Learning Programs With an Easy to Calculate Set of Errors [0]

by

William I. Gasarch [1]
Ramesh K. Sitaraman [2]
Carl H. Smith [3]

and

Mahendran Velauthapillai [4]

I. Introduction

Putnam [12] was the first to notice that there was no mechanism capable of learning all the computable functions. Gold [6] was the first to formally prove, in full generality, Putnam's speculation. In order to enable the automatic learning of larger classes of functions, the Blums [2] relaxed the criteria of successful learning allowing the inference machine to produce programs computing finite variants of desired function. In [3] an infinite hierarchy of larger and larger classes of inferrible functions is exhibited based on counting precisely the number of errors. Inference via programs with infinitely many errors, distributed sparsely through out the domain,

---

[0] A preliminary version of this work appeared at the Workshop on Computational Learning Theory, Cambridge MA, 1988.

[1] Affiliated with the University of Maryland Department of Computer Science and the University of Maryland Institute for Advanced Computer Studies.

Supported, in part, by National Science Foundation Grant CCR 8803641.

[2] Affiliated with the Princeton University Department of Computer Science. Much of this work was done while the second author was affiliated with the University of Maryland Department of Computer Science.

[3] Affiliated with the University of Maryland Department of Computer Science and the University of Maryland Institute for Advanced Computer Studies.

Supported, in part, by National Science Foundation Grant CCR 870110. Much of this work was done while the third author was on leave at the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[4] Affiliated with the Georgetown University Department of Computer Science.

has been investigated in [14,16]. In this paper we investigate inference where to be successful, the mechanism must produce a program that is allowed infinitely many errors, but the anomalous points must form some easy to describe (regular, linear time decidable, etc.) set.

We work with the standard recursion theoretic model of inductive inference [1]. An inductive inference machine (IIM) is an algorithmic device that inputs the graph of some recursive function an ordered pair at a time and, while doing so, outputs programs intended to compute the input function. Natural numbers ($\mathbb{N}$) serve as names for programs. Program i computes the function φ_i. The functions $\varphi_0, \varphi_1, \ldots$ form an *acceptable programming system* [9]. An IIM M, inputting the partial function ψ, *converges* if the sequence of program produced by M is finite *or* almost all of the programs in the sequence are syntactically identical. If the *final* program output by M under such circumstances is i, then we say that $M(\psi)$ converges to i. We may assume without loss of generality that the convergence of M to i is independent of the order in which ψ is presented to M [2]. M identifies, or infers, ψ if $M(\psi)$ converges to an i such that $\psi \subseteq \varphi_i$. EX denotes the class of sets that are inferrible by some IIM.

In previous work on the inference of programs with anomalies, the definition required the IIM to produce a program that was correct except *perhaps* on some set of inputs. For this work, we will require that the final program of the IIM be correct everywhere except on precisely some type of set. Let L be a class of sets, e.g. regular sets, polynomial time decidable sets, etc. An IIM M *L-identifies* ψ if there is an $S \in L$ such that $M(\psi)$ converges to a program i such that $\psi \subseteq \varphi_i$ except for points in S and $\psi(x) \neq \varphi_i(x)$ for all $x \in S$. Note that the function φ_i need not be total. EX^L denotes the class of all sets that are L-identifiable by some IIM. A ramification of this definition is that we discuss the inference of partial recursive functions. Our results will hold even if we demand the stronger convergence condition that $\psi = \varphi_i$.

Various classes L fall within the domain of our results. Since all the singleton sets are regular, and constant time computable, by the results of [3], EX is a strict subset of all the classes EX^L examined below. Furthermore, the well studied class BC contains sets not in the classes EX^L studied below. Whether or not these classes are completely contained in BC is not known.

We prove that the REGULAR-inferrible sets are a strict subset of the CONTEXT FREE-inferrible sets. The only property of the regular and context free language used in the proof is that there is a context free language ($\{1^i0^i \mid i \geq 1\}$) that is (roughly) immune with respect to the regular sets. This suggests a generalization based a notion of *relative immunity* to be defined below. This generalization has several corollaries: the CONTEXT FREE-inferrible sets are a strict subset of the CONTEXT SENSITIVE ones; for any k, the DTIME(n^k)-inferrible sets are a strict subset of the DTIME(n^{k+1})-inferrible sets; and the POLYNOMIAL TIME-inferrible sets are a strict subset of the DTIME(2^n) ones.

