Consistent Partial Identification

Sanjay Jain* **and** **Frank Stephan†**
Department of Computer Science, National University of Singapore, Singapore 117417
Email: sanjay@comp.nus.edu.sg and fstephan@comp.nus.edu.sg

Abstract

This study contrasts consistent partial identification with learning in the limit. Here partial identification means that the learner outputs an infinite sequence of conjectures in which one correct hypothesis occurs infinitely often and all other hypotheses occur only finitely often. Consistency means that every conjecture is correct on all the data seen so far. Learning in the limit means that the learner outputs from some time on always the same correct hypothesis. As the class of all total-recursive functions can be partially identified, the constraint of consistency has to be added to make a meaningful comparison to learning in the limit. For the version of consistency where the learner has to be defined and consistent on all inputs, it is shown that the power of the learning criterion depends on whether the function to be learnt is fed in canonical order or in arbitrary order. In the first case, consistent partial identification is incomparable to learning in the limit; in the second case, it is equivalent to consistent learning in the limit with arbitrarily fed input. Furthermore, the inference degrees of these criteria are investigated. For the case where the function is fed in canonical order, there are just two inference degrees: the trivial one which contains all oracles of hyperimmune free Turing degree and the omniscient one which contains all oracles of hyperimmune Turing degree. In the case that the function is fed in arbitrary order, the picture is more complicated and the omniscient inference degree contains exactly all oracles of high Turing degree.

1 Introduction

A learning situation may be described as follows. Consider a learner receiving data, one piece at a time, about a target concept. As the learner is receiving its data, the learner conjectures its hypothesis describing the target. As the learner gets more and more data, its hypothesis may change. One may consider the learner to be successful if eventually its sequence of hypotheses converges to a correct hypothesis describing the target concept. This is essentially Gold's [10] model of learning in the limit. In this paper we will be mainly concerned about learning functions. The data in this case takes the form of the graph of the function, which is presented one datum at a time. The hypotheses of the learner take the form of a program for computing the function. The model of learning described above is referred to as **Ex**-learning, where the learner is expected to succeed with respect to any order of presentation of the data, as long as all the elements of the graph of the target function is eventually presented to it.

Note that there is no requirement in the above criterion that intermediate conjectures, output by the learner before it converges to its final hypothesis, be consistent with the data seen. In other words, the intermediate hypotheses may be contradicted by the data already seen. If one requires that the intermediate hypotheses be consistent with the data seen upto that point, then we get the criterion of learning called consistent learning [2]. It can be shown that there are concept classes which can be learnt, but no learner which is consistent on all data from the concept class can learn them (using the **Ex**-model of learning discussed above). There are three variation of consistent learning considered in the literature.

- A **Cons**-learner is expected to be consistent only on the data which are from some valid target concept (from the class of concepts being learnt) and the learner may be inconsistent or may even be undefined on data which are not from some target concept in the concept class being learnt [2].
- An **TCons**-learner is expected to be consistent on all possible data, even those which do not belong to any possible target concept [13].
- An **RCons**-learner is expected to be defined on all possible data, though it may not be consistent on data which do not belong to any concept from the class being learnt [4, 21].

Besides the **Ex**-model of learning discussed above, there have been several other learning models considered in the literature. One of the more common one is behaviourally correct learning (**Bc**-learning) [3], where it is required that after some finite time the learner outputs only correct hypotheses,

*Supported in part by NUS grant number R252-000-308-112.
†Supported in part by NUS grant numbers R146-000-114-112 and R252-000-308-112.

but these hypotheses may not all be the same (in other words, the hypotheses semantically converge to a correct one).

Osherson, Stob and Weinstein [18] relaxed the requirement that all but finitely many of the hypotheses be correct. They required that: (a) some fixed correct hypothesis be output infinitely often and (b) only one hypothesis is output infinitely often. This criterion of learning was called partial learning. It was shown by [18] that the class of all total recursive functions can be learnt by some learner in this fashion (in fact, using a suitable definition for learning recursively enumerable languages or partial functions, it can be shown that the class of all recursively enumerable languages, or the class of all partial functions, can be learnt in this fashion). However, the learner doing the above was not consistent. Therefore it is interesting to combine the two notions and to see what can be learnt by consistent partial learners. It turns out that this notion does no longer permit to learn all recursive functions and is a real restriction, the severeness of the restriction depends on whether the functions are presented by the canonical text or an arbitrary text. In this paper we give a detailed study of the resulting criteria.

The paper is organized as follows. In Section 2 we give the preliminaries and define formally the models related to learning considered in this paper. In Section 3 partial learning is defined, along with consistent partial learning. Let **ConsPart**, **TConsPart** and **RConsPart** denote partial learning where the learners are expected to be consistent in the **Cons**, **TCons** and **RCons** sense, respectively, as discussed above. We use the superscript arb or can to denote that the above learning/consistency requirements are for all possible order of presentation of data, or only for canonical presentation of data for the functions.

In Section 4 we consider when a learner can learn the class of all total recursive functions: particularly access to which oracles allows the learner to consistently partially learn the class of all total recursive functions. Let $\mathcal{R}$ denote the class of all total recursive functions. Theorem 6 shows that $\mathcal{R} \in \mathbf{TConsPart}^{can}[A]$ (respectively, $\mathbf{RConsPart}^{can}[A]$ and $\mathbf{ConsPart}^{can}[A]$) iff A has hyperimmune Turing degree. Additionally, Theorem 6 shows that if A has hyperimmune-free Turing degree, then $\mathbf{TConsPart}^{can}[A] = \mathbf{TConsPart}^{can}$. For learning from arbitrary order of presentation of data, Corollary 9 shows that $\mathcal{R} \in \mathbf{TConsPart}^{arb}[A]$ iff A is high. Also, surprisingly $\mathbf{TConsPart}^{arb}[A]$ is same as $\mathbf{TCons}^{arb}[A]$ for all oracles A (Theorem 8).

In Section 5 we compare the consistent partial learning criteria for arbitrary versus canonical texts. We show that for r.e., nonrecursive and non-high oracles A it holds that $\mathbf{TConsPart}^{can} \not\subseteq \mathbf{TConsPart}^{arb}[A]$ and $\mathbf{TConsPart}^{arb}[A] \not\subseteq \mathbf{TConsPart}^{can}$. Furthermore, if $A \leq_T K$ is 1-generic then A is in the omniscient $\mathbf{TConsPart}^{can}$-degree and the trivial $\mathbf{TConsPart}^{arb}$-degree.

In Section 6 we compare consistent partial learning with other criteria of learning such as **Ex** and **Bc** as well as compare various versions of consistency. In particular, we show that there exist classes of recursive functions which are partially learnable by a consistent learner but not **Bc**-learnable (Theorem 15). Furthermore, there are classes of recursive functions which are in **Ex** (even with minimal mind change complexity) which cannot be partially learnt by a consistent learner (Theorems 16 and 18).

Also, there are classes of recursive functions which are **RCons**-learnable, but which are not **TConsPart**-learnable (Corollary 17) and classes of recursive functions which are **Cons**-learnable, but not **RConsPart**-learnable (Theorem 19).

In Section 7 we consider learning partial functions, where the program output infinitely often by the learner is a partial or total extension of the input function. Let $\mathcal{P}$ denote the class of all partial recursive functions and $\mathcal{S}$ denote the class of all partial recursive function which have a total recursive extension. We show that, for learning partial functions by partial extensions, $\mathcal{P} \in \mathbf{ConsPart}^{arb}[A]$ iff $K \leq_T A$ (Theorem 20) and for learning partial functions by total extensions, $\mathcal{S} \in \mathbf{ConsPart}^{arb}[A]$ iff A is high (Theorem 21).

2 Preliminaries

Most recursion-theoretic notations are standard and follow the textbooks of Odifreddi [16, 17], Rogers [19] and Soare [20]. Background on inductive inference can be found in [12]. Let $\mathbb{N}$ denote the set of natural numbers. For a set S, $|S|$ denotes its cardinality. $|S| \leq *$ denotes that S is finite. $\max(S)$ and $\min(S)$ respectively denote the maximum and minimum of the set S, where $\max(\emptyset)$ is taken to be 0 and $\min(\emptyset)$ is taken to be ∞. $A \oplus B$ denotes the set $\{2x : x \in A\} \cup \{2x + 1 : x \in B\}$.

For a function η, we say that $\eta(x)\downarrow$ if $\eta(x)$ is defined and $\eta(x)\uparrow$ if $\eta(x)$ is undefined. A 0-extension of a finite function η is the function f such that $f(x) = \eta(x)$ for all x in the domain of η and $f(x) = 0$ for all x not in the domain of η.

