Proceedings of Machine Learning Research vol 195:1–14, 2023 36th Annual Conference on Learning Theory

Find a witness or shatter: the landscape of computable PAC learning.

Valentino Delle Rose VALENTINO.DELLEROSE@CENIA.CL
Institute for Mathematical and Computational Engineering, PUC, Chile & CENIA, Chile.

Alexander Kozachinskiy AKOZACHINSKYI@CENIA.CL
CENIA, Chile & IMFD, Chile.

Cristóbal Rojas CRISTOBAL.ROJAS@MAT.UC.CL
Institute for Mathematical and Computational Engineering, PUC, Chile & CENIA, Chile.

Tomasz Steifer TSTEIFER@IPPT.PAN.PL
Institute of Fundamental Technological Research, Polish Academy of Sciences & Institute for Mathematical and Computational Engineering, PUC, Chile & IMFD, Chile

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

This paper contributes to the study of CPAC learnability—a computable version of PAC learning—by solving three open questions from recent papers. Firstly, we prove that every improperly CPAC learnable class is contained in a class which is properly CPAC learnable with polynomial sample complexity. This confirms a conjecture by Agarwal et al (COLT 2021). Secondly, we show that there exists a decidable class of hypotheses which is properly CPAC learnable, but only with uncomputably fast-growing sample complexity. This solves a question from Sterkenburg (COLT 2022). Finally, we construct a decidable class of finite Littlestone dimension which is not improperly CPAC learnable, strengthening a recent result of Sterkenburg (2022) and answering a question posed by Hasrati and Ben-David (ALT 2023). Together with previous work, our results provide a complete landscape for the learnability problem in the CPAC setting.

Keywords: PAC learnability, CPAC learnability, VC dimension, Littlestone dimension, computability, foundations of machine learning

1. Introduction

The fundamental problem in the theory of Machine Learning is to understand when a given hypothesis class can be learned by an algorithm that has access to finitely many random samples of an unknown objective function. The goal of the learner is to select a function that approximates the objective function at least as well as any hypothesis from the given class. The fundamental theorem of statistical learning provides a characterization of the existence of learners for a given hypothesis class in terms of the finitude of a combinatorial quantity (VC dimension) associated with the class (Vapnik and Chervonenkis, 1971; Blumer et al., 1989). This characterization is concerned with the existence of learners as abstract mathematical functions and does not take into account their computational properties.

Recently, a new framework combining PAC learning and Computability Theory was proposed by Agarwal et al. (2020). In computable PAC (CPAC) learning, both the learner and the functions it outputs are required to be *computable*, in the sense that they can be computed by a Turing Machine. As observed in Agarwal et al. (2020), the existence of a computable learner no longer follows from finite VC dimension. Moreover, the computable setting is sensible to aspects of the problem that make no difference in the classical setting. For example, it becomes important whether the learner is required to be *proper* (i.e., constrained to only output functions that belong to the hypothesis class) or allowed to be *improper* (can output arbitrary functions). Another issue is whether the *sample complexity*, i.e., the number of samples a learner needs in order to work as requested, can always be bounded by a computable function (a setting referred to as *strong* CPAC learning). This raises a number of natural questions —which of these aspects of the problem actually lead to different versions of computable learnability?

Significant progress was made by Sterkenburg (2022), who gave a characterization of proper strong CPAC learning in terms of the computability of a *Empirical Risk Minimizer* (ERM) and who constructed a class of finite VC dimension which is not CPAC learnable, even in the improper sense. Independently, the framework of computable PAC learning was lifted to continuous spaces and studied in terms of Weihrauch reducibility Ackerman et al. (2022). Very recently, Hasrati and Ben-David (2023) gave the computability-theoretic perspective on a related framework of *online learning*, further improving our understanding of the learning problem in the computable setting.

Main results. The current paper contributes to this line of research, in particular, by solving three open problems raised in these recent papers. First, we provide a characterization of CPAC learning in the *improper* setting, i.e., when the learner is allowed to output a function outside of the given hypothesis class. For that, we introduce the *effective VC dimension*. The classical VC dimension of a hypothesis class $\mathcal{H}$ can be defined as the minimal k such that for any tuple of $k+1$ distinct natural numbers one can indicate a Boolean function on them which is not realizable by hypotheses of $\mathcal{H}$. In the effective version of VC dimension, there also must be an *algorithm*, transforming a tuple of $k+1$ distinct natural numbers into a Boolean function, not realizable on them by $\mathcal{H}$. Our first result states that a hypothesis class is improperly CPAC learnable if and only if its effective VC dimension is finite. As a byproduct, we obtain that every improperly CPAC learnable class is in fact a subclass of a properly CPAC learnable class, settling a conjecture formulated by Agarwal et al. (2021). Secondly, we show that there exists a decidable class of hypotheses $\mathcal{H}$ that has proper computable learners, but only those whose sample complexity cannot be bounded from above by a computable function. This separates CPAC learning from strong CPAC learning in the *proper* setting, solving a question asked by Sterkenburg (2022). Finally, we strengthen a theorem of Sterkenburg (2022), who constructed a decidable class of finite VC dimension which is not improperly CPAC learnable. We show that such a class can be constructed to even have a finite Littlestone dimension, providing an answer to a question posed by Hasrati and Ben-David (2023) in the context of online learning. Altogether, our results provide a comprehensive landscape for the learnability problem in the computable setting.

Organization of the paper. In Section 2, we briefly recall the classical PAC learning framework. In Section 3, we provide a detailed overview of computable PAC learning, go through results and open problems from previous works, and then present precise statements of our results. Proofs are given in the subsequent sections.

2. Preliminaries

2.1. Notation

For any two sets A and B, we denote by B^A the set of functions $f\colon A \to B$. If $f\colon A \to \{0,1\}$, by the support of f, denoted by $\mathsf{supp}(f)$, we mean $f^{-1}(1)$.

