On the Classification of Computable Languages

John Case * — University of Delaware
Efim Kinber † — Sacred Heart University
Arun Sharma ‡ — University of New South Wales
Frank Stephan § — University of Heidelberg

Abstract

A one-sided classifier for a given class of languages converges to 1 on every language from the class and outputs 0 infinitely often on languages outside the class. A two-sided classifier, on the other hand, converges to 1 on languages from the class and converges to 0 on languages outside the class. The present paper investigates one-sided and two-sided classification for classes of computable languages. Theorems are presented that help assess the classifiability of natural classes. The relationships of classification to inductive learning theory and to structural complexity theory in terms of Turing degrees are studied. Furthermore, the special case of classification from only positive data is also investigated.

1 Introduction

Consider the problem of determining whether a language A over $\mathbb{N}$, the set of natural numbers $\{0, 1, 2, \ldots\}$, satisfies a certain property. Let $\mathcal{A}$ denote the class of all languages over $\mathbb{N}$ that satisfy the given property. The question of classification can then be stated thus: if one is given data about A, can one determine if $A \in \mathcal{A}$.

We briefly discuss the various approaches to the study of classification in the literature. One of the earliest attempts was the design of finite automata to decide whether an infinite string (representing the characteristic function of a language) belongs to a given ω-language or not [9, 25, 27, 39]. But the restrictive computational ability of these finite automata led Büchi [9] and his successors to consider non-deterministic automata. The present paper takes the alternate approach of choosing Turing machines as classifiers. In fact this approach had already been initiated by Büchi and Landweber [10, 24].

Smith and Wiehagen [37] introduced a model of classification analogous to the Gold model

*Computer and Information Sciences Department, 101A Smith Hall, University of Delaware, Newark, DE 19716-2586, United States of America, case@cis.udel.edu.

†Computer Science, Sacred Heart University, 5151 Park Avenue, Fairfield, CT 06432-1000, United States of America, kinbere@sacredheart.edu.

‡School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW, 2052, Australia, arun@cse.unsw.edu.au. Supported by a grant from the Australian Research Council.

§Mathematisches Institut der Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany, European Union, fstephan@math.uni-heidelberg.de. Supported by the Deutsche Forschungsgemeinschaft (DFG) grant Am 60/9-1.

of learning [8, 18, 29]. The (computable) classifier M sees longer and longer prefixes σ of the characteristic function of a language $A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \ldots \cup \mathcal{A}_k$ and guesses on each input σ some number $h \in \{1, 2, \ldots, k\}$ to indicate that $A \in \mathcal{A}_h$. These guesses are supposed to converge, for each set $A \in \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$, to a value h such that $A \in \mathcal{A}_h$. Smith, Wiehagen and Zeugmann [38] extended this study in various ways.

Ben-David [6] and Kelly [22] also interestingly studied classification. They call a class *classifiable* iff there exists a (not-necessarily-computable) functional that indicates in the limit for every A whether or not it belongs to a given class $\mathcal{A}$. They obtained topological conditions for classifiable classes. Gasarch, Pleszkoch, Stephan and Velauthapillai [16] extended this study and obtained relations between the Borel hierarchy on classes — which is induced by the space $\{0,1\}^\infty$ with product topology — and the query hierarchy obtained by allowing a certain number of quantifier-alternations during querying a teacher on the target set A.

Later Stephan [35] investigated the limits of (computable) classifiers. He considered classification of languages with respect to one single class $\mathcal{A}$ and used the following two natural models of classification: Two-sided classification which is the computable counterpart to Ben-David's classification in the limit and one-sided classification which is already implicit in the notion of reliable inference (on languages inside the class the learner converges to an index for the language and on languages outside the class the learner makes infinitely many mind changes). These two notions of classes are very natural and coincide with the Δ_2 and Σ_2 classes studied by recursion theorists [19, 28, 31, 32]. Our study derives from these models which we present next. But, first some notation.

We take a *classifier* to be an algorithmic device; M, N and H range over classifiers. Calligraphic letters range over classes, A, B range over sets and U ranges over oracles. We take σ, τ to range over prefixes of strings of characteristic functions of sets. $\sigma \preceq \tau$ means that $\tau(x)\downarrow = \sigma(x)$ for all $x \in dom(\sigma)$. $M(\sigma)$ denotes the guess issued by classifier M on a prefix $\sigma \preceq A$ of the input-set A.

> *Two-Sided Classification*: For all languages A: $M(\sigma) = \mathcal{A}(A)$ for almost all $\sigma \preceq A$.

Here $\mathcal{A}(A)$ is 1 if $A \in \mathcal{A}$ and 0 otherwise, that is, classes and sets are identified with their characteristic function. Two-sided classification may be considered to be too strong a requirement. In some applications it is sufficient if the classifier is able to signal the inclusion of a language in a given class, but only provides a weaker signal if the language is not in the class. Stephan [35] introduced the notion of one-sided classification to model this idea.

> *One-Sided Classification*: For all languages A: if $A \in \mathcal{A}$, then $M(\sigma) = 1$ for almost all $\sigma \preceq A$; if $A \notin \mathcal{A}$, then $M(\sigma) = 0$ for infinitely many $\sigma \preceq A$.

We normally let M and N range over two-sided classifiers and let H range over one-sided classifiers. The notion of one-sided classification is reasonable since the classifier outputs 0 infinitely often thereby guaranteeing that the classifier never locks onto an incorrect conjecture.

In the present paper, we restrict our investigation to classification of computable languages. This restriction may be supported by the fact that practical examples are always computable and assuming an algorithmic view of the universe, it is unlikely that nature generates noncomputable languages. Thus, our classifiers can be relied upon if they are never expected to deliberate upon noncomputable languages. Hence, in the sequel, the statement "for all languages A" in the above two definitions is replaced by "for all computable languages A".[1]

The present paper may also be seen as closing the gap between Stephan's abstract work [35] and the more concrete approach of Smith, Wiehagen and Zeugmann [37, 38]. Before we begin a formal presentation of the results, we give an informal tour of the various sections in the paper.

In Section 2, we introduce the basic definitions and give preliminary results about two-sided and one-sided classification for classes of computable languages. We give concrete classes of languages that can be two-sided and one-sided classified. In particular we observe that one-sided classes are closed under finite monotone Boolean combinations and two-sided classes are closed under all finite Boolean combinations. We also show that every uniformly recursive family of languages is one-sided classifiable. Additionally, if the family is discrete, then it is also two-sided classifiable. As a consequence of this result, the class of pattern languages is two-sided classifiable. As a contrast, however, the class of regular languages is only one-sided classifiable.

Although, from [37] we already know that learning and classification are, in general, incomparable, in Section 3, we provide some pleasant links between learning with bounded mind changes and classification. We show that for the concept of exact PEx-learning, a class is identifiable with a (generalized) bound on the number of mind changes iff it is two-sided. For the notion of Ex-learning a weaker version of this correspondence still holds.

In Section 4 we show that classes identifiable in the limit from informants can be *reliably* identified iff they are one-sided classifiable. We also investigate conditions under which reliable identification in the limit and two-sided classification are linked.

The characteristic function of a language conveys both positive and negative data about the language. In Section 5, we argue that it may not be realistic to assume the availability of both positive and negative data in practice. The experience from empirical studies of learning is that negative data is not always readily available and even when it is available, it is often tedious to obtain. Motivated by such concerns, we also investigate two-sided and one-sided classification from only positive data. Following the practice in inductive inference literature, we model positive data as texts. As expected, we show that classification from texts is very difficult. As a simple consequence of our result, the class of pattern languages is not even one-sided classifiable from texts.

Not deterred by the difficulty of classification from texts, we investigate a weaker version of classification for text presentation, called *partial classification*, that yields some positive results. A class $\mathcal{A}$ is *partially classifiable* just in case there exists a machine that on texts for languages in $\mathcal{A}$ outputs exactly one guess infinitely often and on texts for nonmembers of $\mathcal{A}$ does not output any single guess infinitely often. The motivation here is that a partial classifier gives a weak signal if the language belongs to the class and refuses to give any signal if the language is not a member of the class being classified. We show that partially classifiable classes can be defined in terms of their index sets: $\mathcal{A}$ is partially classifiable iff its index set $\{e : W_e$ is computable and $W_e \in \mathcal{A}\}$ is Σ_3. Furthermore, this general criterion turns out to be independent of the data-presentation: the partially classifiable classes are the same for presenting positive data only and presenting positive and negative data at the same time. Most classes considered in practical applications, for example, the class of pattern languages, have a Σ_3 index set and are therefore partially classifiable.

In Section 6, we investigate structurally the computational limits of classifying computable

[1] So, we ignore noncomputable sets everywhere. Accordingly, set-theoretic notions like the complement of classes are adapted to the computable universe: $\overline{\mathcal{A}} = \{\text{computable } A : A \notin \mathcal{A}\}$.

languages. In particular, we investigate the "computational distance" between one-sided and two-sided classification by determining the kind of noncomputable information that yields a two-sided classifier for a class that was otherwise only one-sided classifiable. This gives insight into what it takes for a class of interest to be two-sided versus one-sided classifiable. We show that access to a high oracle is sufficient to construct a two-sided classifier for a one-sided classifiable class. We also establish that in some cases the power of a high oracle is necessary as there are classes for which any two-sided classifier has high Turing degree. We adapt Post's notion of creative set to describe the one-sided classifiable classes that are effectively not two-sided classifiable. We call a one-sided classifiable class $\mathcal{A}$ *creative* just in case there is a uniformly computable sequence of languages $A_0, A_1, \ldots$ such that for each one-sided classifier H_e, the language A_e is a counterexample to the hypothesis "H_e classifies $\overline{\mathcal{A}}$". The analogy between the two notions of creativity turns out to be quite striking. We give examples of creative classes and show that a creative class is two-sided *only* relative to a high oracle. We discuss some interesting results about one-sided classifiable classes of intermediate complexity and compare our results with the more abstract study of classification by Stephan [35] in which a classifier has to behave correctly on noncomputable languages, too.

