Learning and Classifying

Sanjay Jain*[1], Eric Martin[2] and Frank Stephan**[3]

[1] Department of Computer Science,
National University of Singapore, Singapore 117417, Republic of Singapore.
sanjay@comp.nus.edu.sg
[2] School of Computer Science and Engineering,
The University of New South Wales, Sydney NSW 2052, Australia.
emartin@cse.unsw.edu.au
[3] Department of Mathematics and Department of Computer Science,
National University of Singapore, Singapore 117543, Republic of Singapore.
fstephan@comp.nus.edu.sg

Abstract. We define and study a learning paradigm that sits between identification in the limit and classification. More precisely, we expect a learner to determine in the limit which members of a finite set D of possible data belong to a target language L, where D is arbitrary. So as D becomes larger and larger, the task becomes closer and closer to identifying L. But as D is always finite and L can be infinite, it can still be expected that Ex- and BC-learning are often more difficult than performing this classification task. The paper supports this intuition and makes it precise, taking into account desirable constraints on how the learner behaves, such as bounding the number of mind changes and being conservative. Special attention is given to various forms of consistency. In particular, we might not only require consistency between the members of D to classify, the current data σ and a language L, but also consistency between larger sets of possible data to classify (supersets of D) and the same σ and L: whereas in the classical paradigms of inductive inference or classification, only the available data can grow, here both the available data and the set of possible data to classify can grow. We provide a fairly comprehensive set of results, many of which are optimal, that demonstrate the fruitfulness of the approach and the richness of the paradigm.

Keywords. Classification of predicates; Multiclassification; Learning in the limit; Inductive inference.

1 Introduction

A main purpose of the field of inductive inference is to study whether it is possible to identify a device (usually a Turing machine or one of its equivalents) from an enumeration of the data

* Supported in part by NUS grants C252-000-087-001 and R252-000-420-112.
** Supported in part by NUS grant R252-000-420-112.

it can produce. The identification is to be performed by a mechanical procedure, by analysing growing finite sequences of data, where the mechanical procedure may be subjected to various constraints in its computing abilities. (See [BB75,CS83,Gol67,JORS99,ZZ08] for example.) Two of the important success criteria for learning are based on convergence to a correct answer from some point onwards, when enough data has been seen. The first criterion is that a syntactical description of the device has been correctly discovered and will be issued from the point of convergence onwards (this success criterion is called explanatory or Ex-identification, [Gol67]). The second criterion is that the behaviour of the device has been correctly identified and that from the point of convergence onwards, a syntactical description of the device will be issued but this syntactical description may vary in the face of new data (this success criterion is called behaviourally correct or BC-identification, [Bār74b,CL82,CS83,OW82]).

Another line of research, which has received substantial though considerably less attention from the community, consists in discovering in the limit some property of the device that generates the data being analysed. This line of research is called *classification*. For an overview on classification, see for example [FH94,WS95,SWZ97,CKSS04,GSPM98,Ste01,Jai01,JMS08]. Classification can naturally be generalised to *multiclassification*, defined as the task of finding out whether the device that generates the data to be observed has property P, for any P in a given set of properties. Learning is, in some sense, the case of synchronous multiclassification for (usually infinitely many) disjoint classification tasks: it amounts to discovering (the description of) a device characterized as having or not the property of producing d, for any potential datum d. Therefore, it has to be expected that learning is, in many cases, more difficult than classification. This is still not as obvious as might first appear as we deal with full classification and not *positive* only classification: the work of a classifier is to output either 1 or 0 to indicate whether, on the basis of the sequence of data currently observed, the property holds or does not hold, respectively.

The simplest property one might conceive of is that of membership of some datum to the set of data that the device can produce. The task is trivial with one mind change at most, as a classifier just has to output 0 until the datum appears, if ever. Even if the focus is on the simple properties expressed as data membership, more complex tasks naturally come to mind. In this paper, we will investigate the task of multiclassification—but we will still talk about a "classifier" rather than "multiclassifier". For instance, the task to classify a device in one of the four classes determined by whether or not it produces the datum 0 and whether or not it produces the datum 1 might require two mind changes sometimes, unless the presence of one number guarantees the presence of the other in the set of data that can be generated by any of the target devices. A natural generalisation of this idea then requires from the classifier that it deals with any finite set of basic classification tasks, be they of the form of data membership or more complex ones. For instance, one might want to consider Boolean combinations of data membership of a certain kind as basic classification tasks.

We mentioned that learning is a kind of synchronous multiclassification, where each classification task expresses the property of generating or not some set of data, for all possible such sets. More precisely, learning can be seen as multiclassification of the basic property of data membership, taking together the infinitely many possible data. So what we are describing is a framework where learning is somehow the limit of multiclassification, where the number of basic classification tasks taken together, and required to be finite, grows towards infinity: identifying a device is the limiting case of discovering whether 0, 1, ..., n belong to the set of data that can be generated by that device when n becomes larger and larger. This is the adequate setting, as considering classification tasks individually is trivial, as we have observed already, whereas considering infinitely many tasks together is too hard and too close to the limiting case of considering all predicates. One of the aims of this paper is to make this precise and investigate its consequences. In other words, we examine to which extent successful multiclassification makes learning possible.

Besides just considering classification, we will also consider the effects of requiring the classification process to be consistent or conservative. Formal definitions of these is given in Section 2. Informally, consistent classification requires that after receiving data σ for some language in the class, the classification done by the classifier should be consistent with at least one language L in the class for which σ is a valid data. We also consider a variant called strong consistency, in which we require that not only is the classification consistent, but for any input data σ, for large enough classification task, the classification is based on the same language L' (which may be different from the L mentioned above). This language L' may or may not be in the class under consideration, giving two versions of this stronger consistency requirement. Besides consistency, we also consider conservativeness, where the classifier is not allowed to change its classification unless it sees data which ensures that no language in the class is consistent with the classification.

We proceed as follows. In Section 2, we introduce the background notation and notions, illustrated with a few examples. This covers the classical concepts of finite, Ex- and BC-learnability, and the new concept of classification, possibly constrained by counterparts to the usual notions of mind changes and conservativeness, and three different notions of consistency. Section 3 presents negative results on BC-learnability, showing that it is harder than classification constrained in various ways. In particular, Theorem 8 gives a class which is classifiable using no mind changes but not BC-learnable. Theorems 9 to 11 show similar result for various versions of consistency and conservativeness (where, the mind changes needed for classification is one or two, based on the exact versions of consistency used). The number of mind changes in these results is optimal, as for lower number of mind changes the separations no longer hold. In contrast, Section 4 shows how Ex-learnability can be obtained from various classification strategies. In particular, Theorem 13 shows that the classes which can be strongly consistently and conservatively classified are explanatorily learnable. Section 5 investigates in more depth the relationships between the various forms of classification. In particular, Theorem 16 shows the advantages of conservativeness over consistency, Theorem 17 shows the advantages of strong consistency over strong consistency within class, Theorem 18 shows the advantages of just using consistency over strong consistency. Theorem 21 shows that in general, even for classes which have a consistent classifier, the number of mind changes can be quite big, as a function of the number of predicates to be classified.

