On the Relative Sizes of Learnable Sets[*]

Lance Fortnow[†]
Department of
Computer Science
University of Chicago
1100 E. 58th St.
Chicago, IL 60637 USA
`fortnow@cs.uchicago.edu`

Rūsiņš Freivalds[‡]
Institute of Mathematics
and Computer Science
University of Latvia
Raiņa bulvāris 29
LV-1459, Riga, Latvia
`rusins@mii.lu.lv`

William I. Gasarch[§]
Department of
Computer Science
University of Maryland
College Park, MD 20742
`gasarch@cs.umd.edu`

Martin Kummer
Institut für Logik
Komplexität
und Deduktionssysteme
Universität Karlsruhe
76128 Karlsruhe, Germany
`kummer@ira.uka.de`

Stuart A. Kurtz
Department of
Computer Science
University of Chicago
1100 E. 58th St.
Chicago, IL 60637 USA
`stuart@cs.uchicago.edu`

Carl H. Smith[¶]
Department of
Computer Science
University of Maryland
College Park, MD 20742 USA
`smith@cs.umd.edu`

Frank Stephan[‖]
Mathematical Institute of the
University of Heidelberg
Im Neuenheimer Feld 294
69120 Heidelberg, Germany
`fstephan@ira.uka.de`

Abstract

Measure and category (or rather, their recursion-theoretical counterparts) have been used in theoretical computer science to make precise the intuitive notion "for most of the recursive sets." We use the notions of effective measure and category to discuss the relative sizes of inferrible sets, and their complements. We find that inferable sets become large rather quickly in the standard hierarchies of learnability. On the other hand, the complements of the learnable sets are all large.

1 Introduction

Determining the relative size of denumerable sets, and those with cardinality $\aleph_1$, led mathematicians to develop the notions of measure and category [Oxt71]. We investigate an application of measure and category techniques to a branch of learning theory called inductive inference [AS83]. The models of learning used in this field have been inspired by features of human learning.

The goal of this work is to determine the relative sizes of classes of inferable sets of functions. The idea is to determine when, within the well studied hierarchies of identification criteria, the classes of learnable functions become "large."

Every learning algorithm maintains a space of possible solutions, called the *hypothesis space*. Indeed, a large part of of the computational resources invested by a learning algorithm are devoted to the creation and maintenance of the hypothesis space. A hypothesis space that is just large enough to accommodate a "small" learning problem may not be large enough to include at least one correct solution for every possible instance of a larger learning problem. Hence, to learn a "large" class would require a larger hypothesis space than would be required to learn a "small" class. As the hypothesis space grows, so does the amount of resources required to search through it. We find a point where the size of the hypothesis space takes a significant leap from small, with respect to category and/or measure, to large. The fact that such a crossover point exists is not surprising. What is of interest is that we can isolate where the point is.

We are also interested in the sizes of the complements (with respect to the set of recursive functions) of the learnable classes. The idea is that if the complement of a class is small, then the class itself must be significantly

larger than any class with a large complement. It turns out that unless an inference criteria is sufficiently powerful so as to be able to learn all the recursive functions, then the complement of the learnable sets of functions is large. This means that every practical learning system must be very far from general purpose.

The notions of measure and category have been studied within the context of theoretical computer science. Mehlhorn [Mel73] and Lutz [Lut92] used constructive notions of category and measure to study subrecursive degree structures. Ben-David and Gurvits [BG95] have begun an investigation that relates the VC-dimension of a set of reals with its measure. This may be useful in PAC-learning since the VC-dimension plays a large role there [BEHW89].

2   Technical Preliminaries

We use the following notation throughout this paper.

Notation 2.1

1. The natural numbers are denoted by $\mathbb{N}$. The rational numbers are denoted by $\mathcal{Q}$. The positive rationals are denoted by $\mathcal{Q}^+$.

2. For strings σ and τ, $\sigma \sqsubseteq \tau$ denotes the situation where σ is a prefix of τ. If σ is a proper prefix of τ, we will write $\sigma \sqsubset \tau$. If f is a 0-1 valued function then $\sigma \sqsubset f$ means that σ is an initial segment of f.

3. $\{0,1\}^*$ is the set of all finite sequences of 0's and 1's.

4. $\{0,1\}^\omega$ is the set of all infinite sequences of 0's and 1's.

5. $REC_{0,1}$ is the set of all 0-1 valued recursive functions. (For a formal treatment of recursive functions see [MY78, Rog67, Smi94, Soa87].)

6. If m is a partial function then $m(x) \downarrow$ means that m is defined on arguments x.

2.1 Measure and Category

We will be looking at the size of subsets of $REC_{0,1}$. To do so we identify a function $f : \mathbb{N} \to \{0, 1\}$ with the sequence $f(0)f(1)f(2)\cdots$. Conversely, every element of $\{0, 1\}^\omega$ is associated to a 0-1 valued function, though it need not be recursive.

We now describe two ways to view the size of subsets of $\{0, 1\}^\omega$ and hence the size of subsets of $REC_{0,1}$. In both cases we will end up with a notion of 'small.'

The idea of *measure* comes from [Bor05, Bor14], see [Oxt71]. It is more convenient for us to use the martingale functions [Sch71, Lut92, Lut93]. Intuitively, a martingale m is a betting strategy. A player starts with capital $m(\epsilon)$ and bets on the successive values of a sequence of bits. After he has seen the initial segment σ his capital is $m(\sigma)$ and he bets $m(\sigma b)/2$ that the next bit is b for $b = 0, 1$. In the following definition we require that the wages must sum up to the current capital $m(\sigma)$. If the outcome is b, he receives twice his wager on b and looses his wager on $1 - b$. His capital after σb is therefore $m(\sigma b)$, as intended. This gambling terminology is used in our proofs.

Definition 2.2 A (partial) function $m : \{0, 1\}^\star \to \mathcal{Q}^+ \cup \{0\}$ is a *martingale* if for all σ, if $m(\sigma 0) \downarrow$ or $m(\sigma 1) \downarrow$ then

1. $m(\sigma) \downarrow$, $m(\sigma 0) \downarrow$ and $m(\sigma 1) \downarrow$, and

2. $m(\sigma) = \frac{m(\sigma 0) + m(\sigma 1)}{2}$.

Definition 2.3 Let $f \in \{0, 1\}^\omega$. We say m *wins on* f if

1. For all n, $m(f(0)\cdots f(n)) \downarrow$, and

2. $\limsup_{n \to \infty} m(f(0)\cdots f(n)) = \infty$.

We say m *loses on* f if m does not win on f. If $S \subseteq \{0, 1\}^\omega$ then m *wins on* S if m wins on every $f \in S$.

Let $S \subseteq \{0, 1\}^\omega$. Suppose there exists a martingale m that wins on S. Then every $A \in S$ is somewhat predictable; intuitively, S is small. The next definition is based on this intuition.