A programming system is an arbitrary partial recursive function ψ of two variables. $\lambda x.\psi(e, x)$ is denoted by ψ_e, and e is called a program or index for ψ_e. $\psi_{e,s}$ denotes ψ_e computed within s steps: that is $\psi_{e,s}(x) = \psi_e(x)$, if $x < s$ and $\psi_e(x)$ halts within s steps; $\psi_{e,s}$ is undefined otherwise. φ denotes a fixed acceptable programming system [19]. φ_e denotes the (partial) function computed by the e-th program in the programming system φ. Unless otherwise stated, programs and indices generally mean programs or indices in the φ-system. Φ denotes a fixed Blum complexity measure [5] for the programming system φ. $\mathcal{R}$ denotes the set of all total recursive functions. $\mathcal{P}$ denotes the set of all partial recursive functions.

We say that a total function g dominates a total function f iff for all but finitely many n, $g(n) \geq f(n)$. A set A is *high* iff $K' \leq_T A'$. This can also be characterized in terms of domination: A is high iff there is a function $f \leq_T A$ which dominates every recursive function [15].

A set A is 1-*generic* iff for all r.e. sets $B \subseteq \{0, 1\}^*$ there exists an n such that either $A(0)A(1)\ldots A(n) \in B$ or no extension of $A(0)A(1)\ldots A(n)$ belongs to B.

A set A is said to be *immune* iff it is infinite and it does not contain any infinite recursively enumerable set. A set A is *hyperimmune* iff there is no recursive function f with $|A \cap \{0, 1, 2, \ldots, f(n)\}| \geq n$ for all n. A set B has *hyperimmune Turing degree* iff there is an $A \equiv_T B$ which is hyperimmune; otherwise B has *hyperimmune-free Turing degree*.

We often also consider computations relative to an oracle A. g^A defines a function g relative to an oracle A.

In the following, we review learning-theoretic notation. A text (for a function) is a mapping from $\mathbb{N}$ to $\mathbb{N} \times \mathbb{N} \cup \{\#\}$, such that if (x, y) and (x, z) are in the range of the text then $y = z$. Content of a text T, denoted content(T), is the set of pairs in the range of T (that is range of T, except for $\#$). A text T is for a (possibly partial) function η iff content$(T) = \{(x, \eta(x)) : \eta(x)\downarrow\}$. For a text T, $T[n]$ denotes the initial segment of T of length n. A finite sequence is an initial segment of a text. Content of a sequence σ, denoted content(σ), is the set of pairs in the range of σ. Length of σ, denoted $|\sigma|$, is the number of elements in the domain of σ. SEQ denotes the set of all finite sequences. For finite sequences σ and τ, $\sigma \diamond \tau$ denotes the concatenation of σ and τ, that is, $\sigma \diamond \tau(x)$ is $\sigma(x)$ if $x < |\sigma|$, $\sigma \diamond \tau(x)$ is $\tau(x - |\sigma|)$ if $|\sigma| \leq x < |\sigma| + |\tau|$ and $\sigma \diamond \tau(x)$ is undefined if $x \geq |\sigma| + |\tau|$. Furthermore, we sometimes write $\sigma \diamond (x, y)$ to denote the concatenation of σ with the sequence containing just one element (x, y).

A text T is a canonical text for a total function f if $T(i) = (i, f(i))$. When we are dealing with learning from canonical text, we often identify the function with its canonical text. Thus, $f[n]$ represents $T[n]$, where T is the canonical text for f. CanSEQ denotes the set of all finite initial segments of canonical texts.

A learner is a mapping from SEQ to $\mathbb{N} \cup \{?\}$. Intuitively, ? denotes no conjecture at this point. A learner M converges on a text T to i iff for all but finitely many n, $M(T[n]) = i$.

We mostly consider partial recursive learners only. In some cases, we allow learners access to an oracle A. M (with or without subscripts or superscripts such as primes) ranges over partial recursive learners. M^A denotes a partial recursive learner relative to oracle A. Unless otherwise specified, a learner is assumed to be partial recursive.

The following two definitions give the basic learning criteria for explanatory and behaviourally correct learning.

Definition 1 [6, 10] Let $b \in \mathbb{N} \cup \{*\}$.

- M **Ex**$_b$-*identifies* $f \in \mathcal{R}$ (written: $f \in \mathbf{Ex}_b(M)$) iff for all texts T for f, there exists a program i for f such that M converges on T to i and there are at most b numbers n with $? \neq M(T[n]) \neq M(T[n+1])$.
- M **Ex**$_b$-*identifies* $\mathcal{C} \subseteq \mathcal{R}$ iff M **Ex**$_b$-identifies each $f \in \mathcal{C}$.
- $\mathbf{Ex}_b = \{\mathcal{C} \subseteq \mathcal{R} : (\exists \text{ partial recursive } M)[\mathcal{C} \subseteq \mathbf{Ex}_b(M)]\}$.

$\mathbf{Ex}_*$ is also referred to as **Ex**.

Definition 2 [3] Let $f \in \mathcal{R}$ and $\mathcal{C} \subseteq \mathcal{R}$.

- M **Bc**-*identifies* f (written: $f \in \mathbf{Bc}(M)$) iff for all texts T for f, for all but finitely many n, $\varphi_{M(T[n])} = f$.
- M **Bc**-*identifies* $\mathcal{C} \subseteq \mathcal{R}$ iff M **Bc**-identifies each $f \in \mathcal{C}$.
- **Bc** $= \{\mathcal{C} \subseteq \mathcal{R} : (\exists \text{ partial recursive } M)[\mathcal{C} \subseteq \mathbf{Bc}(M)]\}$.

The following definition formally gives the definition for consistent learning.

Definition 3 [2] A learner M is said to be *consistent* on a text T iff for all $n \in \mathbb{N}$, $M(T[n])\downarrow$ and content$(T[n]) \subseteq \varphi_{M(T[n])}$. A learner is said to be consistent on a function f iff it is consistent on all texts for f.

Definition 4 [2, 4, 13, 21]

- A learner M **Cons**-identifies $\mathcal{C}$ iff it **Ex**-identifies $\mathcal{C}$ and it is consistent on each $f \in \mathcal{C}$.
- A learner M **RCons**-identifies $\mathcal{C}$, iff it **Cons**-identifies $\mathcal{C}$ and it is total.
- A learner M **TCons**-identifies $\mathcal{C}$, iff it **Ex**-identifies $\mathcal{C}$ and it is consistent on each $f \in \mathcal{R}$.

When learning total functions, we often consider learning from canonical texts only. This does not make a difference for $\mathbf{Ex}_b$ or **Bc** learning. But for consistent learning it matters and we use the superscripts can and arb to make it clear.

- $\mathcal{C} \in \mathbf{TCons}^{can}$ iff there is a recursive learner M which is total, consistent on each $f \in \mathcal{R}$, and **Ex**-learns every $f \in \mathcal{C}$ from the text of the form $(0, f(0)), (1, f(1)), \ldots$; such texts are called *canonical texts*.
- $\mathcal{C} \in \mathbf{TCons}^{arb}$ iff there is a recursive learner M which is total, consistent on each $f \in \mathcal{R}$, and **Ex**-learns every $f \in \mathcal{C}$ from every text of the form $(x_0, f(x_0)), (x_1, f(x_1)), \ldots$ where $x_0, x_1, \ldots$ must cover the whole set $\mathbb{N}$; such texts are called *arbitrary texts*.

Similarly one can define these two variants for other criteria of learning. In the case that we do not specify in a definition or theorem which case applies, we always assume that the definition or the result would hold for both variants in the same way.

3 Partial Learning

We now consider the definitions for partial and consistent partial learning.

Definition 5 [18] Let $f \in \mathcal{R}$ and $\mathcal{C} \subseteq \mathcal{R}$.

- M **Part**-*identifies* f (written: $f \in \mathbf{Part}(M)$) iff for all texts T for f, there exists a program i for f such that
  - $M(T[n]) = i$ for infinitely many n and
  - for all $j \neq i$, $M(T[n]) = j$ for only finitely many n.
- M **Part**-*identifies* $\mathcal{C}$ iff M **Part**-identifies each $f \in \mathcal{C}$.
- **Part** $= \{\mathcal{C} : (\exists M)[\mathcal{C} \subseteq \mathbf{Part}(M)]\}$.
- We say that a learner M **ConsPart**-identifies $\mathcal{C}$ iff M **Part**-identifies $\mathcal{C}$ and M is consistent on each $f \in \mathcal{C}$.
- We say that a learner M **TConsPart**-identifies $\mathcal{C}$ iff M **Part**-identifies $\mathcal{C}$ and M is consistent on each $f \in \mathcal{R}$.
- We say that a learner M **RConsPart**-identifies $\mathcal{C}$ iff M **ConsPart**-identifies $\mathcal{C}$ and M is total.

