On the Uniform Learnability of Approximations to Non-Recursive Functions

Frank Stephan[1,*] and Thomas Zeugmann[2,**]

[1] Mathematisches Institut, Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany
fstephan@math.uni-heidelberg.de
[2] Department of Informatics, Kyushu University, Kasuga 816-8580, Japan
thomas@i.kyushu-u.ac.jp

Abstract. Blum and Blum (1975) showed that a class $\mathcal{B}$ of suitable recursive approximations to the halting problem is reliably EX-learnable. These investigations are carried on by showing that $\mathcal{B}$ is neither in NUM nor robustly EX-learnable. Since the definition of the class $\mathcal{B}$ is quite natural and does not contain any self-referential coding, $\mathcal{B}$ serves as an example that the notion of robustness for learning is quite *more restrictive* than intended.
Moreover, variants of this problem obtained by approximating any given recursively enumerable set A instead of the halting problem K are studied. All corresponding function classes $\mathcal{U}(A)$ are still EX-inferable but may *fail* to be reliably EX-learnable, for example if A is non-high and hypersimple. Additionally, it is proved that $\mathcal{U}(A)$ is neither in NUM nor robustly EX-learnable provided A is part of a recursively inseparable pair, A is simple but not hypersimple or A is neither recursive nor high. These results provide more evidence that there is still some need to find an adequate notion for "naturally learnable function classes."

1. Introduction

Though algorithmic learning of recursive functions has been intensively studied within the last three decades there is still some need to elaborate this theory further. For the purpose of motivation, let us shortly recall the basic scenario.

An algorithmic learner is fed growing initial segments of the graph of the target function f. Based on the information received, the learner computes a

* Supported by the Deutsche Forschungsgemeinschaft (DFG) under Research Grant no. Am 60/9-2.
** Supported by the Grant-in-Aid for Scientific Research in Fundamental Areas from the Japanese Ministry of Education, Science, Sports, and Culture under grant no. 10558047. Part of this work was done while visiting the Laboratoire d'Informatique Algorithmique: Fondements et Applications, Université Paris 7. This author is gratefully indebted to Maurice Nivat for providing financial support and inspiring working conditions.

hypothesis on each input. The sequence of all computed hypotheses has to converge to a correct, finite and global description of the target f. We shall refer to this scenario by saying that f is *EX-learnable* (cf. Definition 1).

Clearly, what one is really interested in are powerful learning algorithms that cannot only learn one function but all functions from a given class of functions. Gold [11] provided the first such powerful learner, i.e., the *identification by enumeration* algorithm and showed that it can learn every class contained in NUM. Here NUM denotes the family of all function classes that are subsets of some recursively enumerable class of recursive functions.

There are, however, learnable classes of recursive functions which are not contained in NUM. The perhaps most prominent example is the class $\mathcal{SD}$ of self-describing recursive functions, i.e., of all those functions that compute a program for themselves on input 0. Clearly, $\mathcal{SD}$ is *EX*-learnable.

Since Gold's [11] pioneering paper a huge variety of learning criteria have been proposed within the framework of inductive inference of recursive functions (cf., e.g., [3, 6, 8, 9, 15, 19, 21]). By comparing these inference criteria to one another, it became popular to show separation results by using function classes with self-referential properties. On the one hand, the proof techniques developed are mathematically quite elegant. On the other hand, these separating examples may be considered to be a bit artificial, because of the use of self-describing properties. Hence, Bārzdiņš suggested to look at versions of learning that are closed under computable transformations (cf. [20, 28]). For example, a class $\mathcal{U}$ is *robustly EX-learnable*, iff, for every computable operator Θ such that $\Theta(\mathcal{U})$ is a class of recursive functions, the class $\Theta(\mathcal{U})$ is *EX*-learnable, too (cf. Definition 4). There have been many discussions which operators are admissible in this context (cf., e.g., [10, 14, 16, 20, 23, 28]). At the end, it turned out to be most suitable to consider only general recursive operators, that is, operators which map every total function to a total one. The resulting notion of robust *EX*-learning is the most general one among all notions of robust *EX*-inference.

Next, we state the two main questions that are studied in the present paper.

(1) What is the overall theory developed so far telling us about the learnability of "naturally defined function classes?"
(2) What is known about the robust *EX*-learnability of such "naturally defined function classes?"

Clearly an answer to the first question should tell us something about the usefulness of the theory, and an answer to the second problem should, in particular, provide some insight into the "naturalness" of robust *EX*-learning. However, our knowledge concerning both questions has been severely limited. For function classes in NUM everything is clear, i.e., their learnability has been proved with respect to many learning criteria including robust *EX*-learning. Next, let us consider one of the few "natural" function classes outside NUM that have been considered in the literature, i.e., the class $\mathcal{C}$ of all recursive complexity functions. Then, using Theorem 2.4 and Corollary 2.6. in [23], one can conclude that $\mathcal{C}$ is not robustly *EX*-learnable for many complexity measures including space, since

there is no recursive function that bounds every function in $\mathcal{C}$ for all but finitely many arguments. On the other hand, $\mathcal{C}$ itself is still learnable with respect to many inference criteria by using the identification by enumeration learner.

The latter result already provides some evidence that the notion of robust *EX*-learning may be too restrictive. Nevertheless, the situation may be completely different if one looks at classes of $\{0,1\}$-valued recursive functions, since their learnability differs sometimes considerably from the inferability of arbitrary function classes (cf., e.g., [17, 26]). As far as these authors are aware of, one of the very few "natural classes" of $\{0,1\}$-valued recursive functions that may be a candidate to be not included in *NUM* has been proposed by Blum and Blum [6]. They considered a class $\mathcal{B}$ of approximations to the halting problem K and showed that $\mathcal{B}$ is reliably *EX*-learnable. This class $\mathcal{B}$ is quite natural and not self-describing. It remained, however, open whether or not $\mathcal{B}$ is in *NUM*.

