7-10-2003

On the Intrinsic Complexity of Learning Recursive Functions

Sanjay Jain
National University of Singapore

Efim Kinber
Sacred Heart University

Christophe Papazian
Ecole Normale Supérieure de Lyon

Carl Smith
University Of Maryland College Park

Rolf Wiehagen
Universitat Kaiserslautern

Recommended Citation

Kinber, E., Papazian, C., Smith, C., & Wiehagen, R. (1999). On the intrinsic complexity of learning recursive functions. *Proceedings of the Twelfth Annual Conference on Computational Learning Theory - COLT '99*. doi:10.1145/307400.307465

ACADEMIC PRESS

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Information and Computation 184 (2003) 45–70

Information and Computation

www.elsevier.com/locate/ic

On the intrinsic complexity of learning recursive functions

Sanjay Jain,[a] Efim Kinber,[b] Christophe Papazian,[c] Carl Smith,[d,*] and Rolf Wiehagen[e]

[a] *School of Computing, National University of Singapore, 119260 Singapore*
[b] *Computer Science Department, Sacred Heart University, Fairfield, CT 06432–1000, USA*
[c] *Département de Mathématique et d'Informatique, Ecole Normale Supérieure de Lyon, F-69364 Lyon Cedex 07, France*
[d] *Department of Computer Science, University of Maryland, College Park, MD 20742, USA*
[e] *Fachbereich Informatik, Universität Kaiserslautern, D-67653 Kaiserslautern, Germany*

Received 13 August 1999; revised 2 November 1999

Abstract

The intrinsic complexity of learning compares the difficulty of learning classes of objects by using some reducibility notion. For several types of learning recursive functions, both natural complete classes are exhibited and necessary and sufficient conditions for completeness are derived. Informally, a class is complete iff both its topological structure is highly complex while its algorithmic structure is easy. Some self-describing classes turn out to be complete. Furthermore, the structure of the intrinsic complexity is shown to be much richer than the structure of the mind change complexity, though in general, intrinsic complexity and mind change complexity can behave "orthogonally".

1. Introduction

The problem of learning infinite objects from growing finite samples of their behavior has attracted much attention in recent decades. In inductive inference the objects to be learned are recursive functions, i.e., computable functions being everywhere defined on the set $\mathbb{N}$ of natural numbers. The finite samples given to the learning machine are just initial segments of the infinite sequence of all the values of the corresponding function. The machine is said to learn that function if when fed increasing initial segments, it eventually produces a program of the corresponding function and never changes its mind thereafter.

[*] Corresponding author. Fax: 1-301-405-6707.

E-mail addresses: sanjay@comp.nus.edu.sg (S. Jain), KinberE@sacredheart.edu (E. Kinber), cpapazia@ens-lyon.fr (C. Papazian), smith@cs.umd.edu (C. Smith), wiehagen@informatik.uni-kl.de (R. Wiehagen).

0890-5401/03/$ - see front matter © 2003 Elsevier Science (USA). All rights reserved.
doi:10.1016/S0890-5401(03)00059-2

A machine learns a class of functions if it learns every function from that class. This is basically the concept of learning in the limit introduced in [13]. Other criteria for learning have been studied, see the surveys [2,4,7,12,15,21,24].

In studying any model of learning, two fundamental aspects must be addressed; the qualitative aspect, i.e., which object classes are learnable and which are not, and the quantitative aspect, i.e., how complex are the learning tasks. There has been prior work on trying to get at the complexity of learning, see for example [4–6,13,14]. Our work is different in that we use reducibilities as in both recursion theory [27] and complexity theory [11]. The main idea of the so-called intrinsic complexity introduced in [8] is to *compare* the complexity of learning problems. This is achieved by adopting some formal notion of *reducibility* between learning problems. Namely, if for classes U, V of recursive functions to be learned, U is reducible to V, then, informally, U is at most as hard to learn as V is. Clearly, with every notion of reducibility comes a notion of completeness. A class V is complete for some learning type, if all the classes U from that type are reducible to V. Our main goal consists of exhibiting natural classes which turn out to be complete and characterizing completeness. Surprisingly, the characteristic conditions do not depend much on the concrete learning type under consideration. Informally, these properties consist in being both “topologically complex” and “algorithmically easy”. On the one hand, it seems reasonable that high topological complexity can make learning difficult. On the other hand, the fact that high topological complexity has to be combined only with low algorithmic complexity may seem surprising and, in a sense, counterintuitive. We give some explanation of this fact below, when we will have the corresponding proofs at hand.

Furthermore, we study the relationship between intrinsic complexity and mind change complexity. Under some natural conditions greater mind change complexity is shown to imply greater intrinsic complexity. Moreover, these conditions are necessary to this end. In general, intrinsic complexity and mind change complexity behave “orthogonally” to each other.

In [16–18] the approach of intrinsic complexity was studied for language identification. The problem of characterizing complete classes was not addressed in these papers. There has been one prior study of reductions between learnable classes, see [25]. However, this approach differs from the approach of intrinsic complexity in a fundamental way (see [8] for a more detailed discussion). The rest of this paper is organized as follows. In Section 2 the necessary notation and definitions will be given, including the formalization of the approach of intrinsic complexity. In Section 3 some natural function class and its derivatives are shown to be complete for several learning types. Section 4 is devoted to derive the corresponding characterizations of completeness for these learning types. In Section 5 the intrinsic complexity is compared with the mind change complexity. Finally, in Section 6 we summarize our results and discuss some of their consequences as well as possible future work.

2. Preliminary definitions and notation

For sets A, B, $A \subseteq B$ and $A \subset B$ will denote inclusion and proper inclusion, respectively. $A \setminus B = \{x | x \in A, x \notin B\}$ denotes the difference of A and B. $\emptyset$ stands for the empty set. By $card\, A$, the cardinality of A will be denoted. For $A \subseteq \mathbb{N}$, $\max A$ and $\min A$ will stand for the maximum of A and the minimum of A, respectively. The set of all finite sequences of natural numbers is denoted by $\mathbb{N}^*$.

Let R denote the set of all (total) recursive functions of one argument. For $f \in R$ and $n \in \mathbb{N}$, let $f^n = (f(0), f(1), \ldots, f(n))$ be the initial segment of f up to n. For any functions $f, g \in R$ and $n \in \mathbb{N}$, let $f =_n g$ iff $f^n = g^n$, i.e., if f and g coincide up to n; and $f \neq_n g$ otherwise. Let $f^n \sqsubseteq g$ iff $f =_n g$. In this case we say also that the function g extends the initial segment f^n, or that g is an extension of f^n. Analogously, $f^n \sqsubseteq g^m$ iff $n = \min\{n, m\}$ and $f =_n g$. At several places below we will identify a recursive function with the infinite sequence of its values. Thus, for example, 0^∞ stands for the everywhere zero function and 0^i10^∞ stands for the function f such that $f(x) = 1$, if $x = i$, and $f(x) = 0$ otherwise. Furthermore, let $dist(f, g) = card\{x | f(x) \neq g(x)\}$ denote the *distance* between f and g. The following classes of recursive functions will be used frequently in the sequel:

- the class $FINSUP = \{f | f \in R,\ dist(f, 0^\infty) < \infty\}$ of the functions of finite support, i.e., the class of all recursive functions that have a non-zero value at no more than finitely many arguments,
- the subclasses $FINSUP_m = \{f | f \in R, 1 \leqslant dist(f, 0^\infty) \leqslant m + 1\}$ of $FINSUP$ for any $m \in \mathbb{N}$; note that $0^\infty \notin FINSUP_m$ by definition.

Let $U \subseteq R$ and $f \in R$. Then f is called an *accumulation point* of U iff for any $n \in \mathbb{N}$, there exists a function $g \in U$ such that $g =_n f$ but $g \neq f$. Notice that f can belong to U or not. U is called *dense* iff U is non-empty and, for any $f \in U$, f is an accumulation point of U. Clearly, any dense class must be infinite. U is called *discrete* iff U does not contain any accumulation point of U.

Discrete and dense classes are opposites of each other in the very strong sense that discrete sets contain *no* accumulation points and dense sets contain *only* accumulation points. However, one can show how to build large collections of accumulation points only from discrete sets. We proceed by example. Notice that we will rely on this example in Section 4.

Let F_i denote the largest subset of *FINSUP* containing only functions that have exactly $i + 1$ support points. The classes F_i are discrete. For example, F_0 contains the functions with exactly one support point and can be graphically represented in a fashion suggestive of an effective enumeration, $f_0, f_1, \ldots,$ see Fig. 1. Each row of Fig. 1 represents a function and the xth value in that row contains the value of the function on argument x. Notice the regularity of Fig. 1. For every x, there is a j such that

$$f_j(y) = \begin{cases} 1 & \text{if } y = x; \\ 0 & \text{otherwise.} \end{cases}$$

In fact, j is easily calculated from only x.

$$j = x + \sum_{k=1}^{x-1} k = \frac{x^2 + x}{2}.$$

We have used the convention that if $x - 1 < 1$ then the sum evaluates to 0. Let h denote the function that takes x to $(x^2 + x)/2$.

Suppose σ is the length n initial segment of 0^∞. Notice further that $f_{h(n)}$ agrees with σ, but $f_{h(n)} \neq 0^\infty$. We have just shown (even effectively) that 0^∞ is an accumulation point of F_0.

Suppose now that $k > 0$ and $f \in F_{k-1}$. Then f has exactly k support points. Let σ be the length $n + 1$ initial segment of f, where n is an arbitrary natural number. Then σ has exactly i support points, for some $i \leqslant k$. Define $g = \sigma 1^{k-i+1} 0^\infty$. So g has the i support points from σ and exactly $k - i + 1$ others. Hence, $g \in F_k$. Furthermore, $g =_n f$ by definition of g. Finally, $g \neq f$, since f has fewer support points. Consequently, f is an accumulation point of F_k. Since f was chosen somewhat arbitrarily, we have established that, for all $k > 0$, any function in F_{k-1} is an accumulation point of F_k.

A non-empty class $U \subseteq R$ is called *recursively enumerable* (r.e.) iff there is a universal recursive function u of U; i.e., u is a recursive function of two arguments enumerating exactly the class U, $\{u_i | i \in$

$\mathbb{N}\} = U$ where $u_i = \lambda x u(i, x)$. Then u is called a (recursive) *numbering* of U. Furthermore, if $f \in U$ and $u_i = f$ then i is called a u-index (or a u-number) of f. Note that every infinite r.e. class possesses a *one-one* numbering u, i.e., a numbering such that $u_i \neq u_j$ for any $i \neq j$, see [20]. Clearly, *FINSUP* and all the classes $FINSUP_m,\ m \in \mathbb{N}$, are r.e.

Let φ be any acceptable programming system or, equivalently, any Gödel numbering of all the partial recursive functions of one argument, see [22,26,28]. The natural numbers will then serve as names for programs, and φ_i will denote the function computed by program i. As above, i is called a φ-index or a φ-number of a function f iff $\varphi_i = f$. We will use φ as the basic hypothesis space for all the learning types below. Note that this allows the learning machines to work subsequently in other suitable hypothesis spaces such as in recursive numberings u as well, since any u-index can be effectively translated into an equivalent φ-index.