2.2. Classical PAC learning

In this section, we briefly introduce the classical PAC learning framework. We only work over the domain $\mathbb{N}$. Thus, a *hypothesis class* $\mathcal{H}$ is an arbitrary subset of $\{0,1\}^{\mathbb{N}}$. Elements of $\mathcal{H}$ will be called *hypotheses*. We will say that a class is *finitely supported* if it consists of only hypotheses with finite support. A *sample* of size n is an element of $(\mathbb{N} \times \{0,1\})^n$. A *learner* is any function A from the set of all samples (that is, from $\bigcup_{n\in\mathbb{N}}(\mathbb{N} \times \{0,1\})^n$) to $\{0,1\}^{\mathbb{N}}$. Now, if D is a probability distribution over $\mathbb{N} \times \{0,1\}$, then the *generalization error* of $h \in \{0,1\}^{\mathbb{N}}$ with respect to D is

$$L_D(h) = \Pr_{(x,y)\sim D}[h(x) \neq y].$$

When we write $S \sim D^m$, we mean that the coordinates of the sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ were drawn independently m times from D. In the PAC learning framework, the learner receives a sample $S \sim D^m$ for some sufficiently large m. The learner's task is to select, with high probability over D^m, some function $f\colon \mathbb{N} \to \{0,1\}$ whose generalization error with respect to D is close to the best possible generalization error achievable by functions in some known, a priori given, hypothesis class $\mathcal{H}$.

Definition 1 *Let $\mathcal{H}$ be a hypothesis class and A be a learner. We say that A **PAC learns** $\mathcal{H}$ if for every $n \in \mathbb{N}$ there exists m_n such that for every $m \geq m_n$ and for every probability distribution D over $\mathbb{N} \times \{0,1\}$ we have:*

$$\Pr_{S\sim D^m}[L_D(A(S)) \leq \inf_{h\in\mathcal{H}} L_D(h) + 1/n] \geq 1 - 1/n. \tag{1}$$

*If there exists A which PAC learns $\mathcal{H}$, then $\mathcal{H}$ is called **PAC learnable**.*

*If A PAC learns $\mathcal{H}$, then the **sample complexity** of A w.r.t. $\mathcal{H}$ is a function $m\colon \mathbb{N} \to \mathbb{N}$, where $m(n)$ is the minimal natural number m_n such that, for all $m \geq m_n$, (1) holds for n.*

*We say that A is **proper** for $\mathcal{H}$ if it only outputs hypotheses from $\mathcal{H}$. Otherwise, we say that A is **improper** for $\mathcal{H}$.*

A classical result of learning theory is that PAC learnability admits a combinatorial characterization via the parameter called the Vapnik-Chervonenkis (VC) dimension. More specifically, let $x_1, \ldots, x_k$ be k distinct natural numbers. We say that a hypothesis class $\mathcal{H}$ *shatters* $x_1, \ldots, x_k$ if for all 2^k functions $f\colon \{x_1, \ldots, x_k\} \to \{0,1\}$ there exists $h \in \mathcal{H}$ with $f(x_1) = h(x_1), \ldots, f(x_k) = h(x_k)$. The *VC dimension* of $\mathcal{H}$ is the maximal natural k for which there exist k distinct natural numbers that are shattered by $\mathcal{H}$. If such k distinct natural numbers exist for all k, the VC dimension of $\mathcal{H}$ is $+\infty$. It turns out that a class $\mathcal{H}$ is PAC learnable if and only if its VC dimension is finite.

Another classical result is that any PAC learnable class $\mathcal{H}$ can be learned by a specific type of learners called *Empirical Risk Minimizers* (or ERM for short). To define them, we first have to define the *error of a hypothesis on a sample*. Namely, let $S = ((x_1, y_1), \ldots, (x_m, y_m))$ be a sample and $h\colon \mathbb{N} \to \{0,1\}$. The error of h on S is defined as:

$$L_S(h) = \frac{|\{i \in \{1,\ldots,m\} \mid h(x_i) \neq y_i\}|}{m}.$$

A learner A is an *ERM* for a class $\mathcal{H}$ if for every sample S we have $A(S) \in \arg\min_{h\in\mathcal{H}} L_S(h)$. In other words, for a given sample S an ERM outputs a hypothesis from $\mathcal{H}$ with the least error on S. Note that ERM might be not unique (there might be more than one hypothesis attaining the minimum of $L_S(h)$). Finally, there is another relevant property of a class $\mathcal{H}$ that we will also use. We say that a class of hypothesis $\mathcal{H}$ has the *uniform convergence property* if for every n there exists m_n such that for every $m \geq m_n$ and for every probability distribution D,

$$\Pr_{S\sim D^m}\left[\forall h \in H,\ |L_D(h) - L_S(h)| \leq \frac{1}{n}\right] \geq 1 - 1/n. \tag{2}$$

The fundamental theorem of statistical learning links together the VC dimension, learnability, Empirical Risk Minimization, and the uniform convergence property.

Theorem 2 (Vapnik and Chervonenkis (1971), Blumer et al. (1989)) [1]
For any class of hypotheses $\mathcal{H}$, the following conditions are equivalent:

- *$\mathcal{H}$ is PAC learnable,*
- *the VC dimension of $\mathcal{H}$ is finite;*
- *every ERM for $\mathcal{H}$ PAC learns $\mathcal{H}$, and its sample complexity w.r.t. $\mathcal{H}$ is bounded by $O(n^2 \log n)$, where the constant hidden in $O(\cdot)$ depends only on $\mathcal{H}$;*
- *$\mathcal{H}$ has the uniform convergence property.*

Similarly to the VC dimension for PAC learnability, there is a combinatorial property that characterizes *online learnability* (see Littlestone (1988) and Ben-David and Pál (2009)) and private PAC learning (Alon et al., 2022). Consider a full rooted binary tree T of depth d where each non-leaf node is labeled by some $x \in \mathbb{N}$. We say that T is *shattered* by a hypothesis class $\mathcal{H}$ if for every leaf l of T, there exists a hypothesis $h \in \mathcal{H}$ which *leads to l in T*. I.e., l can be obtained by descending from the root of T in the following manner: if we are in some non-leaf node v (in the beginning, v is the root), and if $x \in \mathbb{N}$ is the label of v in T, then we go to the left child of v if $h(x) = 0$ and we go to the right child of v if $h(x) = 1$, and this continues until we reach a leaf. The *Littlestone dimension of $\mathcal{H}$* (denoted by $\mathrm{Ldim}(\mathcal{H})$) is defined as the maximal depth of a tree which is shattered by $\mathcal{H}$. If $\mathcal{H}$ shatters trees of arbitrary depth, then we set $\mathrm{Ldim}(\mathcal{H}) = \infty$. It is not hard to see that for every hypotheses class $\mathcal{H}$, the VC dimension of $\mathcal{H}$ is at most $\mathrm{Ldim}(\mathrm{H})$ (see e.g. Shalev-Shwartz and Ben-David (2014), Theorem 21.9).

