only finitely many natural numbers to be super-bad, which means that there will be an N such that $\lambda \notin F_N$, which implies that $\lambda \in A := \bigcup_{n \in \mathbb{N}} A_n$. So in order to establish our proposition, it suffices to show that each A_n is nowhere dense in \mathcal{P} .

The first step is to suppose that we are given a set $W \in \mathcal{W}$ of the form $\Pi_k^{-1}(B(\lambda, \varepsilon))$ and to show how to find $W_1 \in \mathcal{W}$ of the form $W_1 = \Pi_{k+1}^{-1}(B(\lambda_1, \varepsilon_1))$, such that $W_1 \subset W$ and (μ, W_1) considers $k + 1$ to be bad.

First, we select λ_1 . For each $w \in \mathcal{B}^k$, if $\mu(w.0) \geq \mu(w.1)$, we set

$$\lambda_1(w.0) = \frac{1}{10} \cdot \lambda(w) \quad \text{and} \quad \lambda_1(w.1) = \frac{9}{10} \cdot \lambda(w);$$

otherwise we set

$$\lambda_1(w.0) = \frac{9}{10} \cdot \lambda(w) \quad \text{and} \quad \lambda_1(w.1) = \frac{1}{10} \cdot \lambda(w).$$

This gives us a well-defined $\lambda_1 \in \mathcal{P}_{k+1}$ that assigns rational values to each string in \mathcal{B}^{k+1} .

We now select $\varepsilon_1 > 0$ as follows: we choose m large enough so that $\varepsilon_1 = 10^{-m}$ is small enough so that for any λ' in $B(\lambda_1, \varepsilon_1)$, for each $w \in \mathcal{B}^k$, if $\mu(w.0) \geq \mu(w.1)$, then

$$\lambda'(w.1) > \frac{8}{10} \cdot \lambda(w),$$

and if $\mu(w.0) < \mu(w.1)$ then,

$$\lambda'(w.0) > \frac{8}{10} \cdot \lambda(w).$$

This process can be iterated. In particular, if we are given $W \in \mathcal{W}$ of the form $\Pi_k^{-1}(B(\lambda, \varepsilon))$, we can run the process once to construct $W_1 \in \mathcal{W}$ with $W_1 \subset W$ such that (μ, W_1) considers $k + 1$ bad; applying the process again (now with W_1 in place of W) yields a $W_2 \in \mathcal{W}$ with $W_2 \subset W_1$ such that (μ, W_2) considers $k + 2$ bad and so on.

So if we are given $W \in \mathcal{W}$ of the form $\Pi_k^{-1}(B(\lambda, \varepsilon))$, we can run the process $k + n$ times to yield $W^* := W_{k+n} \in \mathcal{W}$ such that $W^* \subset W$ and (μ, W^*) considers at least n numbers to be super-bad, so that $W^* \cap A_n = \emptyset$. So A_n is nowhere dense in \mathcal{P} . \square

Corollary 4.3. The set $\{\lambda \in \mathcal{P} \mid \exists \mu \in \mathfrak{F} \text{ such that } \mu \text{ weakly NC-learns } \lambda\}$ is meagre in \mathcal{P} .

It of course follows that the set of sources (strongly) NC-learnable by a given forecaster are likewise meagre—and that set of all sources collectively (strongly) learnable by forecasting machines are likewise meagre.

Remark 4.2. The set of sources not even weakly NC-learnable by a given forecaster μ is a co-meagre subset of \mathcal{P} , and so is uncountable. A variant on the proof of the above proposition shows that even if the continuum hypothesis fails, this set has the cardinality of the continuum. Fix a metric on \mathcal{P} compatible with the weak topology, such as the Prokhorov

metric.³⁷ And let us amend the iterative procedure of the proof of the preceding proposition so that the diameter of W_{k+1} relative to this metric is no more than half of the diameter of W_k . Then if we are given $W \in \mathcal{W}$ and repeatedly apply our revised iterative procedure, we will construct a sequence W_1, W_2, \dots of nested \mathcal{W} -sets such that $\bigcap_{k=1}^{\infty} W_k$ contains a single source, which is not even weakly NC-learnable by μ . There are continuum-many distinct W we could use to initiate this procedure—these determine continuum-many distinct sources not even weakly NC-learnable by μ .

