Weihrauch complexity of PAC learning problems

École polytechnique third-year (M1) internship report

Guillaume Chirache
Supervised by Vasco Brattka
Universität der Bundeswehr München

April to July 2025

Contents

1 Introduction

I am currently completing my École polytechnique third-year internship at the Bundeswehr University in Munich under the supervision of Vasco Brattka. As a continuation of the research project about NP-complete problems in computable analysis I conducted this year with Olivier Bournez, I am working on classifying machine learning problems over the Weihrauch lattice.

While classic computability theory describes inputs and outputs with words or natural numbers, computable analysis allows them to belong to uncountable sets like the real numbers. This unusual approach (continuity is rare in computer science) provides a bridge between computability theory and many analysis problems that could not be considered otherwise, and a formal framework to characterize their computational complexity. It has strong ties with topological properties over the considered spaces, and it is often possible to derive results in classic computability. Analogous to Turing degrees, we can classify problems into Weihrauch degrees to characterize how hard they are to compute, but these degrees are a less abstract construction and are full of natural problems which arise from various areas of mathematics. The reason for this is that many such problems are *discontinuous*, meaning that output symbols depend on the whole input. Discontinuous functions cannot be computable.

I am particularly focused on PAC learning. The goal of this learning paradigm is to identify an "almost" correct *hypothesis* among a set of labelling functions $\mathbb{N} \to \{0, 1\}$ with high probability, provided the data sample is large enough. This set of labelling functions, also known as *hypothesis class*, is crucial since it can be shown that the whole Cantor space $2^{\mathbb{N}}$ is not PAC learnable. For this reason, we are considering problems whose input is a hypothesis class. Because of the cardinality of $\mathcal{P}(2^{\mathbb{N}})$, we need the expressivity power of computable analysis, and even there, we have to make some restrictions which we discuss in subsection 4.1.

Links between PAC learning and computability have already been explored, and they represent a quite natural question since the ultimate goal of machine learning theory is to be applied on real machines. In fact, there has been recent progress about it, with Tom Sterkenburg having found in 2022 a characterization of PAC learnable hypothesis classes with a computable learner ([Sterkenburg, 2022]). However, we are studying them from the perspective of computable analysis for the first time. The motivation is that determining the Weihrauch degree of a mathematical theorem or notion says a lot about its inherent complexity, and can often be the source of other results with concrete applications.

The first two sections describe a state of the art. In section 2, we introduce the main notions of computable analysis which we are using in the rest of the report, especially Weihrauch reducibility. Section 3 is an introduction to the theory of PAC learning and the importance of the VC dimension. Both of these sections mainly rely on existing resources and do not contain any new result, but presentation and comments are mine.

Section 4 is my work on classifying the VC dimension over the Weihrauch lattice. We are interested in this problem because a hypothesis class is learnable if and only if its VC dimension is finite, and the VC dimension provides an upper bound on the sample size required to achieve a given precision with a certain probability. To classify this problem, we need to distinguish between different cases depending on the way the information about the hypothesis class is provided. It turns out that only the *positive* information (i.e., the extendable prefixes) is useful in a sense that we specify. Because this area of the Weihrauch lattice was not extensively studied (although the problem SORT was well identified), this part also contains a few new results about some benchmark problems (subsection 4.2). For instance, computing the limit of a sequence in $\mathbb{N}_\infty$ is easier if we assume the sequence is non-decreasing, while both problems are equivalent for a limit in $\mathbb{N}$.

Section 5 is about the computation of a learner for a hypothesis class. We actually focus on the computation of an *empirical risk minimizer*, which is always a valid learner if the class is learnable. Most results proven in sections 4 and 5 are new and are the bulk of my internship

work. Because this report is submitted more than two weeks before the end of the internship, the work on the degree of finding a learner is still in progress and I hope to complete it with other results.

1.1 Notations

- By $\mathbb{N} = \{0, 1, \dots\}$ we denote the set of natural numbers. We also write $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$.
- By $[\![n, m]\!]$ we denote the set of integers $[n, m] \cap \mathbb{Z}$.
- By B^A we denote the set of (total) functions from A to B.
- A *partial multi-valued function* $f :\subseteq A \rightrightarrows B$ from A to B, where "$\subseteq$" indicates partiality and "$\rightrightarrows$" indicates multi-valuedness, is a subset of $A \times B$ without any further constraints. We define $\operatorname{dom} f = \{x \in A \mid f(x) \neq \varnothing\}$ and $\operatorname{range} f = \bigcup_{x \in A} f(x)$. If for all $x \in \operatorname{dom} f$, $|f(x)| = 1$, we say that f is a partial single-valued function. For such functions, we write $f :\subseteq A \to B$ and if $f(x) = \{y\}$, we simply write $f(x) = y$.
- For two multi-valued functions $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq Y \rightrightarrows Z$, we define the *composition* $g \circ f :\subseteq X \rightrightarrows Z$ of g and f on $\operatorname{dom}(g \circ f) = \{x \in X \mid x \in \operatorname{dom} f \text{ and } f(x) \subseteq \operatorname{dom} g\}$ by:

  $$(g \circ f)(x) = \bigcup_{y \in f(x)} g(y).$$

  We also simply write gf. Note that the domain choice is more conservative than it could be, requiring any element of $f(x)$ for $x \in \operatorname{dom}(gf)$ to have an image under g. Intuitively, this is because we want to be able to finish the computation, whatever the choice made for the image under f.
- A (finite) *word* of length n over the alphabet Σ is a map $w : [\![0, n-1]\!] \to \Sigma$. We write $|w| = n$ and $w = a_0 a_1 \dots a_{n-1}$ if for all $i \in [\![0, n-1]\!]$, $w(i) = a_i$.
- The *empty word* (of length 0) is denoted ε and the set of all words is $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$. In particular, $\mathbb{N}^*$ is the set of finite sequences of natural numbers and **not** $\mathbb{N} \setminus \{0\}$.
- An *infinite word* (or *infinite sequence*) over Σ is a map $p : \mathbb{N} \to \Sigma$. We write $p = a_0 a_1 \dots$ if for all $i \in \mathbb{N}$, $p(i) = a_i$. We consider that $|p| = \infty$ if the context requests it.
- If $a \in \Sigma$, we denote by $\hat{a}$ the constant word $n \mapsto a$.
- We denote by uv the *concatenation* of a finite word u and a finite or infinite word v.
- We write $w\Sigma^{\mathbb{N}} = \{p \in \Sigma^{\mathbb{N}} \mid \forall i \in [\![0, |w| - 1]\!],\ p(i) = w(i)\}$ the *cylinder* of prefix w.
- We identify 0 with the empty set $\varnothing$ and $n + 1$ with $[\![0, n]\!]$ for any $n \in \mathbb{N}$. For instance, $2^{\mathbb{N}}$ is the Cantor space, that is the set of functions $\mathbb{N} \to \{0, 1\}$, and the initial segment of length n of an infinite word $p \in \Sigma^{\mathbb{N}}$ is $p|_n = p|_{[\![0,n-1]\!]} = p(0) \dots p(n-1)$.
- A finite word u is said to be a *prefix* of a finite or infinite word v, in symbols $u \sqsubseteq v$ if $|u| \leqslant |v|$ and for any $i \in [\![0, |u| - 1]\!]$, $u(i) = v(i)$. We also say that u is a *strict prefix* of v and we write $u \sqsubset v$ if additionally $|u| < |v|$ holds.
- If $A \subseteq X$, the *indicator function* of A is the function $\mathbb{1}_A : X \to \{0, 1\}$ defined by $\mathbb{1}_A(x) = 1$ if $x \in A$ and $\mathbb{1}_A(x) = 0$ otherwise. The superset X is supposed to be clear by context.

2 Fundamentals of computable analysis

This first section is an introduction to computable analysis, which focuses on the notions we need for the rest of the report. It is largely inspired by the presentation made in [Brattka, 2018] and [Brattka and Hertling, 2021], although some remarks are mine. My knowledge on computable analysis also comes from the research project [Chirache, 2025] (in French) I made this year with Olivier Bournez, which was also based on [Bournez, 2023] (in French) and [Brattka et al., 2008]. The main two differences here with this project is that we are using the Baire space instead of the Cantor space for infinite sequences, and that we are purely interested in the *computable* aspect of the problems we study, and not their time or space complexity. In the whole report, *complexity* refers to the Weihrauch complexity, introduced in Definition 14.

Even if it is not directly used here because it does not deal with computable analysis, the book *Calculabilité* ([Monin and Patey, 2022], in French) by Benoît Monin and Ludovic Patey has also significantly contributed to my understanding of computability in general.

2.1 Computable functions over the Baire space

The core idea behind computable analysis is to use an uncountable space to describe inputs and outputs of problems, so we can represent much more complex objects. There are two natural possible choices.

Definition 1 (Cantor and Baire spaces)**.**

- The *Cantor space* is the set $2^{\mathbb{N}} = \{0,1\}^{\mathbb{N}}$ endowed with the product topology of the discrete topology on $\{0,1\}$.
- The *Baire space* is the set $\mathbb{N}^{\mathbb{N}}$ endowed with the product topology of the discrete topology on $\mathbb{N}$.

We work with Turing machines whose tapes contain symbols from either $\{0,1\}$ or $\mathbb{N}$, depending on the space we choose. In both cases, the ensuing theory is similar, since we can easily encode elements from one set to the other. As it tends to be the convention in recent works, we choose to work in the Baire space. However, it turns out that the machine learning theory we study will lead us to consider elements of the Cantor space as well.

Let us first try to better understand the topology we are using.

Proposition 2. *Let $X = \{0,1\}$ or $X = \mathbb{N}$. Then:*

- *The topology of $X^{\mathbb{N}}$ is induced by the metric $d : X^{\mathbb{N}} \times X^{\mathbb{N}} \to \mathbb{R}_+$ defined by*

$$d(p,q) = 2^{-\min\{i \in \mathbb{N} | p(i) \neq q(i)\}}$$

  for $p \neq q$ and $d(p,p) = 0$ for $p \in X^{\mathbb{N}}$.
- *Cylinders $wX^{\mathbb{N}}$ for $w \in X^*$ form a basis of the topology of $X^{\mathbb{N}}$.*

We work with Turing machines equipped with a finite number of infinite tapes. All of them are initially blank, except the *input tape*, which carries the input. One of them is called the *output tape* and is one-way, which means that the writing of a symbol cannot be cancelled. A Turing machine on an infinite input p can therefore either halt and produce a (possibly empty) finite output, or it can never halt and produce a finite or infinite output. A way to formally define such machines can be found in the report [Chirache, 2025] of my research project with Olivier Bournez.

Definition 3 (Computable function)**.** A function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is said to be computable if there exists a Turing machine that upon input $p \in \mathrm{dom}(F)$ produces the infinite word $F(p)$ on its one-way output tape in the long run.