3. Computable PAC learning: previous results and our contribution

In the computable version of PAC learning (introduced by Agarwal et al. (2020)) one considers only *computable* learners, i.e. learners that can be implemented by an algorithm. In more detail, we say that a learner A is computable if there exists an algorithm which, given a sample S, outputs a description of a computer program (formally, a Turing machine), implementing $h = A(S)$ (the function that A outputs on S). In particular, with this program, we can evaluate h on any natural number. In this paper, we mainly deal with learners that only output finitely supported hypotheses. To show that such a learner A is computable, it is enough to exhibit an algorithm that transforms

1. The definition of VC dimension and its connection with the uniform convergence property were established in Vapnik and Chervonenkis (1971), while the relationship with PAC learnability is due to Blumer et al. (1989). See e.g. Shalev-Shwartz and Ben-David (2014), Theorem 6.7 for an explanatory presentation.

S into $\mathsf{supp}(A(S))$ – when $\mathsf{supp}(A(S))$ is finite, one can easily construct a Turing machine that outputs 1 exactly on $\mathsf{supp}(A(S))$. Requiring a learner to be computable naturally leads to the following definition:

Definition 3 *A hypothesis class $\mathcal{H}$ is **computably PAC learnable** (or **CPAC learnable** for short) if there exists a computable learner*[2] *that PAC learns it.*

Remark 4 *Previous authors make an explicit assumption for CPAC learnability that a hypothesis class must contain only computable functions. In this paper, we drop this assumption because it is never used in the proofs and we consider it unnecessarily restrictive. Nevertheless, all our results hold with this assumption as well. Moreover, all examples of classes in this paper are computationally simple. Namely, they consist only of functions with finite support. In fact, all these classes are decidable, meaning that there is an algorithm that, when provided the support of a given finitely supported function h, decides whether h belongs to the class.*

The question of whether or not improper CPAC learnability is different from PAC learnability is not obvious and was left as an open problem in Agarwal et al. (2020). It was recently solved by Sterkenburg (2022), who gave an example of a hypothesis class with VC-dimension 1 which is not improperly CPAC learnable. In fact, his class consists of hypotheses with finite support and is decidable.

Agarwal et al. also introduced a more restrictive version of CPAC learnability by constraining learners to only output functions from the given hypothesis class $\mathcal{H}$.

Definition 5 *A hypothesis class $\mathcal{H}$ is **properly CPAC learnable** if there exists a computable learner that PAC learns $\mathcal{H}$ and that is proper for $\mathcal{H}$.*

Remark 6 *The reader should be warned that the choice of names for these definitions is not consistent across the literature. Here we have followed Agarwal et al. (2020). But for example, Sterkenburg (2022) calls CPAC learnability what we have called here proper CPAC learnability.*

In the classical setting, the requirement of being proper does not change anything. Indeed, by Theorem 2, whenever some learner PAC learns $\mathcal{H}$, we have that any ERM for $\mathcal{H}$ PAC learns it as well, and any ERM is proper by definition. However, in the computable setting, the existence of a computable learner does not necessarily guarantee the computability of some ERM, or of any other proper learner. In fact, as shown by Agarwal et al. (2020), the set of properly CPAC learnable classes is indeed strictly contained in the set of CPAC learnable ones. Their example consists of a certain decidable class of hypotheses $\mathcal{H}$ made of functions with the support of size 2 (and thus PAC learnable since its VC dimension is at most 2) which does not admit any proper computable learner, but for which an (improper) computable learner can be easily constructed. Even better, in this case, the class $\mathcal{H}$ is included in the decidable class $\mathcal{H}_2$ of all functions with size support of size 2, which clearly admits a computable ERM and is therefore properly CPAC learnable. A question then naturally arises:

is every improperly CPAC learnable class $\mathcal{H}$ contained in some properly CPAC learnable class $\widehat{\mathcal{H}}$?

2. We stress that in this definition the learner might be *improper*, in the sense that it is allowed to output functions that do not necessarily belong to $\mathcal{H}$.

Observe that indeed all subclasses of a properly CPAC learnable $\widehat{\mathcal{H}}$ are (improperly) CPAC learnable (via the same computable learner that PAC learns $\widehat{\mathcal{H}}$). However, it is not clear if this is the only way a class $\mathcal{H}$ can be CPAC learnable. This question was raised by both Sterkenburg (2022) and Agarwal et al. (2021). In this paper, we shall answer it in the affirmative (see Theorem 11 below).

A similar analysis can be carried out with a focus on a different aspect of the problem –sample complexity. As we have just mentioned, in the classical setting, a class $\mathcal{H}$ that is PAC learnable, is automatically learnable by any ERM, and the sample complexity of ERMs is only polynomial in n. Once again, in the computable setting, ERMs are not necessarily computable. Therefore, for a given computable learner, even the computability of its sample complexity function is not guaranteed. Motivated by this issue, Sterkenburg introduced the following *strong* variant of CPAC learnability.

Definition 7 *A hypothesis class $\mathcal{H}$ is* ***strongly CPAC learnable*** *(or SCPAC learnable for short) if there exists a computable learner that PAC learns $\mathcal{H}$ and whose sample complexity is bounded from above by some total computable function. Moreover, if there exists a learner with these properties which is proper for $\mathcal{H}$, then $\mathcal{H}$ is* ***properly*** *SCPAC learnable.*

Sterkenburg obtained a characterization of *proper* SCPAC learnability.