Gold, in a seminal paper [13], defined the notion called *identification in the limit*. This definition concerned learning by algorithmic devices now called *inductive inference machines* (IIMs). An IIM inputs the graph of a recursive function, an ordered pair at a time, and, while doing so, outputs computer programs. Since we will only discuss the inference of (total) recursive functions, we may assume, without loss of generality, that the input is received by an IIM in its natural domain increasing order, $f(0), f(1), \ldots$ On input from a function f, an IIM $\mathtt{M}$ will output an infinite sequence of programs $p_0 = \mathtt{M}(f^0)$, $p_1 = \mathtt{M}(f^1), \ldots$ The IIM *converges* iff there is a program p such that for all but finitely many i, $p_i = p$. Then we say that the IIM converges to p. In general, there is no effective way to tell when, and if, an IIM has converged.

F_0	0	1	2	3	4	5	$\cdots$
f_0	1	0	0	0	0	0	$\cdots$
f_1	0	1	0	0	0	0	$\cdots$
f_2	2	0	0	0	0	0	$\cdots$
f_3	0	0	1	0	0	0	$\cdots$
f_4	3	0	0	0	0	0	$\cdots$
f_5	0	2	0	0	0	0	$\cdots$
f_6	0	0	0	1	0	0	$\cdots$
f_7	4	0	0	0	0	0	$\cdots$
f_8	0	3	0	0	0	0	$\cdots$
f_9	0	0	2	0	0	0	$\cdots$
f_{10}	0	0	0	0	1	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Fig. 1. The class F_0.

Following Gold, we say that an IIM $\mathtt{M}$ EX-identifies a function f (written: $f \in EX(\mathtt{M})$), if, when $\mathtt{M}$ is given the graph of f as input, it converges to a program p that computes f, i.e., $\varphi_p = f$. More formally, $\mathtt{M}$ is an operator which takes as input the function f, and outputs the sequence $(\mathtt{M}(f^n))_{n\in\mathbb{N}}$ of programs, denoted by $\mathtt{M}(f)$.

Each IIM will learn some set of recursive functions. The collection of all such sets, over the universe of effective algorithms viewed as IIMs, serves as a characterization of the learning power inherent in the Gold model. This collection is symbolically denoted by EX and is defined rigorously by $EX = \{U | U \subseteq R,\ \exists \mathtt{M}(U \subseteq EX(\mathtt{M}))\}$. Note that any r.e. class of recursive functions belongs to EX, see [13].

Now let $a \in \mathbb{N}$. Then we say that an IIM $\mathtt{M}$ EX^a-identifies a function $f \in R$ (written: $f \in EX^a(\mathtt{M})$) iff the sequence $\mathtt{M}(f)$ converges to a program p such that $dist(\varphi_p, f) \leqslant a$; i.e., EX^a-learning allows final hypotheses with at most a anomalies. Let $EX^a = \{U | U \subseteq R,\ \exists \mathtt{M}(U \subseteq EX^a(\mathtt{M}))\}$.

We say that an IIM $\mathtt{M}$ EX^*-identifies a function $f \in R$ (written: $f \in EX^*(\mathtt{M})$) iff $\mathtt{M}(f)$ converges to a program p such that $dist(\varphi_p, f) < \infty$; i.e., EX^*-learning allows final hypotheses with an arbitrary finite number of anomalies. Let $EX^* = \{U | U \subseteq R,\ \exists \mathtt{M}(U \subseteq EX^*(\mathtt{M}))\}$.

In order to define learning with a bounded number of mind changes, notice that without loss of generality we can allow an IIM to output a special symbol ? for a while at the beginning of the learning process. ? can be interpreted as “I don’t know yet”. Clearly, this does not change the limit of the corresponding sequence $\mathtt{M}(f)$. On the other hand, this can save one unnecessary mind change, namely the very first one, which could be forced by requiring to make $\mathtt{M}(f^0)$ a real hypothesis from $\mathbb{N}$. Also without loss of generality, we can assume that $\mathtt{M}$ after producing a first hypothesis from $\mathbb{N}$ will never output ? again. Actually, by simply repeating its actual hypothesis, $\mathtt{M}$ can avoid undesired mind changes without outputting ? again. This way we can also ensure that $\mathtt{M}$ is defined on all possible input segments just by outputting either ? or a hypothesis from $\mathbb{N}$. Now, for any $m \in \mathbb{N}$, we say that an IIM $\mathtt{M}$ EX_m-identifies a function $f \in R$ (written: $f \in EX_m(\mathtt{M})$) iff $\mathtt{M}(f)$ converges to a program p such that $\varphi_p = f$ and $card\{n | ? \neq \mathtt{M}(f^n) \neq \mathtt{M}(f^{n+1})\} \leqslant m$; i.e., on the function f, the machine $\mathtt{M}$ changes its mind no more than m times. Let $EX_m = \{U | U \subseteq R,\ \exists \mathtt{M}(U \subseteq EX_m(\mathtt{M}))\}$.

Theorem 2.1 shows the relationships between the identification types defined above.

Theorem 2.1 [4].
1. $EX_0 \subset EX_1 \subset \cdots \subset EX_m \subset EX_{m+1} \subset \cdots \subset EX$
2. $EX = EX^0 \subset \cdots \subset EX^a \subset EX^{a+1} \subset \cdots \subset EX^*$

In this paper we will be concerned only with the identification types EX, EX^a, EX^*, and EX_m as defined above. Subsequently, we let $\mathcal{I}$ stand for any one of these types.

Proposition 2.1. *There exists an r.e. sequence $\mathtt{M}_0, \mathtt{M}_1, \mathtt{M}_2, \ldots$, of inductive inference machines such that, for any identification type $\mathcal{I}$ considered in this paper, for all $\mathcal{C} \in \mathcal{I}$, there exists an $i \in \mathbb{N}$ such that $\mathcal{C} \subseteq \mathcal{I}(\mathtt{M}_i)$.*

Ref. [15] shows the above for $\mathcal{I} = EX$. Essentially, the same proof can be used for all $\mathcal{I}$ considered in this paper. We assume $\mathtt{M}_0, \mathtt{M}_1, \mathtt{M}_2, \ldots$ to be one such sequence of machines.

In the following we need the notion of admissible sequences of hypotheses as introduced in [8]. Informally, for an identification type $\mathcal{I}$, an $\mathcal{I}$-admissible sequence for a recursive function f is a sequence of hypotheses which is “successful” when learning f in the sense of $\mathcal{I}$. For example, an EX-admissible sequence for $f \in R$ is any sequence of programs $p_0, p_1, \ldots$ converging to some p such that $\varphi_p = f$. Clearly, using this notion one could redefine the identification type EX as follows: $U \in EX$ iff there is an IIM $\mathtt{M}$ such that for any function $f \in U$, $\mathtt{M}(f)$ is an EX-admissible sequence for f. For the other identification types $\mathcal{I}$, the notion of $\mathcal{I}$-admissible sequences is defined analogously.

Besides the notion of admissible sequences we need yet the concept of recursive operators in order to give the basic definition of intrinsic complexity.

Definition 2.1 [27]. A *recursive operator* is an effective total mapping, Θ, from (possibly partial) functions to (possibly partial) functions, which satisfies the following properties:

(a) Monotonicity: For all functions η, η', if $\eta \subseteq \eta'$ then $\Theta(\eta) \subseteq \Theta(\eta')$.

(b) Compactness: For all η, if $(x, y) \in \Theta(\eta)$, then there exists a finite function $\alpha \subseteq \eta$ such that $(x, y) \in \Theta(\alpha)$.

(c) Recursiveness: For all finite functions α, one can effectively enumerate (in α) all $(x, y) \in \Theta(\alpha)$.

In this paper we are concerned with the behavior of Θ on total functions only. Thus, without loss of generality, in (c) above we may additionally assume that $\Theta(\alpha)$ is finite for all finite α, and one can effectively determine $card\, \Theta(\alpha)$ (in addition to being able to enumerate $\Theta(\alpha)$).

We now present some easy results which will be used several times in the sequel. These results show that to some extent, recursive operators preserve the structure of the classes they map. As it will be clear from Lemma 2.1, structure can mean both algorithmic and topological structure. The proof of this lemma is obvious and therefore omitted.

Lemma 2.1. *Let U be any class of recursive functions, and let Θ be any recursive operator mapping every function from U to a recursive function, i.e. $\Theta(U) \subseteq R$. Then:*

1. *If U is r.e., then $\Theta(U)$ is r.e.*
2. *If $h \in U$ is an accumulation point of U and Θ is injective, then $\Theta(h)$ is an accumulation point of $\Theta(U)$.*
3. *If U is not discrete and Θ is injective, then $\Theta(U)$ is not discrete.*
4. *If U is dense and Θ is injective, then $\Theta(U)$ is dense.*

On the other hand, recursive operators can map discrete classes to non-discrete classes as well. Actually, let $U = \{0^i 10^\infty | i \in \mathbb{N}\}$. Then U is both r.e. and discrete. Define a recursive operator Θ as follows: $\Theta(10^\infty) = 0^\infty$, and $\Theta(0^i 10^\infty) = 0^i 10^\infty$ for any $i > 0$. Then Θ is injective and $\Theta(U)$ is not discrete, since this class contains its accumulation point 0^∞.

We now come to the basic definition of intrinsic complexity.

Definition 2.2. Suppose $\mathcal{I}$ is an identification type and $U, V \in \mathcal{I}$. Then U is said to be $\mathcal{I}$-reducible to V (written: $U \leqslant_{\mathcal{I}} V$) iff there exist recursive operators Θ and Ξ such that for any function $f \in U$,

1. $\Theta(f) \in V$,
2. for any $\mathcal{I}$-admissible sequence σ for $\Theta(f)$, $\Xi(\sigma)$ is an $\mathcal{I}$-admissible sequence for f.

Notice that unlike [8] in the definition above we do not require Θ to be injective. This is due to the fact that in learning with anomalies one and the same sequence can be admissible for infinitely many functions. Consequently, there is no ultimate need for the operator Θ to be injective. On the other hand, for the other identification types $\mathcal{I}$ considered in this paper, $\mathcal{I}$-reducibility of U to V by operators Θ and Ξ obviously implies the injectivity of Θ.

Intuitively, if U is $\mathcal{I}$-reducible to V then V is at least as difficult to learn in the sense of $\mathcal{I}$ as U is. Actually, for any IIM $\mathtt{M}$ that $\mathcal{I}$-learns V, one can easily construct an IIM $\mathtt{M}'$ that $\mathcal{I}$-learns U as follows: $\mathtt{M}'(f) = \Xi(\mathtt{M}(\Theta(f)))$. Consequently, in that sense V is *most* difficult for $\mathcal{I}$-learning if *all* classes $U \in \mathcal{I}$ are $\mathcal{I}$-reducible to V.

For an identification type $\mathcal{I}$, a class $V \subseteq R$ is called *$\mathcal{I}$-complete* iff $V \in \mathcal{I}$ and any class $U \in \mathcal{I}$ is $\mathcal{I}$-reducible to V.

For an identification type $\mathcal{I}$ and classes $U, V \in \mathcal{I},\ U$ and V are said to be *$\mathcal{I}$-comparable*, *$\mathcal{I}$-incomparable*, and *$\mathcal{I}$-equivalent*, respectively, iff (1) $U \leqslant_{\mathcal{I}} V$ or $V \leqslant_{\mathcal{I}} U$, (2) neither $U \leqslant_{\mathcal{I}} V$ nor $V \leqslant_{\mathcal{I}} U$, (3) $U \leqslant_{\mathcal{I}} V$ and $V \leqslant_{\mathcal{I}} U$, respectively. Finally, $U <_{\mathcal{I}} V$ iff $U \leqslant_{\mathcal{I}} V$, but not $V \leqslant_{\mathcal{I}} U$.

