Input-Dependence in Function-Learning

Sanjay Jain$^{\star 1}$, Eric Martin$^{\star\star 2}$ and Frank Stephan^{*3}

[1] Department of Computer Science,
National University of Singapore,
Singapore 117590, Republic of Singapore.
sanjay@comp.nus.edu.sg

[2] Department of Computer Science and Engineering,
The University of New South Wales,
Sydney 2052, NSW, Commonwealth of Australia.
emartin@cse.unsw.edu.au

[3] Department of Mathematics,
National University of Singapore,
Singapore 117543, Republic of Singapore.
fstephan@comp.nus.edu.sg

Abstract. In the standard model of inductive inference, a learner gets as input the graph of a function, and has to discover (in the limit) a program for the function. In this paper, we consider besides the graph also other modes of input such as the complement of the graph, the undergraph and the overgraph of the function. The relationships between these models are studied and a complete picture is obtained. Furthermore, these notions are also explored for learning with oracles, learning in teams and learning in the presence of additional information.

Keywords. Inductive inference; recursion theory; various forms of input presentation; team learning; learning with additional information.

Mathematical Subject Classification. 68Q32 (Computational Learning Theory); 03D20 (Recursive Functions).

1 Introduction to the field of inductive inference

The main purpose of the field known as inductive inference is to investigate, in clear mathematical terms, which classes of functions or classes of sets can be learnt, in an idealized sense, as a result of applying an algorithmic procedure that converts data, that is, pieces of information about some object taken from a class of possible objects, to hypotheses, that is, tentative descriptions of the object to be learnt. This can be formalized in different ways, on the basis of notions from recursion theory; therefore the objects to be learnt are — in most scenarios — either classes

* Supported in part by NUS grant numbers R252-000-212-112 and R252-000-308-112.

** E. Martin is jointly appointed at the UNSW and National ICT Australia which is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program.

of recursive functions or classes of recursively enumerable sets. The first formalization has been proposed by Gold [9], who introduced the notion of *learning sets in the limit from positive data*: the learner (a recursive function) maps any finite string of data (that is, finite sequence of members of an underlying set, that might be infinite) to a hypothesis (that is, a tentative exact and complete description of the whole set). The set is assumed to be recursively enumerable and the hypotheses are natural numbers, interpreted as indices with respect to some fixed acceptable numbering W_0, W_1, ... of all recursively enumerable sets. (This numbering corresponds to an acceptable numbering φ_0, φ_1, ... of the partial recursive functions, with W_e being the domain of φ_e for all e's.) More precisely, a set of the form W_e is selected, with e being unknown to the learner, and an enumeration of its members, in some arbitrary order, possibly with repetitions, possibly with the inclusion of a pause symbol, denoted $\#$, with no member of W_e missing from the enumeration, is unveiled to the learner. The enumeration, called a *text* for W_e, is unveiled through a process that takes infinitely many steps, with successive steps revealing successive members of the text, a new step revealing either $\#$ or a member of W_e. The learner M is said to learn W_e from text T for W_e iff there is an n such that $W_{M(T(0)T(1)\ldots T(m))} = W_e$ for all $m \geq n$. In other words, the learner can issue only finitely many incorrect descriptions of W_e. As one cannot check effectively whether $W_e = W_d$ for arbitrary indices d and e, it is restrictive to require that $M(T(0)T(1)\ldots T(n))$ be equal to $M(T(0)T(1)\ldots T(m))$ for all $m \geq n$. So there are (at least) two notions of convergence, that both turned out to be reasonable and fruitful in the formal developments of the field of inductive inference. The more general notion is called *behaviourally correct learning* [3] and the more restrictive notion is called *explanatory learning* [9]. This paper deals exclusively with explanatory learning, but many questions could also be investigated in the more general setting of behaviourally correct learning.

A learner that would be given the task of learning only one set W_e could of course output e in response to any finite string of data; therefore, only the learnability of classes of sets (by the same learner), as opposed to single sets, is of interest. Here are some well known examples.

- The class of finite sets is learnable: in order to be successful, a learner can just output at every stage a hypothesis which covers precisely the elements seen so far; the hypothesis is revised only if a new element appears.
- The class of all sets of the form $\mathbb{N} \setminus \{x\}$ (all natural numbers except x) is learnable: in order to be successful, a learner can just output at every stage a fixed index of $\mathbb{N} \setminus \{n\}$ where n is the least number not in the current input.
- The class of all cofinite sets, the class consisting of $\mathbb{N}$ and all finite subsets of $\mathbb{N}$, the class of all graphs of recursive functions, have all been shown by Gold [9] not to be learnable.

2 Learning total functions from different representations

Given a function f from $\mathbb{N}$ to $\mathbb{N}$, denote by E_f the graph of f:

$$E_f = \{(x, f(x)) : x \in \mathbb{N}\}.$$

Denote by REC the class of all recursive functions from $\mathbb{N}$ to $\mathbb{N}$. The examples at the end of the previous section suggest the definition of a notion of learnability of subsets of REC: say that a subset S of REC is learnable iff $\{E_f : f \in S\}$ is learnable in the sense previously defined. This paper is innovative in that E_f is conceived of as just one possible description of f amongst a number of other, equally legitimate, descriptions of f. What is at issue is how particular representation schemes affect the learnability of a class of functions.

A data presentation scheme I_f may be considered legitimate if it is given by a parametrized uniformly recursive family $U_{a,b}$ such that the following conditions hold:

- $U_{a,b} \subseteq \{a\} \times \mathbb{N}$;
- one can determine b in the limit from a and any text for $U_{a,b}$;
- $I_f = \bigcup_{a \in \mathbb{N}} U_{a,f(a)}$.

In this paper we consider the following types of data presentation schemes which satisfy the above constraints. So the set I_f takes one of the following four forms:

- graph of f, that is, $E_f = \{(x, y) : x, y \in \mathbb{N},\ y = f(x)\}$;
- complement of the graph of f, that is, $D_f = \{(x, y) : x, y \in \mathbb{N},\ y \neq f(x)\}$;
- undergraph of f, that is, $L_f = \{(x, y) : x, y \in \mathbb{N},\ y < f(x)\}$;
- overgraph of f, that is, $G_f = \{(x, y) : x, y \in \mathbb{N},\ y > f(x)\}$.