Theorem 8 (Sterkenburg (2022), Theorem 2) *A class of hypotheses $\mathcal{H}$ is properly SCPAC learnable if and only if it has a computable ERM and its VC dimension is finite.*

One interesting consequence of this result is that whenever $\mathcal{H}$ has a proper computable learner whose sample complexity is bounded by some total computable function, no matter how fast it grows, $\mathcal{H}$ automatically has some other computable learner whose sample complexity is just polynomial (because by Theorem 2, every ERM has polynomial sample complexity). As a main problem, Sterkenburg left open the following natural question:

is proper CPAC learnability equivalent to proper SCPAC learnability?

In this paper, we solve this problem by showing that these two notions can be separated by a decidable class of hypotheses, see Theorem 14 below.

3.1. Statements of our results.

We start with a characterization of CPAC learnability via *effective VC-dimension*.

Definition 9 *Let $\mathcal{H} \subseteq \{0,1\}^{\mathbb{N}}$ be a hypothesis class.* ***A k-witness of VC dimension for*** *$\mathcal{H}$ is any function w, whose domain is the set of all increasing $(k+1)$-tuples of natural numbers and whose range is $\{0,1\}^{k+1}$, such that for every $x_1 < x_2 < \ldots < x_{k+1} \in \mathbb{N}$ and for every $h \in \mathcal{H}$ we have:*

$$\big(h(x_1), \ldots, h(x_{k+1})\big) \neq w((x_1, \ldots, x_{k+1})).$$

The ***effective VC-dimension*** *of $\mathcal{H}$ is the minimal $k \in \mathbb{N}$ such that there exists a* ***computable*** *k-witness of VC-dimension for $\mathcal{H}$. If no such k exists, then the effective VC-dimension of $\mathcal{H}$ is infinite.*

In other words, for every $x_1 < x_2 < \ldots < x_{k+1}$, the witness function w provides a Boolean function on $x_1, \ldots, x_{k+1}$ which is not realizable by hypotheses from $\mathcal{H}$. The minimal k for which such w exists (possibly, not computable) is equal to the ordinary VC dimension of $\mathcal{H}$. Now, if we consider only computable witnesses, we obtain the effective VC dimension. Thus, the effective VC dimension can only be larger than the ordinary one.

Sterkenburg proved that having a finite effective VC dimension is a necessary condition for CPAC learnability (and used this to give an example of a class that is PAC learnable but not CPAC learnable):

Proposition 10 (Sterkenburg (2022), Lemma 1) *If $\mathcal{H}$ is CPAC learnable, then it admits a computable k-witness of VC dimension, for some natural number k.*

We show that these two conditions are, in fact, equivalent. Moreover, they are equivalent to being contained in some properly SCPAC learnable class. Thus, our next theorem settles a question of Sterkenburg (2022); Agarwal et al. (2021).

Theorem 11 *For every $\mathcal{H}$, the following 3 conditions are equivalent:*

- *(a) Effective VC dimension of $\mathcal{H}$ is finite;*
- *(b) $\mathcal{H}$ is CPAC learnable;*
- *(c) $\mathcal{H}$ is contained in some properly SCPAC learnable hypothesis class $\widehat{\mathcal{H}}$.*

Proof Section 4. ∎

We note that the previous result also shows that CPAC learnability is equivalent to SCPAC learnability, as well as to being contained in some properly CPAC learnable class (as the latter two properties are stronger than condition (b) but weaker than condition (c) of Theorem 11).

Next, we observe that one can give a characterization of proper CPAC learnability which is similar in spirit to Proposition 8, Sterkenburg's characterization of proper SCPAC learnability. For that, we need the following relaxed version of ERMs.

Definition 12 *Let $\mathcal{H} \subseteq \{0,1\}^{\mathbb{N}}$ be an hypothesis class. A learner A is called an **asymptotic ERM** for $\mathcal{H}$ if it outputs only hypotheses from $\mathcal{H}$ and if there exists an infinite sequence $\{\varepsilon_m \in [0,1]\}_{m=1}^{\infty}$, converging to 0 as $m \to \infty$, such that for every sample S we have that:*

$$L_S(A(S)) \leq \inf_{H} L_S(h) + \varepsilon_{|S|}.$$

Just like the existence of a computable ERM characterizes proper SCPAC learnability, proper CPAC learnability boils down to the existence of a computable asymptotic ERM.

Proposition 13 *A hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathbb{N}}$ is properly CPAC learnable if and only if its VC dimension is finite and it has a computable asymptotic ERM.*

Proof Section 5. ∎

From the proof of Proposition 13 it is not hard to see that the error ε_m is to an asymptotic ERM for $\mathcal{H}$ exactly as the sample complexity function $m(n)$ is to a corresponding proper learner.

It follows that ε_m can be bounded above by a computable function that decreases to 0, exactly when $\mathcal{H}$ is properly SCPAC learnable. We use this observation to answer another open question of Sterkenburg (2022). Namely, that not every properly CPAC learnable class is properly SCPAC learnable.

Theorem 14 *There is a decidable class of finitely supported hypotheses $\mathcal{H}$ which is properly CPAC learnable but not properly SCPAC learnable.*

Proof Section 6. ∎

Our results, together with previous works, fully determine the landscape of computable PAC learning (see Figure 1). First, by Theorem 14, we have that the set of properly SCPAC learnable classes is strictly included in the set of properly CPAC learnable classes. Next, due to the example of Agarwal et al. (2020), the set of properly CPAC learnable classes is strictly included in the set of CPAC learnable classes. On the other hand, CPAC learnable classes coincide with SCPAC learnable classes, as well as with classes that are subsets of proper CPAC (or SCPAC) learnable superclasses (denoted by $\subseteq$prop. (S)CPAC in Figure 1), by Theorem 13. Finally, by the construction of Sterkenburg (2022), the set of CPAC learnable classes is strictly included in the set of PAC learnable classes. We were able to strengthen this separation by constructing such an example with not only a finite VC dimension but even a finite Littlestone dimension. More precisely, we prove the following result.