2 Definitions and examples

2.1 General notation and learnability

We denote by $\mathbb{N}$ the set of natural numbers. The length of a finite sequence σ is denoted by $\mathrm{len}(\sigma)$. Given a sequence σ, for $i < \mathrm{len}(\sigma)$, $\sigma(i)$ represents the ith member of σ (starting with the 0th element); for $i \leq \mathrm{len}(\sigma)$, $\sigma_{|i}$ represents the initial segment of σ of length i. We use $\diamond$ to denote concatenation between finite sequences, and identify a sequence of length 1 with its unique member. Given $x, i \in \mathbb{N}$, we denote by x^i the unique member of $\{x\}^i$. Given a set E, we denote by $E^\star$ the set of finite sequences of members of E. Given $E \subseteq \mathbb{N}$, we denote by χ_E the characteristic function of E.

We fix an acceptable enumeration $(\varphi_e)_{e\in\mathbb{N}}$ of the partial computable functions over $\mathbb{N}$, and for all $e \in \mathbb{N}$, denote by W_e the domain of φ_e (see for example [Rog67]). We fix a computable coding of all finite sequences of members of $\mathbb{N}$ into $\mathbb{N}$ and denote by $\langle n_1, \ldots, n_k\rangle$ the image of $(n_1, \ldots, n_k)$ under this coding. The coding is chosen such that $\langle n_1, \ldots, n_k\rangle \leq \langle m_1, \ldots, m_h\rangle$ whenever $k \leq h$, $n_1 \leq m_1, \ldots, n_k \leq m_k$.

A language is a set of the form W_e. $\mathcal{L}$ ranges over classes of languages. Given a language L, a *text for L* is an infinite enumeration of the members of L, possibly with repetitions of some members of L and possibly with occurrences of #, the "pause" symbol [Gol67]. A *text* is a text for some language. Given a text t, $t_{|i}$ represents the initial segment of t of length i. Given a recursive function f, the *canonical text for f* is the infinite sequence whose ith term, $i \in \mathbb{N}$, is $\langle i, f(i)\rangle$. We denote $(\mathbb{N} \cup \{\#\})^\star$ by SEQ. Given $\sigma \in \mathrm{SEQ}$, $\mathrm{cnt}(\sigma)$ denotes the set of natural numbers that occur in σ. A member σ of SEQ is said to be *for* a language L if $\mathrm{cnt}(\sigma) \subseteq L$, and *for* $\mathcal{L}$ if σ is for some member of $\mathcal{L}$.

A *learner* is a partial computable function from SEQ into $\mathbb{N} \cup \{?\}$ (where ? is a distinguished symbol which allows the learner to express that it makes no hypothesis). Given a language L, a learner M *Ex-learns* L, see [Gol67], iff for all texts t for L, there exists an $e \in \mathbb{N}$ such that (i) $M(\sigma)$ is defined for all finite initial segments σ of t, (ii) $W_e = L$ and (iii) $M(\sigma) = e$ for cofinitely many finite initial segments σ of t. A learner *Ex-learns* $\mathcal{L}$ iff it Ex-learns all members of $\mathcal{L}$. Given a language L, a learner M *BC-learns* L, see [Bār74b,CL82,OW82], iff for all texts t for L, (i) $M(\sigma)$ is defined for all finite initial segments σ of t and (ii) $W_{M(\sigma)} = L$ for cofinitely many finite initial segments σ of t. A learner *BC-learns* $\mathcal{L}$ iff it BC-learns all members of $\mathcal{L}$. Given a language L, a learner M *finitely learns* L, see [Gol67], iff M Ex-learns L and for all texts t for L and for all finite initial segments σ_1 and σ_2 of t, if $M(\sigma_1)$ and $M(\sigma_2)$ are both distinct from ? then they are equal. A learner *finitely learns* $\mathcal{L}$ iff it finitely learns all members of $\mathcal{L}$. Given $c \in \mathbb{N}$, we say that a learner M that Ex-learns $\mathcal{L}$ *makes at most c mind changes*, see [CS83], iff there is no infinite sequence t of members of $\mathbb{N} \cup \{\#\}$ and no strictly increasing sequence $(i_0, \ldots, i_c)$ of $c+1$ integers such that $M(t_{|i_0}) \neq\ ?$ and $M(t_{|i_k}) \neq M(t_{|i_k+1})$ for all $k \leq c$; when $c = 0$ we rather say that M *makes no mind change*.

2.2 Classification

A *predicate* is a property that can be true or false for any given language (1 or 0 is then the *truth value of* that predicate *for* that language, respectively). Given a sequence τ of predicates, a language L and a sequence τ' of members of $\{0, 1\}$, we say that τ' is *the sequence of truth values of* τ *for* L if τ' has the same length as τ and for all $i < \text{len}(\tau)$, $\tau'(i)$ is the truth value of $\tau(i)$ for L. For $n \in \mathbb{N}$, let $\in_n$ be the predicate that is true for languages that contain n, and false for others. Let In denote the set of all predicates of the form $\in_n$, $n \in \mathbb{N}$. Let Boole denote the set of all predicates that are Boolean combinations of predicates in In.

Definition 1. Let P be a set of predicates.

A *general P-classifier* is a partial function M on $\text{SEQ} \times P^\star$ such that for all $\sigma \in \text{SEQ}$, for all $n \in \mathbb{N}$ and for all $\tau \in P^n$, $M(\sigma, \tau)$ is a member of $\{0, 1\}^n \cup \{?\}$.

A *P-classifier* is a partial computable general P-classifier.

When P is clear from the context, we simply write (*general*) *classifier* for (*general*) *P-classifier*.

Definition 2. Let P be a set of predicates and let C be a general P-classifier.

Given a language L, we say that C *P-classifies* L iff for all $\tau \in P^\star$ and all σ for L, $C(\sigma, \tau)$ is defined and the following holds. For any text t for L and any $\tau \in P^\star$, for cofinitely many finite initial segments σ of t, $C(\sigma, \tau)$ is the sequence of truth values of τ for L.

We say that C *P-classifies* $\mathcal{L}$ iff C P-classifies all members of $\mathcal{L}$.

When P is clear from the context, we simply write *classifies* for *P-classifies.*

Given a set P of predicates, it is useful to refer to any finite sequence of members of P as a *P-classification task.* To say that a general P-classifier C is successful on a P-classification task should be clear from Definition 2.[4] Again, when P is clear from the context, we write more simply *classification task* for *P-classification task.*

Classification, especially when requiring the classifier to decide whether the input language belongs to one of several disjoint classes of languages, has been studied by several authors earlier, see [FH94,WS95,SWZ97,CKSS04,Ste01,Jai01,JMS08] for example. The definitions above are about multiclassification, in which the classifier has to simultaneously perform several classification tasks.