4

Definition 2.4 Let $S \subseteq \{0,1\}^\omega$. *S has measure zero* if there is a martingale m that wins on S. *S* has *(partial) recursive measure* 0 if there exists a (partial) recursive martingale m that wins on S. If m is partial recursive then m has to be defined on all initial segments of elements of S. We denote S having (partial) recursive measure 0 by $(\mu_P(S) = 0)$ $\mu_R(S) = 0$. (Schnorr [Sch71] showed that the above definition of measure 0 is equivalent to the classical one, and related martingales to Martin-Löf randomness.)

Terwijn [Ter95] pointed out that partial recursive measure zero sets are not closed under union (see the remark after Theorem 5.1) and therefore the term "measure" is somewhat euphemistic. However, our intuition as to why partial recursive measure 0 is "small" is still valid.

Next we review an effective notion of category [Kur58, Oxt71]. We will give two definitions of effectively meager. Lisagor [Lis81] (also see Fenner [Fen91, Theorem 3.7]) showed that they are equivalent. We first need to define the notion of a Banach-Mazur game.

Definition 2.5 Let $S \subseteq \{0,1\}^\omega$. The *Banach-Mazur game associated with* S is defined as follows. On the first move player I chooses $\sigma_1 \in \{0,1\}^*$ and player II chooses τ_1 such that $\sigma_1 \sqsubseteq \tau_1$. In all subsequent moves the player picks a proper extension of the current string. If the final infinite string is in S then player I wins, else player II wins.

Definition 2.6 Let $S \subseteq \{0,1\}^\omega$. We refer to the Banach-Mazur game associated with S throughout this definition. A *(recursive) strategy* is a (recursive) function $strat$ from $\{0,1\}^*$ to $\{0,1\}^*$ such that $\sigma \sqsubset strat(\sigma)$. $strat$ is a *winning strategy for player I* if the following sequence of plays leads to player I winning: player I plays $strat(\epsilon)$, player II plays τ_1, player I plays $strat(\tau_1)$, player II plays τ_2, player I plays $strat(\tau_2)$ .... Note that player II's moves are not constrained except that they have to follow the rules of the game and thus be extensions of the current string. A *winning strategy for player II* is defined similarly.

Let $S \subseteq \{0,1\}^\omega$. Imagine that player II has a winning strategy for the Banach-Mazur game associated with S. Then player II can (in the long run) force the string being constructed to not be in S. Hence S is 'small.' The next definition is based on this intuition.

Definition 2.7 A set $C \subseteq \{0,1\}^\omega$ is *meager* iff there is a winning strategy for player II in the Banach-Mazur game associated to C. A set $C \subseteq REC_{0,1}$ is *effectively meager* iff there is a recursive winning strategy for player II in the Banach-Mazur game associated to C. Note that this strategy beats *any* strategy for player I, even nonrecursive ones.

Definition 2.8 (Due to Mehlhorn [Mel73]) A set $C \subseteq REC_{0,1}$ is *effectively meager* iff there is a sequence $\{h_k\}_{k \in \omega}$ of uniformly recursive functions $h_k : \{0,1\}^* \to \{0,1\}^*$ such that $\sigma \sqsubseteq h_k(\sigma)$ for all k, σ, and for every $f \in C$ there is k such that $h_k(\sigma) \not\sqsubseteq f$ for all σ. (This formalizes that C is contained in an effective union of effectively nowhere dense sets.)

2.2 Learning Theory

The basic model of learning used in this paper was first used by philosophers interested in modeling the scientific method [Put75]. The model was studied formally by linguists who were interested in the learning of natural languages [Gol67] and cast recursion theoretically in [BB75]. Since we will be considering learning recursive functions we need a fixed acceptable programming system (see [MY78, Rog67, Smi94, Soa87].)

Notation 2.9 Throughout this paper $M_0, M_1, \ldots$ is a standard list of all Turing machines, $M_0^{()}, M_1^{()}, \ldots$ is a standard list of all oracle Turing machines. $\varphi_0, \varphi_1, \ldots$ is the acceptable programming system, obtained by letting φ_e be the partial recursive function computed by M_e.

Convention 2.10 All the recursive functions we deal with will be in $REC_{0,1}$, hence we will assume that all the φ_e are partial 0-1 valued functions.

The following definitions are from [CS83]. Usually these definitions yield subsets of the recursive functions, however we will take them to yield subsets of $REC_{0,1}$ for this paper.

Definition 2.11 An *inductive inference machine* (IIM) M is a total Turing machine. We informally interpret M to be trying to learn a recursive function f by viewing M as taking as input the values $f(0), f(1), \ldots$ (one value at a

time) and producing output (from time to time) in the form of a program intended to compute f. If almost all the programs are the same, and compute f, then we say that *M EX-identifies f* or *M learns f in the limit*. *EX* stands for EXplains as we are thinking of a program for f as an explanation for f's behaviour. If almost all the programs compute f, but are not necessarily the same, then *M BC-identifies f*. *BC* stands for Behaviorially Correct since we only insist that the programs output behave the same as f does. Formally, M computes a total function from $\{0,1\}^*$ to $\mathbb{N}$. The input is an initial segment of the function to be inferred, and the output is the current guess as to an index of that function. The indices output by IIMs are relative to the acceptable programming system $\{\varphi_e\}_{e=0}^{\infty}$ specified in Notation 2.9.

Convention 2.12 If the output of an IIM is 0 then we interpret this as meaning 'no guess at this time'.

Definition 2.13 Let $S \subseteq REC_{0,1}$. Then $S \in EX$ (BC) if there exists an IIM M such that for all $f \in S$, M *EX*-identifies f (*BC*-identifies f).

Example 2.14 Let
$$S_0 = \{\sigma 0^\omega : \sigma \in \{0,1\}^*\}$$
and
$$S_1 = \{f : 1^e 0 \sqsubset f \wedge \varphi_e = f \wedge f \in REC_{0,1}\}.$$
Note that $S_0 \in EX$ and $S_1 \in EX$. Techniques of Blum and Blum [BB75], or Case and Smith [CS83], can be used to show that $S_0 \cup S_1 \notin EX$.

Definition 2.15 The set S_1 in Example 2.14 is called *the set of self-describing functions*.

We consider several restrictions and enhancements to EX. One restriction of EX is to bound the number of times an IIM outputs a program that is different from the most recently produced program. When this happens, we say the M has made a *mind change*.

Definition 2.16 $S \in EX_n$ if there exists an IIM M such that for all $f \in S$, M EX-identifies f, and changes its mind at most n times.

Example 2.17 Note that the set S_1 of self-describing functions is in EX_0.

Another restriction on learning is to insist that the guesses made are total.

Definition 2.18 $S \in PEX$ if $S \in EX$ via an IIM that, on any input σ, outputs an index of a total function. (The P in PEX stands for "Popperian" since this definition was intended to model Karl Popper's philosophy of science. See [CS83] for further EXplanation.)

One way to expand the class of functions being learned is to allow the final conjecture output to be incorrect on some number of points.

