On a Question About Learning Nearly Minimal Programs

Sanjay Jain
Department of Information Systems and Computer Science
National University of Singapore
Singapore 0511
Republic of Singapore

March 11, 2007

Abstract

Identification of programs for computable functions from their graphs by algorithmic devices is a well studied problem in learning theory. Freivalds and Chen consider identification of 'minimal' and 'nearly minimal' programs for functions from their graphs. The present paper solves the following question left open by Chen: Is it the case that for any collection of computable functions, $\mathcal{C}$, such that some machine can finitely learn a nearly minimal $(n+1)$-error program for every function in $\mathcal{C}$, there exists another machine that can learn in the limit an n-error program (which need not be nearly minimal) for every function in $\mathcal{C}$? We answer this question negatively.
Keywords: Machine learning, inductive inference, recursion theory, minimal programs, methodology of science.

1 Introduction

A *function learning machine* $\mathbf{M}$ is an algorithmic device which returns a sequence of either programs or "don't know" indicators when presented with the graph of a target function f. It learns (identifies, explains) the input function f with at most m errors and n mind changes if its output sequence changes value at most n times and converges to a program which correctly computes f on all but at most m input values. A class of functions is learnable (identifiable), with at most m errors and n mind changes, if some algorithmic device learns each function in the class with at most m errors and n mind changes.

Freivalds [Fre75] and Chen [Che81, Che82] studied the effect of requiring that the final hypothesis held by the learner in the above model be of (nearly) minimal size. For nearly minimal identification with at most m errors and n mind changes one requires, in addition to identification with at most m errors and n mind changes, that the final programs be of nearly minimal size (see formal definitions in Section 3). We direct the reader to [Fre75, Che81, Che82] for results relating nearly minimal identification with other criteria of inference. It was left open in [Che82, Che81] whether any class of functions which can be nearly minimally identified with at most $n+1$ errors and 0 mind changes can also be identified using at most n errors and finite number of mind changes. In this paper we answer this question negatively. Along with the results in [Che82] this completes the relationship between different criteria of explanatory identification and nearly minimal identification.

We now proceed formally.

2 Notation

Recursion-theoretic concepts not explained below are treated in [Rog67]. N denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$; The symbols a, b, i, j, k, l, m, n, s, t, x, and y, with or without decorations (decorations are subscripts, superscripts and the like), range over natural numbers unless otherwise specified. $\mathrm{card}(S) \leq *$ means that cardinality of the set S is finite. $\max(\ ), \min(\ )$ denote the maximum and minimum of a set, respectively. By convention $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$.

$\mathcal{R}$ denotes the set of all total recursive functions. h, f, g, with or without decorations, range over total recursive functions. η, with or without decorations, ranges over partial functions. $\mathrm{domain}(\eta)$ denotes the domain of η. For $a \in N \cup \{*\}$, we say that $\eta_1 =^a \eta_2$ (read: η_1 is an a-variant of η_2) iff $\mathrm{card}(\{x \mid \eta_1(x) \neq \eta_2(x)\}) \leq a$. Thus, $\eta_1 =^* \eta_2$ means that η_1 and η_2 are finite variants of each other.

We let φ denote a standard acceptable programming system. φ_i denotes the partial recursive function computed by the i^{th} program in the standard acceptable programming system φ. We often refer to the i^{th} program as program i. In some contexts, p, with or without decorations, ranges over programs. In other contexts p ranges over total functions, with its range being interpreted as programs. For a recursive function f, $\mathrm{MinProg}(f)$ denotes the minimal program for f (in the φ system), i.e., $\mathrm{MinProg}(f) = \min(\{i \mid \varphi_i = f\})$.

$\langle i, j \rangle$ stands for an arbitrary computable one to one encoding of all pairs of natural numbers onto N [Rog67].

The quantifiers '$\overset{\infty}{\forall}$' and '$\overset{\infty}{\exists}$' means 'for all but finitely many' and 'there exist infinitely many' respectively.

3 Learning Paradigms

For any partial function η and any natural number n such that, for each $x < n$, $\eta(x)\downarrow$, we let $\eta[n]$ denote the finite initial segment $\{(x, \eta(x)) \mid x < n\}$. Let $\mathbf{INIT} = \{f[n] \mid f \in \mathcal{R} \ \wedge \ n \in N\}$. We let σ and τ, with or without decorations, range over **INIT**.

