PAC LEARNING, VC DIMENSION, AND THE ARITHMETIC HIERARCHY

WESLEY CALVERT

Abstract. We compute that the index set of PAC-learnable concept classes is m-complete Σ^0_3 within the set of indices for all concept classes of a reasonable form. All concept classes considered are computable enumerations of computable Π^0_1 classes, in a sense made precise here. This family of concept classes is sufficient to cover all standard examples, and also has the property that PAC learnability is equivalent to finite VC dimension.

1. Introduction

A common method to characterize the complexity of an object is to describe the degree of its index set [5, 6, 7, 8, 10, 13, 17]. In the present paper, we carry out this computation for the class of objects which are machine-learnable in a particular model. In doing so, we solve a problem related to one posed by Linial, Mansour, and Rivest in 1991 [16].

There have been several models of machine learning, dating back at least to Gold's seminal 1967 paper [12]. In Gold's basic model, the goal is that the machine should determine a Σ^0_1-index for a computably enumerable set of natural numbers — that is, an index for a computable function enumerating it, by receiving an initial segment of the string. Of course, many variations are possible, involving, for instance, the receipt of positive or negative information and the strength of the convergence criteria in the task of "determining" an index. This family of models has been studied by the recursion theory community (see, for instance, [11, 14, 21]), but is not the primary focus of this paper. One particular result, however, is of interest to us.

Theorem 1 (Beros [1])**.** *The set of Σ^0_1 indices for uniformly computably enumerable families learnable in each of the following models is m-complete in the corresponding class.*

(1) *TxtFin* — Σ^0_3
(2) *TxtEx* — Σ^0_4
(3) *TxtBC* — Σ^0_5
(4) *TxtEx** — Σ^0_5

1.1. PAC Learning.

The model of learning that concerns us here (*PAC learning*, for "Probably Approximately Correct") was first proposed by Valiant in [22]. Much of our exposition of the subject comes from [15]. The idea of the model is that it should allow some acceptably small error of each of two kinds: one arising from targets to be learned which are somehow too close together to be easily distinguished, and the other arising from randomness in the examples shown to the learner. Neither aspect is easily treated in Gold's framework of identifying indices

for computable enumerations of natural numbers by inspecting initial segments — neither a notion of "close" nor randomness in the inputs.

In the present paper, we will describe a framework in which to model PAC learning in a way which is suitable for recursion-theoretic analysis and which is broad enough to include many of the benchmark examples. We will then calculate the m-degree of the set of indices for learnable concept classes.

Definition 2 (Valiant)**.**

(1) *Let X be a set, called the* instance space*.*
(2) *Let $\mathcal{C}$ be a subset of $\mathcal{P}(X)$, called a* concept class*.*
(3) *The elements of $\mathcal{C}$ are called* concepts*.*
(4) *We say that $\mathcal{C}$ is* PAC Learnable *if and only if there is an algorithm φ_e such that for every $c \in \mathcal{C}$, every $\epsilon, \delta \in (0, \frac{1}{2})$ and every probability distribution $\mathcal{D}$ on X, the algorithm φ_e behaves as follows: On input (ϵ, δ), the algorithm φ_e will ask for some number n of examples, and will be given $\{(x_1, i_1), \ldots, (x_n, i_n)\}$ where x_j are independently randomly drawn according to $\mathcal{D}$, and $i_j = \chi_c(x_j)$. The algorithm will then output some $h \in \mathcal{C}$ so that with probability at least $1 - \delta$ in $\mathcal{D}$, the symmetric difference of h and c has probability at most ϵ in $\mathcal{D}$.*

This is a well-studied model — so well-studied, in fact, that it is more usual to talk about the complexity of the algorithm (in both running time and the number of example calls) than about its existence. Of course, in many learning situations, PAC learning is still impossible without regard to resources. For the present paper, we restrict ourselves to the problem of whether PAC learning is possible. Several examples are well-known.

Example 3. *Let $X = 2^n$, interpreted as assignments of truth values to Boolean variables. Then the class $\mathcal{C}$ of k-CNF expressions — that is, the class of propositional formulas on k Boolean variables, in conjunctive normal form — is PAC learnable (where each expression $c \in \mathcal{C}$ is interpreted as the set of truth assignments that satisfy it).*

Example 4. *Let $X = \mathbb{R}^d$. Then the class $\mathcal{C}$ of linear half-spaces (i.e. of subsets of $\mathbb{R}^d$, each defined by a single linear inequality) is PAC learnable.*

Example 5. *Let $X = \mathbb{R}^2$. Then the class of convex d-gons is PAC learnable for any d.*

1.2. **The Vapnik-Chervonenkis Dimension.** An alternate view of PAC learnability arises from work of Vapnik and Chervonenkis [23]. Again, we follow the exposition of [15].

