8-1997

Learning Recursive Functions From Approximations

John Case
University of Delaware, case@cis.udel.edu

Susanne Kaufmann
Universitat Karlsruhe

Martin Kummer
Universitat Karlsruhe

Efim Kinber
Sacred Heart University, kinbere@sacredheart.edu

Recommended Citation

Case, J. et al. "Learning Recursive Functions From Approximations." Journal of Computer and System Sciences 55.1 (1997): 183-196.

Learning Recursive Functions From Approximations*

John Case[†]
University of Delaware

Susanne Kaufmann[‡]
Universität Karlsruhe

Efim Kinber[§]
Sacred Heart University

Martin Kummer[¶]
Universität Karlsruhe

Abstract

Investigated is algorithmic learning, in the limit, of correct programs for recursive functions f from both input/output examples of f and several interesting varieties of *approximate* additional (algorithmic) information about f. Specifically considered, as such approximate additional information about f, are Rose's *frequency computations* for f and several natural generalizations from the literature, each generalization involving programs for restricted trees of recursive functions which have f as a branch. Considered as the types of trees are those with bounded variation, bounded width, and bounded rank.

For the case of learning final correct programs for recursive functions, EX-learning, where the additional information involves frequency computations, an insightful and interestingly complex combinatorial characterization of learning power is presented as a function of the frequency parameters. For EX-learning (as well as for BC-learning, where a final *sequence* of correct programs is learned), for the cases of providing the types of additional information considered in this paper, the maximal probability is determined such that the entire class of recursive functions is learnable with that probability.

*An extended abstract of this paper appeared in: *Computational Learning Theory–Second European Conference, EuroCOLT'95. Editor: Paul Vitányi.* Lecture Notes in Artificial Intelligence 904, pp. 140–153, Springer-Verlag, Berlin, 1995.

†Department of Computer and Information Sciences, University of Delaware, Newark, Delaware 19176, USA, (Email: `case@cis.udel.edu`).

‡Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, D-76128 Karlsruhe, Germany, (Email: `kaufmann@ira.uka.de`). Supported by the Deutsche Forschungsgemeinschaft (DFG) Grant Me 672/4-2.

§Computer Science, Sacred Heart University, 5151 Park Avenue, Fairfield, CT 06432-1000, USA, (Email: `kinber@shu.sacredheart.edu`).

¶Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, D-76128 Karlsruhe, Germany, (Email: `kummer@ira.uka.de`).

1 Introduction

In the traditional setting of inductive inference the learner receives input/output examples of an unknown recursive function f and has to learn a program for f. In real life a learner usually has "additional information" available. There are several approaches in the literature to incorporate this fact into the learning model, for instance by providing an upper bound for the size of the minimal program which computes f (Freivalds, Wiehagen [16]), by providing a higher-order program for f (Baliga, Case [3]), by allowing access to an oracle (Fortnow et al. [14]), by answering questions about f formulated by the learner in some first-order language (Gasarch, Smith [18]), by presenting "training sequences" (Angluin et al. [2]).

In this paper we follow a different route, we provide additional information in form of algorithms that *approximate* f. In the context of robot planning, Drew McDermott [34] says, "Learning makes the most sense when it is thought of as filling in the details in an algorithm that is already nearly right." As will be seen, the particular approximations we consider can be thought of as algorithms that are nearly right except for needing details to be filled in. The notions of approximation which we consider are also of interest in complexity theory [6] and recursion theory [4].

A classical approximation notion is (m, n)-*computation* (also called *frequency computation*) introduced by Rose [39] and first studied by Trakhtenbrot [42]. Here the approximating algorithm computes, for any n pairwise different inputs $x_1, \dots, x_n$, a vector $(y_1, \dots, y_n)$ such that at least m of the y_i are correct, i.e., are such that $y_i = f(x_i)$.

EX-style learning [9] requires of each function in a class learned that, in the limit, a single correct program be found. In Section 3 below we provide a combinatorial characterization of all m, n, m', n' such that every class which can be EX-learned from (m, n)-computations can also be EX-learned from (m', n')-computations. The combinatorial conditions for this characterization turn out to be interestingly complex. In this same section we also prove an interesting duality result comparing the learning of programs from (m, n)-computations with the learning of (m, n)-computations.

In Section 4 we determine the maximal probability $p > 0$ such that the class of all recursive functions is learnable with probability p from (m, n)-computations by a probabilistic inductive inference machine. We show that for $m \leq n/2$ there is no such probabilistic machine; whereas, for $m > n/2$, that $p = 1/(n-m+1)$ is the maximal p such that there is a probabilistic inductive inference machine which infers all recursive functions with probability p from (m, n)-computations. BC-style learning [9] requires of each function in a class learned that, in the limit, an infinite sequence of correct programs be found. Our results of this section hold for both EX- and BC-learning.

Providing an (m, n)-computation for f can be considered as a special case of providing a partial first-order specification of f (see the discussion at the beginning of Section 5 below). The idea is that the set of all solutions of a partial first-order specification can be pictured as the set of all branches of a recursive tree. Thus it is also natural to look at approximative information in the form of a recursive tree T such that f is a branch of T.

In this regard we consider several classes of recursive trees parameterized by natural numbers: trees of bounded variation, bounded width, or bounded rank. These classes are known from the literature, and they have the pleasing property that all the branches of their trees are recursive (see [21]). In Section 5 below, for each of these classes of approximate additional information, we determine the maximal probability p such that all recursive functions are learnable. In contrast to the special case of frequency computations, a higher maximal probability is obtained in many cases for BC than for EX.

2 Notation and Definitions

The recursion theoretic notation is standard and follows [35, 41].

$\omega = \{0, 1, 2, \ldots\}$. φ_i is the i-th partial recursive function in an acceptable enumeration, and $W_i \subseteq \omega$ is the i-th associated r.e. set (i.e., $W_i = dom(\varphi_i)$). Let REC denote the class of all total recursive functions, and let $REC_{0,1}$ be the class of all $\{0,1\}$-valued functions in REC.

For functions f and g let $f =^* g$ denote that f and g agree almost everywhere, i.e., $(\exists x_0)(\forall x \geq x_0)[f(x) = g(x)]$. $f \restriction y$ denotes the restriction of f to arguments $x < y$. χ_A is the characteristic function of $A \subseteq \omega$. We identify A with χ_A, e.g., we write $A(x)$ instead of $\chi_A(x)$.

ω^* is the set of finite sequences of natural numbers. λ is the empty string. $|\sigma|$ denotes the length of string σ. For instance, $|\lambda| = 0$. For strings σ and τ we write $\sigma \preceq \tau$ if σ is an initial segment of τ. Let $\sigma(x) = b$ if $x < |\sigma|$ and b is the $(x+1)$-th symbol of σ. For $\sigma, \tau \in \omega^n$, $\sigma =^e \tau$ means that σ and τ disagree in at most e components. The concatenation of σ and τ is denoted by $\sigma \star \tau$. We often identify strings with their coding number, e.g., we may regard W_i as the i-th r.e. set of strings.

A *tree* T is a subset of ω^* which is closed under initial segments. $\sigma \in T$ is called a *node* of T. T is r.e. if $W_i = \{\sigma : \sigma \in T\}$ for some i. Such an i is called a Σ_1-*index* of T. T is recursive if χ_T is a recursive function, in which case i is called a Δ_0-*index* of T if $\varphi_i = \chi_T$. $f \in \{0,1\}^\omega$ is a *branch*[1] of T if every finite initial segment of f is a node of T. We also say that $A \subseteq \omega$ is a branch of T if χ_A is a branch of T. $[T]$ is the set of all branches of T. Let $T[\sigma] = \{\tau \in T : \sigma \preceq \tau\}$, the subtree of T below σ.

An *inductive inference machine* (IIM) M is a recursive function from ω^* to ω. M EX-*infers* $f \in REC$ if $\lim_n M(f \restriction n)$ exists and is a φ-index of f. For $S \subseteq REC$, $S \in EX$ if there is an IIM which EX-infers all $f \in S$.

For $a \in \omega$, M BC-*infers* f if there is an n_0 such that for all $n \geq n_0$, $\varphi_{M(f\restriction n)} = f$. For $S \subseteq REC$, $S \in BC$ if there is an IIM which BC-infers all $f \in S$. See [9, 36] for background on these definitions.

In this paper we consider IIMs which receive additional information on f coded into a natural number. In this case an IIM is a recursive function from $\omega \times \omega^*$ to ω. M EX-*infers* $f \in REC$ *from additional information* $e \in \omega$, if $\lim_n M(e, f \restriction n)$ exists and is an index of f; similarly for BC-inference.

As is well-known, every IIM M can be replaced by a primitive recursive (or even polynomially time bounded) machine M' which infers the same set of functions (see

[1]We could consider branches $f \in \omega^\omega$, but, as we shall see in Section 5 below, for this paper, that will not be necessary.

[36]). M' just performs a slow simulation of M. Let $\{M_e\}_{e\in\omega}$ be an effective listing of all primitive recursive IIMs.

3 The Power of Learning from Frequency Computations

In this section we determine the relative power of inductive inference from frequency computations. We give a combinatorial characterization of the parameters m, n, m', n' such that every class which can be learned from (m,n)-computations can also be learned from (m',n')-computations. Our criterion was previously considered for the inclusion problem of frequency computation [13, 23, 28] where it is sufficient but not necessary, and for the inclusion problem of parallel learning where it is necessary but not sufficient [27].

Let us first recall the formal definition of (m,n)-computation which was introduced by Rose [39] and first studied by Trakhtenbrot [42].

Definition 3.1 Let $0 \le m \le n$. A function $f : \omega \to \omega$ is (m,n)-*computable* iff there is a recursive function $F : \omega^n \to \omega^n$ such that for all $x_1 < \cdots < x_n$,

$$(f(x_1), \ldots, f(x_n)) =^{n-m} F(x_1, \ldots, x_n),$$

i.e., F has at least m correct components. In this context, we call F an "(m,n)-operator" and say that f is (m,n)-computable via F.

Trakhtenbrot [42] proved the classical result that, for $m > n/2$, all (m,n)-computable functions are recursive. He also proved that this is optimal, i.e., there exist nonrecursive $(n,2n)$-computable functions. See [19] for a recent survey of these and related results.