Finally, in Section 7, we consider classifiers that, instead of guessing 0 or 1, output programs that converge in the limit to 0 or 1. Such programs may be viewed as generators of trial and error guesses and classifiers that output such programs may be viewed to be of somewhat lower quality (compared to the classifiers that directly guess 0 or 1). We consider two kinds of such classifiers: Ex-style requiring that the sequence of programs converge to a single program that has the correct guess of 0 or 1 in the limit and BC-style requiring that the sequence of programs eventually contain only programs that have the correct guess of 0 or 1 in the limit. We show that the notion of Ex-style classification nicely coincides with two-sided classification. We also show that every one-sided classifier has a BC-style classifier. We conclude with insightful, structural characterizations of BC-style classification.

A previous version of this work has been presented at the Symposium on Theoretical Aspects of Computer Science (STACS 1997) and was published as a technical report [13].

2 Basic Definitions and Results

Formally, a one-sided classifier is just a $\{0,1\}$-valued function on strings and the languages accepted by this classifier are those where the classifier converges to 1.

Definition 2.1 A classifier H is an algorithm that on every string σ outputs a number 0 or 1. It classifies a class $\mathcal{A}$ *one-sided* just in case

- if $A \in \mathcal{A}$, then $H(\sigma) = 1$ for almost all $\sigma \preceq A$;
- if $A \in \overline{\mathcal{A}}$, then $H(\sigma) = 0$ for infinitely many $\sigma \preceq A$.

The classifier H is furthermore *two-sided* iff the statement "for infinitely many" in the second clause can be strengthened to "for almost all". Note that in this definition the variable A ranges over only *computable* sets.

There is an effective list of classifiers H_e such that for each one-sided class there is some H_e classifying it one-sided and for each two-sided class there is some H_e classifying it two-sided. Assuming an acceptable numbering φ_e of all partial computable functions, these classifiers are defined as follows:

$$H_e(\sigma) = \begin{cases} \varphi_e(\tau) & \text{for the longest } \tau \preceq \sigma \text{ such that} \\ & \varphi_e(\tau) \text{ outputs 0 or 1 within } |\sigma| \text{ steps;} \\ 0 & \text{if there is no such } \tau. \end{cases}$$

Now it is easy to verify that whenever φ_e is a one-sided classifier for $\mathcal{A}$, then so is H_e; and whenever φ_e is a two-sided classifier for $\mathcal{A}$, then so is H_e. This normalization has the advantage that now we can assume without loss of generality that all one-sided (two-sided) classes have a total and computable one-sided (two-sided) classifier. Therefore, in the sequel, we will consider H_e instead of the underlying φ_e.

One-sided classes are closed under finite monotone Boolean combinations and two-sided classes are closed under all finite Boolean combinations. Although this fact is a direct corollary from that fact that one-sided and two-sided classification are the Σ_2 and Δ_2 classes restricted to computable sets, the proofs are nevertheless included for the sake of completeness.

Fact 2.2 *$\mathcal{A}$ is two-sided iff $\mathcal{A}$ and $\overline{\mathcal{A}}$ are one-sided. If $\mathcal{A}, \mathcal{B}$ are one-sided classes so are $\mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$. If $\mathcal{A}$ is one-sided so is $\mathcal{B} = \{B : B$ is a finite variant of some $A \in \mathcal{A}\}$.*

Proof The direction ($\Rightarrow$) of the first statement is obvious. For the reverse direction ($\Leftarrow$), let H_1 be a one-sided classifier for $\mathcal{A}$ and let H_2 be one for $\overline{\mathcal{A}}$. Let $M(\lambda) = 0$; we define inductively:

$$M(\sigma w) = \begin{cases} H_1(\sigma w) & \text{if } H_1(\sigma w) \neq H_2(\sigma w); \\ M(\sigma) & \text{otherwise.} \end{cases}$$

We claim that M is a two-sided classifier for $\mathcal{A}$: If a computable set A is in $\mathcal{A}$, then H_1 converges on A to 1 while H_2 outputs on A infinitely many 0s. So there are infinitely many $\tau \preceq A$ with $H_1(\tau) = 1$ and $H_2(\tau) = 0$ but only finitely many $\tau \preceq A$ with $H_1(\tau) = 0$ and $H_2(\tau) = 1$. So M will converge to 1. Similarly M will converge to 0 on any computable set $A \in \overline{\mathcal{A}}$.

For the second statement, let H_1 be a one-sided classifier for $\mathcal{A}$ and H_2 be a one-sided classifier for $\mathcal{B}$. Now $\mathcal{A} \cap \mathcal{B}$ has the one-sided classifier

$$H(\sigma) = \begin{cases} 1 & \text{if } H_1(\sigma) = 1 \text{ and } H_2(\sigma) = 1; \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that H outputs on A infinitely many 0s iff either H_1 or H_2 does. So H converges on A to 1 iff both H_1 and H_2 converge on A to 1. The case $\mathcal{A} \cup \mathcal{B}$ is a bit more involved. The following fact is used in defining the machine:

> "H outputs on A in total at least n 0s if H_1 and H_2 both output on A in total at least n 0s."

This informal idea can be turned into an algorithm as follows: let

$$n_M(\sigma) = |\{\tau \preceq \sigma : M(\tau) = 0\}|$$

for each machine $M \in \{H, H_1, H_2\}$, $H(\lambda) = 1$ and

$$H(\sigma w) = \begin{cases} 0 & \text{if } n_{H_1}(\sigma w) > n_H(\sigma) \text{ and } n_{H_2}(\sigma w) > n_H(\sigma); \\ 1 & \text{otherwise.} \end{cases}$$

It is easy to see that H outputs infinitely many 0s iff both H_1 and H_2 output infinitely many 0s. Therefore, H converges on a set A to 1 if at least one of the machines H_1 and H_2 converge to 1.

The classifier for $\mathcal{B}$ in the last statement is constructed such that it outputs on B at least n 0s iff the classifier for $\mathcal{A}$ outputs on each set of the form $a_0 a_1 \ldots a_n B(n+1)B(n+2)\ldots$ at least n 0s. ∎

Given a computable function $A(x,y)$, let $A_x = \{y : A(x,y) = 1\}$ and $\mathcal{A}$ be the class $\{A_0, A_1, \ldots\}$. Such a class $\mathcal{A}$ is called a *uniformly recursive family.* Angluin [3] initiated the study of learning uniformly recursive families from texts. After the introduction of monotonicity constraints many papers have considered the learnability of these families from texts and informants [21, 40, 41]. A class $\mathcal{A}$ is *closed* iff for each $A \notin \mathcal{A}$ there is a $\sigma \preceq A$ such that no $B \in \mathcal{A}$ extends σ. The first part of the next fact is also an immediate consequence of the fact, that uniformly computable families are Σ_2 classes [31]: $A \in \mathcal{A} \Leftrightarrow (\exists x)\,(\forall y)\,[A(y) = A_x(y)]$.

Fact 2.3 *Every uniformly computable family is one-sided. If it is also closed, then it is two-sided.*

Proof As in the proof of Fact 2.2 let $n_H(\sigma) = |\{\tau \preceq \sigma : H(\tau) = 0\}|$, $H(\lambda) = 1$ and

$$H(\sigma w) = \begin{cases} 1 & \text{if } \sigma w \preceq A_x \text{ for } x = n_N(\sigma); \\ 0 & \text{otherwise.} \end{cases}$$

The intuitive idea behind H is to check the sets $A_0, A_1, \ldots$; whenever A_x turns out to be different from A, H outputs a 0 and moves on to A_{x+1}, otherwise H outputs 1 as long as A_x and A appear to be equal. So H converges on every set A_x to 1 making (at most) x 0s and outputs infinitely many 0s for all $A \notin \mathcal{A}$.

Assume now the same algorithm for a closed class $\mathcal{A}$ and let $A \notin \mathcal{A}$ be computable. Then there is $\tau \preceq A$ such that no A_x extends τ. In particular $\sigma w \not\preceq A_x$ for all $\sigma \succeq \tau$, all x and almost all $\sigma w \preceq A$. If follows that $H(\sigma w) = 0$ for almost all $\sigma w \preceq A$. So H is already a two-sided classifier for $\mathcal{A}$. ∎

Example 2.4 The immediately preceding results yield the following examples.

- $\mathcal{C} = \{A : A \text{ is cofinite}\}$ is one-sided, but not two-sided.
  The classifier is $H(\sigma w) = w$.

- $\mathcal{D} = \{1^\infty, 01^\infty, 001^\infty, 0001^\infty, \ldots\}$ is two-sided.
  The classifier M outputs 1 if $\sigma \in 0^*1^+$ and 0 otherwise.

- $\mathcal{E} = \{A : A \text{ has finite and even cardinality}\}$ is one-sided, but not two-sided.
  The classifier $H(\sigma)$ outputs 1 iff the number of 1s in σ is even and 0 iff this number is odd.

- $\mathcal{F}_\phi = \{A : \text{the formula } \phi(A) \text{ is true}\}$ is two-sided.
  Here $\phi(A)$ means that ϕ is a Boolean formula, such as $[5 \in A \vee [3 \notin A \wedge 4 \notin A]]$, with A being the only free variable representing the input-set A of the same name. Such formulas can be evaluated after having seen a sufficiently long part of the input and from then on the classifier outputs 1 if $\phi(A)$ holds and 0 if $\phi(A)$ does not hold.

- $\mathcal{G} = \{graph(p) : p \text{ is a polynomial}\}$ is one-sided, but not two-sided.
  $\mathcal{G}$ and $\mathcal{R}$ below are uniformly recursive families and, hence, have the one-sided classifier from Fact 2.3.

- $\mathcal{P} = \{A : A \text{ is a pattern language}\}$ is two-sided.
  This is due to the fact that the class of the pattern languages is both closed and uniformly recursive.

- $\mathcal{R} = \{A : A \text{ is regular}\}$ is one-sided, but not two-sided.

There is also a prominent class which is not one-sided: the class $\{A : \varphi_{\min(A)} \text{ computes } A\}$ of the self describing sets. But this class has a one-sided complement.

3 Two-Sided Classification with Bounded Mind Changes

Barzdins and Freivalds [7] introduced the notion of bounded mind changes where a machine has the right to output only a fixed finite number of guesses such that the last one of them is correct. This notion was more generally considered by Case and Smith [14]. Freivalds and Smith [15] generalized this concept further by using constructive ordinal [32] bounds; this concept is equivalent to the second type in the hierarchy below. See [2, 20] for examples of natural classes learnable with ordinal mind change bounds.