Definition 1 [Gol67] A *learning machine* is an algorithmic device which computes a mapping from **INIT** into $N \cup \{?\}$ such that, if $\mathbf{M}(f[n]) \neq ?$, then $\mathbf{M}(f[n+1]) \neq ?$.

We let $\mathbf{M}$, with or without decorations, range over learning machines. In Definition 1 above, '?' denotes the situation when $\mathbf{M}$ outputs "no conjecture" on some $\sigma \in \mathbf{INIT}$.

In Definition 2 below we spell out what it means for a learning machine to converge in the limit.

Definition 2 Suppose $\mathbf{M}$ is a learning machine and f is a computable function. $\mathbf{M}(f)\downarrow$ (read: $\mathbf{M}(f)$ *converges*) just in case $(\exists i)(\overset{\infty}{\forall} n)\ [\mathbf{M}(f[n]) = i]$. If $\mathbf{M}(f)\downarrow$, then $\mathbf{M}(f)$ is defined = the unique i such that $(\overset{\infty}{\forall} n)[\mathbf{M}(f[n]) = i]$, otherwise we say that $\mathbf{M}(f)$ diverges (written: $\mathbf{M}(f)\uparrow$).

3.1 Explanatory Function Identification

We now formally define the criteria of inference considered in this paper.

Definition 3 [Gol67, CS83, BB75, BF74] Suppose $a, b \in N \cup \{*\}$.

(a) A learning machine $\mathbf{M}$ is said to $\mathbf{Ex}_b^a$-*identify* f (written: $f \in \mathbf{Ex}_b^a(\mathbf{M})$) just in case $[(\exists i \mid \varphi_i =^a f)\ (\overset{\infty}{\forall}\ n)[\mathbf{M}(f[n]) = i]\ \wedge\ \mathrm{card}(\{n \mid ? \neq \mathbf{M}(f[n]) \neq \mathbf{M}(f[n+1])\}) \leq b]$.

(b) $\mathbf{Ex}_b^a = \{\mathcal{C} \mid (\exists \mathbf{M})[\mathcal{C} \subseteq \mathbf{Ex}_b^a(\mathbf{M})]\}$.

For a given f and $\mathbf{M}$, we refer to each instance of the case, $? \neq \mathbf{M}(f[n]) \neq \mathbf{M}(f[n+1])$ as a *mind change* by $\mathbf{M}$ on f. We often refer to $\mathbf{Ex}_*^a$ as $\mathbf{Ex}^a$, $\mathbf{Ex}_b^0$ as $\mathbf{Ex}_b$ and $\mathbf{Ex}_*^0$ as $\mathbf{Ex}$.

We next consider identification by nearly minimal programs.

Definition 4 [Fre75, Che82] Suppose $a, b \in N \cup \{*\}$.

(a) Suppose h is a recursive function. A learning machine $\mathbf{M}$ is said to h-$\mathbf{Mex}_b^a$-identify f (written $f \in h$-$\mathbf{Mex}_b^a(\mathbf{M})$) iff $\mathbf{M}$ $\mathbf{Ex}_b^a$-identifies f and $\mathbf{M}(f) \leq h(\mathrm{MinProg}(f))$.

(b) $\mathbf{Mex}_b^a = \{\mathcal{C} \mid (\exists \mathbf{M})(\exists h \in \mathcal{R})[\mathcal{C} \subseteq h\text{-}\mathbf{Mex}_b^a(\mathbf{M})]\}$.

We often refer to $\mathbf{Mex}_*^a$ as $\mathbf{Mex}^a$, $\mathbf{Mex}_b^0$ as $\mathbf{Mex}_b$ and $\mathbf{Mex}_*^0$ as $\mathbf{Mex}$. In Chen [Che82] (see also [Fre75]) it was shown that

Theorem 1 *For all $m, n \in N$, $a \in N \cup \{*\}$.*

(a) $\mathbf{Ex} - \mathbf{Mex}^n \neq \emptyset$.

(b) $\mathbf{Ex}_0^0 - \mathbf{Mex}_m^* \neq \emptyset$.

(c) $\mathbf{Ex}_m^a \subseteq \mathbf{Mex}^a$.

(d) $\mathbf{Ex}^* = \mathbf{Mex}^*$.

(e) $\mathbf{Mex}_{m+1}^0 - \mathbf{Ex}_m^* \neq \emptyset$.