In our new learning theoretic notion, the learner receives input/output examples of f and an index of an (m,n)-operator for f. If $m > n/2$, then any two functions which are (m,n)-computable via the same (m,n)-operator differ in at most $2(n-m)$ places. However, the (m,n)-operator does not reveal too much information about f, even if $m = n-1$: Kinber [22] proved that there is no uniform procedure to compute from an index of an $(n-1,n)$-operator a program which computes, up to finitely many errors, a function which is (m,n)-computable via this operator. This was recently generalized in [21].

Definition 3.2 Let $0 \le m \le n$. A class $S \subseteq REC$ belongs to $(m,n)EX$ iff there is an inductive inference machine M such that for every $f \in S$ and every index e of an (m,n)-operator for f, $\lim_t M(e, f \upharpoonright t)$ exists and is an index of f. Similarly, $(m,n)BC$ is defined.

Remark: Note that $(0,n)EX = EX$. Thus the new notion $(m,n)EX$ generalizes EX-inference. On the other hand, it can also be considered as a special case of EX-inference: For every $S \subseteq REC$ let $\tilde{S}_{m,n} = \{f : \lambda x.\ f(x+1) \in S \ \wedge\ f(0)$ is an index of an (m,n)-operator for $\lambda x.\ f(x+1)\}$. Then, $S \subseteq (m,n)EX$ iff $\tilde{S}_{m,n} \subseteq EX$.

Our next goal is a combinatorial characterization of the parameters m, n, m', n' such that $(m,n)EX \subseteq (m',n')EX$. To this end we consider (m,n)-computations on finite domains. This is a local combinatorial version of (m,n)-computation. It was first studied by Kinber [23] and Degtev [13].

Definition 3.3 Let $\ell \geq n \geq m \geq 0$. A set $V \subseteq \omega^\ell$ is called (m,n)-*admissible* iff for every n numbers x_i $(1 \leq x_1 < \cdots < x_n \leq \ell)$ there exists a vector $b \in \omega^n$ such that $(\forall v \in V)[v[x_1,\ldots,x_n] =^{n-m} b]$. In other words, there exists a function $G : \{1,\ldots,\ell\}^n \to \omega^n$ such that $v[x_1,\ldots,x_n] =^{n-m} G(x_1,\ldots,x_n)$ for all $1 \leq x_1 < \cdots < x_n \leq \ell$. Here $v[x_1,\ldots,x_n]$ denotes the projection of v on the components $x_1,\ldots,x_n$.

It is decidable whether for given m,n,m',n' and $\ell = \max(n,n')$, every (m,n)-admissible set $V \subseteq \omega^\ell$ is (m',n')-admissible. One has to check for all $G : \{1,\ldots,\ell\}^n \to \{1,\ldots,n\binom{\ell}{n}\}^n$ whether there is $H : \{1,\ldots,\ell\}^{n'} \to \{1,\ldots,n\binom{\ell}{n}\}^{n'}$ such that for all $v \in \omega^\ell$, if $\{v\}$ is (m,n)-admissible via G, then it is (m',n')-admissible via H. Also, if there is an (m,n)-admissible set $V \subseteq \omega^\ell$ which is not (m',n')-admissible, then there is a finite such V.

The following characterization says roughly that $(m,n)EX \subseteq (m',n')EX$ iff every finite (m',n')-operator can be transformed into an (m,n)-operator, i.e., (m',n')-computations can be *locally* replaced by (m,n)-computations.

Theorem 3.4 *Let $0 \leq m \leq n, 0 \leq m' \leq n', \ell = \max(n,n')$. Then $(m,n)EX \subseteq (m',n')EX$ iff every (m',n')-admissible set $V \subseteq \omega^\ell$ is (m,n)-admissible.*

Proof: $(\Leftarrow)$: If every (m',n')-admissible set $V \subseteq \omega^\ell$ is (m,n)-admissible, then we can compute from any index of an (m',n')-operator H in a uniform way an index of an (m,n)-operator $\tilde{H}$ such that every recursive function which is (m',n')-computable via H is (m,n)-computable via $\tilde{H}$.

More formally, $\tilde{H}$ is computed as follows: Given $x_1 < \cdots < x_n$, let $x_{n+1} = x_n + 1, \ldots, x_\ell = x_n + \ell - n$. The set

$$V = \{v \in \omega^\ell : (\forall 1 \leq i_1 < \cdots < i_{n'} \leq \ell)[v[i_1,\ldots,i_{n'}] =^{n'-m'} H(x_{i_1},\ldots,x_{i_{n'}})]\}$$

is (m',n')-admissible. By hypothesis there is a function $G : \{1,\ldots,\ell\}^n \to \omega^n$ such that V is (m,n)-admissible via G and, by the remarks above, such a G can be computed from H. Let $\tilde{H}(x_1,\ldots,x_n) = G(1,\ldots,n)$.

It easily follows that $(m,n)EX \subseteq (m',n')EX$: Suppose the IIM M (m,n)-infers $S \subseteq REC$. Given the index i of an (m',n')-operator for $f \in S$ we first compute an index i' of an (m,n)-operator for f and then simulate M with inputs i' and f.

$(\Rightarrow)$: For the converse, assume that there is an (m',n')-admissible set $V \subseteq \omega^\ell$ which is not (m,n)-admissible. By the remarks above, V can be chosen as a finite set, say $V = \{v_1,\ldots,v_k\}$. W.l.o.g., $v_1(1) \neq v_2(1)$. Fix $G : \{1,\ldots,\ell\}^{n'} \to \omega^{n'}$ such that V is (m',n')-admissible via G. Recall that $\{M_e\}_{e\in\omega}$ is an effective listing of all primitive recursive IIMs. For each e we define a function $f_e \in REC$ and an index i of a recursive function $F_e : \omega^{n'} \to \omega^{n'}$ such that f_e is (m',n')-computable via F_e, but

$M_e(i, f_e)$ does not infer f_e. Thus $S = \{f_e : e \geq 0\} \notin (m', n')EX$. But we take care that $S \in (m,n)EX$.

The basic idea for constructing f_e is standard. We try to build an increasing sequence $\tau_0 \prec \tau_1 \prec \cdots$, each time forcing an incorrect guess or a new mindchange, i.e., for each t we want that either $\varphi_{M_e(i,\tau_t)}(|\tau_t|) \neq \tau_{t+1}(|\tau_t|)$ (this corresponds to condition (1.2) below) or $M_e(i,\tau_t) \neq M_e(i,\sigma)$ for some σ with $\tau_t \preceq \sigma \preceq \tau_{t+1}$ (this corresponds to condition (1.3) below). If this succeeds we let $f_e = \lim_t \tau_t$. If we get stuck after building τ_t we let $f_e = \tau_t \star 0^\omega$.

In the construction below we have a variable mc in which we count the current number of errors enforced by the above actions.

The main new ingredient is that we simultaneously try to diagonalize against all (m,n)-operators, i.e., for each j we try to ensure that f_e is not (m,n)-computable via φ_j (this corresponds to condition (1.1) below). However, the diagonalization is allowed only if more than j errors have been enforced. In the variable L we record all j such that φ_j has been diagonalized.

The goal of the additional diagonalization is that f_e becomes inferable from any index j of an (m,n)-operator for f_e: To this end one simulates the construction below. As long as $mc \leq j$ it is assumed that $f_e =^* 0^\omega$. When $mc > j$ the inference algorithm uses the fact that φ_j is never diagonalized. This means that mc goes to infinity and hence $f_e = \lim_t \tau_t$. Thus, as soon as $mc > j$ the algorithm can simply output a program for $\lim_t \tau_t$.

The following construction depends on the parameters e, i. We define a sequence $\tau_0, \tau_1, \ldots$, a function f, and an (m,n)-operator F. Formally all of these objects depend on e, i. To keep the notation simple we omit these additional indices and assume that e, i are fixed. By the recursion theorem we will later obtain a recursive function h such that $i = h(e)$ is an index of $F_{e,i}$.

Construction of the τ-sequence:
Stage 0: Let $t = 0$, $\tau_0 = (e), mc = 0, L = \emptyset$.
Stage $s+1$: Let $I = \{|\tau_t|, \ldots, |\tau_t| + \ell - 1\}$.

1.) Check whether one of the following conditions is satisfied.
   (1.1) There is $j < mc$, $j \notin L$ such that $\varphi_{j,s}(x_1, \ldots, x_n) \downarrow \in \omega^n$ for all $x_1, \ldots, x_n \in I$ with $x_1 < \cdots < x_n$.
   (1.2) There is $b \in \{1,2\}$ such that $\varphi_{c,s}(|\tau_t|) \downarrow \neq v_b(1)$ for $c = M_e(i, \tau_t)$.
   (1.3) There is σ such that $\tau_t \star v_1 \preceq \sigma \preceq \tau_t \star v_1 \star 0^s$ and $M_e(i,\sigma) \neq M_e(i,\tau_t)$.

2.) If none of the conditions holds, then go to stage $s+2$. Otherwise choose the first condition (1.a) which holds, perform step (2.a), and go to stage $s+2$.
   (2.1) Choose the least j such that (1.1) holds. Compute q, $1 \leq q \leq k$, such that there are $x_1, \ldots, x_n \in I$ with $x_1 < \cdots < x_n$ and $\varphi_j(x_1, \ldots, x_n)$ agrees with v_q in at most $m-1$ components. (Note that q exists, since otherwise φ_j witnesses that V is (m,n)-admissible.)
   Let $\tau_{t+1} = \tau_t \star v_q \star 0^s$; $t = t+1$; $L = L \cup \{j\}$.
   (2.2) Choose b as in (1.2) and let $\tau_{t+1} = \tau_t \star v_b \star 0^s$; $t = t+1$; $mc = mc+1$.
   (2.3) Let $\tau_{t+1} = \tau_t \star v_1 \star 0^t$; $t = t+1$; $mc = mc+1$.

End of construction.

Definition of f:
If t is incremented only finitely often, then let t' denote its maximal value and define $f = \tau_{t'} \star v_1 \star 0^\omega$. Otherwise define $f = \lim_t \tau_t$.