As any open set can be written as a union of cylinders, a continuous function $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is a function that such any prefix (or symbol) of the output only depends on finitely many symbols of the input. This yields the following result, which Émile Borel stated as early as 1912 in *Leçons sur la théorie des fonctions*:

Proposition 4. *Any computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is continuous.*

Proof. If F is computable, there is a Turing machine computing it. This machine writes any output symbol after finitely many steps, and in particular it has only read finitely many symbols of the input. All output symbols therefore only depend on finitely many input symbols, which means that F is continuous. □

Naturally, we don't just want to consider sequences of natural numbers as inputs and outputs. We may need to consider combinations of such sequences, and combinations of natural numbers. Fortunately, it is easy to encode such information with the following tupling functions:

Definition 5 (Tupling functions)**.** For $(n, k) \in \mathbb{N}^2$, $(p, q) \in (\mathbb{N}^{\mathbb{N}})^2$ and $(p_n) \in (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}}$, we define

- $\langle n, k\rangle = k + \sum_{i=1}^{n+k} i$,
- $\langle p, q\rangle(2n) = p(n)$ and $\langle p, q\rangle(2n+1) = q(n)$,
- $\langle p_0, p_1, \dots\rangle(\langle n, k\rangle) = p_n(k)$.

The first two functions can be inductively extended by setting for instance $\langle n_0, n_1, \dots, n_k\rangle = \langle n_0, \langle n_1, \dots, n_k\rangle\rangle$ for $k \geqslant 2$. These tupling functions rely on the fact that the map $(n, k) \mapsto k + \sum_{i=1}^{n+k} i$ is bijective from $\mathbb{N}^2$ to $\mathbb{N}$. They show that the expressivity power of the Baire space is quite high, and we will often use them when there is a need to consider complex inputs and outputs.

We say that a function is computable with respect to some oracle q if it can be computed by a Turing machine which additionally has access to a tape carrying q. To avoid formalizing this, we give the following equivalent definition, which makes use of tupling functions:

Definition 6 (Relative computability)**.** A function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is computable with respect to $q \in \mathbb{N}^{\mathbb{N}}$ (or *computable relative* to q) if there exists a computable function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for any $p \in \mathrm{dom}\, F$, $\langle p, q\rangle \in \mathrm{dom}\, G$ and $F(p) = G\langle p, q\rangle$.

Proposition 7. *A function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is continuous if and only if it is computable with respect to some oracle $q \in \mathbb{N}^{\mathbb{N}}$.*

A proof is available in [Brattka, 2018], as well as in the report of my research project [Chirache, 2025].

The intuition behind the previous result is that a function can be non-computable for basically two reasons. Either the information to write does not exist because the functions is discontinuous, or it exists but is inherently uncomputable. For instance, we can set $p(e)$ to be 1 if the (classic) Turing machine of code e halts on input e and 0 otherwise, and the constant function $q \mapsto p$ would be continuous but not computable as the halting problem is undecidable.

2.2 Expressing problems over other spaces

Like with classic computability theory, we can study many more objects than sequences of natural numbers by representing them as elements of $\mathbb{N}^{\mathbb{N}}$.

Definition 8 (Representation)**.** A *representation* of a set X is a surjective (possibly partial) map $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$. We then say that (X, δ) (or simply X) is a *represented space*.

The representation we choose must be chosen carefully because topological and computability properties of functions heavily depend on it.

Definition 9 (Admissible representation)**.**

- Let δ and δ' be two representations of the same set X. Then δ' is said to be topologically reducible to δ, in symbols $\delta' \leqslant_{\mathrm{t}} \delta$, if there is a continuous function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for any $p \in \operatorname{dom} \delta'$, $p \in \operatorname{dom}(\delta F)$ and $\delta F(p) = \delta'(p)$.
- A representation of a topological space X is said to be *admissible* if it is continuous and for any other continuous representation δ' of X, $\delta' \leqslant_{\mathrm{t}} \delta$.

That $\delta' \leqslant_{\mathrm{t}} \delta$ basically means that δ carries less topological information than δ'.

An admissible representation is a representation that "works well" with the topology on the considered space, because it carries the minimum required information to be continuous.

We can now introduce the notion of problem that we are using.

Definition 10 (Problem)**.** Let X and Y be two represented spaces. A problem is a partial multifunction $f :\subseteq X \rightrightarrows Y$.

Definition 11 (Realizer)**.** Let $f :\subseteq X \rightrightarrows Y$ be a problem on represented spaces (X, δ_X) and (Y, δ_Y). We say that $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ realizes f, in symbols $F \vdash f$, if for all $p \in \operatorname{dom}(f\delta_X)$, $p \in \operatorname{dom}(\delta_Y F)$ and $\delta_Y F(p) \in f\delta_X(p)$.

$$\begin{array}{ccc} \mathbb{N}^{\mathbb{N}} & \xrightarrow{F} & \mathbb{N}^{\mathbb{N}} \\ {\scriptstyle \delta_X}\downarrow & & \downarrow{\scriptstyle \delta_Y} \\ X & \xrightarrow[f]{} & Y \end{array}$$

A realizer is a translation of a problem over the Baire space. Because we want to compute such realizers with deterministic programs, we don't allow them to be multi-valued, but they can choose any of the outputs allowed by the problem.

Definition 12. We say that a problem is *computable* (*continuous*) if it has a computable (continuous) realizer.

2.3 Weihrauch reducibility and complexity

Some problems are computable, while others are not. However, these non-computable problems are not all equivalent and can be more or less hard. We need a way to formalize this fact and classify their "computational complexity".

The tool we are using is the Weihrauch reduction, which is a many-one reduction for computable analysis problems. For $F, G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, we define $\langle F, G\rangle(p) = \langle F(p), G(p)\rangle$ for $p \in \operatorname{dom} F \cap \operatorname{dom} G$.

Definition 13 (Weihrauch reduction)**.** Let f and g be two problems (not necessarily on the same represented spaces).

- We say that f is *Weihrauch reducible* to g, in symbols $f \leqslant_{\mathrm{W}} g$ if there exist computable functions $H, K :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for any realizer G of g, $H\langle \mathrm{Id}_{\mathbb{N}^{\mathbb{N}}}, GK\rangle \vdash f$.
- We say that f is *strongly Weihrauch reducible* to g, in symbols $f \leqslant_{\mathrm{sW}} g$ if there exist computable functions $H, K :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for any realizer G of g, $HGK \vdash f$.

This definition is illustrated in figure 1.

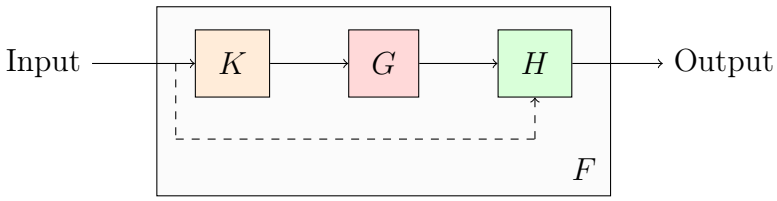

Figure 1: Weihrauch reducibility (the dashed arrow only exists for a weak reduction).

That f is Weihrauch reducible to g intuitively means that it is possible to compute f with a "special machine" making a unique call to an "oracle" g. However, this intuition is not easy to formalize, since such a machine would need to have produced the entirety of the input of g before that call, as g is not necessarily continuous. Instead, to convince ourselves that this reduction is reasonable, we can notice how similar it is to usual many-one reductions, and the fact that a function reducible to a computable function is itself computable. Indeed, in that case, we only need to produce finitely many symbols of the input of g to compute a symbol of its output.

The strong Weihrauch reduction is less natural since the idea of forbidding access to the initial input seems somewhat arbitrary. The intuition is that it is not possible to "recover" data from this input, even in a computable way. For instance, only constant functions are strongly Weihrauch reducible to a constant function, while any computable function is weakly Weihrauch reducible to any constant function. Showing that a Weihrauch reduction is strong is interesting not only because it is a stronger result, but also because strong reducibility behaves better with some properties like the jump (Proposition 18).

It is easy to show that the (strong) Weihrauch reducibility is reflexive and transitive, and therefore induces a preorder on problems. By defining an equivalence relation, we can make it an order.

Definition 14 (Weihrauch degree)**.** We say that f is (strongly) *Weihrauch equivalent* to g, in symbols $f \equiv_{\mathrm{W}} g$ ($f \equiv_{\mathrm{sW}} g$) if $f \leqslant_{\mathrm{W}} g$ and $g \leqslant_{\mathrm{W}} f$ ($f \leqslant_{\mathrm{sW}} g$ and $g \leqslant_{\mathrm{sW}} f$). This defines an equivalence relation and the equivalence class of f is called the (strong) *Weihrauch degree* or (strong) *Weihrauch complexity* of f. We also write $f <_{\mathrm{W}} g$ ($f <_{\mathrm{sW}} g$) if $f \leqslant_{\mathrm{W}} g$ but $f \not\equiv_{\mathrm{W}} g$ ($f <_{\mathrm{sW}} g$ but $f \not\equiv_{\mathrm{sW}} g$).

The Weihrauch reduction is therefore a partial order on Weihrauch degrees. It is actually a lattice, which means that any two elements have a supremum and an infimum. Like Turing degrees, Weihrauch degrees have been extensively studied. An interesting difference is that while Turing degrees are a quite abstract construction, Weihrauch degrees are full of natural problems, which often come from theorems of the analysis.

2.4 Unary operators on problems

There are many possible operations on problems. We mentioned the fact that there are a supremum and an infimum operators. We are going to introduce two unary operators, which we will make extensively use of. Let us first define the following problem:

Definition 15 (The limit map)**.** The limit map (or $\lim_{\mathbb{N}^\mathbb{N}}$ problem) is defined by

$$\begin{array}{rccc} \lim_{\mathbb{N}^\mathbb{N}} :\subseteq & \mathbb{N}^\mathbb{N} & \to & \mathbb{N}^\mathbb{N} \\ & \langle p_0, p_1, \dots \rangle & \mapsto & \lim_{n\to\infty} p_n \end{array}$$

We can also define the same problem $\lim_{2^\mathbb{N}}$ on the Cantor space. It is strongly Weihrauch equivalent to $\lim_{\mathbb{N}^\mathbb{N}}$ since one can encode natural numbers with a binary representation.

Clearly, this problem is not continuous (hence not computable). It can actually be shown that it is Weihrauch equivalent to the Turing jump. This motivates the introduction of the jump of a problem, which is basically its composition with $\lim_{\mathbb{N}^\mathbb{N}}$.

Definition 16 (Jump of a represented space)**.** Let (X, δ) be a represented space. Its *jump* is the represented space $(X, \delta \circ \lim_{\mathbb{N}^\mathbb{N}})$. We usually denote it by X'.