3. Natural examples of complete classes

In this section we will prove some natural classes complete for the types EX, EX^a, EX^*, and EX_m. Notice that these classes essentially differ from the so-called cylinder classes which were used in [8] in order to construct a complete class for a given identification type $\mathcal{I}$ in some uniform way. Informally, to get such a cylinder class for a type $\mathcal{I}$ each recursive function was combined with all the IIMs which were capable to learn that function in the sense of $\mathcal{I}$. Thus, these cylinder classes very directly depend on the learning type under consideration. In contrast, our classes presented below are not distinguished by such a close and direct relationship to the corresponding learning type. Actually, all these classes come from a common natural source, namely the class *FINSUP* of the functions of finite support, that is, all functions $f \in R$ such that $card\{x | f(x) \neq 0\} < \infty$, or, equivalently, $dist(f, 0^\infty)$ is finite. This class itself was shown to be EX-complete in [8].

Theorem 3.1. [8] *FINSUP is EX-complete.*

We show that the same class *FINSUP* is complete for all the types EX^a, $a \in \mathbb{N}$, as well. Thus, surprisingly, EX contains a class being complete for all the EX^a, despite the fact that EX is a smaller and smaller subset of the EX^a. For the types EX^* and $EX_m,\ m \in \mathbb{N}$, we then prove natural modifications of *FINSUP* to be complete.

Theorem 3.2. *For any $a \in \mathbb{N},\ FINSUP$ is EX^a-complete.*

Proof. Let $a \in \mathbb{N}$. Then $FINSUP \in EX^a$, since *FINSUP* is r.e., hence $FINSUP \in EX$, and $EX \subseteq EX^a$. In order to show that every class from EX^a can be reduced to *FINSUP*, we need the following partitioning of the set of all natural numbers into consecutive intervals of length $2a + 1$. For any $n \in \mathbb{N}$, let X_n^a denote the set $\{(2a+1)n, (2a+1)n+1, \ldots, (2a+1)n+2a\}$ of cardinality $2a+1$. Clearly, $X_n^a \cap X_{n'}^a = \emptyset$ if $n \neq n'$. Furthermore, $\bigcup_{n \geqslant 0} X_n^a = \mathbb{N}$. Now let U be an arbitrary class from EX^a. Let $\mathtt{M}$ be any IIM that identifies U in EX^a-style. Without loss of generality assume that $\mathtt{M}$ does not output ? on any input. Then, for any $x \in \mathbb{N}$, find the only $n \in N$ such that $x \in X_n^a$ and define

$$\Theta(f)(x) = \begin{cases} \mathtt{M}(f^n) + 1 & \text{if } n = 0 \text{ or } \mathtt{M}(f^n) \neq \mathtt{M}(f^{n-1}); \\ 0 & \text{if } n \geqslant 1 \text{ and } \mathtt{M}(f^n) = \mathtt{M}(f^{n-1}). \end{cases}$$

Clearly, Θ is a recursive operator mapping U to *FINSUP*, since the machine $\mathtt{M}$ changes its mind only finitely often on any function $f \in U$.

Intuitively, Θ encodes the sequence of hypotheses produced by $\mathtt{M}$ on f into the function $\Theta(f)$ in a way that is "robust with respect to anomalies". Actually, even if at most a among the $2a+1$ consecutively encoded hypotheses will be "destroyed" by the maximum of anomalies allowed in EX^a-learning the function $\Theta(f)$, the remaining $a+1$ "undestroyed" hypotheses will keep the majority. This in turn suggests the following definition of the operator Ξ. Let σ be any EX^a-admissible sequence for a function $\Theta(f)$ where $f \in U$. Then $\Xi(\sigma)$ can be defined as follows:

- Search for the limit of σ, say j; note that $dist(\varphi_j, \Theta(f)) \leqslant a$, hence $\varphi_{j(x)} = 0$ for almost all $X \in \mathbb{N}$, since $\Theta(f) \in$ *FINSUP*,
- search for the maximal $n \in \mathbb{N}$ such that within the interval X_n^a, the function φ_j takes a non-zero value at least $a+1$ times; note that this interval corresponds to the final hypothesis produced by $\mathtt{M}$ on f,
- find the only $y > 0$ such that within X_n^a, the function φ_j takes that value y on at least $a+1$ arguments,
- converge to $y-1$.

Clearly, $y-1$ is just the final hypothesis produced by the IIM $\mathtt{M}$ on the function f. Consequently, the class U is EX^a-reducible to *FINSUP* by the operators Θ and Ξ. □

As we have seen above, the intervals X_n^a of *finite* length $2a+1$ were sufficient to overcome the difficulties caused by the anomalies within the final hypotheses of EX^a-learning, when a is a fixed number. Now, for EX^*-learning, we will need intervals of *infinite* length instead. In the following definition, the sets of powers of the prime numbers will play this role of intervals of infinite length. Therefore let *QUASIFINSUP* denote the set of all recursive functions f such that:

1. For every $x \in \mathbb{N}$, if x is not a power of a prime number, then $f(x) = 0$.
2. For all but finitely many prime numbers p, $f(p^k) = 0$ for all $k \in \mathbb{N}$.
3. For every prime number p, there are $y, n \in \mathbb{N}$ such that either

$$f(p^k) = y \quad \text{for all } k \geqslant 1,$$

or

$$f(p^k) = \begin{cases} y & \text{if } 1 \leqslant k \leqslant n; \\ 0 & \text{otherwise.} \end{cases}$$

Thus, for any sequence $p, p^2, p^3, \ldots$, either the values of f are equal on all arguments from the sequence, or they are equal to a non-zero number on the arguments from a finite initial segment of the sequence and are zero on the arguments from the rest of the sequence.

Note that *QUASIFINSUP* is an r.e. class.

Theorem 3.3. *QUASIFINSUP is EX^*-complete.*

Proof. Obviously, *QUASIFINSUP* is EX^*-learnable, as *QUASIFINSUP* is r.e. Now let U be an arbitrary class from EX^*. Let $\mathtt{M}$ be any IIM that EX^*-learns U. Without loss of generality assume that $\mathtt{M}$ does not output ? on any input. Let p_i denote the ith prime number, where $p_0 = 2$. Then, for any function $f \in U$ and any $x \in \mathbb{N}$, define an operator Θ as follows:

$$\Theta(f)(x) = \text{“Let } i \leqslant x \text{ be the maximal number such that } i = 0 \text{ or } \mathtt{M}(f^i) \neq \mathtt{M}(f^{i-1}).$$
$$\text{If } x = p_i^k \text{ for some } k \geqslant 1, \text{ then let } \Theta(f)(x) = \mathtt{M}(f^i) + 1. \text{ Otherwise, let } \Theta(f)(x) = 0\text{”}.$$

Clearly, Θ is a recursive operator mapping U to *QUASIFINSUP*. Note that for any function $f \in U$, there is exactly one number i such that $\Theta(f)(x) \neq 0$ for all $x = p_i^k$, $k \geqslant 1$, namely just that i where the machine M makes its last mind change on the function f. Moreover, by the definition of Θ, for every other prime $p \neq p_i$, there can be at most finitely many arguments $x = p^k$ with $\Theta(f)(x) \neq 0$. Finally, by the definition of *QUASIFINSUP*, for all but finitely many primes p, $\Theta(f)(x) = 0$ for all $x = p^k$, $k \geqslant 1$. Hence, the number of non-zero values of $\Theta(f)$ on the interval of the powers of p_i will eventually exceed the corresponding number of non-zero values of $\Theta(f)$ on any other interval of prime powers. Clearly, this property remains valid for every function which differs from $\Theta(f)$ on at most finitely many arguments, i.e., especially, for every function φ_j where j is the limit of any EX^*-admissible sequence for $\Theta(f)$. This suggests the following definition of the operator Ξ. Let σ be any EX^*-admissible sequence for a function $\Theta(f)$ where $f \in U$. Then $\Xi(\sigma)$ can be defined as follows:

- Search for the limit of σ, say j; note that $dist(\varphi_j, \Theta(f))$ is finite; hence φ_j has the property mentioned above,
- search for the only number i such that on the arguments $x = p_i^k$, $k \geqslant 1$, the function φ_j takes more non-zero values than on the arguments $x = p^k$ for any other prime $p \neq p_i$,
- find the value y that will be taken by φ_j on all but finitely many arguments $x = p_i^k$, $k \geqslant 1$,
- converge to $y - 1$.

Clearly, $y - 1$ is just the hypothesis the machine M converges to on the function f. Consequently, the class U is EX^*-reducible to *QUASIFINSUP* by the operators Θ and Ξ. □

In order to exhibit classes which are complete for EX_m, $m \in \mathbb{N}$, we will modify the standard definition of EX_m-completeness by defining the notion of “EX-completeness for EX_m”.

Definition 3.1. Let $m \in \mathbb{N}$. A class V of recursive functions is called EX-complete for EX_m iff $V \in EX_m$ and, for any class $U \in EX_m$, U is EX-reducible to V.

Informally, by definition, EX-completeness for EX_m allows all the admissible sequences to be of “EX-style” rather than “EX_m-style”, thus giving the reducing operators Ξ some more freedom.

Now, recall that for any m, *FINSUP*$_m$ denotes the subclass of *FINSUP* consisting of all functions which contain at least one and at most $m + 1$ non-zero points; formally,

$$FINSUP_m = \{f \mid f \in FINSUP,\ 1 \leqslant card\{x \mid f(x) \neq 0\} \leqslant m + 1\}.$$

Recall that each of the classes *FINSUP*$_m$ is r.e.

Theorem 3.4. *For any $m \in \mathbb{N}$, FINSUP$_m$ is EX-complete for EX_m.*

The proof of Theorem 3.4 is pretty analogous to the proof of Theorem 3.1 (see [8]) and therefore omitted.

4. Characterizations of complete classes

Now we are going to characterize completeness for all the identification types EX, EX^a, EX^*, and EX_m, where $a, m \in \mathbb{N}$.

Theorem 4.1. *For any class $U \in EX$, U is EX-complete iff U contains an r.e. dense subclass.*

Proof. Necessity: Let U be EX-complete. Then *FINSUP* is EX-reducible to U by recursive operators Θ and Ξ. Clearly, $\Theta(FINSUP) \subseteq U$ and Θ is injective. Moreover, *FINSUP* is r.e. and dense. Consequently, $\Theta(FINSUP)$ is r.e. and dense by Lemma 2.1.

Sufficiency: Let $U \in EX$ contain the r.e. dense subclass V. Then it suffices to show that *FINSUP* is EX-reducible to V. Actually, since *FINSUP* is EX-complete by Theorem 3.1, this would imply that both V and U are EX-complete as well. We even prove a somewhat more general result namely, that *any* infinite r.e. class is EX-reducible to V. Thus, let W be any infinite r.e. class, and let w be any one-one numbering of W. For any $i, j, i \neq j$, let x_{ij} denote the least number x such that $w_i(x) \neq w_j(x)$. Furthermore, let v be any one-one numbering of V. Then inductively define the operator Θ as follows:

$\Theta(w_0) = v_0;$

and for any $i > 0$,

$\Theta(w_i) =$ "Search for the least w-index $k < i$ such that w_k is "most similar" to w_i, i.e., $x_{ik} = \max\{x_{ij} \mid j < i\}$.
Then search for the least v-index m such that both

- v_m is "sufficiently similar" to $\Theta(w_k)$, i.e., $v_m =_{x_{ik}} \Theta(w_k)$;
  Comment: This ensures the monotonicity of Θ.
- v_m is not in the present range of Θ, i.e., $v_m \notin \Theta(\{w_0, \dots, w_{i-1}\})$;
  Comment: This ensures the injectivity of Θ. Note that v_m must exist, since V is dense. Moreover, v_m can effectively be found, since V is r.e.