Definition 2.19 If f and g are partial recursive functions and $k \in \mathbb{N}$ then $f =^k g$ ($f =^* g$) means that f and g agree on all but at most k points (on all but a finite number of points). If f and g agree on x then it is possible that $f(x) \uparrow$ and $g(x) \uparrow$. If they disagree on x then it is possible that $f(x) \downarrow$ and $g(x) \uparrow$.

Definition 2.20 $S \in EX^k$ ($S \in EX^*$) if there exists an IIM M such that for all $f \in S$, when M is run on initial segments of f, (1) almost all the programs output are the same, say e, and (2) $\varphi_e =^k f$ ($\varphi_e =^* f$). Note that the final conjecture φ_e may diverge on the points where it disagrees with f.

Another way to expand the class of functions being learned is to have more algorithms involved in the learning.

Definition 2.21 Let m, n be such that $1 \leq m \leq n$. A set of recursive functions S is in $[m, n]EX$ (concept from [Smi82, OSW86], notation from [PS88]) if there exist n IIMs $M_1, M_2, \ldots, M_n$ such that for every $f \in S$ there exist $i_1, \ldots, i_m$, $1 \leq i_1 < \cdots < i_m \leq n$, such that $M_{i_1}, \ldots, M_{i_m}$ all EX-infer f. If $m = 1$ then in the literature this is referred to as inferring S by a *team of n IIMs*.

Team learning has been shown to be equivalent to a form of learning resembling the well known "process of elimination" [FKS94] and strongly related to probabilistic learning [PS88, Pit89]. The class $[m,n]EX_0$ has been investigated extensively in [DPVW91, DKV92, DKV93, Vel89, JS90]. We can combine several of the modifications mentioned above.

Definition 2.22 For $\mathcal{I} \in \{PEX, EX, BC\}$, $a, b \in \mathbb{N} - \{0\}$ with $a \leq b$, and $c, d \in \mathbb{N} \cup \{*\}$ one can define $[a,b]\mathcal{I}_c^d$. If $c = *$ then we are allowing unbounded mindchanges which is the standard definition of PEX, EX, and BC. If $c \neq *$ then $BC_c = EX_c$.

The following lemma is well-known [CS83, Smi82]:

Lemma 2.23 *If $\mathcal{I} \in \{PEX, EX\}$, $a, b \in \mathbb{N} - \{0\}$ with $a \leq b$, and $c, d \in \mathbb{N} \cup \{*\}$, then $REC_{0,1} \notin [a,b]\mathcal{I}_c^d$. However, $REC_{0,1} \in BC^*$.*

3   Categoricity of Inferable Classes

In this section we consider a class to be small if it is effectively meager. We show that every set in PEX is small, but that there exists a set in EX_0 that is not small. This shows that the constraint of outputing only indices for total functions is very restrictive: all sets in PEX are small, while the weakest natural inference class that does not have this constraint, EX_0, can learn a set that is not small. Since virtually every inference class is either a subset of PEX or a superset of EX_0 the results here settle virtually all open questions that could be raised.

Theorem 3.1 *Every set in PEX is effectively meager.*

Proof: Suppose $S \in PEX$ via IIM M. We describe a recursive winning strategy *strat* for player II of the Banach-Mazur game for S. Suppose that it is player II's turn to play and that σ is the finite function that has been determined by the game so far. Let x be the least value not in the domain of σ. Let $e = M(\sigma)$. Since $S \in PEX$, φ_e is a recursive function. Player II plays $\sigma \cdot (1 - \varphi_e(x))$. (By Convention 2.10 $\varphi_e(x) \in \{0, 1\}$.) Clearly, the function constructed is not in S.   ∎

Theorem 3.2 *There exists a set $S \in EX_0$ that is not effectively meager.*

Proof: Let $S = S_1$, the set of self describing functions (see Definition 2.15). We show that player II cannot have a recursive winning strategy for the Banach-Mazur game associated to S. Let *strat* be a recursive strategy for player II. We describe a recursive winning strategy that player I can use to defeat player II. The strategy implicitly uses the recursion theorem. Player I first plays 1^e0 where e is the index of the recursive set that is the limit of the sequence defined by

$$
\begin{aligned}
\sigma_1 &= 1^e0; \\
\sigma_{2i} &= strat(\sigma_{2i-1}); \\
\sigma_{2i+1} &= \sigma_{2i}0.
\end{aligned}
$$

From then on player I will always extend the current string by 0. The set f produced is the recursive set described by φ_e. Hence $f \in S$, so *strat* is not a winning strategy. Since this proof was for any recursive *strat*, player II cannot have a recursive winning strategy. ∎

4 Recursive Measure

In this section we consider a class S to be small if $\mu_R(S) = 0$. The behaviour of natural inference classes with respect to this notion of size is identical to that of effective meagerness: every set in PEX is small, but that there exists a set in EX_0 that is not small. Hence again we see that the constraint of outputing only indices for total functions is very restrictive. Since virtually every inference class is either a subset of PEX or a superset of EX_0 the results here settle virtually all open questions that could be raised.

To show that every set in PEX has recursive measure 0 we state a more general theorem. As Terwijn [Ter95] pointed out, this follows from [Lut92, Lemma 3.10] where it is stated in terms of density systems; for the convenience of the reader we present a self-contained proof.

Definition 4.1 Let $S \subseteq REC_{0,1}$. S is *uniformly r.e.* if there exists a recursive function g such that $S = \{\lambda x.g(i, x) : i \geq 0\}$.

Theorem 4.2 *If S is a uniformly r.e. subset of $REC_{0,1}$ then $\mu_R(S) = 0$.*

Proof: Since S is uniformly r.e. there exists a recursive function g such that $S = \{\lambda x.g(i,x) : i \geq 0\}$.

Let $m(\lambda) = 1$. Assume $m(\sigma)$ is defined. We define $m(\sigma 0)$ and $m(\sigma 1)$ simultaneously. Find the minimal $i \leq |\sigma|$ such that for all x, $|x| \leq |\sigma|$, $g(i,x) = \sigma(x)$. Let $g(i, |\sigma|) = b$. We bet on b: Let $m(\sigma b) = \frac{3}{2}m(\sigma)$ and $m(\sigma(1-b)) = \frac{1}{2}m(\sigma)$.