Definition 17 (Jump of a problem)**.** The *jump* of the problem $f :\subseteq X \rightrightarrows Y$ is the same problem $f' :\subseteq X' \rightrightarrows Y$ with the modified input space X'.

Surprisingly, jumps behave badly with weak Weihrauch reductions. It is possible to have both $f \leqslant_\mathrm{W} g$ and $g' <_\mathrm{W} f'$ (see [Brattka and Hertling, 2021]). However, the jump is monotonic for the strong Weihrauch reduction.

Proposition 18 (Monotonicity of the jump)**.** *Let f, g be two problems such that $f \leqslant_\mathrm{sW} g$. Then $f' \leqslant_\mathrm{sW} g'$.*

We also introduce the *parallelization* of a problem. It is basically the parallel resolution of countably many instances of the problem.

Definition 19 (Parallelization)**.**

- Let (X, δ_X) be a represented space. We endow the set $X^\mathbb{N}$ with the representation $\delta_{X^\mathbb{N}}$ defined by
$$\delta_{X^\mathbb{N}}(\langle p_0, p_1, \dots \rangle) = (x_n)_{n\in\mathbb{N}} \iff \forall n \in \mathbb{N},\ \delta_X(p_n) = x_n.$$
- Let $f :\subseteq X \rightrightarrows Y$ be a problem. The *parallelization* of f is the problem $\hat{f} :\subseteq X^\mathbb{N} \rightrightarrows Y^\mathbb{N}$ defined by $(y_n)_n \in f((x_n)_n)$ if and only if for any $n \in \mathbb{N}$, $y_n \in f(x_n)$.

Proposition 20. *If a problem is computable, then its parallelization is computable too.*

This last result can be used to show that parallelization is monotonic for both the weak and the strong Weihrauch reductions.

Proposition 21. *Let f, g be two problems. Then:*

- $f \leqslant_\mathrm{W} g \implies \hat{f} \leqslant_\mathrm{W} \hat{g}$, *and*
- $f \leqslant_\mathrm{sW} g \implies \hat{f} \leqslant_\mathrm{sW} \hat{g}$.

The property tells us that parallelization is monotonic. Clearly, it is also extensive, and it is idempotent since a countable union of countable instance sets is also countable. This exactly means the following proposition.

Proposition 22. *Parallelization is a closure operator for both $\leqslant_\mathrm{W}$ and $\leqslant_\mathrm{sW}$.*

3 PAC learning and VC dimension

3.1 Learning hypotheses from samples

We are interested in classifying machine learning problems over the Weihrauch lattice, so we first need to introduce the notions we are using. The goal of machine learning is to generalize what can be learnt from known data to unseen data. A reference on this subject is the book [Shalev-Shwartz and Ben-David, 2014]. Here, the inputs we consider are elements of $\mathbb{N}$ (which can be used to encode any data from a countable set), to which we want to assign a *label*, that is 0 or 1.

For instance, imagine you are an email provider and you want to determine whether incoming emails are spam. You could encode the email data into an integer, and the label would indicate whether it should be sent to the spam folder. You could then use reports from your users (buttons "Spam" and "Not a spam") and learn from them to improve your automatic classification.

Our goal is, from a *sample* (in our example a set of emails which are already classified), to draw a *hypothesis* (a general rule) in the form of a map $\mathbb{N} \to \{0,1\}$, i.e., an element of the Cantor set. More formally, a sample is a tuple of pairs (n, ℓ), where $n \in \mathbb{N}$ is the input and $\ell \in \{0,1\}$ is the associated label. To measure how a hypothesis is appropriate for a sample, we introduce the *empirical risk*, which is the proportion of sample elements which do not match the hypothesis.

Definition 23 (Sample, hypothesis and empirical risk)**.**

- A *sample* S is an element of the sample set $\mathcal{S} = \bigcup_{n\in\mathbb{N}} (\mathbb{N} \times \{0,1\})^n$.
- A *hypothesis* is a function $h : \mathbb{N} \to \{0,1\}$.
- The *empirical risk* of $h : \mathbb{N} \to \{0,1\}$ over the training sample $S = ((x_1, y_1), \dots, (x_n, y_n)) \in (\mathbb{N} \times \{0,1\})^n$ is
$$L_S(h) = \frac{|\{i \in [\![1, n]\!] \mid h(x_i) \neq y_i\}|}{n}.$$

Note that we have not put many constraints on samples. They can contain redundant or even contradictory information (for instance, a single sample could contain $(12, 0)$ twice and $(12, 1)$ once). The idea is that our data can itself be imperfect: the same email could be reported as a spam by a user, and as legitimate by another one. We need to choose the hypothesis that suits best this information. The function which maps a sample to a hypothesis is called a *learner*.

Intuitively, the best solution is to choose a learner that minimizes the empirical risk. Such a learner is called an *empirical risk minimizer*, or ERM. However, an ERM without any further constraints can be a **very bad** learner. For instance, it can assign 0 to any input it has not seen. This pitfall is known as *overfitting*: we are relying too much on the training data and trying to match it too literally. Instead, we should make some assumptions about what reasonable hypotheses can look like. To do this, we introduce the notion of *hypothesis class*.

Definition 24 (Hypothesis class)**.**

- A *hypothesis class* is a non-empty set of hypotheses, i.e., a subset of the Cantor space $\varnothing \neq \mathcal{H} \subseteq 2^{\mathbb{N}}$.
- A *learning function* (or *learner*) for $\mathcal{H}$ is a map $A : \mathcal{S} \to \mathcal{H}$.
- We say that the learner $A : \mathcal{S} \to \mathcal{H}$ is an $\mathrm{ERM}_{\mathcal{H}}$ (or *empirical risk minimizer* for $\mathcal{H}$) if for any $S \in \mathcal{S}$:
$$A(S) \in \operatorname*{argmin}_{h\in\mathcal{H}} L_S(h).$$
We also denote by $\mathrm{ERM}_{\mathcal{H}}$ the set of all empirical risk minimizers for $\mathcal{H}$.

Conversely, choosing a hypothesis class which is too restricted is another pitfall, since "good" hypotheses that suit the data well would be discarded. This is known as *underfitting*, and we see that there is a trade-off between overfitting and underfitting. To formalize it, we need a way to characterize hypothesis class that can be learnt.

3.2 Probably approximately correct learnability

We want to come up with a definition of learnability that doesn't depend on a given sample, and for this, we introduce the *true error*. We are now considering probability distributions over $\mathbb{N} \times \{0,1\}$, i.e. over input-label pairs. The idea is that we want to find a hypothesis class and a learner that behave well for any such distribution, but we don't know the actual distribution of the data we are studying (otherwise, there would be no point in learning).

Definition 25. The *true error* (or *risk*) of $h : \mathbb{N} \to \{0,1\}$ for the probability distribution $\mathcal{D}$ over $\mathbb{N} \times \{0,1\}$ is:

$$L_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y].$$

Let us decompose this error to better understand it. Notice that given a distribution $\mathcal{D}$ over $\mathbb{N} \times \{0,1\}$, the best hypothesis is its *Bayes optimal predictor* $h_{\mathcal{D}} : \mathbb{N} \to \{0,1\}$ given by $h_{\mathcal{D}}(n) = 1$ if $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = 1 \mid x = n] > 1/2$ and $h_{\mathcal{D}}(n) = 0$ otherwise. Of course, we cannot use this predictor since we do not know $\mathcal{D}$. Now, fix some hypothesis class $\mathcal{H}$ and some $h_0 \in \mathcal{H}$. The true error of h_0 can be decomposed as follows:

$$L_{\mathcal{D}}(h_0) = \varepsilon_{\text{opt}} + \varepsilon_{\text{app}} + \varepsilon_{\text{est}},$$

where:

- $\varepsilon_{\text{opt}} = L_{\mathcal{D}}(h_{\mathcal{D}})$ is the minimal yet inevitable error: it stems from the distribution when it doesn't provide a unique label for a given input. We cannot do anything about it.
- $\varepsilon_{\text{app}} = \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - \varepsilon_{\text{opt}}$ is the approximation error: it stems from the hypothesis class when it does not contain the Bayes optimal predictor.
- $\varepsilon_{\text{est}} = L_{\mathcal{D}}(h_0) - \varepsilon_{\text{app}}$ is the estimation error: it stems from the chosen hypothesis when it is not the best one of the hypothesis class.

Restricting $\mathcal{H}$ decreases ε_{est} (since it is then easier to choose the "right" hypothesis in the class) but increases ε_{app}. **Overfitting** happens when $\mathcal{H}$ is too large, while **underfitting** happens when it is too small. To characterize "good" hypothesis classes, we want to select those such that the estimation error can be made as small as needed, with a probability as high as needed, provided the sample is large enough and in a uniform way (independently of the distribution). This is exactly the definition of *probably approximately correct learning*, or *PAC learning*.

Definition 26 (PAC learnability)**.** We say that $A : \mathcal{S} \to \mathcal{H}$ *PAC learns* $\mathcal{H}$ if for every $\varepsilon, \delta > 0$, there exists $N \in \mathbb{N}$ such that for every $n \geqslant N$, and for every probability distribution $\mathcal{D}$ over $\mathbb{N} \times \{0,1\}$:

$$\mathbb{P}_{S\sim\mathcal{D}^n}\left[L_{\mathcal{D}}(A(S)) \leqslant \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon\right] \geqslant 1 - \delta,$$

where $S \sim \mathcal{D}^n$ means that the n elements of S are i.i.d. according to $\mathcal{D}$.

A hypothesis class $\mathcal{H}$ is called *PAC learnable* if there exists such an A.

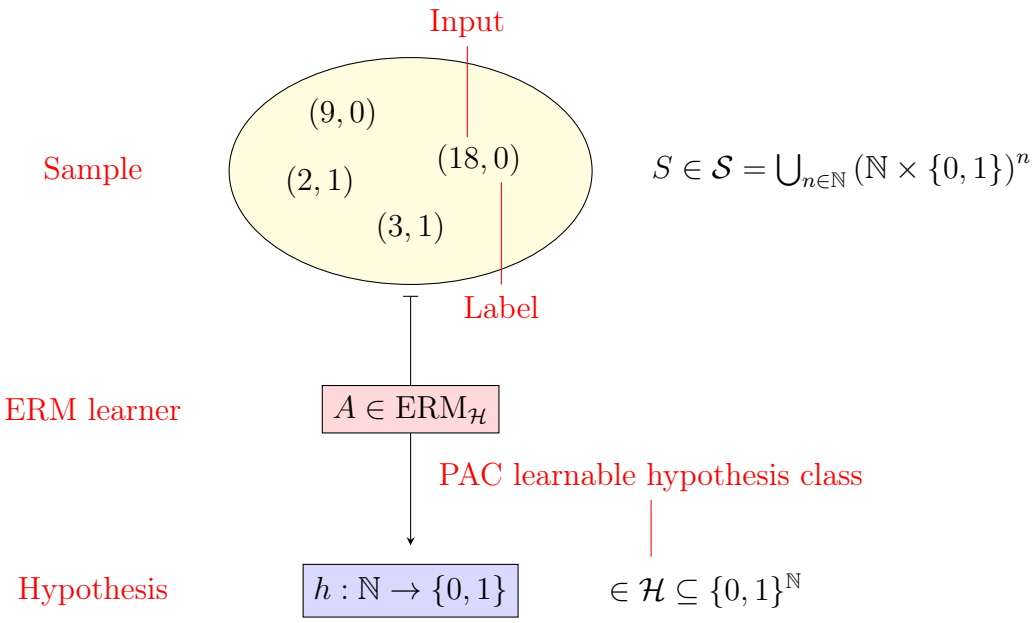

Figure 2: PAC learning main vocabulary

Note that the PAC learning paradigm is quite constraining. There are other options in machine learning which provide fewer guarantees but can be applied to a wider range of hypothesis class. Here, we have chosen to focus on PAC learnability.