We now consider the important constraint of consistency. In the context of this paper, and as opposed to the classical notion of consistency used for learnability (see [Ang80,Bār74a,JB81,WL76]), it is natural to impose that the decisions on the predicates to be dealt with be consistent not only

[4] Formally, it means that if τ denotes the classification task then $C(\sigma, \tau)$ is defined for all σ for $\mathcal{L}$, and for all texts t for a member L of $\mathcal{L}$, for cofinitely many finite initial segments σ of t, $C(\sigma, \tau)$ is the sequence of truth values of τ for L (so C P-classifies $\mathcal{L}$ iff C is successful on all P-classification tasks).

with the available data (which can always be easily achieved when deciding a finite number of predicates), but also with a language in the class under consideration. As classifiers have to deal with arbitrarily large finite sets of predicates, one can further consider whether the classification, on a particular input, is compatible with some fixed language L, provided that the predicate set is large enough: otherwise, it may seem like the classifier is changing its classification when larger and larger sets of questions from P are asked, even when the data provided is not changed. This language L may or may not be required to be in the class of languages under consideration. The difference is captured in the next definition by the notions of strong consistency within class and strong consistency, respectively. In this definition, the only difference between C being *strongly consistent on* $\mathcal{L}$ and C being *strongly consistent within class on* $\mathcal{L}$ is that the former refers to a language L, whereas the latter refers to a language $L \in \mathcal{L}$.

Notions of conservativeness [Ang80] and mind changes [CS83] will also play an important role in this paper, but they are straightforward adaptations of the classical notions considered in inductive inference. The definition below formalises both notions as well as the various concepts of consistency previously discussed.

Definition 3. Let a class of languages $\mathcal{L}$, a set of predicates P, and a general P-classifier C be given.

We say that C is *consistent on* $\mathcal{L}$ iff for all $\sigma \in \mathrm{SEQ}$ and $\tau \in P^\star$, if σ is for $\mathcal{L}$ then there exists $L \in \mathcal{L}$ such that σ is for L and $C(\sigma, \tau)$ is the sequence of truth values of τ for L.

We say that C is *strongly consistent on* $\mathcal{L}$ iff C is consistent on $\mathcal{L}$ and for all $\sigma \in \mathrm{SEQ}$, if σ is for $\mathcal{L}$ then there exists a language L and a $\tau \in P^\star$ such that σ is for L and for all $\tau' \in P^\star$ such that the set of predicates in τ is contained in the set of predicates in τ', $C(\sigma, \tau')$ is the sequence of truth values of τ' for L.

We say that C is *strongly consistent within class on* $\mathcal{L}$ iff C is consistent on $\mathcal{L}$ and for all $\sigma \in \mathrm{SEQ}$, if σ is for $\mathcal{L}$ then there exists an $L \in \mathcal{L}$ and a $\tau \in P^\star$ such that σ is for L and for all $\tau' \in P^\star$ such that the set of predicates in τ is contained in the set of predicates in τ', $C(\sigma, \tau')$ is the sequence of truth values of τ' for L.

(Based on [Ang80]) We say that C is *conservative on* $\mathcal{L}$ iff for all $\sigma \in \mathrm{SEQ}$, $x \in \mathbb{N} \cup \{\#\}$ and $\tau \in P^\star$, if there exists $L \in \mathcal{L}$ such that $\sigma \diamond x$ is for L and $C(\sigma, \tau)$ is the sequence of truth values of τ for L then $C(\sigma, \tau) = C(\sigma \diamond x, \tau)$.

When $\mathcal{L}$ is clear from the context, we omit "on $\mathcal{L}$" in the previous expressions.

Given a set P of predicates, a general P-classifier C that classifies $\mathcal{L}$ and $c \in \mathbb{N}$, the notion C *makes at most c mind changes* is defined by adapting in a straightforward manner the corresponding definition for learners from [CS83] recalled at the end of Section 2.1.

2.3 Illustrative examples

The following first examples give an idea how classification works and show some straightforward relations to learning theory.

Example 4. *Assume that $\mathcal{L}$ is some superclass of the class of all finite sets. Then a consistent and conservative* In-*classifier C for $\mathcal{L}$ works as follows: $C\big(\sigma, (\in_{n_0}, \in_{n_1}, \ldots, \in_{n_k})\big) = (\chi_S(n_0), \chi_S(n_1), \ldots, \chi_S(n_k))$, where $S = \mathrm{cnt}(\sigma)$. Note that the class of all finite sets is Ex-learnable; however, whenever $\mathcal{L}$ is a proper superset of the class of all finite sets, then $\mathcal{L}$ is neither Ex-learnable nor BC-learnable [Gol67]. So classifiability does not imply learnability.*

Assume that $\mathcal{L}$ consists of all sets with exactly 5 elements. Then $\mathcal{L}$ has an In-*classifier C which works as follows: If $S = \mathrm{cnt}(\sigma)$ has exactly 5 elements, then, $C\big(\sigma, (\in_{n_0}, \in_{n_1}, \ldots, \in_{n_k})\big) = (\chi_S(n_0), \chi_S(n_1), \ldots, \chi_S(n_k))$; otherwise $C\big(\sigma, (\in_{n_0}, \in_{n_1}, \ldots, \in_{n_k})\big) = ?$. This* In-*classifier does not make any mind change when the input text is for some language in $\mathcal{L}$. However, no consistent* In-*classifier can classify $\mathcal{L}$ with four mind changes or less, as consistent classifiers cannot take the value ?. The same trade-off applies for Ex-learners versus consistent Ex-learners (if it were the case that ? is forbidden for consistent Ex-learners as well).*

The next example deals with nonrecursive sets. It shows that the lack of computability necessarily leads to mind changes by the classifier. In subsequent results, we will mostly avoid such examples and obtain separations of the notions via classes consisting of recursive sets only.

Example 5. *Assume that $\mathcal{L} = \{A\}$ for some r.e. nonrecursive set A. While $\mathcal{L}$ is Ex-learnable by the trivial learner which always outputs the same correct hypothesis for A and thus never makes a mind change, no* In-*classifier can classify $\mathcal{L}$ with a bounded number of mind changes: If $\mathcal{L}$ were* In-*classifiable with n mind changes then one could feed a recursive enumeration of A as a text t into the classifier and, for every $\tau \in \mathrm{In}^{n+1}$, enumerate up to $n+1$ sequences, at least one of which is a sequence of truth values of τ for A. This contradicts Beigel's Nonspeedup Theorem and Kummer's Cardinality Theorem [GM98]: informally, Kummer's cardinality theorem says that a set A is recursive if, for any $m \geq 1$, there exists a Turing machine which, for every $x_1, x_2, \ldots, x_m$, enumerates at most m elements, at least one of which is the cardinality of $\{x_1, x_2, \ldots, x_m\} \cap A$. Hence $\mathcal{L}$ is not* In-*classifiable with n mind changes, for any n.*

The fact that a successful classifier of a learnable class does not immediately provide a learner of that class is strengthened in the next example. It exhibits a class of languages $\mathcal{L}$ and a consistent and conservative In-classifier that classifies $\mathcal{L}$ using at most one mind change, such that for all finite initial segments of a text for some member of $\mathcal{L}$, the classifier is incorrect on infinitely many In-classification tasks.