Define $\Theta(w_i) = v_m$".

Clearly, Θ is a recursive operator mapping W injectively to V. Moreover, the following claim implied by the definition of Θ will be useful to define the second operator Ξ.

Claim A. Given any v-index m such that $v_m \in \Theta(W)$, one can effectively find the corresponding w-index i such that $\Theta(w_i) = v_m$.

Proof of Claim A. Consecutively, for $i = 0, 1, 2, \dots,$ look at the functions $\Theta(w_i)$, as defined above, until the right w-index i has been found. □

In order to define the operator Ξ we need yet another technical algorithm which, in the limit, allows us to translate φ-indices into equivalent v-indices.

Claim B. For any r.e. set S of recursive functions, given any numbering s of S and any φ-index j such that $\varphi_j \in S$, one can effectively produce a sequence of s-indices converging to an s-index of φ_j.

Proof of Claim B. Given the numbering s, the φ-index j and any $n \in \mathbb{N}$, define $m_n = \min\{i | i \leqslant n, s_i =_n \varphi_j\} \cup \{n\}$. Thus, s_{m_n} is the first function among $s_0, s_1, \ldots, s_n$, if any, which coincides with φ_j up to n. Clearly, m_n can be found effectively, since φ_j and all the s_i, $i \in \mathbb{N}$, are recursive functions. Moreover, obviously, the sequence $(m_n)_{n \in \mathbb{N}}$ converges to the least s-index of the function φ_j. □

We now are ready to define the operator Ξ. Therefore, let σ be any EX-admissible sequence of φ-indices for any function $\Theta(f)$, $f \in W$. Then $\Xi(\sigma)$ can be defined as follows:

- Search for the limit of σ, say j; note that $\varphi_j = \Theta(f)$,
- in the limit, find a v-index m of φ_j by applying the algorithm from Claim B,
- find the w-index i such that $\Theta(w_i) = v_m$ by applying the algorithm from Claim A,
- converge to a φ-index of w_i.

Clearly, since $v_m = \varphi_j = \Theta(f)$, we get $w_i = f$ by the injectivity of Θ. This completes the proof that W is EX-reducible to V by the operators Θ and Ξ. □

Next, we will present a characterization of EX^a-complete classes for any $a \in \mathbb{N}$. Therefore, we need an additional property of r.e. dense classes, namely that all the functions of such a class can be chosen to have sufficiently large distance from each other. This property will enable the reducing operators to deal with the anomalies allowed in EX^a-learning. The following lemma just states that all r.e. dense classes possess this property.

Lemma 4.1. *For any $d \in \mathbb{N}$, any r.e. dense class contains an r.e. dense subclass where all the distinct functions have distance at least d.*

Proof. We start with the following Claim which, intuitively, states that given an r.e. dense class W and a function $f \in W$, one can find another function $g \in W$ such that g both "arbitrarily much" coincides with f and "arbitrarily much" differs from f.

Claim. Let W be any r.e. dense class and let w be any numbering of W. Then, given any w-index i and any $n, m \in \mathbb{N}$, one can effectively find a w-index j such that $w_j =_n w_i$ and $dist(w_j, w_i) \geqslant m$.

Proof of Claim. Let W and w as above. Let $i, n \in \mathbb{N}$ be arbitrarily fixed; set $f = w_i$. Then we proceed by induction on m. Obviously, for $m = 0$, $j = i$ suffices.

Now, by induction, suppose that for every $p < m$, $m \geqslant 1$, one can effectively find a w-index j_p such that $w_{j_p} =_n f$ and $dist(w_{j_p}, f) \geqslant p$. Let $h = w_{j_{m-1}}$. Clearly, $h =_n f$ and $dist(h, f) \geqslant m - 1$. Let $x \geqslant n$ be the least argument such that $card\{y | y \leqslant x,\ h(y) \neq f(y)\} \geqslant m - 1$. Since $h \in W$ and W is dense, there must be a function $g \in W$ such that $g =_x h$ and $g \neq h$. Moreover, since W is r.e., a w-index of one such g can easily be found by searching for the least $j \in \mathbb{N}$ such that $w_j =_x h$ and $w_j \neq h$. Clearly, $dist(g, f) \geqslant m$ or $dist(h, f) \geqslant m$. Furthermore, it is straightforward to effectively fix one of the functions g or h (and, hence, a w-index of this function) with the desired distance property. This completes the proof of the Claim. □

Now let W be any r.e. dense class, and let w be any numbering of W. Furthermore, let $d \in \mathbb{N}$ be given. Then we can inductively define a numbering v of a subclass V of W with the desired properties as follows.

Let $(i_s)_{s\in\mathbb{N}}$ denote the sequence $0, 0, 1, 0, 1, 2, 0, 1, 2, 3, \ldots$ in which every natural number occurs infinitely often. Notice that $i_s < s$ for any $s \geqslant 1$.

Stage 0.
Define $v_0 = w_0$, and go to stage 1.

Stage $s, s \geqslant 1$.
By induction assume that for any v-index $i < s$, a w-index i' can effectively be computed such that $v_i = w_{i'}$. Also by induction assume that for any distinct $i, j < s$, $dist(v_i, v_j) \geqslant d$. Then effectively find an arbitrary $n \geqslant s$ such that for any distinct $i, j < s$, $card\{x|x \leqslant n, v_i(x) \neq v_j(x)\} \geqslant d$, i.e., on their initial segments of length n the functions enumerated by v so far have pairwise distance at least d. Now, by applying the Claim above, search for a w-index j such that

$$w_j =_n w_{(i_s)'} (= v_{i_s}) \text{ and } dist(w_j, w_{(i_s)'}) \geqslant d.$$

Define $v_s = w_j$, and go to stage $s + 1$.

Clearly, $V = \{v_i|i \in \mathbb{N}\}$ is an r.e. subclass of W. Moreover, by the choice of both n and v_s, each new function v_s has distance at least d, from all the previously enumerated functions $v_i, i < s$, i.e., including v_{i_s}. This proves the desired distance property of V. Finally, by the choice of the sequence $(i_s)_{s\in\mathbb{N}}$, arbitrarily large initial segments of each function $v_i, i \in \mathbb{N}$, will be extended. More exactly, for each function v_i and any $x \in \mathbb{N}$, there is some stage s such that $i_s = i$, $v_s =_x v_i$ and $v_s \neq v_i$. Hence v_i is an accumulation point of V and, consequently, V is dense. □

Then we get the following characterization of EX^a-completeness.

Theorem 4.2. *For any $a \in \mathbb{N}$ and for any class $U \in EX^a$, U is EX^a-complete iff U contains an r.e. dense subclass.*

Proof. Necessity: Let U be any EX^a-complete class, $a \in \mathbb{N}$. Let V be any r.e. dense class such that for any distinct functions $f, g \in V$, $dist(f, g) > 2a$. Note that V exists by Lemma 4.1. Clearly, $V \in EX \subseteq EX^a$, since V is r.e. Consequently, V is EX^a-reducible to U by some operators Θ and Ξ. We claim that Θ has to be injective. Actually, otherwise distinct functions $f, g \in V$ with $\Theta(f) = \Theta(g)$ would exist. Hence, for each EX^a-admissible sequence for the function $\Theta(f) = \Theta(g)$, the operator Ξ had to construct a sequence being EX^a-admissible for both functions f and g. But this is impossible, since $dist(f, g) > 2a$ by the definition of V. This contradiction proves Θ to be injective. Hence $\Theta(V) \subseteq U$ is r.e. and dense by Lemma 2.1.

Sufficiency: Let $U \in EX^a$, $a \in \mathbb{N}$, contain an r.e. dense subclass W. Then, by Theorem 3.2, it suffices to show that *FINSUP* is EX^a-reducible to W. In order to do this, we need to map the functions from *FINSUP* to an r.e. dense *subclass* V of W where for any distinct functions $g, h \in V$, $dist(g, h) > 2a$. Note that such a class V exists by Lemma 4.1. Informally, this additional property of V will enable the second operator Ξ to identify the functions $f \in FINSUP$ from any EX^a-admissible sequence for the function $\Theta(f)$. Formally, we need this property in the proof of Claim B.

For the following, let w denote any one-one numbering of *FINSUP*, and let v denote any one-one numbering of V. Then, since V is both r.e. and dense, the operator Θ can be defined in the same way as in the proof of Theorem 4.1; again, Θ is injective. Also, Claim A of that proof remains valid, even with exactly the same proof.

Claim A. Given any v-index m such that $v_m \in \Theta(FINSUP)$, one can effectively find the corresponding w-index i such that $\Theta(w_i) = v_m$.

Proof of Claim A. Consecutively, for $i = 0, 1, 2, \ldots,$ look at the functions $\Theta(w_i)$, as defined above, until the right w-index i has been found. □

However, in contrast to the sufficiency proof of Theorem 4.1, the operator Ξ now gets only EX^a-admissible sequences for the functions $\Theta(f)$ rather than EX-admissible ones. Hence, we have to modify Claim B in the following way.

Claim B. Given any φ-index j such that $dist(\varphi_j, g) \leqslant a$ for some $g \in V$, one can effectively produce a sequence of v-indices converging to the v-index of g.

Proof of Claim B. First note that the function $g \in V$ above is unique due to the property that the distance of any distinct functions from V exceeds $2a$. Second note that the function φ_j may be undefined on at most a arguments. This leads to the following algorithm which, intuitively, keeps any v-index s until it will be clear that $dist(\varphi_j, v_s) > a$, and hence v_s must differ from the function g.

“Go to stage 0.
Stage s, $s \geqslant 0$.
Output s. Check if there are at least $a + 1$ arguments x such that $\varphi_j(x)$ is defined and
$\varphi_j(x) \neq v_s(x)$, in which case go to stage $s + 1$”.

Clearly, due to the distance property of V, the algorithm above will converge to the v-index of g. □

Then the operator Ξ can be defined analogously to the sufficiency proof of Theorem 4.1 . Therefore, let σ be any EX^a-admissible sequence of φ-indices for any function $\Theta(f)$, $f \in FINSUP$. Then define $\Xi(\sigma)$ as follows:

- Search for the limit of σ, say j; note that $dist(\varphi_j, g) \leqslant a$ for some unique function $g \in V$,
- in the limit, find the v-index m of g by applying the algorithm from Claim B,
- find the w-index i such that $\Theta(w_i) = v_m$ by applying the algorithm from Claim A,
- converge to a φ-index of w_i.

Clearly, since $g = v_m = \Theta(w_i)$, we get $w_i = f$ by the injectivity of Θ. Consequently, *FINSUP* is EX^a-reducible to V (and hence to W) by the operators Θ and Ξ. □

In order to characterize EX^*-complete classes in an analogous way as done in Theorem 4.2 for EX^a-complete classes, we would need the following strengthening of Lemma 4.1: Any r.e. dense class contains an r.e. dense subclass where all the distinct functions have *infinite* distance. However, in general, this strengthening is not valid. Actually, just the r.e. and dense class *FINSUP* provides a counterexample, since all the functions from *FINSUP* have *finite* distance from each other. On the other hand, this infinite distance property turns out to be really necessary in order to deal with EX^*-admissible sequences. Consequently, we have to insert this property directly into the characterization.