The regular sets are particularly interesting to us because of their simplicity. Consequently, we examine subsets of $EX^{REGULAR}$. Let REG_n denote the subclass of the regular sets that are recognized by an n state deterministic finite automaton. We show that for all n, $EX^{REG_n} \subset EX^{REG_{n+1}}$.

We also consider inference paradigms where the number of times an IIM can alter its conjecture is restricted [3]. EX_n denotes the subset of EX that includes only the sets that can be inferred by IIMs that change their conjecture at most n times. The classes EX_n^L are defined similarly. We prove $EX_{n+1} - EX_n^L \neq \emptyset$, extending a result of [3].

II. Inference with a regular set of errors

We need a new notion of immunity [13] to state our results.

Definition: If A is a class of sets (languages) and L is set, then L is *A-immune* if no infinite subset of L is in A.

The proofs depend on the slightly stronger notion of relative immunity given by the following.

Definition: If A is a class of sets (languages) and L is set, then L is *A-fvimmune* if no finite variant of an infinite subset of L is in A.

To use these notions, it will be necessary to discuss functions that are equivalent except on a set. Often this set will be described as words of some language. We make implicit the straightforward association between words over a given alphabet and the natural numbers. As a technical convention, we assume that 0 corresponds to the empty string. With this in mind, if L is a language and f and g are functions such that $f = g$ except perhaps on some elements of L, we write $f =^{\bar{L}} g$. For a function f and a set L, f restricted to domain L is denoted by $f|_L$. To save on some notational complexity, instead of proving our most general result, we first prove.

THEOREM 1. $EX^{REGULAR} \subset EX^{CONTEXT\ FREE}$.

Proof: The fact that every regular language is also context free establishes the inclusion $EX^{REGULAR} \subseteq EX^{CONTEXT\ FREE}$. Let $L = \{1^i 0^i \mid i \geq 1\}$, a language known to be Context Free [7]. Note that L is REGULAR-fvimmune. Let

$$S = \{f \mid \text{range } f|_L \subseteq \{0,1\} \text{ and } \varphi_{f(0)} =^{\bar{L}} f\}.$$

Note that the set S may contain *partial* recursive functions.

Let M_0 be an IIM that, on input from the graph of some $f \in S$, waits until the value $f(0)$ has been input and outputs a program for the following partial recursive function:

$$\psi(x) = \begin{cases} 2 & \text{if } x \in L; \\ \varphi_{f(0)}(x) & \text{otherwise.} \end{cases}$$

The only program output by M_0 computes f everywhere except precisely on arguments $x \in L$. Hence, $S \in EX^{CONTEXT\ FREE}$.

To complete the proof, it remains to show that $S \notin EX^{REGULAR}$. This is accomplished by the construction of an $f \in S$ such that $f \notin EX^{REGULAR}(M)$ uniformly in an IIM M. Let M, an IIM, be given. Below we describe a program e that computes a partial recursive function with an extension being the desired f.

Begin program e. On input x, successively execute stages $s \geq 0$ below until (if ever) $\varphi_e(x)$ is defined. The finite portion of φ_e determined prior to stage s will be denoted by σ^s. Initialize the construction by setting $\sigma^0 = \{(0,e)\}$ via the recursion theorem [8]. Let δ^s denote the set of values $\leq s$ that are *not* in the domain of σ^s. At stage s we will attempt to define φ_e on arguments in δ^s.

Begin Stage s. Let $q = M(\sigma^s)$, M's most recent conjecture on the portion of φ_e defined so far. First look for a $\tau \supset \sigma^s$ such that $\tau \subseteq (\sigma^s \cup \{(x,0) \mid x \in \delta^s\})$ and $M(\tau) \neq q$. Since there are only finitely many candidate τ's, the search for one takes a finite amount of time. If such a τ is found, set $\sigma^{s+1} = \tau$ and go to stage $s+1$. Otherwise, let $C = \{x \mid x \in \delta^s \text{ and } \varphi_q(x) \text{ is defined in } \leq s \text{ steps}\}$ and set

$$\sigma^{s+1} = \sigma^s \cup \{(x, \varphi_q(x)) \mid x \in C \wedge x \notin L\} \cup \{(x, 1 \dot{-} \varphi_q(x)) \mid x \in C \wedge x \in L\}.$$

End Stage s.