Proposition 4.10. Let μ be a forecaster. Then there is a second forecaster μ^\dagger such that the sets of sources weakly NC-learned by μ and by μ^\dagger are disjoint. If μ is computable, we can take μ^\dagger to be likewise computable.

Proof. Let μ be given. We construct a map $\nu : \mathcal{B}^* \rightarrow [0, 1]$ inductively as follows:

- a) $\nu(\emptyset) = 1$;
- b) Supposing that $\nu(w)$ is given, we define $\nu(w.0)$ and $\nu(w.1)$ as follows:
 - i) if $\mu(0 | w) \leq \mu(1 | w)$, then $\nu(w.0) = 9/10 \cdot \nu(w)$ and $\nu(w.1) = 1/10 \cdot \nu(w)$;
 - ii) if $\mu(0 | w) > \mu(1 | w)$, then $\nu(w.0) = 1/10 \cdot \nu(w)$ and $\nu(w.1) = 9/10 \cdot \nu(w)$.

Clearly, for any $w \in \mathcal{B}^*$, $\nu(w) = \nu(w.0) + \nu(w.1)$. So by the Carathéodory Extension Theorem, ν extends to a unique Borel probability measure on \mathcal{C} , which we take as our μ^\dagger .

For any non-empty $w \in \mathcal{B}^*$, $|\mu(1 | w) - \mu^\dagger(1 | w)| \geq \frac{2}{5}$. So for any $\lambda \in \mathcal{P}$, any $\sigma \in \mathcal{C}$, and $n \in \mathbb{N}$ we have:

$$\max\{|\mu(1 | \sigma[n]) - \lambda(1 | \sigma[n])|, |\mu^\dagger(1 | \sigma[n]) - \lambda(1 | \sigma[n])|\} \geq \frac{1}{5}.$$

So there can be no $\lambda \in \mathcal{P}$ such that for every σ in a set of λ -measure one, there is a set K of natural numbers of asymptotic density 1, such that for sufficiently large $n \in K$, $\mu(1 | \sigma[n])$ and $\mu^\dagger(1 | \sigma[n])$ are both arbitrarily close to $\lambda(1 | \sigma[n])$ —i.e., there is no source λ that is weakly NC-learned by both μ and μ^\dagger . \square

Of course, it follows that μ and μ^\dagger also (strongly) NC-learn disjoint sets of sources.

So we have no-free-lunch theorems for (weak) NC-learning: each forecaster, computable or not, (weakly) NC-learns an uncountable and dense but meagre set of sources; and for each (computable) forecaster there is another that (weakly) NC-learns a disjoint set of sources.

Remark 4.3. Lehrer and Smorodinsky [22] show that if $\mu \in \mathcal{F}$ NC-learns $\lambda \in \mathcal{P}$, then any nontrivial mixture of μ with any $\nu \in \mathcal{P}$ weakly NC-learns λ . Ryabko and Hutter [23, Proposition 10] show that this result is sharp: they give an example of measures μ, ν , and λ where μ NC-learns λ but any non-trivial mixture of μ and ν merely weakly NC-learns λ . So there is no prospect of using the strategy of the proof of Proposition 4.7 above to prove an analogous result for NC-learning.

4.3. FORECASTING OF COMPUTABLE SOURCES

Let us now specialize to problem of (strong, weak) NC-learning for computable forecasters facing data streams generated by computable sources.³⁸

³⁷For details see, e.g., [25, pp. 72 f.].

³⁸Vitányi and Chater [57] introduce a model of learning in which agents facing a data stream generated by a computable source attempt to guess a code number for that source (so this model of learning stands to next chance learning as identification stands to extrapolation). See also [58, 59].

For $\mu \in \mathfrak{F}$, we denote by $\mathcal{NC}(\mu)$ the set of computable sources that are NC-learned by μ . We use \mathcal{NC} to denote:

$$\{S \subset \mathfrak{P} \mid \exists \mu \in \mathfrak{F} \text{ with } S \subseteq \mathcal{NC}(\mu)\}.$$

Let us likewise use $\mathcal{NC}^s(\mu)$ and $\mathcal{NC}^w(\mu)$ to denote the set of computable sources strongly NC-learned and weakly NC-learned by the forecasting machine μ and use \mathcal{NC}^s and \mathcal{NC}^w to denote the class of subsets of \mathfrak{P} that can be strongly/weakly NC-learned by some forecasting machine. It is immediate from the definitions that $\mathcal{NC}^s \subseteq \mathcal{NC}$ and that $\mathcal{NC} \subseteq \mathcal{NC}^w$. A variant on the proof of Proposition 3.9 shows that the latter containment is proper.