Definition of F:
We define $F(y_1, \ldots, y_{n'}) = (b_1, \ldots, b_{n'})$ as follows for $y_1 < \cdots < y_{n'}$:
Let $s = y_{n'}$ and let t' denote the value of t at the end of stage $s+1$. Choose $z_1, \ldots, z_{n'}$ such that $1 \leq z_1 < \cdots < z_{n'} \leq \ell$ and $\{y_j : 1 \leq j \leq n' \ \wedge \ |\tau_{t'}| \leq y_j < |\tau_{t'}| + \ell\} \subseteq \{|\tau_{t'}| + z_j - 1 : 1 \leq j \leq n'\}$.
If $y_j < |\tau_{t'}|$, then let $b_j = \tau_{t'}(y_j)$.
If $y_j \geq |\tau_{t'}| + \ell$, then let $b_j = 0$.
If $y_j = |\tau_{t'}| + z_{j'} - 1$ for some $1 \leq j' \leq n'$, then let $b_j = G(z_1, \ldots, z_{n'})[j']$.

Note that the definition of F is uniform in e, i and that F is defined for all n'-tuples $y_1 < \cdots < y_{n'}$. The definition of f is non-uniform, but f is in any case a total recursive function.

Claim 0: f is (m', n')-computable via F.
Proof: Consider $y_1 < \cdots < y_{n'}$ and let $s, t', z_1, \ldots, z_{n'}, b_1, \ldots, b_{n'}$ be as above. If $y_j < |\tau_{t'}|$, then $b_j = \tau_{t'}(y_j) = f(y_j)$ since $\tau_{t'} \prec f$. If $y_j \geq |\tau_{t'}| + \ell$, then $b_j = 0 = f(y_j)$ since $\tau_{t'} \star v \star 0^s \prec f$ for some $v \in V$. Otherwise, $|\tau_{t'}| \leq y_j < |\tau_{t'}| + \ell$. Suppose that there are a such y_j's. Since the other $n' - a$ components are correct, we need to show that at least $m' - (n' - a)$ of the corresponding b_j's are correct. Note that the b_j's are components of a projection of $G(z_1, \ldots, z_{n'})$ on a set of size a. By construction, $G(z_1, \ldots, z_{n'}) =^{n'-m'} (f(|\tau_{t'}| + z_1 - 1), \ldots, f(|\tau_{t'}| + z_{n'} - 1))$. Thus any projection on a components has at least $m' - (n' - a)$ correct components. $\Box$

Claim 1: $M_e(i, f)$ does not converge to an index of f.
Proof: a.) Suppose that t is incremented only finitely often and reaches its maximal value t' at stage s'. Then conditions (1.2) and (1.3) do not hold at any later stage. Thus $\varphi_{M_e(i,\tau_{t'})}(|\tau_{t'}|)$ is undefined and $M_e(i, \tau_{t'}) = M_e(i, \tau_{t'} \star v_1 \star 0^s)$ for all s, i.e., $M_e(i, f)$ converges to an index of a non-total function.

b.) If t is incremented infinitely often, then also mc is incremented infinitely often. (If mc does not change, then t can be incremented only via (1.1). But this can happen at most mc times.) Thus, $M_e(i, f)$ makes infinitely many mindchanges or for infinitely many $\tau \prec f$ we have $\varphi_{M_e(i,\tau)}(|\tau|) \neq f_e(|\tau|)$. In particular, $M_e(i, f)$ does not converge to an index of f. $\Box$

Definition of f_e, F_e, and S:
Let $F_{e,i}, f_{e,i}$ denote the recursive functions F, f in the construction with parameters e, i. Since the construction of $F_{e,i}$ is uniform in e, i, there is a recursive function g such that $F_{e,i} = \varphi_{g(e,i)}$. By the recursion theorem with parameters there is a recursive function h such that $\varphi_{h(e)} = \varphi_{g(e,h(e))}$ for all e. Let $F_e = F_{e,h(e)}$, $f_e = f_{e,h(e)}$, and $S = \{f_e : e \geq 0\}$.

Claim 2: $h(e)$ is an index of an (m', n')-operator for f_e.
Proof: By Claim 0, F_e is an (m', n')-operator of f_e. By definition of h, $h(e)$ is an index of F_e. $\Box$

Claim 3: $S \notin (m', n')EX$.
Proof: Suppose that $S \in (m', n')EX$. Then there is an e such that M_e infers S. By

Claim 1, $M_e(h(e), f_e)$ does not converge to an index of f_e. Since, by Claim 2, $h(e)$ is an index of an (m', n')-operator for f_e, we obtain a contradiction. $\Box$

Claim 4: $S \in (m,n)EX$.
Proof: The following algorithm infers S: Given $f \in S$ and an index j of an (m,n)-operator for f. First obtain $e = f(0)$ and compute $i = h(e)$. Then simulate the construction of the τ-sequence with parameters e, i. As long as $mc \leq j$ assume that $f_e =^* 0^\omega$ and perform identification by enumeration. If it is discovered that $mc > j$, then output a program which computes $\lim_t \tau_t$.

It remains to show that this algorithm is correct. If at each stage $mc \leq j$, then t is incremented only finitely often and $f_e =^* 0^\omega$. If $mc > j$ and t is incremented only finitely often, then there is a stage at which j is the least number for which (1.1) holds, so φ_j would be diagonalized which contradicts the hypothesis that φ_j is an (m,n)-operator for f_e. Thus, t is incremented infinitely often and $f_e = \lim_t \tau_t$, i.e., the final guess of the algorithm is correct. $\Box$ $\blacksquare$

Remarks: a.) As $\{0,1\}^n$ is (trivially) $(0,n)$-admissible, but not $(1,n)$-admissible, it follows that $EX \subset (1,n)EX$ for all $n \geq 1$. This shows that even if very weak operators are provided, one can still learn more than without them.

b.) In the proof of ($\Rightarrow$) we construct recursive functions such that every (m,n)-operator of f has high running time. Indeed, in the simulation one uses the running-time of the program which computes the operator rather than the extensional information provided by the operator. This is inevitable: Suppose $S \in (1,n)EX$ and every $f \in S$ is $(1,n)$-computable by an operator which is easily computable, say primitive recursive. Then $S \in EX$, since we can successively try all primitive recursive $(1,n)$-operators as additional inputs, until we settle down on one which is consistent with f. — Note however, that even if we restrict all operators to be computable in polynomial time, they can still $(n-1,n)$-compute arbitrarily complex recursive functions (see [1, 22]).

It is also natural to define a notion of inference where we want to learn an approximation of f instead of f, i.e., a program of an (m,n)-operator for f instead of a program for f. Call this notion $EX(m,n)$. We get the following interesting and nontrivial duality between both notions.

Theorem 3.5 *$EX(m,n) \subseteq EX(m',n')$ iff $(m',n')EX \subseteq (m,n)EX$.*

Proof sketch: We use the characterization of Theorem 3.4.

If $(m',n')EX \subseteq (m,n)EX$, then every (m,n)-operator can be uniformly transformed into an (m',n')-operator; hence, if we can learn an (m,n)-operator for f we can also learn an (m',n')-operator.

For the other direction, if $(m',n')EX \not\subseteq (m,n)EX$, then there is an (m,n)-admissible finite set V which is not (m',n')-admissible. We can use V to diagonalize over machines which learn (m',n')-operators while constructing an (m,n)-operator. This is formally similar to (but easier than) the proof of Theorem 3.4 ($\Rightarrow$). The details are left to the reader. $\blacksquare$

A couple of explicit results on (m,n)-admissible sets are listed in [27, Section 3.3] (see also [21, Section 5]). For instance, Kinber [23] showed that, for $n \geq 2$, every $(n, n+1)$-admissible set is $(n+1, n+2)$-admissible. If $n - m > n' - m'$, then the set of all binary vectors with at most $n - m$ ones is (m,n)-admissible but not (m', n')-admissible. The set $\{1^\ell, 2^\ell, \ldots, n^\ell\}$ is $(1,n)$-admissible but not (m', n')-admissible for $\ell = \max(n, n')$ and $m'/n' > 1/n$. Hence, we get the following corollary.

Corollary 3.6

a.) $(n, n+1)EX = (n+1, n+2)EX$ *for all* $n \geq 2$.

b.) $(m,n)EX \subset (m+1, n)EX$ *for all* $1 \leq m < n$.
In particular, $REC \notin (n-1, n)EX$.

c.) $(m', n')EX \not\subseteq (1,n)EX$ *if* $1/n < m'/n'$.

4 Probabilistic Learning from Frequency Computations

We have shown that REC is not inferable by an IIM even if $(n-1,n)$-computations of f are provided. In this section we answer the question whether REC is inferable from (m,n)-computations by a *probabilistic* IIM with positive probability. We show that this is indeed the case if $m/n > 1/2$. Further, we determine the maximal $p = p(m,n)$ such that REC can be learned from (m,n)-computations with probability p.

We first recall some notation and results from [38]. Let $EX_{prob}(p)$ denote the set of all $S \subseteq REC$ that can be EX-inferred by a probabilistic IIM with probability at least p. Let $EX[k]$ denote the set of all S which can be EX-inferred by a team of k IIMs. The same notation is used for BC instead of EX. Pitt [38] proved the following surprising connection between probabilistic inference and team inference.

Proposition 4.1 [38] *For all natural numbers* $k \geq 1$ *and all real numbers* $p \in (0,1]$*:*

$$EX_{prob}(p) \subseteq EX[\lfloor 1/p \rfloor] \ \wedge \ EX[k] \subseteq EX_{prob}(1/k).$$

The same holds for BC *instead of* EX.

Using Smith's team hierarchy result [40] that $EX[k] \subset EX[k+1]$ and $BC[k] \subset BC[k+1]$ for all $k \geq 1$, Pitt concluded that the probabilistic classes form an infinite discrete hierarchy with breakpoints of the form $1/k$.

Proposition 4.2 [38, 40] *For all natural numbers* $k \geq 1$ *and all real numbers* $p \in (0,1]$*:*

$$EX_{prob}(p) = EX[k] \iff \frac{1}{k+1} < p \leq \frac{1}{k} \iff BC_{prob}(p) = BC[k].$$

In particular, $REC \notin EX_{prop}(p)$.