End Program e.

Notice that, by virture of the initialization, $\varphi_e \in S$. If M, on input from φ_e, never converges, then M fails to identify φ_e by any EX type of inference criteria. In particular, $\varphi_e \notin EX^{REGULAR}(M)$. Suppose then that $M(\varphi_e)$ converges to $q = M(\sigma^t)$ for all $t \geq s$, for some s. Then at and past stage s, φ_e is defined to match φ_q on precisely $\bar{L}$ and to disagree on precisely L. Let $\delta = \lim_{s\to\infty} \delta^s$, e.g. the complement of the domain of φ_e. Two cases must be considered.

Case 1. $L \cap \delta$ is finite. In this case, φ_e is defined on all but finitely many elements of L. Since only finitely many elements of L were placed in the domain of φ_e before M converged to q and after that point, φ_e is defined to be different from φ_q everywhere except $L \cap \delta$, program $M(\varphi_e) = q$ is wrong precisely on a finite variant of L. Since L is REGULAR-fvimmune, $\varphi_e \notin EX^{REGULAR}(M)$.

Case 2. $L \cap \delta$ is infinite. In this case, φ_e is undefined on infinitely many elements of L. If φ_e is defined on infinitely many elements of L then φ_e agrees with φ_q, except on a finite variant of those (infinitely many) points. Since the points where φ_e and φ_q disagree is a finite varaint of an infinite subset of L, $\varphi_e \in S - EX^{REGULAR}(M)$. On the other hand, suppose φ_e is defined on only finitely many elements of L. Let $\psi = \varphi_e \cup \{(x,0) \mid x \in L \cap \delta\}$ a partial recursive function. Since ψ extends φ_e only on elements of L and $\psi(0) = e$, $\psi \in S$. Furthermore, since the construction explicitly searches for mind changes, $M(\psi) = M(\varphi_e) = q$. There are infinitely many points in the domain of ψ and in L, but not in the domain of φ_q. The set of points where φ_e and φ_q disagree is a finite variant of this infinte subset of L. Hence, $\psi \in S - EX^{REGULAR}(M)$. ☒

A slight modification to the above argument yields our main result:

THEOREM 2. Suppose A and B are two classes of sets such that $A \subseteq B$. If there is a recursive $L \in B$ that is A-fvimmune, then EX^A is properly contained in EX^B.

Proof: Same as Theorem 1, but using the A-fvimmune set S instead of L, A for $REGULAR$ and B for $CONTEXT\ FREE$. ☒

Note that if B is closed under finite variation then we can replace A-fvimmuity with A-immunity in the above result. In fact, in the applications of this theorem below, we do so implicitly. It turns out that that there are many pairs of sets that are relative immune. Consequently, there are several corollaries of Theorem 2, including Theorem 1.

COROLLARY 3. $EX^{CONTEXT\ FREE} \subset EX^{CONTEXT\ SENSITIVE}$.

Proof: The language $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ is context sensitive. By the pumping lemma for context free languages, no infinite subset of L is context free. Hence, L is CONTEXT FREE-fvimmune and the result follows from Theorem 2. ☒

COROLLARY 4. For all $k \in \mathbb{N}$, $EX^{DTIME(n^k)} \subset EX^{DTIME(n^{k+1})}$.

Proof: It is easy to construct, by a wait and see diagonalization argument, a language L in $DTIME(n^{k+1})$ which is $DTIME(n^k)$-immune, see [5]. ☒

COROLLARY 5. $EX^P \subset EX^{DTIME(2^n)}$.

Proof: Similar to the proof of Corollary 4. ☒

The regular sets are particulaly interesting to us because of their simplicity. Let REG_n denote the subclass of the regular sets that are recognized by an n state deterministic finite automaton.

COROLLARY 6. For all n, $EX^{REG_n} \subset EX^{REG_{n+1}}$.