Proposition 4.11. $\mathcal{NC} \subset \mathcal{NC}^w$.

Proof. Let V be the subset of \mathfrak{P} consisting of δ -function measures concentrated on computable binary sequences in which 0's have vanishing asymptotic density. Let $\mu \in \mathfrak{F}$ be the measure that on input of any $w \in \mathcal{B}^n$, considers the chance of seeing a 0 next to be 2^{-n} . We have $V \in \mathcal{NC}^w(\mu)$. But suppose that $V \subseteq \mathcal{NC}(\nu)$ for some $\nu \in \mathcal{F}$. We define $\sigma \in \mathcal{C}$ as follows: σ is of the form $1^{n_1}.0.1^{n_2}.0.1^{n_3}.0\dots$ where where each n_j is chosen to be the smallest n larger than 2^j such that $\nu(1^{n_1}.0.1^{n_2}.0\dots.1^{n_{j-1}}.0.1^n) > .9$ (such n_j must exist, since by assumption ν \mathcal{NC} -learns each delta-function measure concentrated on a sequence containing only finitely many 0's). The delta-function measure concentrated on σ is in V . But $\sigma \notin \mathcal{NC}(\nu)$, since when fed σ , there are infinitely many occasions on which ν issues forecast probabilities for seeing a 0 next of less than .1, when the true chance is 1. \square

Proposition 4.12. For any $\mu \in \mathfrak{F}$, the following are dense subsets of \mathcal{P} :

- (a) $\mathcal{NC}^s(\mu)$.
- (b) The complement of $\mathcal{NC}^s(\mu)$ in \mathfrak{P} .
- (c) $\mathcal{NC}(\mu)$.
- (d) The complement of $\mathcal{NC}(\mu)$ in \mathfrak{P} .
- (e) $\mathcal{NC}^w(\mu)$.
- (f) The complement of $\mathcal{NC}^w(\mu)$ in \mathfrak{P} .

Proof. The claim concerning (a) follows via straightforward adaptation of the proof of Proposition 4.4, while that of (f) follows from Proposition 4.15 below. The other sets listed are supersets of (a) or (f). \square

For any of our senses of probabilistic learning, for any computable forecaster, that forecaster succeeds in learning a countable infinity of computable sources and fails to learn a countable infinity of computable sources. So we have parity between the learnable and the unlearnable at the level of cardinality. And this parity persists at the level of topology: a version of Sierpiński's Theorem tells us that, up to homeomorphism, there is only one countable metrizable topological space without isolated points [60, §2].

But it is straightforward to extend Mehlhorn's notion of an effectively meagre set to the context of \mathfrak{P} , with the elements of the basis \mathcal{W} for \mathcal{P} of Section 4.2 playing the role that the basic open sets B_w played in our discussion of effectively meagre subsets of \mathcal{C} in Section 3.3 above. And, as in the case of next-value learning, we find that the for our species of next-chance learning, this notion allows us to isolate a sense in which failure is incomparably more common than success.

Recall that elements of \mathcal{W} are specified by specifying a natural number k , a rational-valued measure $\mu \in \mathcal{P}_k$ (which is determined in turn by specifying the values that it assigns each

$w \in \mathcal{B}^k$), and a rational $\varepsilon > 0$. So the elements of \mathcal{W} can be effectively represented by binary strings. In the following definition we take such a coding scheme to be fixed.

Definition 4.7. Let A be a subset of \mathfrak{P} . Let $f : \mathcal{W} \rightarrow \mathcal{W}$ be a computable function. Then A is *effectively nowhere dense via f* if for each $W \in \mathcal{W}$:

- i) $f(W) \subset W$;
- ii) $A \cap f(W) = \emptyset$.

Definition 4.8. A subset A of \mathfrak{P} *effectively nowhere dense* if there is a computable $F : \mathbb{N} \times \mathcal{W} \rightarrow \mathcal{W}$ such that:

- i) for each $n \in \mathbb{N}$ there is an A_n that is effectively nowhere dense via $F(n, \cdot) : \mathcal{W} \rightarrow \mathcal{W}$;
- ii) $A = \bigcup A_n$.