One can similarly identify the learning criteria **TConsPart**, **RConsPart** and **ConsPart** with the collections of all classes of functions learnable under the respective criterion. When considering learning with oracles [1, 7, 9, 14], we use $\mathbf{I}[A]$ to denote learning under criterion **I** where the learners are allowed access to oracle A. We note that, for the notions **I** of consistent learning defined here, $\mathbf{I}^{can}$ and $\mathbf{I}^{arb}$ are different. Furthermore, for many oracles A, $\mathbf{I}^{can}[A]$ and $\mathbf{I}^{arb}[A]$ are different.

4 Some Characterizations of Consistent Partial Learning

In this section, we first characterize for every oracle A the class **TConsPart**$^{can}[A]$. There are only two cases, either A permits to learn the class of all recursive functions under this criterion or A is trivial and the classes learnable are characterized by some quite natural domination property. Oracles of hyperimmune degree fall into the first and oracles of hyperimmune-free degree fall into the second case.

Theorem 6 **(a)** *If A has hyperimmune Turing degree then it holds that $\mathcal{R} \in$ **ConsPart**$^{can}[A]$, $\mathcal{R} \in$ **TConsPart**$^{can}[A]$ and $\mathcal{R} \in$ **RConsPart**$^{can}[A]$.*
(b) *If A has hyperimmune-free Turing degree then a class $\mathcal{C}$ is in **TConsPart**$^{can}[A]$ iff there is a strictly increasing recursive function g such that every $f \in \mathcal{C}$ has an index e which satisfies, for infinitely many n,*

$$\forall m \leq n\, [\Phi_e(m) \leq g(\max(\{n, f(0), f(1), \ldots, f(n)\}))].$$

*In particular, $\mathcal{R} \notin$ **ConsPart**$^{can}[A]$, $\mathcal{R} \notin$ **TConsPart**$^{can}[A]$ and $\mathcal{R} \notin$ **RConsPart**$^{can}[A]$.*

Proof. (a) Suppose A is hyperimmune. We show that $\mathcal{R} \in$ **TConsPart**$^{can}[A]$. A hyperimmune oracle A allows one to construct a function h^A not dominated by any recursive function. Let ψ be a one-one numbering of all partial recursive functions [8]. Let $r(\cdot)$ be a reduction from ψ to φ. Now a learner M using oracle A is defined as follows.

$M(f[\langle n, e\rangle]) = r(e)$, if $f[\langle n, e\rangle] \subseteq \psi_{e,h^A(n)}$ and $\psi_{e,m} \subseteq f[\langle n, e\rangle]$, where m is the number of times $r(e)$ has been output on proper initial segments of $f[\langle n, e\rangle]$. Otherwise, $M(f[\langle n, e\rangle]) = z$ such that z has not been output by M on proper initial segments of $f[\langle n, e\rangle]$ and $f[\langle n, e\rangle] \subseteq \varphi_z$.

For any recursive function ψ_e, let $g(n) = \min(\{s : \psi_e[\langle n, e\rangle] \subseteq \psi_{e,s}\})$.

Then $h^A(n)$ is, for infinitely many n, larger than $g(n)$. For these infinitely many n, M on $\psi_e[\langle n, e\rangle]$ will output $r(e)$ (note that the number of times $r(e)$ is output on proper initial segments of $\psi_e[\langle n, e\rangle]$ is at most $\langle n, e\rangle$ and $\psi_{e,\langle n,e\rangle} \subseteq \psi_e[\langle n, e\rangle]$). Thus, $r(e)$ is output infinitely often by M on ψ_e.

For $e' \neq e$, if $\psi_e[\langle n, e'\rangle] \not\subseteq \psi_{e'}$, then $r(e')$ would not be output beyond $\psi_e[\langle n, e'\rangle]$. Any number z not in the range of r is output at most once. Thus, M **TConsPart**$^{can}[A]$-identifies $\mathcal{R}$.

(b) ($\Rightarrow$) Let M be an A-recursive **TConsPart**$^{can}[A]$-learner for $\mathcal{C}$. Now one constructs a function g^A as follows: $g^A(n)$ is the maximum of $\Phi_e(x)$ such that there is a finite sequence $\sigma = (0, y_0)\,(1, y_1)\,\ldots\,(m, y_m)$ with $\max(\{m, y_0, y_1, \ldots, y_m\}) \leq n$, $x \leq m$ and $M(\sigma) = e$. This function g^A is then bounded by a strictly increasing recursive function g as A is hyperimmune-free Turing degree [20]. When learning $f \in \mathcal{C}$, M outputs one index e infinitely often. Then, for each n such that the learner outputs e on input $(0, f(0)), (1, f(1)), \ldots, (n, f(n))$, all values $\Phi_e(m)$ with $m \leq n$ are bounded by $g(\max(\{n, f(0), f(1), \ldots, f(n)\}))$. This completes one direction.

($\Leftarrow$) Assume that there is a recursive function g such that every $f \in \mathcal{C}$ has an index e which satisfies for infinitely many n that

$$\forall m \leq n\, [\Phi_e(m) \leq g(\max(\{n, f(0), f(1), \ldots, f(n)\}))].$$

Let pad be a recursive padding function (that is $\varphi_{pad(i,k)} = \varphi_i$, for all i, k, and pad is one–one). Now the learner M works as follows.

1. Read the current input $(0, f(0)), (1, f(1)), \ldots, (n, f(n))$.
2. Determine the least $e \leq n$ such that either $e = n$ or it holds for all $k \leq n$ that $\Phi_e(k) \leq g(\max(\{n, f(0), f(1), \ldots, f(n)\})) \wedge \varphi_e(k) = f(k)$.
3. Determine the number p of pairs (m, d) with $m \leq n$ and $d < e$ such that for all $k \leq m$, $\Phi_d(k) \leq g(\max(\{m, f(0), f(1), \ldots, f(m)\})) \wedge \varphi_d(k) = f(k)$.
4. Output $pad(e, p)$.

Given $f \in \mathcal{C}$, there is a least index e of f such that, for infinitely many n and all $k \leq n$, it holds that $\Phi_e(k) \leq g(\max(\{n, f(0), f(1), \ldots, f(n)\})) \wedge \varphi_e(k) = f(k)$. Thus the number p of all pairs (m, d) with $d < e$ and $\Phi_d(k) \leq g(\max(\{m, f(0), f(1), \ldots, f(m)\})) \wedge \varphi_d(k) = f(k)$ for all $k \leq m$ is finite. It follows that M outputs $pad(e, p)$ infinitely often. Furthermore, each other possible value $pad(d, q)$ with $(e, p) \neq (d, q)$ is output only finitely often: if $\varphi_d \neq f$ then d passes the test in step 2 only for finitely many n; if $\varphi_d = \varphi_e$ and $d < e$, then d passes the test in step 2 only for finitely many n (by the hypothesis on e being the least); if $\varphi_d = \varphi_e$ and $d > e$, then p as computed in step 3, is unbounded for the corresponding d and, thus, each $pad(d, q)$ is output only for finitely many n. Hence $\mathcal{C} \in$ **TConsPart**^{can}.

As $\mathcal{R}$ is dense, any **ConsPart**$^{can}[A]$ learner for $\mathcal{R}$ is also a **TConsPart**$^{can}[A]$ learner for $\mathcal{R}$. Thus, it suffices to show that $\mathcal{R} \notin$ **TConsPart**$^{can}[A]$. It is well-known that for every recursive function g, there is a $\{0, 1\}$-valued recursive function f such that for all e with $\varphi_e = f$ and almost all n, $\Phi_e(n) > g(n)$. Hence, $\mathcal{R}$ does not satisfy the characterization of the classes in **TConsPart**$^{can}[A]$ for oracles A of hyperimmune-free Turing degree given in part (b). □

Note that the characterization is independent of A and hence the learning power of all the oracles of hyperimmune-free degree is the same as **TConsPart**^{can}.

Corollary 7 *If A has hyperimmune-free Turing degree then* **TConsPart**$^{can}[A] =$ **TConsPart**^{can}.

The next result characterizes **TConsPart**$^{arb}[A]$ and shows that it can be mapped back to a known criterion. Note that the condition in the third item on domination implies that there are infinitely many inference degrees for this criterion.

Theorem 8 *For every oracle A and class $\mathcal{C}$, the following conditions are equivalent.*

- *$\mathcal{C} \in$ **TConsPart**$^{arb}[A]$;*
- *$\mathcal{C} \in$ **TCons**$^{arb}[A]$;*
- *There is a strictly increasing A-recursive function g^A such that every $f \in \mathcal{C}$ has an index e which satisfies $\Phi_e(x) \leq g^A(\max(\{x, f(x)\}))$ for almost all x.*

Proof. By definition, $\mathbf{TCons}^{arb}[A] \subseteq \mathbf{TConsPart}^{arb}[A]$ is true.