Proof: An argument similar to the pumping lemma for regular languages can be used to show that the set $L = \{1^i \mid i \equiv n \bmod (n+1)\}$ is REG_n-immune. The set L is clearly in REG_{n+1}. Theorem 2 yields the desired result. ☒

III. Inference with a bounded number of mind changes

In this section, we consider inference schemes with a restraint on the number of times an IIM can alter its conjecture [3]. EX_n denotes the subset of EX that includes only the sets that can be inferred by IIMs that change their conjecture at most n times. The classes EX_n^A are defined similarly.

THEOREM 7. $EX_0^{CONTEXT\ FREE} - EX^{REGULAR} \neq \emptyset$.

Proof: This theorem directly follows from the proof of Theorem 1. ☒

THEOREM 8. Let $\mathcal{A}$ be a class of sets such that there exists a recursive language L that is $\mathcal{A}$-fvimmune. For all n, $EX_{n+1} - EX_n^{\mathcal{A}} \neq \emptyset$.

Proof: We may assume without loss of generality that $0 \in \bar{L}$. First we prove the $n = 0$ case. Let S be the set of partial recursive functions, f, such that

(a.) $\forall x[x \in \bar{L} \Rightarrow \varphi_{f(0)}(x) = f(x)]$ and

(b.) The function $f|_L$ either has range $\{0\}$ or there is a y such that

(1.) $[x \in L \text{ and } x < y] \Rightarrow f(x) = 0$ and

(2.) $[x \in L \text{ and } x \geq y] \Rightarrow f(x) = 1$.

The following process EX_1 identifies S: wait for input $f(0)$ and output a program for the function:

$$g(x) = \begin{cases} \varphi_{f(0)}(x) & \text{if } x \in \bar{L}, \\ 0 & \text{if } x \in L. \end{cases}$$

If ever an $a \in L$ is discovered such that $f(a) = 1$ then output a program for the following function:

$$g(x) = \begin{cases} \varphi_{f(0)}(x) & \text{if } x \in \bar{L}, \\ 0 & \text{if } x \in L \text{ and } x < a, \\ 1 & \text{if } x \in L \text{ and } x \geq a. \end{cases}$$

We show that S is not $EX_0^{\mathcal{A}}$ inferrible. Suppose by way of contradiction that M is an IIM such that $S \subseteq EX_0^{\mathcal{A}}(M)$. We exhibit an $f \in S - EX_0^{\mathcal{A}}(M)$. Below we define a program, e, by implicit use of the recursion theorem, such that φ_e or some extension thereof computes the desired f. Let σ be the shortest initial segment of the following function:

$$\psi(x) = \begin{cases} e & \text{if } x = 0, \\ 0 & \text{otherwise.} \end{cases}$$

such that M, on input σ outputs a program. If no such σ exists, then ψ as defined above is a member of S that M fails to infer, by any criteria of success. Suppose then, without loss of generality, that such a σ exists and that $M(\sigma) = q$. Program e from the point of coercing the first conjecture from M, simulates that conjecture. In other words,

$$\varphi_e(x) = \begin{cases} \sigma(x) & \text{if } x \in \text{domain } \sigma, \\ \varphi_q(x) & \text{otherwise.} \end{cases}$$

There are two cases to consider.

Case 1. There are infinitely many $x \in L$ such that $\varphi_q(x)$ diverges or converges to a nonzero value. Define the possibly partial recursive function ψ by:

$$\psi(x) = \begin{cases} e & \text{if } x = 0, \\ 0 & \text{if } 1 \leq x < \text{length}(\sigma) \text{ or } x \in L, \\ \varphi_q(x) & \text{otherwise.} \end{cases}$$

Clearly, $\psi \in S$. The partial functions φ_q and ψ disagree on

(a.) the points of L where φ_q is nonzero or divergent, and

(b.) possibly a finite set of points in the domain of σ.

Thus the set of points where φ_q and ψ disagree is precisely a finite variant of an infinite subset of L. Since L is fvimmune, this set is not in $\mathcal{A}$.