When an inductive learner or a classifier abandons its currently held guess and conjectures a new guess, it is said to make a *mind change*. The general idea behind bounded mind changes learning or classification is that during the learning or classification process, the machine has the right to make only a bounded number of mind change. To realize this, the classifier can be viewed as being equipped with a counter in such a way that a mind change is permitted only if the counter is decreased (that is, changes to a lower number) at the same time. In the more concrete cases, the counter starts with a natural number, say 5, and goes down with every mind change, for example, from 5 to 4, then from 4 to 3 until it reaches 0 and can't be decreased further. More abstract realizations use counters which range over ordinals. In the context of inductive learning Sharma, Stephan and Ventsov [33] have considered the following four types of bounded mind changes whose definitions can be directly adapted for classification:

first type: constant bound. The counter can take as value only a natural number.

second type: ordinal bound. The counter is a rational number which is always member of a well-ordered subset Q of the rationals.

third type: linear bound. The set Q of permitted rational numbers has no recursive decreasing infinite sequence but may have a nonrecursive one.

fourth type: general bound. The set Q of permitted counter values is equipped with a partial ordering $\sqsubseteq$ such that there are no recursive infinite descending chains.

A fifth type where Q may have a infinite recursive descending chain is just the same as having no restriction at all. In the context of classification, this fifth type is equivalent to one-sided classification. Note that each such definition requires that the set Q of possible counter values is recursively enumerable and the ordering on Q is recursive with domain Q.

Theorem 3.1 *For a class $\mathcal{A}$ the following is equivalent:*
(a) *$\mathcal{A}$ can be classified using a classifier with bounded mind changes of the second type;*

(b) *$\mathcal{A}$ can be classified using a classifier which converges on every set – also on every* nonrecursive *set;*
(c) *$\mathcal{A}$ can be classified using a classifier of fourth type with the additional requirement that the partially ordered set Q of counter-values does not have any decreasing infinite sequence — also no nonrecursive one.*

The proof of this theorem is similar to that in the case of inductive inference [33]; so it is omitted here. The next theorem however is quite different to that setting since there are inferable classes which do not have bounded mind changes of the fourth type.

Theorem 3.2 *Every two-sided class has a mind change bound of the fourth type.*

Proof Let M be a two-sided classifier for $\mathcal{A}$. Then let Q contain the string λ and all strings σa such that $M(\sigma a) \neq M(\sigma)$. The ordering $\sqsubset$ is just given by $\tau \sqsubset \sigma$ iff τ is a proper extension of σ. Furthermore M assigns to each mind change on each input σa the string $\sigma a \in Q$ as the counter for the mind change. It remains to show that Q has no descending recursive sequence: Assume that $\sigma_0 \sqsubset \sigma_1 \sqsubset \ldots$ would be such a sequence. Each string σ_k has length at least k and all σ_k coincide on their common domain: so they have a recursive limit A given by $A(x) = \sigma_{x+1}(x)$. M makes at "the end" of each σ_k a mind change, thus M makes infinitely many mind changes on A in contradiction to the fact that M is a two-sided classifier for A. So the theorem follows. ∎

The next theorem again transfers directly from the case in inductive inference [33].

Theorem 3.3 *The hierarchy given by the four notions of bounded mind changes is proper.*

For the restrictive notion of Popperian Explanatory identification (PEx), a strong relation between learnability with bounded mind changes and classifiability is established. Recall that PEx-learning [11, 14] means that the learner Ex-identifies the class with the additional requirement that any guess — also on illegal data — is an index of a total recursive function. This has some consequences as the one that it can be checked immediately whether a guess is consistent with the data seen so far. Furthermore M learns a class $\mathcal{A}$ exact iff M converges only on sets in $\mathcal{A}$ to a correct program; on computable sets outside $\mathcal{A}$ either M diverges or converges to a program which computes something else.

Theorem 3.4 *Let $\mathcal{A}$ be exactly* PEx-*learnable. Then $\mathcal{A}$ is two-sided iff $\mathcal{A}$ can be exactly* PEx-*learned with bounded mind changes of the fourth type.*

Proof M exactly PEx-learns $\mathcal{A}$ iff for any computable $A \notin \mathcal{A}$, M either does not converge or converges to a false program, that is, a program which computes for some x a wrong output. Note that for inductive learning, mind change bound of the fourth type means that the learner converges to some program on every computable set.

For the first direction let N be an exact PEx-learner for $\mathcal{A}$ which respects a mind change bound on the fourth type, in other words, N converges on every computable set. Now the classifier M works as follows:

$$M(\sigma) = \begin{cases} 1 & \text{if } \varphi_{N(\sigma)} = \sigma(x) \text{ for all } x \in dom(\sigma); \\ 0 & \text{otherwise, that is, } \varphi_{N(\sigma)}(x) \neq \sigma(x) \text{ for some } x \in dom(\sigma). \end{cases}$$

Since the guesses $N(\sigma)$ are always total programs, M is computable. Furthermore N converges on every computable set to a fixed program and so M also converges on this computable set either to 1 if this last program is a program for the set and to 0 otherwise. So the convergence of M follows from that of N. By the exactness, M classifies $\mathcal{A}$ correctly.

For the other direction let M be a two-sided classifier and N be a PEx-learner for $\mathcal{A}$. Furthermore let e be an index of some function in $\mathcal{A}$. Now the new exact PEx-learner respecting the mind change bound of the fourth type works as follows:

$$H(\sigma) = \begin{cases} N(\sigma) & \text{if } M(\sigma) = 1; \\ e & \text{otherwise, that is, } M(\sigma) = 0. \end{cases}$$

If $A \in \mathcal{A}$ then H converges on A to the same program as N and so identifies A. If $A \notin \mathcal{A}$ then M converges on A to 0 and so H on A to e. So H converges on all computable sets (the others are not considered for mind change bounds of the fourth type). Furthermore H converges to a program for some set in $\mathcal{A}$ while $A \notin \mathcal{A}$, so H is exact. ∎

This result could be improved to stating that $\mathcal{A}$ is two-sided via a classifier respecting mind change bounds of the k-th type iff $\mathcal{A}$ is exactly PEx-learnable via a machine respecting mind change bounds of the k-th type. Nevertheless the result depends on the fact that every PEx-learnable class is one-sided. Replacing PEx by Ex, only a weaker version holds.

Theorem 3.5 *Every two-sided classifiable class $A \in$* Ex *can be* Ex*-learned with bounded mind changes of the fourth type, but some two-sided classifiable class $\mathcal{A} \in$* Ex *cannot be* Ex*-learned with bounded mind changes of second type.*

Proof The construction of the PEx-learner H from a two-sided classifier M and an PEx-learner N in the previous theorem can be carboncopied in order to construct an Ex-learner H from a two-sided classifier M and an Ex-learner N. Again H respects mind change bounds of the fourth type.

For the second statement, consider a *simple* set $S = \{a_0, a_1, \ldots\}$ which is defined as a recursively enumerable set whose complement is infinite but does not contain any infinite recursive set [30]. Let $S = \{a_0, a_1, \ldots\}$ be a simple set and $\mathcal{A} = \{A : |A| \text{ is finite and even and } A \subseteq \overline{S}\}$. A two-sided classifier on input σ checks first whether $\sigma(a_k) = 1$ for some $a_k \in dom(\sigma)$ with $k \leq |\sigma|$. If so, then the classifier outputs 0. Otherwise the output is 1 if the number of all x with $\sigma(x)\!\downarrow = 1$ is even and is 0 if this number is odd. Since no infinite computable set is disjoint from S, this two-sided classifier for $\mathcal{A}$ is correct.

On the other hand $\mathcal{A}$ cannot be learned with bounded mind changes of second type. Let M be an inductive inference machine which learns $\mathcal{A}$ which satisfies a mind change bound of the second type. For each set $A \in \mathcal{A}$ let $q(A)$ be the minimal q_i output during the inference of A. The set $\{q(A) : A \in \mathcal{A}\}$ has a minimum q_j since it is well-ordered. $q_j = q(A)$ for some fixed set A. Now A has finite and even cardinality and there is some $\sigma \preceq A$ such that $M(\sigma)$ is an index for A and M has output q_j while reading this σ. Since $\overline{S}$ is infinite there are $x, y \in \overline{S} - A - dom(\sigma)$ and M has to infer $A \cup \{x, y\}$. Since also $\sigma \preceq A \cup \{x, y\}$, M has to make a mind change after σ and also output a rational $q_i < q_j$. So $q(A \cup \{x, y\}) < q_j$ in contradiction to the choice of q_j and such a machine M does not exist. ∎

Theorem 3.6 *If $\mathcal{A}$ can be exactly* Ex*-learned with bounded mind changes of the fourth type, then $\overline{\mathcal{A}}$ is one-sided. Furthermore some class, namely the class of all self-describing sets, is* Ex*-learnable with bounded mind changes of the first type but is not one-sided.*

Proof Let M be a learner for $\mathcal{A}$ which respects mind change bounds of the fourth type, that is, which converges on every computable set A to some program; this program is a program for A iff $A \in \mathcal{A}$. Now the classifier H for $\overline{\mathcal{A}}$ is specified as follows:

> H outputs at least n 0s for an input-set A iff there is $m \geq n$ such that the index $e = M(A(0)A(1)\ldots A(m))$ computes the first $n+1$ values of A: $\varphi_e(0)\downarrow = A(0)$, $\varphi_e(1)\downarrow = A(1)$, ..., $\varphi_e(n)\downarrow = A(n)$.

That this condition is not computable but recursively enumerable does not matter since H is not required to output the n-th 0 immediately but only after some suitable time. If H outputs finitely many 0s, then M does not converge to an index e which computes A; so H classifies A correctly to be inside $\overline{\mathcal{A}}$. If H outputs infinitely many 0s, then M converges to a program e which coincides on arbitrary long prefixes with A, thus computes A. So H classifies again correctly A to be outside $\overline{\mathcal{A}}$.