We now present the notions that determine a learning paradigm in a more formal way. A *learner* is a recursive function from the set of finite sequences of members of $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$. A learner's domain is viewed as a set that contains all finite initial segments of pairs taken from E_f, D_f, L_f or G_f where f is a member of the subset of REC to be learnt, depending on the kind of data selected by the paradigm. Each of the learner's output is viewed as an index for the function to be learnt. The way functions are coded by indices can vary from paradigm to paradigm. Unless specified otherwise, these indices are assumed to be from some implicit acceptable numbering of all the partial recursive functions. We use E, D, L and G to refer to the notions of learnability determined by E_f, D_f, L_f and G_f, respectively: these notions are defined as above, with texts for E_f being replaced by texts for E_f, D_f, L_f and G_f, respectively.

Definition 1. Given a recursive function f, an *D-text*, *E-text*, *L-text* or *G-text for f* is a text for D_f, E_f, L_f or G_f, respectively. Here a text for a set I_f is a, not necessarily recursive, infinite listing of I_f which can also possibly contain the pause symbol #.

The # symbol represents a pause in the presentation of the data. Note that E-, D- and G-texts could be defined without the # symbol; the # symbol is necessary to define L-texts for the function 0^∞.

Definition 2. Given a recursive function f and a learner M, M is said to *E-learn*, *D-learn*, *L-learn* or *G-learn* f iff for all E-texts, D-texts, L-texts or G-texts T for f, respectively, M outputs a fixed index for f in response to all but finitely many initial segments of T.

We use D, E, G and L to represent the set of all classes S of recursive functions such that some learner D-, E-, G- or L-learns all members of S, respectively. For consistency with other papers, it would have been perhaps better to call these criteria DEx, EEx, GEx and LEx, but we dropped the suffix Ex as it is redundant within the present work. The criterion E coincides with explanatory function learning (Ex or Lim) [9] as the course of values of a function f can be easily reconstructed from any E-text for f. Here are two examples that illustrate the notions that have been introduced.

Example 3. *Call a family $(f_i)_{i\in\mathbb{N}}$ of recursive functions* recursively enumerable *if the binary function $(n, x) \mapsto f_n(x)$ is recursive.*

- *Every recursively enumerable family $(f_i)_{i\in\mathbb{N}}$ of recursive functions is E-learnable and D-learnable: in order to be successful, a learner just has to output a fixed index of the function f_n where n is the least natural number such that no pair of the form (x, y) with $y \neq f(x)$ (for E-learnability) or $y = f(x)$ (for D-learnability) occurs in the data.*
- *The class $\{1^n0^\infty : n \in \mathbb{N}\} \cup \{1^\infty\}$ is not L-learnable but it is E-learnable, D-learnable and G-learnable. Gold* [9] *showed that the class $\{\{0, 1, 2, \ldots, n\} : n \in \mathbb{N}\} \cup \{\mathbb{N}\}$ is not learnable from text. This implies that $\{1^n0^\infty : n \in \mathbb{N}\} \cup \{1^\infty\}$ is not L-learnable. A G-learner for $\{1^n0^\infty : n \in \mathbb{N}\} \cup \{1^\infty\}$ conjectures 1^∞ until it has seen some pair $(n, 1)$; from then onwards it outputs 1^n0^∞ for the least n such that $(n, 1)$ has shown up in the input. Since it is recursively enumerable, the class $\{1^n0^\infty : n \in \mathbb{N}\} \cup \{1^\infty\}$ is also D-learnable and E-learnable.*
- *Similarly, the class $\{0^n1^\infty : n \in \mathbb{N}\}\cup\{0^\infty\}$ is not G-learnable but it is L-learnable, D-learnable and E-learnable.*
- *The recursively enumerable class $\{f : \exists x \forall y > x\,(f(y) = f(x))\}$ is neither L-learnable nor G-learnable but it is both D-learnable and E-learnable.*

Example 4. *Let S denote the class of* self-describing *functions, that is, the set of all recursive functions f such that $\varphi_{f(0)} = f$. Then S is E-, D-, L- and G-learnable, as one can discover $f(0)$ in the limit from any text for E_f, D_f, L_f and G_f, respectively.*

Recall the following standard notation:

- We denote by $\varphi_{e,s}$ the partial recursive function such that for all $x \in \mathbb{N}$, $\varphi_{e,s}(x)$ takes the value $\varphi_e(x)$ if $x < s$ and the computation of $\varphi_e(x)$ converges within s steps; otherwise $\varphi_{e,s}(x)$ is undefined.
- We denote by $W_{e,s}$ the domain of $\varphi_{e,s}$.

3 The Basic Relationships

The next theorem gives the complete relationship between the four basic notions E, D, L and G considered in this paper.

Theorem 5. *The inclusions $L \subseteq D$, $D \subseteq E$, $L \subseteq E$ and $G \subseteq E$ hold. No other inclusion between two distinct criteria out of E, D, L and G holds.*

Proof. The inclusion $L \subseteq E$ is obtained by transitivity from $L \subseteq D$ and $D \subseteq E$, thus only the other three inclusions are shown. Observe that for all recursive functions f, the following equalities hold:

$$\begin{aligned} L_f &= \{(x,y) : \forall z \leq y\,((x,z) \in D_f)\}; \\ D_f &= \{(x,y) : \exists z \neq y\,((x,z) \in E_f)\}; \\ G_f &= \{(x,y) : \exists z < y\,((x,z) \in E_f)\}. \end{aligned}$$

Given a recursive function f, these equalities allow one to effectively translate a D-text for f into an L-text for f, an E-text for f into a D-text for f and an E-text for f into a G-text for f. Hence a learner that L-, D- or G-learns a class S of recursive functions can be effectively transformed into a learner that D-, E- or E-learns S, respectively. Hence the inclusions $L \subseteq D$, $D \subseteq E$ and $G \subseteq E$ hold.