Case 2. For all but finitely many $x \in L$, $\varphi_q(x) = 0$. Let ψ be:

$$\psi(x) = \begin{cases} e & \text{if } x = 0, \\ 0 & \text{if } 1 \leq x < \text{length}(\sigma), \\ \varphi_q(x) & \text{if } x > \text{length}(\sigma) \text{ and } x \in \bar{L}, \\ 1 & \text{if } x > \text{length}(\sigma) \text{ and } x \in L. \end{cases}$$

Clearly, $\psi \in S$. By reasoning similar to the previous case, M does not $EX_0^{\mathcal{A}}$ infer ψ.

For the general case of $EX_{n+1} - EX_n^{\mathcal{A}} \neq \emptyset$, take S to be the set of partial recursive functions that, when restricted to $\bar{L}$ may step from 0 to 1 to ... to $n+1$. ☒

Corollary 9. $\forall n, EX_{n+1} - EX_n^{REGULAR} \neq \emptyset$.

Proof: Immediate by Theorem 8. ☒

Corollary 10. $\forall n, EX_{n+1} - EX_n^{CONTEXT\ FREE} \neq \emptyset$.

Proof: Immediate by Theorem 8. ☒

IV. Relations to BC type inference

Here we consider a criteria where the IIM does not necessarily converge to a single program but rather to a sequence of programs. As long as the IIM eventually outputs nothing but programs to compute the input function, then the prediction strategy which always uses the IIM's most current conjecture will be **b**ehaviorally correct (BC) [3]. Formally an IIM M BC^a identifies f (written $f \in BC^a(M)$) if and only if when M fed with the graph of f outputs over time an infinite sequence of programs p_0, p_1, $\cdots$, such that $(\forall n)[\varphi_{p_n} =^a f]$. Note for EX type inference we require syntactic convergence and for BC type inference semantic convergence is required. Our final results shows that BC type inference and $EX^{REGULAR}$ are incomparable.

THEOREM 11. $BC - EX^{REGULAR} \neq \emptyset$.

Let $S = \{f \mid \overset{\infty}{\forall} x, \varphi_{f(x)} = f\}$, a set of partial recursive functions. Clearly, $S \in BC$. Let M be an IIM. The proof is completed by constructing, via the operator recursion theorem, an $f \in S$ that can not be $EX^{REGULAR}$ identified by M. Program e_0 is constructed in effective stages of finite extension below. Program e_1 is syntactically different from e_0 but computes the same function. At stage s program $p(s)$ will be constructed. These programs will refer to one another in accordance with the operator recursion theorem and one of them will be the sought after witness f. The idea is to diagonalize against the program output by M on the portion of φ_{e_0} determined so far. The diagonalizations must be performed only on some non regular set. Let $L \subset \mathbb{N}$ correspond to the set $\{1^n b 0 n \mid n \in \mathbb{N}\}$. $\varphi_{e_0}^s$ denotes the finite amount of φ_{e_0} determined prior to stage s, $\varphi_{e_0}^0 = \emptyset$

Stage s: Initialize $\varphi_{p(0)}$ to be $\varphi_{e_0}^s$. Let $q = M(\varphi_{e_0}^s)$, e.g. M's guess on all the input currently determined. (For stage 0, $q = \perp$.) Initialize $\sigma = \varphi_{e_0}^s$. Simultaneously, perform two sub computations specified below.

1. Repeatedly redefine σ by adding points $(x, p(s))$ where x is the least value in L and not in the domain of $\varphi_{e_0}^s$. This process stops only when a σ is arrived at such that $M(\sigma) \neq q$ or when an interrupt is received from the computation described in step 2. If a σ causing M to change its conjecture is found, then an interrupt is sent to the computation described in step 2.
2. Look for (by dovetailing) an x such that $x \in L$, x is not in the domain of $\varphi_{e_0}^s$, and $\varphi_q(x)$ is defined. If such an x is found, stop the dovtailing procedure and interrupt the computation described in step 1.

Notice that if neither of the two sub computations halts (and interrupts the other) then step 1 will define $\varphi_{p(s)}$ to be a finite variant of the constant $p(s)$ function. Otherwise, there are two ways of extending φ_{e_0}, depending on which of the above sub computations interrupts the other.