The class $\mathcal{B} = \{A : \varphi_{\min(A)} \text{ computes } A\}$ of the self describing sets can be learned with bounded mind changes. The learner waits for the first 1 to appear in the characteristic function and then outputs the place e where this 1 was found. No mind change is necessary, since e is correct whenever the input set is self describing, that is, in $\mathcal{B}$.

The index set of the self describing sets is Π_2: e is an index of a self describing set if φ_e is total and φ_e outputs 0 on input $x < e$ and 1 on input e. It can be shown that this index set is also Π_2 complete.

But if $\mathcal{B}$ would be one-sided via some computable machine M then also the complement of the index set of all self describing functions would be Π_2 yielding a contradiction: Given e, e is not the index of a self-describing function iff M outputs infinitely many 0s on every set beginning with 0^e1, that is, iff for each n there is an m such that M outputs on all strings in $0^e1 \cdot \{0,1\}^m$ at least n 0s. ▮

4 Reliable Learning and One-Sided Classification

A learner is said to *reliably* Ex-identify a class of languages if it either diverges or converges to a correct index, but it never converges to a false one — see Minicozzi for the definition and related results [26]. Among the several variants of reliable learning, we pick the one where the learner is only on the *computable sets* required either to diverge or to infer the set. We denote by REx the class of language classes that can be reliably Ex-identified. Exact REx contains those classes in REx where the reliable learner does learn only the sets in the class. The reader may observe that a reliably inferred class is also in some sense classified since convergence indicates membership in the class and divergence indicates membership in its complement. Hence, it might be expected that there are interesting links between reliable learning and classification.

Theorem 4.1 *Let $\mathcal{A}$ be in* Ex. *Then $\mathcal{A} \in$* exact REx *iff $\mathcal{A}$ is one-sided.*

Proof ($\Rightarrow$): Let $\mathcal{A}$ be exactly reliably learnable. The classifier outputs 0 if the learner changes its mind and outputs 1 if there is no mind change. Whenever the learner converges to an index, then the classifier outputs only finitely many 0s and thus accepts the language. Whenever the learner does not converge to an index, that is, the language does not belong to $\mathcal{A}$, then the classifier rejects the language by outputting infinitely many 0s. So the classifier accepts just the languages in $\mathcal{A}$ and is correct.

($\Leftarrow$): If $\mathcal{A}$ is learnable and one-sided classifiable, then a mind change can be introduced into the learning algorithm by padding at every place where the classifier outputs 0, that is, if the learner outputs for σ and σw the same guess e, but the classifier outputs a 0 for σw, then the learner's output at σw is replaced by an equivalent but different index for the characteristic function computed by e. This does not effect convergence on $A \in \mathcal{A}$ since there these new mind changes are inserted only finitely often. But if $A \notin \mathcal{A}$, then the classifier outputs infinitely many 0s which induce infinitely many mind changes on the modified learner; so this modified learner diverges. Thus the modified learner is reliable, that is, it converges on a computable A if and only if it learns A. Furthermore, the learner is exact since it learns exactly the languages in $\mathcal{A}$. ∎

The reader may have observed that in the above proof of Theorem 4.1, at no point are the guesses evaluated. Therefore this result can be translated to similar notions as long as the following two conditions are satisfied: padding is available and infinitely many mind changes (as in the notion of behaviorally correct learning) are not permitted.

Case, Jain and Sharma [12] introduced the notion of learning limiting recursive programs or "trial-and-error-guesses" in the context of learning functions. We adapt this notion to identification in the limit of limiting programs for characteristic functions of computable languages. According to this criterion of learning, a learner has to converge on the characteristic function of every language $A \in \mathcal{A}$ to a total program e in two variables which computes A in the limit:

$$(\forall x)\,(\exists y)\,(\forall z > y)\,[A(x) = \varphi_e(x, z)].$$

We denote by LimEx the class of computable language classes that can be inferred in the above sense. Reliable LimEx identification is then a variant of the above criterion in which the learner is required to converge to a limiting program for every $A \in \mathcal{A}$ and has to make infinitely many mind changes for all computable $B \notin \mathcal{A}$. Since LimEx satisfies the two conditions above, Theorem 4.1 also holds for learning limiting recursive programs.

Theorem 4.2 *Let $\mathcal{A}$ be in* LimEx*. Then $\mathcal{A}$ is exactly reliably* LimEx *learnable iff $\mathcal{A}$ is one-sided.*

The definitions of reliable learning and bounded mind changes conflict since the first one requires infinitely many mind changes on sets outside $\mathcal{A}$ while the second one requires also in this case finitely many mind changes. This artificial conflict can be removed via permitting a special symbol "?" to indicate non-convergence which also does not count as additional mind change. Indeed one directly receives in the next theorem by translating any guess of the bounded mind change reliable learner to 1 and any output of the symbol "?" to 0: if the learner converges to a guess then the corresponding classifier converges to 1 and if the learner converges to "?" in order to indicate that the data belongs to an input-set outside $\mathcal{A}$ then the classifier converges to 0.

Theorem 4.3 *If $\mathcal{A}$ can be exactly reliably learned by a machine with bounded mind changes of the fourth type, then $\mathcal{A}$ is two-sided classifiable. Furthermore every two-sided class $\mathcal{A} \in$* Ex *is exactly reliably learnable with bounded mind changes of the fourth type.*

5 Classification From Only Positive Data

Gold [18] introduced the notion of identification from text. A text for a language is an infinite sequence of numbers and the symbol "#" such that each element of the language appears at least once and no nonelement of the language ever appears in the sequence. Analogously to Gold's notion of inference, we can define classification from texts: a classifier for $\mathcal{A}$, upon being fed a text for some language A, converges to 1 iff $A \in \mathcal{A}$. As in the case of standard classification, one can define the obvious variants of one-sided and two-sided classification from texts.

Example 5.1 *Every class $\mathcal{F}_\phi$ of all languages satisfying the Boolean formula ϕ is two-sided classifiable from text.*

Proof The classifier is relatively easy and for each input σ evaluates $\phi(range(\sigma))$. Since ϕ accesses the set A only at a finite number of places, all sufficient long $\sigma \preceq T$ for a given text T satisfy $x \in range(\sigma) \Leftrightarrow x \in A$ for the x where ϕ evaluates $A(x)$. For example, if $\phi(A) = (3 \in A \wedge 4 \notin A)$, then all sufficiently large $\sigma \preceq T$ satisfy $3 \in range(\sigma) \Leftrightarrow 3 \in A$ and $4 \in range(\sigma) \Leftrightarrow 4 \in A$. So the result of evaluating ϕ on $range(\sigma)$ for these σ is the same as for evaluating ϕ on A. ∎

Theorem 5.2 *If $\mathcal{A}$ and $\mathcal{B}$ are both two-sided classifiable from text and a finite set belongs to $\mathcal{A}$ iff it belongs to $\mathcal{B}$, then $\mathcal{A} = \mathcal{B}$.*

Proof Assume that $\mathcal{A}$ and $\mathcal{B}$ are both two-sided classifiable from text, that each finite set belongs to $\mathcal{A}$ iff it belongs to $\mathcal{B}$ and that A is an infinite and computable set. Furthermore, let M_1 classify $\mathcal{A}$ and M_2 classify $\mathcal{B}$ from text and let $a_0, a_1, \ldots$ be a recursive enumeration of A. Now define inductively over k a text $T = a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots$ such that $M_1(a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots a_k\#^{n_k})$ $= M_2(a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots a_k\#^{n_k})$ for all k; the numbers n_k must all exist since M_1 and M_2 classify each finite set $\{a_0, a_1, a_2, \ldots, a_k\}$ in the same way and thus converge on each text $a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots a_k\#^{\infty}$ to the same value. So both, M_1 and M_2, take on T infinitely often the same value and both converge on T; therefore both converge to the same limit-value and A is in $\mathcal{A}$ iff A is in $\mathcal{B}$. ∎

One might ask whether the following chain-condition on two-sided classifiable $\mathcal{A}$ must hold.

Whenever an ascending chain $A_0 \subset A_1 \subset \ldots$ belongs to $\mathcal{A}$ so does some infinite set.

It needn't as the following counterexample $\mathcal{A}$ shows. Let

$$\mathcal{A} = \{A : A \cap S = \emptyset\},$$

where S is a simple set, that is, a recursively enumerable set whose infinite complement does not contain any infinite recursive set. This set S has a recursive enumeration $a_0, a_1, \ldots$ and the two-sided classifier M just checks whether the text seen so far intersects an approximation of S:

$$M(\sigma) = \begin{cases} 0 & \text{if } a_k \in range(\sigma) \text{ for some } k \leq |\sigma|; \\ 1 & \text{otherwise.} \end{cases}$$

Now let $\overline{S} = \{b_0, b_1, \ldots\}$ (where the sequence $b_0, b_1, \ldots$ is of course not computable). Then $\{b_0\}, \{b_0, b_1\}, \{b_0, b_1, b_2\}, \ldots$ forms this ascending chain of sets in $\mathcal{A}$. But $\mathcal{A}$ has no infinite member since every infinite and computable set intersects S; non-computable members of $\mathcal{A}$ are not considered in this paper.

Furthermore, Theorem 5.2 does not hold for one-sided classification. An example is $\mathcal{A}$ as the class of all finite sets and $\mathcal{B}$ as the class of all sets. Obviously $\mathcal{B}$ can be classified one-sided from text by always outputting 1. For $\mathcal{A}$ the algorithm is a bit more difficult: $H(\lambda) = 1$ and $H(\sigma w)$ is 1 if $w \in range(\sigma)$ and 0 if $w \notin range(\sigma)$. Thus if the text is for an infinite set, then infinitely often a new element is added and so H outputs infinitely often a 0. If the text is for a finite set, then only finitely often w is a new element and so the classifier converges to 1.

Theorem 5.3 *If $\mathcal{A}$ is one-sided classifiable from text and contains only infinite languages, then $\mathcal{A}$ is void. In particular the class $\mathcal{P}$ of all pattern-languages is not classifiable from text.*

Proof Let H be a classifier for $\mathcal{A}$ and $A = \{a_0, a_1, \ldots\}$ be an infinite set. Then there is a text $T = a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots$ such that $H(a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots a_k\#^{n_k}) = 0$ for all k since H must output on each text $a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\ldots a_k\#^{\infty}$ for each finite set $\{a_0, a_1, \ldots, a_k\}$ infinitely many often a 0. So each set A has a text T such that H outputs on T infinitely many 0s.