(f) $\mathbf{Mex}^{n+1} - \mathbf{Ex}^n \neq \emptyset$.

However it was left open, whether $\mathbf{Mex}_0^{n+1} \subseteq \mathbf{Ex}^n$. We answer this question negatively in the next section. Along with the results in [Che82] this completes the relationship between different $\mathbf{Ex}_n^m$ and $\mathbf{Mex}_{n'}^{m'}$ criteria of inference.

The following proposition facilitates the proof of our result.

Proposition 1 *There exists a recursively enumerable sequence $\mathbf{M}_0, \mathbf{M}_1, \ldots$ of learning machines such that, for all learning machines $\mathbf{M}$,*

$$(\exists i)(\forall a \in N \cup \{*\})[\mathbf{Ex}^a(\mathbf{M}) \subseteq \mathbf{Ex}^a(\mathbf{M}_i)]$$

For a proof of the above proposition see for example [OSW86]. We let $\mathbf{M}_0, \mathbf{M}_1, \ldots$ be one such enumeration.

4 Result

Theorem 2 $(\forall n)[\mathbf{Mex}_0^{n+1} - \mathbf{Ex}^n \neq \emptyset]$.

PROOF. Fix n. We will describe a recursive 1–1 monotone increasing recursive function p (using operator recursion theorem [Cas74]) and a sequence of functions $f_0, f_1, \ldots$ below. These will satisfy the following properties:

(P1) $f_i(0) = i$.
(P2) $f_i(1) \leq i$.
(P3) $\varphi_{p(\langle i, f_i(1)\rangle)} =^{n+1} f_i$.
(P4) $\mathrm{MinProg}(f_i) \geq i$.
(P5) $f_i \notin \mathbf{Ex}^n(\mathbf{M}_i)$.

Let $\mathcal{C} = \{f_i \mid i \in N\}$. Let h be such that $h(i) = \max(\{p(\langle j,k\rangle) \mid j,k \leq i\})$.

For $f \in \mathcal{C}$, $\varphi_{p(\langle f(0), f(1)\rangle)} =^{n+1} f$ and $p(\langle f(0), f(1)\rangle) \leq h(f(0)) \leq h(\mathrm{MinProg}(f))$. Thus $\mathcal{C} \in \mathbf{Mex}_0^{n+1}$.

Also by property (P5) above it follows that $\mathcal{C} \notin \mathbf{Ex}^n$. It now remains to construct p and f_i's as claimed above. We first define p. f_i's will be defined later. By operator recursion theorem [Cas74] there exists a recursive monotone increasing p such that the functions $\varphi_{p(\langle i,j\rangle)}$ may be defined as follows (we only need to consider $\varphi_{p(\langle i,j\rangle)}$, for $j \leq i$).

Definition of $\varphi_{p(\langle i,j\rangle)}$.

$\varphi_{p(\langle i,j\rangle)}$ is defined in stages.

Initially let $\varphi_{p(\langle i,j\rangle)}(0) = i$, $\varphi_{p(\langle i,j\rangle)}(1) = j$.

Let x_s denote the least x such that $\varphi_{p(\langle i,j\rangle)}(x)$ is not defined before stage s. Thus, $x_0 = 2$. Note that if stage s is started, then at the beginning of stage s, $\varphi_{p(\langle i,j\rangle)}$ is defined on exactly the inputs $0, 1, \ldots, x_s - 1$.

Intuitively, step 3 below searches for suitable values for $\varphi_{p(\langle i,j\rangle)}$, on inputs $x_s, x_s+1, \ldots, x_s+n$, so that a mind change (by $\mathbf{M}_i$) can be forced. While this search is being done, step 2 extends $\varphi_{p(\langle i,j\rangle)}$ by defining it on inputs $x_s+n+1, x_s+n+2, \ldots$.