It is possible to show that any finite class is PAC learnable, while the whole Cantor space $2^{\mathbb{N}}$ is not PAC learnable. This means that there is no way to find a "universal" PAC learner. Intuitively, this is because the learner is never able to gather enough information from the training set to fully characterize the distribution that has produced it.

Figure 2 illustrates the notions we have seen so far. We now want to come up with a precise characterization of PAC learnable hypothesis classes.

3.3 Characterization of learnable classes

It turns out that the correct metric to characterize the PAC learnability of a hypothesis class is its *VC dimension*. The VC dimension is often the number of natural numbers one needs to describe an element of the hypothesis class. More formally, the VC dimension of $\mathcal{H}$ is the highest possible cardinality for a subset of $\mathbb{N}$ such that any labelling function on this subset is a restriction of a function of $\mathcal{H}$.

Definition 27. Let $\mathcal{H} \subseteq 2^{\mathbb{N}}$ be a hypothesis class and $X \subseteq \mathbb{N}$ be finite. We say that $\mathcal{H}$ shatters X if for any $f : X \to \{0,1\}$, there exists $h \in \mathcal{H}$ such that $h|_X = f$.

Definition 28 (VC dimension)**.** Let $\mathcal{H} \subseteq 2^{\mathbb{N}}$ be a hypothesis class. The *VC dimension* of $\mathcal{H}$ is

$$\mathrm{VCdim}(\mathcal{H}) = \sup\{|X| \mid X \subseteq \mathbb{N} \text{ finite}, \mathcal{H} \text{ shatters } X\} \in \mathbb{N}_\infty.$$

The following statement is a direct consequence of the definition.

Lemma 29 (VC dimension is non-decreasing)**.** *Let $\mathcal{H}, \mathcal{H}' \subseteq 2^{\mathbb{N}}$ be such that $\mathcal{H} \subseteq \mathcal{H}'$. Then,* $\mathrm{VCdim}(\mathcal{H}) \leqslant \mathrm{VCdim}(\mathcal{H}')$.

Proof. Any finite $X \subseteq \mathbb{N}$ shattered by $\mathcal{H}$ is shattered by $\mathcal{H}'$ as well. □

We can now state the fundamental theorem of PAC learning.

Theorem 30 (Fundamental theorem of PAC learning)**.** *Let $\mathcal{H}$ be a hypothesis class. Then the following are equivalent:*

- *$\mathcal{H}$ is PAC learnable.*
- *Any* $\mathrm{ERM}_{\mathcal{H}}$ *is a PAC learner for $\mathcal{H}$.*
- $\mathrm{VCdim}(\mathcal{H}) < \infty$*.*

A full proof is available in [Shalev-Shwartz and Ben-David, 2014], chapters 4 and 6. Even if not detailed here, there is also an upper bound on the minimal N we need in Definition 26 which depends on ε, δ, and $\mathrm{VCdim}(\mathcal{H})$.

This result tells us that it is interesting to exactly compute the VC dimension and not only to know whether it is finite. Furthermore, we also know that finding an ERM is sufficient to get a PAC learner, and it is a notion whose definition is easier to manipulate since it does not involve probability distributions. In the next two sections, we will therefore focus on computing the VC dimension and an ERM of a hypothesis class.

4 Computing the VC dimension

4.1 Representation of closed hypothesis classes

Because we want to consider problems whose input is a hypothesis class, we need to find a way to represent such classes. The issue is that the power set $\mathcal{P}(2^{\mathbb{N}})$ has a higher cardinality than the set of infinite words and therefore cannot be represented. We need to find a restriction to a reasonable set of hypothesis classes, such that considering only these hypotheses doesn't make us lose expressivity.

An immediate idea is to consider only open or closed subsets. As we have seen in subsection 2.1, any open set can be described by a union of cylinders (that is of prefixes), and this union is necessarily countable.

Choosing open subsets doesn't seem interesting, as these are either empty or have an infinite VC dimension. Indeed, the cylinder $w2^{\mathbb{N}}$ shatters $[\![|w|, |w| + n]\!]$ for any $n \in \mathbb{N}$.

In contrast, closed sets seem more suitable. We can first notice the following statement.

Proposition 31. *For any $\mathcal{H} \subseteq 2^{\mathbb{N}}$, $\mathrm{VCdim}(\mathcal{H}) = \mathrm{VCdim}(\overline{\mathcal{H}})$.*

Proof. The inequality $\mathrm{VCdim}(\mathcal{H}) \leqslant \mathrm{VCdim}(\overline{\mathcal{H}})$ directly follows from Lemma 29. For the other direction, let $X \subseteq \mathbb{N}$ be a finite set shattered by $\overline{\mathcal{H}}$, and $f : X \to \{0, 1\}$. We have some $h \in \overline{\mathcal{H}}$ with $h|_X = f$. Because h is in the closure of $\mathcal{H}$, there exists a sequence $(h_n) \in \mathcal{H}^{\mathbb{N}}$ converging to h. In particular, there is some $N \in \mathbb{N}$ such that $h_N|_{\max X} = h|_{\max X}$, so $h_N|_X = f$, and $\mathcal{H}$ therefore shatters X as well. Thus, $\mathrm{VCdim}(\mathcal{H}) \geqslant \mathrm{VCdim}(\overline{\mathcal{H}})$. □

This means that if we wanted to consider a hypothesis class that would happen not to be closed, we could consider its closure instead and it would not affect its PAC learnability. Moreover, adding limit points does not appear to have a major impact in foreseeable cases. Let us say we assume that the positive instances of our learning problem are those above and below set thresholds. That means we would consider the hypothesis class

$$\mathcal{H} = \{\mathbb{1}_{[\![a,b]\!]} \mid a < b \in \mathbb{N}\}.$$

Its closure is $\overline{\mathcal{H}} = \mathcal{H} \cup \{\hat{0}\} \cup \{0^a\hat{1} \mid a \in \mathbb{N}\}$, meaning we are also allowing the interval to be empty or the upper threshold not to exist. This doesn't seem to be a problem, since these additional hypotheses are very unlikely to be chosen by the learner if the sample set has the expected form. We therefore elect to consider **only closed hypothesis classes**, i.e., closed subsets of the Cantor space.

The set of all closed sets over the Cantor space $2^{\mathbb{N}}$ is denoted[1] by $\mathscr{A}(2^{\mathbb{N}})$. It turns out that there are two natural ways of representing such sets. The first one is to see them as complements of open sets, which can themselves be described as countable unions of cylinders. The prefixes of these cylinders are those which cannot occur in the elements of the closed set that we describe. This is called *negative information*.

In order to properly represent prefixes, let $\nu : \mathbb{N} \to 2^*$ be a computable bijective encoding of finite bit sequences.

Definition 32 (Representation with negative information)**.** The representation $\delta_{\mathscr{A}_-(2^{\mathbb{N}})} :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathscr{A}(2^{\mathbb{N}})$ of the closed sets of the Cantor space using negative information is defined by:

$$\delta_{\mathscr{A}_-(2^{\mathbb{N}})}(p) = A \iff \{\nu(i-1) \mid i \in \mathrm{range}(p) \setminus \{0\}\} = \{w \in 2^* \mid w2^{\mathbb{N}} \cap A = \varnothing\}.$$

This means that a closed set is represented by the list of the prefixes that cannot be extended to a sequence of this set. Because this list may be finite or even empty, we shift the indices by 1, so that for instance, $\hat{0}$ can represent the entire space $2^{\mathbb{N}}$.

[1] Where $\mathscr{A}$ stands for "abgeschlossen" (closed in German).

Remark 33. The fact that we need all "forbidden prefixes" would be problematic if the Cantor space $2^{\mathbb{N}}$ was not locally compact. For the Baire space, defining the negative representation this way would not work since if an open set is described for instance as the union of all $05n\mathbb{N}^{\mathbb{N}}$, $n \in \mathbb{N}$, there would be no computable way to detect that the prefix 05 should be in the negative information of the complement too. For the Cantor space, this is not an issue since the numbers of extensions to detect before concluding that a prefix is forbidden is finite.

Instead of receiving negative information, we can also describe closed sets with *positive information*, i.e., a list of prefixes which can be extended to an infinite word of the closed set.

Definition 34 (Representation with positive information)**.** We define $\delta_{\mathscr{A}_+(2^{\mathbb{N}})} :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathscr{A}(2^{\mathbb{N}})$ by:

$$\delta_{\mathscr{A}_+(2^{\mathbb{N}})}(p) = A \iff \{\nu(i-1) \mid i \in \operatorname{range}(p) \setminus \{0\}\} = \{w \in 2^* \mid w2^{\mathbb{N}} \cap A \neq \varnothing\}.$$

Note that these representations contain a lot of redundant information. For instance, if a prefix w is included in the negative representation, it will also be the case for all prefixes extending it. Including them doesn't harm though, since an incomplete list can be completed in a computable way (given Remark 33 and the fact that even if there are countably many prefixes to add, we can parallelize the writing process).

We can also combine positive and negative information:

Definition 35 (Representation with full information)**.** The representation $\delta_{\mathscr{A}_\pm(2^{\mathbb{N}})} :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathscr{A}(2^{\mathbb{N}})$ is defined by:

$$\delta_{\mathscr{A}_\pm(2^{\mathbb{N}})}\langle p, q\rangle = A \iff \delta_{\mathscr{A}_+(2^{\mathbb{N}})}(p) = \delta_{\mathscr{A}_-(2^{\mathbb{N}})}(q) = A.$$

We usually write $\mathscr{A}_+(2^{\mathbb{N}})$, $\mathscr{A}_-(2^{\mathbb{N}})$ or $\mathscr{A}_\pm(2^{\mathbb{N}})$ to indicate the representation we endow the set with.

In order to compute the VC dimension, we also need a representation for the set $\mathbb{N}_\infty$, which the VC dimension belongs to.