Definition 6. *Let $\mathcal{C}$ be a concept class.*

(1) *Let $S \subseteq X$. Then $\Pi_{\mathcal{C}}(S) = |\{S \cap c : c \in \mathcal{C}\}|$.*
(2) *The Vapnik-Chervonenkis (VC) dimension of $\mathcal{C}$ is the greatest integer d such that $\Pi_{\mathcal{C}}(S) = 2^d$ for some S with cardinality d, if such an integer exists. Otherwise, the VC dimension of $\mathcal{C}$ is ∞.*

For example, if $\mathcal{C}$ is the class of linear half-spaces of $\mathbb{R}^2$, and if S is a set of size 4, suppose that k is the least natural number such that all of S is contained in the convex hull of $k \leq 4$ points of S. If $k < 4$, take a set S_0 of size k such that the

convex hull of S_0 contains S. The subset $S_0 \subset S$ cannot be defined by intersecting S with a linear half-space. If $k = 4$, then let S_0 be a diagonal pair, which again cannot be defined by intersection with a linear half-space. Consequently, the VC dimension of $\mathcal{C}$ must be at most 3. One can also show that this bound is sharp.

The connection between VC dimension and learnability is a theorem of Blumer, Ehrenfeucht, Haussler, and Warmuth showing that under some reasonable measure-theoretic hypotheses (which hold in all examples shown so far, and in all examples that will arise in the present paper), finite VC dimension is equivalent to PAC learnability [3].

Definition 7 (Ben-David, as described in [3])**.** *Let $R \subseteq \mathcal{P}(X)$, and let $\mathcal{D}$ be a probability distribution on X, and $\epsilon > 0$.*

1. *We say that $N \subseteq X$ is an ϵ-transversal for R with respect to $\mathcal{D}$ if and only if for any $c \in R$ with $P_{\mathcal{D}}(c) > \epsilon$ we have $N \cap R \neq \emptyset$.*
2. *For each $m \geq 1$, we denote by $Q^m_\epsilon(R)$ the set of $\vec{x} \in X^m$ such that the set of distinct elements of $\vec{x}$ does not form an ϵ-transversal for R with respect to $\mathcal{D}$.*
3. *For each $m \geq 1$, we denote by $J^{2m}_\epsilon(R)$ the set of all $\vec{x}\vec{y} \in X^{2m}$ with $\vec{x}$ and $\vec{y}$ each of length m such that there is $c \in R$ with $P_{\mathcal{D}}(c) > \epsilon$ such that no element of c occurs in $\vec{x}$, but elements of c have density at least $\frac{\epsilon m}{2}$ in $\vec{y}$.*
4. *We say that a concept class $\mathcal{C}$ is* well-behaved *if for every Borel set b, the sets $Q^m_\epsilon(R)$ and $J^{2m}_\epsilon(R)$ are measurable where $R = \{c \triangle b : c \in \mathcal{C}\}$.*

This notion of "well-behaved" is exactly the necessary hypothesis for the equivalence:

Theorem 8 ([3])**.** *Let $\mathcal{C}$ be a nontrivial, well-behaved concept class. Then $\mathcal{C}$ is PAC learnable if and only if $\mathcal{C}$ has finite VC dimension.*

In [16], Linial, Mansour, and Rivest asked for the complexity of computing the VC dimension of a finite family of concepts over a finite instance space. They showed that this problem can be solved in time $O(rn^{lg(r)})$, where r is the number of concepts in the class, n is the number of elements in the instance space, and $lg(r)$ denotes the base 2 logarithm of r. Schäfer [19] later showed that the analogous problem in a different representation system was Σ^P_3-complete. Schäfer's instance spaces and concept classes are still finite, which is a severe limitation in view of the ubiquitous examples one finds in the literature (linear half-spaces of $\mathbb{R}^d$, for example) in which both the instance space and the concept class are infinite.

2. Concepts and Concept Classes

The most general context in which PAC learning makes sense is far too broad to say anything meaningful about the full problem of determining whether a class is learnable. If we were to allow the instance space to be an arbitrary set, and a concept class an arbitrary subset of the powerset of the instance space, we would quickly be thinking about a non-trivial fragment of set theory.

In practice, on the other hand, one usually fixes the instance space, and asks whether (or how efficiently, or just by what means) a particular class is learnable. This approach is too narrow for the main problem of this paper to be meaningful. The goal of this section, then, is to describe a context broad enough to cover many of the usual examples, but constrained enough to be tractable.

Many of the usual examples of machine learning problems can be systematized in the framework of Π^0_1 classes, which will now be introduced. The following result is well-known, but a proof is given in [9], which is also a good general reference on Π^0_1 classes.