These notions can be transferred in a straightforward way to our setting:

Let $(m,n)EX_{prob}(p)$ denote the set of all $S \subseteq REC$ such that there is a probabilistic IIM M such that for every $f \in S$ and every index e of an (m,n)-operator of f, $M(e,f)$ converges to an index of f with probability at least p.

Let $(m,n)EX[k]$ denote the set of all $S \subseteq REC$ such that there is a team of k IIMs $M_1,\dots,M_k$ such that for every $f \in S$ and every index e of an (m,n)-operator for f there is $i, 1 \le i \le k$ such that $\lim_t M_i(e, f \restriction t)$ exists and is an index of f. The classes $(m,n)BC_{prob}(p)$ and $(m,n)BC[k]$ are defined analogously.

The proof of Pitt's Proposition 4.1 can be straightforwardly transferred and yields the following.

Proposition 4.3 *For all natural numbers k,m,n, with $k \ge 1$, and all real numbers $p \in (0,1]$:*

$$(m,n)EX_{prob}(p) \subseteq (m,n)EX[\lfloor 1/p \rfloor] \ \wedge \ (m,n)EX[k] \subseteq (m,n)EX_{prob}(1/k).$$

The same holds for BC instead of EX.

Our first result shows that no probabilistic IIM can infer REC with positive probability from frequency computations with frequency less than or equal to $1/2$.

Theorem 4.4 *If $0 \le m \le \frac{n}{2}$ and $0 < p \le 1$, then $REC_{0,1} \notin (m,n)BC_{prob}(p)$.*

Proof: Let $C \subseteq REC_{0,1}$ be the set of all recursive functions g such that there is a sequence $a_0, a_1, \dots$ with g the characteristic function of $\{\langle a_0,\dots,a_i\rangle : i \ge 0\}$. It is easy to see that there is a $(1,2)$-operator F such that every $g \in C$ is $(1,2)$-computable via F. It follows that for every m,n with $m/n \le 1/2$ there is a fixed (m,n)-operator $F_{m,n}$ such that every $g \in C$ is (m,n)-computable via $F_{m,n}$.

Suppose for a contradiction that $C \in (m,n)BC_{prob}(p)$ with $p \in (0,1]$. Let $k = \lfloor 1/p \rfloor$. Then, by Proposition 4.3, $C \in (m,n)BC[k]$. Let e be an index of $F_{m,n}$. There is a team of k machines which BC-infers C with additional information e. If this constant additional information is hard-wired into the IIMs, we obtain $C \in BC[k]$. Note that every $f \in REC$ can be transformed into a unique $g \in C$ and vice versa, by recursive operators. Thus it follows that $REC \in BC[k]$. This contradicts the team hierarchy result of Smith [40]. ∎

Now we turn to frequencies greater than $1/2$. In this case there exist probabilistic IIMs which can infer REC from frequency computations. We determine the maximal probability p for which this can be done.

Theorem 4.5 *Let $\frac{n}{2} < m \le n$. Then $REC \in (m,n)EX_{prob}(\frac{1}{n-m+1})$, but $REC_{0,1} \notin (m,n)BC_{prob}(p)$ for any probability $p > \frac{1}{n-m+1}$.*

Proof: Let $m,n \ge 1$ be given with $\frac{n}{2} < m \le n$. By Proposition 4.3 is suffices to show the upper bound $REC \in (m,n)EX[n-m+1]$ and the lower bound $REC_{0,1} \notin (m,n)BC[n-m]$.

a.) Proof of $REC \in (m,n)EX[n-m+1]$: This requires a combination of methods from [19] and [21]. Given an (m,n)-operator R we define uniformly as in [19, p. 684] a

recursive tree $T \subseteq \{0,1\}^*$ whose branches represent the graphs of all partial functions which are (m,n)-computable via R.
More formally, we call a string σ *single valued* if

$$(\forall \langle x, y_1\rangle < |\sigma|)(\forall \langle x, y_2\rangle < |\sigma|)[(\sigma(\langle x, y_1\rangle) = 1 \ \wedge \ \sigma(\langle x, y_2\rangle) = 1) \implies y_1 = y_2].$$

We call a string σ *R-consistent* if for all $x_1 < \cdots < x_n$, if $R(x_1, \ldots, x_n) = (z_1, \ldots, z_n)$ and $\langle x_1, z_1\rangle, \ldots, \langle x_n, z_n\rangle < |\sigma|$, then $|\{i : \sigma(\langle x_i, z_i\rangle) = 1\}| \geq m$. Then we define T as follows.

$$T = \{\sigma \in \{0,1\}^* : \sigma \text{ is single valued and } R\text{-consistent}\}.$$

Assume that $f \in REC$ is (m,n)-recursive via R. Then the characteristic function of $\mathrm{Graph}(f) = \{\langle x, f(x)\rangle : x \in dom(f)\}$ is a branch of T. Conversely suppose that $A \in [T]$, i.e., χ_A is a branch of T. Then there is a partial function g such that $A = \mathrm{Graph}(g)$ and for all $x_1 < \cdots < x_n$, $|\{i : x_i \in dom(g) \ \wedge \ (R(x_1, \ldots, x_n))_i = g(x_i)\}| \geq m$. Since $m > n/2$ it follows that $f =^{2(n-m)} g$. In particular, there are at most $2(n-m)$ arguments for which g is undefined.

The Vapnik-Chervonenkis dimension of $[T]$, $dim(T)$, is the maximal number d such that there exist $z_1 < \cdots < z_d$ with

$$(\forall \tau \in \{0,1\}^d)(\exists A \in [T])[\tau = (\chi_A(z_1), \ldots, \chi_A(z_d))].$$

See [7] for more information on this notion. Note that we have $dim(T) \leq n - m$. Otherwise there exist pairwise distinct numbers $z_1 = \langle x_1, y_1\rangle, \ldots, z_{n-m+1} = \langle x_{n-m+1}, y_{n-m+1}\rangle$ and branches of T whose characteristic functions on $z_1, \ldots, z_{n-m+1}$ realize all possible 0/1-vectors of length $n-m+1$. Since every branch is single valued, it follows that the x_i's are pairwise distinct. Assume that $x_1 < \cdots < x_{n-m+1}$ and let $(a_1, \ldots, a_n) = R(x_1, \ldots, x_{n-m+1}, x_{n-m+1}+1, \ldots, x_{n-m+1}+m-1)$. Choose a branch A such that $[A(z_i) = 1 \Leftrightarrow y_i \neq a_i]$ for $1 \leq i \leq n-m+1$. But this means that an initial segment of A is not R-consistent, a contradiction.

It is shown in [21, Lemma 3.12] that if T is an infinite recursive tree with $dim(T) \leq d$ such that any two branches agree almost everywhere, then one can compute uniformly from any Δ_0-index of T the indices of $d+1$ partial recursive functions such that one of them is total recursive and computes a branch of T up to finitely many errors. If we combine the results presented so far we get the following.

Claim: There is a uniform procedure to compute from any index of an (m,n)-operator R a list of $n-m+1$ indices $i_1, \ldots, i_{n-m+1}$ such that if there is $f \in REC$ which is (m,n)-recursive via R, then there is $1 \leq j \leq n-m+1$ and such that φ_{i_j} is total, $\{0,1\}$-valued, and $\varphi_{i_j} = \mathrm{Graph}(g)$ for some g with $f =^* g$.

Now the inference procedure for $REC \in (m,n)EX[n-m+1]$ is clear: On input (e,f), where e is an index of an (m,n)-operator for f, each team member computes the list $i_1, \ldots, i_{n-m+1}$ as in the claim. The j-th team member assumes that φ_{i_j} is total, $\{0,1\}$-valued and $\varphi_{i_j} = \mathrm{Graph}(g)$ for some g with $f =^* g$. While reading f it checks whether $f(x) = g(x)$ and outputs a program for g where all differences with f that have been discovered so far are patched. By the claim, for one of the team members the assumption is correct. Thus, this team member will eventually output a correct program for f.

b.) $REC_{0,1} \notin (m,n)BC[n-m]$: Suppose for a contradiction that there is team of $n-m$ machines $M_1, \ldots, M_{n-m}$ which infers $REC_{0,1}$ from (m,n)-computations. We combine the proof of the lower bound in [21, Theorem 3.5] with a diagonalization method for teams and construct a function $f \in REC_{0,1}$ and an (m,n)-operator R for f. By the recursion theorem, we can use an index e of R in the construction. For $1 \le i \le n-m$, we ensure that $M_i(e,f)$ does not BC-infer f.

The function f is initialized as the constant zero function. During the construction $f(x)$ may be updated from zero to one. For each i we are looking for possibilities to force an error in the inference process of M_i with inputs e and f. To this end we are looking for r such that $\varphi_{M_i(e,f\restriction r)}(r) = 0 = f(r)$ and then update $f(r) = 1$ and ensure that $f(x)$ does not change for $x \le r$. If this can be done for infinitely many r, then $M_i(e,f)$ produces infinitely many incorrect hypotheses. If this can be done only finitely often, then almost all hypotheses of $M_i(e,f)$ are incorrect. In any case, $M_i(e,f)$ does not BC-infer f.

Since there is a conflict between the diagonalization and preservation actions for different machines, we are using a priority ordering of the machines that is updated during the construction according to the 'least recently used principle': If $q = (a_1, \ldots, a_{n-m})$ is the current ordering of machine indices and there are several candidates for diagonalization, then we select the machine with the leftmost index, say $i = a_k$. $f(r)$ is updated accordingly, and it is ensured that all later diagonalization actions of M_{a_j} with $j \ge k$ start at values greater than r (thereby preserving $f \restriction (r+1)$ with priority k). In the updated sequence q', we insert i at the last position, i.e., $q' = (a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_{n-m}, a_k)$.

This update rule for the diagonalization values will automatically allow us to compute an (m,n)-operator for f.