The complement of an effectively meagre subset of \mathfrak{P} is *effectively co-meagre*.

The proofs of Propositions 3.12 and 3.13 are easily adapted to yield:

Proposition 4.13. The set of effectively meagre subsets of \mathfrak{P} is closed under the following operations:

- (i) taking subsets;
- (ii) taking finite unions;
- (iii) taking effective unions.

Proposition 4.14. For any $W \in \mathcal{W}$, the set $W \cap \mathfrak{P}$ is not effectively meagre.

So it is again natural to consider the elements of effectively meagre subsets of \mathfrak{P} to be incomparably less common than the elements of effectively co-meagre subsets of \mathfrak{P} .

Proposition 4.15. For any $\mu \in \mathfrak{F}$, $NC^w(\mu)$ is an effectively meagre subset of \mathfrak{P} .

Proof. A straightforward adaptation of the proof of Proposition 4.9, appealing to the fact that when μ is computable, the map $(n, W) \mapsto W^* \notin A_n$ used there is computable. \square

So there is a natural sense in which, for any computable forecaster μ , among computable sources, those weakly NC-learnable by μ are incomparably less common than those not NC-weakly learnable by μ . And, *a fortiori*, those computable sources (strongly) NC-learnable by μ are incomparably less common than those not (strongly) NC-learnable by μ . Learning in this setting is formidably difficult. And hard choices must be made: the proof of Proposition 4.10 above shows that each $\mu \in \mathfrak{F}$ has an evil twin $\mu^\dagger \in \mathfrak{F}$ such that the two weakly NC-learn disjoint sets of measures.

Remark 4.4 (Partial Forecasting Machines.) In Remark 3.6 above, we saw above that liberalizing our notion of NV-learning of computable sequences by allowing merely partial computable extrapolating machines made an interesting difference: while every set in \mathcal{NV} is effectively meagre, this is not true of every set in \mathcal{NV}' (let alone \mathcal{NV}''). What is the ‘partial’ analog of NC-learning?

Recall that a *semi-measure* on \mathcal{C} is a map $\mu : \mathcal{B}^* \rightarrow [0, 1]$ satisfying $\mu(\emptyset) \leq 1$ and $\mu(w) \geq \mu(w.0) + \mu(w.1)$ for all $w \in \mathcal{B}^*$ (with equality of course being required for measures). Such a μ is *lower semi-computable* if there exists a partial computable $\phi : \mathcal{B}^* \times \mathbb{N} \rightarrow [0, 1]$ with $\mu(w) = \lim_{\ell \rightarrow \infty} \phi(w, \ell)$ and $\phi(w, k) \leq \phi(w, k+1)$, for all $w \in \mathcal{B}^*$ and $k \in \mathbb{N}$. According to the approach deriving from [19] and [61], as computable functions stand to partial computable functions,

computable measures stand to lower semi-computable semi-measures.³⁹ This approach leads to a stunning result: there exists a lower semi-computable semi-measure μ_S that next-chance learns (in various natural senses) every computable measure (there are many such learners, in fact). But this feat is less impressive than it might at first appear [62, 63]: while μ_S is lower semi-computable, the corresponding conditional probability function $\mu_S(1|w)$ —i.e., the thing we need to calculate in order to make predictions—is merely limit computable (the problem being that the quotient of two lower semi-computable numbers need not be lower semi-computable).