Example 3 shows the noninclusions $D \not\subseteq L$, $D \not\subseteq G$, $E \not\subseteq L$, $E \not\subseteq G$, $L \not\subseteq G$ and $G \not\subseteq L$. The noninclusion $G \not\subseteq D$ follows from Theorem 8 below. This noninclusion directly implies $E \not\subseteq D$. □

4 Oracles

Given a set A of natural numbers, the notion of learner is generalized to the notion of an A-recursive learner: the learner is permitted to use an oracle for A, that is, the learner can ask membership questions about A. Given an oracle A, the notions of E-, D-, L- and G-learnability are generalized to the notions of $E[A]$-, $D[A]$-, $L[A]$- and $G[A]$-learnability. In the following, A, A_1, A_2, A_3, A_4 and B will denote sets of natural numbers which are used as oracles and K will denote the set $\{x : \varphi_x(x)\downarrow\}$. A' denotes the set $\{x : \varphi_x^A(x)\downarrow\}$, where φ_i^A is the partial function computed by the i-th Turing machine using the oracle A. A set A is said to be *high* iff $K' \leq A'$ (see [16]). Adleman and Blum [1] established the following fundamental result, which characterizes the oracles A such that the class of recursive functions is $E[A]$-learnable.

Theorem 6 (Adleman and Blum [1]). *$REC \in E[A]$ iff A is high.*

This result also holds for D-learnability as we now show.

Theorem 7. *$REC \in D[A]$ iff A is high.*

Proof. If A is high then there is an A-recursive procedure $Tot(e,t)$ such that for every e, $\lim_{t\to\infty} Tot(e,t) = 1$ if φ_e is total, and $\lim_{t\to\infty} Tot(e,t) = 0$ if φ_e is partial. Consider an A-recursive learner M that proceeds as follows. Given a sequence σ of data of length s, output the least e such that $Tot(e, s) = 1$ and for all $x \leq s$, if $\varphi_{e,s}(x)$ is defined then $(x, \varphi_e(x))$ does not occur in σ. It is easily verified that M $D[A]$-learns REC.

The proof that $D \subseteq E$ given in Theorem 5 is easily generalized to a proof that $D[A] \subseteq E[A]$. This together with Theorem 6 shows that if $REC \in D[A]$ then A is high. $\Box$

Theorem 6 can be generalized neither to L- nor to G-learnability, as the separations in Example 3 also hold for learning with oracles.

We know from Theorem 5 that D-learnability is not as powerful as G-learnability. A natural question is whether this limitation of D-learnability can be overcome with the use of oracles: in other words, does there exist an oracle A such that $D[A]$-learnability is at least as powerful as G-learnability? The next result provides a positive answer and characterizes the oracles A that yield that positive answer.

Theorem 8. *$G \subseteq D[A]$ iff A is high.*

Proof. If A is high then the inclusion $G \subseteq D[A]$ follows immediately from Theorem 7.

Conversely, for all $e \in \mathbb{N}$, let f_e be the recursive function such that for all $x \in \mathbb{N}$, $f_e(x)$ is equal to the sum of e with the least natural number s such that $W_{e,s}$ contains at least $\min\{x, |W_e|\}$ elements (recall that $W_{e,s}$ is included in $\{0, 1, \ldots, s-1\}$); in particular, $f_e(0) = e$. Let $S = \{f_e : e \in \mathbb{N}\}$.

In order to G-learn S, a learner on a G-text for f can proceed as follows. Let y be the least number such that $(0, y+1)$ has already appeared in the input; if no such y exists then the learner outputs 0. Note that $y = f(0)$ in the limit. Let s denote the length of the input data. Given $x \leq s$, say that x is *good* if the data contains no pair of the form (x, z) such that $W_{y,z}$ contains less than x elements. If all $x \leq s$ are good then output an index for the partial recursive function that maps any $x \in \mathbb{N}$ to $s + e$ for the least s with $|W_{e,s}| \geq x$. Otherwise, let x be the least integer that is not good. Then output a fixed index for a function g with the following properties.

- For $x' \leq x$, let y' be the greatest number such that $(x', 0), (x', 1), \ldots, (x', y')$ do not appear in the G-text for f. Then $g(x')$ is equal to this y'.
- For all $x' > x$ let $g(x') = g(x)$.

It is immediately verified that a learner that proceeds as described G-learns S. Suppose M is an A-recursive learner which D-learns S. In order to complete the proof of the theorem, it suffices to show that A is high. Note that the set of all $(e, x, y) \in \mathbb{N}^3$ with $(x, y) \in D_{f_e}$ is r.e., an immediate consequence of the fact that for all $e, x, y \in \mathbb{N}$, $(x, y) \in D_{f_e}$ iff at least one of the following four conditions holds:

- $y < e$;
- $y \geq e$ and there exists $s < y - e$ such that $|W_{e,s}| \geq x$;
- $y \geq e$ and there exists $s < y - e$ such that $|W_{e,s}| = |W_{e,y-e}|$;
- $y \geq e$, $|W_{e,y-e}| < x$ and there exists $s \in \mathbb{N}$ with $|W_{e,y-e}| < |W_{e,s}|$.

Thus one can construct a text T_e for D_{f_e} effectively from e. Let $h(e)$ denote the index to which M^A converges on T_e. Clearly, for all natural numbers e, $\varphi_{h(e)}$ is total and W_e is finite iff there exist $x, s \in \mathbb{N}$ such that $e+s = \varphi_{h(e)}(x)$ and $|W_{e,s}| < x$. Thus, using oracle A, one can effectively determine in the limit whether W_e is finite and thus the set $\{e : |W_e| < \infty\}$ is A'-recursive. As the set $\{e : |W_e| < \infty\}$ is Σ_2^0-complete [17, Section IV.3] it holds that $K' \leq_T A'$, completing the verification that A is high. □

In the following let $A_1 \oplus A_2 \oplus \ldots \oplus A_n$ be the set $\{nx + y : y < n \wedge x \in A_{y+1}\}$. The operation $\oplus$ is called the join-operation. Similarly one can define the join of n functions $f_1, f_2, \ldots, f_n$ as $(f_1 \oplus f_2 \oplus \ldots \oplus f_n)(nx+y) = f_{y+1}(x)$ for $x \in \mathbb{N}$ and $y < n$. The join $S_1 \oplus S_2 \oplus \ldots \oplus S_n$ of n classes of functions is the class of all functions $f_1 \oplus f_2 \oplus \ldots \oplus f_n$ with $f_1 \in S_1, f_2 \in S_2, \ldots, f_n \in S_n$.