Theorem 9. *Let $c \subseteq 2^\omega$. Then the following are equivalent:*

1. *c is the set of all infinite paths through a computable subtree of 2^ω*
2. *c is the set of all infinite paths through a Π^0_1 subtree of 2^ω (i.e. a co-c.e. subtree)*
3. *$c = \{x \in 2^\omega : \forall n\ R(n, x)\}$ for some computable relation R, i.e. a relation R for which there is a Turing functional Φ such that $R(n, x)$ is defined by $\Phi^x(n)$.*

This equivalence (and other similar formulations could be added) gives rise to the following definition:

Definition 10. *Let $c \subseteq 2^\omega$. We say that c is a Π^0_1 class if and only if it satisfies one of the equivalent conditions in Theorem 9.*

Example 11. *There is a natural and uniform representation of all well-formed formulas of classical propositional calculus, each as a Π^0_1 class. We regard 2^ω as the assignment of values to Boolean variables, so that for $f \in 2^\omega$, the value $f(n) = k$ indicates a value of k for variable x_n. Let φ be a propositional formula. We construct a Π^0_1 subtree $T_\varphi \subseteq 2^\omega$ such that $f \in T_\varphi$ if and only if f satisfies φ. At stage n, for each $\sigma \in 2^{<\omega}$ of length n, we include $\sigma \in T_\varphi$ if and only if there is an extension $f \supset \sigma$ such that $f \models \varphi$. This condition can be checked effectively. Consequently, T_φ is a Π^0_1 subtree of 2^ω — intuitively, an infinite path f may fall out of T_φ at some point when we see a long enough initial segment to detect non-satisfiability, but unless it falls out at some finite stage, it is included.*

Example 12. *There is a natural and uniform representation of all* closed intervals *of $\mathbb{R}$ with computable endpoints, each as a Π^0_1 class. We take the usual representation of real numbers by binary strings. Let I be a closed interval with computable endpoints. We construct a Π^0_1 tree $T_I \subseteq 2^\omega$ such that the set of paths through T_I is equal to I. At stage s, we include in T_I all binary sequences σ of length s such that there is an extension $f \supset \sigma$ with $f \in I$. This condition can be checked effectively, by the computability of the endpoints of I. Consequently, T_I is a Π^0_1 subtree of 2^ω.*

Example 13. *There is a natural and uniform representation of all* closed linear half-spaces *of $\mathbb{R}^d$ which are defined by hyperplanes with computable coefficients, each half-space as a Π^0_1 class. We encode $\mathbb{R}^d$ as 2^ω in the following way: the ith coordinate of the point represented by the path f is given by the sequence $(f(k) : k \equiv i \bmod d)$. Now we encode a linear subspace into a subtree in the same way as with intervals in the previous example.*

Example 14. *There is a natural and uniform representation of all* convex d-gons in $\mathbb{R}^2$ *with computable vertices, with each d-gon represented by a Π^0_1 class. A convex d-gon is an intersection of d closed linear half-spaces, and so we exclude a node $\sigma \in 2^\omega$ from the tree for our d-gon if and only if it is excluded from the tree for at least one of those linear half-spaces.*

Note that the requirement of computable boundaries of these examples is not a practical restriction.

Proposition 15. *For any probability measure μ on $\mathbb{R}^d$ absolutely continuous with respect to Lebesgue measure, any $\epsilon > 0$, and any hyperplane given by $f(\bar{x}) = 0$, there is a hyperplane given by $\bar{f}(\bar{x}) = 0$ where $\bar{f}$ has computable coefficients, and where the linear half-spaces defined by these hyperplanes are close in the following sense: If H_f is defined by $f(\bar{x}) \leq 0$, if H_f^0 is defined by $f(\bar{x}) < 0$, and $H_{\bar{f}}$ is defined by $\bar{f}(\bar{x}) \leq 0$, then $\mu\left(H_f \triangle H_{\bar{f}}\right) < \epsilon$ and $\mu\left(H_f^0 \triangle H_{\bar{f}}\right) < \epsilon$.*

Proof. Since a hyperplane has Lebesgue measure 0, it suffices to show that we can achieve $\mu\left(H_f \triangle H_{\bar{f}}\right) < \epsilon$, since $\mu\left(H_f \triangle H_f^0\right) = 0$. Let $F_\mu(\bar{x})$ be the cumulative distribution function, that is,

$$F_\mu(\bar{x}) = \mu\left((-\infty, x_1] \times (-\infty, x_2] \times \cdots \times (-\infty, x_d]\right).$$

Now $\lim_{\bar{x} \to (-\infty, \ldots, -\infty)} F(\bar{x}) = 0$ and $\lim_{\bar{x} \to (+\infty, \ldots, +\infty)} F(\bar{x}) = 1$, so that for each i there are $x'_{i,\ell}$ and $x'_{i,u}$ such that

$$\lim_{\bar{y} \to (-\infty, \ldots, -\infty)} F(y_1, y_2, \ldots, y_{i-1}, x'_{i,\ell}, y_{i+1}, \ldots, y_d) < \frac{\epsilon}{4d}$$

and

$$\lim_{\bar{y} \to (+\infty, \ldots, +\infty)} F(y_1, y_2, \ldots, y_{i-1}, x'_{i,u}, y_{i+1}, \ldots, y_d) > 1 - \frac{\epsilon}{4d}.$$

Since computable points are dense in $\mathbb{R}^d$, we can find $\bar{x}_{i,\ell}$ and $\bar{x}_{i,u}$ which also satisfy these inequalities.