Within the present work, it is shown that $\mathcal{B}$ is neither in *NUM* nor robustly *EX*-learnable. Moreover, we study generalizations of Blum and Blum's [6] original class by considering classes $\mathcal{U}(A)$ of approximations for any recursively enumerable set A. In particular, it is shown that all these classes remain *EX*-learnable but not necessarily reliably *EX*-inferable (cf. Theorems 14 and 16). Furthermore, we show $\mathcal{U}(A)$ to be neither in *NUM* nor robustly *EX*-learnable provided A is part of a recursively inseparable pair, A is simple but not hypersimple or A is neither recursive nor high (cf. Theorems 13 and 17).

Thus the results obtained enlarge our knowledge concerning the learnability of "naturally defined" function classes. Additionally, all those classes $\mathcal{U}(A)$ which are not in *NUM* as well as $\mathcal{B}$ are natural examples for a class which is on the one side not self-describing and on the other side not robustly learnable. So all these $\mathcal{U}(A)$ provide some incidence that the presently discussed notions of robust and hyperrobust learning [1, 7, 10, 14, 16, 23, 28] destroy not only coding tricks but also the learnability of quite natural classes.

Due to the lack of space, many proofs are only sketched or omitted. We refer the reader to [25] for a full version of this paper.

2. Preliminaries

Unspecified notations follow Rogers [24]. $\mathbb{N} = \{0,1,2,\ldots\}$ and $\mathbb{N}^*$ denote the set of all natural numbers and the set of all finite sequences of natural numbers, respectively. $\{0,1\}^*$ stands for the set of all finite $\{0,1\}$-valued sequences and for all $x \in \mathbb{N}$ we use $\{0,1\}^x$ for the set of all $\{0,1\}$-valued sequences of length x.

The classes of all partial recursive and recursive functions of one, and two arguments over $\mathbb{N}$ are denoted by $\mathcal{P}$, $\mathcal{P}^2$, $\mathcal{R}$, and $\mathcal{R}^2$, respectively. $f \in \mathcal{P}$ is said to be monotone provided for all $x, y \in \mathbb{N}$ we have, if both $f(x)$ and $f(y)$ are defined then $f(x) \leq f(y)$. $\mathcal{R}_{0,1}$ and $\mathcal{R}_{mon}$ denotes the set of all $\{0,1\}$-valued recursive functions and of all monotone recursive functions, respectively.

Furthermore, we write f^n instead of the string $(f(0),\ldots,f(n))$, for any $n \in \mathbb{N}$ and $f \in \mathcal{R}$. Sometimes it will be suitable to identify a recursive function with

the sequence of its values, e.g., let $\alpha = (a_0, \ldots, a_k) \in \mathbb{N}^*$, $j \in \mathbb{N}$, and $p \in \mathcal{R}_{0,1}$; then we write αjp to denote the function f for which $f(x) = a_x$, if $x \leq k$, $f(k+1) = j$, and $f(x) = p(x-k-2)$, if $x \geq k+2$. Furthermore, let $g \in \mathcal{P}$ and $\alpha \in \mathbb{N}^*$; we write $\alpha \preceq g$ iff α is a prefix of the sequence of values associated with g, i.e., for all $x \leq k$, $g(x)$ is defined and $g(x) = a_x$.

Any function $\psi \in \mathcal{P}^2$ is called a numbering. Moreover, let $\psi \in \mathcal{P}^2$, then we write ψ_i for the function $x \to \psi(i,x)$ and set $\mathcal{P}_\psi = \{\psi_i \mid i \in \mathbb{N}\}$ as well as $\mathcal{R}_\psi = \mathcal{P}_\psi \cap \mathcal{R}$. Consequently, if $f \in \mathcal{P}_\psi$, then there is a number i such that $f = \psi_i$. If $f \in \mathcal{P}$ and $i \in \mathbb{N}$ are such that $\psi_i = f$, then i is called a ψ–program for f. Let ψ be any numbering, and $i, x \in \mathbb{N}$; if $\psi_i(x)$ is defined (abbr. $\psi_i(x)\downarrow$) then we also say that $\psi_i(x)$ *converges*. Otherwise, $\psi_i(x)$ is said to *diverge* (abbr. $\psi_i(x)\uparrow$).

A numbering $\varphi \in \mathcal{P}^2$ is called a Gödel numbering or acceptable numbering (cf. [24]) iff $\mathcal{P}_\varphi = \mathcal{P}$, and for any numbering $\psi \in \mathcal{P}^2$, there is a $c \in \mathcal{R}$ such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. In the following, let φ be any fixed Gödel numbering. As usual, we define the halting problem to be the set $K = \{i \mid i \in \mathbb{N},\ \varphi_i(i)\downarrow\}$. Any function $\Phi \in \mathcal{P}^2$ satisfying $dom(\varphi_i) = dom(\Phi_i)$ for all $i \in \mathbb{N}$ and $\{(i,x,y) \mid i,x,y \in \mathbb{N},\ \Phi_i(x) \leq y\}$ is recursive is called a *complexity measure* (cf. [5]).

Furthermore, let $NUM = \{\mathcal{U} \mid (\exists \psi \in \mathcal{R}^2)[\mathcal{U} \subseteq \mathcal{P}_\psi]\}$ denote the family of all subsets of all recursively enumerable classes of recursive functions.

Next, we define the concepts of learning mentioned in the introduction.

Definition 1. Let $\mathcal{U} \subseteq R$ and $M : \mathbb{N}^* \to \mathbb{N}$ be a recursive machine.

(a) (Gold [11]) M is an *EX-learner* for $\mathcal{U}$ iff, for each function $f \in \mathcal{U}$, M converges syntactically to f in the sense that there is a $j \in \mathbb{N}$ with $\varphi_j = f$ and $j = M(f^n)$ for all but finitely many $n \in \mathbb{N}$.

(b) (Angluin [2]) M is a *conservative EX-learner* for $\mathcal{U}$ iff M EX-learns $\mathcal{U}$ and M makes in addition only *necessary hypothesis changes* in the sense that, whenever $M(\sigma\eta) \neq M(\sigma)$ then the program $M(\sigma)$ is inconsistent with the data $\sigma\eta$ by either $\varphi_{M(\sigma)}(x)\uparrow$ or $\varphi_{M(\sigma\eta)}(x)\downarrow \neq \sigma\eta(x)$ for some $x \in dom(\sigma\eta)$.