Theorem 9. *Given four recursively enumerable sets A_1, A_2, A_3 and A_4, there are four classes SE_{A_1}, SD_{A_2}, SL_{A_3} and SG_{A_4} of recursive functions such that*

$$\begin{aligned} SE_{A_1} \in E[B] &\text{ iff } A_1' \leq_T B'; \\ SD_{A_2} \in D[B] &\text{ iff } A_2' \leq_T B'; \\ SL_{A_3} \in L[B] &\text{ iff } A_3' \leq_T B'; \\ SG_{A_4} \in G[B] &\text{ iff } A_4' \leq_T B'. \end{aligned}$$

Furthermore, if A_1, A_2, A_3 and A_4 are such that

$$(*) \quad A_1 \leq_T A_2,\ A_2 \leq_T A_3,\ A_2 \leq_T A_4,\ K <_T A_1',\ A_1' <_T A_2',\ A_3' \not\leq_T A_4' \text{ and } A_4' \not\leq_T A_3'$$

then the class $S = SE_{A_1} \oplus SD_{A_2} \oplus SL_{A_3} \oplus SG_{A_4}$ satisfies for every oracle B that

$$\begin{aligned} S \in E[B] &\text{ iff } A_1' \leq_T B'; \\ S \in D[B] &\text{ iff } A_2' \leq_T B'; \\ S \in L[B] &\text{ iff } A_3' \leq_T B'; \\ S \in G[B] &\text{ iff } A_4' \leq_T B'. \end{aligned}$$

Finally, there are four r.e. sets A_1, A_2, A_3 and A_4 that satisfy $(*)$ above.

Proof. Kummer and Stephan [12] have exhibited a class TE_{A_1} of total functions from $\mathbb{N}$ into $\{0, 1\}$ which is $E[B]$-learnable iff $A_1' \leq_T B'$. Define SE_{A_1} as the set of all recursive functions g such that

$$\exists h \in TE_{A_1}\, \forall n \in \mathbb{N}\, [g(2n) = h(n) \wedge g(2n+1) = 1 - h(n)].$$

For all $n \in \mathbb{N}$, $g(2n) + g(2n+1) = 1$. Hence approximations of $g(2n)$ and $g(2n+1)$ from below or from above allow one to discover both values in the limit. This implies that $A_1' \leq_T B'$ is equivalent to any of $SE_{A_1} \in E[B]$, $SE_{A_1} \in D[B]$, $SE_{A_1} \in L[B]$ and $SE_{A_1} \in G[B]$.

Since A_2' is recursively enumerable relative to K, there is a total binary function $\Psi(x, s)$ that approximates A_2', in the sense that for all $x \in \mathbb{N}$, $x \in A_2'$ iff $\Psi(x, s) = 1$ for all but finitely many

$s \in \mathbb{N}$. For all $x \in \mathbb{N}$, let f_x be defined as

$$f_x(y) = \begin{cases} x & \text{if } y = 0; \\ s & \text{for the first } s > f_x(y-1) \text{ with } \Psi(x,s) \neq 1 \\ & \text{if } y > 0 \text{ and such an } s \text{ exists;} \\ f_x(y-1) & \text{otherwise.} \end{cases}$$

Note that the case-distinction is nonuniform in x. However, f_x is recursive as either the second case applies for all $y > 0$ or f_x is eventually constant. Let $SD_{A_2} = \{f_0, f_1, \ldots\}$. Although given x, one cannot effectively compute an index for f_x, one can compute an index $u(x)$ of the following variant of f_x:

$$\varphi_{u(x)}(y) = \begin{cases} x & \text{if } y = 0; \\ s & \text{for the first } s > \varphi_{u(x)}(y-1) \text{ with } \Psi(x,s) \neq 1 \\ & \text{if } y > 0 \text{ and such an } s \text{ exists;} \\ \uparrow & \text{otherwise.} \end{cases}$$

For all $n \in \mathbb{N}$, let D_{f_x} and L_{f_x} be the two r.e. sets defined as follows. First define D_{f_x} as the set of all $(a,b) \in \mathbb{N}^2$ for which one of the following conditions holds:

- $\exists c \leq a \ [\varphi_{u(x)}(c)\downarrow > b]$ or
- $\exists c \geq a \ [\varphi_{u(x)}(c)\downarrow < b]$ or
- $b \neq x \wedge \Psi(x,b) = 1$.

Second define L_{f_x} as the set of all $(a,b) \in \mathbb{N}^2$ that satisfy the first condition in the definition in D_{f_x}, namely

$$\exists c \leq a \ [\varphi_{u(x)}(c)\downarrow > b].$$

Note that both D_{f_x} and L_{f_x} are uniformly recursively enumerable in x. Hence, if SD_{A_2} is $L[B]$-learnable or $D[B]$-learnable then $A_2' \leq_T B'$ as one can, for any given x, simulate the learner on L_{f_x} or D_{f_x}, respectively, find the final hypothesis e of the learner using B' and determine whether there is an y with $\varphi_e(y) = \varphi_e(y+1)$ using B' again; the latter has the answer YES if $x \in A_2'$ and the answer NO if $x \notin A_2'$.

The class SD_{A_2} is E-learnable and G-learnable, as the values $f_x(0)$ and x can be determined from the data in the limit. It can also be determined in the limit whether there exists $y \in \mathbb{N}$ such that $\varphi_{u(x)}(y)$ is defined with the property that $(y+1, \varphi_{u(x)}(y)) \in E_{f_x}$, whether there exists $y \in \mathbb{N}$ such that $\varphi_{u(x)}(y)$ is defined with the property that $(y+1, \varphi_{u(x)}(y)+1) \in G_{f_x}$, and whenever the first or second property holds, a value y with that property can be found in the limit. In the latter case, the learner knows that f_x agrees with $\varphi_{u(x)}$ on all natural numbers smaller than that y, and maps all natural numbers at least equal to y to a constant value. If no such y is found in the limit, then the learner will conjecture $\varphi_{u(x)}$ from the time it has determined x.