We then let $B \subseteq \mathbb{R}^d$ be the d-orthotope with vertices $\left\{\bar{b} : b_i \in \{x_{i,\ell}, x_{i,u}\}\right\}$. Note that $\mu(B) > 1 - \frac{\epsilon}{2}$.

Now either $f(\bar{x}) = 0$ has empty intersection with the interior of B, or it intersects at least d faces of B. In the first case, we may take $\bar{f}$ to be such that $\bar{f}(x) = 0$ contains the face of B nearest $f(\bar{x}) = 0$ (of course, it is possible that the nearest point of B to $f(\bar{x}) = 0$ has more than one incident face, e.g. if it is on an edge, in which any of the incident faces will suffice). Then the symmetric difference of H_f and $H_{\bar{f}}$ is contained in the complement of B, and so has measure less than $\frac{\epsilon}{2}$.

In the second case, let $\Phi_1, \ldots, \Phi_d$ be the faces of B which intersect $f(\bar{x}) = 0$. Note that d points, one taken taken from $\Phi_i - \bigcup_{j \neq i} \Phi_j$ for each i, are sufficient to determine a hyperplane.

Since computable points are dense in $\mathbb{R}$, we can find, in each face Φ_i, a computable point $\bar{a}_i$ so close to $f(\bar{x}) = 0$ that if $\bar{f}(\bar{x}) = 0$ is the hyperplane determined by the set of points $\{\bar{a}_i : i \leq d\}$, then

$$\mu\left(\left(H_f \triangle H_{\bar{f}}\right) \cap B\right) < \frac{\epsilon}{2}.$$

Although there are, of course, infinitely many choices for these a_i, one could determine them by starting with any computable points $(a_{0,i} : i \leq d)$ in the relevant faces, and then as long as the inequality above fails, find the i such that $a_{t,i}$ is farthest from $f(\bar{x}) = 0$ and replacing it with $a_{t+1,i}$ which is less than half as far away from $f(\bar{x}) = 0$, while setting $a_{t+1,j} = a_{t,j}$ for all $j \neq i$.

The coefficients of $\bar{f}$ are computable since the points $\bar{a}_i$ are computable. Furthermore,

$$\mu\left(H_f \triangle H_{\bar{f}}\right) < \frac{\epsilon}{2} + (1 - \mu(B)) < \epsilon.$$

□

Examples could be multiplied, of course, and it seems likely that many of the more frequently encountered machine learning situations could be included in this framework — certainly, for instance, any example in [2], [15], or [18].

We will work, for the purposes of the present paper, with instance space 2^ω and with concepts which are Π^0_1 classes. Unless otherwise noted, we use the standard product topology on 2^ω. It remains to describe the concept classes to be used. We make the following preliminary definitions:

Definition 16. (1) *Let $f, g \in 2^{\leq\omega}$. Then $d(f,g)$ is defined to be 2^{-n} where n is the least natural number such that $f(n) \neq g(n)$.*
(2) *Let $f \in 2^{\leq\omega}$ and $r \in \mathbb{R}$. We denote by $B_r(f)$ the set $\{g \in 2^{\leq\omega} : d(f,g) < r\}$.*
(3) *Let $S \subseteq 2^\omega$. We say that S is* computable *if and only if there is a computable function $f_S : 2^{<\omega} \times \mathbb{Q} \to \{0,1\}$ such that*

$$f_S(\sigma, r) = \begin{cases} 1 & \text{if } B_r(\sigma) \cap S \neq \emptyset \\ 0 & \text{if } B_{2r}(\sigma) \cap S = \emptyset \\ 0 \text{ or } 1 & \text{otherwise} \end{cases}$$

Part 3 of the definition is standard for metric spaces which, like 2^ω, may be given a computable structure (see [4, 24]).

Lemma 17. *The following conditions on a set S of paths through a computable tree $T \subseteq 2^\omega$ are equivalent:*
(1) *S is computable*
(2) *S is the set of paths through a computable tree T' with no dead ends.*

Proof. Suppose first that S is computable. We form T' by using f_S to examine each finite string $\sigma \in T$, as well as its immediate successors, $\sigma 0$ and $\sigma 1$. If $f_S(\sigma, 2^{-|\sigma|}) = 0$, we exclude σ from T'. If

$$f_S(\sigma 0, 2^{-|\sigma|-1}) = f_S(\sigma 1, 2^{-|\sigma|-1}) = 0$$

then we exclude σ from T'. Otherwise, we include σ in T'.