(c) (Bārzdins [4], Case and Smith [8]) M is a *BC-learner* for $\mathcal{U}$ iff, for each function $f \in \mathcal{U}$, M converges semantically to f in the sense that $\varphi_{M(f^n)} = f$ for all but finitely many $n \in \mathbb{N}$.

A class $\mathcal{U}$ is EX-learnable iff it has a recursive EX-learner and EX denotes the family of all EX-learnable function classes. Similar we define when a class is conservatively EX-learnable or BC-learnable. We write BC for the family of all BC-learnable function classes.

Note that $EX \subset BC$ (cf. [8]). As far as we are aware of, it has been open whether or not conservative learning constitutes a restriction for EX-learning of recursive functions. The negative answer is provided by the next proposition.

Proposition 2. EX = conservative-EX.

Nevertheless, whenever suitable, we shall design a conservative learner instead of just an EX-learner, thus avoiding the additional general transformation given by the proof of Proposition 2.

Next, we define reliable inference. Intuitively, a learner M is reliable provided it converges if and only if it learns. There are several variants of reliable learning, so we will give a justification of our choice below.

Definition 3 (Blum and Blum [6], Minicozzi [21]). Let $\mathcal{U} \subseteq R$; then $\mathcal{U}$ is said to be *reliably EX-learnable* if there is a machine $M \in \mathcal{R}$ such that

(1) M EX-learns $\mathcal{U}$ and
(2) for all $f \in \mathcal{R}$, if the sequence $(M(f^n))_{n\in\mathbb{N}}$ converges, say to j, then $\varphi_j = f$.

By $\mathcal{R}EX$ we denote the family of all reliably EX-learnable function classes.

Note that one can replace the condition "$f \in \mathcal{R}$" in (2) of Definition 3 by "$f \in \mathcal{P}$" or "all total f." This results in a different model of reliable learning, say $\mathcal{P}EX$ and $\mathcal{T}EX$, respectively. Then for every $\mathcal{U} \subseteq \mathcal{R}_{0,1}$ such that $\mathcal{U} \in \mathcal{P}EX$ or $\mathcal{U} \in \mathcal{T}EX$ one has $\mathcal{U} \in NUM$ (cf. [6, 12, 26]). On the other hand, there are classes $\mathcal{U} \subseteq \mathcal{R}_{0,1}$ such that $\mathcal{U} \in \mathcal{R}EX \setminus NUM$ (cf. [12]). As a matter of fact, our Theorem 6 below together with Blum and Blum's [6] result $\mathcal{B} \in \mathcal{R}EX$ provides a much easier proof of the same result than Grabowski [12].

Finally, we define robust EX-learning. This involves the notion of general recursive operators. A general recursive operator is a computable mapping that maps functions over $\mathbb{N}$ to functions over $\mathbb{N}$ and every total function has to be mapped to a total function. For a formal definition and more information about general recursive operators the reader is referred to [13, 22, 27].

Definition 4 (Jain, Smith and Wiehagen [16]). Let $\mathcal{U} \subseteq \mathcal{R}$; then $\mathcal{U}$ is said to be *robustly EX-learnable* if $\Theta(\mathcal{U})$ is EX-learnable for every general recursive operator Θ. By robust-EX we denote the family of all robustly EX-learnable function classes.

3. Approximating the Halting Problem

Within this section, we deal with Blum and Blum's [6] class $\mathcal{B}$. First, we define the class of approximations to the halting problem considered in [6].

Definition 5. Let $\tau \in \mathcal{R}$ be such that for all $i \in \mathbb{N}$

$$\varphi_{\tau(i)}(x) = \begin{cases} 1, & \text{if } \Phi_i(x)\downarrow \text{ and } \Phi_x(x) \leq \Phi_i(x) \\ 0, & \text{if } \Phi_i(x)\downarrow \text{ and } \neg[\Phi_x(x) \leq \Phi_i(x)] \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now, we set $\mathcal{B} = \{\varphi_{\tau(i)} \mid i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}_{mon}\}$.

Blum and Blum [6] have shown $\mathcal{B} \in \mathcal{R}EX$ but left it open whether or not $\mathcal{B} \in NUM$. It is not, as our next theorem shows.

Theorem 6. $\mathcal{B} \notin NUM$.

Proof. First, recall that K is part of a recursively inseparable pair (cf. [22, Exercise III.6.23.(a)]). That is, there is an r.e. set H such that $K \cap H = \emptyset$ and for

every recursive set $A \supseteq H$ we have $|A \cap K| = \infty$. Now, we fix any enumeration $k_0, k_1, k_2, \ldots$ and $h_0, h_1, h_2, \ldots$ of K and H, respectively. Suppose to the contrary, that there exists a numbering $\psi \in \mathcal{R}^2$ such that $\mathcal{B} \subseteq \mathcal{R}_\psi$. Next, we define for each ψ_e a function $g_e \in \mathcal{P}$ as follows. For all $e, x \in \mathbb{N}$ let

$g_e(x) =$ "Search for the least n such that for $n = s + y$ either (A), (B) or (C) happens:

(A) $y = h_s \ \wedge \ \psi_e(y) = 1$
(B) $y = k_s \ \wedge \ \psi_e(y) = 0 \ \wedge \ y > x$
(C) $\psi_e(y) > 1$

If (A) happens first, then set $g_e(x) = s + y$.
If (B) happens first, then let $g_e(x) = \Phi_y(y) + y$.
If (C) happens first, then let $g_e(x) = 0$."

Claim 1. *$g_e \in \mathcal{R}$ for all $e \in \mathbb{N}$.*

If there is at least one y such that $\psi_e(y) > 1$, then $g_e \in \mathcal{R}$. Now let $\psi_e \in \mathcal{R}_{0,1}$ and suppose that there is an $x \in \mathbb{N}$ with $g_e(x)\uparrow$. Then there are no s, y such that $y = h_s$ and $\psi_e(y) = 1$. Hence, $M = \{y \mid y \in \mathbb{N} \wedge \psi_e(y) = 0\} \supseteq H$ and M is recursive. Thus, $|M \cap K| = \infty$. So there must be a $y > x$ such that $\psi_e(y) = 0$ and an $s \in \mathbb{N}$ with $y = k_s$. Thus (B) must happen, and since $y = k_s$, we conclude $\Phi_y(y)\downarrow$. Hence, $g_e(x)\downarrow$, too, a contradiction. This proves Claim 1.