The adaptation to $\mathcal{P}$ uses the fact that every pattern language which contains two different elements already is infinite. Thus the construction starts with $a_0a_1\#^{n_1}$ and then proceeds in the same way. ▮

Indeed the construction can be strengthened to prove the existence of some kind of locking-set: If $\mathcal{A}$ can be one-sided classified from text and if $A \in \mathcal{A}$ is infinite, then there is a finite set $F \subseteq A$ such that every computable set B with $F \subseteq B \subseteq A$ belongs to $\mathcal{A}$. Similarly if $\mathcal{A}$ is two-sided classifiable from text, also each set outside $\mathcal{A}$ has a locking set $F \subseteq A$ such that no set B between F and A ($F \subseteq B \subseteq A$) belongs to $\mathcal{A}$. Using this fact it is possible to show that one almost natural property does not hold for classification from text: Infinite one-sided classes do sometimes not have infinite two-sided subclasses.

Theorem 5.4 *The infinite class $\mathcal{A} = \{\{0, 1, \ldots, a\} : a \in \mathbb{N}\}$ is one-sided classifiable from text but every subclass $\mathcal{B} \subseteq \mathcal{A}$ which is two-sided classifiable from text is finite.*

Proof First it is necessary to show that $\mathcal{A}$ is one-sided classifiable from text. This is witnessed by the one-sided classifier M given by $M(\lambda) = 0$ and

$$M(\sigma w) = \begin{cases} 0 & \text{if } w \in range(\sigma) \text{ or } range(\sigma w) \text{ is not of the form } \{0, 1, \ldots, a\}; \\ 1 & \text{otherwise.} \end{cases}$$

Let $\mathcal{B}$ be a subclass of $\mathcal{A}$ which is two-sided classifiable from text. Then $\mathbb{N}$ is not in $\mathcal{B}$ since $\mathbb{N}$ is not in $\mathcal{A}$ and $\mathbb{N}$ has a locking set F. F has a maximum b and so no set $\{0, 1, \ldots, a\}$ with $a \geq b$ is in $\mathcal{B}$; thus $\mathcal{B}$ is finite. ▮

The preceding theorems showed the limitations of classifying from text. So it is suitable to look for a weaker convergence criterion in order to make it possible to classify more realistic classes from text.

Definition 5.5 A machine H classifies a class $\mathcal{A}$ *partially* iff H on any text T for any set A outputs an infinite sequence of numbers such that $A \in \mathcal{A}$ iff exactly one number appears in the output infinitely often and $A \notin \mathcal{A}$ iff no number appears in the output infinitely often.

It is easy to see that every class which can be one-sided classified from texts can also be partially classified from texts. But there are classes which can be partially classified but cannot be one-sided classified from text. Indeed the class of the partially classifiable sets can be classified in terms of its index set. Since the intersection of two Σ_3 sets is a Σ_3 set and since the index set $\{e : W_e \text{ is computable}\}$ is a Σ_3 set, one can ignore those indices e where W_e is not computable. Theorem 5.6 holds indeed even also for all classes of recursively enumerable sets; it is to a certain extent just the counterpart of the fact, that the class of all recursively enumerable sets can be learned from text under the criterion of partial identification [29].

Theorem 5.6 *A class $\mathcal{A}$ is partially classifiable iff its index-set $\{e : W_e \in \mathcal{A}\}$ is Σ_3.*

Proof The sets W_e have all a uniform enumeration $a_{e,0}, a_{e,1}, \ldots$ which might contain the pause symbol # in order to deal with the empty set and indices where it is unknown whether the corresponding set is empty or not. Now the predicate

$$(\exists x)\,(\forall y)\,(\exists z > y)\,[M(a_{e,0}a_{e,1}\ldots a_{e,z}) = x]$$

witnesses that the index-set of $\mathcal{A}$ is Σ_3. For the converse direction one uses the fact that one can make the first quantifyer to have either one or zero solutions. More precisely, there is a formula of the following type:

$$\begin{aligned} W_e \in \mathcal{A} &\;\Leftrightarrow\; (\exists \text{ exactly one } x)\,(\forall y)\,(\exists z)\,[P(e,x,y,z)]; \\ W_e \notin \mathcal{A} &\;\Leftrightarrow\; (\forall x)\,(\exists y)\,(\forall z)\,[\neg P(e,x,y,z)]. \end{aligned}$$

This formula can be obtained by using an m-reduction f from the Σ_3 set $\{e : W_e \in \mathcal{A}\}$ to $\{e : W_e$ is cofinite$\}$ [34, Corollary IV.3.5] and then letting

$$P(e,x,y,z) \;\Leftrightarrow\; x, x+1, \ldots, x+y \in W_{f(e),z} \wedge (x = 0 \vee x-1 \notin W_{f(e),y}).$$

Furthermore, let E be a recursively enumerable set which contains for every set A exactly one index e with $W_e = A$. Such a set exists since one can obtain it as the range of the translation of a Friedberg numbering into the given acceptable numbering. Now one defines that a classifier M outputs a number coding the pair $\langle e, x\rangle$ at least n times iff there is a prefix σ of the text such that

- $|\sigma| \geq n$ and e is enumerated into E within $|\sigma|$ steps;
- $range(\sigma) \cap \{0,1,\ldots,n\} = W_{e,|\sigma|} \cap \{0,1,\ldots,n\}$;
- $(\forall y \leq n)\,(\exists z \leq |\sigma|)\,[P(e,x,y,z)]$.

If now a number representing the pair $\langle e, x\rangle$ appears infinitely often in the output of M, then the first and second conditions imply that the given text is a text for W_e and $e \in E$ while the third condition implies that x witnesses that $W_e \in \mathcal{A}$. So only sets $A \in \mathcal{A}$ can be partially classified and it is necessary to show that they indeed are partially classified.

Let now $A \in \mathcal{A}$ and T be a text for A. Furthermore, let e be the index of $\mathcal{A}$ in E and x be the corresponding unique number such that $(\forall y)\,(\exists z)\,[P(e,x,y,z)]$. Now, for given n, let $\sigma \preceq T$ be so long that e is enumerated into E within $|\sigma|$ stages, all elements of A up to n have appeared in

σ and been enumerated into $W_{e,|\sigma|}$ and every $y \leq n$ has a witness $z \leq |\sigma|$ such that $P(e,x,y,z)$ holds. Then it follows that $\langle e,x\rangle$ is output at least n times and as a consequence, $\langle e,x\rangle$ is output infinitely often. By the preceding paragraph, any further pair $\langle e',x'\rangle$ is output infinitely often only if $e' \in E$, $W_{e'} = A$ and $(\forall y)\,(\exists z)\,[P(e',x',y,z)]$ what implied $e' = e$ and $x' = x$. So the uniqueness of the infinitely often output number is guaranteed and the proof is completed. ∎

Every uniformly recursively enumerable class has a Σ_3 index set, that is, every class for which there is a recursive function f with $\mathcal{A} = \{W_{f(0)}, W_{f(1)}, \ldots\}$. In particular the classes $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{G}$, $\mathcal{P}$, $\mathcal{R}$ and all classes $\mathcal{F}_\phi$ from Example 2.4 can be partially classified. Assume now that M is a one-sided classifier for $\mathcal{A}$ working on informants. Then

$$W_e \in \mathcal{A} \Leftrightarrow (\exists x)\,(\forall y \geq x)\,[M(W_e(0)W_e(1)\ldots W_e(y)) = 1]$$

which is a Σ_3 condition since it is a Σ_2 condition with a K-recursive predicate; the oracle K is used to evaluate $W_e(z)$ for $z = 0,1,\ldots,y$. So $\mathcal{A}$ is partially identifiable. The converse does not hold since the class of all infinite computable languages is not one-sided classifiable from informant but has a Σ_3 index set.

Corollary 5.7 *If $\mathcal{A}$ is one-sided classifiable from informant, then $\mathcal{A}$ is partially classifiable from text. The converse does not hold.*

So partial classification is very powerful. It is even so powerful, that the method of data-presentation does not matter.

Theorem 5.8 *A class $\mathcal{A}$ can be partially classified from text iff $\mathcal{A}$ can be partially classified from informant.*

Proof Only the direction "informant $\Rightarrow$ text" has to be shown, since the other one is obvious. Within this proof, the fact that only the behaviour on the computable sets W_e matters, is crucial — otherwise one could consider a class consisting only of sets of the form $A \oplus K$ where K is the halting problem; this class could be partially classified from informant iff the index-set $\{e : (\exists A)\,[W_e = A \oplus K \wedge W_e \in \mathcal{A}]\,\}$ is Σ_4, which is much more powerful than Σ_3. Now let $\mathcal{A}$ be a class with Σ_3 index set. The following predicate assigns to every e an index of the complement e' and can be satisfied only for computable sets W_e:

$$W_{e'} = \overline{W_e} \Leftrightarrow (\forall x,s)\,(\exists t > s)\,[W_{e',t}(x) + W_{e,t}(x) = 1]$$

Now a set W_e is in $\mathcal{A}$ iff it has a complement $W_{e'}$ and there is an u such that the partial classifier outputs on the characteristic function of W_e — which is verified to be correct using $W_{e'}$ — infinitely often this u. Formally,

$W_e \in \mathcal{A} \Leftrightarrow$ there are e' and u such that

- $(\forall x,s)\,(\exists t > s)\,[W_{e',t}(x) + W_{e,t}(x) = 1]$
- $(\forall y)\,(\exists x \geq y)\,(\exists t)\,[0,1,\ldots,x \in W_{e,t} \cup W_{e',t} \wedge M(W_{e,t}(0)W_{e,t}(1)\ldots W_{e,t}(x)) = u]$.

This is an Σ_3 predicate and thus $\mathcal{A}$ is also partially classifiable from text by Theorem 5.6. ∎

While every class, which is one-sided classifiable from text, either contains or is disjoint to an infinite class which is two-sided classifiable from text, this does not longer hold for partial classification versus one-sided classification.