Definition 36 (Representation of $\mathbb{N}_\infty$)**.** We define $\delta_{\mathbb{N}_\infty} :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}_\infty$ by $\delta_{\mathbb{N}_\infty}(p) = \infty \iff p = \hat{1}$ and $\delta_{\mathbb{N}_\infty}(p) = n \iff p = 1^n\hat{0}$ for $n \in \mathbb{N}$.

This representation is admissible for the topology on $\mathbb{N}_\infty$ generated by the singletons $\{n\}$ and the intervals $[\![n, \infty[\![$, $n \in \mathbb{N}$.

We can finally formulate the problem of computing the VC dimension. We need to distinguish between three cases, depending on the information we receive about the closed set we want to compute the VC dimension.

Definition 37. Let $*$ be $+$, $-$, or $\pm$. We define the problem:

$$\begin{array}{rccc} \mathrm{VCdim}_{\mathscr{A}_*(2^{\mathbb{N}})} : & \mathscr{A}_*(2^{\mathbb{N}}) & \to & \mathbb{N}_\infty \\ & \mathcal{H} & \mapsto & \mathrm{VCdim}(\mathcal{H}) \end{array}$$

4.2 Around the SORT problem

We introduce a few problems to compare them with the VC dimension. We represent the Baire and the Cantor spaces with trivial identity functions.

Definition 38 (Benchmark problems)**.** Let X be either $\mathbb{N}$ or $\mathbb{N}_\infty$.

- The problem SORT : $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is defined by $\mathrm{SORT}(p) = \hat{0}$ if p contains infinitely many zeroes, and $\mathrm{SORT}(p) = 0^n\hat{1}$ if p contains n zeroes.

- The problem $\lim_X :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ is defined by $\lim_X(p) = \ell \iff p(i) \underset{i\to\infty}{\longrightarrow} \ell$ for $\ell \in X$.
- The problem $\lim_X^{\nearrow} :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ is defined by, for $\ell \in X$, $\lim_{\mathbb{N}_\infty}^{\nearrow}(p) = \ell$ iff $(p(i))_i$ is a non-decreasing sequence and $\lim_{i\to\infty} p(i) = \ell$.

Proposition 39. *We have* $\mathrm{SORT} \equiv_{\mathrm{sW}} \lim_{\mathbb{N}_\infty}^{\nearrow}$.

Proof. $\boxed{\mathrm{SORT} \leqslant_{\mathrm{sW}} \lim_{\mathbb{N}_\infty}^{\nearrow}}$ Given a sequence $p \in 2^{\mathbb{N}}$, produce a sequence $q \in \mathbb{N}^{\mathbb{N}}$ by setting $q(0) = 1-p(0)$ and, for $i \in \mathbb{N}$, $q(i+1) = q(i)$ if $p(i+1) = 1$ and $q(i+1) = q(i)+1$ if $p(i+1) = 0$. This way, $q(i)$ is the number of zeroes in $p|_{i+1}$, and $(q(i))$ is a non-decreasing sequence.

Then, apply $\lim_{\mathbb{N}_\infty}^{\nearrow}$ on q to get the number of zeroes in $\mathrm{SORT}(p)$. It is then easy to produce $\mathrm{SORT}(p)$ by flipping zeroes and ones.

$\boxed{\lim_{\mathbb{N}_\infty}^{\nearrow} \leqslant_{\mathrm{sW}} \mathrm{SORT}}$ Given a non-decreasing sequence $p \in \mathbb{N}^{\mathbb{N}}$, first write $p(0)$ zeroes. Then, for every read symbol $p(i)$ $(i \geqslant 1)$, write a one followed by $p(i) - p(i-1)$ zeroes. The resulting sequence q is such that $q|_i$ contains $p(i)$ zeroes, and as p is non-decreasing, q contains $\lim p(i)$ zeroes. Apply SORT to q and then flip zeroes and ones to produce a name of $\lim_{\mathbb{N}_\infty}^{\nearrow}(p)$. □

Proposition 40. *We have however* $\lim_{\mathbb{N}_\infty}^{\nearrow} <_{\mathrm{W}} \lim_{\mathbb{N}_\infty}$.

Remark 41. This result is surprising, since the very same problem with only finite limits is Weihrauch equivalent to its non-decreasing variation, i.e., $\lim_{\mathbb{N}} \equiv_{\mathrm{W}} \lim_{\mathbb{N}}^{\nearrow}$. Indeed, given a converging sequence $(p_n) \in \mathbb{N}^{\mathbb{N}}$, we can construct a sequence (q_n) so that the sequence of codes $(\langle p_n, q_n\rangle)_n$ is non-decreasing. Then we can get the limit of that sequence, and deduce that of (p_n). This last step would fail with $\mathbb{N}_\infty$ since after reading many ones, a machine can not know if it is reading $\hat{1}$ (a name of ∞) and should start writing a 1, or if it reads a code of the form $\langle 0, N\rangle$ (with N big). In other words, the decoding process is discontinuous.

In order to prove Proposition 40, let's first show the following lemma:

Lemma 42. *We have* $\lim_{\mathbb{N}}^{\nearrow} <_{\mathrm{W}} \lim_{\mathbb{N}_\infty}^{\nearrow}$.

In other words, allowing the limit to be infinite makes the problem strictly harder.

Proof of Lemma 42. Clearly, $\lim_{\mathbb{N}}^{\nearrow} \leqslant_{\mathrm{sW}} \lim_{\mathbb{N}_\infty}^{\nearrow}$. Assume $\lim_{\mathbb{N}_\infty}^{\nearrow} \leqslant_{\mathrm{W}} \lim_{\mathbb{N}}^{\nearrow}$ and let $H, K :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ be like in Definition 13.

We will construct an increasing (for the $\sqsubseteq$ relation) sequence $(w_n) \in (2^*)^{\mathbb{N}}$ such that, for any $n \in \mathbb{N}$, $w_n \sqsubset w_{n+1}$, $w_{n+1} \sqsubseteq w_n\hat{n}$, and:

- The image of $w_n\hat{n}$ under K contains a symbol equal to $\lim_{i\to\infty} K(w_n\hat{n})(i)$ which only depends on the prefix w_{n+1},
- The output of H fed with $w_n\hat{n}$ and the limit of $K(w_n\hat{n})$ only depends on that limit and the prefix w_{n+1}.

We construct (w_n) the following way:

- We choose $w_0 = \varepsilon$.
- Let $n \in \mathbb{N}$. Denote by u the prefix of $w_n\hat{n}$ required to write the first symbol of $K(w_n\hat{n})$ with value $\lim_{i\to\infty} K(w_n\hat{n})(i)$ and by v the prefix of $w_n\hat{n}$ required to write $1^{\lim_{i\to\infty}(w_n\hat{n})(i)}0$ in the output of H. Note that both prefixes exist as K and H are continuous. Then choose w_{n+1} as the longest prefix of $w_n\hat{n}$ between u and v, extending it if necessary so $w_n \sqsubset w_{n+1}$.

By induction, let's now show that for any $n \in \mathbb{N}$, $\lim K(w_n\hat{n}) \geqslant n$. Clearly, $\lim K(w_0\hat{0}) \geqslant 0$. For $n \in \mathbb{N}$, assume $\lim K(w_n\hat{n}) \geqslant n$. Because H outputs 1^n0 on the prefix $w_n \sqsubseteq w_{n+1}\widehat{n+1}$ and the value of $\lim K(w_n\hat{n})$, we have necessarily $\lim K(w_{n+1}\widehat{n+1}) \neq \lim K(w_n\hat{n})$, and therefore, as $(K(w_{n+1}\widehat{n+1})(i))_i$ is non-decreasing and contain a $\lim K(w_n\hat{n})$ symbol, $\lim K(w_{n+1}\widehat{n+1}) \geqslant \lim K(w_n\hat{n}) + 1 \geqslant n+1$, which completes the induction.

Now, as (w_n) is increasing for $\sqsubseteq$, we can define $p \in \mathbb{N}^{\mathbb{N}}$ as the only sequence which extends all w_n. Then $p(i) \underset{i\to\infty}{\longrightarrow} \infty$, but by construction, $K(p)$ contains an integer $\geqslant n$ for any n, meaning that $K(p)$ cannot converge and is therefore not in the domain of $\lim^{\nearrow}_{\mathbb{N}}$. We arrive at a contradiction. □

Proof of Proposition 40. It is clear that $\lim^{\nearrow}_{\mathbb{N}_\infty} \leqslant_{\mathrm{sW}} \lim_{\mathbb{N}_\infty}$. Assume the reciprocal reduction holds, that is there exist computable $H, K :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ so that for any realizer G of $\lim^{\nearrow}_{\mathbb{N}_\infty}$, $H\langle \mathrm{Id}_{\mathbb{N}^{\mathbb{N}}}, GK\rangle \vdash \lim_{\mathbb{N}_\infty}$.

First, by transitivity and from Lemma 42, we know that $\lim^{\nearrow}_{\mathbb{N}} <_{\mathrm{W}} \lim_{\mathbb{N}_\infty}$, so there exists $p \in \mathbb{N}^{\mathbb{N}}$ such that $K(p)(i) \underset{i\to\infty}{\longrightarrow} \infty$ (otherwise $\lim_{\mathbb{N}_\infty}$ would be reducible to $\lim^{\nearrow}_{\mathbb{N}}$).

Now, if $\lim p > 0$ (resp. $\lim p = 0$), as H is continuous, the first one (resp. zero) of the output only depends on finitely many symbols of p, as well as a finite number of ones (say N) in the name of $\lim K(p) = \infty$. Because $(K(p)(i))_i$ is non-decreasing, the fact that its limit is greater than or equal to N also only depends on finitely many symbols of p. Therefore, it is sufficient to change all others symbols of p to 0 (resp. 1) to arrive at a contradiction. □

4.3 Weihrauch complexity of $\mathrm{VCdim}_{\mathscr{A}_+(2^{\mathbb{N}})}$ and $\mathrm{VCdim}_{\mathscr{A}_\pm(2^{\mathbb{N}})}$

Let us state our first theorem regarding the VC dimension.

Theorem 43. *We have* $\mathrm{VCdim}_{\mathscr{A}_+(2^{\mathbb{N}})} \equiv_{\mathrm{W}} \mathrm{VCdim}_{\mathscr{A}_\pm(2^{\mathbb{N}})} \equiv_{\mathrm{W}} \mathrm{SORT}$.

Proof. We use Proposition 39 to compare these problems with $\lim^{\nearrow}_{\mathbb{N}_\infty}$ as well.

$\boxed{\mathrm{VCdim}_{\mathscr{A}_+(2^{\mathbb{N}})} \leqslant_{\mathrm{W}} \lim^{\nearrow}_{\mathbb{N}_\infty}}$ Given $p \in \mathbb{N}^{\mathbb{N}}$ a name for $\mathcal{H} \in \mathscr{A}_+(2^{\mathbb{N}})$, read for $i \in \mathbb{N}$ the prefixes in $p|_i$ and denote by $q(i)$ the cardinal of the largest subset of $\mathbb{N}$ that is shattered by the read prefixes. This defines a non-decreasing sequence $(q(i))$.