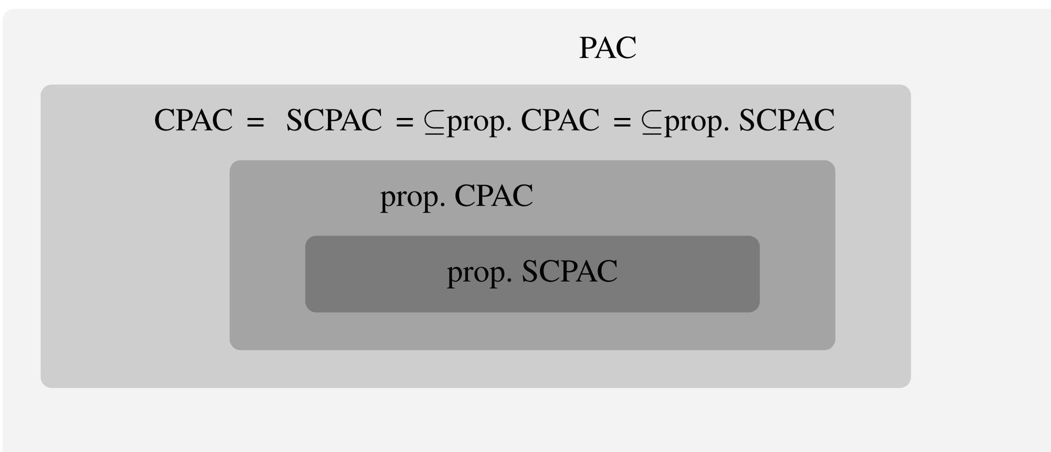

Figure 1: The landscape of computable PAC learning. Note that the strict inclusions hold even in the case of decidable classes of hypotheses.

Theorem 15 *There is a decidable class of finitely supported hypotheses $\mathcal{H}$ with $\mathrm{Ldim}(\mathcal{H}) = 1$ which is not (improperly) CPAC learnable.*

Proof Section 7. ∎

This theorem answers an open question from Hasrati and Ben-David (2023). It also establishes the separation between the classical online learnability (recently shown to be equivalent to a version of private PAC learning by Alon et al. (2022)) and its computable counterpart, even for decidable classes of hypotheses (see details in Hasrati and Ben-David (2023), section 6).

4. Proof of Theorem 11

Proof Implication $(b) \implies (a)$ has been already shown in Sterkenburg (2022) (Proposition 10 above), and $(c) \implies (b)$ follows directly from the definitions. It remains to establish that $(a) \implies (c)$.

Assume that the effective VC dimension of $\mathcal{H}$ is finite. Then for some $k \in \mathbb{N}$ there exists a computable k-witness w of VC dimension for $\mathcal{H}$. Let us say that a function $h\colon \mathbb{N} \to \{0,1\}$ is *good* if h has finite support and for every $x_1 < x_2 < \ldots < x_{k+1} < \max \mathsf{supp}(h)$ we have that h disagrees with w on $x_1 \ldots, x_{k+1}$, i.e., $\big(h(x_1), \ldots, h(x_{k+1})\big) \neq w((x_1, \ldots, x_{k+1}))$. Let $\mathcal{H}_{good}$ denote the set of all good $h\colon \mathbb{N} \to \{0,1\}$. Define $\widehat{\mathcal{H}} = \mathcal{H} \cup \mathcal{H}_{good}$. Obviously, $\mathcal{H} \subseteq \widehat{\mathcal{H}}$. It remains to show that $\widehat{\mathcal{H}}$ is SCPAC learnable. By Theorem 8, it is sufficient to show two things: that $\widehat{\mathcal{H}}$ has finite VC dimension and that $\widehat{\mathcal{H}}$ has computable ERM.

We first show that the VC dimension of $\widehat{\mathcal{H}}$ is at most $k+1$. Indeed, take any $x_1 < x_2 < \ldots < x_{k+2} \in \mathbb{N}$. We claim that

$$\big(h(x_1), \ldots, h(x_{k+1}), h(x_{k+2})\big) \neq \big(w((x_1, \ldots, x_{k+1})), 1\big) \text{ for all } h \in \widehat{\mathcal{H}}. \tag{3}$$

Indeed, if $h \in \mathcal{H}$, then $\big(h(x_1), \ldots, h(x_{k+1})\big) \neq w((x_1, \ldots, x_{k+1}))$ because w is a k-witness of VC dimension for $\mathcal{H}$, and hence (3) holds as well. Now, assume that $h \in \mathcal{H}_{good}$. If $h(x_{k+2}) = 0$, then (3) holds. In turn, if $h(x_{k+2}) = 1$, observe that $x_1 < x_2 < \ldots < x_{k+1} < x_{k+2} \leq \max \mathsf{supp}(h)$. Hence, by definition of a good hypothesis, we have $\big(h(x_1), \ldots, h(x_{k+1})\big) \neq w((x_1, \ldots, x_{k+1}))$, and this implies (3).

It remains to show that $\widehat{\mathcal{H}}$ has a computable ERM. The key is to note that for any given sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ with $M = \max\{x_1, \ldots, x_m\}$, in order to find a hypothesis with minimal error on S, it is enough to go through all good hypotheses whose support is a subset of $\{0, 1, \ldots, M\}$. There are finitely many of them, and we can effectively list them because w is computable. It remains to show that the resulting learner is an ERM for $\mathcal{H}$. For that, it is enough to establish that for any $h \in \widehat{\mathcal{H}}$ there exists a good h_1 with $\mathsf{supp}(h_1) \subseteq \{0, 1, \ldots, M\}$ such that $L_S(h) = L_S(h_1)$. Indeed, consider h_1 that coincides with h on $\{0, 1, \ldots, M\}$ and equals 0 otherwise (so that $\mathsf{supp}(h_1) \subseteq \{0, \ldots, M\}$). Clearly, $L_S(h) = L_S(h_1)$ because S only involves numbers up to M. It remains to show that h_1 is good, i.e., that it disagrees with w on every tuple $x_1 < x_2 < \ldots < x_{k+1}$ with $x_{k+1} < \max \mathsf{supp}(h_1)$. Indeed, note that $\max \mathsf{supp}(h_1) \leq M$. Hence, h agrees with h_1 on $x_1 < x_2 < \ldots < x_{k+1}$ and on $x_{k+2} = \max \mathsf{supp}(h_1)$ (which means that $h(x_{k+2}) = 1$). Thus, h_1 must disagree with w on $x_1 < x_2 < \ldots < x_{k+1}$ because otherwise we have $(h(x_1), \ldots, h(x_{k+1})) = w((x_1, \ldots, x_{k+1}))$ and $h(x_{k+2}) = 1$, which contradicts (3) for h. ∎