Example 6. *For all $i \in \mathbb{N}$, let $L_i = \{2j \mid j \leq i\} \cup \{2i+1\}$ and suppose that $\mathcal{L}$ is equal to $\{2\mathbb{N}\} \cup \{L_i \mid i \in \mathbb{N}\}$. Let C be an* In-*classifier such that for all $\sigma \in \mathrm{SEQ}$ for $\mathcal{L}$ and members k, $n_0, \ldots, n_k$ of $\mathbb{N}$, the following holds.*

– *If there exists $i \in \mathbb{N}$ with $2i+1 \in \mathrm{cnt}(\sigma)$ then $C\big(\sigma, (\in_{n_0}, \ldots, \in_{n_k})\big) = (\chi_{L_i}(n_0), \ldots, \chi_{L_i}(n_k))$.*
– *Otherwise, if the greatest number in $\{n_0, \ldots, n_k\}$, say n, is both odd and greater than any number in $\mathrm{cnt}(\sigma)$, then for all $j \leq k$, $C\big(\sigma, (\in_{n_0}, \ldots, \in_{n_k})\big)(j) = 1$ iff n_j is even or $n_j = n$.*
– *Otherwise, for all $j \leq k$, $C\big(\sigma, (\in_{n_0}, \ldots, \in_{n_k})\big)(j) = 1$ iff n_j is even.*

It is easy to see that C In-classifies $\mathcal{L}$, is consistent and conservative, and makes at most one mind change; also, for all $\sigma \in \mathrm{SEQ}$, if σ is for $2\mathbb{N}$ then there exist infinitely many sequences of the form $(\in_{n_0}, \ldots, \in_{n_k})$ such that $C\big(\sigma, (\in_{n_0}, \ldots, \in_{n_k})\big) \neq (\chi_{2\mathbb{N}}(n_0), \ldots, \chi_{2\mathbb{N}}(n_k))$.

3 Classification versus BC-learnability

We first provide a preliminary result on BC-learnability which will be useful to establish our first results on multiclassification.

Given a set L, the retraceable set R_L determined from L is defined as follows. Write L as $\{x_0, x_1, x_2, \ldots\}$ where $x_0 < x_1 < x_2 \ldots$; then, $R_L = \{\langle x_0, x_1, \ldots, x_n\rangle \mid n \leq \text{cardinality of } L\}$.

Proposition 7. *Let $e \in \mathbb{N}$ be given. Let $\mathcal{L}_e = \{L \mid \min(L) = e, L \text{ is recursive and infinite}\}$. Then $\mathcal{L}'_e = \{R_L \mid L \in \mathcal{L}_e\}$ is not BC-learnable.*

Proof. BC-learnability of $\mathcal{L}'_e$ implies BC-learnability of the class of infinite recursive languages from informant, which is false [CL82]. □

We start our investigations with three results on Boole-classification. Almost all other results deal with In-classification.

Our first result shows that some class can be classified using no mind changes, even though it cannot be BC-learnt. So as hypothesised in the abstract, identification can be harder than multiclassification, even when the latter is the easiest one.

Theorem 8. *There exists a class $\mathcal{L}$ of languages that some* Boole-*classifier classifies making no mind change, but no learner BC-learns $\mathcal{L}$.*

Proof. Let $\mathcal{L} = \{R_L \mid L \text{ is an infinite recursive language}\}$. Then $\mathcal{L}$ is not BC-learnable by Proposition 7.

Let C be a Boole-classifier defined as follows. $C((), \tau) =?$. For $\sigma \neq ()$ and $\tau \in \mathrm{Boole}^\star$, C first computes the maximum n such that $\in_n$ is used in one of τ's predicates. If σ is not for any member of $\mathcal{L}$ or if $\mathrm{cnt}(\sigma)$ does not contain an element $\langle x_1, x_2, \ldots, x_m\rangle$ with $m > n$, then $C(\sigma, \tau) = ?$; otherwise, $C(\sigma, \tau)$ is the sequence of truth values of τ for R_L, for any set L such that σ is for R_L (note that all such L give the same output for $C(\sigma, \tau)$). It is easily verified that C classifies $\mathcal{L}$ making no mind change. □

Our next result shows that some class can be classified strongly consistently within class, using a non-recursive classifier, by making at most one mind change, even though it cannot be BC-learnt. Note that this is optimal as if it allows no mind change, then the output of a consistent classifier on input $((), \tau)$ must be correct. So the same lesson holds as for Theorem 8, this time with classifiers whose classifications are better behaved, but which are not computable.

Theorem 9. *There exists a class $\mathcal{L}$ of languages such that some strongly consistent within class general* Boole-*classifier classifies $\mathcal{L}$ making at most one mind change, but no learner BC-learns $\mathcal{L}$.*

Proof. Let $M_0, M_1, \ldots$ denote an effective enumeration of all partial computable learners. For all $e \in \mathbb{N}$, let L_e be an infinite recursive set with $\min(L) = e$ such that R_{L_e} is not BC-learnt by M_e. Note that for all $e \in \mathbb{N}$, there exists such an L_e by Proposition 7. Let $\mathcal{L} = \{\emptyset\} \cup \{R_{L_e} \mid e \in \mathbb{N}\}$. Then, clearly $\mathcal{L}$ is not BC-learnable as no learner M_e BC-learns $R_{L_e} \in \mathcal{L}$.

Define a general classifier C as follows. $C(\sigma, \tau)$ is the sequence of truth values of τ in $\emptyset$, if σ is for $\emptyset$ or σ is not for any R_{L_e}. Otherwise, $C(\sigma, \tau)$ is the sequence of truth values of τ in R_{L_e}, where e is the unique number such that σ is for R_{L_e} (note that unless $\mathrm{cnt}(\sigma) = \emptyset$, σ can be for at most one R_{L_e}). It is easy to verify that C classifies $\mathcal{L}$. □

Our next result shows that some class can be classified strongly consistently within class, using a partial-recursive classifier, by making at most two mind changes, even though it cannot be BC-learnt. Note that this is optimal as if only one mind change is allowed, then Corollary 15 shows that such a result is not possible. So compared with Theorem 9, there is a cost of one mind change for multiclassification to be computable.

Theorem 10. *There exists a class $\mathcal{L}$ of languages such that some strongly consistent within class* Boole-*classifier classifies $\mathcal{L}$ making at most two mind changes, but no learner BC-learns $\mathcal{L}$.*

Proof. Let $M_0, M_1, \ldots$ denote an effective enumeration of all partial computable learners. Let $\mathcal{L} = \{\mathbb{N}\} \cup \{R_L \mid L \text{ is infinite}\}$. Then $\mathcal{L}$ is not BC-learnable by Proposition 7.

Define a classifier C as follows. If $\sigma = ()$ or if σ is not for any R_L, then $C(\sigma, \tau)$ is the sequence of truth values of τ in $\mathbb{N}$. For $\sigma \neq ()$ and $\tau \in \mathrm{Boole}^\star$, C first computes the maximum n such that $\in_n$ is used in one of τ's predicates. If $\mathrm{cnt}(\sigma)$ does not contain an element $\langle x_1, x_2, \ldots, x_m \rangle$ with $m > n$, then $C((), \tau)$ is the sequence of truth values of τ in $\mathbb{N}$; otherwise, $C(\sigma, \tau)$ is the sequence of truth values of τ for R_L, for any set L such that σ is for R_L (note that all such L give the same output for $C(\sigma, \tau)$). It is easily verified that C classifies $\mathcal{L}$ making at most two mind changes. □

We now focus on In-classification. The last result of this section exhibits a case where BC-learnability fails whereas consistent and conservative classification succeeds, even when the latter is constrained to using very few mind changes. Note that the number of mind changes for classification in the below theorem cannot be improved due to Theorem 13. So compared with Theorem 10, the classifiers under consideration are not so well behaved in terms of consistency, but enjoy an additional good behaviour, that of conservativeness.