Claim 2. *Let e be any number such that $\psi_e = \varphi_{\tau(i)}$ for some $\varphi_{\tau(i)} \in \mathcal{B}$. Then $g_e(x) > \Phi_i(x)$ for all $x \in \mathbb{N}$.*

Assume any i, e as above, and consider the definition of $g_e(x)$. Suppose $g_e(x) = s + y$ for some s, y such that $y = h_s$ and $\psi_e(y) = 1$. Since $\psi_e(y) = \varphi_{\tau(i)}(y) = 1$ implies $\Phi_y(y) \leq \Phi_i(y)$, and hence $y \in K$, we get a contradiction to $K \cap H = \emptyset$. Thus, this case cannot happen.

Consequently, in the definition of $g_e(x)$ condition (B) must have happened. Thus, some s, y such that $y > x$, $y = k_s$ and $\psi_e(y) = 0$ have been found. Since $y = k_s$, we conclude $\Phi_y(y)\downarrow$ and thus $g(x) > \Phi_y(y)$. Because of $\psi_e(y) = \varphi_{\tau(i)}(y) = 0$, we obtain $\Phi_i(y) < \Phi_y(y)$ by the definition of $\varphi_{\tau(i)}$. Now, putting it all together, we get $g(x) > \Phi_y(y) > \Phi_i(y) \geq \Phi_i(x)$, since $y > x$ and $\Phi_i \in \mathcal{R}_{mon}$. This proves Claim 2.

Claim 3. *For every $b \in \mathcal{R}$ there exists an $i \in \mathbb{N}$ such that $\Phi_i \in \mathcal{R}_{mon}$ and $b(x) < \Phi_i(x)$ for all $x \in \mathbb{N}$.*

Let $r \in \mathcal{R}$ be such that for all $j, x \in \mathbb{N}$ we have

$$\varphi_{r(j)}(0) = \begin{cases} 0, & \text{if } \neg[\Phi_j(0) \leq b(0)] \\ \varphi_j(0) + 1, & \text{otherwise} \end{cases}$$

and for $x > 0$

$$\varphi_{r(j)}(x) = \begin{cases} 0, & \text{if } \Phi_j(n) \text{ is defined for all } n < x \ \wedge \\ & \quad \neg[\Phi_j(x) \leq \Phi_j(x-1) \ \vee \ \Phi_j(x) \leq b(x)] \\ \varphi_j(x) + 1, & \text{if } \Phi_j(n) \text{ is defined for all } n < x \ \wedge \\ & \quad [\Phi_j(x) \leq \Phi_j(x-1) \ \vee \ \Phi_j(x) \leq b(x)] \\ \uparrow, & \text{otherwise.} \end{cases}$$

By the fixed point theorem [24] there is an $i \in \mathbb{N}$ such that $\varphi_{r(i)} = \varphi_i$. Now, one inductively shows that $\varphi_i = 0^\infty$, $\Phi_i \in \mathcal{R}_{mon}$ and $b(x) < \Phi_i(x)$ for all $x \in \mathbb{N}$ and Claim 3 follows.

Finally, by Claim 1, all $g_e \in \mathcal{R}$, and thus there is a function $b \in \mathcal{R}$ such that $b(x) \geq g_e(x)$ for all $e \in \mathbb{N}$ and all but finitely many $x \in \mathbb{N}$ (cf. [6]). Together with Claim 2, this function b contradicts Claim 3, and hence $\mathcal{B} \notin \mathit{NUM}$. ∎

The next result can be obtained by looking at $\mathcal{U}(K)$ in Theorems 15 and 17.

Theorem 7. *$\mathcal{B}$ is $\mathcal{R}$EX-inferable but not robustly EX-learnable.*

Theorems 6 and 7 immediately allow the following separation, thus reproving Grabowski's [12] Theorem 5.

Corollary 8. $\mathit{NUM} \cap \wp(\mathcal{R}_{0,1}) \subset \mathcal{R}\mathit{EX} \cap \wp(\mathcal{R}_{0,1})$.

Finally, we ask whether or not the condition $\Phi_i \in \mathcal{R}_{mon}$ in the definition of the class $\mathcal{B}$ is necessary. The affirmative answer is given by our next theorem. That is, instead of $\mathcal{B}$, we now consider the class $\tilde{\mathcal{B}} = \{\varphi_{\tau(i)} \mid i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}\}$.

Theorem 9. *$\tilde{\mathcal{B}}$ is not BC-learnable.*

Next, we generalize the approach undertaken so far by considering classes $\mathcal{U}(A)$ of approximations to any recursively enumerable (abbr. r.e.) set A.

4. Approximating Arbitrary r.e. Sets

The definition of Blum and Blum's [6] class uses implicitly the measure Φ_K defined as $\Phi_K(x) = \Phi_x(x)$ for measuring the speed by which K is enumerated. Using this notion Φ_K, the class $\mathcal{B}$ of approximations of K is defined as

$$\mathcal{B} = \{f \in \mathcal{R}_{0,1} \mid (\exists \Phi_e \in \mathcal{R}_{mon})\,(\forall x)\,[f(x) = 1 \Leftrightarrow \Phi_K(x) \leq \Phi_e(x)]\}.$$

Our main idea is to replace K by an arbitrary r.e. set A and to replace Φ_K by a measure Φ_A of (the enumeration speed of) A. Such a measure satisfies the following two conditions:

- The set $\{(x, y) \mid \Phi_A(x)\!\downarrow \leq y\}$ is recursive.
- $x \in A \Leftrightarrow (\exists y)\,[\Phi_A(x) \leq y]$.

Here, Φ_A is intended to be taken as the function Φ_i of some index i of A, but sometimes we might also take the freedom to look at some other functions Φ_A satisfying the two requirements above. The natural definition for a class $\mathcal{U}(A)$ corresponding to the class $\mathcal{B}$ in the case $A = K$ based on an underlying function Φ_A is the following.