Construction:
Stage 0: Initialize $q = (1, 2, \ldots, n-m)$. Let $f = \lambda x.\ 0$; $x_i = 0$ for $i = 1, \ldots, n-m$.
Stage $s+1$: If there is an i for which there exists (a least) r such that

$$x_i < r \le s \ \wedge\ f(r) = 0 \ \wedge\ \varphi_{c,s}(r) = 0 \ \text{ for } c = M_i(e, f \restriction r),$$

then select (i,r) such that i appears in the leftmost position in q, say $i = a_k$.
Update $f(r) = 1$, let $x_{a_j} = 2s$ for $k \le j \le n-m$.
Move i to the rear of q, i.e., let $q = (a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_{n-m}, a_k)$.
End of construction.

The (m,n)-operator $R(y_1, \ldots, y_n)$ is defined as follows:
Given $y_1 < \cdots < y_n$ let $s = y_n$ and let f_s be the function f at the end of stage $s+1$.
Then let $R(y_1, \ldots, y_n) = (f_s(y_1), \ldots, f_s(y_n))$.

From the update rule for the x_i's, it follows that f is (m,n)-recursive via R.

Let I be the set of all i such that i is selected at infinitely many stages. Let I' be the set of all i which are selected only finitely often. Then, by the update rule for q, there is a stage t_0 such that in all stages $t > t_0$, all elements from I' occupy the first $|I'|$ positions of q and the $x_i, i \in I'$, do not change. If $|I'| = n-m$, then $f = f_{t_0}$. If $|I'| = k-1 < n-m$, then $f(x) = f_t(x)$ for $t = (\mu s > t_0)[x_{a_k,s} > x]$ where $x_{a_k,s}$ denotes the value of x_{a_k} at the end of stage $s+1$. In particular, f is recursive.

From the update rule for the x_i's, it follows that $f(r) = 0$ for infinitely many r.

No $M_i, i \in I'$, BC-infers f: Let x'_i be the final value of x_i. Then for all $r > x_i$ such that $f(r) = 0$, $M_i(e, f \restriction r)$ outputs a program which is undefined at r or computes a nonzero value (otherwise i would eventually be selected and x_i would increase). Thus, $M_i(e, f)$ outputs infinitely often an incorrect program.

Now suppose for a contradiction that $i \in I$ and $M_i(e, f \restriction r)$ is an index of f for all $r \geq r_0$. Consider a stage $s+1 > t_0$ with $x_i > r_0$ where i occupies the $(|I'|+1)$-th position in q and is selected (by the update rule for q there are infinitely many such stages). At stage $s+1$ we put $f(r) = 1 \neq 0 = \varphi_c(r)$ for $c = M_i(e, f_s \restriction r)$ and some $r > r_0$. By the choice of t_0 and the update rule for the x_j's we have $f_s \restriction (r+1) = f \restriction (r+1)$. Thus $c = M_i(e, f \restriction r)$ is not a program for f, a contradiction.

Therefore, none of the M_i's BC-infers f with additional information e. ▮

We obtain the following interesting corollary on team inference. It shows that there are natural team hierarchies of arbitrary *finite* length.

Corollary 4.6

a.) *If* $\frac{n}{2} < m \leq n$, *then* $(m,n)EX[k] \subset (m,n)EX[k+1]$ *for* $1 \leq k \leq n-m$, *and* $(m,n)EX[k] = (m,n)EX[k+1] = 2^{REC}$ *for* $k > n-m$.

b.) *If* $0 \leq m \leq \frac{n}{2}$, *then* $(m,n)EX[k] \subset (m,n)EX[k+1]$ *for all* $k \geq 1$.

The same holds for BC instead of EX.

Proof: a.) Let $\frac{n}{2} < m \leq n$. By proof of Theorem 4.5 it remains to show that $(m,n)EX[k] \subset (m,n)EX[k+1]$ and $(m,n)BC[k] \subset (m,n)BC[k+1]$ for $1 \leq k \leq n-m$. By a modification of the proof that $REC_{0,1} \notin (m,n)BC[n-m]$ one can even show the following:

$$\text{If } 1 \leq k \leq n-m, \text{ then } EX[k+1] - (m,n)BC[k] \neq \emptyset.$$

To this end we diagonalize over all k-tuples of IIMs. For the i-th tuple we use the old construction to build a function f_i with $1^i0 \preceq f_i$ and an index $g(i)$ of an (m,n)-operator for f_i such that none of the IIMs in the i-th tuple infers f_i with additional information $g(i)$. The function $g \in REC$ is obtained by the recursion theorem with parameters. Let $S = \{f_i : i \geq 0\}$. By construction, $S \notin (m,n)BC[k]$. It remains to verify that $S \in EX[k+1]$:

On input f the EX-team first determines i such that $1^i0 \preceq f$. Then it simulates the construction of f_i. The j-th team member, $1 \leq j \leq k+1$, assumes that $j-1$ is maximal such that an initial segment of length $j-1$ of the queue q is almost always constant. It is not difficult to check that the team member with the correct guess can EX-infer f_i.

b.) By the team hierarchy result of Smith [40] there is a set $S \subseteq REC$ with $S \in EX[k+1] - BC[k]$. Let C be the set as defined in the proof of Theorem 4.4. As we saw there, for any $S' \subseteq C$, all $\ell \geq 1$, and all m, n with $1 \leq m \leq \frac{n}{2}$ we have $[S' \in (m,n)EX[\ell] \Leftrightarrow S' \in EX[\ell]]$, and the same for BC instead of EX. Further, S can be translated into a subset S' of C such that $S' \in EX[k+1] - BC[k]$. Thus the second part of the corollary follows. ▮

5 Other Notions of Approximative Information

In this section we consider other notions of approximative information and determine the maximal probability p with which all total recursive $\{0,1\}$-valued functions are learnable. In each case we provide indices of recursive or r.e. trees with certain properties such that the function which is to be learned is an infinite branch of the tree. If one generalizes from binary to arbitrary trees (and thus arbitrary $f \in REC$) one gets a notion which corresponds to r.e. trees in the binary case. Therefore, we only consider the $\{0,1\}$-valued case.

Recursive trees capture a wide range of approximative information: Suppose we have a first-order specification of f, i.e., an r.e. set S of sentences containing the function symbol f. Then, the set of all consistent interpretations $f' : \omega \to \omega$ of f are just the branches of a recursive tree T which can be computed uniformly from S: By the compactness theorem, f' is inconsistent with S iff there is an initial segment $\sigma = (y_0, \dots, y_n) \prec f'$ such that $S_\sigma = S \cup \{f(0) = y_0, \dots, f(n) = y_n\}$ is an inconsistent set of formulas, which is an r.e. property of σ. Let $\sigma_0, \sigma_1, \dots$ be a recursive enumeration of all such σ. Define $T = \{\tau : \sigma_i \not\preceq \tau \text{ for all } i \leq |\tau|\}$.

For all notions of approximative information which we consider the analogue of Proposition 4.3 holds. Therefore we first state our results in terms of team inference. At the end of this section we state the corresponding results for probabilistic inference.

5.1 Trees of Bounded Variation

We consider trees where any two branches differ in at most a constant number of arguments.

Definition 5.1 For $A, B \subseteq \omega$, let $A \Delta B$ denote the symmetric difference of A and B. For any tree $T \subseteq \{0,1\}^*$, let $(\Delta T) = \sup\{|A \Delta B| : A, B \text{ branches of } T\}$. We say that T has *bounded variation* if $(\Delta T) < \infty$.

If a recursive tree $T \subseteq \{0,1\}^*$ has bounded variation, then every branch of T is recursive [42] (see also [19, 21]). We now determine, for each n, the optimal team size such that all recursive functions are learnable given recursive trees T with $(\Delta T) \leq n$ as additional information.

Definition 5.2 Let $d_{EX}(n)$ denote the least team size k such that there is a team of k IIMs that EX-infers every $f \in REC_{0,1}$ given any Δ_0-index of a recursive tree $T \subseteq \{0,1\}^*$ such that $(\Delta T) \leq n$ and f is a branch of T. $d_{BC}(n)$ is defined analogously for BC- instead of EX-inference.

Theorem 5.3 *For $n \geq 0$, $d_{EX}(n) = n+1$ and $d_{BC}(n) = \lceil \frac{n+1}{2} \rceil$.*

Proof: a.) $d_{EX}(n) \leq n+1$: Fix n. It is shown in [21] that there is a uniform procedure to compute, for any Δ_0-index of an infinite recursive tree $T \subseteq \{0,1\}^*$ with $(\Delta T) \leq n$, a set of $n+1$ partial recursive functions such that one of these functions is total and computes a branch of T up to finitely many errors. Each of the team members computes one of these functions and patches all differences with f. The team member which got the *total* finite variant of f successfully EX-infers f.

b.) $d_{EX}(n) > n$: We modify the proof of the lower bound in [21, Theorem 3.13] to diagonalize a team of n EX-machines. Suppose for a contradiction that each $f \in REC_{0,1}$ is EX-inferred by the team $M_1, \ldots, M_n$ from Δ_0-indices of recursive trees $T \subseteq \{0,1\}^*$ such that $(\Delta T) \leq n$ and f is a branch of T.

We construct a recursive function f and a tree T with $(\Delta T) \leq n$ and $f \in [T]$. By the recursion theorem we can use a Δ_0-index e of T in the construction. The construction is a slight modification of the construction in the proof of Theorem 4.5.

Construction:
Stage 0: Initialize $q = (1, 2, \ldots, n)$. Let $f = \lambda x.\ 0$; $x_i = i$ for $i = 1, \ldots, n$.
Stage $s+1$: If there is an i such that one of the following conditions holds:

(1) $\varphi_{c,s}(x_i) = 0$ for $c = M_i(e, f \restriction x_i)$,
(2) $(\exists r)[x_i < r \leq s \ \wedge\ M_i(e, f \restriction x_i) \neq M_i(e, f \restriction r)]$,

then select that i which appears in the leftmost position in q, say $i = a_k$.
If (1) holds, then update $f(x_i) = 1$.
In both cases let $x_{a_j} = sn + a_j$ for $k \leq j \leq n$ and move i to the rear of q, i.e., let $q = (a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_n, a_k)$.
End of construction.

Note that in (1) we look for a diagonalization at x_i and in (2) we look for a mindchange. If from some point on, neither (1) nor (2) holds and $M_i(e, f)$ converges to an index c, then $\varphi_c(x_i) \neq 0 = f(x_i)$.