Let f be the recursive function being fed to the martingale. Let e be the minimal i such that $(\forall x)[g(i,x) = f(x)]$. The i picked in determining $m(|\sigma|)$ will reach a limit of e. Once it reaches that limit the martingale always bets on the correct answer. Hence the theorem is established.   ∎

Corollary 4.3 *If $S \in PEX$ then $\mu_R(S) = 0$.*

Proof: Let $S \in PEX$ via M. Note that $S \subseteq S' = \{\varphi_{M(\sigma)} : \sigma \in \{0,1\}^*\}$. Since $S \in PEX$ we know that M only outputs indices of total machines, so $S' \subseteq REC_{0,1}$. S' is easily seen to be uniformly r.e., hence, by Theorem 4.2, $\mu_R(S') = 0$. Since $S \subseteq S'$, $\mu_R(S) = 0$.   ∎

Theorem 4.4 *There exists $S \in EX_0$ such that $\mu_R(S) \neq 0$.*

Proof: Let $S = S_1$, the set of self describing functions (see Definition 2.15). We show that if m is any recursive martingale then there exists an $f \in S$ such that m loses on f. Let m be a martingale. Recall that, for all σ, either $m(\sigma 0) \leq m(\sigma)$ or $m(\sigma 1) < m(\sigma)$.

We construct f by an initial segment argument. At the end of stage s we have $\sigma_s \in \{0,1\}^*$. f will be $\lim_{s \to \infty} \sigma_s$.

BEGIN CONSTRUCTION

Stage 0: $\sigma_0 = 1^e 0$ where e is the index of the function we are constructing (we use the recursion theorem implicitly).

Stage s+1: Let $x = |\sigma_s|$, the least undefined point. Let

$$\sigma_{s+1} = \begin{cases} \sigma_s 0 & \text{if } m(\sigma_s 0) \leq m(\sigma_s); \\ \sigma_s 1 & \text{if } m(\sigma_s 1) < m(\sigma). \end{cases}$$

11

END OF CONSTRUCTION

One can easily show by induction that $(\forall s)[m(\sigma_s) \leq m(\sigma_0)]$. Hence m loses on f. ∎

5 Partial Recursive Measure

In this section we consider a class S to be small if $\mu_P(S) = 0$. We show that every set in EX_0 is small, but that there exists a set in EX_1 that is not small. The question arises as to how much power we must add to EX_0 before we obtain an inference class $\mathcal{I}$ such that there exists $S \in \mathcal{I}$ with S not small. We answer this completely:

1. If $c > \frac{d}{2}$ then every $S \in [c,d]EX_0^*$ is small.

2. If $c \leq \frac{d}{2}$ then there exists $S \in [c,d]EX_0$ which is not small. (This is easy since $EX_1 \subseteq [c,d]EX_0$ in this case.)

In summary, for this notion of size, the step from 0 mindchanges to 1 mindchange is critical. This still holds even if we enhance the EX_0 machine in various ways.

Theorem 5.1 *If $S \in EX_0$ then $\mu_P(S) = 0$.*

Proof: Let $S \in EX_0$ via M. We define a partial recursive martingale m as follows for $\sigma \in \Sigma^*$ and $i \in \Sigma$:

- Let $m(\lambda) = 1$.

- If $M(\sigma) = 0$ then $m(\sigma b) = 1$ for $b = 0, 1$. (No bet is placed.)

- If $M(\sigma)$ outputs a value larger than 0 and $\varphi_{M(\sigma)}(|\sigma|) = b$ then let $m(\sigma b) = 2m(\sigma)$ and $m(\sigma(1-b)) = 0$. (All the money is bet on b.)

We show that, for all $f \in S$, m wins on f. Let $\sigma \sqsubset f$ be the shortest initial segment of f such that $M(\sigma) > 0$. Since $S \in EX_0$ via M we have $\varphi_{M(\sigma)} = f$. Hence all bets placed after σ is observed will win. Hence m wins on f. ∎

Terwijn [Ter95] noted that $S_0 \cup S_1$ is an example of a class S with $\mu_P(S) \neq 0$ and such that S is the union of two (partial recursive) measure zero classes.

We now show that there exists a set $S \in EX_1$ such that $\mu_P(S) = 0$. We need the following technical lemma. This lemma will enable us to extend a string σ by "not too many ones" such that for the resulting $\tau \sqsupseteq \sigma$ the value of $m(\tau)$ is "not too large." The lemma will also be used in Section 6.

Lemma 5.2 *Let m be a partial recursive martingale. Given $\sigma \in \{0,1\}^*$ and $c \in \mathcal{Q}$ such that*

    a. m is defined on all strings in $\{\sigma\eta : |\eta| \leq \frac{m(\sigma)}{c-m(\sigma)} + 1\}$, and

    b. $m(\sigma) < c$,

one can recursively determine a τ such that the following hold.

    1. τ extends σ.

    2. $|\tau| \leq \frac{m(\sigma)}{c-m(\sigma)} + |\sigma|$.

    3. $(\forall \eta)[\sigma \sqsubseteq \eta \sqsubseteq \tau \to m(\eta) \downarrow \leq c]$. ($m(\eta) \downarrow$ is guaranteed by a and 2.)

    4. $m(\tau 0) \downarrow \leq c$ and $m(\tau 1) \downarrow \leq c$. (convergence is guaranteed by a and 2.)

    5. There exists $k \leq \frac{m(\sigma)}{c-m(\sigma)}$ such that $m(\sigma 1^k 0) \downarrow \leq c$. Hence if $c = m(\sigma) + \frac{1}{2^s}$ then $k \leq 2^s m(\sigma)$. (This is the only part we will be using now.)

Proof: Let $\eta_0 = \sigma$ and let

$$\eta_{i+1} = \begin{cases} \eta_i 0 & \text{if } m(\eta_i 1) \downarrow > c; \\ \eta_i 1 & \text{if } m(\eta_i 0) \downarrow > c; \\ \eta_i & \text{otherwise.} \end{cases}$$

We will show that there exists $i \leq \frac{m(\sigma)}{c-m(\sigma)}$ such that $\eta_{i+1} = \eta_i$, hence by a the least such i can be found recursively.

Since m is a martingale $m(\eta_i) = \frac{m(\eta_i 0) + m(\eta_i 1)}{2}$. Using this one can show (by induction) that $(\forall i)[m(\eta_i) \leq m(\sigma)]$. We use this later.

We show that there exists $i \leq \frac{m(\sigma)}{c-m(\sigma)}$ such that $\eta_{i+1} = \eta_i$. Since m maps to positive numbers it suffices to show that if $\eta_{i+1} \neq \eta_i$ then $m(\eta_{i+1}) < m(\eta_i) - (c - m(\sigma))$.

13

If $\eta_{i+1} = \eta_i b$ ($b \in \{0,1\}$) then, by the definition of a martingale,

$$m(\eta_{i+1}) = m(\eta_i b) = 2m(\eta_i) - m(\eta_i(1-b)).$$

By the definition of η_{i+1}, $m(\eta_i(1-b)) > c$. Hence

$$m(\eta_{i+1}) < 2m(\eta_i) - c = m(\eta_i) - c + m(\eta_i) \leq m(\eta_i) - (c - m(\sigma)).$$

(this last inequality came from $m(\eta_i) \leq m(\sigma)$).

Let i be the least number such that $\eta_i = \eta_{i+1}$. Let $\tau = \eta_i$. Items 1 and 2 clearly hold. Since $(\forall i)[m(\eta_i) \leq m(\sigma) < c]$ item 3 holds. Since $m(\tau 0) \leq c$ and $m(\tau 1) \leq c$ item 4 holds.