Theorem 5.9 *There is a class $\mathcal{A}$ partially classifiable from text such that any infinite class $\mathcal{B}$ which is one-sided classifiable from text is neither subclass of $\mathcal{A}$ nor of $\overline{\mathcal{A}}$.*

Proof There is a set $E \leq_T \{e : W_e \text{ is finite}\}$ such that E is biimmune relative K, that is, there is no infinite set B which is enumerable in the limit and either satisfies $B \subseteq E$ or $B \subseteq \overline{E}$. Now the class $\mathcal{A}$ is defined as follows: $\mathcal{A}$ contains all infinite sets and all finite nonempty sets D with $\max(D) \in E$. The index-set of $\mathcal{A}$ is Σ_3, thus $\mathcal{A}$ is partially classifiable from text.

So it remains to show that no infinite one-sided class $\mathcal{B}$ is contained either in $\mathcal{A}$ or in $\overline{\mathcal{A}}$. Assume that $\mathcal{B}$ is an infinite class with one-sided classifier N. If $\mathcal{B}$ contains some infinite set, then $\mathcal{B}$ contains also some finite set F and all finite sets between this F and the infinite set; therefore independent of the fact how many infinite sets $\mathcal{B}$ has, $\mathcal{B}$ has also infinitely many finite sets. The set B given by $B = \{x : (\exists D \in \mathcal{B})\,[x = \max(D)]\}$ is infinite. Furthermore B is enumerable in the limit: x is in B iff there is some text with maximum x on which N converges to 1. Since there is also a sufficient easy text for N iff there is one, B can be described via one existential and one universal quantifier,

$$x \in B \Leftrightarrow (\exists \eta \in \{0, 1, \ldots, x\}^*)\,(\forall k)\,[N(\eta x^k) = 1].$$

The fact that B is enumerable relative to K and infinite implies that B is neither contained in A nor in $\overline{A}$. Thus $\mathcal{B}$ can also be neither a infinite subclass of $\mathcal{A}$ nor of $\overline{\mathcal{A}}$. ∎

A similar result does not hold for one-sided versus two-sided learning from text. Indeed for every one-sided text-classifiable class $\mathcal{A}$ there is a subclass $\mathcal{B}$ of either $\mathcal{A}$ or $\overline{\mathcal{A}}$ which is infinite and two-sided classifiable from text. This is due to the fact that the set $\{x : \{x\} \in \mathcal{A}\}$ is enumerable relative to K and has either a K-recursive infinite subset or is disjoint to a K-recursive infinite set. This set — call it B in both cases — defines the class $\mathcal{B} = \{\{x\} : x \in B\}$ which is infinite, two-sided classifiable from text and is either a subclass of $\mathcal{A}$ or of $\overline{\mathcal{A}}$.

6 Structural Properties of Classification

Soare [34] contains an extensive study on the relation between recursively enumerable and computable sets. As Stephan [35] has already noted, the situation of one-sided versus two-sided classification is similar to that of recursively enumerable versus computable sets. This relationship not only holds in the setting of classifying all sets but also in setting of the present paper of classifying computable sets.

This section shows that if only computable sets are to be classified, then the analogy with recursively enumerable versus computable sets is even more striking. Turing degrees, an important tool for studying recursively enumerable sets, also turn out to be useful in analyzing the complexity of one-sided classification. The next result shows that — similarly to Stephan's general setting [35] — every one-sided class is two-sided relative to a sufficiently complex oracle.

An oracle U is *Turing reducible* to V (written: $U \leq_T V$) iff U can be computed by a machine which has access to a database containing V by the membership-queries "Is $x \in V$?". For an oracle U the relativized halting problem U' to U is defined as $U' = \{e : \varphi_e^U(e)\downarrow\}$.[2] U is *high* iff $K' \leq_T U'$.[3] An alternative characterization is that there is a function u computable relative to

[2] φ_e^U is the e-th partial recursive in U function.

[3] This differs slightly from Soare's definition [34, Definition IV.4.2]: Soare defined "$K' \equiv_T U'$" instead of "$K' \leq_T U'$" since he considers only oracles $U \leq_T K$.

U which dominates every recursive function, that is, which satisfies $(\forall^\infty x)\,[u(x) > f(x)]$ for all $f \in$ REC. Adleman and Blum [1] showed that high oracles play a significant role in inductive inference: REC can be Ex-identifiable relative to U iff U is high. Theorems 6.1 and 6.5 show that the high oracles play a similar special role in classification.

Theorem 6.1 *For each high oracle U, every one-sided class $\mathcal{A}$ has a two-sided classifier which is computable relative to U.*

Proof Let H be a one-sided classifier for a class $\mathcal{A}$ of computable sets. Furthermore let u be a function computable relative to U which dominates every computable function. Now the two-sided classifier is defined as follows where $n_H(\sigma)$ denotes as in Fact 2.2 the number of prefixes $\tau \preceq \sigma$ with $H(\tau) = 0$. The idea is now to repeat each 0 of H a large but finite number of times such that M still converges to 1 if H does but M converges to 0 if H only diverges.

If $u(n_H(\sigma)) > |\sigma|$, then let $M(\sigma) = 0$ else let $M(\sigma) = 1$.

If $A \in \mathcal{A}$, then there is only a finite number n of prefixes $\tau \preceq A$ with $H(\tau) = 0$. Almost all prefixes σ of A have length at least $u(n)$. So $|\sigma| \geq u(n) \geq u(n_H(\sigma))$ and $M(\sigma) = 1$ for these prefixes σ. If $A \notin \mathcal{A}$ and A is computable, then also the function $f_A(n) = \min\{m : n_H(A(0)A(1)\dots A(m)) \geq n\}$ is computable and thus u dominates f_A. There is an n with $u(m) > f(m)$ for all $m \geq n$. In particular whenever a prefix $\sigma \preceq A$ has at least the length $u(n)$, then $u(n_H(\sigma)) > f_A(n_H(\sigma)) \geq |\sigma|$ and $M(\sigma) = 0$. So M converges on every computable set outside $\mathcal{A}$ to 0 and M is two-sided. ▮

Nevertheless there are hard problems, that is, there are one-sided classes $\mathcal{A}$ which require that every two-sided classifier for $\mathcal{A}$ has high Turing degree.

Theorem 6.2 *If M two-sided classifies the class $\mathcal{C} = \{C : C \text{ is cofinite}\}$, then M is not computable and its Turing degree is high.*

Proof The machine $H(\sigma w) = w$ classifies $\mathcal{C}$ one-sided. Assume now that M is a (non-computable) two-sided classifier for $\mathcal{C}$. It is shown that the index-set $I = \{e : W_e \text{ is finite}\}$ can be computed relative to M in the limit and thus the Turing-degree of M must be high. Let $W_{e,s}$ be an uniform recursive class of finite sets enumerating the sets W_e and let

$$A_e(s) = \begin{cases} 0 & \text{if } W_{e,s+1} \neq W_{e,s}; \\ 1 & \text{otherwise, that is, } W_{e,s+1} = W_{e,s}. \end{cases}$$

So A_e is cofinite iff W_e is finite, that is, $I(e) = 1$ iff $A_e \in \mathcal{C}$ iff M converges on A_e to 1. Furthermore, the sets A_e are uniformly recursive and so M converges on every set A_e. So $I(e) = \lim_n M(A_e(0)A_e(1)\dots A_e(n))$ and I is computable in the limit relative to M. The Turing degree of M is high and in particular, M is not computable. ▮

A recursively enumerable set E is called *creative* [34, Definition II.4.3] iff there is an effective procedure which disproves for every e the hypothesis "$W_e = \overline{E}$" by a counterexample $f(e)$, that is, either $f(e) \in \overline{E} - W_e$ or $f(e) \in W_e - \overline{E}$. The name "creative" derives from the fact that such an f creates a new element $f(e) \in \overline{E}$ outside W_e whenever $W_e \subseteq \overline{E}$. This concept is adapted to the context of classifying computable sets.

Definition 6.3 A one-sided classifiable class $\mathcal{A}$ is *creative* iff there is an uniformly computable array $A_0, A_1, \ldots$ such that for each one-sided classifier H_e the set A_e is a counterexample to the hypothesis "H_e classifies $\overline{\mathcal{A}}$".

The next theorem shows that there is a creative class, namely the class of all cofinite sets. So this class is effectively not two-sided.

Theorem 6.4 *The class $\mathcal{C}$ of all cofinite sets is creative.*

Proof Let inductively $A_e(0) = 0$ and $A_e(n+1) = H_e(A_e(0)A_e(1)\ldots A_e(n))$. If H_e converges on this set to 1, then it is cofinite and not in $\overline{\mathcal{C}}$. Otherwise H_e does not converge to 1 and outputs infinitely many 0. Then also A_e is coinfinite and belongs to $\overline{\mathcal{C}}$. So A_e proves that H_e is not an one-sided classifier for the complement of $\mathcal{C}$. $\mathcal{C}$ is creative since its complement is effectively not one-sided. ∎

All creative sets are 1-equivalent to K and have in particular the same Turing degree as K, that is, belong to the greatest recursively enumerable Turing degree. So it is natural to ask how complex are the creative classes. The next theorem shows that there is indeed an analogous result that only the high oracles allow them to be two-sided classified.