5. Proof of Proposition 13

Proof First, assume that the VC dimension of $\mathcal{H}$ is finite and that it admits a computable asymptotic ERM A. We show that A PAC learns $\mathcal{H}$. Since A is computable and only outputs functions from $\mathcal{H}$

by definition, this means that $\mathcal{H}$ is properly CPAC learnable. Since the VC dimension of $\mathcal{H}$ is finite, Theorem 2 ensures that it has the uniform convergence property. Let then m_n be such that for every $m \geq m_n$ and every probability distribution D, it holds that

$$\Pr_{S \sim D^m}\left[\forall h \in \mathcal{H},\ |L_D(h) - L_S(h)| \leq \frac{1}{3n}\right] \geq 1 - \frac{1}{n}.$$

Since A only outputs hypothesis from $\mathcal{H}$, we have that with probability at least $1 - \frac{1}{n}$, it holds that $|L_D(A(S)) - L_S(A(S))| \leq \frac{1}{3n}$, as well as $|L_D(h^*) - L_S(h^*)| \leq \frac{1}{3n}$ for any $h^* \in \arg\min_{h \in \mathcal{H}} L_S(h)$. Furthermore, by definition of asymptotic ERM, there exists a sample size m'_n such that for any sample of size at least m'_n we have

$$L_S(A(S)) \leq L_S(h^*) + \frac{1}{3n}.$$

It follows that for any sample S of size $m \geq \max(m_n, m'_n)$,

$$\Pr_{S \sim D^m}\left[L_D(A(S)) \leq \inf_{h \in \mathcal{H}} L_D(h) + \frac{1}{n}\right] \geq 1 - \frac{1}{n}$$

holds, which shows that A is a PAC learner for $\mathcal{H}$.

Now assume that $\mathcal{H}$ is CPAC learnable and let A be a computable learner. We construct a computable asymptotic ERM $\widehat{A}$. For any sample S, consider the probability distribution $D(S)$ which assigns probability $\frac{1}{|S|}$ to any element $(x, y) \in S$ (in case S has repetitions, we simply sum the corresponding probabilities). Note that with this definition for $D(S)$ we have that

$$L_S(h) = L_{D(S)}(h) \quad \text{for all } h \in \mathcal{H}. \tag{4}$$

Let us denote by $\widehat{\mathcal{S}}$ the set of all samples S' of size $m = |S|$ that can be drawn from $D(S)$. We then define the output of $\widehat{A}$ on S by

$$\widehat{A}(S) = \arg \min_{\{A(S')\,:\, S' \in \widehat{\mathcal{S}}\}} L_S(A(S')).$$

It is clear that $\widehat{A}$ is computable since A and $D(S)$ are. To show that $\widehat{A}$ is an asymptotic ERM, let $m(n)$ be the sample complexity of A w.r.t. $\mathcal{H}$. Then, for every n and every m such that $m(n) \leq m < m(n+1)$, let $\epsilon_m = \frac{1}{n}$. Since A is a PAC learner for $\mathcal{H}$, we have that for such an m and $S' \sim D(S)^m$, with probability at least $1 - 1/n$ it holds that

$$L_{D(S)}(A(S')) \leq \inf_{h \in \mathcal{H}} L_{D(S)}(h) + \epsilon_{|S|}.$$

This means that there exists some $S^* \in \widehat{\mathcal{S}}$ for which this holds, which together with equality (4) above gives $L_S(\widehat{A}(S)) \leq \inf_h L_S(h) + \epsilon_{|S|}$, as it was to be shown. ■

6. Proof of Theorem 14

Proof We start by defining the class $\mathcal{H}$. First, we partition all even numbers into blocks of increasing sizes: $I_1 = \{2\}$, $I_2 = \{4, 6\}$, $I_3 = \{8, 10, 12\}$ and so on so that the size of $I_k = \{n_{k1}, \ldots, n_{kj}, \ldots n_{kk}\}$ is k. Then, for every $k \geq 1$ and $1 \leq j \leq k$ we let

$$h_{kj}(n) = \begin{cases} 1 & \text{if } n \in I_k \setminus \{n_{kj}\}, \\ 0 & \text{otherwise,} \end{cases}$$

and put it into $\mathcal{H}$. Then we consider a total injective computable function $f \colon \mathbb{N} \to \mathbb{N}$ such that $f(\mathbb{N})$ is undecidable. For example, one can take any enumerable undecidable set $S \subseteq \mathbb{N}$ and let $f(a)$ be the ath natural number in a computable enumeration of S (without repetitions to ensure that f is injective). Now, for every $a \in \mathbb{N}$, we define a hypothesis h_a which is equal to 1 on $I_{f(a)} \cup \{2a+1\}$ and nowhere else. We put all hypotheses h_a into $\mathcal{H}$ as well. This finishes the description of $\mathcal{H}$, which consists only of functions with finite support.

We now show that $\mathcal{H}$ is decidable. Let $h \colon \mathbb{N} \to \mathbb{N}$ be any function with finite support. We first check whether the support of $\mathcal{H}$ intersects I_k for exactly one k (otherwise, $h \notin \mathcal{H}$). If it does, there are two possibilities for h to be in $\mathcal{H}$. The first possibility is when $\mathsf{supp}(h)$ is a subset of I_k of size $k-1$. In this case, $h = h_{kj}$ for some j and thus $h \in \mathcal{H}$. The only other possibility is when $\mathsf{supp}(h) = I_k \cup \{2a+1\}$ for some a. Then $h \in \mathcal{H}$ if and only if $k = f(a)$, and we check this by computing $f(a)$.