Go to stage 0.

```
Stage $s$
1.  Dovetail steps 2 and 3 until, if ever, step 3 succeeds. If and when step 3 succeeds, go to
      step 4.
2.  Let $y = x_s + n + 1$.
    repeat
          Let $\varphi_{p(\langle i,j\rangle)}(y) = 0$.
          Let $y = y + 1$.
    forever
3.  For each $k \in N$, let $g_k$ denote the function
```

$$g_k(x) = \begin{cases} \varphi_{p(\langle i,j\rangle)}(x), & \text{if } x < x_s; \\ k, & \text{if } x_s \leq x \leq x_s + n; \\ 0, & \text{otherwise.} \end{cases}$$

```
        Search for a $k \in N$ and $m > x_s + n$ such that $\mathbf{M}_i(g_k[m]) \neq \mathbf{M}_i(g_k[x_s])$.
    4.  If and when step 3 above succeeds, let $k, m$ be as found in step 3.
        For $x_s \leq x \leq x_s + n$, let $\varphi_{p(\langle i,j \rangle)}(x) = k$.
        For $x_s + n < x \leq m$, let $\varphi_{p(\langle i,j \rangle)}(x) = 0$.
        Go to stage $s + 1$.
    End stage $s$
End of definition of $\varphi_{p(\langle i,j \rangle)}$.
```

Now, we define f_i. Fix i. Pick $j \leq i$ such that, for all $l < i$, $\varphi_l(1) \neq j$ (by pigeonhole principle there exists such a j). Below, we will pick a suitable extension of $\varphi_{p(\langle i,j \rangle)}$ as f_i. Thus f_i would clearly satisfy properties P1 and P2. Since $\varphi_{p(\langle i,j \rangle)}$ is undefined on atmost $n+1$ inputs, f_i would also satisfy property P3. Note that, by the choice of j, if $\varphi_{p(\langle i,j \rangle)} \subseteq g$, then $\text{MinProg}(g) \geq i$ (since, for such g, $g(1) = j$, and for all $l < i$, $\varphi_l(1) \neq j$). Thus f_i would satisfy property P4. We now pick f_i based on the following two cases.

Case 1: In the construction of $\varphi_{p(\langle i,j \rangle)}$ all stages terminate.

In this case let $f_i = \varphi_{p(\langle i,j \rangle)}$. Since all stages terminate, $\mathbf{M}_i(f_i)$ diverges. Thus f_i satisfies property P5. Thus all the properties of f_i as claimed above are satisfied.

Case 2: In the construction of $\varphi_{p(\langle i,j \rangle)}$, some stage s starts but does not terminate.

In this case, for each $k \in N$, let g_k be as defined in stage s of the construction of $\varphi_{p(\langle i,j \rangle)}$. Note that for each $k \in N$, $\varphi_{p(\langle i,j \rangle)} =^{n+1} g_k$. Let x_s be as defined in the construction of $\varphi_{p(\langle i,j \rangle)}$. Now $\mathbf{M}_i$ on each g_k converges to $\mathbf{M}_i(\varphi_{p(\langle i,j \rangle)}[x_s])$. However, a program can compute an n variant of g_k, for at most finitely many k. Thus, there exists a g_k such that $\mathbf{M}_i$ does not $\mathbf{Ex}^n$-identify g_k and $\text{MinProg}(g_k) \geq i$. Let $f_i =$ one such g_k. Thus f_i satisfies property P5. Thus all the properties of f_i as claimed above are satisfied. ∎

As a corollary to Theorems 1 and 2 we have

Corollary 1 *Let* $a, b, a', b' \in N \cup \{*\}$.
$\mathbf{Mex}_b^a \subseteq \mathbf{Ex}_{b'}^{a'}$ *iff* $[a \leq a' \ \wedge \ b \leq b']$.
$\mathbf{Ex}_b^a \subseteq \mathbf{Mex}_{b'}^{a'}$ *iff* $[b' = * \ \wedge \ a \leq a' \ \wedge \ [a' = * \ \vee \ b \neq *]]$.

5 Acknowledgements

I would like to thank Ganesh Baliga, John Case, Suraj Mandayam and Arun Sharma for helpful discussions and comments. I thank the referees for several helpful suggestions.

References

[BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[BF74] J. M. Barzdin and R. Freivalds. Prediction and limiting synthesis of recursively enumerable classes of functions. *Latvijas Valsts Univ. Zimatm. Raksti*, 210:101–111, 1974.

[Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.

[Che81] K. Chen. *Tradeoffs in Machine Inductive Inference*. PhD thesis, SUNY at Buffalo, 1981.

[Che82] K. Chen. Tradeoffs in inductive inference of nearly minimal sized programs. *Information and Control*, 52:68–86, 1982.

[CS83] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[Fre75] R. Freivalds. Minimal Gödel numbers and their identification in the limit. *Lecture Notes in Computer Science*, 32:219–225, 1975.

[Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[OSW86] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Mass., 1986.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967. Reprinted by MIT Press, Cambridge, Massachusetts in 1987.