Any finite set $X \subseteq \mathbb{N}$ that is shattered by $\mathcal{H}$ will eventually be discovered once all prefixes of length $\max X + 1$ in the positive information have been read. Therefore, $q(i) \underset{i\to\infty}{\longrightarrow} \mathrm{VCdim}(\mathcal{H})$.

Now, apply $\lim^{\nearrow}_{\mathbb{N}_\infty}$ on the sequence q: the result is a name of $\mathrm{VCdim}_{\mathscr{A}_+(2^{\mathbb{N}})}(\mathcal{H})$.

$\boxed{\mathrm{SORT} \leqslant_{\mathrm{W}} \mathrm{VCdim}_{\mathscr{A}_\pm(2^{\mathbb{N}})}}$ Given a string $p \in \mathbb{N}^{\mathbb{N}}$, consider the hypothesis class

$$\mathcal{H} = \{h : \mathbb{N} \to \{0,1\} \mid \forall i \in \mathbb{N},\ p(i) = 1 \implies h(i) = 0\}.$$

Observe that the VC dimension of $\mathcal{H}$ is the number of indices over which the functions of $\mathcal{H}$ can take both values 0 and 1, which by definition is exactly the numbers of zeroes in p.

To produce a name of $\mathcal{H}$, iterate over all natural numbers $i \in \mathbb{N}$, starting from $i = 0$. For each i, the extendable prefixes of length $i+1$ are those of the form $a_0 \dots a_i \in 2^{i+1}$ respecting the constraint $a_k = 0$ for all $k \leqslant i$ such that $p(k) = 1$. Read $p|_i$, identify these prefixes and write them in the positive information, then write all other prefixes of length $i+1$ in the negative information.

Then, apply $\mathrm{VCdim}_{\mathscr{A}_\pm(2^{\mathbb{N}})}$ to get the VC dimension of $\mathcal{H}$, which is the number of zeroes in p. Flip zeroes and ones to write $\mathrm{SORT}(p)$.

$\boxed{\mathrm{VCdim}_{\mathscr{A}_\pm(2^{\mathbb{N}})} \leqslant_{\mathrm{W}} \mathrm{VCdim}_{\mathscr{A}_+(2^{\mathbb{N}})}}$ Given $\langle p, q\rangle$, simply apply $\mathrm{VCdim}_{\mathscr{A}_+(2^{\mathbb{N}})}$ on p and discard the negative information q. □

There are two important conclusions to draw from this result. First, computing the VC dimension of a closed set described by its positive information is equivalent to sorting an infinite sequence. Second, the negative information doesn't provide us with anything useful here, since including it doesn't change the Weihrauch complexity.

Intuitively, the irrelevance of the negative information can be understood in the following way: this information can help us know that a certain subset is not shattered by the hypothesis class we consider. But we cannot conclude anything about the VC dimension based on that, since a subset of the same size could very well be shattered but simply involve much higher (and still unexplored) indices.

4.4 Weihrauch complexity of $\mathrm{VCdim}_{\mathscr{A}_-(2^{\mathbb{N}})}$

Let us now see what happens if we only include the negative information. Surprisingly (but consistently with the previous result), this information can not be exploited in a better way than translating it into positive information before using it.

Lemma 44 (Translation into positive information)**.** *The problem of translating the negative information of a closed set of the Cantor space into its positive information*

$$\begin{array}{rccc} \mathrm{CONVERT} & : \mathscr{A}_-(2^{\mathbb{N}}) & \to & \mathscr{A}_+(2^{\mathbb{N}}) \\ & A & \mapsto & A \end{array}$$

is strongly Weihrauch reducible to the $\lim_{\mathbb{N}^{\mathbb{N}}}$ *problem:* $\mathrm{CONVERT} \leqslant_{\mathrm{sW}} \lim_{\mathbb{N}^{\mathbb{N}}}$.

Proof. Given $p \in \mathbb{N}^{\mathbb{N}}$ with $\delta_{\mathscr{A}_-(2^{\mathbb{N}})}(p) = A$, produce a sequence $(q_n) \in (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}}$ the following way: let $q_0 = \hat{0}$ and, for $n \in \mathbb{N}$, define q_{n+1} by extending q_n with all indices of prefixes of length n, and then, for any $i \in \mathbb{N}$ such that $q_{n+1}(i) \in \mathrm{range}(p|_n) \setminus \{0\}$, replace $q_{n+1}(i)$ with 0. This way, $\langle q_0, q_1, \dots \rangle$ can be computed out of p, and we claim that (q_n) converges to some $q \in \mathbb{N}^{\mathbb{N}}$ such that $\delta_{\mathscr{A}_+(2^{\mathbb{N}})}(q) = A$.

That (q_n) converges follows from the fact that for any $i \in \mathbb{N}$, the value at index i can change at most twice. For any $k \in \mathbb{N}$ such that $\nu(k)2^{\mathbb{N}} \cap A \neq \varnothing$, $k+1$ is included in the sequence range at step $n = |\nu(k)|$, and never removed. For any $k \in \mathbb{N}$ such that $\nu(k)2^{\mathbb{N}} \cap A \neq \varnothing$, $k+1$ may be included, but is removed at some point, as it is in the range of p. Therefore, $\delta_{\mathscr{A}_+(2^{\mathbb{N}})}(\lim q_n) = A$.

We can then apply $\lim_{\mathbb{N}^{\mathbb{N}}}$ on $\langle q_0, q_1, \dots \rangle$ to get a valid name of $\mathrm{CONVERT}(A)$. □

This result motivates us to consider the jump (Definition 17) of SORT.

Proposition 45. *We have* $\mathrm{VCdim}_{\mathscr{A}_-(2^{\mathbb{N}})} \leqslant_{\mathrm{sW}} \mathrm{SORT}'$.

Proof. Given $p \in \mathbb{N}^{\mathbb{N}}$ with $\delta_{\mathscr{A}_-(2^{\mathbb{N}})}(p) = \mathcal{H}$, produce $\langle q_0, q_1, \dots \rangle$ like in the proof of Lemma 44 and feed it into a realizer of $\mathrm{VCdim}'_{\mathscr{A}_+(2^{\mathbb{N}})}$. As (q_n) converges to a $\delta_{\mathscr{A}_+(2^{\mathbb{N}})}$-name of $\mathcal{H}$, the ensuing result is a name of $\mathrm{VCdim}(\mathcal{H})$.

Therefore $\mathrm{VCdim}_{\mathscr{A}_-(2^{\mathbb{N}})} \leqslant_{\mathrm{sW}} \mathrm{VCdim}'_{\mathscr{A}_+(2^{\mathbb{N}})} \equiv_{\mathrm{sW}} \mathrm{SORT}'$, using Proposition 18. □

To finish classifying the VC dimension, we need to show that SORT' is Weihrauch reducible to computing a VC dimension from negative information.

Theorem 46. *We have* $\mathrm{SORT}' \leqslant_{\mathrm{sW}} \mathrm{VCdim}_{\mathscr{A}_-(2^{\mathbb{N}})}$*, and therefore* $\mathrm{VCdim}_{\mathscr{A}_-(2^{\mathbb{N}})} \equiv_{\mathrm{sW}} \mathrm{SORT}'$.

Proof. Let $(p_n) \in (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}}$ be a sequence converging to $p \in 2^{\mathbb{N}}$, and let $\ell \in \mathbb{N}_\infty$ denote the number of zeroes in p. We want to compute the negative information of a hypothesis class $\mathcal{H}$ whose VC dimension is ℓ. We'll perform the construction so that for any $h \in \mathcal{H}$ and $n \in \mathbb{N}$, $h(n) = 0$, except for certain n, which we refer to as *open paths*, where $h(n)$ can take both values 0 and 1. We want the number of open paths to be ℓ, so that we would have $\mathrm{VCdim}(\mathcal{H}) = \ell$.

To do that, let's keep track of a variable `open_paths` containing a list of elements of $\mathbb{N} \times \mathbb{N}$. For $(i, k) \in \mathbb{N} \times \mathbb{N}$ in this list, i denotes an index which we keep open, while k denotes the position of the zero in (p_n) we associate with i. Also keep a variable `N`, initialized at 0, which is the latest index we have handled in the description of $\mathcal{H}$.

Now, to construct $q \in \mathbb{N}^{\mathbb{N}}$ such that $\delta_{\mathscr{A}_-(2^{\mathbb{N}})}(q) = \mathcal{H}$, repeat the following steps:

- Read $p_{\mathtt{N}|_{\mathtt{N}+1}}$.
- For any $k \leqslant \mathtt{N}$ such that $p_{\mathtt{N}}(k) = 0$ and there is no $i \in \mathbb{N}$ with (i, k) in `open_paths`, add an open path for k, that is add $(\mathtt{N}, k)$ to `open_paths`, and then increment `N`.
- For any $k \leqslant \mathtt{N}$ such that $p_{\mathtt{N}}(k) \neq 0$ with some $i \in \mathbb{N}$ with (i, k) in `open_paths` (i.e., k was "wrongly left open"), remove (i, k) from `open_paths`.
- Increment `N` by one so that one additional path is closed.
- Effectively close all paths that have not been left open at this point, that is write all $m \in \mathbb{N}$ in q such that $|\nu(m-1)| \leqslant \mathtt{N}+1$ **and** there is $i \leqslant \mathtt{N}$ with $\nu(m-1)(i) = 1$ and yet no k such that (i, k) is in `open_paths`.

Note that we have written the negative information of a closed set $\mathcal{H} \subseteq 2^{\mathbb{N}}$ so that $h \in \mathbb{N}$ iff $h(i) = 0$ for all i that are either never added in `open_paths`, or added and then removed. Indeed, any prefix violating this condition is added to the negative information, while no other prefix is added, because prefixes of length $\mathtt{N}+1$ are only written in the negative information **after** the index $k = \mathtt{N}$ is added to `open_paths`.

Moreover, for any $k \in \mathbb{N}$, there is either zero or one $i \in \mathbb{N}$ such that (i, k) indefinitely remains in `open_paths`, and there is one iff $p(k) = 0$. Indeed, if $p(k) = 0$ holds, there is a minimal $N \in \mathbb{N}$ such that $p_n(k) = 0$ for all $n \geqslant N$, and k is added when $p_N(k)$ is read and then never removed. If $p(k) = 1$, any addition of k to `open_paths` is eventually cancelled.

Therefore, once q is produced, it can be fed into a realizer of $\mathrm{VCdim}_{\mathscr{A}_-(2^{\mathbb{N}})}$, and then we can just flip the zeroes and ones in the output to produce a name for the outcome of SORT′. □

The figure 3 in the conclusion summarizes all our results.