Theorem 4.3. *For any class $U \in EX^*$, U is EX^*-complete iff U contains an r.e. dense subclass where all the distinct functions have infinite distance.*

Proof. Necessity: Let U be any EX^*-complete class. Then, by Theorem 3.3, *QUASIFINSUP* is EX^*-reducible to U by some operators Θ and Ξ. Let Q denote the subclass of *QUASIFINSUP* such that for any function $f \in Q$ and any prime p, the function f takes the same value on all arguments p^m, $m \geqslant 1$. We claim that $\Theta(Q)$ will be the desired subclass of U. In order to show this note that Q is r.e. and dense. Moreover, all the distinct functions from Q have infinite distance.

Claim. For any distinct functions $f, g \in Q$, $dist(\Theta(f), \Theta(g)) = \infty$.

Proof of Claim. Assume to the contrary that for some distinct functions $f, g \in Q$, $dist(\Theta(f), \Theta(g))$ is finite. Then there is a sequence σ which is EX^*-admissible for both functions $\Theta(f)$ and $\Theta(g)$. Hence, by definition, $\Xi(\sigma)$ has to converge to some index of a function with finite distance from both f and g. But this is impossible, since $f \in Q$ and $g \in Q$ are of infinite distance. This contradiction completes the proof. □

The Claim above immediately implies that both the operator Θ is injective on Q and all the distinct functions from $\Theta(Q)$ have infinite distance. Moreover, $\Theta(Q)$ is r.e. and dense by Lemma 2.1. Thus, $\Theta(Q)$ is the desired subclass of U.

Sufficiency: Informally, this proof follows the same line as the sufficiency proof of Theorem 4.2 replacing the distance bound of $2a$ from that proof by the infinite distance property. The latter leads to some modification of both the statement of Claim B below and its proof. Furthermore, *QUASIFINSUP* is used rather than *FINSUP*, of course.

Let V be an r.e. dense subclass of U where all the distinct functions have infinite distance. Let v be a one-one numbering of V. It suffices to show that *QUASIFINSUP* is EX^*-reducible to V. Let w denote any one-one numbering of *QUASIFINSUP*. Define the operator Θ mapping *QUASIFINSUP* to V as in the sufficiency proof of Theorem 4.2, and hence as in the sufficiency proof of Theorem 4.1. Then Θ is injective.

Claim A. Given any v-index m such that $v_m \in \Theta(QUASIFINSUP)$, one can effectively find the corresponding w-index i such that $\Theta(w_i) = v_m$.

Proof of Claim A. Consecutively, for $i = 0, 1, 2, \ldots,$ look at the functions $\Theta(w_i)$, as defined above, until the right w-index i has been found. □

Claim B. Given any φ-index j such that $dist(\varphi_j, g) < \infty$ for some $g \in V$, one can effectively produce a sequence of v-indices converging to the v-index of g.

Proof of Claim B. Note that the function $g \in V$ is unique, since all the distinct functions from V have infinite distance. Moreover, the function φ_j may be undefined on at most finitely many arguments. This leads to the following algorithm which, intuitively, comes back to each v-index i arbitrarily often and keeps the present index i as long as no further point of difference between v_i and φ_j will be developed. Therefore, let $(i_s)_{s\in\mathbb{N}}$ denote the sequence $0, 1, 0, 1, 2, 0, 1, 2, 3, \ldots$ in which every natural number occurs infinitely often. For any $i \in \mathbb{N}$, set $X_i = \emptyset$; intuitively, X_i will denote the set of all arguments x developed so far on which v_i and φ_j are different. Then the algorithm can be defined as follows.

"Go to stage 0.

Stage s, $s \geqslant 0$.
Output i_s. Check if there is an $x \notin X_{i_s}$ such that $\varphi_j(x)$ is defined and $\varphi_j(x) \neq v_{i_s}(x)$, in which case set $X_{i_s} = X_{i_s} \cup \{x\}$ and go to stage $s+1$".

Now let m denote the only v-index of the function g. Let i be an arbitrary number such that $i \neq m$. Since $dist(v_i, v_m) = \infty$, each stage s with $i_s = i$ entered by the algorithm above will eventually be left. Hence the algorithm must reach some stage s such that $i_s = m$ and X_{i_s} already contains all the arguments x such that $\varphi_j(x)$ is defined and $\varphi_j(x) \neq v_m(x)$. Consequently, stage s will never be left, and the algorithm converges to m. □

Now let σ be any EX^*-admissible sequence for any function $\Theta(f)$, $f \in$ *QUASIFINSUP*. Then define $\Xi(\sigma)$ as follows:

- Search for the limit of σ, say j; note that $dist(\varphi_j, g)$ is finite for exactly one function $g \in V$,
- in the limit, find the v-index m of g by applying the algorithm from Claim B,
- find the w-index i such that $\Theta(w_i) = v_m$ by applying the algorithm from Claim A,
- converge to a φ-index of w_i.

Since $g = v_m = \Theta(w_i)$, we get $w_i = f$ by the injectivity of Θ. Hence *QUASIFINSUP* is EX^*-reducible to V by the operators Θ and Ξ. □

Finally, we will characterize EX-completeness for EX_m. Therefore we have to modify the notion of density in the following way.

Definition 4.1. Let $U \subseteq R$ and $m \in \mathbb{N}$. Then U is called m-dense iff there are pairwise disjoint infinite classes $U_0, U_1, \ldots, U_m$ such that $\bigcup_{i \leqslant m} U_i = U$, and, for any $i < m$, each function from U_i is an accumulation point of U_{i+1}.
Furthermore, if U is r.e., then $U_0, U_1, \ldots, U_m$ are r.e. as well.

A typical example for an m-dense class is just FINSUP$_m$, see the discussion around Fig. 1 in Section 2.

Theorem 4.4. *For any $m \in \mathbb{N}$ and any class $U \in EX_m$, U is EX-complete for EX_m iff U contains an r.e. m-dense subclass.*

For the proof of Theorem 4.4, the reader is referred to [19].

The characterization given by Theorem 4.4 is especially easy for $m = 0$, namely: For any class $U \in EX_0$, U is EX-complete for EX_0 iff U contains an infinite r.e. subclass. This follows immediately from Theorem 4.4 and the definition of m-density.

We now want to point out a consequence of our completeness characterizations above, namely that there are classes which are both complete and "self-describing". Actually, in [8] it was proved that the standard self-describing class $S = \{f \mid f \in R, \varphi_{f(0)} = f\}$, i.e., the class of all recursive functions that, on argument 0, return a program computing themselves, is not EX-complete. However, this class is EX-complete for EX_0. Indeed, by the Recursion Theorem [27], there is a function $g \in R$ such that for any $i \in \mathbb{N}$, $\varphi_{g(i)} = g(i)i^\infty$. Hence, the class $\{\varphi_{g(i)} \mid i \in \mathbb{N}\}$ is an infinite r.e. subclass of S. Consequently, S is EX-complete for EX_0. Moreover, classes which are both complete and self-describing exist at the bottom of the mind change hierarchy, and also within *every* level including the very top level of unbounded mind changes. In order to see the latter point, consider the class $C = \{\alpha i p | \alpha \in \mathbb{N}^*, i \geqslant 2, p \in R_{0,1}, \varphi_i = \alpha i p\}$, where $R_{0,1}$ denotes the set of all 0–1-valued functions from R. Clearly, C is a self-describing class; in order to EX-learn this class a learning machine has only to find the last value $i \geqslant 2$ and to converge to this self-describing value. Moreover, C is dense. This easily follows from the fact that C is "initially complete", i.e., for every initial segment $\alpha \in \mathbb{N}^*$, there is a function in C that is consistent with α. Obviously, every initially complete class is dense. Furthermore, one can show that C is not contained in any recursively enumerable class (otherwise, $R_{0,1}$ would be contained in an r.e. class, a contradiction). Nevertheless, C contains a subclass D which is both r.e. and dense. In order to see this note that by use of the Recursion Theorem, for any $\alpha \in \mathbb{N}^*$, one can uniformly construct an $i_\alpha \in \mathbb{N}$ such that $\alpha i_\alpha 0^\infty \in C$. Hence the class $D = \{\alpha i_\alpha 0^\infty | \alpha \in \mathbb{N}^*\}$ is r.e. Moreover, D is initially complete and hence dense. Consequently, C is EX-complete by Theorem 4.1. Analogously, one can show that for any $m \geqslant 1$, the EX_m-versions C_m of the class C above, i.e., $C_m = \{f | f \in C, card\{x | f(x) \geqslant 2\} \leqslant m + 1\}$, are EX-complete for EX_m.

5. Intrinsic complexity versus mind change complexity

In this section we mainly want to explore how intrinsic complexity and mind change complexity relate to each other. First we will show that under some natural conditions, greater mind change complexity implies greater intrinsic complexity, Theorem 5.1. However, greater mind change complexity does not always imply greater intrinsic complexity, Theorem 5.2. Thus, mind change complexity and intrinsic complexity are in a sense "orthogonal". Nevertheless, further results, Theorems 5.3 and 5.4, then yield that the structure of the intrinsic complexity is much richer than the quasi-linearly ordered structure of the mind change complexity. Informally, these theorems state that on any levels of the mind change hierarchy, there are classes which are intrinsically "unrelated". Finally, we prove that in general, maximal mind change complexity does not imply maximal intrinsic complexity, Theorem 5.5. This is due to the fact that high mind change complexity does not always imply high topological complexity, as it would be necessary for maximal intrinsic complexity by the results of Section 4. Notice that all these results above can be shown for *recursively enumerable* function classes, and hence for "natural" classes.

Note that for mind changes bounded by *ordinal* numbers (see [9]), there are results similar to those announced above, but we do not include them here because of the technical machinery which would be necessary to state and to prove these results. We only want to mention one fact in this respect. In [1] it is shown (within the framework of learning languages from text) that a class is learnable with an ordinal mind change bound if this class is learnable by a machine which converges on any input sequence, even on non-computable ones. Using this result one can prove that no dense class, in particular no complete class, can be learned with an ordinal mind change bound. Thus, mind change bounds are always a sign for incompleteness.

In order to state and to prove the first result of this section we need a few more notions. Suppose $U \in EX$. If $U \in \bigcup_{m \geqslant 0} EX_m$, then define $mcc(U) = \min\{m | U \in EX_m\}$, where mcc stands for "mind change complexity". If $U \in EX \setminus \bigcup_{m \geqslant 0} EX_m$, then let $mcc(U) = *$. By definition, $m < *$ for any $m \in \mathbb{N}$. Now, let V be any non-empty class of recursive functions. The class V is called *bounded* iff for any function $f \in V$ and any $n \in \mathbb{N}$, there are at most finitely many values $y \in \mathbb{N}$ such that $f^n y \sqsubseteq g$ for some function $g \in V$; furthermore an upper bound on such y can be effectively found from f^n. That is, for any initial segment f^n from V, one can effectively determine a b such that, for any $y > b$, no extension of $f^n y$ belongs to V. For example, all classes of predicates (i.e., functions taking only the values 0, 1) are bounded. Note that, in definition of bounded class above, we allow $\{f(0)|f \in V\}$ to be unbounded. Though this is not crucial for our results, it makes some of the proofs easier. Another consequence of a class being bounded is that the tree formed by this class is of bounded degree (except, possibly, at the root). This allows to apply König's Lemma, as we will do below.