Here is an alternative approach.⁴⁰ A *confirmation function* is a map $\mu : \mathcal{B} \times \mathcal{B}^* \rightarrow (0, 1)$ such that $\mu(1|w) + \mu(0|w) = 1$ (as noted above, the Kolmogorov Consistency Theorem tells us that specifying a confirmation function is the same thing as specifying a forecaster). A *semi-confirmation function* is a map $\mu : \mathcal{B} \times \mathcal{B}^* \rightarrow (0, 1)$ such that $\mu(1|w) + \mu(0|w) \leq 1$. We say that a semi-confirmation function μ is *lower semi-computable* if there is a partial computable $\phi : \mathcal{B} \times \mathcal{B}^* \times \mathbb{N} \rightarrow [0, 1]$ such that for each $s \in \mathcal{B}$, $w \in \mathcal{B}^*$, and $k \in \mathbb{N}$ we have: (a) $\mu(s|w) = \lim_{\ell \rightarrow \infty} \phi(s, w, \ell)$; and (b) $\phi(s, w, k) \leq \phi(s, w, k + 1)$. We call a lower-semi computable semi-confirmation function a *partial forecasting machine*. We say that the partial forecasting machine μ *NC'-learns* the computable source λ if with λ -probability one, when μ is fed a data stream σ generated by λ , we have: (i) $\mu(1|\sigma[k]) + \mu(0|\sigma[k]) = 1$ for each k ; and $\mu(1|\sigma[k])$ converges to $\lambda(1|\sigma[k])$ as $k \rightarrow \infty$. We say that μ *NC''-learns* λ if the previous definition holds with clause (i) weakened to allow finitely many exceptions.

We define the classes \mathcal{NC}' and \mathcal{NC}'' in the obvious way. It is straightforward to show that if a partial extrapolating machine m NV-/NV'-/NV''-learns each sequence in a set S_0 of computable sequences, then there is a partial forecasting machine μ that NC-/NC'-/NC''-learns each delta-function measure corresponding to an element of S_0 : for $w \in \mathcal{B}^n$, and $k \in \mathbb{N}$, set $\phi(1, w, k) = \phi(0, w, k) = 0$ unless simulating the computation of m on w for k steps shows that $m(w) = s$, in which case set $\phi(s, w, k) = 1 - 2^{-n}$ and set $\phi(1 - s, w, k) = 2^{-n}$. Further, if a partial forecasting machine μ NC-/NC'-/NC''-learns each member of a set S_1 of computable delta-function measures, then there is a partial forecasting machine m that NV-/NV'-/NV''-learns each sequence that is the support of one of the measures in S_1 : for $w \in \mathcal{B}^n$, compute the conditional probabilities $\mu(s|w)$ until one of them is at least a half and take the corresponding bit to be $m(w)$. So from the fact that $\mathcal{NV} \subset \mathcal{NV}' \subset \mathcal{NV}''$ it follows that $\mathcal{NC} \subset \mathcal{NC}' \subset \mathcal{NC}''$. Since there are sets in \mathcal{NV}' that are not effectively meagre in \mathfrak{C} , it is natural to expect that there are sets in \mathcal{NC}' that are not effectively meagre in \mathfrak{P} . It is known, however, that $\mathfrak{P} \notin \mathcal{NC}''$ [63].

5. DISCUSSION

Over the course of the last century, it became widely accepted that successful inductive learning is possible only against a background of biases that favour some hypotheses over others (see, e.g., [64–67]).⁴¹ No-free-lunch results substantiate this insight. If we don't presuppose anything about the binary sequence being revealed to us, then we face a formidably difficult learning problem: no matter what approach to learning we adopt, the situations that we might face, those in which we fail are incomparably more common than those in which we succeed. And no approach dominates all rivals in its range of success: for any

³⁹For full details and motivation see [11].

⁴⁰In what follows, I am indebted to discussions with Tom Sterkenburg.

⁴¹Arguably, this theme can be found already in Leibniz—see Item 6 in [68].

approach, there are others that succeed in situations in which the given one fails; indeed, for any approach, there is another that succeeds in a disjoint set of situations. To adopt an approach to learning is to make a bet about what the world is like.

No-free-lunch results place upper bounds on our reasonable ambitions.⁴² Suppose that one is interested in the question: Why should someone interested in arriving at the truth proceed inductively (expecting the future to be like the past) rather than counter-inductively? Consider how this question looks in the simplest of our contexts, in which an agent being shown a binary sequence bit by bit aims to eventually be able to correctly predict each new bit on the basis of the bits seen so far. Here each method of learning succeeds on a countable dense subspace of the space of binary sequences. And all such subspaces are isomorphic (Sierpiński’s Theorem again). So unless we impose more structure on our problem, we have parity between the set of possibilities in which an inductive extrapolator m is successful and the set of possibilities in which counter-inductive extrapolator m^\dagger is successful.⁴³