If a σ forcing a mind change is found in step 1 before (or at the same time as) an x is found in step 2, then set $\varphi_{e_0}^{s+1} = \sigma$, commit program $p(s)$ to simulate program e_0 from here on (there by making programs e_0 and $p(s)$ compute the same function) and go to stage $s + 1$.

If an x is found in step 2 before a mind change is found in step 1 then do the following. Set $i = 0$ unless $\varphi_q(x) = e_0$ in which case $i = 1$. Set $\varphi_{e_0}^{s+1} = \varphi_{e_0}^{s} \cup \{(x, e_i)\}$ and go to stage $s + 1$.

End Stage s.

There are two cases to consider, depending on whether or not each stage terminates. In either case, we will find a function $f \in L$ that cannot be $EX^{REGULAR}$ identified by M.

Case 1. Each stage s terminates. Then φ_{e_0} is a partial recursive function with an infinite domain. Let $f = \varphi_{e_0}$. The range of f includes e_0 and e_1 which are programs for f. If $p(s)$ is placed in the range of f, for some value of s, then $p(s)$ will be another program for f. Hence, $f \in L$. If $M(f)$ does not converge, then M cannot $EX^{REGULAR}$ identify f. Suppose that $M(f)$ converges, say to program i. Past some point in the construction of f, all extensions must be made by virtue of sub computation 2. Hence there are infinitely many $x \in \mathcal{C}$ such that $f(x) \neq \varphi_i(x)$. These x's correspond to an infinite subset of $\{a^n b^n \,|\, n \in \mathbb{N}\}$.

Case 2. Some stage s never terminates; Let s be the least such stage. Then program $p(s)$ computes a partial function with infinite domain. Let $f = \varphi_{p(s)}$. By the failure of sub computation 1 to terminate stage s, we know that $M(f)$ converges, say to i. Program i however is undefined on all $x \in \mathcal{C}$ that are not in the domain of $\varphi_{e_0}^{s}$, precisely where f is defined. ☒

THEOREM 12. $BC - EX^{CONTEXT\ FREE} \neq \emptyset$.

Proof: In the proof of Theorem 11, when constructing f the diagonalization was performed on a non regular set $\{a^n b^n \,|\, n \in N\}$. For the proof of this theorem, use a non context free set $\{a^n b^n c^n \,|\, n \in n\}$. The rest of the proof is similar to the proof of Theorem 11.

THEOREM 13. $(\forall a \in N)\ EX_0^{REGULAR} - BC^a \neq \emptyset$.

Proof: Let $L = 0^+$, clearly L is regular. Define

$$S = \{f \,|\, \text{range } (f|_L) \subseteq \{0,1\} \text{ and } \varphi_{f(0)} =^{\bar{L}} f\}.$$

Notice that the set S may contain partial recursive functions.

Let M_0 be an IIM that, on input from the graph of some $f \in S$, waits until the value $f(0)$ has been input and outputs a program for the following partial recursive function:

$$\psi(x) = \begin{cases} 2 & \text{if } x \in L; \\ \varphi_{f(0)}(x) & \text{otherwise.} \end{cases}$$

The only program output by M_0 computes f everywhere except precisely on arguments $x \in L$. Hence, $S \in EX^{REGULAR}$.

To complete the proof, it remains to show that $S \notin BC^a$. This is accomplished by the construction of an $f \in S$ such that $f \notin BC^a(M)$ for any IIM M.

Let M be any IIM and $a \in N$. Using the recursion theorem we will construct a program e_0 which will enable us to construct an $f \in S$ such that $f \notin BC^a(M)$. The construction is done in effective stages of finite extension. At stage s, the program e_0 tries to diagonalize against M's current output at $a + 1$ points.

Begin program e_0. On input x, succesively execute the stages $s \geq 0$ below until (if ever) $\varphi_{e_0}(x)$ is defined. $\varphi^s_{e_0}$ denotes the finite initial segment of φ_{e_0} determined prior to stage s. Set $\varphi^0_{e_0} = \emptyset$. σ^s_0 denotes $\varphi^s_{e_0}$.

Stage s. Search for distinct natural numbers $x_0, \cdots, x_a$ which belong to L and finite initial segments τ and ρ with range τ, range $\rho \subseteq \{0, 1\}$ such that $\sigma^s \subset \tau \subset \rho$ and $(\forall j \leq a)$ $[x_j \in$ domain $(\rho - \tau)$ and $\varphi_{M(\tau)}(x_j)$ converges $\neq \rho(x_j)]$.