The implication from the third condition to the second can be obtained by applying the algorithm to learn by enumeration [10]. On input σ, the learner determines m to be the maximum of all numbers which occur as first or second component of a pair in content(σ) and then outputs the least $e \leq m$ such that $\Phi_e(x) \leq g^A(m) \wedge \varphi_e(x) = y$ for all $(x, y) \in$ content(σ); if such an e does not exist, the learner outputs some canonical index of the function which coincides with the seen data and maps all unseen inputs to 0. Clearly, the algorithm above is consistent on all inputs. Furthermore, for any $f \in \mathcal{C}$, on any text T for f, the algorithm converges to the least e such that (i) $\varphi_e = f$ and (ii) for all but finitely many n, and $(x, f(x)) \in$ content($T[n]$): $e \leq m_n$ and $\Phi_e(x) \leq g^A(m_n)$, where m_n is the maximum of all numbers which occur as first or second component of a pair in $T[n]$. (Note that there exists such a e due to the third condition). Thus the algorithm witnesses $\mathbf{TCons}^{arb}[A]$-identification of $\mathcal{C}$.

For the remaining implication from the first to the third condition, assume that M is an A-recursive $\mathbf{TConsPart}^{arb}$-learner for $\mathcal{C}$. Now let $\tilde{g}^A(n)$ be the maximum value of all $\Phi_{M(\sigma)}(x)$ where σ, x satisfy that $|\sigma| \leq n$, content(σ) contains only pairs with both components in $\{0, 1, \ldots, n\}$ and x is the first coordinate of a pair in content(σ). The function $\tilde{g}^A$ is total and A-recursive as it takes only the maximum over finitely many sequences and M only outputs hypotheses which are consistent on the data seen so far. Furthermore, one defines for every finite set D of indices the index $amal(D)$ of the amalgamation of the indices such that $\varphi_{amal(D)}(x)$ takes the value y if there is an s and an $e \in D$ such that $\varphi_e(x) = y$, $\Phi_e(x) = s$ and for all $d \in D$, $[\Phi_d(x) > s$ or $[\Phi_d(x) = s$ and $\varphi_d(x) \geq y]]$. As the amalgamation is an effective operation, there is a strictly increasing A-recursive function g^A such that $\Phi_{amal(D)}(m) \leq g^A(n)$ whenever $D \subseteq \{0, 1, \ldots, n\}$, $m \leq n$ and there is a $d \in D$ with $\Phi_d(m) \leq \tilde{g}^A(n)$.

Fix $f \in \mathcal{C}$. Now one tries to noneffectively define, by induction, the following text T:

Stage 0. Set $T(0) = (0, f(0))$.

Stage $2n + 1$. Let $E_n = \{M(T[r]) : r \leq 2n + 1\}$. Choose $x > 2n + 2$ and $x > \max(\{z, f(z)\})$ for all pairs $(z, f(z)) \in$ content($T[2n + 1]$) such that either $\Phi_e(x) > \tilde{g}^A(\max(\{x, f(x)\}))$ or $\varphi_e \neq f$ for all $e \in E_n$. Let $T(2n + 1) = (x, f(x))$.

Stage $2n + 2$. Let $(x, f(x)) = T(2n + 1)$. Let y be the least number such that either $y = x$ or $(y, f(y))$ does not occur in content($T[2n+2]$) and $\max(\{y, f(y)\}) \leq \max(\{x, f(x)\})$. Let $T(2n + 2) = (y, f(y))$.

We first show that the construction does not succeed in finding a x in some stage $2n + 1$. So suppose by way of contradiction that the construction always finds the required x in every stage $2n + 1$. Then the construction gives a text T for f. Furthermore, for all n, M on $T[2n + 2]$ or $T[2n + 3]$ does not output any of the indices in E_n unless that index is not an index for f (since, by definition of $\tilde{g}$, $\Phi_{M(T[2n+2])}(x)$ and $\Phi_{M(T[2n+3])}$ are both bounded by $\tilde{g}^A(\max\{x, f(x)\})$, where x is as found in stage $2n + 1$). It follows that indices for f are only repeated at most once by M and hence M does not $\mathbf{TConsPart}^{arb}[A]$-learn f, in contradiction to the assumption.

Hence, the construction fails to find the required x in stage $2n + 1$ for some n. In this case it holds that, for almost all x, there is an $e \in E_n$ with $\varphi_e = f \wedge \Phi_e(x) \leq \tilde{g}^A(\max(\{x, f(x)\}))$. Now let $d = amal(\{e \in E_n : \varphi_e = f\})$. Note that $\varphi_d = f$ as all indices used in the amalgamation are indices of f and the set of indices used is not empty. Furthermore, for all sufficiently large x, $E_n \subseteq \{0, 1, \ldots, x\}$ and $\Phi_d(x) \leq g^A(\max(\{x, f(x)\}))$. □

Corollary 9 *A is high iff $\mathcal{R} \in \mathbf{TConsPart}^{arb}[A]$ iff $\mathcal{R} \in \mathbf{RConsPart}^{arb}[A]$ iff $\mathcal{R} \in \mathbf{ConsPart}^{arb}[A]$.*

Proof. Note that $\mathcal{R}$ is dense. Hence it holds for all oracles A that $\mathcal{R} \in \mathbf{TConsPart}^{arb}[A]$ iff $\mathcal{R} \in \mathbf{RConsPart}^{arb}[A]$ iff $\mathcal{R} \in \mathbf{ConsPart}^{arb}[A]$.

If A is high, then there is an A-recursive function g^A which dominates every recursive function. Thus if A is high, then it follows using Theorem 8 that $\mathcal{R} \in \mathbf{TConsPart}^{arb}[A]$.

For the other direction, suppose $\mathcal{R} \in \mathbf{TConsPart}^{arb}[A]$. Then $\mathcal{R} \in \mathbf{TCons}^{arb}[A]$ and $\mathcal{R} \in \mathbf{Ex}[A]$. Hence A is high by a result of Adleman and Blum [1]. □

Note that the above characterizations show that the inference criteria $\mathbf{TConsPart}^{can}$ and $\mathbf{TConsPart}^{arb}$ are closed under union; this also holds for their relativized versions. That is, if $\mathcal{C}_0, \mathcal{C}_1 \in \mathbf{TConsPart}^{can}$ then $\mathcal{C}_0 \cup \mathcal{C}_1 \in \mathbf{TConsPart}^{can}$. This property does not hold for the criteria $\mathbf{RConsPart}^{can}$, $\mathbf{RConsPart}^{arb}$, $\mathbf{ConsPart}^{can}$ and $\mathbf{ConsPart}^{arb}$ by Theorem 18 below (since every class in $\mathbf{Ex}_1$ is a union of two classes in $\mathbf{Ex}_0$, and $\mathbf{Ex}_0 \subseteq \mathbf{RConsPart}^{arb}$).

One might ask whether the equivalence in Theorem 8 can be generalized.

Open Problem 10 **(a)** *Is $\mathbf{RConsPart}^{arb} = \mathbf{RCons}^{arb}$?*
(b) *Is $\mathbf{ConsPart}^{arb} = \mathbf{Cons}^{arb}$?*

While for $\mathbf{TConsPart}^{can}$ there are only two inference degrees, we show below that the degree structures for the criteria $\mathbf{RConsPart}^{can}$ and $\mathbf{ConsPart}^{can}$ are more complicated and in each case there are uncountably many degrees. The next theorem shows that for sets A, B of hyperimmune-free degree there is a class $\mathcal{C} \in \mathbf{RConsPart}^{can}[A]$ which is not in $\mathbf{ConsPart}^{can}[B]$ unless $A \leq_T B'$. This implies that all nonomniscient inference degrees are countable for these two criteria.

Theorem 11 **(a)** *If $\mathbf{ConsPart}^{can}[A] \subseteq \mathbf{ConsPart}^{can}[B]$ then $A \leq_T B'$ or B has hyperimmune Turing degree.*
(b) *If $\mathbf{RConsPart}^{can}[A] \subseteq \mathbf{RConsPart}^{can}[B]$ then $A \leq_T B \oplus K$ or B has hyperimmune Turing degree.*

Proof. Let A be fixed and assume without loss of generality that for every x, either $2x \in A$ or $2x + 1 \in A$ but not both. In the following let $\mathcal{C} = \{f : f(0) \in A \wedge f = \varphi_{f(1)}\} \cup \{f : f(0) \notin A \wedge \forall^\infty x\, [f(x) = 0]\}$.