The class V is said to be *closed* iff V contains all of its accumulation points. Equivalently, for any function $f \in R$, if every initial segment $f^n, n \in \mathbb{N}$, can be extended to a function from V, then f itself must belong to V as well. For example, all of the classes $FINSUP_m, m \in \mathbb{N}$, are not closed, since, by definition, they all do not contain their accumulation point 0^∞. However, $FINSUP_0 \cup \{0^\infty\}$ is closed. Also *FINSUP* is not closed, since *every* recursive function is an accumulation point of *FINSUP*; hence, the "closure" of *FINSUP* would be the set R of all recursive functions.

Finally, we call the class V *decidable* iff the set $\{f^n|f \in V \text{ and } n \in \mathbb{N}\}$ of all the initial segments of functions from V is decidable. In other words, for an arbitrary initial segment, one can effectively find out if there is a function from the class which extends that initial segment.

Theorem 5.1. *For any EX-comparable classes U, V such that V is bounded, closed, and decidable, if $mcc(U) < mcc(V)$ then $U <_{EX} V$.*

Proof. Let U, V be any EX-comparable classes such that V is bounded, closed, and decidable. For simplicity we below assume that V is "binarily bounded", i.e., V is a class of predicates ($f(x) \in \{0, 1\}$ for any $f \in V$ and any $x \in \mathbb{N}$). Using the decidability of V the proof can straightforwardly be generalized to arbitrary bounded classes. Furthermore, let $mcc(U) < mcc(V)$. Consequently, for some $m \in \mathbb{N}, U \in EX_m$ and $V \in EX \backslash EX_m$. Since U, V are EX-comparable, it suffices to prove that $V \leqslant_{EX} U$ does not hold. Assume to the contrary that $V \leqslant_{EX} U$ by operators Θ and Ξ; clearly, Θ must be injective. We will show that this implies $V \in EX_m$, a contradiction.

Suppose that U is EX_m-identified by an IIM $\mathtt{M}$. Without loss of generality we may assume that $\mathtt{M}$ makes no more than m mind changes on *any* input function. Consider the tree $\mathcal{T}$ formed by all the initial segments of functions from V (including the empty segment λ which forms the root of $\mathcal{T}$) where each initial segment f^n, $f \in V$ and $n \in \mathbb{N}$, is represented by a node in $\mathcal{T}$. Since V is closed, V consists of all the infinite branches in $\mathcal{T}$. Also, there is no leaf in $\mathcal{T}$, i.e., every node in $\mathcal{T}$ has at least one child. We call a node f^n in $\mathcal{T}$ *marked* iff $\mathtt{M}(\Theta(f^n)) \neq \mathtt{M}(\Theta(f^{n-1}))$; for $n = 0$, f^0 will be marked iff $\mathtt{M}(\Theta(f^0)) \neq ?$.

Claim A. For any node f^n in $\mathcal{T}$ and any distinct extensions $g, h \in V$ of f^n, at least one of the nodes from $\{g^r|r > n\} \cup \{h^r|r > n\}$ is marked.

Proof of Claim A. Otherwise $\mathtt{M}$ would not EX-identify at least one of the distinct (by the injectivity of Θ) functions $\Theta(g)$ and $\Theta(h)$, a contradiction to $U \in EX$. □

Claim B. Any infinite branch of $\mathcal{T}$ has at most $m + 1$ marked nodes.

Proof of Claim B. This follows immediately from the hypothesis that $\mathtt{M}$ changes its mind at most m times. □

The next claim informally says that all branches on at least one side of any node f^n in $\mathcal{T}$ cause a mind change before some $n' > n$. For proving this claim, we need that V is bounded, since we apply König's Lemma.

Claim C. For any node f^n in $\mathcal{T}$, there exists $n' > n$ such that $\{h^{n'} | h^{n'}$ in $\mathcal{T}$, $f^n 0 \sqsubseteq h^{n'}$ and no one of $h^{n+1}, h^{n+2}, \ldots, h^{n'}$ is marked $\} = \emptyset$
or
$\{h^{n'} | h^{n'}$ in $\mathcal{T}$, $f^n 1 \sqsubseteq h^{n'}$ and no one of $h^{n+1}, h^{h+2}, \ldots, h^{n'}$ is marked $\} = \emptyset$.

Proof of Claim C. Otherwise, by König's Lemma, both $f^n 0$ and $f^n 1$ are extended by infinite branches in $\mathcal{T}$ with no marks beyond f^n. This contradicts Claim A. □

Now, define a function *Prog* such that for any marked f^n in $\mathcal{T}$, the following holds:

1. For $x \leqslant n$, $\varphi_{Prog(f^n)}(x) = f(x)$.
2. Suppose $\varphi_{Prog(f^n)}(x)$ has been defined for all $x \leqslant y$ (below let $\varphi^y_{Prog(f^n)}$ be denoted by g^y). Search for $n' > y$ such that
   (2.1) $\{h^{n'} | h^{n'}$ in $\mathcal{T}$, $g^y 0 \sqsubseteq h^{n'}$ and no one of $h^{y+1}, h^{y+2}, \ldots, h^{n'}$ is marked $\} = \emptyset$, or
   (2.2) $\{h^{n'} | h^{n'}$ in $\mathcal{T}$, $g^y 1 \sqsubseteq h^{n'}$ and no one of $h^{y+1}, h^{y+2}, \ldots, h^{n'}$ is marked $\} = \emptyset$.
   In case (2.1) let $\varphi_{Prog(f^n)}(y) = 1$, and in case (2.2) let $\varphi_{Prog(f^n)}(y) = 0$.
   Intuitively, *Prog* chooses the branch (if any) which does not seem to cause any mind change.
3. Go to step 2.

Clearly, by the decidability of V, the function *Prog* is computable.

Claim D. For any f^n in $\mathcal{T}$ and any $g \in V$ extending f^n such that no node from $\{g^{n'} | n' > n\}$ is marked in $\mathcal{T}$, $\varphi_{Prog(f^n)} = g$.

Proof of Claim D. This easily follows from Claim C by induction. □

Now define an IIM $\mathtt{M}'$ as follows:

$$\mathtt{M}'(f^n) = \begin{cases} Prog(f^x), & \text{where } x \leqslant n \text{ is the maximal number (if any)} \\ & \text{such that } f^x \text{ is marked in } \mathcal{T} \\ ? & \text{otherwise.} \end{cases}$$

Clearly, by Claim B, on any function $g \in V$, $\mathtt{M}'$ outputs at most $m + 1$ hypotheses, thus making at most m mind changes. Moreover, by Claim D, the final hypothesis output by $\mathtt{M}'$ on g is a φ-program for g. Consequently, V is EX_m-identified by $\mathtt{M}'$, a contradiction to $V \notin EX_m$. □

Roughly, Theorem 5.1 says that under some natural conditions greater mind change complexity implies greater intrinsic complexity. On the other hand, the conditions provided by Theorem 5.1 (boundedness, closedness, and decidability) turn out to be really necessary, as we will see below. In other words, in general, greater mind change complexity does not imply greater intrinsic complexity. Actually, there are classes U, V such that $mcc(U) < mcc(V)$ but $V \leqslant_{EX} U$; see Theorem 5.2. Thus, mind change complexity and intrinsic complexity are in a sense "orthogonal" to each other.

Theorem 5.2. *There are classes $U, V \in EX$ such that $mcc(U) < mcc(V)$ but $V \leqslant_{EX} U$.*

Proof. Let $(\mathtt{M}_i)_{i\in\mathbb{N}}$ denote an effective enumeration of IIM's as in Proposition 2.1. First, we will define the class V by some explicit diagonalization procedure yielding among others that $mcc(V) > 1$. More exactly, V will be the union of classes $V_i, i \in \mathbb{N}$, where each V_i is defined by uniformly diagonalizing against the machine $\mathtt{M}_i$. Then the class U will be defined just as $\Theta(V)$, where Θ simultaneously serves as the first operator realizing the EX-reduction of V to $U = \Theta(V)$. Moreover, this definition will yield $mcc(U) = 1$.

Definition of $V_i,\ i \in \mathbb{N}$.

$$V_i = \begin{cases} \{(i+4)^\infty\} & \text{if for all } k \in \mathbb{N}, \mathtt{M}_i((i+4)^k) = ? \\ \\ \{(i+4)^k 0^\infty\} \cup & \text{if } k \geqslant 1 \text{ is the least number such that for} \\ \{(i+4)^k 0^r 1^\infty | r \geqslant 1\} \cup & \text{some } j \in \mathbb{N}, \mathtt{M}_i((i+4)^k) = j \text{ and, for} \\ \{(i+4)^k 1^r 0^\infty | r \geqslant 1\} & \text{this } j, \varphi_j(k) \text{ is undefined or } \varphi_j(k) \neq 0 \\ \\ \{(i+4)^k 1^\infty\} \cup & \text{if } k \geqslant 1 \text{ is the least number such that for} \\ \{(i+4)^k 0^r 1^\infty | r \geqslant 1\} \cup & \text{some } j \in \mathbb{N}, \mathtt{M}_i((i+4)^k) = j \text{ and, for} \\ \{(i+4)^k 1^r 0^\infty | r \geqslant 1\} & \text{this } j, \varphi_j(k) = 0. \end{cases}$$

Let $V = \bigcup_{i\geqslant 0} V_i$.

Notice that V is r.e. by construction. Hence $V \in EX$.

Now we define an operator Θ as follows:

$\Theta((i+4)^k) = (i+4)^k$,

$\Theta((i+4)^k 0^r) = \Theta((i+4)^k 1^r) = (i+4)^k 0^r$, for any $r \in \mathbb{N}$.

Comment: This definition will not violate the injectivity of Θ on the class V, since by definition, for any $i \in \mathbb{N}$, at most one of the functions $(i+4)^k 0^\infty$ and $(i+4)^k 1^\infty$ belongs to V. On the other hand, mapping both functions $(i+4)^k 0^\infty$ and $(i+4)^k 1^\infty$ to the only function $(i+4)^k 0^\infty$ just allows to save one mind change in learning $\Theta(V)$ compared to learning V itself, as it follows from the proof of Claim B below.

$\Theta((i+4)^k 0^r 1^s) = (i+4)^k 0^r 2^s$, for any $r, s \geqslant 1$.

$\Theta((i+4)^k 1^r 0^s) = (i+4)^k 0^r 3^s$, for any $r, s \geqslant 1$.

Clearly, Θ is a recursive operator which injectively maps the class V to some class $\Theta(V) \subseteq R$. Define $U = \Theta(V)$.

Claim A. $V \leqslant_{EX} U$

Proof of Claim A. Let Θ be the operator from the definition of U. The operator Ξ is defined as follows. Let σ be any EX-admissible sequence for a function $g \in U$ where $g = \Theta(f)$, $f \in V$. Then, by definition, $\varphi_w = g$ where w is the limit of σ. On σ the operator Ξ searches for w (and finds w in the limit) and evaluates the function φ_w in order to identify the corresponding function f where g is "coming from". The following cases are possible:

Case 1. φ_w behaves as $(i+4)^k$, $k \geqslant 1$.