Definition 10. Given an r.e. set A, an enumeration Φ_A and a total function Φ_e, let

$$f_e(x) = \begin{cases} 1, & \text{if } \Phi_A(x) \leq \Phi_e(x) \\ 0, & \text{if } \neg[\Phi_A(x) \leq \Phi_e(x)]. \end{cases}$$

Now $\mathcal{U}(A)$ consists of all those f_e where $\Phi_e \in \mathcal{R}_{mon}$.

Next, comparing $\mathcal{U}(K)$ to the original class $\mathcal{B}$ of Blum and Blum [6] one can easily prove the following. For every $f \in \mathcal{B}$ there is a function $g \in \mathcal{U}(K)$ such that for all $x \in \mathbb{N}$ we have $f(x) = 1$ implies $g(x) = 1$. Hence, the approximation g is at least as good as f. The converse is also true, i.e., for each $g \in \mathcal{U}(K)$ there is an $f \in \mathcal{B}$ such that $g(x) = 1$ implies $f(x) = 1$ for all $x \in \mathbb{N}$. Therefore, we consider our new classes of approximations as natural generalizations of Blum and Blum's [6] original definition.

Moreover, note that there is a function gen_A which computes for every e of a monotone Φ_e a program $gen_A(e)$ for the function f associated with Φ_e:

$$\varphi_{gen_A(e)}(x) = \begin{cases} 1, & \text{if } \Phi_A(x) \leq \Phi_e(x)\downarrow \ \wedge\ (\forall y < x)[\Phi_e(y) \leq \Phi_e(y+1)] \\ 0, & \text{if } \neg[\Phi_A(x) \leq \Phi_e(x)\downarrow] \ \wedge\ (\forall y < x)[\Phi_e(y) \leq \Phi_e(y+1)] \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now, if A is recursive, everything is clear, since we have the following.

Theorem 11. *If A is recursive then $\mathcal{U}(A) \in NUM$.*

The direct generalization of Theorem 6 would be that $\mathcal{U}(A)$ is not in *NUM* for every non-recursive r.e. set A and every measure Φ_A. Unfortunately, there are some special cases where this is still unknown to us.

We obtained many intermediate results which give incidence that $\mathcal{U}(A)$ is not in *NUM* for any non-recursive r.e. set A. First, every non-recursive set A has a sufficiently "slow" enumeration such that $\mathcal{U}(A) \notin NUM$ for this underlying enumeration and the corresponding Φ_A. Second, for many classes of sets we can directly show that $\mathcal{U}(A) \notin NUM$, whatever measure Φ_A we choose. Besides the cases where A is part of a recursively inseparable pair or A is simple but not hypersimple, the case of the non-recursive and non-high sets A is interesting, in particular, since the proof differs from that for the two previous cases.

Recall that a set A is *simple* iff A is both r.e. and infinite, $\overline{A}$ is infinite but there is no infinite recursive set R disjoint to A. A set A is *hypersimple* iff A is both r.e. and infinite, and there is no function $f \in \mathcal{R}$ such that $f(n) \geq \overline{a}_n$ for all $n \in \mathbb{N}$, where $\overline{a}_0, \overline{a}_1, \ldots$ is the enumeration of $\overline{A}$ in strictly increasing order (cf. Rogers [24]). Using this definition of hypersimple sets, one can easily show the following lemma.

Lemma 12. *A set $A \subseteq \mathbb{N}$ is hypersimple iff*
(a) *A is r.e. and both A and $\overline{A}$ are infinite*
(b) *for all functions $g \in \mathcal{R}$ with $g(x) \geq x$ for all $x \in \mathbb{N}$ there exist infinitely many $x \in \mathbb{N}$ such that $\{x, x+1, \ldots, g(x)\} \subseteq A$.*

Now, we are ready to state the announced theorem.

Theorem 13. *$\mathcal{U}(A)$ is not in NUM for the following r.e. sets A.*
(a) *A is part of a recursively inseparable pair.*
(b) *A is simple but not hypersimple.*
(c) *A is neither recursive nor high.*

Proof. We sketch only the proof of Assertion (c) here. Assume by way of contradiction $\mathcal{U}(A) \in NUM$. Thus, there is a $\psi \in \mathcal{R}^2$ such that $\mathcal{U}(A) \subseteq \mathcal{R}_\psi$. Assume without loss of generality that $0 \in A$. The A-recursive function $d_A(x) = \max\{\Phi_A(y) \mid y \leq x \text{ and } y \in A\}$ is total and recursive relative to A. If now $m(x) \geq d_A(x)$, then the function generated by m in accordance to Definition 10 is equal to the characteristic function of A.

$$A(x) = f_m(x) = \begin{cases} 1, & \text{if } \Phi_A(x) \leq m(x) \\ 0, & \text{if } \neg[\Phi_A(x) \leq m(x)]. \end{cases}$$

So one can define the following A-recursive function h:

$$h(x) = \min\{y \geq x \mid (\forall j \leq x)\,(\exists z)[(x \leq z \leq y) \wedge \psi_j(z) \neq A(z)]\}.$$

Since A is not recursive, no function ψ_j can be a finite variant of $A(x)$, and thus h is total. Using h we next the following total A-recursive function g by $g(x) = \sum_{y=x}^{h(x)} d_A(y)$. Since A is not high, there is a function $b \in \mathcal{R}$ such that $b(x) \geq g(x)$ for infinitely many x. By Claim 3 in the demonstration of Theorem 6, there exists an $e \in \mathbb{N}$ such $\Phi_e \in \mathcal{R}_{mon}$ and $\Phi_e(x) > b(x)$ for all $x \in \mathbb{N}$. Thus, $\Phi_e(x) \geq g(x)$ for infinitely many x.