This class $\mathcal{C}$ is **RConsPart**$^{can}[A]$-learnable by a learner which on input $(0, f(0)), (1, f(1)), ..., (n, f(n))$ conjectures a canonical index for $f(0)f(1)\dots f(n)0^\infty$ unless $f(0) \in A$ and $n \geq 1$; in the latter case the learner conjectures $f(1)$.

Assume now that the learner M **ConsPart**$^{can}[B]$-learns $\mathcal{C}$ and that B has hyperimmune-free Turing degree. Note that the proof of Theorem 16 below can be modified to show that, for each x, the class $\{f : f(0) = x \wedge f = \varphi_{f(1)}\}$ is not **TConsPart**$^{can}[B]$-learnable. Hence, for every $x \in A$, there is a function $f \notin \mathcal{C}$ with $f(0) = x$ such that for some n, either M (using oracle B) is undefined on $(0, f(0)), (1, f(1)), \dots, (n, f(n))$ or M (using oracle B) on $(0, f(0)), (1, f(1)), \dots, (n, f(n))$ outputs an index e which is not consistent with $(0, f(0)), (1, f(1)), \dots, (n, f(n))$. Therefore, there is a B'-recursive function ψ which finds such $(0, f(0)), (1, f(1)), \dots, (n, f(n))$, for every $x \in A$; for the $x \notin A$, there is no such witness as M must always output consistent conjectures on data from any function f with $f(0) \notin A$. Hence A is B'-r.e. and, as exactly one of each pair $2x, 2x + 1$ of numbers is in A, $A \leq_T B'$.

One can improve the above analysis in the case that M is a **RConsPart**$^{can}[B]$-learner, M is everywhere defined and hence $\psi(x)$ is a partial $B \oplus K$-recursive function as $\psi(x)$ alternately evaluates M on potential inputs $(0, f(0)), (1, f(1)), \dots, (n, f(n))$ using B and then checks using K whether the output obtained is consistent. Hence one has $A \leq_T B \oplus K$. □

5 Arbitrary versus Canonical Text

In the present section the influence of the data presentation (arbitrary versus canonical text) on the learning power of consistent partial learners is investigated. The next result shows that for non-high oracles A, **TConsPart**$^{arb}[A]$-learnability of classes of $\{0, 1\}$-valued functions implies that the corresponding class is also **TConsPart**^{can}-learnable while the corresponding inclusion does not hold in general when the condition of $\{0, 1\}$-valuedness is dropped. Furthermore, some oracles A are in the omniscient **TConsPart**^{can}-degree and trivial **TConsPart**^{arb}-degree; that is, depending on the permitted form of data presentation to the learner, the oracle is either omniscient or useless.

Theorem 12 **(a)** *If A is not high then* **TConsPart**$^{arb}[A] \cap \mathcal{R}_{0,1} \subseteq$ **TConsPart**^{can}.
(b) *If A is r.e. and not recursive then* **TConsPart**$^{arb}[A] \not\subseteq$ **TConsPart**^{can}.
(c) *If A is 1-generic and $A \leq_T K$ then* **TConsPart**$^{arb}[A]$ = **TConsPart**^{arb} *and* $\mathcal{R} \in$ **TConsPart**$^{can}[A]$.

Proof. (a) Assume that $\mathcal{C} \in \mathcal{R}_{0,1} \cap$ **TConsPart**$^{arb}[A]$ where A is not high. The characterization of **TConsPart**$^{arb}[A]$ in Theorem 8 and the fact that every $f \in \mathcal{C}$ is $\{0, 1\}$-valued imply that there is an A-recursive function g^A such that every $f \in \mathcal{C}$ has an index e with

$$\forall^\infty n\, \forall m \leq n\, [\Phi_e(m) \leq g^A(n)].$$

As A is not high, there is a recursive function g with $g(n) > g^A(n)$ for infinitely many n. Now every $f \in \mathcal{C}$ has an index e with

$$\exists^\infty n\, \forall m \leq n\, [\Phi_e(m) \leq g(n)].$$

This implies then that $\mathcal{C} \in$ **TConsPart**^{can} by Theorem 6.

(b) As A is r.e. but not recursive, A is the range of a recursive one-one sequence $a_0, a_1, a_2, \dots$ from which the convergence module c_A can be defined as $c_A(n) = \max(\{m : m = n \vee a_m \leq n\})$. As A is not recursive, c_A is not dominated by any recursive function.

Given an index d of a total recursive function, one defines a recursive function $F_d(n, m)$ as follows. $F_d(n, m)$ is the first k found (in some algorithmic search) such that $k > n$, $k > m$, $c_A(k) > \varphi_d(k)$, $c_A(k) > \Phi_d(k)$ and $k > \varphi_e(n)$ for all $e \leq n$ with $\Phi_e(n) \leq \varphi_d(k)$. Note that, as φ_d is recursive, there exists such a k and one can recursively find one such k from n, m. Furthermore, the complexity to find this k is bounded by $h(d, n, m, k, c^A(k))$ where h is a suitable strictly increasing recursive function; note that it needs to be $c^A(k)$ and not $c^A(n)$. Thus, one can compute $F_d(n, m)$ within time $h'(d, n, m, F_d(n, m), c^A(F_d(n, m)))$, for some strictly increasing recursive function h'.

Given any index d for a total recursive function, one can define a strictly increasing recursive function f_d inductively as follows. $f_d(0) = F_d(0, 0)$ and $f_d(n+1) = F_d(n, f_d(n))$. Consider the class $\mathcal{C} = \{f_d : \varphi_d \text{ is total}\}$.

Consider any index d for a recursive function. Note that, by definition of F_d and f_d, for all n, for all $e < n$, either $\varphi_e(n) \neq f_d(n)$ or $\Phi_e(n) > \varphi_d(f_d(n)) = \varphi_d(\max(\{n, f_d(0), f_d(1), \dots, f_d(n)\}))$. Thus there exists no program e computing f_d such that $\Phi_e(n) \leq \varphi_d(\max(\{n, f_d(0), f_d(1), ..., f_d(n)\}))$ for infinitely many n. Thus, $\mathcal{C} \notin$ **TConsPart**^{can} by Theorem 6. On the other hand, since $F_d(m, n)$ can be computed within time $h'(d, n, m, F_d(n, m), c^A(F_d(n, m)))$, we have that the function f_d has an index e with $\Phi_e(n) \leq h''(d, n-1, f_d(n-1), f_d(n), c_A(f_d(n)))$, for some increasing recursive function h''. Now we take h''' as $h'''(n, m) = \max(\{h''(d, n-1, m', m, c_A(m)) : d, m' \leq m\})$ and have that $\Phi_e(n)$ is bounded by $h'''(n, f_d(n))$, for almost all n. It follows that $\mathcal{C} \in$ **TConsPart**$^{arb}[A]$ − **TConsPart**^{can}.

(c) Let $A \leq_T K$ and A be 1-generic. It is clear that $\mathcal{R} \in$ **TConsPart**$^{can}[A]$ as every 1-generic set is hyperimmune and thus has hyperimmune Turing degree.

Let $\mathcal{C} \in$ **TConsPart**$^{arb}[A]$ be a given class. Then, by Theorem 8, there is an A-recursive function g^A such that every $f \in \mathcal{C}$ has an index e with $\Phi_e(n) \leq g^A(\max(\{n, f(n)\}))$ for almost all n. Let $\alpha_0, \alpha_1, \dots$ be a recursive approximation to A by finite strings (such approximation exists as $A \leq K$). Define $g(n) = g^{\alpha_{m(n)}}(n)$, where $m(n)$ is the least $k \geq n$ for which the computation of $g^{\alpha_k}(n)$ terminates with $|\alpha_k|$ steps. The function g is recursive.

Now let f and an index e of f be given such that $\Phi_e(n) > g(\max(\{n, f(n)\}))$ for infinitely many n. The set $\{\alpha_{m(n)} : \Phi_e(n) > g(\max(\{n, f(n)\}))\}$ is r.e. and infinite; hence it contains for every finite string $A(0)A(1)\dots A(k)$ an extension. As A is 1-generic, $\alpha_{m(n)}$ is a prefix of $A(0)A(1)\dots$ for infinitely many n and thus there are infinitely many n with $\Phi_e(n) > g^A(\max(\{n, f(n)\}))$.

Thus, for every $f \in \mathcal{C}$, for the index e of f such that $\Phi_e(n) \leq g^A(\max(\{n, f(n)\}))$ for almost all n, we also have that $\Phi_e(n) \leq g(\max(\{n, f(n)\}))$ for almost all n. It follows that $\mathcal{C} \in$ **TConsPart**^{arb}.