Let k be the maximal number such that $\sigma 1^k \sqsubseteq \tau$. Clearly $k \leq i \leq \frac{m(\sigma)}{c - m(\sigma)}$. If $\tau = \sigma 1^k$ then, by item 4, $m(\sigma 1^k 0) \leq c$. If $\sigma 1^k 0 \sqsubseteq \tau$ then by item 3 $m(\sigma 1^k 0) \leq c$. Hence, in any case, $m(\sigma 1^k 0) \leq c$. Thus item 5 holds. ∎

Theorem 5.3 *There exists $S \in EX_1$ such that $\mu_P(S) \neq 0$.*

Proof: A string $\sigma = 1^e 0 1^{a_1} 0 1^{a_2} 0 \cdots 0 1^{a_n}$ is said to have *few ones* if $0 \leq a_i \leq 2^{i+2}$ for $i = 1, 2, \ldots, n$. A function has *few ones* iff each of its initial segments has few ones. Let S be

$$\{\varphi_e : \varphi_e \text{ is total}, 1^e 0 \sqsubseteq \varphi_e \text{ and } \varphi_e \text{ has few ones}\} \ \cup$$
$$\{f : f = \sigma 1^\omega \text{ and } \sigma \text{ has few ones}\}.$$

S is EX_1-inferable via the following inference algorithm. On input σ do the following. If $\sigma \sqsupseteq 1^e 0$ has few ones, then guess φ_e. Otherwise let $\tau \sqsubset \sigma$ denote the longest initial segment of σ with few ones and guess $\tau 1^\omega$.

Assume, by way of contradiction, that $\mu_P(S) = 0$ via partial recursive martingale m; without loss of generality assume that $m(1^e 0) = 1$ for all e. We construct, by an initial segment argument, a function $f \in S$ such that m loses on f. At the end of stage s we will have σ_s. The final f will be $\lim_{s \to \infty} \sigma_s$.

BEGIN CONSTRUCTION

Stage 0: $\sigma_0 = 1^e 0$ where e is the index of the function we are constructing (we use the recursion theorem implicitly).

Stage s+1: Inductively assume that

14

1. $\sigma_s = 1^e 0 1^{a_1} 0 \cdots 1^{a_s} 0$,

2. σ_s has few ones, and

3. for all $\eta \sqsubseteq \sigma$, $m(\eta) \leq \sum_{i=0}^{s} 2^{-i}$.

Let $c = m(\sigma_s) + 2^{-s-1}$, $\sigma = \sigma_s$, and $i = \frac{m(\sigma)}{c-m(\sigma)} = m(\sigma)2^{s+1} < 2^{s+2}$. For $\eta \in \{0,1\}^{\leq i+1}$ the string $\sigma\eta$ has few ones so $\sigma\eta 1^\omega \in S$; hence $m(\sigma\eta)$ is defined. We can thus apply Lemma 5.2.5 to σ to obtain $k \leq 2^{s+2}$ such that $m(\sigma 1^k 0) \leq m(\sigma) + 2^{-s-1}$. Let $\sigma_{s+1} = \sigma_s 1^k 0$. It is easy to show that items 1,2, and 3 still hold.

END OF CONSTRUCTION

Clearly, the function constructed is in S. Since $m(\eta) \leq \sum_{i=0}^{n} 2^{-i}$ on all initial segments of f the martingale is bounded on f by 2. Hence the martingale m loses on f. This is a contradiction. ∎

Theorem 5.4 *Let c, d be such that $c > \frac{d}{2}$. If $S \in [c,d]EX_0^*$ then $\mu_P(S) = 0$.*

Proof: Assume $S \in [c,d]EX_0^*$ via IIM's $M_1, \ldots, M_d$. We describe a partial recursive martingale m for S intuitively; we formalize it later. Assume that if σb ($\sigma \in \{0,1\}^*$, $b \in \{0,1\}$) is input then σ is an initial segment of a fixed 0-1 valued function $f \in S$. We define $m(\sigma 0)$ and $m(\sigma 1)$ simultaneously. They are how much we are betting that $f(|\sigma|) = 0$ and $f(|\sigma|) = 1$. Since we can compute $m(\tau)$ for all $\tau \sqsubseteq \sigma$ recursively we assume that we have access to all prior bets and outcomes.

If $|\{i : M_i(\sigma) \neq 0\}| < c$ then we bet on 0 and 1 equally. Once we have σ such that $|\{i : M_i(\sigma) \neq 0\}| \geq c$ we set up the following parameters.

1. e_i is the guess that the M_i makes as to the index of f. Initially it is $M_i(\sigma)$ if it exists, and undefined otherwise. It may be undefined now but become defined later.

2. k is our current guess for the least number such that $\varphi_{e_i} =^k f$ (see Definition 2.19). Initially $k = 0$. This parameter increases when its current guess looks infeasible.

3. E_i is the number of errors φ_{e_i} has made. Initially $E_i = 0$.

15

4. $POSS$ is the set of all i such that we currently think $\varphi_{e_i} =^k f$ is possible. $POSS$ will always be $\{i : E_i \leq k\}$. This set may (1) grow when more e_i's are defined, (2) shrink when a φ_e is seen to be wrong more than $k + 1$ times, and (3) grow when k is increased. This set will always have at least c elements.

5. $CANCEL$ is the set of all i such that φ_{e_i} has been seen to make $k + 1$ errors. It will always be $\{i : E_i \geq k + 1\}$. Indices that have been cancelled may be resurrected if k increases.

On input σ we do the following. Look for $(x, i, b) \in \mathbb{N} \times POSS \times \{0, 1\}$ such that $x \geq |\sigma|$ and $\varphi_{e_i}(x) \downarrow= b$. (Since $c > \frac{d}{2}$ and $|POSS| \geq c$ there exists $i \in POSS$ such that $\varphi_{e_i} =^* f$, hence an (x, i, b) will be found.) We will now plan to (1) bet even money on $\tau 0$ and $\tau 1$ for all τ such that $|\sigma| \leq |\tau| \leq x - 1$, and (2) bet $\frac{3}{4}$ that $f(x) = b$ and $\frac{1}{4}$ that $f(x) = 1 - b$. If we later find out that this prediction is wrong then we will set $E_i = E_i + 1$. If $|E_i| \geq k + 1$ then we cancel e_i and remove i from $POSS$. If $|POSS| < c$ then the value of k was too low so we set $k := k + 1$, $POSS := POSS \cup CANCEL$, and $CANCEL := \emptyset$.

We now present this algorithm formally. Since we will be calling it recursively we make each execution of the algorithm yield several parameters as well as the answer. The parameters are those listed above (e.g. $POSS$ and $CANCEL$) and also a tuple (x, i, b). For some inputs the output will not yield these parameters. In this case we say the parameters are undefined. We will be able to tell that this is the case recursively.