Similarly as in the previous proof it follows that f is recursive and f is not EX-inferred by any of the M_i's.

It remains to give a uniform definition of T such that $(\Delta T) \leq n$ and $f \in [T]$. This is analogous to the proof in [21, Theorem 3.13]. Note in each stage $x_i \equiv i \bmod n$. Thus the values of x_i, x_j for $i \neq j$ are different. Let $x_{i,s}$ denote the value of x_i at the end of stage $s+1$. Define

$$T = \{\sigma \in \{0,1\}^* : (\forall x < |\sigma|)[x \notin \{x_{1,|\sigma|}, \ldots, x_{n,|\sigma|}\} \rightarrow \sigma(x) = f_{|\sigma|}(x)]\}.$$

Clearly $f \in [T]$. Let ℓ be the number of x_i's which are incremented only finitely often and let $z_1, \ldots, z_\ell$ be their final values. Then we get $[T] = \{g \in \{0,1\}^\omega : (\forall x)[x \notin \{z_1, \ldots, z_\ell\} \rightarrow f(x) = g(x)\}$. Thus $(\Delta T) = \ell \leq n$.

c.) $d_{BC}(n) \leq \lceil \frac{n+1}{2} \rceil$: Fix n. It is shown in [21] that there is a uniform procedure to compute for any Δ_0-index of an infinite recursive tree $T \subseteq \{0,1\}^*$ with $(\Delta T) \leq n$ a set of $\lceil \frac{n+1}{2} \rceil$ partial recursive functions such that one of these functions computes a branch of T up to finitely many errors. (Note that, in contrast to a.), it is possible that none of the functions is total.) Each of the $\lceil \frac{n+1}{2} \rceil$ team members outputs programs for one of these functions which are patched with the correct values of f on arbitrary large initial segments. The team member which received the finite variant of f successfully BC-infers f.

d.) $d_{BC}(n) \geq \lceil \frac{n+1}{2} \rceil$: Trakhtenbrot [42] (see also [19, 21]) proved that if $k/2 < h \leq k$, then one can compute in a uniform way for any (h,k)-operator F a recursive tree $T \subseteq \{0,1\}^*$ with $(\Delta T) \leq 2(k-h)$ such that every $\{0,1\}$-valued function f which is (h, k)-recursive via F is a branch of T. Therefore, the lower bound from Theorem 4.5, for $h = k + 1, k = 2k + 1$, implies that $d_{BC}(2k + 1) \geq d_{BC}(2k) \geq k + 1$. ▮

Remark: R.e. trees of bounded variation are of less help. One can show that no finite team size suffices to infer $REC_{0,1}$ from indices of r.e. trees, even for r.e. trees with exactly one branch.

5.2 Trees of Bounded Width

We consider trees which have at most a constant number of nodes in each level.

Definition 5.4 The *width* $w(T)$ of a tree $T \subseteq \{0,1\}^*$ is the maximum number of nodes on any level, i.e., $w(T) = \max\{|T \cap \{0,1\}^k| : k \geq 0\}$.

If a recursive tree $T \subseteq \{0,1\}^*$ has bounded width, then every branch of T is recursive. In fact, this holds also for r.e. trees of bounded width [37]. We determine, for both the recursive and the r.e. cases, the optimal team size such that all recursive functions are inferable given such trees as additional information.

Definition 5.5 Let $w_{EX}(n)$ denote the least team size k such that there is a team of k IIMs that EX-infers every $f \in REC_{0,1}$ given any Δ_0-index of a recursive tree $T \subseteq \{0,1\}^*$ such that $w(T) \leq n$ and f is a branch of T. If Σ_1-indices are provided for T the corresponding team size is denoted by $w_{EX}^{re}(n)$. The analogous numbers for BC-teams are $w_{BC}(n)$ and $w_{BC}^{re}(n)$.

Theorem 5.6 *For $n \geq 1$, $w_{EX}(n) = w_{EX}^{re}(n) = w_{BC}^{re}(n) = n$ and $w_{BC}(n) = 1$.*

Proof: If T has bounded width and f is a branch of T, then there is $\sigma_0 \prec f$ such that f is the unique branch of T which extends σ_0. If we have a Δ_0-index of T and any τ with $\sigma_0 \preceq \tau \prec f$, we can compute an index of f. Using this fact it easily follows that $w_{BC}(n) = 1$.
Clearly $w_{EX}(n) \leq w_{EX}^{re}(n)$ and $w_{BC}^{re}(n) \leq w_{EX}^{re}(n)$.

a.) $w_{EX}^{re}(n) \leq n$: If f is an infinite branch of T let $w(T, f) = \sup\{w(T[\sigma]) : \sigma \prec f\}$. It is shown in [21] that given k, σ, $\sigma \prec f$, and a Σ_1-index of T with $w(T[\sigma]) = w(T, f) = k$ we can uniformly compute an index of f.

For each k, $1 \leq k \leq n$, we have a team member M_k which assumes that $w(T, f) = k$ and works as follows: At the beginning it initializes a local variable $\sigma = \lambda$ and outputs an index of f on the assumption that $w(T[\sigma]) = w(T, f) = k$. Then it enumerates T. If after s steps it is discovered that $w(T[\sigma]) > k$, then it updates $\sigma = (f(0), \ldots, f(s))$ and outputs a new index for f, etc. Clearly, if $k = w(T, f)$, then after finitely many steps $w(T[\sigma]) = k$ and from then on M_k outputs a fixed correct index of f.

b.) $w_{BC}^{re}(n) > n - 1$: Suppose for a contradiction that each $f \in REC_{0,1}$ is BC-inferred by the team $M_1, \ldots, M_{n-1}$ from Σ_1-indices of r.e. trees $T \subseteq \{0,1\}^*$ such that $w(T) \leq n$ and f is a branch of T.

We construct a recursive function f and an r.e. tree T with $w(T) \leq n$ and $f \in [T]$. By the recursion theorem we can use a Σ_1-index e of T in the construction. The construction is just the diagonalization in the proof of Theorem 4.5 where $n - m$ is replaced by $n - 1$.

Let f_s denote the version of f at the end of stage $s + 1$. We define a tree T as follows:

$$T = \{\sigma \in \{0,1\}^* : (\exists s)[\sigma \preceq f_s \upharpoonright s]\}.$$

Clearly T is a tree which is uniformly r.e., and f is a branch of T. We claim that $w(T) \leq n$: Consider any level k, let $s_1 = k + 1$, and let $s_2 < \cdots < s_d$ be those $s > s_1$ such that $f_s \upharpoonright (k+1) \neq f_{s-1} \upharpoonright (k+1)$. It follows that $|T \cap \{0,1\}^k| = d$. At each stage s_j, $2 \leq j \leq d$, some i with $x_i < k$ is selected and $f(r)$ is updated for some r with $x_i < r \leq k \leq s_j$. Then x_i is updated to $2s_j > k$. Hence for each i there is at most one such stage and therefore $d \leq n$.

c.) $w_{EX}(n) > n - 1$: The construction is a modification of the diagonalization in the proof of Theorem 5.3, b.), where n is replaced by $n - 1$. The point is that we strengthen the update rule for f such that if $f(r)$ is set from 0 to 1 at stage $s + 1$, then we *reset* $f(r') = 0$ for all $r' > r$.

It is still the case that $f \in REC$ and f is not EX-inferred by any M_i, with additional input e. Let $x_{i,s}$ denote the value of x_i at the end of stage $s+1$. We define a set T as follows:

$$\begin{aligned} T = \;& \{f_s \upharpoonright s : s \geq 0\} \\ & \cup \{\sigma \in \{0,1\}^* : (\exists i, s)[|\sigma| = s \;\wedge\; x_{i,s} < s \;\wedge\; \sigma = (f_s \upharpoonright x_{i,s}) \star 1 \star 0^{s-(x_{i,s}+1)}]\}. \end{aligned}$$

Clearly T is uniformly recursive and every initial segment of f belongs to T. Also, by the update rule for the x_i's, $|T \cap \{0,1\}^s| \leq n$. It remains to verify that T is a tree. This is done by induction on s. In the inductive step we have to show that the predecessor of every $\sigma \in T$ of length $s > 0$ belongs to T. This is easy to see if no i is selected at stage $s + 1$. If some i is selected, then, using the new reset rule, $(f_{s-1} \upharpoonright x_{i,s-1}) \star 1 \star 0^{s-x_{i,s-1}} \in T$ is an initial segment of f_s and $x_{j,s} > s + 1$ for all j with $x_{j,s-1} \geq x_{i,s-1}$. Thus, also in this case the predecessor of every $\sigma \in T \cap \{0,1\}^s$ belongs to T. ∎

Remark: One obtains more general classes by considering (m, n)-verboseness operators, see [4, 5, 6]. The corresponding inference notions can be studied along the lines of Sections 3, 4 above.

We now present an application for learning when an upper bound of the descriptional complexity of f is given as additional information. The following considerations hold for our arbitrary acceptable numbering φ; though usually these notions are considered only for "optimal numberings" or "Kolmogorov numberings" [15, 30]. Let $\lg(i) = \lfloor \log_2(i+1) \rfloor$ denote the size of the number i, i.e., the number of bits in the i-th binary string. The *descriptional complexity* $C(\sigma)$ of a string $\sigma \in \{0,1\}^n$ is defined as

$$C(\sigma) = \lg(\min\{i : \varphi_i(n) = \sigma\}).$$

Thus $C(\sigma)$ is just the well-known (length conditional) Kolmogorov complexity of σ with respect to φ. See [30] for background information.
The descriptional complexity $C(f)$ of $f \in REC_{0,1}$ is defined as

$$C(f) = \lg(\min\{i : \varphi_i = f\}).$$

Finally, we define the *weak descriptional complexity* $C'(f)$ of f as

$$C'(f) := \sup\{C(f \restriction n) : n \geq 0\}.$$

Note that there is a recursive function t such that $C'(f) \leq t(C(f))$ for all $f \in REC_{0,1}$. For optimal Gödelnumberings one has $t(e) = e + O(1)$. Since there are less than 2^c functions with $C'(f) < c$, $C'(f)$ indeed measures, in some sense, bits of information of f, as Chaitin [10, Section 4] pointed out. He called $C'(f)$ the "Loveland information measure" and proved that $C'(f)$ can be much smaller than $C(f)$. If $f \in REC_{0,1}$, then $C'(f)$ is finite. The converse appears in a paper of Loveland [31] where it is credited to A. R. Meyer. Actually, as was noted in [21], Meyer's result is roughly equivalent to the fact that trees of bounded width have only recursive branches.