So the $\mathbf{TConsPart}^{can}$-degree of A is omniscient and the $\mathbf{TConsPart}^{arb}$-degree of A is trivial. $\square$

Theorem 13 *If* $\mathbf{TCons}^{can} \subseteq \mathbf{ConsPart}^{arb}[A]$ *then* A *is high.*

Proof. Given a total recursive function φ_e, let $\varphi_{s(e)}(2n) = \Phi_e(n)+\varphi_e(n)$ and $\varphi_{s(e)}(2n+1)$ be the least number m such that $m \neq \varphi_d(2n+1)$ for all $d < n$ with $\Phi_d(2n+1) \leq \varphi_e(n)$. Let $\mathcal{C}_0 = \{\varphi_{s(e)} : \varphi_e \text{ is total}\}$, let $\mathcal{C}_1$ be all functions which are almost everywhere 0 and let $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1$.

One can construct a $\mathbf{TCons}^{can}$-learner for $\mathcal{C}$ as follows. One can compute from $f(0)f(1)...f(n)$ and e whether $\varphi_{s(e)}$ coincides with f on inputs $\leq n$; in the case that this is true for some $e \leq n$ the learner outputs $s(e)$ for the least such e; otherwise the learner outputs a canonical index for the almost everywhere 0 function $f(0)f(1)\ldots f(n)0^\infty$. It is easy to verify that the above learner $\mathbf{TCons}^{can}$-identifies $\mathcal{C}$.

The class $\mathcal{C}$ is dense and hence it is in $\mathbf{ConsPart}^{arb}[A]$ iff it is in $\mathbf{TConsPart}^{arb}[A]$. It follows now from Theorem 8 that there is a function $g^A \leq_T A$ such that every function $\varphi_{s(e)}$ with e being an index of a total function has an index d with $\Phi_d(2n+1) \leq g^A(2n+1)$ for almost all n; note here that $\varphi_{s(e)}(2n+1) \leq n$. Thus, using definition of $\varphi_{s(e)}$, one can conclude that $\varphi_e(n) \leq g^A(2n+1)$ for almost all n. Hence, the function $n \mapsto g^A(2n+1)$ dominates all recursive functions and A is high. $\square$

As a corollary we get the following result of Grieser [11] who solved with this result an open problem from [22].

Corollary 14 (Grieser [11]) $\mathbf{TCons}^{can} \not\subseteq \mathbf{Cons}^{arb}$.

6 Comparison of Learning Criteria

The first result is to show that consistent partial identification from functions in canonical input order is not captured by behaviourally correct learning. The next results after that show that the criteria $\mathbf{Ex}_0$ and $\mathbf{Ex}_1$ fail to be included in various criteria of partial consistent learning.

Theorem 15 $\mathbf{TConsPart}^{can} \not\subseteq \mathbf{Bc}$.

Proof. Let $M_0, M_1, M_2, \ldots$ be a recursive sequence of all partial-recursive learners. For every given e, we construct a function f_e effectively in e as follows. For this, one constructs in parallel to f_e a strictly ascending sequence $x_0, x_1, \ldots$, depending on e, starting with $x_0 = 0$ and $f_e(0) = e$.

In stage n, define x_{n+1} and f_e on all y with $y \in \{x_n+1, x_n+2, \ldots, x_{n+1}\}$ as follows. $x_{n+1} = x_n+z+1$ for the first z found (in some algorithmic search) such that $\varphi_d(x_n + z + 1) = 0$ for $d = M_e((0, f_e(0))\,(1, f_e(1))\,\ldots\,(x_n, f_e(x_n))\,(x_n+1, 0)\,(x_n+2, 0)\,\ldots\,(x_n + z, 0))$. Let s be the time in which z is found. Define $f_e(x_{n+1}) = f_e(x_n) + s + 1$ and $f_e(y) = 0$ for $y \in \{x_n + 1, x_n + 2, \ldots, x_n + z\}$. Note that the search for z in some stage n might fail and then f_e is partial. Note that index for f_e can be found effectively from e. Let t be a recursive function such that $t(e)$ is an index of f_e.

Now let $\mathcal{C}$ contain all functions f_e where f_e is total plus all functions which are almost everywhere 0. Whenever M_e $\mathbf{Bc}$-learns all functions which are almost everywhere 0, in each stage n, the number z can be found and hence f_e is total. Furthermore, $M_e(f_e[x_{n+1}])$ is not a program for f_e and hence M_e is not a $\mathbf{Bc}$-learner for $\mathcal{C}$.

Note that there is a recursive function g such that whenever x_{n+1} exists in the definition of f_e and $n > e$ then $\Phi_{t(e)}(y) \leq g(f_e(x_{n+1}))$ for all $y \leq x_{n+1}$; this is because of the complexity of finding the z to define x_{n+1} is incorporated into the definition of $f_e(x_{n+1})$. Hence, using the characterization of Theorem 6, g is a recursive function that witnesses that $\{f_e : f_e \text{ is total}\}$ is in $\mathbf{TConsPart}^{can}$. Furthermore, the class of all almost everywhere 0 functions is in $\mathbf{TConsPart}^{can}$. As $\mathbf{TConsPart}^{can}$ is closed under union, $\mathcal{C} \in \mathbf{TConsPart}^{can}$. $\square$

This stands in contrast to the result from Theorem 8 that $\mathbf{TConsPart}^{arb} = \mathbf{TCons}^{arb} \subseteq \mathbf{Ex} \subseteq \mathbf{Bc}$. The next result shows that the self-describing functions are in $\mathbf{Ex}^0$ but not in $\mathbf{TConsPart}^{can}$. The result is based on the fact that the self-describing functions do not satisfy the domination-property stated in Theorem 8. Here a function f is called self-describing if $\varphi_{f(0)} = f$.

Theorem 16 *The class* $\{f : \varphi_{f(0)} = f\}$ *of self-describing functions is in* $\mathbf{Ex}_0$ *but not in* $\mathbf{TConsPart}^{can}$ *and not in* $\mathbf{TConsPart}^{arb}[A]$ *for non-high oracles* A.

Proof. Let $\mathcal{C} = \{f : \varphi_{f(0)} = f\}$. Clearly, $\mathcal{C} \in \mathbf{Ex}_0$. Suppose by way of contradiction that $\mathcal{C} \in \mathbf{TConsPart}^{can}$ as witnessed by M. Then, by Kleene's recursion theorem, there exists an e such that $\varphi_e(0) = e$, and $\varphi_e(x+1) = \min(\{y > \varphi_e(x) : M(\varphi_e[x+1] \diamond (x+1, y)) > x\})$.

Note that φ_e is total, by consistency of M. It follows that M on φ_e outputs any program only finitely often, and thus does not $\mathbf{TConsPart}^{can}$-learn $\varphi_e \in \mathcal{C}$.

The second part of the theorem is witnessed by the fact that for every recursive function f, there exists a self-describing function g such that, for all indices e of g and almost all x, $g(x) \leq x$ and $\Phi_e(x) > f(x)$. Such a g can be constructed from f by using Kleene's recursion theorem such that $\varphi_{g(0)} = g$ and, for all $x > 0$, $g(x)$ is the minimal value $y \notin \{\varphi_e(x) : e < x \wedge \Phi_e(x) \leq f(x)\}$. Note that $g(x) \leq x$ for all $x \geq 1$.

Hence, whenever a function h dominates Φ_e for some index e of the self-describing function g then h dominates f. Thus it follows from the characterization in Theorem 8 that $\mathcal{C} \in \mathbf{TConsPart}^{arb}[A]$ only if there is an A-recursive function h dominating all recursive functions. It follows that such an oracle A must be high. $\square$

It is easy to show that $\mathbf{Ex}_0 \subseteq \mathbf{RConsPart}^{arb}$ by simulating the corresponding $\mathbf{Ex}_0$-learner: as long as the $\mathbf{Ex}_0$-learner has not issued a hypothesis, the $\mathbf{RConsPart}^{arb}$-learner outputs functions extending the data in a canonical way; then, when the $\mathbf{Ex}_0$-learner has output its unique hypothesis, the $\mathbf{RConsPart}^{arb}$-learner just copies it.

Corollary 17 $\mathbf{RCons}^{arb} \not\subseteq \mathbf{TConsPart}^{can}$.

While $\mathbf{Ex}_0 \subseteq \mathbf{RConsPart}^{arb}$, this cannot be improved to permitting one mind change as shown by the following theorem. Only for omniscient oracles holds the inclusion, if a non-omniscient oracle is on the right side, the inclusion fails.