Then Ξ outputs a fixed φ-program for $(i+4)^\infty$.

Case 2. φ_w behaves as $(i+4)^k 0^r 2^s$, $r, s \geqslant 1$.
Then Ξ outputs a fixed φ-program for $(i+4)^k 0^r 1^\infty$.
Case 3. φ_w behaves as $(i+4)^k 0^r 3^s$, $r, s \geqslant 1$.
Then Ξ outputs a fixed φ-program for $(i+4)^k 1^r 0^\infty$.
Case 4. φ_w behaves as $(i+4)^k 0^r$, $r \geqslant 1$.
Then Ξ outputs a fixed φ-program for $(i+4)^k 0^\infty$. In parallel, Ξ checks if $\varphi_{\mathtt{M}_i((i+4)^k)}(k) = 0$ in which case Ξ stops outputting the program for $(i+4)^k 0^\infty$ and outputs a fixed φ-program for $(i+4)^k 1^\infty$ instead. Comment: Case 4 occurs if $f = (i+4)^k 0^\infty$ or $f = (i+4)^k 1^\infty$. The additional check allows Ξ to find out the "right" function f.

Clearly, the operators Θ and Ξ witness $V \leqslant_{EX} U$. □

Claim B. $mcc(U) \leqslant 1$

Proof of Claim B. For any $i \in \mathbb{N}$, let p_i denote a φ-program such that

$$\varphi_{p_i} = \begin{cases} (i+4)^\infty & \text{if} \quad \mathtt{M}_i((i+4)^k) = ? \text{ for all } k \in \mathbb{N}, \\ (i+4)^k 0^\infty & \text{if} \quad k \geqslant 1 \text{ is the least number such that} \\ & \qquad \mathtt{M}_i((i+4)^k) \neq ? \end{cases}$$

Now, for all $k, r, s \geqslant 1$, define an IIM $\mathtt{M}$ as follows:
$\mathtt{M}((i+4)^k) = p_i$,
$\mathtt{M}((i+4)^k 0^r) = p_i$,
$\mathtt{M}((i+4)^k 0^r 2^s) =$ fixed φ-program for $(i+4)^k 0^r 2^\infty$,
$\mathtt{M}((i+4)^k 0^r 3^s) =$ fixed φ-program for $(i+4)^k 0^r 3^\infty$.
Then, on any function from U, the machine $\mathtt{M}$ makes at most one mind change. Furthermore, by the definition of U and the p_i's, $\mathtt{M}$ clearly EX-identifies U. Hence $U \in EX_1$ and $mcc(U) \leqslant 1$. □

Obviously, $mcc(U) = 1$, but for proving the theorem $mcc(U) \leqslant 1$ suffices.

Claim C. $mcc(V) > 1$.

Proof of Claim C. Assume to the contrary that $V \in EX_1(\mathtt{M}_i)$ for some $i \in \mathbb{N}$. Then there must be a least number $k \geqslant 1$ such that $\mathtt{M}_i((i+4)^k) \neq ?$, since otherwise $\mathtt{M}_i$ would not identify the function $(i+4)^\infty \in V_i \subseteq V$. Let $j = \mathtt{M}_i((i+4)^k)$. Now the following cases are possible.
Case 1. $\mathtt{M}_i((i+4)^k 0^r) \neq \mathtt{M}_i((i+4)^k)$ for some r.
Then $\mathtt{M}_i$ fails to EX_1-identify all but at most one of the functions from $\{(i+4)^k 0^t 1^\infty | t \geqslant r\} \subseteq V$.
Case 2. Not Case 1, i.e., $\mathtt{M}_i((i+4)^k 0^r) = \mathtt{M}_i((i+4)^k)$ for all $r \in \mathbb{N}$.
Case 2.1. $\varphi_j(k)$ is undefined or $\varphi_j(k) \neq 0$.
Then $\mathtt{M}_i$ fails to EX-identify the function $(i+4)^k 0^\infty \in V$.
Case 2.2. $\varphi_j(k) = 0$.
Then $\mathtt{M}_i$ either fails to EX-identify the function $(i+4)^k 1^\infty \in V$, or (in case $\mathtt{M}_i$ makes a mind change on $(i+4)^k 1^s$ for some $s \geqslant 1$) $\mathtt{M}_i$ fails to EX_1-identify all but at most one of the functions from $\{(i+4)^k 1^t 0^\infty | t \geqslant s\}$. □

The theorem immediately follows from Claims A, B, and C. □

We proceed with some remarks concerning Theorem 5.2 and its consequences. First notice that the classes U, V were built “near the bottom” of the EX_m-hierarchy. However, by correspondingly modifying the proof, the same effect can be proved on any higher level of this hierarchy. Next notice that the class V and, by Lemma 2.1, also the class $U = \Theta(V)$ are both r.e. Hence these classes are in a sense “natural”. Furthermore, note that the class V is bounded and decidable, but not closed. In other words, V misses *exactly one* of the conditions from Theorem 5.1, which results in that Theorem 5.1 does not remain valid then. Analogously, we can show that Theorem 5.1 will no longer be true, if one of the other two conditions is violated. Thus, the three conditions provided by Theorem 5.1, i.e., boundedness, closedness, and decidability of V, turn out to be really necessary for making this result hold. Hence, in this sense, Theorem 5.1 is optimal.

Notice that Theorem 5.1 is also “optimal” concerning another detail, namely in assuming that the classes U, V are EX-comparable. Actually, in general, $mcc(U) < mcc(V)$ does not imply $U \leqslant_{EX} V$ (even if V is bounded, closed, and decidable), as our next result will show immediately for “arbitrarily different” levels of the EX_m-hierarchy.

Theorem 5.3. *For any $m \in \mathbb{N}$, there exist r.e. classes $V \in EX_{m+1} \setminus EX_m$ and $U \in EX_1$ such that U is not EX-reducible to V.*

Proof. Let $m \in \mathbb{N}$. Let $(\mathtt{M}_i)_{i\in\mathbb{N}}$ be an effective enumeration of IIMs as in Proposition 2.1. Then, for any $i \in \mathbb{N}$, by straightforward diagonalization against $\mathtt{M}_i$, one can uniformly generate a finite class V_i of recursive functions such that

1. $card\, V_i \leqslant m + 2$,
2. for any $f \in V_i$, $f(0) = i$ and $dist(f, i^\infty) \leqslant m + 1$,
3. $V_i \nsubseteq EX_m(\mathtt{M}_i)$.

Let $V = \bigcup_{i\in\mathbb{N}} V_i$. Obviously, V is r.e. and $V \in EX_{m+1} \setminus EX_m$. Note that V does not possess any accumulation point. Now consider the class $U = \{0^r 10^\infty | r \in \mathbb{N}\} \cup \{0^\infty\}$. Clearly, U is r.e. and $U \in EX_1$. However, U contains its accumulation point 0^∞. Consequently, U is not EX-reducible to V by Lemma 2.1. □

Clearly, since the classes V_i from the proof of Theorem 5.3 above are all finite and functions from different V_i differ on input 0, the class $V = \bigcup_{i\geqslant 0} V_i$ is bounded, closed, and decidable, thus fulfilling these conditions of Theorem 5.1. Nevertheless, this does not guarantee that $U \leqslant_{EX} V$ holds. In a sense, Theorem 5.3 may be interpreted in that the structure of the intrinsic complexity is much richer than the quasi-linearly ordered structure of the mind change complexity. Our next result emphasizes this point of view by stating that each level $m > 0$ of the mind change hierarchy contains r.e. classes which turn out to be EX-incomparable.

Theorem 5.4. *For any $m > 0$, there are r.e. EX-incomparable classes $U, V \in EX_m \setminus EX_{m-1}$.*

Proof. We will consider only the $m = 1$ case . The construction can be easily extended to any $m > 1$. Let $U = \{0^i 10^\infty | i \in \mathbb{N}\} \cup \{0^\infty\}$. Obviously, U is r.e. Furthermore, $U \in EX_1$ and $U \notin EX_0$. The latter follows from the fact that U is not discrete, since it contains its accumulation point 0^∞. On the other hand, any class from EX_0 must be discrete, see [10]. Now, let $K = \{x|\varphi_x(x) \text{ is defined}\}$ be the halting set. Let $V = \{k0^\infty|k \in \mathbb{N}\} \cup \{k0^i10^\infty|k \in K, \varphi_k(k) \text{ halts in exactly } i \text{ steps}\}$. Clearly, V is both r.e. and discrete. Moreover, $V \in EX_1$ and $V \notin EX_0$, where the latter follows by a similar argument as below. Actually, any IIM that would EX_0-learn the class V could be used to decide K, a contradiction. Since V is discrete while U is not, U cannot be EX-reduced to V by Lemma 2.1 (Clause 3). Now assume to the contrary that V is EX-reducible to U. Let Θ be the first reducing operator. As Θ is injective, we have $\Theta(k0^\infty) \neq 0^\infty$ for all but, may be, one $k \in \mathbb{N}$. Then the following algorithm can be used to decide the halting set K (except maybe for one element k): “For any $k \in \mathbb{N}$, find the minimal $j \in \mathbb{N}$ such that $\Theta(k0^j) = 0^i1$ for some $i \in \mathbb{N}$. Then $k \in K$ iff $\varphi_k(k)$ terminates in at most j steps”. Indeed, if $\varphi_k(k)$ would terminate in $t > j$ steps then, by the monotonicity of Θ, $\Theta(k0^t10^\infty) = \Theta(k0^\infty)$, a contradiction to the injectivity of Θ. Consequently, V is not EX-reducible to U. □

In a sense, Theorem 5.4 shows that, as far as learning functions from finite samples is concerned, “poor” topological structure (the class V is discrete) and relatively “high” algorithmic complexity (the class V simulates the halting problem) cannot be traded for “rich” topological structure (the class U contains an accumulation point) and “low” algorithmic complexity (the class U has a computable numbering that provides complete finite descriptions for every function in the class).

Note that Theorem 5.4 does not remain valid for EX_0. Actually, recall that all infinite r.e. classes from EX_0 are EX-complete for EX_0, as it directly follows from Theorem 4.4. Thus, these classes are equivalent rather than incomparable. Consequently, EX_0 is a “singular point” among all the identification types EX_m, $m \in \mathbb{N}$, with respect to the properties exhibited by Theorems 5.3 and 5.4.

Finally, we want to point out another difference between mind change complexity and intrinsic complexity. Informally, we will show that in general, maximal mind change complexity does not imply maximal intrinsic complexity. More exactly, we will exhibit a class U from $EX \setminus \bigcup_{m\geqslant 0} EX_m$, i.e., with unbounded and hence “maximal” mind change complexity. Moreover, U itself is r.e. On the other hand, U is far from being of maximal intrinsic complexity, i.e., far from being EX-complete. Actually, U does not only not contain any dense subclass (as it would be necessary for being EX-complete by Theorem 4.1), but U even does not contain (even does not possess!) any accumulation point.