Freivalds and Wiehagen [16] proved that $REC_{0,1}$ is EX-learnable if an upper bound of $C(f)$ is given as additional information for $f \in REC_{0,1}$. In contrast we show that upper bounds of $C'(f)$ do not provide sufficient information to learn all $f \in REC_{0,1}$. This follows as a corollary of Theorem 5.6.

Corollary 5.7 *For all $k \geq 1$, $REC_{0,1}$ is not $BC[k]$-learnable if an upper bound for $C'(f)$ is given as additional information for $f \in REC_{0,1}$.*

Proof: Define a recursive function g such that $\varphi_{g(e,j)}(n)$ is the j-th string σ of length n which appears in W_e (i.e., there is an s such that $\sigma \in W_{e,s}$ and $|\{\tau \in \{0,1\}^n : (\exists t)[\langle \tau, t\rangle < \langle \sigma, s\rangle \wedge\ \tau \in W_{e,t}\}| = j-1$) and is undefined if σ does not exist.

Suppose for a contradiction that there is a team of k IIMs which BC-infers every $f \in REC_{0,1}$ given an upper bound of $C'(f)$ as additional information. Let $h(e) = \max\{g(e,j) : 1 \leq j \leq k+1\}$. If e is a Σ_1-index of a tree T with $w(T) \leq k+1$ and $f \in [T]$, then for each n there is $j, 1 \leq j \leq k+1$, such that $f \restriction n = \varphi_{g(e,j)}(n)$. Thus, $C'(f) \leq h(e)$ and one of the team members BC-infers f from additional information $h(e)$. Since $h \in REC$ we obtain a team of k machines which BC-infers every $f \in REC_{0,1}$ from any Σ_1-index of a tree T of width at most $k+1$ which has f as a branch. This contradicts $w^{re}_{BC}(k+1) > k$ which was shown in Theorem 5.6. ∎

5.3 Trees of Bounded Rank

A larger class of trees is obtained if we consider finite rank instead of finite width.

Definition 5.8 $B_n = \{0,1\}^{\leq n}$ is the full binary tree of depth n. A mapping $g : B_n \to T$ is an embedding of B_n into T if

$$(\forall \sigma)[|\sigma| < n \to [g(\sigma \star 0) \succeq g(\sigma) \star 0\ \wedge\ g(\sigma \star 1) \succeq g(\sigma) \star 1]].$$

$rk(T)$, the *rank* of T, is the supremum of all n such that B_n is embeddable into T.

If an r.e. tree $T \subseteq \{0,1\}^*$ has finite rank, then every branch of T is recursive (see [21, 26]). We consider both r.e. and recursive trees of finite rank which are given as additional information to the IIM.

Definition 5.9 Let $rk_{EX}(n)$ denote the least team size k such that there is a team of k IIMs that EX-infers every $f \in REC_{0,1}$ given any Δ_0-index of a recursive tree $T \subseteq \{0,1\}^*$ such that $rk(T) \leq n$ and f is a branch of T. If Σ_1-indices are provided for T, the corresponding team size is denoted by $rk^{re}_{EX}(n)$. The analogous numbers for BC-teams are $rk_{BC}(n)$ and $rk^{re}_{BC}(n)$.

Theorem 5.10 *For $n \geq 0$, $rk_{EX}(n) = rk^{re}_{EX}(n) = rk^{re}_{BC}(n) = n+1$ and $rk_{BC}(n) = \max(1,n)$.*

Proof: a.) The lower bounds for $rk_{EX}(n), rk^{re}_{BC}(n)$ follow from the corresponding lower bounds of Theorem 5.6, since $[w(T) \leq n+1 \Rightarrow rk(T) \leq n]$.

If f is a branch of T, let $rk(T,f) = \sup\{rk(T[\sigma]) : \sigma \prec f\}$. It is shown in [21] that given k, σ and a Σ_1-index of T with $rk(T[\sigma]) = rk(T,f) = k \ \wedge \ \sigma \prec f$ we can uniformly compute an index of f. Hence, for the upper bounds we can argue as in the proof of Theorem 5.6. Note that we have $n+1$ possible values for k (including $k=0$); thus $n+1$ team members suffice.

b.) For the upper bound $rk_{BC}(n) \leq \max(1,n)$ it suffices to show that $rk_{BC}(1) = 1$. Then we apply the argument of a.) above and note that the cases $k = 0,1$ can be handled by a single IIM. Thus we can save one team member and therefore n team members are enough for $n \geq 1$.

Given a Δ_0-index of a tree $T \subseteq \{0,1\}^*$, $rk(T) \leq 1$, such that f is a branch of T, the BC-algorithm works as follows:

On input $\sigma = (f(0), \ldots, f(n))$ it outputs a program e_n such that:

$\varphi_{e_n}(x) = \tau(x)$ if there is $\tau \in T$, $\sigma \preceq \tau$ such that either τ is the only extension of σ in T with $|\tau| = x+1$, or $|\tau| > x+1$ and $\tau \star 0, \tau \star 1$ both belong to T.

Since $rk(T) \leq 1$, either there is $\sigma_0 \prec f$ such that T has no branching node τ with $\sigma_0 \preceq \tau$, or for every $\sigma \prec f$ there is $\tau \succ \sigma$ such that $\tau \star 0, \tau \star 1 \in T$. In the latter case, all such τ must be an initial segment of f. (Otherwise, B_2 is embeddable in T.) Thus, in both cases $\varphi_{e_n} = f$ for almost all n.

c.) Clearly $rk_{BC}(0) = 1$. For $n \geq 1$ and the lower bound $rk_{BC}(n) \geq n$, we add two features to the diagonalization in the proof of Theorem 4.5. First, the reset rule which we already used in the proof of Theorem 5.6. Second, an additional restriction of diagonalization points. In the original construction all $r > x_i$ were available to diagonalize M_i. This time we may, in the course of the construction, exclude certain points, e.g., if some j with $x_j > x_i$ is selected at stage $s+1$, then all r with $x_j < r \leq s$ are henceforth excluded for diagonalizing M_i. We use an additional set variable L_i to record the excluded points. These restrictions are needed for the construction of a recursive tree of rank at most n which contains f as a branch. They may delay the diagonalization process, but it still goes through.

Now we turn to the formal details. Suppose for a contradiction that the team $M_1, \ldots, M_{n-1}$ BC-infers every $f \in REC_{0,1}$ given Δ_0-indices of trees of rank at most n as additional information. We construct a function $f \in REC_{0,1}$ and a Δ_0-index e of a recursive tree T, $rk(T) \leq n$ such that $f \in [T]$ but f is not BC-inferred by any M_i with additional information e. Since the construction of T will be uniform, we may assume by the recursion theorem that e is given in advance.

Construction:
Stage 0: Initialize $q = (1, 2, \ldots, n-1)$. Let $f = \lambda x.\ 0$. Let $x_i = 0$; $L_i = \emptyset$ for $i = 1, \ldots, n-1$.
Stage $s+1$: If there is an i for which there exists r such that

$$r \notin L_i \ \wedge\ x_i < r \leq s \ \wedge\ f(r) = 0 \ \wedge\ \varphi_{c,s}(r) = 0 \ \text{ for } c = M_i(e, f \restriction r),$$

then select that i which appears in the leftmost position in q, say $i = a_k$.
Update $f(r) = 1$ and reset $f(r') = 0$ for all $r' > r$.
Let $L_{a_j} = L_{a_j} \cup \{x : x_i < x \leq s\}$ for $1 \leq j < k$.
Let $x_{a_j} = 2s$ for $k \leq j \leq n-1$.
Move i to the rear of q, i.e., let $q = (a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_{n-1}, a_k)$.
End of construction.

Definition of T:
Let $f_s, x_{i,s}, L_{i,s}$ denote the values of f, x_i, L_i at the end of stage $s+1$.

$$\begin{aligned} T = \ & \{f_s \restriction s : s \geq 0\} \\ & \cup \{\sigma \in \{0,1\}^* : (\exists i, r, s)[|\sigma| = s \ \wedge\ x_{i,s} < r \leq s \ \wedge\ r \notin L_{i,s} \\ & \quad \wedge\ \sigma = (f_s \restriction r) \star 1 \star 0^{s-r+1)}]\}. \end{aligned}$$

Clearly T is uniformly recursive and $f \in [T]$. It is verified by induction on s that T is a tree. If i acts at stage $s+1$ and sets $f(r) = 1$, then f_s extends $(f_{s-1} \restriction r) \star 1 \star 0^{s-r}$ for some $r \notin L_{i,s-1}$. Also, $[r, s] \subseteq L_{j,s}$ for all j with $x_{j,s} \leq s$ and therefore $f_s \restriction r' = f_{s-1} \restriction r'$ for all $r' \leq s$ with $r' \notin L_{j,s}$.

$rk(T) \leq n$: Suppose for a contradiction that g is an embedding of B_{n+1} into T. Let $\tau_0 = g(\lambda)$, $\tau_j = g(0^j)$ for $j = 1, \ldots n-1$. Then $\tau_j \star 0 \preceq \tau_{j+1}$ for $j = 0, \ldots, n-2$. There must be a stage t_j where $\tau_j \preceq f_{t_j}$ and $f(|\tau_j|)$ is set to 1. (Otherwise B_1 is not embeddable in the subtree $T[\tau_j \star 1]$.) It follows that $t_{j+1} < t_j$ for $0 \leq j < n-1$, since $f_t \restriction (|\tau_j| + 1) \neq \tau_j \star 0$ for all $t \geq t_j$. Let i_j denote the i which is selected at stage t_j. Then $x_{i_j,t} > t_j$ for all $t \geq t_j$. Thus all i_j's are pairwise distinct. This contradicts the fact that there are at most $n-1$ different i_j's.
None of the team members infers f from additional information e: Let $(a_1, \ldots, a_k)$, $k \geq 0$, denote the maximal initial segment of q which stays almost always constant, say from stage s_0 onwards. If $k = n$, then there are only finitely many stages where some i is selected and f changes only finitely often. Clearly, in this case none of the machines infers f.