We now show that T' has no dead ends and has the same infinite paths as T. Suppose that $\sigma \in T$. If $f_S(\sigma, 2^{-|\sigma|}) = 1$ then either σ extends to an infinite path in T, or there is some τ such that τ extends to an infinite path in T and $d(\sigma, \tau) \leq 2^{-|\sigma|+1}$ (corresponding, respectively, to the first and third clauses of the definition of f_S, above); that is, τ differs from σ at the $|\sigma| - 1$ place, so that τ and σ share an immediate predecessor. Similarly, if $f(\sigma, r) = 0$ for some $r > 0$, then σ does not extend to an infinite path. Now if σ was included in T', then we have both $f_S(\sigma, 2^{-|\sigma|}) = 1$ and at least one of $f_S(\sigma 0, 2^{-|\sigma|-1}) = 1$ and $f_S(\sigma 1, 2^{-|\sigma|-1}) = 1$. Without loss of generality, say $f_S(\sigma 0, 2^{-|\sigma|-1}) = 1$. Then either $\sigma 0$ extends to a path or σ extends to a path. In either case, σ extends to a path. On the other hand, if σ extends to a path in T, then both $f_S(\sigma, 2^{-|\sigma|}) = 1$ and at least one of $f_S(\sigma 0, 2^{-|\sigma|-1}) = 1$ and $f_S(\sigma 1, 2^{-|\sigma|-1}) = 1$, so that σ is included in T'.

Suppose, on the other hand, that S is the set of paths through a computable tree T' with no dead ends. We define a function f_S to witness that S is computable. Let $\sigma \in 2^{<\omega}$ and $r \in \mathbb{Q}$. If $r \geq 1$, then set $f_S(\sigma, r) = 0$ if S is empty, and $f_S(\sigma, r) = 1$ if S is nonempty. If $r < 1$, let s be a natural number such that $2^{-s} < r < 2^{-(s-1)}$. Let $U_s = \{\tau \in 2^{<\omega} : |\tau| \leq s\}$. Note that U_s is finite.

To compute $f_S(\sigma, r)$ where $r < 1$, we check all $\tau \in U_S$. If there is a $\tau \in U_S$ such that $d(\sigma, \tau) < r$ and $\tau \in T'$, then set $f_S(\sigma, r) = 1$. Otherwise, set $f_S(\sigma, r) = 0$.

To show that this function works, note that if $g \in S$ with $g \in B_r(\sigma)$, then there is some restriction $\tau \subseteq g$ in U_s such that $d(\sigma, \tau) < r$, and it will have caused us to set $f_S(\sigma, r) = 1$. On the other hand, if $d(\sigma, s) \geq 2r$, then for any $\tau \in U_s$ with $d(\sigma, \tau) < r$, the element τ in $2^{<\omega}$ cannot extend to an infinite path in S, so that $\tau \notin T'$. □

There is an unfortunate clash of terminology in that the concept *classes* will have, for their members, Π^0_1 *classes*. In this paper, we will never use the term ambiguously, but because both terms are so well-established it will be necessary to use both of them.

Definition 18. *A weakly effective concept class is a computable enumeration $\varphi_e : \mathbb{N} \to \mathbb{N}$ such that $\varphi_e(n)$ is a Π^0_1 index for a Π^0_1 tree $T_{e,n}$.*

Naturally, we interpret each index enumerated as the Π^0_1 class of paths through the associated tree. We also freely refer to the indices (or trees, or Π^0_1 classes) in the range of a concept class as its elements.

This definition is almost adequate to our needs. We would like, however, one additional property: that a finite part of an effective concept class $\mathcal{C}$ should not be able to distinguish a non-computable point of 2^ω from all computable points, in the sense that if $y \in 2^\omega$ is noncomputable, then any finite Boolean combination of members of $\mathcal{C}$ containing y should also contain a computable point. This is reasonable: it would strain our notion of an "effective" concept class if it should fail. And yet it can fail with a weakly effective concept class: our classes may have no computable members at all, for instance. For that reason, we define an effective concept class as follows.

Definition 19. *An effective concept class is a weakly effective concept class φ_e such that for each n, the set c_n of paths through $T_{e,n}$ is computable as a subset of 2^ω.*

Note that for the set c_n to be computable, it is not necessary (or even likely) that all of its elements are computable.

In addition to the useful property mentioned above, which we will soon prove, there is another reason for preferring this stronger definition: Typically when we want a computer to learn something, it is with the goal that the computer will then be able to act on it. Computability of each concept is a necessary condition for this. The restriction corresponds, in the examples, to the restriction that a linear half-space, for instance, be defined by computable coefficients. The classes we consider in this paper will be effective concept classes.

Proposition 20. *Let $\mathcal{C}$ be an effective concept class, and let $c_1, \ldots, c_k \in \mathcal{C}$. Then for any $y \in 2^\omega$, there is a computable $x \in 2^\omega$ such that for each $i \in \{1, \ldots, k\}$, we have $x \in c_i$ if and only if $y \in c_i$.*

Proof. Let $y, c_1, \ldots, c_k$ be as described in the statement of the Proposition. Let I be the set of i such that $y \in c_i$ and J be the set of i such that $y \notin c_i$.