Theorem 6.5 *Every creative class, in particular $\mathcal{C}$, is two-sided only relative to high oracles.*

Proof It is easy to code an infinite array of machines $H_{s(e)}$ such that the machines are independent on the actual input A and that $H_{s(e)}$ outputs on any input A infinitely many 0s iff W_e is infinite. This can be achieved easily by

$$H_{s(e)}(\sigma) = \begin{cases} 0 & \text{if } W_{e,|\sigma|+1} \neq W_{e,|\sigma|}; \\ 1 & \text{otherwise, that is, if } W_{e,|\sigma|+1} = W_{e,|\sigma|}. \end{cases}$$

So if W_e is finite, then $H_{s(e)}$ suggests the class to contain all computable sets; and if W_e is infinite, then $H_{s(e)}$ suggests the class to contain no computable set. Thus in the first case, the counterexample has to be outside $\mathcal{B}$ and in $\mathcal{A}$ and in the second case, the counterexample has to be in $\mathcal{B}$ and so outside $\mathcal{A}$. If now some machine M classifies $\mathcal{A}$ two-sided, then M classifies in particular each set $A_{s(e)}$. It follows that

$$\begin{array}{lllll} W_e \text{ is finite} & \Rightarrow & A_{s(e)} \in \mathcal{A} & \Rightarrow & M \text{ converges on } A_{s(e)} \text{ to } 1; \\ W_e \text{ is infinite} & \Rightarrow & A_{s(e)} \notin \mathcal{A} & \Rightarrow & M \text{ converges on } A_{s(e)} \text{ to } 0. \end{array}$$

So using M it can be computed in the limit whether W_e is finite or infinite and thus the Turing degree of M must be high. ∎

While the preceding results mainly dealt with creative classes, this one deals with several degrees of non-creativeness. First it is shown that there are one-sided classes of intermediate complexity: they are two-sided relative to some non-high oracle but not relative to the empty oracle. In particular they are also not creative by Theorem 6.5.

Theorem 6.6 *For each $U \geq_T K$ which is also enumerable relative to K there is a one-sided class $\mathcal{A}$ such that, for every oracle V, there two-sided V-recursive classifier for $\mathcal{A}$ iff U is computable relative to the jump of V. In particular there are intermediate one-sided classes; these are neither two-sided nor creative.*

Proof Let $U \geq_T K$ be enumerable relative to K. Then there is a uniformly computable sequence $U_0, U_1, \ldots$ such that

$$(\forall x)\, [x \in U \Leftrightarrow (\forall^\infty y)[x \in U_y]\,].$$

Now let

$$\mathcal{A} = \{A : A = 0^x 1^\infty \text{ for some } x \in U\}.$$

Using the sequence $U_0, U_1, \ldots$ it is possible to give the following one-sided classifier for $\mathcal{A}$:

$$M(\sigma) = \begin{cases} 1 & \text{if } \sigma = 0^x 1^y \text{ and } x \in U_y; \\ 0 & \text{otherwise.} \end{cases}$$

If $x \in U$, then x is also in almost all U_y and M outputs 1 on almost all inputs $0^x 1^y$, that is, M classifies $0^x 1^\infty$ to be in $\mathcal{A}$. If $x \notin U$, then x is also not in infinitely many U_y and it follows that $M(x^0 1^y) = 0$ for infinitely many y. Thus M classifies $0^x 1^\infty$ not to be in $\mathcal{A}$. It is furthermore easy to see that M classifies every input of a form different from $0^x 1^\infty$ also not to be in $\mathcal{A}$, so M is indeed a one-sided classifier for $\mathcal{A}$.

It is easy to see that relative to any oracle V it is equally difficult to classify two-sided the set $\{x, x+1, \ldots\}$ (with the characteristic function $0^x 1^\infty$) and to compute $U(x)$ in the limit. From this fact it follows that there is some two-sided classifier N computable relative to V iff U is computable relative to V in the limit, that is, iff $U \leq_T V'$. Since there are oracles U which are enumerable in K and between K and K' ($K <_T U <_T K'$), the corresponding classes $\mathcal{A}$ are neither two-sided nor creative. ∎

There are two kinds of immunity-properties for classes:

- For a class $\mathcal{A}$ there is no uniformly computable array $A_0, A_1, \ldots$ of pairwise different sets such that $\{A_0, A_1, \ldots\} \subseteq \mathcal{A}$.
- No infinite two-sided class $\mathcal{B}$ is contained in $\mathcal{A}$.

The following theorems investigate the extent to which one-sided classes and their complements satisfy these requirements. But, the first result shows that a one-sided class and its complement can never be simultaneously immune.

Theorem 6.7 *Let $\mathcal{A}$ be a one-sided class. Then there is an uniformly computable array $A_0, A_1, \ldots$ of pairwise distinct sets such that the class $\mathcal{B} = \{A_0, A_1, \ldots\}$ is two-sided and either $\mathcal{B} \subseteq \mathcal{A}$ or $\mathcal{B} \subseteq \overline{\mathcal{A}}$.*

Proof If there is a string σ such that every computable $A \succeq \sigma$ is in $\mathcal{A}$, then the array given by $A_e = \sigma 0^e 1^\infty$ forms a two-sided subclass of $\mathcal{A}$.

Otherwise it is possible to construct such an array outside $\mathcal{A}$. Let M be a one-sided classifier for $\mathcal{A}$. For each e define $A_e = \lim_n \sigma_n$ by $\sigma_0 = 0^e 1$ and σ_{n+1} being the first proper extension of σ_n with $M(\sigma_{n+1}) = 0$. Since each σ_n is prefix of some $A \notin \mathcal{A}$ there is always such extension σ_{n+1} and the so defined sets A_e are uniformly computable. Furthermore, M outputs on every A_e infinitely often a zero and thus no A_e is in $\mathcal{A}$.

It remains in both cases to show that the class $\mathcal{B}$ is two-sided. The algorithm is in both cases the same, the σ in it is the empty string in the second case and the common prefix in the definition of the A_e in the first case. Let

$$M(\tau) = \begin{cases} 1 & \text{if } \sigma 0^e 1 \preceq \tau \preceq A_e \text{ for some } e; \\ 0 & \text{otherwise.} \end{cases}$$

If the set to be classified is some A_e, then M outputs 1 for all input beyond the prefix $\sigma 0^e 1$ and thus M converges on A_e to 1. Otherwise the set A to be classified is different from all A_e. Either no $\sigma 0^e 1$ is prefix of A and then M always outputs 0 or some prefix $\sigma 0^e 1 \eta$ is a prefix of A but not of A_e. In this case also $M(\tau) = 0$ for all input beyond this prefix. So M always converges to the correct value and is a two-sided classifier for $\mathcal{B}$. ∎

Theorem 6.8 *There is an infinite two-sided class $\mathcal{A}$ which contains no subclass $\mathcal{B} = \{A_0, A_1, \ldots\}$ consisting of a uniformly computable array of pairwise distinct sets.*

Proof Let U be an immune set below K. Then $\mathcal{A} = \{0^x 1^\infty : x \in U\}$ is two-sided but for each uniformly computable array $A_0, A_1, \ldots$ of sets in $\mathcal{A}$ is finite since $\{x : (\exists e)\, [0^x 1 \preceq A_e]\}$ is a recursively enumerable subset of U and therefore finite. ∎

The next theorem states that there is something analogous to simple sets which are recursively enumerable and coinfinite but intersect every infinite computable set.

Theorem 6.9 *There is an infinite one-sided class such that its complement has no two-sided infinite subclass.*

Proof Let U be a set which is enumerable relative to K but whose infinite complement does not have an infinite K-recursive subset, that is, U is a set which is simple relative to K. Now the class

$$\mathcal{A} = \{A : A \cap U \neq \emptyset\}$$

is infinite but its complement does not have an infinite two-sided subclass. ∎

It is well-known that every infinite recursively enumerable set has an infinite computable subset. Stephan [35] showed that this easy observation does not generalize to one-sided classification versus two-sided in his model which requires correct classification of non-computable sets. Furthermore Theorem 5.4 shows something similar for classification from texts. Since the classification of only computable sets from informants is more well-behaved than the two previously mentioned settings, the following problem might still have a positive solution.

Problem Does every infinite one-sided class have an infinite two-sided subclass?

7 Classification By Finding Trial-And-Error Programs

Baliga, Case, Jain, Sharma and Suraj studied in several papers [4, 5, 12] the concept of learning (or using) limiting or mind-changing programs (equivalently, K-recursive programs) instead of ordinary programs for classes of computable functions. This concept transfers quite naturally to classification: Instead of guesses 0 and 1, the classifier produces a sequence of programs such that each of this program converges in the limit to either 0 or 1 which then stands for the guess of the classifier. More formally such a classifier assigns to every input σ a primitive recursive program e such that $L(e) = \lim_n \varphi_e(n)$ exists and is either 0 or 1. As in inductive inference there are two notions of convergence.

- Ex-*style classification*: For every computable set A, the classifier outputs for almost all $\sigma \preceq A$ the same guess e and $L(e) = \mathcal{A}(A)$.
- BC-*style classification*: For every computable set A, the classifier outputs for almost all $\sigma \preceq A$ an index e_σ such that $L(e_\sigma) = \mathcal{A}(A)$.

Theorem 7.1 Ex-*style classification and two-sided classification coincide.*

Proof It is easy to see that outputting a constant 0 or 1 can be transferred into outputting a program which converges in the limit to 0 or 1, respectively. So only the direction to transfer an Ex-style classifier into an two-sided classifier for the same class is interesting. Given an Ex-style classifier M the new two-sided classifier N is defined by $N(\sigma) = \varphi_{M(\sigma)}(|\sigma|)$. Since M always outputs indices of primitive recursive functions, N is total. Assume now that A is computable. Then M outputs for almost all $\sigma \preceq A$ the same index e. Furthermore $\varphi_e(n) = \mathcal{A}(A)$ for almost all n. It follows that $N(\sigma) = \mathcal{A}(A)$ for almost all $\sigma \preceq A$. ∎

Theorem 7.2 *Every one-sided class has a* BC-*style classifier.*

Proof By Theorem 6.1 every one-sided class is classifiable two-sided relative to a high oracle, in particular it has a K-recursive classifier M. By the Limit-Lemma there is a primitive recursive function N such that $M(\sigma) = \lim_x N(\sigma, x)$. Using the substitution-theorem there is a primitive recursive procedure assigning to each σ and index $e(\sigma)$ for the function $f(x) = N(\sigma, x)$. This index $e(\sigma)$ is then the output of the BC-style classifier which classifies the same sets as M. ∎

It is easy to see that the concept of BC-style classification is closed under complementation. Thus the inclusion of one-sided classification into a BC-style classifier is proper. The proof showed that every class which is two-sided relative to the oracle K is already BC-style classifiable. This can be extended to a characterization of BC-style classification by the following theorem.