Using a similar argument, the class SD_{A_2} is seen to be both $L[B]$-learnable and $D[B]$-learnable for all oracles B with $A_2' \leq_T B'$. Again, the learner can obtain from the data in the limit the values $f_x(0)$ and x. Furthermore, it can use the oracle B to determine in the limit whether

$x \in A'_2$. If so, it finds in the limit the maximal y such that $\varphi_{u(x)}(y)$ is defined and knows that f_x equals the function which is $\varphi_{u(x)}$ below y and is constant from y onwards. If $x \notin A'_2$ then $f_x = \varphi_{u(x)}$ and the learner converges to the hypothesis $u(x)$.

For the sets SL_{A_3} and SG_{A_4}, one considers functions f^{A_3} and f^{A_4} which are ${A_3}'$-recursive and ${A_4}'$-recursive, respectively, such that one can solve A'_3 using any function dominating f^{A_3} as an oracle and one can solve A'_4 using any function dominating f^{A_4} as an oracle. Let

$$SL_{A_3} = \{(2n+1)^\infty : n \in \mathbb{N}\} \cup \{(2n+1)^m (2n)^\infty : n \in \mathbb{N}, m \leq f^{A_3}(n)\};$$
$$SG_{A_4} = \{(2n)^\infty : n \in \mathbb{N}\} \cup \{(2n)^m (2n+1)^\infty : n \in \mathbb{N}, m \leq f^{A_4}(n)\}.$$

The class SL_{A_3} is in $L[B]$ iff ${A_3}' \leq_T B'$. Consider the case of learning a function $g \in SL_{A_3}$ from an L-text for g. If ${A_3}' \leq_T B'$ then one can find $g(0)$ and the parameter n in the limit; one can then find in the limit an upper bound k for $f^{A_3}(n)$ relative to B; one can then determine $g(0)$, $g(1)$, $\ldots$, $g(k)$ in the limit, at which point the learner can output a fixed index for the function represented as $g(0)g(1)\ldots g(k)(g(k))^\infty$. For the converse, recall that a stabilizing sequence [4, 8] for M on L_f is a finite sequence σ such that σ contains only elements from L_f, and for all τ such that τ contains elements only from L_f, $M(\sigma) = M(\sigma\tau)$. If a learner M infers SL_{A_3} with oracle B then for each function f of the form $(2n+1)^\infty$, one can in the limit discover a stabilizing sequence for M on L_f as well as the largest first coordinate $h(n)$ of the pairs that occur in this stabilizing sequence; then $h \leq_T B'$ and $h(n) \geq f^{A_3}(n)$ for all n; hence ${A_3}' \leq_T B'$.

The argumentation for $SG_{A_4} \in G[B]$ iff $A'_4 \leq_T B'$ is symmetric. Furthermore, one can easily see that SL_{A_3} belongs to G, D and E, and that SG_{A_4} belongs to L, D and E.

For the second part of the theorem, recall that the class S defined as $SE_{A_1} \oplus SD_{A_2} \oplus SL_{A_3} \oplus SG_{A_4}$ consists of all functions g for which there are $g_1 \in SE_{A_1}$, $g_2 \in SD_{A_2}$, $g_3 \in SL_{A_3}$ and $g_4 \in SG_{A_4}$ such that for all $x \in \mathbb{N}$ and for all $y \in \{1, 2, 3, 4\}$, $g(4x + y - 1) = g_y(x)$. The hardness result is an immediate consequence of the hardness results about the 4 components. Consider a function $g \in S$ and an oracle B. If $A'_1 \leq_T B'$ and a text for E_g is given, then one can learn the component g_1 using the oracle B and the components g_2, g_3, g_4 without any oracle. If $A'_1 \leq_T B'$, $A'_2 \leq_T B'$ and a text for D_g is given, then the components g_1 and g_2 can be learnt using the oracle B and the components g_3 and g_4 do not require an oracle. If $A'_1 \leq_T B'$, $A'_2 \leq_T B'$, $A'_3 \leq_T B'$ and a text for L_g is given, then g_1, g_2, g_3 can be learnt using the oracle B and learning the component g_4 does not need an oracle. If $A'_1 \leq_T B'$, $A'_2 \leq_T B'$, $A'_4 \leq_T B'$ and a text for G_g is given, then the components g_1, g_2 and g_4 can be learnt using the oracle B and the component g_3 does not need an oracle. Since $A'_1 \leq_T A'_2 \leq_T A'_3$ and $A'_1 \leq_T A'_2 \leq_T A'_4$, one can conclude that S satisfies the claim of the theorem. □

5 Degrees of Inference

The question for which oracles A and B the inclusion $E[A] \subseteq E[B]$ holds has been investigated exhaustively [1, 6, 12]. Adleman and Blum [1] established that exactly the high oracles B satisfy that $E[A] \subseteq E[B]$ for all oracles A. Kummer and Stephan [12] showed that for non-high recursively enumerable sets A and B, $E[A] \subseteq E[B]$ iff $A \leq_T B$. Besides the criterion E — which corresponds to explanatory function learning — many other convergence criteria have been studied [6, 12]. The approach taken in the present work differs from these by the fact that the mode of convergence (explanatory or learning in the limit) does not vary, but the mode of presentation of the input varies. The criterion E coincides with explanatory learning (and hence the results coincide with that of explanatory learning), but for the criteria L and G some results differ from standard explanatory learning.

Remark 10. Assume that $E[A] \not\subseteq E[B]$ for some oracles A and B. Then there is a class R of $\{0,1\}$-valued functions witnessing this [12]. Now one defines

$$S = \{g : \exists h \in R \; \forall x \, (g(2x) = h(x) \wedge g(2x+1) = 1 - h(x))\}.$$

The class S satisfies for every oracle C that

$$R \in E[C] \Leftrightarrow S \in E[C] \Leftrightarrow S \in L[C] \Leftrightarrow S \in G[C] \Leftrightarrow S \in D[C].$$

As a consequence, one has that $L[A] \not\subseteq L[B]$, $G[A] \not\subseteq G[B]$ and $D[A] \not\subseteq D[B]$.