Suppose first that y is not in the boundary ∂c_i of c_i for each i. Then

$$y \in N := \left(\bigcap_{i \in I} c_i^\circ \right) \cap \left(\bigcap_{i \in J} (\overline{c_i})^\circ \right),$$

where S° denotes the interior of s and $\bar{S}$ the complement of S. Since I and J are finite, N is open. Since $y \in N$, the set N is nonempty, and must contain a basic open set of 2^ω, and so must contain a computable member, x, as required.

Now suppose that y is in ∂c_i for some i. Then we can compute y, using the function f_{c_i}, so that y is itself computable and we take $x = y$. □

We note that all of the examples given so far are effective concept classes.

Example 21. *The class of well-formed formulas of classical propositional calculus, and the class of k-CNF expressions (for any k) are effective concept classes, by the example above. Whether a given $y \in 2^\omega$ satisfies a particular formula can be determined by examining only finitely many terms of y.*

Example 22. *The class $\mathcal{C}$ of linear half-spaces in $\mathbb{R}^d$ bounded by hyperplanes with computable coefficients is an effective concept class. Recall that each linear half-space with computable coefficients is a computable set, since the distance of a point from the boundary can be computed.*

Example 23. *The class of convex d-gons in $\mathbb{R}^2$ with computable vertices is an effective concept class.*

Again, it appears that any example in any of the standard references is an effective concept class.

A pleasant feature of the effective concept classes is that they are always well-behaved.

Lemma 24. *A weakly effective concept class has finite VC dimension if and only if it is PAC learnable.*

Proof. Let $\mathcal{C}$ be an effective concept class. In [3], a proof of Ben-David is given that if $\mathcal{C}$ is universally separable — that is, if there is a countable subset $\mathcal{C}^*$ such that every point in $\mathcal{C}$ can be written as the pointwise limit of some sequence in $\mathcal{C}^*$ — then $\mathcal{C}$ is well-behaved. Since an effective concept class is always countable (i.e. it contains only countably many Π^0_1 classes), $\mathcal{C}$ is trivially universally separable. By Theorem 8, the conclusion holds. □

3. Bounding the Degree of the Index Set

We now turn toward the main problem of the paper, which we can now express exactly.

Problem 25. *Determine the m-degree of the set of all natural numbers e such that φ_e is a PAC-learnable effective concept class.*

One minor refinement in the problem remains: the difficulty of saying that e is the index for an effective concept class competes with that of saying that this concept class is learnable. Indeed, since determining that n is an X-index for an X-computable tree is m-complete $\Pi^0_2(X)$ (see [13, 20]), it follows that determining that n is a Π^0_1 index for a Π^0_1 tree is m-complete Π^0_3.

Since we will see that finite VC dimension can be defined at Σ^0_3, a driving force in the m-degree described in the problem above will be that it must compute all Π^0_3 sets. This tells us nothing about the complexity of learnability, but only about the complexity of determining whether we have a concept class. The usual way to deal with this issue is by the following definition.

Definition 26 ([5]). *Let $A \subseteq B$, and let Γ be some class of sets (e.g. Π^0_3).*

(1) *We say that A* is Γ within B *if and only if $A = R \cap B$ for some $R \in \Gamma$.*
(2) *We say that $S \leq_m A$* within B *if and only if there is a computable $f : \omega \to B$ such that for all n we have $n \in S \Leftrightarrow f(n) \in A$.*
(3) *We say that A* is m-complete Γ within B *if and only if A is Γ within B and for every $S \in \Gamma$ we have $S \leq_m A$ within B.*

We can now present the question in its final form.

Problem 27. *Let L be the set of indices for effective concept classes, and K the set of indices for effective concept classes which are PAC learnable. What is the m-degree of K within L?*

The solution to the problem will have two parts. In the present section, we will show that K is Σ^0_3 within L. In the following section, we show that K is m-complete Σ^0_3 within L.

We first reduce the problem to one where dimension is witnessed by computable paths through 2^ω.

Proposition 28. *An effective concept class $\mathcal{C}$ has infinite VC dimension if and only if for every d there are (not necessarily uniformly) computable elements*

$$(x_i : i < d)$$

such that $\Pi_{\mathcal{C}}(x_i : i < d) = 2^d$.