The results developed above presuppose that we are operating in an austere setting—one in which we countenance arbitrary (computable) data streams or data streams generated by sampling from arbitrary (computable) probability measures. In more tightly constrained settings, learning becomes tractable—e.g., if one knows that the data stream is generated by a Bernoulli measure, then it is a straightforward task to use the data to successfully estimate the relevant parameter. But this observation illustrates rather than undercuts the perspective of the preceding paragraphs, making the point that although a universal learning algorithm is an impossibility, learning becomes possible when sufficiently strong presuppositions are in play. Of course, one would ultimately like to know more about where the boundaries lie of the class of learning problems in which failure is typical and of the class of learning problems in which success is typically achievable.⁴⁴

The results developed above are absolute in the sense that they do not presuppose the choice of a privileged measure on Cantor space or on the space of probability measures on Cantor space. But most of them *do* depend on the choice of topology. For the results concerning (weak) learning of sequences by extrapolators, this is not very worrying. In the vast majority of applications in statistics, economics, and computer science, the space of binary sequences is equipped with the product topology. And with good reason: this topology can be thought of as the topology of point-wise convergence and motivated by thinking of binary sequences as encoding real numbers in the usual way. The situation is not quite as straightforward with the space of probability measures on Cantor space. Certainly, the weak topology is extremely natural—but it is only one of several natural options. So it is natural to wonder whether the intractability of our learning problems would hold under other reasonable choices of topology.

⁴²That is one of their uses. Of the results canvassed in Section 1 above, Putnam’s was designed to expose a serious flaw in objective Bayesian approaches in the tradition of [69] while that of Shalev–Shwartz and Ben–David was devised to provide an elegant motivation for the definition of VC-dimension.

⁴³Some formal learning theorists take the view that learning strategies prone to mind-changes are to be eschewed and establish, in some contexts, a link between counter-inductive behaviour and mind changes—see, e.g., [70] and [71].

⁴⁴See [12] for some results of this kind for the problem of identification of sequences.

REFERENCES

- [1] D. Wolpert, W. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation* 1 (1997) 67–82.
- [2] D. Wolpert, The supervised learning no-free-lunch theorems, in: R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, F. Hoffmann (Eds.), *Soft Computing and Industry*, Springer, 2002, pp. 25–42.
- [3] Y.-C. Ho, D. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, *Journal of Optimization Theory and Applications* 115 (2002) 549–570.
- [4] U. von Luxburg, B. Schölkopf, Statistical learning theory: Models, concepts, and results, in: D. Gabbay, S. Hartmann, J. Woods (Eds.), *Inductive Logic*, Elsevier, 2011, pp. 651–706.
- [5] T. Sterkenburg, The no-free-lunch theorems of supervised learning, Unpublished (2020).
- [6] B. Rao, D. Gordon, W. Spears, For every generalization action, is there really an equal and opposite reaction? Analysis of the conservation law for generalization performance, in: A. Prieditis, S. Russell (Eds.), *ICML '95: Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995, pp. 471–479.
- [7] S. Shalev-Shwartz, S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.
- [8] H. Putnam, *Probability and Confirmation*, US Information Agency, 1963.
- [9] J. Case, C. Smith, Comparison of identification criteria for machine inductive inference, *Theoretical Computer Science* 25 (1983) 193–220.
- [10] E. M. Gold, Language identification in the limit, *Information and Control* 10 (1967) 447–474.
- [11] M. Li, P. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, 4th Edition, Springer, 2019.
- [12] L. Fortnow, R. Freivalds, W. Gasarch, M. Kummer, S. Kurtz, C. Smith, F. Stephan, On the relative sizes of learnable sets, *Theoretical Computer Science* 197 (1998) 139–156.
- [13] H. Putnam, ‘Degree of confirmation’ and inductive logic, in: P. Schilpp (Ed.), *The Philosophy of Rudolf Carnap*, Open Court, 1963, pp. 761–783.
- [14] J. Bārzdiņš, Prognostication of automata and functions, in: C. Frieman (Ed.), *Information Processing '71*, Vol. 1, North-Holland, 1972, pp. 81–84.
- [15] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, *Information and Control* 28 (1975) 125–155.
- [16] K. Podnieks, Comparing various concepts of function prediction. I, *Scientific Proceedings of Latvia State University* 210 (1974) 68–81 (in Russian).
- [17] K. Mehlhorn, On the size of sets of computable functions, in: R. Book, A. Borodin, F. Lewis, A. Mukhopadhyay, A. Rosenberg, R. Strong, J. Ullman (Eds.), *14th Annual Symposium on Switching and Automata Theory*, IEEE Computer Society Publications Office, 1973, pp. 190–196.
- [18] J. Bārzdiņš, R. Freivalds, On the prediction of general recursive functions, *Soviet Mathematics Doklady* 13 (1972) 1224–1228.
- [19] R. Solomonoff, A formal theory of inductive inference. I, *Information and Control* 7 (1964) 1–22.
- [20] D. Blackwell, L. Dubins, Merging of opinions with increasing information, *The Annals of Mathematical Statistics* 33 (1962) 882–886.