Theorem 5.5. *There is an r.e. class U of recursive functions such that*
1. $U \in EX\setminus \bigcup_{m\geqslant 0} EX_m$,
2. *U is not EX-complete.*

Proof. We need a slight generalization of the idea used in the proof of Theorem 5.3. Therefore, let $\langle .,. \rangle$ denote any effective bijection of all the pairs of natural numbers onto the natural numbers. Let $(\mathtt{M}_i)_{i\in\mathbb{N}}$ be any effective enumeration of the IIMs as in Proposition 2.1. Then, for any $i, m \in \mathbb{N}$, by straightforward diagonalization against the IIM $\mathtt{M}_i$, one can uniformly generate a finite class $U_{i,m}$ of recursive functions such that

(a) $card\, U_{i,m} \leqslant m + 2$,
(b) for any function $f \in U_{i,m}$, $f(0) = \langle i, m\rangle$ and $dist(f, \langle i, m\rangle^\infty) \leqslant m + 1$,
(c) $U_{i,m} \notin EX_m(\mathtt{M}_i)$.

Now define $U = \bigcup_{i\geqslant 0, m\geqslant 0} U_{i,m}$.

Clearly, U is r.e., since all of the finite classes $U_{i,m}$ are uniformly effectively generable. Hence $U \in EX$, by [13]. On the other hand, due to condition (c), U cannot be EX_m-learnable for any $m \in \mathbb{N}$.

Moreover, by condition (b), for every natural number y $(= \langle i, m\rangle$ for some unique $i, m \in \mathbb{N})$, there are only finitely many functions $f \in U$ such that $f(0) = y$. Hence the class U does not possess any accumulation point. Thus, U cannot contain a dense subclass. Consequently, by Theorem 4.1, U cannot be EX-complete. □

On the other hand, maximal intrinsic complexity always implies maximal mind change complexity. Actually, it is very easy to show that any dense class cannot be identified with a bounded number of mind changes.

6. Conclusions

We wanted to find out what makes classes of recursive functions most difficult to learn or, in a formal sense, complete within the framework of intrinsic complexity [8]. Informally, the characteristic properties for completeness consist in being both “topologically complex” and “algorithmically easy”. Here topologically complex means being dense, i.e., consisting of accumulation points only, and algorithmically easy means being recursively enumerable. Actually, the common structure of all of our completeness characterizations is roughly the following: A class is complete if and only if this class contains a recursively enumerable dense subclass.

On the one hand, it seems intuitively clear that the density of a class can make this class difficult to learn. It is well-known that not only density but already the presence of a *single* accumulation point can make learning impossible at all. Actually, this is true for learning superfinite classes of languages in the limit from text (see [13]), as well as for finite learning of recursive functions (see [10]).

On the other hand, we feel that some explanation is needed for the counterintuitive fact that the property of high topological complexity has to be combined with the property of low algorithmic complexity, namely recursive enumerability, in order to yield completeness. Here recursive enumerability is said to be a property of low algorithmic complexity, since being r.e. is kind of being “well-structured” rather than more or less “unstructured”, what one could expect when the most difficult learning situations are to be described. One might argue that density yields already so much difficulty that it needs some compensation in order to keep the class learnable at all. But this is at best half the truth, since, clearly, the recursive enumerability of the characteristic subclass ensures the learnability of *this subclass*, but in general this cannot guarantee the learnability of the *whole* complete class. A better explanation stems from the fact that our complete example classes from Section 3 have to be reducible to an arbitrary complete class. Recall that all these concrete complete classes are r.e. themselves. Hence, by Lemma 2.1 (Clause 1), any reducing operator transmits this recursive enumerability to some subclass of any complete class; moreover, by Lemma 2.1 (Clause 4), any injective reducing operator transmits the density to the same subclass of the corresponding complete class. A final, substantial explanation of the fact above is the following. In order to be a complete class, every class from the corresponding learning type must be mapped to this class by some recursive operator. But to be able to do this the operator must “know” which functions of the complete class it may choose as its “targets”. One possibility of formalizing this “knowledge” consists just in getting the operator acquainted with a suitable recursively enumerable subclass of the complete class. Thus, informally, knowing such an r.e. subclass enables the operator "to hit the target". Conversely, one can provide technical evidence that the *non*-existence of such an r.e. subclass results just in "missing the target". Actually, by (non-effectively) diagonalizing against all the recursive operators, one can construct a subclass of *FINSUP* which, though still being dense, cannot be complete, since no injective recursive operator is able to map even the trivial class of all constant functions to that class. Interestingly, a similar effect of being both topologically difficult and algorithmically easy has been exhibited recently in the framework of classifying languages, see [3,29].

A consequence of our completeness characterizations is the fact that there are classes which are both complete and "self-describing". Recall that it was commonly believed that self-describing classes are easy to learn. Specifically, the self-describing class $C = \{\alpha i p | \alpha \in \mathbb{N}^*,\ i \geqslant 2,\ p \in R_{0,1}, \varphi_i = \alpha i p\}$ turned out to be EX-complete. Notice that this class is not only equally hard to learn as the class *FINSUP* in a *formal* sense (both classes are EX-complete), but also in the following *intuitive* sense. When learning *FINSUP* one never can know if and when all the points of finite support have been found; analogously, when learning the class C one never can know if and when the last value exceeding 1 has been found. One might wish to further argue that learning the class C at all is "artificially/unnaturally easy", since, for any function from C, it suffices to find *one* only suitable value of that function in order to know the whole function. This is formally correct. But, consider the class of all polynomials on the reals. This class has an analogous property! Actually, for any polynomial of degree n, an arbitrarily small interval, more exactly, *finitely many* points, even *arbitrary* $n + 1$ of the denumerably many ones suffice to recreate (to learn) the whole function. Hardly, anyone would call the polynomials an artificial or unnatural class therefore. Naturally, we cannot exclude that for some approach to the complexity of learning, all the self-describing classes are really easy to learn. Answering this question in a rigorous way would require to *formalize* the notion of self-description, though.

In general, each approach to formalize the notion of complexity of learning will focus on specific features of this broad and diverse notion and will also have its specific implications. In this sense, we feel it quite justified to study further approaches to this notion which may more or less differ from the approach of intrinsic complexity. Slight modifications of the present approach could consist in allowing the first reducing operator to be limit-recursive when learning in the limit is considered, or to require that this operator is "length-preserving" in a reasonable sense. On the other hand, the approach by Nessel [23] formalizing the complexity of learning also in a reducibility fashion strongly differs from our approach by showing that in that approach the class *FINSUP* is not only not complete, but it has even *minimal* complexity. This is due to the fact that there the learning of "basic rules" is considered as the main goal (the basic rule for the whole class *FINSUP* is one and the same, namely the everywhere zero function, and learning a single function is intuitively of minimal complexity, since it can be thought to be known a priori), whereas collecting the finitely many and "arbitrarily chaotic" exceptions on which any function of *FINSUP* may differ from this basic rule is not taken into the complexity account by definition. Also, in contrast to our r.e. property in the context of intrinsic completeness as considered above, we cannot exclude that for other approaches just some kind of *high* algorithmic complexity will turn out to be necessary for a class to belong to the most difficult to learn ones. Thus, it would be desirable to study alternative approaches to the complexity of learning as well in order to yield a more complete picture and thus further deepen our understanding of the diverse nature of this important notion.

Acknowledgments

We thank the referees for several helpful comments which improved the presentation of the paper. The first author was supported by NUS grant number R252-000-127-11Z. The second author was supported by the URCG grant of Sacred Heart University. The fourth author was partly supported by the National Science Foundation under grant CCR-9732692. The fourth and the fifth author would like to thank the Deutsche Forschungsgemeinschaft (DFG) for supporting a guest professorship of the fourth author at the University of Kaiserslautern.

References

[1] A. Ambainis, S. Jain, A. Sharma, Ordinal mind change complexity of language identification, Theoret. Comput. Sci. 220 (1999) 323–343.

[2] D. Angluin, C.H. Smith, Inductive inference: theory and methods, Comput. Surv. 15 (1983) 237–269.

[3] J. Case, E. Kinber, A. Sharma, F. Stephan, On the classification of computable languages, in: STACS 97, 14th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 1200, Springer Verlag, Berlin, 1997, pp. 225–236.

[4] J. Case, C.H. Smith, Comparison of identification criteria for machine inductive inference, Theoret. Comput. Sci. 25 (1983) 193–220.

[5] R.P. Daley, C.H. Smith, On the complexity of inductive inference, Inform. Control 69 (1986) 12–40.

[6] R. Freivalds, J. Bārzdiņš, K. Podnieks, Inductive inference of recursive functions: complexity bounds, in: Baltic Computer Science, Lecture Notes in Computer Science, vol. 502, 1991, pp. 111–155.

[7] R. Freivalds, Inductive inference of recursive functions: qualitative theory, in: Baltic Computer Science, Lecture Notes in Computer Science, vol. 502, 1991, pp. 77–110.

[8] R. Freivalds, E. Kinber, C.H. Smith, On the intrinsic complexity of learning, Inform. Comput. 123 (1995) 64–71.

[9] R. Freivalds, C. Smith, On the role of procrastination in machine learning, Inform. Comput. 107 (1993) 237–271.

[10] R. Freivalds, R. Wiehagen, Inductive inference with additional information, J. Inform. Process. Cybernet. (EIK) 15 (1979) 179–185.

[11] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman and Company, New York, 1979.

[12] W. Gasarch, C. Smith, Recursion theoretic models of learning: some results and intuitions, Ann. Math. Artif. Intell. 15 (1995) 151–166.

[13] E.M. Gold, Language identification in the limit, Inform. Control 10 (1967) 447–474.

[14] K.P. Jantke, H.-R. Beick, Combining postulates of naturalness in inductive inference, J. Inform. Process. Cybernet. (EIK) 17 (1981) 465–484.

[15] S. Jain, D. Osherson, J.S. Royer, A. Sharma, Systems that Learn, second ed., MIT Press, Cambridge, 1999.

[16] S. Jain, A. Sharma, The intrinsic complexity of language identification, J. Comput. System Sci. 52 (1996) 393–402.

[17] S. Jain, A. Sharma, The structure of intrinsic complexity of learning, J. Symbolic Logic 62 (1997) 1187–1201.

[18] S. Jain, A. Sharma, Elementary formal systems, intrinsic complexity, and procrastination, Inform. Comput. 132 (1997) 65–84.

[19] E. Kinber, C. Papazian, C. Smith, R. Wiehagen, On the intrinsic complexity of learning infinite objects from finite samples, Technical Report LSA-99-01E, Centre for Learning Systems and Applications, Department of Computer Science, University of Kaiserslautern, 1999.

[20] M. Kummer, A learning-theoretic characterization of classes of recursive functions, Inform. Process. Lett. 54 (1995) 205–211.

[21] R. Klette, R. Wiehagen, Research in the theory of inductive inference by GDR mathematicians – a survey, Inform. Sci. 22 (1980) 149–169.

[22] M. Machtey, P. Young, An Introduction to the General Theory of Algorithms, North-Holland, New York, 1978.

[23] J. Nessel, Birds can fly. . . , in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM Press, New York, 1998, pp. 56–63.

[24] D. Osherson, M. Stob, S. Weinstein, Systems that Learn, MIT Press, Cambridge, 1986.

[25] L. Pitt, M.K. Warmuth, Prediction-preserving reducibility, J. Comput. System Sci. 41 (1990) 430–467.

[26] H. Rogers Jr., Gödel numberings of partial recursive functions, J. Symbolic Logic 23 (1958) 331–341.

[27] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, McGraw Hill, New York, 1967.

[28] C. Smith, A Recursive Introduction to the Theory of Computation, Springer, Berlin, 1994.

[29] F. Stephan, On one-sided versus two-sided classification, Technical Report 25/1996, Mathematical Institute, University of Heidelberg, 1996.