Next, for every $\psi_k \in \mathcal{R}_\psi$ there exists an $x > k$ such that $\Phi_e(x) > g(x)$. Consider all $y = x, x+1, \ldots, h(x)$. By the definition of g and by $\Phi_e \in \mathcal{R}_{mon}$, we have $\Phi_e(y) \geq d_A(y)$ for all these y. Thus, by the choice of d_A and the definition of $\varphi_{gen_A(e)}$ we arrive at $\varphi_{gen_A(e)}(y) = A(y)$ for all $y = x, x+1, \ldots, h(x)$. But now the definition of the function h guarantees that $\psi_k(z) \neq \varphi_{gen_A(e)}(z)$ for some z with $x \leq z \leq h(x)$. Consequently, $\varphi_{gen_A(e)}$ differs from all f_k in contradiction to the assumption $\mathcal{U}(A) \in NUM$. ∎

5. Reliable and EX-Learnability of $\mathcal{U}(A)$

Blum and Blum [6] showed $\mathcal{B} \in \mathcal{R}EX$. The EX-learnability of $\mathcal{U}(A)$ alone can be generalized to every r.e. set A, but this is not possible for reliability. But before dealing with $\mathcal{R}EX$-inference, we show that every $\mathcal{U}(A)$ is EX-learnable.

Theorem 14. *$\mathcal{U}(A)$ is EX-learnable for all r.e. sets A.*

Proof. If A is recursive, then $\mathcal{U}(A) \in NUM$ (cf. Theorem 11) and thus EX-learnable. So let A be non-recursive and let Φ_A be a recursive enumeration of A. An EX-learner for the class $\mathcal{U}(A)$ is given as follows.

- On input σ, disqualify all e such that there are $x \in dom(\sigma)$ and $y \leq |\sigma|$ satisfying one of the following three conditions:
  (a) $\Phi_{gen_A(e)}(x) \leq |\sigma|$ and $\varphi_{gen_A(e)}(x) \neq \sigma(x)$
  (b) $\sigma(x) = 0$, $\Phi_A(x) \leq y$ and $\neg[\Phi_e(x) \leq y]$
  (c) $\Phi_e(x+1) \leq y$ and $\neg[\Phi_e(x) \leq y]$.
- Output $gen_A(e)$ for the smallest e not yet disqualified.

The algorithm disqualifies only such indices e where $\varphi_{gen_A(e)}$ is either defined and false or undefined for some $x \in dom(\sigma)$. Thus the learner is conservative.

Since the correct indices are never disqualified, it remains to show that the incorrect ones are. This clearly happens if $\varphi_{gen_A(e)}(y)\downarrow \neq \sigma(y)$ for some y. Otherwise let z be the first undefined place of $\varphi_{gen_A(e)}$. This undefined place is either due to the fact that $\Phi_e(x) > \Phi_e(x+1)$ for some $x < z$ or that $\Phi_e(z)\uparrow$. In the first case, e is eventually disqualified by condition (c), in the second case, either $\Phi_e(x+1)\downarrow$ for some first $x \geq z$, then e is again eventually disqualified by condition (c) or $\Phi_e(x)\uparrow$ for some $x \in A$ above z and so e is disqualified by condition (b). Hence, the learning algorithm is correct. ∎

The result that $\mathcal{B}$ is reliably EX-learnable can be generalized to halves of recursively inseparable pairs and to simple but not hypersimple sets.

Theorem 15. *$\mathcal{U}(A)$ is reliably EX-learnable if*
(a) *A is part of a recursively inseparable pair or*
(b) *A is simple but not hypersimple.*

Proof. The central idea of the proof is that conditions (a) and (b) allow to identify a class of functions which contains all recursive functions which are too difficult to learn and on which the learner then signals infinitely often divergence. The recursive functions outside this class turn out to be EX-learnable and contain the class $\mathcal{U}(A)$.

The learner M does not need to succeed on functions $f \notin \mathcal{R}_{0,1}$ or if $f(x) = 1$ for almost all $x \in A$. Now, the second condition can be checked indirectly for $f \in \mathcal{R}_{0,1}$ and the A in the precondition of the theorem.

In case (a), let A and $B = \{b_0, b_1, \ldots\}$ form a recursively inseparable pair. If $f(x) = 1$ for almost all $x \in A$ then $f(b_s) = 1$ for some b_s. So one defines that σ disqualifies if $\sigma(x) \geq 2$ for some x or if $\sigma(b_s)\downarrow = 1$ for some $s \leq |\sigma|$.

In case (b), the set A is simple but not hypersimple. By Lemma 12 there is a function $g \in \mathcal{R}$ with $g(x) \geq x$ for all $x \in \mathbb{N}$ such that $\overline{A}$ intersects every interval $\{x, x+1, \ldots, g(x)\}$. But if $f(x) = 1$ for almost all $x \in A$, then, by the simplicity of A, $f(x) = 1$ for almost all x and there is an x with $f(y) = 1$ for all $y \in \{x, x+1, \ldots, g(x)\}$. So one defines that σ disqualifies if $\sigma(x) \geq 2$ for some x or if there is an x and $\sigma(y) = 1$ for all $y \in \{x, x+1, \ldots, g(x)\}$.

The reliable EX-learner N is a modification of the learner M from Theorem 14 which copies M on all σ except those which disqualify — on them, N always outputs a guess for $\sigma 0^\infty$ and thus either converges to some $\sigma 0^\infty$ or diverges by infinitely many changes of the hypothesis. Let $e(\sigma)$ be a program for $\sigma 0^\infty$ and let

$$N(\sigma) = \begin{cases} e(\sigma), & \text{if } \sigma \text{ is disqualified} \\ M(\sigma), & \text{otherwise.} \end{cases}$$

For the verification, note that for every $f \in \mathcal{U}(A)$ we have $f(x) = 0$ for all $x \in \overline{A}$. Thus, if $f \in \mathcal{U}(A)$ then no $\sigma \preceq f$ is disqualified and therefore N is an EX-learner for $\mathcal{U}(A)$.

Assume now that N converges to an e' on some recursive function f. If

this happens for a function f such that some $\sigma' \preceq f$ has been disqualified then $f = \sigma 0^\infty$ and so also $\varphi_{e'} = \sigma 0^\infty$ for some $\sigma \preceq f$. Thus, N converges to a correct program for f in this case.

Otherwise, no $\sigma' \preceq f$ is disqualified. Since N copies the indices of M and those are all of the form $gen_A(e)$, there is a least e with $e' = gen_A(e)$. If $f(x) = 0$ for infinitely many $x \in A$, then M converges only to $gen_A(e)$ if $\varphi_{gen_A(e)} = f$ and the algorithm is correct in that case.