- [21] E. Kalai, E. Lehrer, Weak and strong merging of opinions, *Journal of Mathematical Economics* 23 (1994) 73–86.
- [22] E. Lehrer, R. Smorodinsky, Merging and learning, in: T. Ferguson, L. Shapley, J. MacQueen (Eds.), *Statistics, Probability and Game Theory: Papers in honor of David Blackwell.*, Institute of Mathematical Statistics, 1996, pp. 147–168.
- [23] D. Ryabko, M. Hutter, On sequence prediction for arbitrary measures, in: 2007 IEEE International Symposium on Information Theory, IEEE, 2007, pp. 2346–2350.
- [24] K. Parthasarathy, *Probability Measures on Metric Spaces*, Academic Press, 1967.
- [25] P. Billingsley, *Convergence of Probability Measures*, 2nd Edition, Wiley, 1999.
- [26] A. Kechris, *Classical Descriptive Set Theory*, Springer, 1995.
- [27] J. Reimann, Effectively closed sets of measures and randomness, *Annals of Pure and Applied Logic* 156 (2008) 170–182.
- [28] A. Nies, *Computability and Randomness*, Oxford University Press, 2009.
- [29] J. Oxtoby, The Banach-Mazur game and Banach category theorem, in: M. Dresher, A. Tucker, P. Wolfe (Eds.), *Contributions to the Theory of Games*, Vol. III, Princeton University Press, 1957, pp. 159–163.
- [30] R. Klette, R. Wiehagen, Research in the theory of inductive inference by GDR mathematicians—a survey, *Information Sciences* 22 (1980) 149–169.
- [31] D. Angluin, C. Smith, Inductive inference: Theory and methods, *ACM Computing Surveys* 15 (1983) 237–269.
- [32] P. Odifreddi, *Classical Recursion Theory*, Vol. II, Elsevier, 1999.
- [33] T. Zeugmann, S. Zilles, Learning recursive functions: A survey, *Theoretical Computer Science* 397 (2008) 4–56.
- [34] K. Podnieks, Comparing various concepts of function prediction. II, *Scientific Proceedings of Latvia State University* 233 (1975) 35–44 (in Russian).
- [35] Z. Szabó, Stratified inductive hypothesis generation, in: K. Jantke (Ed.), *Analogical and Inductive Inference*, Springer, 1987, pp. 177–195.
- [36] S. Jain, D. Osherson, J. Royer, A. Sharma, *Systems that Learn: An Introduction to Learning Theory*, 2nd Edition, MIT press, 1999.
- [37] C. Jockusch, P. Schupp, Generic computability, Turing degrees, and asymptotic density, *Journal of the London Mathematical Society* 85 (2012) 472–490.
- [38] H. Zenil (Ed.), *A Computable Universe: Understanding and Exploring Nature as Computation*, World Scientific, 2013.
- [39] L. Pitt, Probabilistic inductive inference, *Journal of the ACM* 36 (1989) 383–433.
- [40] J. Royer, Inductive inference of approximations, *Information and Control* 70 (1986) 156–178.
- [41] S. Jain, Program synthesis in the presence of infinite number of inaccuracies, *Journal of Computer and System Sciences* 53 (1996) 583–591.
- [42] A. Dasgupta, *Set Theory: With an Introduction to Real Point Sets*, Springer, 2014.
- [43] L. Lisagor, The Banach-Mazur game, *Mathematics of the USSR-Sbornik* 38 (1981) 201–216.
- [44] L. Báez-Duarte, $C(X)^*$ and Kolmogorov’s consistency theorem for Cantor spaces, *Studies in Applied Mathematics* 49 (4) (1970) 401–403.
- [45] R. Solomonoff, Complexity-based induction systems: Comparisons and convergence theorems, *IEEE Transactions on Information Theory* 24 (1978) 422–432.