Theorem 18 **(a)** $\mathbf{Ex}_1 \subseteq \mathbf{ConsPart}^{can}[A]$ *iff A has hyperimmune Turing degree;*
(b) $\mathbf{Ex}_1 \subseteq \mathbf{ConsPart}^{arb}[A]$ *iff A is high.*

Proof. The class $\mathcal{C}$, which is used in both parts (a) and (b), is the union of the following two classes:

$$\begin{aligned}\mathcal{C}_0 &= \{f : \varphi_{f(0)-1} = f \wedge \forall x\,[f(x) \geq 1]\};\\ \mathcal{C}_1 &= \{f : \exists x \forall y [f(y) = 0 \Leftrightarrow y \geq x]\}.\end{aligned}$$

It is easy to see that the class $\mathcal{C}$ is in $\mathbf{Ex}_1$: the learner first conjectures $f(0)-1$ and then updates the current conjecture to $f(0)f(1)\ldots f(n)0^\infty$ at the first time it has seen all the data $(0, f(0)), (1, f(1)), \ldots, (n, f(n))$ and $f(n) = 0$.

A $\mathbf{ConsPart}^{can}$-learner or $\mathbf{ConsPart}^{arb}$-learner for $\mathcal{C}$ is defined and consistent on all inputs σ which do not contain a pair of the form $(x, 0)$. This is due to the fact that every such σ is extended by a function in $\mathcal{C}_1$. Thus, from a given $\mathbf{ConsPart}^{can}$-learner ($\mathbf{ConsPart}^{arb}$-learner) for $\mathcal{C}$, one can build a $\mathbf{TConsPart}^{can}$-learner ($\mathbf{TConsPart}^{arb}$-learner) for the class of self-describing functions by translating in the data every pair (x, y) to $(x, y+1)$ and then translating back the hypothesis e to e' with $\varphi_{e'}(x) = y \Leftrightarrow \varphi_e(x) = y+1$. By Theorem 16 and Corollary 7, the class of self-describing functions is not in $\mathbf{TConsPart}^{can}[A]$ for sets A of hyperimmune-free Turing degree and hence $\mathcal{C} \notin \mathbf{ConsPart}^{can}[A]$ for these A. By Theorem 16, the class of self-describing functions is not in $\mathbf{TConsPart}^{arb}[A]$ for non-high oracles A; so it follows that $\mathcal{C} \notin \mathbf{ConsPart}^{arb}[A]$ for non-high A. This completes the negative parts of (a) and (b).

The positive part of (a) follows from Theorem 6 and the positive part of (b) follows from Corollary 9. $\square$

The next result provides a class which is $\mathbf{Ex}_1$-learnable and $\mathbf{Cons}^{arb}$-learnable but which is not $\mathbf{RConsPart}^{can}$-learnable. One could modify the class such that the positive parts would be realized by the same learner, but that would make the whole proof very technical and therefore only the slightly weaker version is given. The idea would be to use the class $\{g_f : f \in \mathcal{C}\}$ where $g_f(n) = \langle f(0), f(1), \ldots, f(n)\rangle$.

Theorem 19 $\mathbf{Ex}_1 \cap \mathbf{Cons}^{arb} \not\subseteq \mathbf{RConsPart}^{can}$.

Proof. Let $M_0, M_1, \ldots$ denote a recursive enumeration of all the learning machines. We first define (recursively in i) a function g_i (which may be partial) and satisfies $g_i(0) = i$.

In stage $s > 0$, the algorithm will try to define g_i on input s. If g_i is total then it will not be $\mathbf{RConsPart}^{can}$-identified by M_i. Otherwise, a suitable extension of g_i would not be $\mathbf{RConsPart}^{can}$-identified by M_i.

Let $P_i^s = \{M_i(g_i[m]) : m \leq s\}$, denote the set of programs output by M_i on g_i as known at the beginning of stage s. Note that, if stage s exists, then we have defined g_i on inputs below s prior to the start of stage s, thus $g_i[s]$ is known at the beginning of stage s. Note that $|P_i^s| \leq s+1$.

Stage 0:

1. Let $g_i(0) = i$.
2. Go to stage 1.

End stage 0.

Stage s with $s > 0$:

1. Search for $y \leq s+1$ such that $M_i(g_i[s] \diamond (s, y))\!\downarrow\, \notin P_i^s$.
2. If and when such a y is found, let $g_i(s) = y$.
3. Go to stage $s+1$.

End stage s.

If g_i is not total, let s be the stage which starts but does not finish and let

$$f_{i,j}(x) = \begin{cases} g_i(x), & \text{if } x < s;\\ j, & \text{otherwise.}\end{cases}$$

Now define

$$\begin{aligned}\mathcal{C} &= \{g_i : g_i \text{ is total and } M_i \text{ is total}\}\\ &\cup \{f_{i,j} : g_i \text{ is not total and } M_i \text{ is total}\}.\end{aligned}$$

It is easy to verify that $\mathcal{C} \notin \mathbf{RConsPart}^{can}$. For this suppose M_i is total. If g_i is total then M_i does not $\mathbf{RConsPart}^{can}$-identify g_i (as the program output by M_i at $g_i[s+1]$ is not present in P_i^s). If g_i is not total (where stage s started but did not finish), then either M_i is not total or M_i is inconsistent on $g_i[s] \diamond (s, j)$, for some $j \leq s+1$ (note that P_i^s contains at most $s+1$ programs); thus M_i does not $\mathbf{RConsPart}^{can}$-identify at least one of $f_{i,j}$.

On the other hand, it can easily be verified that one can $\mathbf{Cons}^{arb}$-identify $\mathcal{C}$. A learner may output a program for 0-extension of the input, if $(0, i)$ is not present in the input for any i. Otherwise, learner outputs program for g_i (where $(0, i)$ is in the input) until it receives an input σ such that for some s with $1 \leq s \leq \max(\{w : (w, x) \in \text{content}(\sigma)\})$, in stage s in the construction of g_i, $M_i(g_i[s] \diamond (s, y))$ is defined for all $y \leq s+1$ and each of these $M_i(g_i[s] \diamond (s, y))$ are members of P_i^s; in that case, the learner outputs a canonical program for f given as

$$f(x) = \begin{cases} y, & \text{if } (x, y) \in \text{content}(\sigma);\\ z, & \text{otherwise;}\\ & \text{where } (x', z) \in \text{content}(\sigma) \text{ and } x' =\\ & \max(\{y : (y, w) \in \text{content}(\sigma) \text{ for some } w\}).\end{cases}$$

It is easy to see that the above learner would $\mathbf{Cons}$-identify $\mathcal{C}$.

To $\mathbf{Ex}_1$-identify $\mathcal{C}$, note that a learner can initially output ? until it sees $(0, i)$ in the input, for some i. At which point the learner outputs a program for g_i (which can be obtained effectively from i). The learner then continues to output this program until it receives an input σ and finds an s with $1 \leq s \leq \max(\{w : (w, x) \in \text{content}(\sigma)\})$ such that, in stage s in the construction of g_i, $M_i(g_i[s] \diamond (s, y))$ is defined for all $y \leq s+1$ and each of these $M_i(g_i[s] \diamond (s, y))$ are members of P_i^s and the input σ contains (x, y_x), for some y_x, for all $x \leq s$; in that case, the learner outputs a canonical program for

$$f(x) = \begin{cases} y, & \text{if } x < s,\ (x, y) \in \text{content}(\sigma);\\ z, & \text{otherwise; where } (s, z) \in \text{content}(\sigma).\end{cases}$$

and never changes its mind from then onwards. $\square$

It is open at present whether above theorem can be generalized such that, for every A of hyperimmune-free Turing degree, $\mathbf{Cons}^{arb} \not\subseteq \mathbf{RConsPart}^{can}[A]$. Similarly, it is open whether, for non-high A, $\mathbf{Cons}^{arb} \not\subseteq \mathbf{RConsPart}^{arb}[A]$.

7 Learning Partial Functions

In this section we consider learning of partial functions. In this situation, the input to the learner is a text for the partial function. For learnability, instead of requiring the learnt program e to be a program for η we require it to be a program for an extension of η. (This learnt program is the final program obtained in the limit, for **Ex**-learning, and the program output infinitely often for **Part**-learning.) The notion of consistency is as before.

The learnt program above may be required to be a partial extension or total extension of the input; by a result of Gold [10] it is impossible to avoid extensions when learning all partial-recursive functions, even if one uses a very strong oracle. The two theorems below consider each of these possibilities.

Theorem 20 *For learning partial functions by partial extensions,* $\mathcal{P} \in \mathbf{ConsPart}^{arb}[A]$ *iff* $K \leq_T A$.