Proof. Let $(y_i : i < d)$ witness that $\mathcal{C}$ has VC dimension at least d, and denote by $D_1, \ldots, D_{2^d}$ elements of $\mathcal{C}$ which distinguish distinct subsets of $(y_i : i < d)$. For each $i < d$, there is a computable element x_i such that for every $j \leq 2^d$ we have $x_i \in D_j$ if and only if $y_i \in D_j$, by Proposition 20. Then $x_1, \ldots, x_d$ witness that $\mathcal{C}$ has VC dimension at least d. The converse is obvious. □

Proposition 29. *The set of indices for effective concept classes of infinite VC dimension is Π^0_3 within L.*

Proof. We begin by noting that if f is a computable function and T is a Π^0_1 tree, then it is a Π^0_1 condition that f is a path of T, and a Σ^0_1 condition that it is not, uniformly in a Π^0_1 index for T and a computable index for f. Further, if $\mathcal{C} = \varphi_e$ is an effective concept class, then for any $k \in \omega$, the condition that $k \in ran(\varphi_e)$ is a Σ^0_1 condition, uniformly in e and k.

Let $(x_1, \ldots, x_n)$ be a sequence of computable functions, $S \subseteq \{1, 2, \ldots, n\}$, and c a Π^0_1 class, represented by a Π^0_1 index for a tree in which it is the set of paths. We abbreviate by $[c \upharpoonright_n = S](\bar{x})$ the statement that for each $i \in \{1, \ldots n\}$, we have $x_i \in c$ if and only if $i \in S$. Now $c \upharpoonright_n = S$ is a d-Σ^0_1 condition, uniformly in the indices for the x_i and c.

We now note that $\mathcal{C} = \varphi_e$ has infinite VC dimension if and only if

$$\bigwedge\!\!\!\bigwedge_{n \in \mathbb{N}} \exists x_1, \ldots, x_n \bigwedge_{S \subseteq (n+1)} \exists k \, [\varphi_e(k) \upharpoonright_n = S](\bar{x}).$$

From the comments above, this definition is Π^0_3. □

4. Sharpness of the Bound

The completeness result in this section will finish our answer to the main question of the paper.

Theorem 30. *The set of indices for effective concept classes of infinite VC dimension is m-complete Π^0_3 within the set of indices for effective concept classes, and the set of indices for effective concept classes of finite VC dimension is m-complete Σ^0_3 within the set of indices for effective concept classes.*

Proof. It only remains to show completeness. For each Π^0_3 set S, we will construct a sequence of effective concept classes $(\mathcal{C}_n : n \in \mathbb{N})$ such that $\mathcal{C}_n$ has infinite VC dimension if and only if $n \in S$. In the following lemma, to simplify notation, we suppress the dependence of f on n.

Lemma 31. *There is a Δ^0_2 function $f : \mathbb{N} \to 2$ such that $f(s) = 1$ for infinitely many s if and only if $n \in S$.*

Proof. It suffices (see [20]) to consider S of the form $\exists^\infty x \forall y R(x, y, n)$, where R is computable. Now we set

$$f(x) = \begin{cases} 1 & \text{if } \forall y R(x, y, n) \\ 0 & \text{otherwise} \end{cases}.$$

This function is Δ^0_2-computable, and has the necessary properties. □

Now by the Limit Lemma, there is a uniformly computable sequence

$$(f_s : s \in \mathbb{N})$$

of functions such that for each x, for sufficiently large s, we have $f_s(x) = f(x)$.

We now define a set of functions that will serve as the elements that may eventually witness high VC dimension. Let $\{\pi_{s,t,j} : s, t, j \in \mathbb{N}, j < s\}$ be a discrete uniformly computable set of distinct elements of 2^ω such that $\pi_{s,t,j}(q) = \pi_{s,t',j'}(q)$ whenever $q < \min\{t, t'\}$.

We also initialize $G_{s,0} = \emptyset$ for each s. Denote by P_t a bijection

$$P_t : \mathcal{P}(\{1, \ldots, t\}) \to \{1, \ldots, 2^t\}.$$

At stage s of the construction, we consider $f_s(t)$ for each $t \leq s$. If $f_s(t) = 0$, then no action is required.

If $f_s(t) = 1$, then we find the least k such that $k \notin G_{t,s}$. Let $\{e_{t,i} : i < 2^t\}$ be Π^0_1 indices for trees such that $T_{e_{t,i}}$ consists exactly of the initial segments τ of $\pi_{t,k,j}$ where $j = P_t(S)$ for some $S \subseteq \{1, \ldots, t\}$ and $|\tau|$ is less than the first $z > s$ such that $f_z(t) = 0$. This can be done effectively exactly because we are looking for Π^0_1 indices, and the search is uniform. We then let i_s be the least such that $\mathcal{C}_n(i_s)$ is undefined, and take $\mathcal{C}_n(i_s + \ell) = e_{t,\ell}$ for each $\ell < 2^t$. We also set $G_{t,s} = G_{t,s-1} \cup k$.

Now for each t with $f(t) = 1$, there will be some s such that $f_{s'}(t) = f_s(t) = 1$ for all $s' > s$. Then at stage s we have added to $\mathcal{C}_n$ the Π^0_1 indices $\{e_{t,i} : i < 2^t\}$ guaranteeing that $\{\pi_{t,k,j} : j < t\}$ is shattered for some k.