- [46] M. Hutter, On universal prediction and Bayesian confirmation, *Theoretical Computer Science* 384 (2007) 33–48.
- [47] D. Freedman, On the asymptotic behavior of Bayes’ estimates in the discrete case, *The Annals of Mathematical Statistics* 34 (1963) 1386–1403.
- [48] Y. Noguchi, Merging with a set of probability measures: A characterization, *Theoretical Economics* 10 (2015) 411–444.
- [49] E. Szpilrajn, Remarques sur les fonctions complètement additives d’ensemble et sur les ensembles jouissant de la propriété de Baire, *Fundamenta Mathematicae* 22 (1934) 303–311.
- [50] E. Marczewski, R. Sikorski, Remarks on measure and category, *Colloquium Mathematicum* 2 (1949) 13–19.
- [51] O. Zindulka, Killing residual measures, *Journal of Applied Analysis* 5 (1999) 223–238.
- [52] E. Dekel, Y. Feinberg, Non-Bayesian testing of a stochastic prediction, *The Review of Economic Studies* 73 (2006) 893–906.
- [53] G. Koumoullis, Baire category in spaces of measures, *Advances in Mathematics* 124 (1996) 1–24.
- [54] L. Dubins, D. Freedman, Measurable sets of measures, *Pacific Journal of Mathematics* 14 (1964) 1211–1222.
- [55] J. Oxtoby, *Measure and Category: A Survey of the Analogies between Topological and Measure Spaces*, 2nd Edition, Springer, 1980.
- [56] A. Sandroni, R. Smorodinsky, The speed of rational learning, *International Journal of Game Theory* 28 (1999) 199–210.
- [57] P. Vitányi, N. Chater, Identification of probabilities, *Journal of Mathematical Psychology* 76 (2017) 13–24.
- [58] L. Bienvenu, S. Figueira, B. Monin, A. Shen, Algorithmic identification of probabilities is hard, *Journal of Computer and System Sciences* 95 (2018) 98–108.
- [59] G. Barmpalias, N. Fang, F. Stephan, Equivalences between learning of data and probability distributions, and their applications, *Information and Computation* 262 (2018) 123–140.
- [60] P. Neumann, Automorphisms of the rational world, *Journal of the London Mathematical Society* 2 (1985) 439–448.
- [61] L. Levin, V. V’yugin, Invariant properties of informational bulks, in: J. Gruska (Ed.), *Mathematical Foundations of Computer Science 1977*, Springer, 1977, pp. 359–364.
- [62] J. Leike, M. Hutter, On the computability of Solomonoff induction and knowledge-seeking, in: K. Chaudhuri, C. Gentile, S. Zilles (Eds.), *International Conference on Algorithmic Learning Theory*, Springer, 2015, pp. 364–378.
- [63] T. Sterkenburg, Putnam’s diagonal argument and the impossibility of a universal learning machine, *Erkenntnis* 84 (2019) 633–656.
- [64] H. Jeffreys, Probability, statistics, and the theory of errors, *Proceedings of the Royal Society of London. Series A* 140 (1933) 523–535.
- [65] T. S. Kuhn, *The Structure of Scientific Revolutions*, University of Chicago Press, 1963.
- [66] N. Chomsky, *Aspects of the Theory of Syntax*, MIT Press, 1965.
- [67] C. G. Hempel, *Philosophy of Natural Science*, Prentice–Hall, 1966.
- [68] L. Loemker (Ed.), *Gottfried Wilhelm Leibniz: Philosophical Papers and Letters*, 2nd Edition, D. Reidel, 1969.
- [69] R. Carnap, On inductive logic, *Philosophy of Science* 12 (1945) 72–97.

- [70] K. Kelly, Simplicity, truth, and probability, in: P. Bandyopadhyay, M. Forster (Eds.), *Philosophy of Statistics*, Elsevier, 2011, pp. 983–1024.
- [71] H. Lin, Modes of convergence to the truth: Steps toward a better epistemology of induction, Forthcoming in *The Review of Symbolic Logic*. (2018).

DEPARTMENT OF PHILOSOPHY, UNIVERSITY OF MICHIGAN

BELOT@UMICH.EDU