Theorem 11. *There exists a class $\mathcal{L}$ of languages such that some consistent and conservative* In-*classifier classifies $\mathcal{L}$ making at most two mind changes, but no learner BC-learns $\mathcal{L}$.*

Proof. By the operator recursion theorem [Cas74], there exists a 1–1 recursive function $p : \mathbb{N}^2 \to \mathbb{N}$ that is increasing in both arguments and such that the following holds.

Let $\mathcal{L} = \{W_{p(i,j)} \mid W_{p(i,j)} \text{ is infinite}\}$.

Let $M_0, M_1, \dots$ denote an effective enumeration of all partial computable learners. For each $i \in \mathbb{N}$, we construct $W_{p(i,\cdot)}$ as follows (the construction is run separately for each i, using variables that are implicit functions of i). The procedure is very similar to the one used by Case and Smith [CS83] to exhibit a class of functions that is not learnable by any BC-learner.

Initially, enumerate $\langle 0, p(i,0)\rangle$ in $W_{p(i,0)}$. Let $s \in \mathbb{N}$ be given, and let $W^s_{p(i,0)}$ denote $W_{p(i,0)}$ as enumerated up to the start of stage s. Let $x_0 = 0$. Let σ_0 denote $\big(\langle 0, \varphi_{p(i,0)}(0)\rangle\big)$. We now describe stage s, for $s = 0, 1, \dots$, of the construction of $W_{p(i,\cdot)}$. An element is enumerated in $W_{p(i,\cdot)}$, only if it is done via these stages (or the initialization above).

Stage s

1. Let j_s be such that (i) $j_s > s$, (ii) j_s is larger than any j such that $\langle x, p(i,j)\rangle$ is in $W^s_{p(i,0)}$, and (iii) $j_s >$ the number of computation steps that have taken place to define $W_{p(i,0)}$ up to now.
2. Enumerate $\langle x_s + 1, p(i, j_s)\rangle$ into $W_{p(i,0)}$. Let $W_{p(i,j_s)} = W^s_{p(i,0)} \cup \{\langle x, p(i,j_s)\rangle \mid x \geq x_s\}$.
3. For all $x_s \leq y$, $y \in \mathbb{N}$, let γ^y_s denote $\sigma_s \diamond \big(\langle x, p(i,j_s)\rangle\big)_{x_s \leq x \leq y}$.
4. Search for $y > x_s$ such that $W_{M_i(\gamma^y_s)}$ contains $\langle y+1, p(i,j_s)\rangle$.
   If and when such a y is found, enumerate $\{\langle x, p(i,j_s)\rangle \mid x_s < x \leq y\}$ into $W_{p(i,0)}$, let $x_{s+1} = y$ and $\sigma_{s+1} = \gamma^y_s$ and go to stage $s+1$.

End stage s

We now show that for all $i \in \mathbb{N}$, M_i does not BC-learn $\mathcal{L}$. Let $i \in \mathbb{N}$ be given and consider the construction above for $W_{p(i,\cdot)}$ for the corresponding i. We consider two cases.

- Case 1: There exists a (unique) stage s that starts but does not finish. Then for all $y > x_s$, $\langle y+1, p(i,j_s)\rangle \notin W_{M_i(\gamma^y_s)}$. Thus M_i does not BC-learn $W_{p(i,j_s)}$.
- Case 2: All stages finish. Then $W_{p(i,0)}$ is infinite. Now for each $s \in \mathbb{N}$ and for $y = x_{s+1}$, $\langle y+1, p(i,j_s)\rangle \in W_{M_i(\gamma^y_s)} \setminus W_{p(i,0)}$. Here note that $\gamma^y_s = \sigma_{s+1}$ for all $s \in \mathbb{N}$, and $\bigcup_{s\in\mathbb{N}} \sigma_s$ is a text for $W_{p(i,0)}$. Thus M_i does not BC-learn $W_{p(i,0)}$.

This completes the proof that no learner BC-learns $\mathcal{L}$. We now define a consistent and conservative In-classifier C and show that it classifies $\mathcal{L}$. Note that each language $W_{p(i,j)}$, for $j > 0$ is either $\emptyset$ or a recursive language (a decision procedure for which can be effective found from i, j, in case we know that $W_{p(i,j)}$ is not empty); $W_{p(i,0)}$ is also recursive, though we may not be able to effectively find a decision procedure for $W_{p(i,0)}$ from i.

Let $\sigma \in \mathrm{SEQ}$ be for $\mathcal{L}$, and let $\tau \in \mathrm{In}^\star$ be given. Define $C(\sigma, \tau)$ as follows. Note that for all $i, j \in \mathbb{N}$, one can determine whether $\langle x, p(i,j)\rangle$ is in $W_{p(i,0)}$ for some x, by running the construction above for j steps (see step 1 in the construction above). If $\mathrm{cnt}(\sigma) = \emptyset$, then C outputs a sequence of nothing but 0's. Otherwise, C determines the unique i such that $\mathrm{cnt}(\sigma) \subseteq \{\langle x, p(i,j)\rangle \mid x, j \in \mathbb{N}\}$. Let j be largest such that $\in_{\langle x,p(i,j)\rangle}$ is in τ for some x. C first determines the stage s which the construction of $W_{p(i,\cdot)}$ above reaches in time j. Then, C outputs the sequence of truth values of τ in the language L defined as follows.

- If $\mathrm{cnt}(\sigma) \subseteq W_{p(i,j_s)}$, then $L = W_{p(i,j_s)}$.
- If $\mathrm{cnt}(\sigma) \not\subseteq W_{p(i,j_s)}$ but there exists (a necessarily unique) $j \in \{j_0, j_1, \ldots, j_{s-1}\}$ such that $\mathrm{cnt}(\sigma) \subseteq W_{p(i,j)}$, then $L = W_{p(i,j)}$. Note that in this case, $W_{p(i,j)}$ is the only language in $\mathcal{L}$ which is consistent with σ.
- If $\mathrm{cnt}(\sigma) \not\subseteq W_{p(i,j)}$ for all $j \in \{j_0, j_1, \ldots, j_s\}$, then j_{s+1} must be defined. Then $L = W_{p(i,j_{s+1})}$. Note that in this case, the sequence of truth values of τ for $W_{p(i,j_{s+1})}$ is the same as the sequence of truth values of τ for $W_{p(i,0)}$ (if it is infinite). Thus the sequence of truth values of τ for $W_{p(i,j_{s+1})}$ is consistent with σ (irrespective of which $W_{p(i,\cdot)}$ is the input language, as long as it is not $W_{p(i,j)}$, $j \in \{j_0, j_1, \ldots, j_s\}$).

It is easy to verify that the above classifier is conservative and makes at most two mind changes. Note that if the input σ is not for $W_{p(i,j)}$, for any $j \in \{j_0, j_1, \ldots, j_s\}$, where i and s are as defined above, then for σ to be for a language in the class, j_{s+1} must be defined (otherwise, all $W_{p(i,j)}$, $i, j \in \mathbb{N}$ are either finite or not consistent with the input). Furthermore, for all x such that $\in_x$ is a predicate in τ, $x \in W_{p(i,0)}$ iff $x \in W_{p(i,j_{s+k})}$ for any $k \geq 1$ such that $W_{p(i,j_{s+k})}$ is not empty. Thus, the above classifier classifies $\mathcal{L}$ and is consistent on $\mathcal{L}$. $\square$

4 Classification versus finite and Ex-learnability

The first result of this section exhibits a class of languages that is easy to learn, as it is learnable with no mind change, whereas classification requires sometimes to go through all possibilities of making n predicates true or false before converging to the correct answer. So it is not always true that learning is harder than multiclassification.