If $k < n$ then for each $i \notin \{a_1, \ldots, a_k\}$ there are infinitely many stages $s+1 > s_0$ where $i = a_{k+1}$ and i is selected. This makes the guess of $M_i(e, f \restriction r)$ incorrect for some r with $x_{i,s-1} \leq r \leq s$. Since x_i grows unbounded, $M_i(e, f)$ infinitely often outputs an incorrect guess.

Suppose for a contradiction that $M_i(e, f)$ BC-infers f for some $i \in \{a_1, \ldots, a_k\}$. Then there is $s_1 > 2s_0 \geq x_i$ such that $\varphi_{M_i(e, f \restriction t)}$ is an index of f for all $t \geq s_1$. Let $s_2 + 1 > s_1$ be a stage where some j with $j = a_{k+1}$ acts. Then $x_{a_{k'},t} \geq 2s > s_2 + 1$ for $k' \geq k+1$ and $t \geq s_2$. Thus, $[s_2 + 1, 2s_2) \cap L_{i,t} = \emptyset$ and $f(s_2 + 1) = 0$. Choose $s_3 > s_2$ such that $\varphi_{M_i(e, f \restriction (s_2+1)), s_3}(s_2 + 1) = 0$. Then i satisfies the condition in stage $s_3 + 1$ and therefore some $l \leq k$ is selected, a contradiction. ∎

By adapting Proposition 4.3 to our new inference notions we obtain that inference with probability p implies team inference with team size $\lfloor 1/p \rfloor$. And team inference with size k implies probabilistic inference with probability $1/k$.

Hence as a corollary of our results on team inference we obtain the desired results on probabilistic inference. This is depicted in the following table where the maximal probabilities p are given such that $REC_{0,1}$ is inferable w.r.t. $EX_{prob}(p)$ and $BC_{prob}(p)$ from additional information.

Additional information	$REC_{0,1} \in EX_{prob}(p)$	$REC_{0,1} \in BC_{prob}(p)$
(m, n)-comp., $m \leq n/2$	0	0
(m, n)-comp., $m > n/2$	$1/(n-m+1)$	$1/(n-m+1)$
T rec., $(\Delta T) \leq n$	$1/(n+1)$	$1/\lceil \frac{n+1}{2} \rceil$
T rec., $width(T) \leq n$	$1/n$	1
T r.e., $width(T) \leq n$	$1/n$	$1/n$
T rec., $rank(T) \leq n$	$1/(n+1)$	$1/\max(1, n)$
T r.e., $rank(T) \leq n$	$1/(n+1)$	$1/(n+1)$

6 Conclusion and Future Work

We believe the present paper provides hope for escaping from the dilemma in computational learning theory (as well as in work with real robots [8]) that learning is too unsolvable or infeasible. We have provided above some reasonable forms of additional information that yield at least slightly positive *solvability* results.

Future work could investigate improved forms of practically available additional information toward finding increasingly useful, solvable and feasible learnability.

We intend to consider, for example, the learning of useful programs for maps, including route finding programs [33], motivated by robot navigation problems. As in [12], we would model the spaces to be navigated as graphs with vertices representing locally distinct places [24, 25, 29] and with edges representing conduits between them. We plan to consider, as *natural* additional information, bird's eye views, aerial shots, or satellite photos, graph theoretically modeled as (possibly noisy) homomorphic images of the maps to be learned, i.e., as (approximate) copies of the maps with some vertices coalesced. This approach would be complementary to that in [20]. Our work in the present paper suggests, for example, using homomorphic images which limit, in each of various regions, how many vertices from the map are coalesced. In animal learning of spatial routes to goals, the animals attend to global, macroscopic shape information before local clues (see, for example, [11, 17, 32]). Homomorphic image is also a good first approximation to global, macroscopic shape information.

References

[1] A. Amir, W. I. Gasarch. Polynomial terse sets. *Information and Computation*, 77:37–56, 1988.

[2] D. Angluin, W. I. Gasarch, C. H. Smith. Training sequences. *Theoretical Computer Science*, 66:255–272, 1989.

[3] G. Baliga, J. Case. Learning with higher order additional information. In *Proceedings 4th AII'94*, pp. 64–75, Lecture Notes in Computer Science 872, Springer-Verlag, Berlin, 1994.

[4] R. Beigel, W. I. Gasarch, J. Gill, J. C. Owings, Jr. Terse, superterse, and verbose sets. *Information and Computation*, 103:68–85, 1993.

[5] R. Beigel, M. Kummer, F. Stephan. Quantifying the amount of verboseness. *Information and Computation*, 118:73–90, 1995.

[6] R. Beigel, M. Kummer, F. Stephan. Approximable sets. *Information and Computation*, 120:304–314, 1995.

[7] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.

[8] R. Brooks, M. Mataric. Real robots, real learning problems. In *Robot Learning (Edited by J. Connell and S. Mahadevan)*, pp. 193–234, Kluwer Academic Publishers, Boston, 1993.

[9] J. Case, C. H. Smith. Identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[10] G. J. Chaitin. Program size, oracles, and the jump operation. *Osaka Journal of Mathematics*, 14:139–149, 1977.

[11] K. Chang. A purely geometric module in the rat's spatial representation. *Cognition*, 23:149–178, 1986.

[12] T. Dean, D. Angluin, K. Bayse, S. Engelson, L. Kaelbling, E. Kokkevis, O. Maron. Inferring finite automata with stochastic output functions and an application to map learning. In *Proceedings Tenth National Conference on Artificial Intelligence AAAI-92*, pp. 208–214, Menlo Park, CA, 1992.

[13] A. N. Degtev. On (m,n)-computable sets. In *Algebraic Systems (Edited by D.I. Moldavanskij)*, Ivanova Gos. Univ. pp. 88–99, 1981. (in Russian) (see MR 86b:03049)

[14] L. Fortnow, W. I. Gasarch, S. Jain, E. Kinber, M. Kummer, S. Kurtz, M. Pleszkoch, T. Slaman, R. Solovay, F. Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.

[15] R. V. Freivalds. Inductive inference of recursive functions: Qualitative theory. In *Baltic Computer Science*, Lecture Notes in Computer Science, Vol. 502, pp. 77–110, Springer-Verlag, Berlin, 1991.

[16] R. V. Freivalds, R. Wiehagen. Inductive inference with additional information. *EIK*, 15:179–185, 1979.

[17] C. Gallistel. *The organization of learning.* MIT Press, Cambridge, MA, 1990.

[18] W. I. Gasarch, C. H. Smith. Learning via queries. *Journal of the ACM*, 39:649–674, 1992.

[19] V. Harizanov, M. Kummer, J. C. Owings, Jr. Frequency computation and the cardinality theorem. *Journal of Symbolic Logic*, 57:677–681, 1992.

[20] A. Hayashi, T. Dean. Locating a mobile robot using local observations and a global satellite map. In *Third International Symposium on Intelligent Control 1988*, pp. 135–140, IEEE, Piscataway, NJ, 1988.

[21] S. Kaufmann, M. Kummer. On a quantitative notion of uniformity. To appear in: *Fundamenta Informaticae.* (An extended abstract appeared in: *Proceedings MFCS'95*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1995.)

[22] E. Kinber. On frequency calculations of general recursive predicates. *Sov. Math. Dokl.*, 13:873–876, 1972.

[23] E. Kinber. *Frequency-computable functions and frequency-enumerable sets.* Candidate Dissertation, Riga, 1975. (in Russian)

[24] B. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978.

[25] B. Kuipers, Y. Byun. A robust, qualitative method for robot spatial reasoning. In *Proceedings Sixth National Conference on Artificial Intelligence AAAI-88*, pp. 774–779, 1988.

[26] M. Kummer. A proof of Beigel's cardinality conjecture. *Journal of Symbolic Logic*, 57:682–687, 1992.

[27] M. Kummer, F. Stephan. Inclusion problems in parallel learning and games. To appear in: *Journal of Computer and System Sciences.* (An extended abstract appeared in: *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory, COLT'94*, pp. 287–298, ACM Press, 1994.)

[28] M. Kummer, F. Stephan. The power of frequency computation. In *Proceedings FCT'95*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1995.

[29] T. Lewitt, D. Lawton, D. Chelberg, P. Nelson. Qualitative landmark-based path planning and following. In *Proceedings Fifth National Conference on Artificial Intelligence AAAI-87*, pp. 689–694, 1987.

[30] M. Li, P. Vitányi. *An Introduction to Kolmogorov complexity and its applications.* Springer-Verlag, Berlin, 1993.

[31] D. W. Loveland. A variant of the Kolmogorov concept of complexity. *Information and Control*, 15:510–526, 1969.

[32] J. Margules, C. Gallistel. Heading in the rat: Determination by environmental shape. *Animal Learning and Behavior*, 16:404–410, 1988.

[33] M. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8:304–312, 1992.

[34] D. McDermott. Robot planning. *AI Magazine*, 13(2):55–79, 1992.

[35] P. Odifreddi. *Classical recursion theory.* North-Holland, Amsterdam, 1989.

[36] D. Osherson, M. Stob, S. Weinstein. *Systems that learn.* MIT Press, Cambridge, MA, 1986.

[37] J. C. Owings, Jr. A cardinality version of Beigel's nonspeedup theorem. *Journal of Symbolic Logic*, 54:761–767, 1989.

[38] L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.

[39] G. F. Rose. An extended notion of computability. In *Abstr. Intern. Congr. for Logic, Meth., and Phil. of Science*, Stanford, California, 1960.

[40] C. H. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.

[41] R. I. Soare. *Recursively enumerable sets and degrees.* Springer-Verlag, Berlin, 1987.

[42] B. A. Trakhtenbrot. On frequency computation of functions. *Algebra i Logika*, 2:25–32, 1963. (in Russian)