BEGIN ALGORITHM

1. Input$(\sigma 0, \sigma 1)$.

2. If $\sigma = \lambda$ or $|\{i : M_i(\sigma) \neq 0\}| < c$ then set $m(\sigma 0) = m(\sigma 1) = 1$. All other parameters remain undefined.

3. If $|\{i : M_i(\sigma) \neq 0\}| \geq c$ and no proper prefix of σ has this property

then we initialize parameters.

$$
\begin{aligned}
e_i &:= \begin{cases} M_i(\sigma) & \text{if } M_i(\sigma) \neq 0 \\ \text{undefined} & \text{otherwise} \end{cases} \\
k &:= 0 \\
E_i &:= \begin{cases} 0 & \text{if } e_i \text{ is defined} \\ \text{undefined} & \text{otherwise} \end{cases} \\
POSS &:= \{i : E_i \leq k\} \\
CANCEL &:= \emptyset
\end{aligned}
$$

4. (We can assume $|\{i : M_i(\sigma) \neq 0\}| \geq c$ and all parameters have been initialized.) Compute $m(\sigma)$. From this computation we extract two parameters: $POSS$ and (x, i, b) (which might be undefined, but determining this is recursive). If (x, i, b) is defined then x is a number such that we would like to bet that $f(x) = b$, since $\varphi_{e_i}(x) = b$.

5. If (x, i, b) is undefined then find (by dovetailing) a triple $(x, i, b) \in \mathbb{N} \times POSS \times \{0, 1\}$ such that $x \geq |\sigma|$ and $\varphi_{e_i}(x) \downarrow = b$. Define (x, i, b) to be the first such triple found. This value will be used in the next three steps.

6. If $x > |\sigma|$ then set $m(\sigma 0) = m(\sigma 1) = m(\sigma)$. All parameters retain their values. (We are not betting. We are waiting to bet on x later.)

7. If $x = |\sigma|$ then we bet on x: set $m(\sigma b) = \frac{3}{2} m(\sigma)$ and $m(\sigma(1 - b)) = \frac{1}{2} m(\sigma)$. All parameters retain their values. (In the next stage we will see if this bet was correct and take appropriate action.)

8. If $x = |\sigma| - 1$ then do the following.

   (a) If b is not the last bit of σ then set $E_i := E_i + 1$.

   (b) If $E_i > k$ then $POSS := POSS - \{i\}$ and $CANCEL := CANCEL \cup \{i\}$.

   (c) If $|POSS| < c$ then $k := k + 1$, $POSS := POSS \cup CANCEL$, and $CANCEL := \emptyset$.

   (d) Declare (x, i, b) undefined.

   (e) If $(\exists j)[(j \notin CANCEL \cup POSS) \wedge (M_j(\sigma) \neq 0)]$ then for all such j set $e_j := M_j(\sigma)$, $E_j = 0$, and $POSS := POSS \cup \{j\}$.

17

END OF ALGORITHM

We show this martingale wins on every $f \in S$. Let $f \in S$. Let $\sigma \sqsubseteq f$ and let $k', k'' \in \mathbb{N}$ be such that the following hold.

1. Let $e_i = M_i(\sigma)$ if it exists, and undefined otherwise. Let $CORR = \{i : \varphi_{e_i} =^{k'} f\}$. We require $|CORR| \geq c$.

2. For all $i \in CORR$ if $\varphi_{e_i}(x) \neq f(x)$ then $x < |\sigma|$.

3. When $m(\sigma)$ is defined the value of k is k''.

We look at how the martingale behaves on inputs $\tau \sqsubseteq f$ such that $|\tau| > |\sigma|$. If the (x, i, b) picked is such that $i \in CORR$ then the martingale will win. The number of times an $i \notin CORR$ can be picked such that the martingale loses is at most $(d - c) \max\{k', k''\}$. Hence eventually the martingale will always win. $\blacksquare$

6 Complements of Inference Classes are Large

We have been looking at the size (defined in various ways) of sets $S \in \mathcal{I}$ for various inference classes $\mathcal{I}$. Another way to judge the size of S is to look at the size of $REC_{0,1} - S$. In this section we show that if $S \in EX$ (or most of the other inference classes considered in this paper) then $REC_{0,1} - S$ is not small. We begin with some definitions. The only new one is the definition below of *reasonable*.

Definition 6.1 Two strings are *incompatible* iff neither is a prefix of the other.

Definition 6.2 An *infinite binary tree* is a mapping $\mathcal{T}$ from $\{0,1\}^*$ to $\{0,1\}^*$ such that (1) $\sigma \sqsubset \tau$ implies $\mathcal{T}(\sigma) \sqsubset \mathcal{T}(\tau)$, and (2) $\mathcal{T}(\sigma 0)$ and $\mathcal{T}(\sigma 1)$ are incompatible with respect to $\sqsubseteq$. Let $d_0 d_1 \cdots \in \{0,1\}^\omega$. Let $A = \lim_{n \to \infty} \mathcal{T}(d_0 \cdots d_n)$. A is *a branch of* $\mathcal{T}$. A *recursive binary tree* is a tree computed by a recursive function.

Definition 6.3 Let $\mathcal{T}$ be a recursive binary tree and let $S \subseteq REC_{0,1}$. We would like to define the set of branches of $\mathcal{T}$ that are 'guided' by functions in S. Formally let

$$RECB(\mathcal{T}, S) = \{f : (\exists \phi \in S)[f = \lim_{n \to \infty} \mathcal{T}(\phi(0)\phi(1) \cdots \phi(n)]\}.$$

Definition 6.4 An inference class $\mathcal{I}$ is *reasonable* if, for all recursive trees $\mathcal{T}$ and sets $S \subseteq REC_{0,1}$,

$$S \in \mathcal{I} \text{ iff } RECB(\mathcal{T}, S) \in \mathcal{I}.$$

Most inference classes are reasonable. In particular $[a, b]I_c^d$, is reasonable for $\mathcal{I} \in \{EX, BC, PEX\}$ and any choice of $a, b \in \mathbb{N}$ with $a \leq b$ and $c, d \in \mathbb{N} \cup \{*\}$. The inference classes dealt with in the study of learning via queries [GS92], are not reasonable. (See Appendix.)

Definition 6.5 A set $S \subseteq \{0, 1\}^\omega$ is *dense* if for all $\sigma \in \{0, 1\}^*$ there exists $f \in S$ such that $\sigma \sqsubseteq f$.

Lemma 6.6 *Let S be a dense subset of $REC_{0,1}$ such that $\mu_P(S) = 0$ or such that S is effectively meager. Then there exists a recursive binary tree $\mathcal{T}$ such that no branch of $\mathcal{T}$ is in S.*

Proof: There are two cases:

Case 1: $\mu_P(S) = 0$. Let m be a partial recursive martingale for S. Since S is dense, m is total recursive.