Theorem 12. *There exists a class $\mathcal{L}$ of finite languages such that some learner finitely learns $\mathcal{L}$ and some consistent and conservative* In*-classifier classifies $\mathcal{L}$. Moreover, for all consistent* In*-classifiers C and for all $n \in \mathbb{N}$, there is $\tau \in \mathrm{In}^n$ and a text t for $\mathcal{L}$ such that $\{C(t_{|i}, \tau) \mid i \in \mathbb{N}\}$ has cardinality 2^n.*

Proof. Consider a 1–1 computable mapping from $\{(i, n) \mid 0 < n, i < n\}$ into $2\mathbb{N}$ and for all $n > 0$ and $i < n$, denote by $x_{i,n}$ the image of (i, n) under this mapping. Consider a 1–1 computable mapping from $\{(I, n) \mid 0 < n, I \subseteq \{0, \ldots, n-1\}\}$ into $2\mathbb{N}+1$ and for all $n > 0$ and $I \subseteq \{0, \ldots, n-1\}$, denote by $y_{I,n}$ the image of (I, n) under this mapping. Set

$$\mathcal{L} = \big\{\{x_{i,n} \mid i \in I\} \cup \{y_{J,n} \mid J \neq I\} \;\big|\; n > 0,\, I \subseteq \{0, \ldots, n-1\}\big\}.$$

Trivially, some learner finitely learns $\mathcal{L}$ by outputting ? until, for some $n > 0$ and $I \subseteq \{0, 1, \ldots, n-1\}$, all elements of the form $y_{J,n}$, $J \neq I$, appear in the input; at which point the learner outputs an index for $\{x_{i,n} \mid i \in I\} \cup \{y_{J,n} \mid J \neq I\}$.

Let an In-classifier C be defined as follows. Let a member σ of SEQ be for $\mathcal{L}$. If $\mathrm{cnt}(\sigma) = \varnothing$ then for all $\tau \in \mathrm{In}^\star$, $C(\sigma, \tau)$ is the sequence of $\mathrm{len}(\tau)$ 0's. Suppose that $\mathrm{cnt}(\sigma) \neq \varnothing$, and let n be the unique nonzero natural number such that all members of $\mathrm{cnt}(\sigma)$ are of the form $x_{i,n}$ or $y_{J,n}$. Let I be the lexicographically first member of the powerset of $\{0, \ldots, n-1\}$ (identified with $\{0,1\}^n$) such that σ is for $L = \{x_{i,n} \mid i \in I\} \cup \{y_{J,n} \mid J \neq I\}$. Note that L can be effectively determined from σ. Then for all $m \in \mathbb{N}$, $\tau \in \mathrm{In}^m$ and $j < m$, let $C(\sigma, \tau)(j)$ be the truth value of $\tau(j)$ for L. It is immediately verified that C is consistent and conservative and classifies $\mathcal{L}$.

Now let C be a consistent In-classifier, and let $n \in \mathbb{N} \setminus \{0\}$ be given. Let τ denote $(\in_{x_{0,n}}, \ldots, \in_{x_{n-1,n}})$. By consistency of C, there exists a unique enumeration $(I_m)_{m<2^n}$ of the powerset of $\{0, \ldots, n-1\}$ such that for all $m < 2^n$ and for all $j < n$, $C\big((y_{I_0,n}, \ldots, y_{I_{m-1},n}), \tau\big)(j) = 1$ iff $x_{j,n} \in I_m$. Then $\sigma = (y_{I_0,n}, \ldots, y_{I_{2^n-2},n})$ is for $\{x_{i,n} \mid i \in I_{2^n-1}\} \cup \{y_{J,n} \mid J \neq I_{2^n-1}\}$ and for all $j < n$, $C(\sigma, \tau)(j) = 1$ iff $x_{j,n} \in I_{2^n-1}$; therefore, $\{C(\sigma_{|i}, \tau) \mid i < 2^n\}$ is of cardinality 2^n. $\square$

The next results in this section show how to construct an Ex-learner from a classifier constrained in the maximum number of mind changes it is allowed to make, and by consistency and conservativeness requirements. It is the first result which does not only compare the power of multiclassification to the power of learning, but reveals a deeper relationship between both.

Theorem 13. *Let $\mathcal{L}$ be a class of languages that some strongly consistent and conservative* In-*classifier classifies making at most k mind changes for some k in $\mathbb{N}$. Then some learner Ex-learns $\mathcal{L}$. Moreover, all members of $\mathcal{L}$ are recursive.*

Proof. For all $n \in \mathbb{N}$, set $\tau_n = (\in_0, \ldots, \in_n)$. Let C be a strongly consistent and conservative In-classifier that classifies $\mathcal{L}$ making at most k mind changes. Define a learner M as follows. Let $\sigma \in \mathrm{SEQ}$ be given. Then $M(\sigma)$ is an integer e such that for all $n \in \mathbb{N}$, $n \in W_e$ iff either $n \leq \mathrm{len}(\sigma)$ and $C(\sigma, \tau_{\mathrm{len}(\sigma)})(n) = 1$, or $n > \mathrm{len}(\sigma)$ and $C(\sigma, \tau_n)(n) = 1$. Note that for all $e \in \mathbb{N}$ such that $M(\sigma) = e$, W_e is recursive, a decision procedure for which can be effectively obtained from σ. Furthermore, for learners outputting total decision procedures, BC-learning and Ex-learning coincide. Thus, to complete the proof of the theorem, it suffices to verify that M BC-learns $\mathcal{L}$.

Let a text t for $L \in \mathcal{L}$ be given. We determine a finite sequence $(\sigma_i)_{i \leq k}$ of increasing finite initial segments of t in such a way that we can then prove that for all finite initial segments σ of t which extend σ_k, $M(\sigma)$ is an index for L. Intuitively, we force a mind change by C for each transition from σ_i to σ_{i+1}, for large enough τ_n.

The definition of $(\sigma_i)_{i\leq k}$ is performed inductively together with the definition a sequence $(n_i)_{i\leq k}$ of natural numbers such that for all $i < k$, $\mathrm{len}(\sigma_i) \leq n_i \leq \mathrm{len}(\sigma_{i+1})$. Let σ_0 be the empty sequence, and let $n_0 \in \mathbb{N}$ be such that for all $n \geq n_0$ and $n' \geq n$, $C(\sigma_0, \tau_n)$ is an initial segment of $C(\sigma_0, \tau_{n'})$ (such an n_0 exists since C is strongly consistent). Let $i < k$ be given and assume that for all $j \leq i$, σ_j and n_j have been defined.