For each t such that $f(t) = 0$ and each s such that $f_s(t) = 1$, there is some later stage s' such that $f_{s'}(t) = 0$, so any indices added at stage s will be indices for a tree with no paths — that is, for the empty concept.

Note that if the same t receives attention infinitely often — that is, if infinitely many different sets of classes are added to $\mathcal{C}_n$ to guarantee that the VC dimension

of $\mathcal{C}_n$ is at least t, this does not inflate the VC dimension beyond t. Indeed, the sets of witnesses will be pairwise disjoint, so no concept in $\mathcal{C}_n$ will include any mixture of witnesses from different treatments; the resulting sets will not be shattered.

We further note that all the Π^0_1 classes in $\mathcal{C}_n$ are computable. Indeed, each $c \in \mathcal{C}_n$ consists of finitely many (perhaps no) computable paths. Thus, $\mathcal{C}_n$ is an effective concept class.

Now if $n \notin S$, then $f(s) = 1$ for at most finitely many s, so that the VC dimension of $\mathcal{C}_n$ is finite. If $n \in S$, then $f(s) = 1$ for infinitely many s, so that the VC dimension of $\mathcal{C}_n$ is infinite (since sets of arbitrarily large size will be shattered). □

References

1. A. Beros, *Learning theory in the arithmetic hierarchy*, Journal of Symbolic Logic **79** (2014), 908–927.
2. C. M. Bishop, *Pattern recognition and machine learning*, Information Science and Statistics, Springer, 2006.
3. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, *Learnability and the Vapnik-Chervonenkis dimension*, Journal of the ACM **36** (1989), 929–965.
4. M. Braverman and M. Yampolsky, *Computability of julia sets*, Algorithms and Computation in Mathematics, no. 23, Springer, 2009.
5. W. Calvert, *The isomorphism problem for computable Abelian p-groups of bounded length*, Journal of Symbolic Logic **70** (2005), 331–345.
6. W. Calvert, D. Cenzer, V. Harizanov, and A. Morozov, *Effective categoricity of equivalence structures*, Annals of Pure and Applied Logic **141** (2006), 61–78.
7. ———, *Δ^0_2-categoricity of Abelian p-groups*, Annals of Pure and Applied Logic **159** (2009), 187–197.
8. W. Calvert, V. Harizanov, J. F. Knight, and S. Miller, *Index sets of computable structures*, Algebra and Logic **45** (2006), 306–325.
9. D. Cenzer, *Π^0_1 classes in computability theory*, Handbook of Computability, Studies in Logic and the Foundations of Mathematics, no. 140, Elsevier, 1999, pp. 37–85.
10. D. Cenzer and J. Remmel, *Index sets for Π^0_1 classes*, Annals of Pure and Applied Logic **93** (1998), 3–61.
11. M. Friend, N. B. Goethe, and V. Harizanov, *Induction, algorithmic learning theory, and philosophy*, Logic, Epistemology, and the Unity of Science, vol. 9, Springer, 2007.
12. E. M. Gold, *Language identification in the limit*, Information and Control **10** (1967), 447–474.
13. S. S. Goncharov and J. F. Knight, *Computable structure and non-structure theorems*, Algebra and Logic **41** (2002), 351–373.
14. V. Harizanov and F. Stephan, *On the learnability of vector spaces*, Journal of Computer and System Sciences **73** (2007), 109–122.
15. M. J. Kearns and U. V. Vazirani, *An introduction to computational learning theory*, MIT Press, 1994.
16. N. Linial, Y. Mansour, and R. L. Rivest, *Results on learnability and the Vapnik-Chervonenkis dimension*, Information and Computation **90** (1991), 33–49.
17. A. G. Melnikov and A. Nies, *The classification problem for compact computable metric spaces*, The Nature of Computation: Logic, Algorithms, Applications, Lecture Notes in Computer Science, vol. 7921, Springer, 2013, pp. 320–328.
18. S. Russell and P. Norvig, *Artificial intelligence*, 3rd ed., Prentice Hall, 2010.
19. M. Schaefer, *Deciding the Vapnik-Chervonenkis dimension is Σ^P_3-complete*, Journal of Computer and System Sciences **58** (1999), 177–182.
20. R. I. Soare, *Recursively enumerable sets and degrees*, Springer-Verlag, 1987.
21. F. Stephan and Y. Ventsov, *Learning algebraic structures from text*, Theoretical Computer Science **268** (2001), 221–273.
22. L. G. Valiant, *A theory of the learnable*, Communications of the ACM **27** (1984), 1134–1142.
23. V. N. Vapnik and A. Ya. Chervonenkis, *On the uniform convergence of relative frequencies of events to their probabilities*, Theory of probability and its applications **16** (1971), 264–280.
24. K. Weihrauch, *Computable analysis*, Texts in Theoretical Computer Science, Springer, 2000.