We define $\mathcal{T}$ as follows. Let $\mathcal{T}(\lambda) = \lambda$. Assume $\mathcal{T}(\sigma)$ is already defined. Let $c = m(\mathcal{T}(\sigma)) + 2^{-|\sigma|}$. By Lemma 5.2 there exists a $\tau \sqsupseteq \mathcal{T}(\sigma)$ such that

$$(\forall \eta)[\mathcal{T}(\sigma) \sqsubseteq \eta \sqsubseteq \tau \to m(\eta) \leq c]$$

and

$$m(\tau 0) \leq c \text{ and } m(\tau 1) \leq c.$$

We can find such a τ by searching. Let $\mathcal{T}(\sigma 0) = \tau 0$ and $\mathcal{T}(\sigma 1) = \tau 1$.

We show that no branch of $\mathcal{T}$ is in S. Let $d_1 d_2 d_3 \cdots \in \{0,1\}^\omega$. By induction we have

$$[\sigma \sqsubseteq \mathcal{T}(d_1 d_2 \cdots d_n)] \rightarrow \left[m(\sigma) \leq m(\mathcal{T}(\lambda)) + \sum_{i=0}^{n} 2^{-i} \right].$$

Hence if $A = b_1 b_2 \cdots$ is any branch of $\mathcal{T}$ then

$$\lim_{n \to \infty} m(b_1 \cdots b_n) \leq m(\mathcal{T}(\lambda)) + \sum_{i=0}^{\infty} 2^{-i} \leq m(\mathcal{T}(\lambda)) + 2.$$

Since m is a martingale that wins on S we have $A \notin S$.

Case 2: S is effectively meager. Let $\{h_k\}_{k \in \omega}$ be the associated uniformly recursive functions (see Definition 2.8). Let $\mathcal{T}(\lambda) = \lambda$ and for, $b \in \{0,1\}$ let $\mathcal{T}(\sigma b) = h_{|\sigma|}(\mathcal{T}(\sigma)b)$. Clearly no branch of $\mathcal{T}$ is in S. ∎

Theorem 6.7 *Let $\mathcal{I}$ be a reasonable inference class. If there exists $S \in \mathcal{I}$ such that $REC_{0,1} - S$ is effectively meager or $\mu_P(REC_{0,1} - S) = 0$ then $REC_{0,1} \in \mathcal{I}$.*

Proof: Let $S' = REC_{0,1} - S$. There are two cases.

Case 1: S' is not dense. Hence $(\exists \sigma)(\forall f \in REC_{0,1})[\sigma \sqsubseteq f \rightarrow f \in S]$. Hence $\sigma \cdot REC_{0,1} \subseteq S$, so $\sigma \cdot REC_{0,1} \in \mathcal{I}$. Since $\mathcal{I}$ is reasonable $REC_{0,1} \in \mathcal{I}$.

Case 2: S' is dense. By Lemma 6.6 there exists a recursive tree $\mathcal{T}$ such that no branch of $\mathcal{T}$ is in S'. Hence every recursive branch of $\mathcal{T}$ is in S, or in our notation $RECB(\mathcal{T}, REC_{0,1}) \subseteq S$. Therefore $RECB(\mathcal{T}, REC_{0,1}) \in \mathcal{I}$. Since $\mathcal{I}$ is reasonable $REC_{0,1} \in \mathcal{I}$. ∎

Corollary 6.8 *Let $\mathcal{I} \in \{PEX, EX, BC\}$, $a, b \in \mathbb{N}$ with $a \leq b$, and $c, d \in \mathbb{N} \cup \{*\}$. Assume that $d \neq *$ or $\mathcal{I} \neq BC$. If $S \in [a,b]\mathcal{I}_c^d$ then $REC_{0,1} - S$ is not effectively meager and $\mu_P(REC_{0,1} - S) \neq 0$.*

Proof: This follows from Theorem 6.7 and Lemma 2.23. ∎

7   Conclusions and Open Problems

We have shown that there are sets in EX_0 that are not effectively meager, hence, even very limited learning algorithms must have very large hypothesis spaces to search through. For recursive measure, the learnable sets become large at the same place in the hierarchy of learnable classes. For partial recursive measure the point is at EX_1.

The complements of the learnable sets are all large with respect to both measure and category. This indicates that unless a technique is guaranteed to learn all the recursive functions, then it will fail to learn a large set of them.

It is an open problem to extend this work to other types of learning. In particular one can ask such questions about PAC-learning using a more restricted notion of measure and category.

8   Appendix

The reader is assumed to know the definitions $Q_i EX[L]$ from [GS92].

Theorem 8.1 *The class $QEX[Suc]$ is not reasonable. (using Definition 6.4).*

Proof: Assume, by way of contradiction, that $QEX[Suc]$ is reasonable. We describe a set S and a tree $\mathcal{T}$ such that that the following occurs.

1. $S \in QEX[Suc]$.

2. $S = RECB(\mathcal{T}, REC_{0,1})$.

Since $REC_{0,1} \notin QEX[Suc]$ this will be a contradiction. Let

$$S = \{B(0)01^{a_0}0B(1)01^{a_1}0B(2)01^{a_2}0\cdots : B \in REC_{0,1}\}$$

where a_0, a_1, a_2, ... is a recursive enumeration of K. $S \in Q_1 EX[S]$ since queries *about* $A \in S$ can be used to ask queries *to* K; from which one can easily EX infer S. To ask "$n \in K$" (for $n > 1$) you can ask

$$(\exists x)[x \notin A \wedge x + 1 \in A \wedge \cdots \wedge x + n \in A \wedge x + n + 1 \notin A].$$

S is the set of all recursive branches of the binary tree given by $T(\lambda)=\lambda$ and $T(\sigma b) = T(\sigma)b01^{a_{|\sigma|}}0$. Now $S = RECB(\mathcal{T}, REC_{0,1})$ can be inferred under the criterion $QEX[Suc]$ while the criterion fails to infer $REC_{0,1}$, so it is not reasonable. ∎

Indeed Theorem 8.1 uses only queries with one existential quantifier. Also $+$ instead of S might be used. Furthermore the proof works for $Q_2EX[<]$. The case $Q_1EX[<]$ needs a special proof.

Lemma 8.2 *If $S \in Q_1EX[<]$ then $S \subseteq S' \cup S_0 \cup S_0'$ for some $S' \in EX$ where S_0 denotes the class of finite and S_0' the class of cofinite sets.*

Proof: Let S' be the set of all $f \in S$ such that $(\exists^{\infty}x)[f(x) = 0]$ and $(\exists^{\infty}x)[f(x) = 1]$. We show that $S' \in EX$. This easily implies the lemma.

Since $S' \subseteq S$ we have $S' \in Q_1EX[<]$. We show that any query made in the $S' \in Q_1EX[<]$ inference, when asked of a function in S', is equivalent to a query with no quantifiers. This easily implies $S' \in EX$.