If suitable $x_0, \cdots, x_a$, τ and ρ are found then set

$\varphi^{s+1}_{e_0} = \varphi^s_{e_0} \cup \{(x_j, 1 \dot{-} \rho(x_j)) \mid (0 \leq j \leq a)\} \cup \{(x, 0) \mid x \in \{\text{domain } (\rho - \varphi^s_{e_0} - \{x_0, \cdots, x_a\})\}$.

End stage s.

End program e_0.

Case 1. Suppose φ_{e_0} is total. Then, by construction, $\varphi_{e_0} \in S$. Past every stage s a τ is found such that $\tau \subset \varphi_{e_0}$ and $\varphi_{M(\tau)}$ is not an a-variant of φ_{e_0}. Therefore $\varphi_{e_0} \notin BC^a(M)$.

Case 2. Suppose φ_{e_0} is not total. Then choose the least stage s such that $\varphi_{e_0} = \varphi^s_{e_0}$ and set $f = \sigma^s \cup \{(x, 0) \mid x \in (L-$ domain $(\sigma^s)\}$. By construction $f \in S$, and for all $\tau \supset \sigma^s$, such that range $\tau \subseteq \{0, 1\}$, $\varphi_{M(\tau)}$ is not defined on infinitely many elements of L. Otherwise a suitable $x_0, \cdots, x_a$ would be found in some stage $s' \geq s$. Hence $f \notin BC^a(M)$. ☒

COROLLARY 14. $(\forall a \in N)\ EX_0^{CONTEXT\ FREE} - BC^a \neq \emptyset$.

Proof: $EX_0^{REGULAR} \subset EX_0^{CONTEXT\ FREE}$ the result follows. ☒

A stronger version of Theorem 13 holds.

THEOREM 15. For all infinite $S \subseteq \mathbb{N}$, for all $a \in \mathbb{N}$, $EX_0^S - BC^a \neq \emptyset$.

Proof: Replace the set L, in the proof of Theorem 13, with the set S. ☒

¿From Theorem 13 and Theorem 11 we can conclude that the BC classes and the $EX^{REGULAR}$ classes are incomparable. Simillarly from Theorem 12 and Corollary 14 we can also conclude that the BC classes and the $EX^{CONTEXT\ FREE}$ classes are incomparable.

V. Team inference

The Blums [2] constructed two sets of inferrible recursive functions whose union was not inferrible. Consequently, arbitrarily large collections of IIMs were considered [15]. A set of functions S is inferred by the team $M_1, M_2, \ldots, M_n$ if for each $f \in S$ there is an i, $1 \leq i \leq n$, such that $f \in EX(M_i)$. This leads to the following definition. For $n \geq 1$, $[1,n]EX_b^a = \{S \mid (\exists M_1, M_2, \ldots, M_n)$ IIMs, and for each $f \in S$ there is an $1 \leq i \leq n$ such that $f \in EX_b^a(M_i)\}$. Here you require at least one IIM to EX_b^a infer the input function. Also note that, for different f's in the set S, the machine which infers f in the team could be different.

Pitt [10] investigated and characterized probabilistic inductive inference. Suppose that M is an IIM that has a fair coin to toss that is trying to learn a program for the function f. For a fixed enumeration of the graph of f, the outcome of M applied to f depends only on the results of the coin tosses. Using the standard Borel measure on the possible sequences of coin tosses, the set of sequences for which $M(f) \downarrow$ to a program for f is measurable. Let M be an IIM, $a \in N$ and $0 \leq p \leq 1$, we say that $f \in EX^a\langle p\rangle(M)$ if and only if M EX^a infers f with probability p. The classes $EX^a\langle p\rangle$ is defined analogously. For $0 \leq p \leq 1$ and $a \in N$ $EX^a\langle p\rangle = \{S \mid (\exists M)[S \subseteq EX^a\langle p\rangle(M)]\}$.