The inference-degrees of D and E might be quite similar, as previous results already establish that the high oracles are also exactly those which are omniscient for D. It can even be conjectured that they are the same. For L and G, things are a bit different.

Remark 11. Jain and Sharma [10] proved that there is no maximal inference-degree for learning recursively enumerable sets from positive data. Kummer and Stephan [12] strengthened this result by showing the following: if every A-recursively learnable class of languages is also B-recursively learnable then $A' \leq_T B'$. To show this, they constructed a class $S_{A,B}$ consisting of recursive sets. The class of characteristic functions of these sets witnesses the following corresponding result: if $L[A] \subseteq L[B]$ then $A' \leq_T B'$; similarly, if $G[A] \subseteq G[B]$ then $A' \leq_T B'$. In particular, for each oracle there is still a strictly more powerful oracle and so the structure of inference-degrees for L and G differs from that of E and D because for the latter two models, all high oracles form the maximal inference degree.

For many criteria the trivial inference degree consisting of all oracles which do not help to learn anything unlearnable has been determined. In particular for explanatory learning of functions and explanatory learning of sets from positive data, this trivial inference degree coincides with all sets $A \leq_T K$ such that either A is recursive or has 1-generic Turing degree. These results can be generalized to E-learning, D-learning, L-learning and G-learning.

Proposition 12. *For an oracle A, the following five conditions are equivalent:*

1. *$E[A] = E$, that is, A is trivial for E-learning;*
2. *$D[A] = D$, that is, A is trivial for D-learning;*

3. $L[A] = L$, that is, A is trivial for L-learning;
4. $G[A] = G$, that is, A is trivial for G-learning;
5. $A \leq_T K$ and the Turing degree of A is either recursive or 1-generic.

Conjecture 13. *The inference-degrees of E and D coincide.*

Conjecture 14. *The inference-degrees of L and G both coincide with the inference-degrees of learning recursively enumerable sets from positive data.*

6 Additional Information and Teams

An oracle supplies information independent of the concrete learning task, only nonrecursive oracles are relevant. In contrast to this, models of additional information are usually linked to the concrete learning task, although they might not provide any nonrecursive information. Indeed additional information is even finite, mostly an index or a bound on an index for some aspect of the learning task. Freivalds and Wiehagen [7] showed that one can learn the class of all recursive functions if the additional information supplied is an upper bound on an index of the function to be learnt. This result holds independently of the chosen data type.

Proposition 15. *REC is D-learnable, L-learnable and G-learnable if one is given an upper bound on an index of the function to be learnt as additional information.*

Proof. The proof uses the following notation: given finite set Z, let e_Z be an index such that $\varphi_{e_Z}(x) = \varphi_z(x)$ for the first $z \in Z$ (found in some algorithmic search) such that $\varphi_z(x)$ converges.

Let f and a bound b on an index for f be given. The learner converges in the limit to e_Z for a set Z defined as follows. Call an index $i \leq b$ *bad* if there is an x in the domain of φ_i such that the corresponding following condition holds:

- in the case of D-learning: $(x, \varphi_i(x))$ is present in the input D-text;
- in the case of L-learning: $(x, \varphi_i(x))$ is present in the input L-text for f or $\exists j \leq b\,(x$ is the least number with $\varphi_i(x) > \varphi_j(x)\!\downarrow$ and $(x, \varphi_j(x))$ is not present in the input L-text for $f)$;
- in the case of G-learning: $(x, \varphi_i(x))$ is present in the input G-text for f or $\exists j \leq b\,(x$ is the least number with $\varphi_i(x) < \varphi_j(x)\!\downarrow$ and $(x, \varphi_j(x))$ is not present in the input G-text for $f)$.

It is easy to verify that the learner can determine the set of bad indices in the limit. We let Z be the set of indices $i \leq b$ which are not bad.

For the verification, let $j \leq b$ be an index for f. Then j is not bad. Furthermore, if $i \leq b$ and φ_i is inconsistent with f then i is bad, as witnessed using j in the above conditions with x being the minimal value such that $\varphi_i(x)\!\downarrow \neq \varphi_j(x)\!\downarrow$. Thus Z is the set of indices below b which are consistent with the input function and e_Z is an index of the input function. □

As the additional information about an upper bound on the index of the function being learnt gives no new result, the following more general setting is investigated.

Given $I, J, H \in \{E, D, L, G\}$, a class S is $IJ(H)$-learnable iff there is a learner M learning every function $f \in S$ in the following setting.

- M receives text for I_f as input, as in the concept of I-learning in previous sections;
- M outputs hypotheses of type J, that is, the output sequence of M, on initial segments of the text of I_f, converges to an r.e. index of the recursively enumerable set J_f;
- The additional information provided is an r.e. index of a recursively enumerable set V such that $V = H_f$.

Additionally, we also consider the criteria of IJ-learning, where the first two conditions above hold, but no additional information is supplied.

The reader should note that in the definition above, r.e. indices instead of characteristic indices of E_f, D_f, L_f and G_f are considered. Note that r.e. indices for E_f and indices of f as a recursive function can be translated into each other, hence the criteria IE and I coincide for $I \in \{D, E, L, G\}$. However, r.e. indices of D_f, L_f, G_f cannot be translated into characteristic indices for these sets.

Clearly, $REC \in IJ(J)$ as the additional information already solves the learning task. Similarly, the inclusions from Section 3 carry over to corresponding values of I (but fixed J, H). Furthermore, REC belongs to both $IL(D)$ and $IJ(E)$ for $I, J \in \{D, L, G, E\}$: an r.e. index for D_f given as additional information can be converted to an r.e. index for L_f, and r.e. index for E_f given as additional information can be converted to an r.e. index for J_f; so one can simply translate the additional information into the required form. In some cases, data and additional information can complement each other: for example, $EJ \subseteq LJ(G)$ as the additional information G plus the data L allow one to generate the data E.