Finally, consider the subcase that $f(x) = 0$ for only finitely many $x \in A$. Consequently, in case (a) $f(x) = 1$ for some $x \in B$ and in case (b) there must be an x such that $f(y) = 1$ for all $y = x, x+1, \ldots, g(x)$. In both cases, some $\sigma' \preceq f$ is disqualified, thus this case cannot occur. Hence, N is reliable. ∎

Theorem 16. *If A is hypersimple and not high then $\mathcal{U}(A) \notin \mathcal{REX}$.*

Proof. Let A be a hypersimple non-high set, let Φ_A be a corresponding measure, and assume to the contrary that $\mathcal{U}(A) \in \mathcal{REX}$. Then also the union

$$\mathcal{U}(A) \cup \{\alpha 1^\infty \mid \alpha \in \{0,1\}^*\}$$

is EX-learnable, since every class in NUM is also in $\mathcal{REX}$ and $\mathcal{REX}$ is closed under union (cf. [6, 21]). Given an EX-learner M for the above union, one can define the following function h_1 by taking

$$h_1(x) = \min\{s \geq x \mid (\forall \sigma \in \{0,1\}^x)\,(\forall y \leq x)\,[M(\sigma 1^s) \neq M(\sigma) \ \vee \ \Phi_{M(\sigma)}(y) \leq s \ \wedge \ \varphi_{M(\sigma)}(y) = \sigma(y)]\}.$$

The function h_1 is total since any guess $M(\sigma)$ either computes the function $\sigma 1^\infty$ or is eventually replaced by a new guess on $\sigma 1^\infty$. Note that $h_1 \in \mathcal{R}_{mon}$ and $h_1(x) \geq x$ for all x.

Since A is not recursive, there is no total function dominating Φ_A. Thus one can define a recursive function $h_2(x)$ by taking

$$h_2(x) = \text{ the smallest } s \text{ such that there is a } y \text{ with } x \leq y \leq s \wedge h_1(y + h_1(y)) < s \wedge \Phi_A(y) + \Phi_A(y+1) + \ldots + \Phi_A(y + h_1(y)) < s\,.$$

Since A is hypersimple, we directly get from Lemma 12 that $h_2 \in \mathcal{R}$. Consider for every $f \in \mathcal{U}(A)$ the index i to which M converges and an index j with $f = \varphi_{gen_A(j)}$.

Assume now that M has converged to i at $z \leq x$. Consider the y, s from the definition of h_2 and let $\sigma = f(0), \ldots, f(y)$. If $M(\sigma 1^{h_1(y)}) \neq M(\sigma)$ then there is some $y' \in \{y, y+1, \ldots, y + h_1(y)\}$ with $f(y') = 0$. As a consequence, $\Phi_j(y') < \Phi_A(y') < h_2(y)$. Since $\Phi_j \in \mathcal{R}_{mon}$, we know $\Phi_j(y) < s$. Otherwise, $\Phi_i(x) \leq h_1(y)$ and $\varphi_i(x)$ has converged. Since $y \leq h_2(x)$, we conclude $\Phi_i(x) \leq h_1(h_2(x))$. So one can give the following definition for f by case-distinction where the first case is taken which is applicable and where $\sigma = f(0), \ldots, f(z)$.

$$\varphi_{e(i,j,\sigma)}(x) = \begin{cases} \sigma(x), & \text{if } x \in dom(\sigma) \\ \varphi_i(x), & \text{if } \Phi_i(x) \leq h_1(h_2(x)) \\ 1, & \text{if } \Phi_A(x) \leq \Phi_j(x) \leq h_2(x) \\ 0, & \text{otherwise.} \end{cases}$$

Since the search-conditions in the second and third case are bounded by a recursive function in x, the family of all $\varphi_{e(i,j,\sigma)}$ contains only total functions and its universal $i, j, \sigma, x \to \varphi_{e(i,j,\sigma)}(x)$ is computable in all parameters. Furthermore, for the correct i, j, σ as chosen above, $\varphi_{e(i,j,\sigma)}$ equals the given f since, for all $x > z$, either $\varphi_i(x)$ converges within $h_1(h_2(x))$ steps to $f(x)$ or $\Phi_A(x) \leq \Phi_j(x) \leq h_2(x)$. It follows that this family covers $\mathcal{U}(A)$ and that $\mathcal{U}(A)$ is in *NUM* which, a contradiction to Theorem 13, since A is neither recursive nor high. ∎

6. Robust Learning

A mathematical elegant proof method to separate learning criteria is the use of classes of self-describing functions. On the one hand, these examples are a bit artificial, since they use coding tricks. On the other hand, natural objects like cells contain a description of themselves. Nevertheless, from a learning theoretical point some criticism remains in order, since a learner needs only to fetch some code from the input.

Therefore, Bārzdiņš suggested to look at restricted versions of learning: For example, a class S is robustly EX-learnable, iff, for every operator Θ, the class $\Theta(S)$ is EX-learnable. There were many discussions, which operators Θ are admissible in this context and how to deal with those cases where Θ maps some functions in S to partial functions. At the end, it turned out that it is most suitable to consider only general recursive operators Θ which map every total function to a total one [16]. This notion is among all notions of robust EX-learning the most general one in the sense that every class S which is robustly EX-learnable with respect to any criterion considered in the literature is also robustly EX-learnable with respect to the model of Jain, Smith and Wiehagen [16].

Although the class $\mathcal{B}$ is quite natural and does not have any obvious self-referential coding, the class $\mathcal{B}$ is not robustly EX-learnable — so while on the one hand the notion of robust EX-learning still permits topological coding tricks [16, 23], it does on the other hand already rule out the natural class $\mathcal{B}$. The provided example gives some incidence, that there is still some need to find a adequate notion for a "natural EX-learnable class."

Every class in *NUM* is robustly EX-learnable, in particular the class $\mathcal{U}(A)$ for a recursive set A (cf. Theorem 11). The next theorem shows that $\mathcal{U}(A)$ is not robustly EX-learnable for any nonrecursive sets A which are part of a recursively inseparable pair, which are simple but not hypersimple or which are neither recursive nor high. Thus, here the situation is parallel to the one at Theorem 13.