5 Computing ERMs and PAC learners

Now that we know how to compute the VC dimension, we can consider the problem of computing the learner itself if the VC dimension if finite (i.e., the hypothesis class is PAC learnable). Because any ERM is a learner in that case and the definition of an ERM is less complex than that of a learner, the first natural question is the complexity of computing an ERM.

5.1 Weihrauch complexity of the ERM problem

Recall that given a hypothesis class $\mathcal{H}$, an $\mathrm{ERM}_{\mathcal{H}}$ is a function $A : \mathcal{S} \to \mathcal{H}$ which maps any sample to a hypothesis $h \in \mathcal{H}$ minimizing the empirical risk $L_S(h)$. There are therefore two natural ways of defining the ERM problem in computable analysis: we can either consider $\mathcal{H}$ as the input and expect a function $\mathcal{S} \to \mathcal{H}$ as the output (basically an element of $(2^{\mathbb{N}})^{\mathcal{S}}$), or take a pair $(\mathcal{H}, S)$ as the input and expect an output in $2^{\mathbb{N}}$ which is an $\operatorname{argmin}_{h\in\mathcal{H}} L_S(h)$.

Note that the set $\mathcal{S}$ is easy to represent since it is countable (for instance, we can represent $((x_1, y_1), \dots, (x_n, y_n)) \in \mathcal{S}$ as $(\langle x_1, y_1\rangle + 1) \dots (\langle x_n y_n\rangle + 1)\hat{0} \in \mathbb{N}^{\mathbb{N}}$). If $\mu : \mathbb{N} \to \mathcal{S}$ is a computable bijection, we can also represent $A \in (2^{\mathbb{N}})^{\mathcal{S}}$ as $\langle A(\mu(0)), A(\mu(1)), \dots\rangle \in \mathbb{N}^{\mathbb{N}}$, making use of tupling functions (Definition 5).

Definition 47. Let $*$ be $+$, $-$, or $\pm$. We define the problems

$$\begin{array}{rccc} \mathrm{ERM}_{\mathscr{A}_*(2^{\mathbb{N}})} & : & \mathscr{A}_*(2^{\mathbb{N}}) \setminus \{\varnothing\} & \rightrightarrows & (2^{\mathbb{N}})^{\mathcal{S}} \\ & & \mathcal{H} & \mapsto & \mathrm{ERM}_{\mathcal{H}} \end{array}$$

and

$$\begin{array}{rccc} \mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_*(2^{\mathbb{N}})} & : & \big(\mathscr{A}_*(2^{\mathbb{N}}) \setminus \{\varnothing\}\big) \times \mathcal{S} & \rightrightarrows & 2^{\mathbb{N}} \\ & & (\mathcal{H}, S) & \mapsto & \operatorname{argmin}_{h\in\mathcal{H}} L_S(h) \end{array}$$

Clearly, the first problem is a "restricted parallelization" of the second one in the sense that we want the result for all samples, but the same hypothesis. This immediately yields the following lemma.

Lemma 48. *Let $*$ be $+$, $-$, or $\pm$. Then $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_*(2^{\mathbb{N}})} \leqslant_{\mathrm{W}} \mathrm{ERM}_{\mathscr{A}_*(2^{\mathbb{N}})} \leqslant_{\mathrm{W}} \widehat{\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_*(2^{\mathbb{N}})}}$.*

Our first result is the following:

Theorem 49. *The problem $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_\pm(2^{\mathbb{N}})}$ is computable, and so is therefore $\mathrm{ERM}_{\mathscr{A}_\pm(2^{\mathbb{N}})}$.*

Proof. Given a closed hypothesis class $\mathcal{H}$ for which we have the positive and negative information and a sample $S = ((x_1, y_1), \dots, (x_n, y_n))$, let $x_{\max}$ be the maximum index for which the sample gives a label, that is $x_{\max} = \max\{x \in \mathbb{N} \mid \exists i \in [\![1, n]\!],\ x = x_i\}$.

Notice that two hypotheses have the same empirical risk if they coincide over $[\![0, x_{\max}]\!]$. Therefore, read the positive and negative information until you know for all prefixes of length $x_{\max}+1$ (there are $2^{x_{\max}+1}$ of them) whether they can be extended or not. Compute the empirical risk for all extendable prefixes and choose the best of them. Then extend it to an infinite sequence with the positive information. □

Now, the question is naturally what happens when we are deprived of the positive or negative information. First, consider the $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})}$ problem. If we miss the negative information, the issue occurs in the first phase of the algorithm in the proof of Theorem 49, because we cannot tell anything if a prefix doesn't appear in the positive information (it may be in the negative information, but may as well show up later on in the positive information). We may need to change our mind if a better prefix appears. But we also know that there are up to 2^{N+1} extendable prefixes, which puts an upper bound on the number of mind changes which can be known from the beginning. Intuitively, we have to look for problems which have this property.

Definition 50 (Benchmark problems)**.**

- The $\mathrm{K}_{\mathbb{N}}$ problem is defined as follows:

$$\begin{array}{rcccc} \mathrm{K}_{\mathbb{N}} & : & \{(m,p) \in \mathbb{N} \times \mathbb{N}^{\mathbb{N}} \mid [\![0,m]\!] \not\subseteq \operatorname{range} p\} & \rightrightarrows & \mathbb{N} \\ & & (m,p) & \mapsto & [\![0,m]\!] \setminus \operatorname{range} p \end{array}$$

- The $\min_{\mathbb{N}}$ problem is defined as follows:

$$\begin{array}{rcccc} \min_{\mathbb{N}} & : & \mathbb{N}^{\mathbb{N}} & \to & \mathbb{N} \\ & & p & \mapsto & \min\{p(i) \mid i \in \mathbb{N}\} \end{array}$$

These problems are clearly discontinuous and therefore non-computable. They can only be computed when finitely many mind changes are allowed, but the machine is able to give from the beginning an upper bound of the number of mind changes it will actually need (at most m and $p(0)$, respectively). They are however not equivalent, as $\min_{\mathbb{N}}$ is harder than $\mathrm{K}_{\mathbb{N}}$[2], so $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})}$ should be equivalent to one of them.

It turns out (and I was initially going in the wrong direction) that $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})} \equiv_{\mathrm{W}} \min_{\mathbb{N}}$, which we are now going to prove.

Proposition 51. *We have* $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})} \leqslant_{\mathrm{W}} \min_{\mathbb{N}}$.

Proof. Given a hypothesis class $\mathcal{H} \in \mathscr{A}_+(2^{\mathbb{N}})$ for which we know the positive information and a sample $S = ((x_1, y_1), \dots, (y_1, y_n)) \in \mathcal{S}$, consider $x_{\max} = \max\{x \in \mathbb{N} \mid \exists i \in [\![1,n]\!],\ x = x_i\}$ like previously.

Compute the empirical risk associated with all prefixes of length $x_{\max}+1$ and sort them by ascending risk by defining a bijection $\mu : [\![0, 2^{x_{\max}+1} - 1]\!] \to 2^{x_{\max}+1}$.

Produce an entry for $\min_{\mathbb{N}}$ the following way: write in p the images under μ^{-1} of all prefixes of length $x_{\max}+1$ that appear in the positive information. There is at least one of them since $\mathcal{H} \neq \varnothing$ and we can repeat it to get an infinite sequence.

Now, apply $\min_{\mathbb{N}}$. The result is a prefix in the positive information with the lowest empirical risk. Extend it with the positive information to get an $\mathrm{ERM}_{\mathcal{H}}$. □

Theorem 52. *We have* $\min_{\mathbb{N}} \leqslant_{\mathrm{W}} \mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})}$ *and therefore* $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})} \equiv_{\mathrm{W}} \min_{\mathbb{N}}$.

Proof. Let p be an input for $\min_{\mathbb{N}}$. Choose the sample $S = ((i,0))_{i=0}^{p(0)}$ and the hypothesis class $\mathcal{H} = \{\mathbb{1}_{[\![0,i]\!]}\}_{i \in [\![0,p(0)]\!] \cap \operatorname{range} p}$.

This way, we have associated any $i \in [\![0,p(0)]\!]$ with a hypothesis that has an empirical risk of $\frac{i}{p(0)+1}$ for the sample S. The hypotheses that are effectively in $\mathcal{H}$ are those corresponding to the range of p. It is easy to write the positive information of $\mathcal{H}$ (just add the prefixes corresponding to any $i \in [\![0,p(0)]\!]$ that appears in p). Then, use a realizer of $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})}$ to get the best hypothesis, and count how many times it takes the value 1 over $[\![0,p(0)]\!]$. This is the minimum of p. □

Remark 53. This equivalence cannot be strong, since $\min_{\mathbb{N}}$ has countably many possible outputs, while $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})}$ has uncountably many of them.

Now, consider the global problem $\mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})}$. From Lemma 48, it is easier than the parallelization of $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_+(2^{\mathbb{N}})}$. We also know from [Brattka and Hertling, 2021] the following result.

[2]Actually, $\mathrm{K}_{\mathbb{N}} \equiv_{\mathrm{sW}} \mathrm{LLPO}^*$ and $\min_{\mathbb{N}} \equiv_{\mathrm{sW}} \mathrm{LPO}^*$, where LPO is finding whether a bit sequence contains a 1 while LLPO is finding whether a bit sequence with at most one 1 has only zeroes at odd or even positions, and * is a finite parallelization operator.

Proposition 54. *We have* $\widehat{\mathrm{LPO}} \equiv_{\mathrm{sW}} \widehat{\min}_{\mathbb{N}} \equiv_{\mathrm{sW}} \lim_{\mathbb{N}^{\mathbb{N}}}$*, where* $\mathrm{LPO} : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ *is defined by* $\mathrm{LPO}(p) = \hat{1}$ *if* p *contains a* 1 *and* $\mathrm{LPO}(p) = \hat{0}$ *otherwise.*

So it is sufficient to show that the parallelization of LPO is reducible to $\mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})}$ to have that $\mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})} \equiv_{\mathrm{W}} \lim_{\mathbb{N}^{\mathbb{N}}}$.

Theorem 55. *We have* $\widehat{\mathrm{LPO}} \leqslant_{\mathrm{W}} \mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})}$ *and therefore* $\mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})} \equiv_{\mathrm{W}} \lim_{\mathbb{N}^{\mathbb{N}}}$.

Proof. Let $\langle p_0, p_1, \dots \rangle$ be an input for $\widehat{\mathrm{LPO}}$. Consider the hypothesis class:

$$\mathcal{H} = \{h : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N},\ (\forall i \in \mathbb{N},\ p_n(i) = 0) \implies h(n) = 0\}.$$

It is possible to produce the positive information for $\mathcal{H}$ the following way: write prefixes that only allow a zero in positions corresponding to sequences where no 1 has been seen, and then add the prefixes allowing a one at position n when a 1 is read in p_n.