Example 16. *There is a uniformly r.e. family $D_{f_0}, D_{f_1}, D_{f_2}, \ldots$ of sets such that for each x, the function f_x is recursive, $f_x(0) = x$ and the x-th learner M_x does not learn f_x from E_{f_x}. Then the class $\{f_0, f_1, f_2, \ldots\}$ is in ID for $I \in \{E, D, L, G\}$, because one can find x from the input text in the limit, and then an r.e. index for D_{f_x} from x. However, the above class is not in $EE(D)$ as it is not in EE and the additional information for D is redundant, since it can be found in the limit from the input data.*

Proof. The idea is to define each D_{f_x}, diagonalizing against the x-th machine M_x as follows. The enumeration keeps an approximation g_s of f_x from below and an increasing diagonalization marker m_s. The initialization is $m_0 = 1$ and $g_0 = (x)^\infty$. For $s \leq x$, the approximation remains unchanged and for $s > x$, the following update rule is applied:

> if there is a y with $m_s < y < s$ such that M_x on $(0, g_s(0))\,(1, g_s(1))\,\ldots\,(y, g_s(y))$ outputs an index e (within s steps) with $\varphi_{e,s}(y+1)\downarrow\, < s$, then let $g_{s+1}(t) = s + 1$ for $t > y$, $g_{s+1}(t) = g_s(t)$ for $t \leq y$ and $m_{s+1} = y + 1$; otherwise let $g_{s+1} = g_s$ and $m_{s+1} = m_s$.

The resulting function is either almost everywhere constant or it receives for each y the final value whenever m_s becomes larger than y. Therefore f_x is recursive. Furthermore, the set D_{f_x} is easily seen to be r.e. as $(y,z) \in D_{f_x}$ if $g_s(y) \neq z$ for some $s > z$. Furthermore, M_x does not learn f_x from E_{f_x} by the explicit diagonalization above. □

The next theorem characterizes that a criterion J is more powerful than a criterion I iff REC can be $HI(J)$-learnt for $H \in \{E, D, L, G\}$ iff one can effectively convert r.e. indices for J_f to r.e. indices for I_f, uniformly in all f.

Theorem 17. *For all members I and J of $\{E, D, L, G\}$, the following statements are equivalent.*

1. *One can effectively translate every r.e. index of J_f into an r.e. index of I_f (uniformly in all f);*
2. *$I \subseteq J$, that is, without any additional information and using hypotheses of type E, every class learnable from data of type I is also learnable from data of type J;*
3. *$EE(I) \subseteq EE(J)$, that is, for every class S, if S can be learnt with data E, with hypothesis space E and with additional information I then S can also be learnt with data E, hypothesis space E and additional information J;*
4. *$REC \in EI(J)$;*
5. *$REC \in DI(J)$;*
6. *$REC \in LI(J)$;*
7. *$REC \in GI(J)$.*

Proof. The claim of the theorem is trivial when $I = J$, so suppose otherwise. Let (I, J) be one of (L, D), (D, E), (L, E) and (G, E). Then $I \subseteq J$ by Theorem 5 and it is easy to verify that every r.e. index for J_f can be translated into an r.e. index for I_f. Thus, 1–7 hold.

Note that item 4 implies item 3 because if $REC \in EI(J)$ then one can convert J-indices using E-text into I-indices in the limit and thus every $EE(I)$-learnable class can be $EE(J)$-learnt by converting the index of the given J-additional information to an index for I-additional information. Similarly, item 1 implies item 3. Furthermore, items 5-7 clearly imply item 4.

So assume now that $(I, J) \notin \{(E,E),(D,D),(G,G),(L,L),(L,D),(D,E),(L,E),(G,E)\}$. By the discussion in the previous paragraph, it suffices to prove that items 2 and 3 do not hold.

Theorem 5 implies that $I \not\subseteq J$ and hence item 2 does not hold.

Item 3 is now disproven by considering each relevant case.

$EE(G) \not\subseteq EE(L)$ and $EE(L) \not\subseteq EE(G)$: these two non-inclusions follow from the observation of Case and Lynes [5]. Their result [5, Theorem 1] implies the existence of a class $S_1 \in EL - EE$ of recursive $\{0,1\}$-valued functions (one can take S_1 to be the set of characteristic functions of the languages in the class of languages which witnesses that one can synthesize grammars in the limit, but cannot synthesize decision procedures in the limit, from E-texts for characteristic functions of languages as input). By symmetry, there is also a class $S_0 \in EG - EE$ of recursive $\{0,1\}$-valued functions (by letting $S_0 = \{g_f : f \in S_1\}$, where $g_f(x) = 1 - f(x)$).

As $S_1 \in EL$, it follows that $S_1 \in EE(G)$ because one can use the additional information for G_f for translating an r.e. index for L_f into an r.e. index for E_f. On the other hand, $S_1 \notin EE(L)$ as an r.e. index for L_f can be found in the limit from input E_f (since $S_1 \in EL$) and is therefore useless as additional information for learning S_1. Hence $S_1 \in EE(G) - EE(L)$. Similarly $S_0 \in EE(L) - EE(G)$. These examples also show that $EE(D) \not\subseteq EE(L)$ and $EE(D) \not\subseteq EE(G)$ as for $\{0,1\}$-valued functions, $EE(D)$ and $EE(E)$ coincide.

Example 16 gives a class which is in ED but not in $EE(D)$. This class is in $EE(G)$ too, as an r.e. index for G_f along with an r.e. index for D_f can be used to generate an r.e. index for E_f:

$$E_f = \{(x,y) : \exists z > y\,[(x,z) \in G_f \wedge \forall u \in \{0,1,2,\ldots,z\} - \{y\}\,((x,u) \in D_f)]\}.$$

Hence this example witnesses $EE(G) \not\subseteq EE(D)$ and $EE(E) \not\subseteq EE(D)$. □

A further important topic is the question of team-learning where a class S is $[m,n]IJ$-learnable iff there is a team of n learners such that on every text for a set I_f for a function f in S at least m members of the team converge to an r.e. index for the language J_f. Here are some sample results which could be extended with well-known techniques.