- If for all initial segments σ of t and all n such that $n \geq \mathrm{len}(\sigma) \geq n_i$, $C(\sigma, \tau_n) = C(\sigma_i, \tau_n)$, then clearly $M(\sigma)$ is an index for L, for all initial segments σ of t with $\mathrm{len}(\sigma) \geq n_i$.
- So suppose that there exists a finite initial segment σ of t and n such that $n \geq \mathrm{len}(\sigma) \geq n_i$ and $C(\sigma, \tau_n) \neq C(\sigma_i, \tau_n)$. Since C is conservative, $C(\sigma_i, \tau_n)$ is not the sequence of truth values of τ_n for L', for any $L' \in \mathcal{L}$ which contains $\mathrm{cnt}(\sigma)$. So since C is consistent, $C(\sigma_i, \tau_n)$ is not an initial segment of $C(\sigma, \tau_{n'})$ for any $n' \geq n$. We then set $\sigma_{i+1} = \sigma$ and let $n_{i+1} \geq n$ be such that for all $n'' \geq n_{i+1}$ and $n''' \geq n''$, $C(\sigma_{i+1}, \tau_{n''})$ is an initial segment of $C(\sigma_{i+1}, \tau_{n'''})$ (such an n_{i+1} exists since C is strongly consistent.) Note that for all $m \geq n_{i+1}$, $C(\sigma_{i+1}, \tau_m) \neq C(\sigma_i, \tau_m)$: this follows from the respective definitions of σ_i, σ_{i+1}, n_i and n_{i+1}, the fact that $n_i \leq n \leq n_{i+1}$, and the fact that $C(\sigma_i, \tau_n)$ is an initial segment of $C(\sigma_i, \tau_m)$, but not of $C(\sigma_{i+1}, \tau_m)$.

Since C makes no more than k mind changes, we conclude that for all finite initial segments σ of t that extend σ_k and are of length greater than n_k, $M(\sigma)$ is an index of the language L. $\square$

Now we examine whether the behaviour of a classifier can be improved "for free". If at most one mind change is allowed, then indeed a better form of consistency can be achieved from the weakest one.

Theorem 14. *Let $\mathcal{L}$ be a class of languages such that some consistent* In-*classifier classifies $\mathcal{L}$ making at most one mind change. Then some strongly consistent* In-*classifier classifies $\mathcal{L}$ making at most one mind change.*

Proof. Let C be a consistent In-classifier that classifies $\mathcal{L}$ making at most one mind change on any text, for any τ. Given $i \in \mathbb{N}$, let τ_i denote $(\in_0, \ldots, \in_i)$, and let A_i be the set of all $j \leq i$ with $C\big((), \tau_i\big)(j) = 1$. Let B denote the set of all members of $\bigcup \mathcal{L}$ that do not belong to A_i for infinitely many $i \in \mathbb{N}$, and let E denote the set of members of $\bigcup \mathcal{L}$ that belong to A_i for all but finitely many $i \in \mathbb{N}$. We will first make an observation on the relationship between B and $\mathcal{L}$, then use it to show that E is r.e., and then prove that E is actually recursive. This together with another observation, this time on the relationship between $\mathcal{L}$ and E, will allow us to define a strongly-consistent In-classifier which classifies $\mathcal{L}$ making at most 1 mind change.

Given $x \in \mathbb{N}$ and $i \geq x$, let S^i_x denote $C\big((), \tau_i\big)$ if $x \in A_i$, and $C\big((x), \tau_i\big)$ otherwise. First note that for all $x \in B$, there exists a unique language L_x in $\mathcal{L}$ such that $x \in L_x$, and for all $i \geq x$, S^i_x is the sequence of truth values of τ_i for L_x. Indeed, for $x \in B$, since C makes at most one mind change, $C\big((x),\tau_i\big)$ is necessarily the sequence of truth values of τ_i for L_x for the infinitely many $i \in \mathbb{N}$ such that $x \notin A_i$; this uniquely determines L_x. Since C is consistent, $C\big((),\tau_i\big)$ is then necessarily the sequence of truth values of τ_i for L_x for all $i \in \mathbb{N}$ such that $x \in A_i$.

We first show that E is r.e. There is nothing to prove if E is recursive, so assume otherwise for a contradiction. For another contradiction, suppose that there exists $x \in E$ such that for all $i \geq x$, S_x^i is an initial segment of S_x^{i+1}. Since $S_x^i = C\big((),\tau_i\big)$ for cofinitely many $i \in \mathbb{N}$, it follows from the definition of A_i, $i \in \mathbb{N}$, that for all $y \geq x$, either $y \in E$ and $S_x^y(y) = 1$, or $y \notin E$ and $S_x^y(y) = 0$, which contradicts the hypothesis that E is not recursive. Thus, for all $x \in E$, there exists an $i \geq x$ such that S_x^i is not an initial segment of S_x^{i+1}. Together with the observation, in the previous paragraph, on the members of B, this implies that E is the set of all $x \in \bigcup_{i\in\mathbb{N}} A_i$ such that S_x^i is not an initial segment of S_x^{i+1} for some $i \geq x$, hence E is r.e.

We now show that E is recursive. First, consider the case that there exists a sequence $(x_i)_{i\in\mathbb{N}}$ of pairwise distinct members of E and a sequence $(n_i)_{i\in\mathbb{N}}$ of members of $\mathbb{N}$ such that for all $i \in \mathbb{N}$, $n_i \geq x_i$ and $x_i \notin A_{n_i}$. One can assume without loss of generality that the sequences $(x_i)_{i\in\mathbb{N}}$ and $(n_i)_{i\in\mathbb{N}}$ are recursive, where $x_i < x_{i+1}$. For all $i \in \mathbb{N}$, set $S^i = C\big((x_i),\tau_{n_i}\big)$. Then for all $i \in \mathbb{N}$, S^i is the sequence of truth values of τ_{n_i} in every language in $\mathcal{L}$ that contains x_i; moreover, since x_i belongs to A_j for cofinitely many $j \in \mathbb{N}$, S^i is an initial segment of $C\big((),\tau_j\big)$ for cofinitely many $j \in \mathbb{N}$. Hence for all $i,j \in \mathbb{N}$, one of S^i and S^j is an initial segment of the other. Furthermore, (a) for all $x \in E$, for all $i \geq x$, $S^i(x) = 1$, as otherwise, x does not belong to A_j for cofinitely many $j \in \mathbb{N}$ and (b) for all $y \in B$ with $y \leq n_i$, $S^i(y) = 0$, as otherwise, y belongs to A_j for cofinitely many $j \in \mathbb{N}$. Thus E is the language L such that for all $i \in \mathbb{N}$, S^i is the sequence of truth values of τ_{n_i} for L. Moreover, since the sequence $(S^i)_{i\in\mathbb{N}}$ is r.e., E is recursive. On the other hand, if there exists no infinite sequence $(x_i)_{i\in\mathbb{N}}$ of pairwise distinct members of E and corresponding sequence $(n_i)_{i\in\mathbb{N}}$ of members of $\mathbb{N}$ such that for all $i \in \mathbb{N}$, $n_i \geq x_i$ and $x_i \notin A_{n_i}$, then $\bigcup_i A_i \cap B$ is r.e. (as $\bigcup_i A_i \cap B$ consist of all $x \in \bigcup_i A_i$ such that, for some $j \geq x$, $x \notin A_j$ and $x \notin E$; here note that only finitely many members x of E satisfy that for some $j \geq x$, $x \notin A_j$). Thus E is recursive, as both E and $\{i : i \in A_i\} \setminus E = \{i : i \in A_i\} \cap (\bigcup_i A_i \cap B)$ are recursively enumerable, and the recursive set $\{i : i \in A_i\}$ contains all but finitely many elements in E.