Use a realizer of $\mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})}$ to get an $\mathrm{ERM}_{\mathcal{H}}$. Then, to know if p_n contains a one, check the hypothesis h which the $\mathrm{ERM}_{\mathcal{H}}$ outputs on the sample $((n,1))$ and see whether $h(n) = 1$. It can only be the case if p_n contains a one, and if p_n does contain a one, the chosen hypothesis will have the property $h(n) = 1$ since it minimizes the empirical risk. □

A natural question is whether the classification is different if we assume that the VC dimension of the input is finite or even if we provide its value in the input. This is a fair question to ask since in the context of PAC learning, we are mainly interested in computing ERMs for learnable hypothesis classes. A reason to think the complexity may be different is that the $\widehat{\mathrm{LPO}}$ problem restricted to inputs that only contain finitely many sequences with a 1 is (weakly) reducible to $\lim^{\nearrow}_{\mathbb{N}}$. However, we can adjust the hypothesis class in the proof and choose for instance

$$\mathcal{H} = \hat{0} \cup \{\mathbb{1}_{\{n\}} \mid n \in \mathbb{N},\ \exists i \in \mathbb{N},\ p_n(i) = 1\}.$$

The VC dimension of this closed hypothesis class is 1, its positive information can still be produced, and the sample $((n,1))$ can still be used to know whether p_n contains a one. This means that even under the additional assumption that the VC dimension is 1, the $\mathrm{ERM}_{\mathscr{A}_+(2^{\mathbb{N}})}$ problem is equivalent to $\widehat{\mathrm{LPO}} \equiv_{\mathrm{W}} \lim_{\mathbb{N}^{\mathbb{N}}}$.

The classification of the ERM problem is therefore complete for the $\mathscr{A}_\pm(2^{\mathbb{N}})$ and $\mathscr{A}_+(2^{\mathbb{N}})$ cases. The problems $\mathrm{ERM}^{\mathrm{loc}}_{\mathscr{A}_-(2^{\mathbb{N}})}$ and $\mathrm{ERM}_{\mathscr{A}_-(2^{\mathbb{N}})}$ remain unclassified. An upper bound on their complexity is of course $\lim_{\mathbb{N}^{\mathbb{N}}}$ since this problem is sufficient to turn the negative information into positive information. There are mainly two challenges: finding the best prefix among those that are not listed (this is to classify), and then extend that prefix using negative information (this is equivalent to $\widehat{\mathrm{K}_{\mathbb{N}}}$).

5.2 The general PAC learner problem

The previous subsection was dedicated to computing an ERM. However, we are naturally interested in the more general problem of computing a PAC learner for a hypothesis class which is closed and has a finite VC dimension.

Definition 56. Let $*$ be $+$, $-$, or $\pm$. We define the problem:

$$\begin{array}{rccc} \mathrm{PAC}_{\mathscr{A}_*(2^{\mathbb{N}})} & :\subseteq & \mathscr{A}_*(2^{\mathbb{N}}) \setminus \{\varnothing\} & \rightrightarrows & (2^{\mathbb{N}})^{\mathcal{S}} \\ & & \mathcal{H} & \mapsto & \{A : \mathcal{S} \to \mathcal{H} \mid A \text{ PAC learns } \mathcal{H}\} \end{array}$$

Of course, we can deduce from Theorem 30 that $\mathrm{PAC}_{\mathscr{A}_*(2^{\mathbb{N}})} \leqslant_{\mathrm{sW}} \mathrm{ERM}_{\mathscr{A}_*(2^{\mathbb{N}})}$, which already gives us an upper bound and implies that $\mathrm{PAC}_{\mathscr{A}_\pm(2^{\mathbb{N}})}$ is computable. The question is whether both problems are equivalent, or if there is an easier way to get a PAC learner than computing an ERM.

We can wonder whether there exist learners which are not ERMs. Of course, we can take a function that doesn't behave like an ERM for small samples and it would still fit the definition, but a learner is not necessarily an ERM, even for large sample sizes.

Example 57. *Define $h_i : n \mapsto i$ for $i \in \{0,1\}$ and $\mathcal{H} = \{h_0, h_1\}$. Then $A : \mathcal{S} \to \mathcal{H}$ defined by*

$$A((x_1,y_1),\dots,(x_n,y_n)) = \begin{cases} h_1 & \text{if } 2\,|\{i \in [\![1,n]\!] \mid y_i = 1\}| \geqslant n+2 \\ h_0 & \text{otherwise} \end{cases}$$

PAC learns $\mathcal{H}$, while A is not an $\mathrm{ERM}_{\mathcal{H}}$ (it is biased in favor of h_0).

Proof sketch. Fix some $\varepsilon, \delta > 0$ and let $n \in \mathbb{N}$. Let $\mathcal{D}$ be a distribution over $\mathbb{N} \times \{0,1\}$. Denote by p the probability $\mathbb{P}_{(x,y)\sim\mathcal{D}}(y=1)$. Then $\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) = \min(p, 1-p)$ while, for a given sample $S \in (\mathbb{N}\times\{0,1\})^n$ which contains X_n times the label 1, $L_{\mathcal{D}}(A(S))$ is $1-p$ if $2N \geqslant |S|+2$ and p otherwise.

The "bad" event $E : L_{\mathcal{D}}(A(S)) > \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon$ can therefore only happen if $p < \frac{1+\varepsilon}{2}$ or $p > \frac{1+\varepsilon}{2}$. Note that the random variable X_n follows $\mathscr{B}(n,p)$.

If $p > \frac{1+\varepsilon}{2}$, we can write:

$$\begin{aligned} \mathbb{P}_{S\sim\mathcal{D}^n}(E) &= \mathbb{P}_{X_n\sim\mathscr{B}(n,p)}\left(\frac{X_n-2}{n} < \frac{1}{2}\right) \\ &< \mathbb{P}_{X_n\sim B(n,\frac{1+\varepsilon}{2})}\left(\frac{X_n-2}{n} < \frac{1}{2}\right) \underset{n\to\infty}{\longrightarrow} 0, \end{aligned}$$

using the weak law of large numbers.

The case $p < \frac{1+\varepsilon}{2}$ is purely analogous.

So there is some rank n from which $\mathbb{P}_{S\sim\mathcal{D}^n}(E) < \delta$, regardless of the distribution $\mathcal{D}$. □

We have therefore at this point no strong argument in favor or against this Weihrauch equivalence. Naturally, we can see that the previous example is *close* to being an ERM. Usually, probabilistic versions of problems equivalent to $\lim_{\mathbb{N}^{\mathbb{N}}}$ tend to have connections with SORT, so it may be the case here. This is still to investigate.

6 Conclusion

The work carried out during the internship has enabled us to classify some machine learning problems on the Weihrauch lattice, thereby improving our understanding of some of its regions. Remarkably, all the problems seen so far have been assigned to known equivalence classes. This highlights the stability of this lattice and its ability to support natural problems. Knowing the Weihrauch complexity of these problems also leads to a better intrinsic understanding of their essence, such as the primordiality of positive information for computing the VC dimension of a closed subset of $2^{\mathbb{N}}$.

Some results are still to be shown, in particular about the computation of an ERM for a set described by its negative information, and that of a learner that could not be an ERM. It will then be possible to revisit existing results about PAC learnability in classic computability theory. This approach has often proven fruitful for other areas.

This internship helped me progress a lot, both technically in computable analysis, but also scientifically in general: appropriating and learning a subject not from lectures but from scientific articles, writing an article and looking for the right level of precision in proofs. I would like to express my sincere thanks to Vasco Brattka for agreeing to supervise me and for all the exchanges we had: this work would naturally not have been possible without his many words of advice and clarification. I am also grateful to Olivier Bournez for his follow-up from Paris and for initially introducing me to computable analysis. Thanks also to my colleagues Emmanuel Rauzy, Patrick Uftring and Philip Janicki for all our exchanges and the time spent together.

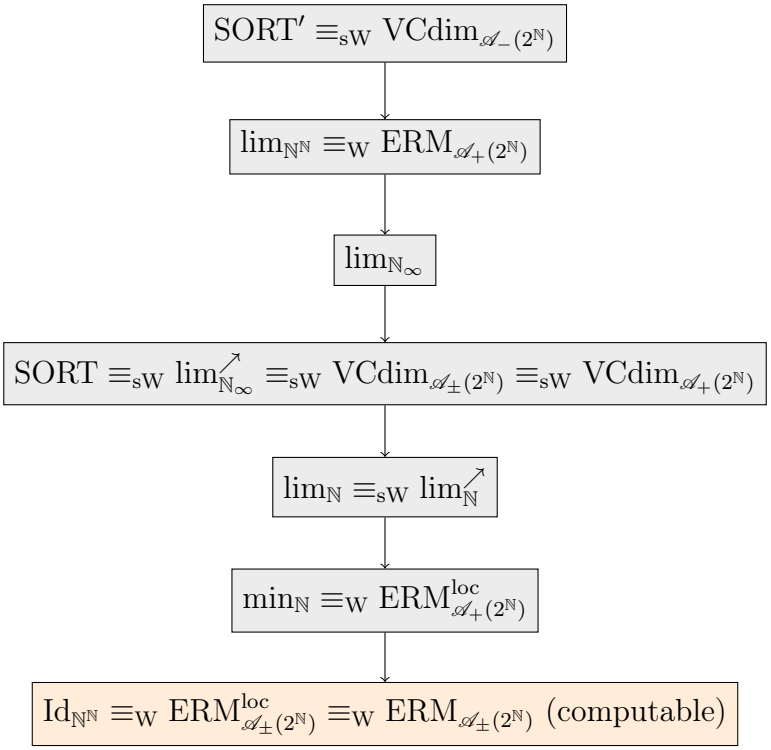

Figure 3: The Weihrauch lattice with our new results

7 References

[Bournez, 2023] Bournez, O. (2023). *Fondements de l'informatique : logique, modèle, calculs*.

[Brattka, 2018] Brattka, V. (2018). Computability and analysis. Work in progress, unpublished.

[Brattka and Hertling, 2021] Brattka, V. and Hertling, P. (2021). *Handbook of Computability and Complexity in Analysis*. Springer Nature.

[Brattka et al., 2008] Brattka, V., Hertling, P., and Weihrauch, K. (2008). *A Tutorial on Computable Analysis*, pages 425–491. Springer New York, New York, NY.

[Chirache, 2025] Chirache, G. (2025). Problèmes NP-complets en analyse calculable : rapport de projet de recherche. École polytechnique, encadré par Olivier Bournez.

[Monin and Patey, 2022] Monin, B. and Patey, L. (2022). *Calculabilité*. Calvage & Mounet.

[Shalev-Shwartz and Ben-David, 2014] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

[Sterkenburg, 2022] Sterkenburg, T. F. (2022). On characterizations of learnability with computable learners.