Proposition 18. $[2,3]IE = [1,1]IE$ *for all* $I \in \{E,D,L,G\}$. *That is, for all modes of input presentation and as long as the hypotheses have to be of type E, any* $[2,3]IE$*-learner can be replaced by a* $[1,1]IE$*-learner which is the same as an* I*-learner.*

Proof. Let M_1, M_2, M_3 be a given $[2,3]IE$-team learning S. Without loss of generality, the size of the hypotheses of the machines increases at every mind change. Then a new learner N simulates M_1, M_2, M_3 and considers at every stage s the current outputs of the learning machines; in order of their size they are denoted as e_1, e_2, e_3. In case $\varphi_{e_1}(x) = \varphi_{e_2}(x)$ for all $x \le s$ where these two values are output within s steps of computation, the output of N does not depend on e_3 and is a value $f(e_1,e_2)$ such that $\varphi_{f(e_1,e_2)}(x)$ takes $\varphi_{e_1}(x)$ or $\varphi_{e_2}(x)$, whichever converges first. Otherwise the learner outputs $g(e_1,e_2,e_3)$ such that $\varphi_{g(e_1,e_2,e_3)}(x)$ is y iff at least two of the values $\varphi_{e_1}(x)$, $\varphi_{e_2}(x)$ and $\varphi_{e_3}(x)$ are y. For the verification, note that at least two machines converge and their final hypotheses e_1 and e_2 are eventually below the hypothesis of the third machine. If φ_{e_1} and φ_{e_2} are consistent then $\varphi_{f(e_1,e_2)}$ equals f as one of these two functions must be f. If they are inconsistent, the third machine must converge as well to some final value e_3 and $\varphi_{g(e_1,e_2,e_3)}$ is equal to f. □

This proof technique can be generalized to obtain the following result which was for EE already obtained by Pitt and Smith [15].

Theorem 19. *For* $I \in \{E,D,L,G\}$, $[m,n]IE = [1,k]IE$ *iff* $\frac{1}{k+1} < \frac{m}{n} \le \frac{1}{k}$.

In the case of ED-learning where the data is of type E and the hypotheses are of type D, one can assume that every inconsistent hypothesis, that is, every hypothesis e for a set D_e with $(x,f(x)) \in D_e$, is eventually updated and replaced by another one. Then $[2,3]ED = [1,1]ED$ by simply taking the union of the two oldest hypotheses in each stage. Again one can generalize the result.

Proposition 20. *$[m, n]ED = [1, k]ED$ iff $\frac{1}{k+1} < \frac{m}{n} \leq \frac{1}{k}$.*

The results can be very different if neither the data nor the hypotheses are of type E; then they have some similarity to the case of language learning.

Remark 21. Translating results on language learning obtained by Jain and Sharma [11] yields $[1, 1]GG \subset [2, 3]GG \subset [1, 2]GG = [3, 6]GG \subset [2, 4]GG$ and $[1, 1]LL \subset [2, 3]LL \subset [1, 2]LL = [3, 6]LL \subset [2, 4]LL$.

7 Conclusion

One of the key distinctions in the criteria that determine a learning paradigm is between "positive data only" and "both positive and negative data", or more generally, between implicit and explicit information about the structure to identify. That distinction takes a simple form in case the structure to identify is a set of objects taken from some universe, with the potential properties of the set being expressed as: "is this object in the universe a member of the set?": a positive datum is then a member of the set and a negative datum a nonmember of the set. Moreover, the situation is symmetric. Not only do positive data fully determine negative data, which allows them to leave them implicit; negative data also fully determine positive data, and if the negative data were all provided to the learner then the positive data could all be left implicit.

Although the graph E_f of a function f is a set, the distinction between positive and full information does not carry over to this special type of sets as one can derive that $(x, y) \notin E_f$ after discovering that there is some $z \neq y$ with $(x, z) \in E_f$. Hence the idea behind the current work is to take an alternative approach and look at the sets D_f, L_f, G_f instead of E_f and to present them by text, that is, in the form of positive data. It was shown that these four types of information are all different and that the only inclusions to hold between two distinct criteria are $D \subseteq E$, $L \subseteq E$, $G \subseteq E$ and $L \subseteq D$, that is, every class of functions learnable from D-text is also learnable from E-text and so on. Then it was investigated to which extent oracles can support learning and bridge the gaps between various types of data presentation. This turned out to be possible for E versus D as the class REC of all recursive functions is learnable from D-texts using suitable oracles, but oracles cannot bridge the gaps found between learning from L-texts and G-texts.

For inference degrees, there were parallels between the inference degrees of learning language-classes from text and learning functions from L-text or G-text, respectively. It is an open problem whether these three types of inference degrees are the same. Similarly, it is open whether the inference degrees of learning from D-texts are the same as those for learning from E-texts.

Besides attacking these two open problems, further work could also study the impact of additional information for learning from these novel types of data presentations in the future. Furthermore, team hierarchies could be investigated in more detail.

Acknowledgments. We thank the anonymous referees for several useful comments.

References

1. Lenny Adleman and Manuel Blum. Inductive inference and unsolvability. *Journal of Symbolic Logic*, 56:891–900, 1991.
2. Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
3. Janis Bārzdiņš. Two Theorems on the Limiting Synthesis of Functions. *Theory of Algorithm and Programs* I:82–88. Latvian State University, 1974.
4. Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.
5. John Case and Christopher Lynes. Inductive inference and language identification. *Proceedings of the of the Ninth International Colloquium on Automata, Languages and Programming* — ICALP, edited by Mogens Nielsen and Erik Meineche Schmidt, LNCS 140, 107–115, Springer-Verlag, Heidelberg, 1982.
6. 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.
7. Rūsins Freivalds and Rolf Wiehagen. Inductive inference with additional information. *Electronische Informationverarbeitung und Kybernetik*, 15:179–195, 1979.
8. Mark Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.
9. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
10. Sanjay Jain and Arun Sharma. On the non-existence of maximal inference degrees for language identification. *Information Processing Letters*, 47:81–88, 1993.
11. Sanjay Jain and Arun Sharma. Computational limits on team identification of languages. *Information and Computation*, 130(1):19–60, 1996.
12. 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.
13. Piergiorgio Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
14. Daniel Osherson, Michael Stob and Scott Weinstein. *Systems That Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists.* Bradford — The MIT Press, Cambridge, Massachusetts, 1986.
15. Leonard Pitt and Carl H. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.
16. Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.
17. Robert Soare. *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets.* Springer-Verlag, Heidelberg, 1987.