A natural combination of the notions of team inference and probabilistic inference results in the definition of some new classes of functions [11]. For $m, n \geq 1$, $a \in N$ and $0 \leq p \leq 1$, a set of functions S is in the class $[m,n]EX^a\langle p\rangle$ if and only if $m \leq n$ and there exists probabilistic IIMs $M_1, M_2, \ldots, M_n$ such that for each function $f \in S$ there are $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that for all $1 \leq j \leq m$, $f \in EX^a\langle p\rangle(M_{i_j})$. Here we require at least m out of the n machines in the team, to infer f with probability at least p. Note that probabilistic inference is a special case of probabilistic team inference, the case where $m = 1$.

THEOREM 16. Let A be a class of sets such that there exists a language L that is A-fvimmune, then $(\forall n \in N)$ $[1, n+1]EX_0^0 - EX_n^A \neq \emptyset$.

Proof: By Theorem 8 we have that $EX_{n+1}^0 - EX_n^A \neq \emptyset$. But $EX_{n+1}^0 \subset [1, n+1]EX_0^0$ [4]. Hence the theorem follows. ☒

THEOREM 17. $(\forall n \geq 1)$ $(\forall a \in N)$ $EX_0^{REGULAR} - [1, n]EX_\star^a \neq \emptyset$.

Proof: Let $a \in N$ and $n \geq 1$ be given. Then by Theorem 5.1 of [15] $EX_0^{n(a+1)} - [1, n]EX_\star^a \neq \emptyset$. But clearly $EX_0^{n(a+1)} \subset EX_0^{REGULAR}$. Hence the theorem follows. ☒

COROLLARY 18. $(\forall p > 0)(\forall a \in N)$ $EX_0^{REGULAR} - EX^a\langle p\rangle \neq \emptyset$.

Proof: By [10] $EX_*^a = EX^a\langle p\rangle$, hence the theorem follows. ☒

VI. Conclusions

We continued the study of learning an approximation to the desired function. Rather than measure the variance between the desired function and the approximation, we accounted for the difficulty of deciding membership in the set points comprising the variance. Our results indicate that the more complex a decision procedure is allowed, the larger the class of functions that become inferrible.

References

1. ANGLUIN, D. AND SMITH, C. H. Inductive inference: theory and methods. *Computing Surveys 15* (1983), 237–269.
2. BLUM, L. AND BLUM, M. Toward a mathematical theory of inductive inference. *Information and Control 28* (1975), 125–155.
3. CASE, J. AND SMITH, C. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science 25,* 2 (1983), 193–220.
4. FREIVALDS, R., SMITH, C., AND VELAUTHAPILLAI, M. Trade–offs amongst parameters effecting the inductive inferribility of classes of recursive functions. *Information and Computation* (1989). To appear.
5. GESKE, J., HUYNH, D., AND SELMAN, A. Hierarchy Theorems for Almost Everywhere Hard Sets. *Lecture Notes in Computer Science (To Appear)* (1987). STACS 1987 Proceedings.
6. GOLD, E. M. Language identification in the limit. *Information and Control 10* (1967), 447–474.
7. HOPCROFT, J. AND ULLMAN, J. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Co., Reading, Mass., 1979.
8. KLEENE, S. On notation for ordinal numbers. *Journal of Symbolic Logic 3* (1938), 150–155.

9. MACHTEY, M. AND YOUNG, P. *An Introduction to the General Theory of Algorithms.* North-Holland, New York, New York, 1978.

10. PITT, L. A Characterization of Probabilistic Inference. *Journal of the ACM.* To appear.

11. PITT, L. AND SMITH, C. Probability and plurality for aggregations of learning machines. *Information and Computation 77* (1988), 77–92.

12. PUTNAM, H. Probability and confirmation. In *Mathematics, Matter and Method*, 1, Cambridge University Press, 1975. Originally appeared in 1963 as a Voice of America Lecture.

13. ROGERS, H. JR. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967.

14. ROYER, J. S. Inductive inference of approximations. *Information and Control 70,* 2/3 (1986), 156–178.

15. SMITH, C. H. The power of pluralism for automatic program synthesis. *Journal of the ACM 29,* 4 (1982), 1144–1165.

16. SMITH, C. H. AND VELAUTHAPILLAI, M. On the inference of approximate explanations. *Theoretical Computer Science.* To appear.