Theorem 17. *$\mathcal{U}(A)$ is not robustly EX-learnable for the following r.e. sets A.*
(a) *A is part of a recursively inseparable pair.*
(b) *A is simple but not hypersimple.*
(c) *A is neither recursive nor high.*

7. Conclusions

The main topic of the present investigations have been the class $\mathcal{B}$ of Blum and Blum [6] and the natural generalizations $\mathcal{U}(A)$ of it obtained by using r.e. sets A as a parameter. It is has been shown that for large families of r.e. sets A, these classes $\mathcal{U}(A)$ are not in NUM. Furthermore, they can be always EX-learned. Moreover, for some but not all sets A there is also a $\mathcal{R}EX$-learner. Robust EX-learning is impossible for all non-recursive sets A that are part of recursively inseparable pair, for simple but not hypersimple sets A and for all sets A that are non-high and non-recursive. Since the classes $\mathcal{U}(A)$ are quite natural, this result adds some incidence that "natural learnability" does not coincide with robust learnability as defined in the current research.

Future work might address the remaining unsolved question whether $\mathcal{U}(A)$ is outside NUM for all non-recursive sets A. Additionally, one might investigate whether $\mathcal{U}(A)$ is robustly BC-learnable for some sets A such that $\mathcal{U}(A)$ is not robustly EX-inferable. It would be also interesting to know whether or not $\mathcal{U}(A)$ can be reliably BC-learned for sets A with $\mathcal{U}(A) \notin \mathcal{R}EX$ (cf. [18] for more information concerning reliable BC-learning). Finally, there are some ways to generalize the notion of $\mathcal{U}(A)$ to every K-recursive set A and one might investigate the learning theoretic properties of the so obtained classes.

References

1. A. Ambainis and R. Freivalds. Transformations that preserve learnability. In *Proccedings of the 7th International Workshop on Algorithmic Learning Theory* (ALT'96) (S. Arikawa and A. Sharma, Eds.) Lecture Notes in Artificial Intelligence Vol. 1160, pages 299–311, Springer-Verlag, Berlin, 1996.
2. D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
3. D. Angluin and C.H. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–289, 1983.
4. J. Bārzdins. Prognostication of automata and functions. *Information Processing* '71, (1) 81–84. Edited by C. P. Freiman, North-Holland, Amsterdam, 1971.
5. M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the Association for Computing Machinery*, 14:322–336.
6. L. Blum and M. Blum. Towards a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.
7. J. Case, S. Jain, M. Ott, A. Sharma and F. Stephan. Robust learning aided by context. In *Proceedings of 11th Annual Conference on Computational Learning Theory* (COLT'98), pages 44–55, ACM Press, New York, 1998.
8. J. Case and C.H. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science* 25:193–220, 1983.
9. R. Freivalds. Inductive inference of recursive functions: Qualitative theory. In *Baltic Computer Science* (J. Bārzdiņš and D. Bjørner, Eds.), Lecture Notes in Computer Science Vol. 502, pages 77–110. Springer-Verlag, Berlin, 1991.

10. M. Fulk. Robust separations in inductive inference. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science* (FOCS), pages 405–410, St. Louis, Missouri, 1990.
11. M.E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
12. J. Grabowski. Starke Erkennung. *In* Strukturerkennung diskreter kybernetischer Systeme, (R. Linder, H. Thiele, Eds.), Seminarberichte der Sektion Mathematik der Humboldt-Universität Berlin Vol. 82, pages 168–184, 1986.
13. J.P. Helm. On effectively computable operators. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik (ZML)*, 17:231–244, 1971.
14. S. Jain. *Robust Behaviourally Correct Learning.* Technical Report TRA6/98 at the DISCS, National University of Singapore, 1998.
15. S. Jain, D. Osherson, J.S. Royer and A. Sharma. *Systems That Learn: An Introduction to Learning Theory.* MIT-Press, Boston, MA., 1999.
16. S. Jain, C. Smith and R. Wiehagen. On the power of learning robustly. In *Proceedings of Eleventh Annual Conference on Computational Learning Theory* (COLT), pages 187–197, ACM Press, New York, 1998.
17. E.B. Kinber and T. Zeugmann. Inductive inference of almost everywhere correct programs by reliably working strategies. *Journal of Information Processing and Cybernetics*, 21:91–100, 1985.
18. E.B. Kinber and T. Zeugmann. One-sided error probabilistic inductive inference and reliable frequency identification. *Information and Computation*, 92:253–284, 1991.
19. R. Klette and R. Wiehagen. Research in the theory of inductive inference by GDR mathematicians – A survey. *Information Sciences*, 22:149–169, 1980.
20. S. Kurtz and C.H. Smith. On the role of search for learning. In *Proceedings of the 2nd Annual Workshop on Computational Learning Theory* (R. Rivest, D. Haussler and M. Warmuth, Eds.) pages 303–311, Morgan Kaufman, 1989.
21. E. Minicozzi. Some natural properties of strong-identification in inductive inference. *Theoretical Computer Science*, 2:345–360, 1976.
22. P. Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
23. M. Ott and F. Stephan. Avoiding coding tricks by hyperrobust learning. *Proceedings of the Fourth European Conference on Computational Learning Theory* (EuroCOLT) (P. Fischer and H.U. Simon, Eds.) Lecture Notes in Artificial Intelligence Vol. 1572, pages 183–197, Springer-Verlag, Berlin, 1999.
24. H.Jr. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw–Hill, New York, 1967.
25. F. Stephan and T. Zeugmann. On the Uniform Learnability of Approximations to Non-Recursive Functions. DOI Technical Report DOI-TR-166, Department of Informatics, Kyushu University, July 1999.
26. T. Zeugmann. A-posteriori characterizations in inductive inference of recursive functions. *Journal of Information Processing and Cybernetics (EIK)*, 19:559–594, 1983.
27. T. Zeugmann. On the nonboundability of total effective operators. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik (ZML)*, 30:169–172, 1984.
28. T. Zeugmann. On Bārzdiņš' conjecture. In *Proceedings of the International Workshop on Analogical and Inductive Inference* (AII'86) (K.P. Jantke, Ed.), Lecture Notes in Computer Science Vol. 265, pages 220–227. Springer-Verlag, Berlin, 1986.