We now show that $\mathcal{H}$ is properly CPAC learnable. By Proposition 13, it is enough to show two things: that $\mathcal{H}$ has finite VC dimension and that $\mathcal{H}$ has a computable asymptotic ERM. We now show that the VC dimension of $\mathcal{H}$ is at most 3. For that, we take any 4 distinct natural numbers x_1, x_2, x_3, x_4 and show that not all 2^4 Boolean functions on $S = \{x_1, x_2, x_3, x_4\}$ can be realized by hypotheses from $\mathcal{H}$. First, observe that the support of every hypothesis from $\mathcal{H}$ intersects exactly one block I_k. Hence, if S intersects two different blocks, we cannot realize the all-ones function on S. Likewise, we cannot realize the all-ones function if S has two distinct odd numbers (every hypothesis from S has at most one odd number in its support). The only case left is when S intersects exactly one block I_k and, besides that, possibly has exactly one odd number. Then at least 3 elements of S are from I_k. Observe that no function from $\mathcal{H}$ can be equal to 0 on two of these elements and to 1 on the third one.

We now construct a computable asymptotic ERM A for $\mathcal{H}$. Assume that A receives on input a sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ of size m. Then A works as follows. First, it constructs a finite set of hypotheses $H_S \subseteq \mathcal{H}$ (we describe how A does it later). Then it goes through all hypotheses of H_S and outputs one which minimizes $L_S(h)$ among them. We will argue that

$$\min_{h \in H_S} L_S(h) \leq \min_{h \in \mathcal{H}} L_S(h) + \varepsilon_m, \qquad \varepsilon_m = \frac{1}{\min\left[f(\mathbb{N}) \setminus f(\{1, \ldots, m\})\right]}. \tag{5}$$

Since f is an injection, we have that $\varepsilon_m \to 0$ as $m \to \infty$. By (5) we will have that A is an asymptotic ERM for $\mathcal{H}$. We now explain how A constructs H_S. Let $M = \max\{x_1, \ldots, x_m\}$. First, A puts h_{M1} into H_S. Observe that h_{M1} equals 0 on $\{1, \ldots, M\}$. Then A puts into H_S all hypotheses of the form h_{kj} such that I_k intersects $\{1, \ldots, M\}$. Note that there are finitely many such h_{kj}, and we can effectively list them. This is because I_k is disjoint from $\{1, \ldots, M\}$ for $k \geq M$. Finally, A puts $h_1, \ldots, h_{\max\{m,M\}}$ into H_S. To compute these hypotheses, A computes $f(1), \ldots, f(\max\{m, M\})$.

To show that H_S satisfies (5), we take any $h \in \mathcal{H}$ and show that there exists $h' \in H_S$ with $L_S(h') \le L_S(h) + \varepsilon_m$. Assume first that $h = h_{kj}$ for some k, j. If $\mathsf{supp}(h_{kj})$ is disjoint from $\{1, \ldots, M\}$, we let $h' = h_{M1} \in H_S$. Observe that $L_S(h') = L_S(h_{kj})$ because both h and h' are equal to 0 on $\{1, \ldots, M\}$. Now, if $\mathsf{supp}(h_{kj})$ intersects $\{1, \ldots, M\}$, then $I_k \supset \mathsf{supp}(h_{kj})$ intersects $\{1, \ldots, M\}$ as well. Hence, $h_{kj} \in H_S$ in this case by definition of H_S. We then simply take $h' = h_{kj}$.

Next, assume that $h = h_a$ for some a. If $a \le \max\{m, M\}$, then h is also in H_S, so we can set $h' = h$. Likewise, we are done if h_a equals 0 on $\{1, \ldots, M\}$ (then again we can set $h' = h_{M1}$). The only remaining case is when $a > \max\{m, M\}$ and $\mathsf{supp}(h_a)$ intersects $\{1, \ldots, M\}$. Recall that $\mathsf{supp}(h_a) = I_{f(a)} \cup \{2a+1\}$. Since $a > M$, we see that $I_{f(a)}$ must intersect $\{1, \ldots, M\}$ in this case. Hence, H_S contains all hypotheses of the form $h_{f(a)j}$. Note that any of these $h_{f(a)j}$ differs from h_a exactly at two points: $2a+1$ and the jth element of $I_{f(a)}$. Hence, the difference between $L_S(h_a)$ and $L_S(h_{f(a)j})$ can be bounded by the number of times these two points appear in the sample (divided by m, the size of the sample). Since $a > M$, the point $2a+1$ actually does not appear in the sample, and there exists j such that the jth element of $I_{f(a)}$ appears in S at most $m/|I_{f(a)}| = m/f(a)$ times. Hence, $L_S(h_a) \le L_S(h_{f(a)j}) + \frac{1}{f(a)}$ for some j. Finally, since $a > m$ and f is injective, we have that $f(a) \in f(\mathbb{N}) \setminus f(\{1, \ldots, m\})$ and hence

$$\frac{1}{f(a)} \le \frac{1}{\min\left[f(\mathbb{N}) \setminus f(\{1, \ldots, m\})\right]} = \varepsilon_m.$$

It remains to show that $\mathcal{H}$ is not properly SCPAC learnable. Equivalently, by Theorem 8, we have to show that $\mathcal{H}$ does not have a computable ERM. Assume for contradiction that it does, and call it A. We deduce from it that $f(\mathbb{N})$ is decidable. Take any $k \in \mathbb{N}$. Our goal is to check if $k \in f(\mathbb{N})$. Define S to be a sample of size k, having each element of I_k exactly once, all of them labeled by 1. The only possible hypothesis in $\mathcal{H}$ which has 0-error on S (equivalently, is equal to 1 everywhere on I_k) is h_a for a with $f(a) = k$. Thus, such a hypothesis exists if and only if $k \in f(\mathbb{N})$, which can then be decided by running A on S, and checking whether the output hypothesis has 0-error on S, a contradiction. ■

7. Proof of Theorem 15

Proof We start by observing that if all the hypotheses in a class $\mathcal{H}$ have pairwise disjoint supports, then the Littlestone dimension of this class is at most 1. Indeed, consider any tree of depth 2. We show that $\mathcal{H}$ does not shatter it. Assume that its root is labeled by $x \in \mathbb{N}$. Note that there is at most one hypothesis $h \in \mathcal{H}$ with $h(x) = 1$. Hence, only one leaf under the right child of x can be reached via a hypothesis from $\mathcal{H}$. This means that $\mathcal{H}$ does not shatter this tree.