Theorem 7.3 *For a class $\mathcal{A}$ of computable sets the following is equivalent:*
(a) *$\mathcal{A}$ is* BC-*style classifiable.*
(b) *$\mathcal{A}$ is two-sided relative to K.*
(c) *Both index sets $\{e : W_e$ is computable and $W_e \in \mathcal{A}\}$ and $\{e : W_e$ is computable and $W_e \in \overline{\mathcal{A}}\}$ are Σ_3.*

Proof It suffices to show (a $\Rightarrow$ c) and (c $\Rightarrow$ b).

Given a BC-style classifier M, the following Σ_3 predicate describes the set of all e on which M converges to c:

$$\mathcal{A}(W_e) = c \Leftrightarrow (\exists x)\,(\forall y \geq x)\,(\forall s)\,(\exists t \geq s)\,[\varphi_{M(W_{e,t}(0)W_{e,t}(1)\ldots W_{e,t}(y))}(t) = 1].$$

Since M converges on every computable set $\mathcal{A}$, the intersection of these predicates with a Σ_3 predicate defining that W_e is computable gives the transition from (a) to (c).

For every Σ_3 set E there is a K-recursive predicate P such that $e \in E \Leftrightarrow (\exists x)\,(\forall y)\,[P(e,x,y)]$. Let P_0 and P_1 the corresponding P's for the Σ_3 formulas for membership and non-membership of computable sets W_e in $\mathcal{A}$. Now the K-recursive two-sided classifier M for $\mathcal{A}$ works as follows:

On input σ, mark all (e,x) as disqualified such that either $W_e(y) \neq \sigma(y)$ for some $y \in dom(\sigma)$ or $(\exists y, y' \leq |\sigma|)\,[\neg P_0(e,x,y) \wedge \neg P_1(e,x,y')]$. Find the first pair (e,x) not yet disqualified and output 0 if $(\forall y \leq |\sigma|)\,[P_0(e,x,y)]$ and 1 otherwise.

For any computable set A, the following holds: If $W_e \neq A$ then all pairs (e,x) are eventually disqualified. If $W_e = A$, then there is an x such that either $P_0(e,x,y)$ or $P_1(e,x,y)$ holds for all y. Then this pair (e,x) is never disqualified and thus, for almost all $\sigma \preceq A$, there is the same first pair (e,x) which is not disqualified. So there is a unique $c \in \{0,1\}$ such that $(\forall y)\,[P_c(e,x,y)]$ — this c is just $\mathcal{A}(A)$. If σ is long enough, then also the y with $\neg P_{1-c}(e,x,y)$ is bounded by $|\sigma|$ and $M(\sigma)$ outputs c. So M is a two-sided K-recursive classifier for $\mathcal{A}$. ∎

So it follows that every BC-style classifiable class is also partially identifiable, but the converse does not hold since there are classes $\mathcal{A}$ where $\{e : W_e$ is computable and $W_e \in \mathcal{A}\}$ is Σ_3 but $\{e : W_e$ is computable and $W_e \notin \mathcal{A}\}$ not.

References

[1] Leonard Adleman and Manuel Blum: Inductive Inference and Unsolvability. *Journal of Symbolic Logic* 56 (1991) 891–900.

[2] Andris Ambainis, Sanjay Jain and Arun Sharma: Ordinal Mind Change Complexity of Language Identification, *Proceedings of the Third European Conference on Computational Learning Theory, Jerusalem*, (1997), Lecture Notes in Artificial Intelligence, Springer-Verlag.

[3] Dana Angluin: Inductive Inference of Formal Languages from Positive Data, *Information and Control* 45 (1980) 117–135.

[4] Ganesh Baliga and John Case: Learning with Higher Order Additional Information. *Proceedings of the Fifth International Workshop on Algorithmic Learning Theory, Reinhardsbrunn*, (1994) 64–75.

[5] Ganesh Baliga, John Case, Sanjay Jain and Mandayam Suraj: Machine Learning of Higher Order Programs. *Journal of Symbolic Logic*, 59 (1994) 486–500.

[6] Shai Ben-David: Can Finite Samples Detect Singularities of Real-Valued Functions? *Proceedings of the 24th Annual ACM Symposium on the Theory of Computer Science, Victoria, B.C.*, (1992) 390–399.

[7] Janis Barzdins and Rūsiņš Freivalds: Prediction and Limiting Synthesis of Recursively Enumerable Classes of Functions. In: *Theory of Algorithms and Programs, Vol. 1* (Latvian State University, edited by J. M. Barzdins, Riga 1974) 112–128 (in Russian).

[8] Lenore Blum and Manuel Blum: Toward a mathematical Theory of Inductive Inference. *Information and Control*, 28 (1975) 125–155.

[9] J. Richard Büchi: On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, Standford University Press, Standford, California, 1960.

[10] J. Richard Büchi and Lawrence H. Landweber: Definability in the Monadic Second Order Theory of Successor. *Journal of Symbolic Logic* 34 (1969) 166–170.

[11] John Case, Sanjay Jain and Suzanne Ngo Manguelle: Refinements of Inductive Inference by Popperian and Reliable Machines. *Kybernetika*, 30 (1994) 23–52.

[12] John Case, Sanjay Jain and Arun Sharma: On Learning Limiting Programs. *International Journal of Foundations of Computer Science*, 3 (1992) 93–115.

[13] John Case, Efim Kinber, Arun Sharma and Frank Stephan: On the Classification of Computable Languages. *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, Springer LNCS 1200 (1997) 225-236. Technical Report No. 9603, School of Computer Science and Engineering, The University of New South Wales, Sydney NSW 2052, Australia.

[14] John Case and Carl H. Smith: Comparison of Identification Criteria for Machine Inductive Inference. *Theoretical Computer Science* 25 (1983) 193–220.

[15] Rūsiņš Freivalds and Carl H. Smith: On the Role of Procrastination for Machine Learning, *Information and Computation* 107 (1993) 237–271.

[16] William Gasarch, Mark Pleszkoch, Frank Stephan and Mahendran Velauthapillai: Classification Using Information. *Annals of Mathematics and Artificial Intelligence* 23 (1998) 147–168.

[17] William Gasarch, Mark Pleszkoch and Mahendran Velauthapillai: Classification Using Information — Conference Version. *Proceedings of the Fifth International Workshop on Algorithmic Learning Theory, Reinhardsbrunn*, (1994) 290–300, Lecture Notes in Artificial Intelligence, 872, Springer Verlag, Heidelberg, 1994.

[18] E. Mark Gold: Language Identification in the Limit. *Information and Control*, 10 (1967) 447–474.

[19] Peter G. Hinman: *Recursion-Theoretic Hierarchies.* Springer-Verlag, Heidelberg, 1978.

[20] Sanjay Jain and Arun Sharma: Elementary Formal Systems, Intrinsic complexity, and Procrastination. *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, (1996), 181–192, ACM Press, 1996.

[21] Klaus Peter Jantke: Monotonic and Non-Monotonic Inductive Inference. *New Generation Computing* 8 (1991) 349–360.

[22] Kevin Kelly: *The Logic of Reliable Inquiry.* Oxford University Press, Oxford, 1995.

[23] Ker-I Ko: *Complexity Theory of Real Functions.* Birkhäuser, Boston, 1991.

[24] Lawrence H. Landweber: Decision Problems for ω-Automata. *Mathematical Systems Theory*, 3 (1969) 376–384.

[25] Robert McNaughton: Testing and Generating Infinite Sequences by a Finite Automaton. *Information and Control* 9 (1966) 434–448.

[26] Eliana Minicozzi: Some Natural Properties of Strong Identification in Inductive Inference. *Theoretical Computer Science* 2 (1976) 345–360.

[27] Maurice Nivat and Dominique Perrin (editors): *Automata on Infinite Words.* Lecture Notes to Computer Science 192, Springer-Verlag, Heidelberg, 1984.

[28] Piergiorgio Odifreddi: *Classical recursion theory.* North-Holland, Amsterdam, 1989.

[29] Daniel N. Osherson, Michael Stob and Scott Weinstein: *Systems that learn.* Bradford / MIT Press, London, 1986.

[30] Emil Post. Recursively enumerable sets of positive integers and their decision problems, *Bulletin of the American Mathematical Society*, 50:284–316, 1944.

[31] Hartley Rogers, Jr.: *Theory of Recursive Functions and Effective Computability.* McGraw-Hill Book Company, New York, 1967.

[32] Gerald E. Sacks: *Higher Recursion Theory*, Perspectives in Mathematical Logic, Springer-Verlag, Heidelberg, 1990.

[33] Arun Sharma, Frank Stephan and Yuri Ventsov: Generalized notions of mind change complexity. *Proceedings of the 10th Conference on Computational Learning Theory* (1997) 96–108.

[34] Robert I. Soare: *Recursively enumerable sets and degrees.* Springer-Verlag, Heidelberg, 1987.

[35] Frank Stephan: On One-Sided Versus Two-Sided Classification. *Archive for Mathematical Logic*, to appear.

[36] Alan M. Turing: On Computable Numbers with an Application to the Entscheidungsproblem. *Proc. London Math. Society* 42 (1936) 230–265.

[37] Rolf Wiehagen and Carl H. Smith: Generalization versus Classification. *Journal of Experimental and Theoretical Artificial Intelligence*, 7, 1995. Shorter version in *Proceedings 5th Annual Workshop on Computational Learning Theory*, (1992) 224–230. ACM Press, New York.

[38] Carl H. Smith, Rolf Wiehagen and Thomas Zeugmann: Classifying Predicates and Languages. *International Journal of Foundations of Computer Science* 8 (1997) 15–42.

[39] Boris A. Trakhtenbrot: Finite Automata and the Logic of One Place Predicates. *Siberian Mathematical Journal* 3 (1962) 103–131 [in Russian].

[40] Thomas Zeugmann and Steffen Lange: A Guided Tour Across the Boundaries of Learning Recursive Languages. *Algorithmic Learning for Knowledge-Based Systems* (K. P. Jantke and S. Lange, Eds.), Lecture Notes in Computer Science 961 (1995) 193–262.

[41] Thomas Zeugmann, Steffen Lange and Shyam Kapur: Characterizations of Monotonic and Dual Monotonic Language Learning, *Information and Computation* 120 (1995) 155–173.