Proof. ($\Rightarrow$) Suppose by way of contradiction that $K \not\leq_T A$ and M $\mathbf{ConsPart}^{arb}[A]$-learns $\mathcal{P}$ from arbitrary texts. Note that M must be total and consistent on all members of SEQ. Define A-recursive function g as follows.

$$\begin{aligned} g^A(n) \quad = \quad & \min(\{s : \forall \sigma \in \text{SEQ with } |\sigma| \leq 2n \\ & \text{and content}(\sigma) \subseteq \{(x,y) : y \leq x, x \leq n\} \\ & [\text{content}(\sigma) \subseteq \varphi_{M^A(\sigma),s}]\}) \end{aligned}$$

Let $\eta(x) = \min(\mathbb{N} - \{\varphi_{e,\Phi_x(x)}(x) : e < x\})$ (here $\eta(x)$ is undefined, if $\varphi_x(x)\uparrow$).

Now as $K \not\leq_T A$, $g^A(x) \leq \Phi_x(x) < \infty$ for infinitely many $x \in K$. Let $u_0, u_1, \ldots$ be an ascending listing of these numbers x. Now consider M on the input text $(u_0, \eta(u_0))$, $(x_0, \eta(x_0)), (u_1, \eta(u_1)), (x_1, \eta(x_1)), \ldots$, where $x_0, x_1, \ldots$ is an ascending list of the domain of η.

As M $\mathbf{Part}^{arb}$-identifies η, there is a number e which is output infinitely often on the above text. If $n > e$ and e is output after $(u_n, \eta(u_n))$ or $(x_n, \eta(x_n))$ then

- $\varphi_{e,g^A(u_n)}(u_n) = \eta(u_n)$, by definition of g and consistency of M, and
- $\eta(u_n)\downarrow \neq \varphi_{e,\Phi_{u_n}(u_n)}(u_n)$, by definition of η.

But this contradicts $g^A(u_n) \leq \Phi_{u_n}(u_n)$.

($\Leftarrow$) Using oracle for K, one can check consistency. Thus, one can convert any partial learner M (without oracles) for $\mathcal{P}$ to a consistent partial learner M' by outputting the output of M, if it is consistent, and some new program (not output earlier) consistent with the input, otherwise. As $\mathcal{P} \in \mathbf{Part}^{arb}$ (see [18]) the direction ($\Leftarrow$) follows. $\square$

Theorem 21 *Let* $P = \{\eta \in \mathcal{P} : (\exists f \in \mathcal{R})[\eta \subseteq f]\}$.

For learning partial functions by total extensions, $P \in \mathbf{ConsPart}^{arb}[A]$ *iff A is high.*

Proof. The necessity of A to be high follows from Corollary 9.

To see that $P \in \mathbf{ConsPart}^{arb}[A]$, for high A, consider the following learner.

Let H be an A-recursive function such that, for all $i \in \mathbb{N}$, $\lim_{t\to\infty} H^A(i,t) = 1$ if φ_i is total, and $\lim_{t\to\infty} H^A(i,t) = 0$ if φ_i is not total.

The learner, on input σ, outputs the first i found such that:

(a) content(σ) $\subseteq \varphi_i$,

(b) there exists a $t \geq |\sigma|$ with $H^A(i,t) = 1$ and

(c) for all $j < i$, either there exists a $t \geq |\sigma|$ such that $H^A(j,t) = 0$, or there exists $(x,y) \in$ content(σ) such that $\varphi_j(x)\downarrow \neq y$.

It can be easily verified that the above learner witnesses $P \in \mathbf{ConsPart}^{arb}[A]$. $\square$

8 Conclusion

In this paper we considered consistent partial identification of functions. We gave several characterizations about which oracles allow a learner to consistenly partially identify all the recursive functions. In particular, Theorem 6 showed that the class $\mathcal{R}$ of all recursive functions can be $\mathbf{ConsPart}^{can}[A]$-identified ($\mathbf{TConsPart}^{can}[A]$-identified, $\mathbf{RConsPart}^{can}[A]$-identified) iff A has a hyperimmune Turing degree. Furthermore, Corollary 9 showed that $\mathcal{R}$ can be $\mathbf{ConsPart}^{arb}$-identified ($\mathbf{TConsPart}^{arb}[A]$-identified, $\mathbf{RConsPart}^{arb}[A]$-identified) iff A is high. Additionally, some characterizations of $\mathbf{TConsPart}^{can}[A]$ and $\mathbf{TConsPart}^{arb}[A]$-learnability in terms of A-recursive functions being, infinitely often or almost always, above the time to compute the function to be learnt were obtained for arbitrary oracles A, see Theorem 6 and Theorem 8. Theorem 11 showed that $\mathbf{ConsPart}^{can}[A] \subseteq \mathbf{ConsPart}^{can}[B]$ iff B has hyperimmune Turing degree or $A \leq_T B'$ and $\mathbf{ConsPart}^{can}[A] \subseteq \mathbf{ConsPart}^{can}[B]$ iff B has hyperimmune Turing degree or $A \leq_T B \oplus K$.

We also compared learning from canonical versus arbitrary texts, and showed in particular that there are oracles A for which $\mathbf{TConsPart}^{arb}$ degree is trivial but $\mathbf{TConsPart}^{can}$ degree is omniscient. This is in particular true for an oracle A which is 1-generic and satisfies $A \leq_T K$.

We also showed relationships between various consistent partial identification criteria without oracles and compared them with **Bc** and **Ex**-learning criteria. In particular, there are classes which can be $\mathbf{TConsPart}^{can}$-identified but not **Bc**-identified. On the other hand, there are classes which can be **Ex**-identified with one mind change, but cannot be $\mathbf{ConsPart}^{can}$-identified.

One of our surprising results, see Theorem 8, is the equivalence $\mathbf{TConsPart}^{arb} = \mathbf{TCons}^{arb}$. It is an open problem whether this also holds for the other consistency criteria: Is $\mathbf{RConsPart}^{arb} = \mathbf{RCons}^{arb}$ and $\mathbf{ConsPart}^{arb} = \mathbf{Cons}^{arb}$? These equivalences do not hold for learning from canonical text.

Acknowledgments. We would like to thank Thomas Zeugmann for discussions on the topics of this paper. We also thank the anonymous referees for several helpful comments.

References

[1] Lenny Adleman and Manuel Blum. Inductive inference and unsolvability. *The Journal of Symbolic Logic*, 56:891–900, 1991.

[2] Janis Bārzdiņš. Inductive inference of automata, functions and programs. In *Proceedings of the 20th International Congress of Mathematicians, Vancouver*, pages 455–560, 1974. In Russian. English translation in American Mathematical Society Translations: Series 2, 109:107-112, 1977.

[3] Janis Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 82–88. Latvian State University, 1974. In Russian.

[4] Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[5] Manuel Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[6] John Case and Carl Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[7] Lance Fortnow, William Gasarch, Sanjay Jain, Efim Kinber, Martin Kummer, Steven Kurtz, Mark Pleszkoch, Theodore Slaman, Robert Solovay and Frank Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.

[8] Richard Friedberg. Three theorems on recursive enumeration. *Journal of Symbolic Logic*, 23(3):309–316, 1958.

[9] William Gasarch and Mark Pleszkoch. Learning via queries to an oracle. *Proceedings of the Second Annual Conference on Computational Learning Theory* (COLT), 214–229, 1989.

[10] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[11] Gunter Grieser. Reflective inductive inference of recursive functions. *Theoretical Computer Science A*, 397(1–3):57–69, 2008. Special Issue on Forty Years of Inductive Inference. Dedicated to the 60th Birthday of Rolf Wiehagen.

[12] Sanjay Jain, Daniel Osherson, James Royer and Arun Sharma. *Systems that Learn: An Introduction to Learning Theory*. MIT Press, Cambridge, Massachusetts, second edition, 1999.

[13] Klaus Peter Jantke and Hans-Rainer Beick. Combining postulates of naturalness in inductive inference. *Journal of Information Processing and Cybernetics (EIK)*, 17:465–484, 1981.

[14] Martin Kummer and Frank Stephan. On the structure of degrees of inferability. Journal of Computer and System Sciences, Special Issue COLT 1993, 52:214–238, 1996.

[15] Donald Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:295–310, 1966.

[16] Piergiorgio Odifreddi. *Classical Recursion Theory*. North-Holland, Amsterdam, 1989.

[17] Piergiorgio Odifreddi. *Classical Recursion Theory, Volume II*. Elsevier, Amsterdam, 1999.

[18] Daniel Osherson, Micheal Stob and Scott Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, 1986.

[19] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967. Reprinted by MIT Press in 1987.

[20] Robert I. Soare. *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets*. Springer-Verlag, Heidelberg, 1987.

[21] Rolf Wiehagen and Walter Liepe. Charakteristische Eigenschaften von erkennbaren Klassen rekursiver Funktionen. *Journal of Information Processing and Cybernetics (EIK)*, 12:421–438, 1976.

[22] Rolf Wiehagen and Thomas Zeugmann. Learning and consistency. *Algorithmic Learning for Knowledge-Based Systems*, GOSLER Final Report, Springer LNAI 961:1–24, 1995.