By Proposition 10, every CPAC learnable class admits a computable k-witness of VC dimension, for some $k \in \mathbb{N}$. We will construct a class that is not CPAC learnable by diagonalizing against all possible computable k-witnesses, for all k. We will guarantee that for every potential computable k-witness w, there is a block of natural numbers and a hypothesis from $\mathcal{H}$ which agrees with w on this block. We will use disjoint blocks for different potential witnesses to ensure that hypotheses in our class have pairwise disjoint supports. Thus, the class will have Littlestone dimension 1. We have to take into account that

Let $((M_e, k_e))_{e\in\mathbb{N}}$ be a computable enumeration of all pairs of the form (M, k), where M is a Turing machine and k is a natural number. Partition the set of even numbers into consecutive blocks $(I_e)_{e\in\mathbb{N}}$, where the size of I_e is k_e. Let $I_e(i)$ denote the ith smallest element of I_e. Also, fix a computable bijection $p\colon \mathbb{N}^2 \to \mathbb{N}$.

We now define the class of hypotheses $\mathcal{H}$. For every $e, s \in \mathbb{N}$ such that M_e halts on (a code for) the tuple $(I_e(1), \dots, I_e(k_e))$ in exactly s steps and outputs a binary word $x = x_1 \dots x_{k_e}$ of length k_e, we define $h_{es}\colon \mathbb{N} \to \{0, 1\}$ by

$$h_{es}(n) = \begin{cases} 1 & \text{if } n = 2p(e, s) + 1, \\ x_i & \text{if } n = I_e(i), \\ 0 & \text{otherwise,} \end{cases}$$

and let $\mathcal{H}$ be the collection of all such hypothesis. Notice that every h in $\mathcal{H}$ has finite support. We claim that, moreover, any two hypotheses from $\mathcal{H}$ have disjoint supports. Indeed, for any e, there is at most one s such that h_{es} is in our class. Hence, for any block I_e of even numbers, there is at most one hypothesis whose support intersects this block. Thus, two distinct hypotheses cannot have a common even number in their support. Likewise, each odd number can be in the support of at most one h_{es}, because p is a bijection. Hence, $\mathcal{H}$ has Littlestone dimension 1.

Next, we observe that $\mathcal{H}$ is decidable. We give an algorithm that, for a function $h\colon \mathbb{N} \to \{0, 1\}$ with finite support, decides, whether $h \in \mathcal{H}$. Note that h can be a member of $\mathcal{H}$ only if h has exactly one odd number n in its support. Using the fact that p is a computable bijection, we find unique e, s such that $n = 2p(e, s) + 1$ (via brute force). Note that h belongs to $\mathcal{H}$ if and only if h_{es} does and $h_{es} = h$. We then run M_e on the code for the tuple $(I_e(1), \dots, I_e(k_e))$. If it halts in exactly s steps and outputs a binary string $x_1 \dots x_{k_e}$ of length k_e, we know that h_{es} belongs to $\mathcal{H}$. It remains to check if $h_{es} = h$.

To complete the proof, suppose for contradiction that $\mathcal{H}$ is CPAC learnable and w is a computable k-witness of VC dimension for $\mathcal{H}$. There exists e such that M_e realizes w and $k_e = k + 1$. In particular, there exists s such that M_e halts on $(I_e(1), \dots, I_e(k_e))$ in exactly s steps and outputs $x = w((I_e(1), \dots, I_e(k_e))$. Observe that h_{es} belongs to $\mathcal{H}$ but agrees with w on $(I_e(1), \dots, I_e(k_e))$, a contradiction.

■

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143. Delle Rose, Kozachinskiy, and Rojas were also funded by the National Center for Artificial Intelligence CENIA FB210017, Basal ANID. Additionally, Kozachinskiy and Steifer were funded by the Millennium Science Initiative Program - Code ICN17002.

References

Nathanael Ackerman, Julian Asilis, Jieqi Di, Cameron Freer, and Jean-Baptiste Tristan. Computable pac learning of continuous features. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12, 2022.

Sushant Agarwal, Nivasini Ananthakrishnan, Shai Ben-David, Tosca Lechner, and Ruth Urner. On learnability with computable learners. In *Algorithmic Learning Theory*, pages 48–60. PMLR, 2020.

Sushant Agarwal, Nivasini Ananthakrishnan, Shai Ben-David, Tosca Lechner, and Ruth Urner. Open Problem: Are all VC-classes CPAC learnable? In *Conference on Learning Theory*, pages 4636–4641. PMLR, 2021.

Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *ACM Journal of the ACM (JACM)*, 69(4):1–34, 2022.

Shai Ben-David and Shai Pál, Dávid Shalev-Shwartz. Agnostic Online Learning. In *Annual Conference Computational Learning Theory*, 2009.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

Niki Hasrati and Shai Ben-David. On computable online learning. In *International Conference on Algorithmic Learning Theory*, pages 707–725. PMLR, 2023.

Nick Littlestone. Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Mach. Learn.*, 2(4):285–318, apr 1988. ISSN 0885-6125. doi: 10.1023/A:1022869011914. URL https://doi.org/10.1023/A:1022869011914.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. doi: 10.1017/CBO9781107298019.

Tom F Sterkenburg. On characterizations of learnability with computable learners. In *Conference on Learning Theory*, page 178:1–15. PMLR, 2022.

VN Vapnik and A Ya Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications*, 16(2):264, 1971.