By standard techniques any existential query in the langaugue $[<]$ is the disjunction of queries of the form

$$(\exists x_1, .., x_n)[c_{m+1} < x_1 < x_2 < .. < x_n < c_{m+2} \wedge f(x_1) = a_1 \wedge \cdots \wedge f(x_n) = a_n \wedge f(c_1) = b_1 \wedge \cdots \wedge f(c_m) =$$

where $a_i, b_i \in \{0, 1\}$, $c_1, \ldots, c_{m+1} \in \mathsf{N}$ and $c_{m+1} \in \mathsf{N} \cup \{\infty\}$. If $c_{m+2} \neq \infty$ then there is an upper bound on the x_i, hence the query can be rewritten without any quantifiers. If $c_{m+2} = \infty$ then, since for $f \in S'$ $(\exists^{\infty}x)[f(x) = 0]$ and $(\exists^{\infty}x)[f(x) = 1]$, we know that

$$(\exists x_1, .., x_n)[c_{m+1} < x_1 < x_2 < .. < x_n < c_{m+2} \wedge f(x_1) = a_1 \wedge \cdots \wedge f(x_n) = a_n]$$

is true. Hence for $f \in S'$ the query is equivalent to the quantifier-free query $f(c_1) = b_1 \wedge \cdots \wedge f(c_m) = b_m$.
∎

Lemma 8.3 $Q_1EX[<]$ *is not reasonable.*

Proof: Let $S \in Q_1 EX[<]$. By Lemma 8.2 $S \subseteq S' \cup S_0 \cup S_0'$ for some $S' \in EX$ where S_0 denotes the class of finite and S_0' the class of cofinite sets. It is well-known that $S_0 \cup S_1 \notin EX$ for S_1 the self describing functions (see Definition 2.15). Now the tree given by $T(\lambda) = \lambda$ and $T(\sigma b) = T(\sigma)01b$ maps $S_0 \cup S_1$ to a family S_2 of sets which are never finite nor cofinite. Since EX is reasonable,

$$S_0 \cup S_1 \notin EX \Rightarrow S_2 \notin EX \Rightarrow S_2 \notin Q_1 EX[<]$$

and $Q_1 EX[<]$ is not reasonable. ∎

Corollary 8.4 *None of the classes $Q_i EX[Suc]$, $Q_i EX[<]$, $Q_i EX[+]$, $QEX[Suc]$, $QEX[<]$ and $QEX[+]$ with $i \geq 1$ is reasonable.*

It is open to determine if there is a query inference class with a nontrivial measure zero or meager complement.

References

[AS83]   D. Angluin and C. H. Smith. Inductive inference: Theory and methods. *Computing Surveys*, 15:237–269, 1983.

[Bar74]   J. Bārzdiņš. Two theorems on the limiting synthesis of functions. In Bārzdiņš, editor, *Theory of Algorithms and Programs*, volume 1, pages 82–88. Latvian State University, Riga, U.S.S.R., 1974.

[BG95]   S. Ben-David and L. Gurvits. A note on VC-dimension and measure of sets of reals. *Proceedings of the Eighth Annual Workshop on Computational Learning Theory*, pages 454-461, ACM Press, 1995.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

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

[Bor05]   E. Borel. *Leçons sur les fonctions de variables réeles.* Gauthier-Villars, Paris, 1905.

[Bor14]   E. Borel. *Leçons sur la théorie des fonctions.* Gauthier-Villars, Paris, 1914.

[CS83]   J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25(2):193–220, 1983.

[DKV92]   R. Daley, B. Kalyanasundaram, and M. Velauthapillai. Breaking the probability 1/2 barrier in fin-type learning. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 203–217, ACM Press, 1992.

[DKV93]   R. Daley, B. Kalyanasundaram, and M. Velauthapillai. Capabilities of fallible finite learning. *Proceedings of the Sixth Annual Workshop on Computational Learning Theory*, pages 199–208, ACM Press, 1993.

[DPVW91]   R. Daley, L. Pitt, M. Velauthapillai, and T. Will. Relations between probabilistic and team one-shot learners. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 228–239, Morgan Kaufmann Publishers, 1992.

[Fen91]   S. Fenner. Notions of resource-bounded category and genericity. *Proceedings of the Sixth Annual Conference on Structure in Complexity Theory*, pages 196–212. IEEE Computer Society, 1991.

[FKS94]   R. Freivalds, M. Karpinski, and C. Smith. Co-learning of total recursive functions. *Proceedings of the Seventh Annual Workshop on Computational Learning Theory*, pages 190–197, ACM, 1994.

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

[GS92]   W. Gasarch and C. H. Smith. Learning via queries. *Journal of the ACM*, 39(3):649–675, 1992. A shorter version is in 29th FOCS conference, 1988, pp. 130-137.

[JS90]   S. Jain and A. Sharma. Finite learning by a team. *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 163–177, Morgan Kaufmann Publishers, 1990.

[Kur58]   C. Kuratowski. *Topologie Vol. 1,* 4th *edition*, volume 20 of *Monografie Matematyczne*, Panstwowe Wydawnictwo Naukowe, 1958.

[Lis81]   L. Lisagor. The Banach-Mazur game. Translated Version of *Matematicheskij Sbornik*, 38:201–206, 1981.

[Lut92]   J. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and Systems Science*, 44:226-258, 1992.

[Lut93]   J. Lutz. The quantitative structure of exponential time. In *Proceedings of the Eighth Annual Conference on Structure in Complexity Theory Conference*, pages 158–175. IEEE Computer Society, 1993.

[Mel73]   K. Mehlhorn. On the size of sets of computable functions. *Proceedings of the Fourteenth Annual Symposium on Switching and Automata Theory*, pages 190–196, IEEE Computer Society, 1973.

[MY78]   M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms*. North-Holland, New York, 1978.

[OSW86]   D. N. Osherson, M. Stob, and S. Weinstein. Aggregating inductive expertise. *Information and Control*, 70:69–95, 1986.

[Oxt71]   J. Oxtoby. *Measure and Category*. Springer-Verlag, 1971.

[Pit89]   L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36(2):383–433, 1989.

[PS88]   L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.

[Put75]   H. Putnam. Probability and confirmation. In *Mathematics, Matter and Method*, volume 1. Cambridge University Press, 1975.

[Rog67]   H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967.

[Sch71]   C. P. Schnorr *Zufälligkeit und Wahrscheinlichkeit.* Springer-Verlag Lecture Notes in Mathematics, Vol. 218, 1971.

[Smi82]   C. H. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29(4):1144–1165, 1982.

[Smi94]   C. Smith. *A Recursive Introduction to the Theory of Computation.* Springer-Verlag, 1994.

[Soa87]   R. I. Soare. *Recursively Enumerable Sets and Degrees.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987.

[Ter95]   S. A. Terwijn, private communication, August 1995.

[Vel89]   M. Velauthapillai. Inductive inference with a bounded number of mind changes. *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 200–213, Palo, Morgan Kaufmann, 1989.