Now observe that no member L of $\mathcal{L}$ is strictly included in E. Otherwise, there would be a member σ of SEQ with $\mathrm{cnt}(\sigma) \subseteq L$ and $i \in \mathbb{N}$ such that $x \in E \cap A_i \setminus L$ but $C(\sigma,\tau_i)(x) = 0$. But then there exists a $\sigma' \in$ SEQ extending σ such that $\mathrm{cnt}(\sigma') \subseteq E$ and $C(\sigma',\tau_i)(x) = 1$. Furthermore, since $x \in A_i$, we have $C((),\tau_i)(x) = 1$. Thus, $C((),\tau_i) \neq C(\sigma,\tau_i) \neq C(\sigma',\tau_i)$, a contradiction to C making at most 1 mind change.

We can now define a strongly-consistent In-classifier C' as follows. Let $\sigma \in$ SEQ be for $\mathcal{L}$, and let $\tau \in \mathrm{In}^\star$ be given. If σ is for E then $C'(\sigma,\tau)$ is the sequence of truth values of τ for E. Otherwise, pick least $x \in \mathrm{cnt}(\sigma)$ that does not belong to E, and then $C'(\sigma,\tau)$ is the sequence of truth values of τ for L_x, the unique member of $\mathcal{L}$ that contains x. (By the definition of L_x, $x \in B$, $C'(\sigma,\tau)$ can be effectively determined using the sequences S_x^i, $i \geq x$.) Obviously, C' classifies $\mathcal{L}$ with at most one mind change. $\square$

As a corollary to the previous theorem, we can obtain another result which, similarly to Theorem 13, reveals a relationship between multiclassification and learning.

Corollary 15. *Let $\mathcal{L}$ be a class of languages. If some consistent* In*-classifier classifies $\mathcal{L}$ making at most one mind change then some learner Ex-learns $\mathcal{L}$.*

Proof. Note that every consistent In-classifier which makes at most one mind change is also a conservative In-classifier which makes at most one mind change. Thus, the corollary follows from Theorems 13 and 14. $\square$

5 Limitations of classification variously constrained

The results in this section have the flavour of the standard results in inductive inference that compare how various constraints imposed on the learners affect their power to learn as opposed to other constraints. These matters are investigated here in the context of our notion of classification.

The next four results consider classifiers which are conservative, perhaps consistent in some way and possibly restricted in the maximal number of mind changes they can afford, and show that they can sometimes classify classes which cannot be classified by classifiers not required to be conservative, but subjected to more stringent conditions on consistency or on mind changes.

Theorem 16. *There exists a class $\mathcal{L}$ of languages such that some conservative* In*-classifier classifies $\mathcal{L}$ using at most one mind change, but no consistent* In*-classifier classifies $\mathcal{L}$.*

Proof. Let $(C_e)_{e\in\mathbb{N}}$ denote an effective enumeration of all partial computable In-classifiers. For all $e \in \mathbb{N}$, let L_e denote $\big\{\langle e,0\rangle\big\}$ if $C_e\big((\langle e,0\rangle),(\langle e,1\rangle)\big)$ is defined and equal to (1), and $\big\{\langle e,0\rangle,\langle e,1\rangle\big\}$ otherwise. Set $\mathcal{L} = \big\{L_e \bigm| e \in \mathbb{N}\big\}$. One can easily define a conservative In-classifier, which starts with ?, and classifies $\mathcal{L}$ (using at most one mind change). On the other hand, for all $e \in \mathbb{N}$, C_e is not a consistent In-classifier for $\mathcal{L}$. $\square$

Theorem 17. *There exists a class $\mathcal{L}$ of languages such that some conservative and strongly consistent* In*-classifier classifies $\mathcal{L}$ using at most one mind change, but no strongly consistent within class* In*-classifier classifies $\mathcal{L}$.*

Proof. Let $(C_e)_{e\in\mathbb{N}}$ denote an effective enumeration of all partial computable In-classifiers. For all $i \in \mathbb{N}$, let S_i, σ_i and τ_i denote $\{2x \mid x \le i\}$, $(0,2,\dots,2i)$ and $(\in_0,\in_1,\dots,\in_{2i-1},\in_{2i})$, respectively. We inductively define for all $e \in \mathbb{N}$ an integer n_e and a language L_e; it will be the case that $n_e < n_{e+1}$. $\mathcal{L}$ will then consist of all these L_e. Intuitively, each language L_e would be of the form $S_i \cup \{2i+1\}$, for some i; we call this i, n_e below.

Let $e \in \mathbb{N}$ be given, and assume that for all $e' < e$, $n_{e'}$ has been defined. Let k be the least integer greater than $n_{e'}$ for all $e' < e$. If there exists $i \geq k$ and $j \geq i$ such that for all $j' \geq j$, $C_e(\sigma_k, \tau_{j'})$ is defined and is the sequence of truth values of $\tau_{j'}$ for $S_i \cup \{2i+1\}$ then we set $n_e = i+1$ and $L_e = S_{i+1} \cup \{2i+3\}$; otherwise, we set $n_e = k$ and $L_e = S_{n_e} \cup \{2n_e+1\}$. Let $\mathcal{L}$ consist of all sets of the form L_e, $e \in \mathbb{N}$. It is immediately verified that for all $e \in \mathbb{N}$, C_e is not strongly consistent within class by the definitions of L_e and n_e.

Let C be an In-classifier defined as follows. Let $\sigma \in \mathrm{SEQ}$ be for $\mathcal{L}$. If no odd number occurs in $\mathrm{cnt}(\sigma)$ then for all $\tau \in \mathrm{In}^*$ and for all $i < \mathrm{len}(\tau)$, $C(\sigma,\tau)(i) = 1$ iff $\tau(i)$ is of the form $\in_{2j}$. If some odd number n occurs in σ then for all $\tau \in \mathrm{In}^*$ and for all $i < \mathrm{len}(\tau)$, $C(\sigma,\tau)(i) = 1$ iff either $\tau(i) = \in_n$ or $\tau(i) = \in_{2j}$ for $2j < n$. It is easy to verify that C is conservative and strongly consistent and classifies $\mathcal{L}$ using at most one mind change. □

The above diagonalization is optimal, as any consistent classifier using no mind changes outputs only correct classifications (for any language in the class) on empty σ. It is thus also strongly consistent within class making no mind change. Note that by Theorem 14, the following is also optimal.

The next theorem should be considered in parallel with Theorem 11 as both assess the existence of a class $\mathcal{L}$ of languages which some consistent and conservative In-classifier classifies making at most two mind changes. But we diagonalise against the class of BC-learns in one case and the class of strongly consistent In-classifiers in the other case, which resulted in two totally different choices of $\mathcal{L}$ to witness both results.

Theorem 18. *There exists a class $\mathcal{L}$ of languages such that some consistent and conservative* In-*classifier classifies $\mathcal{L}$ making at most two mind changes, but no strongly consistent* In-*classifier classifies $\mathcal{L}$.*

Proof. For the sake of defining $\mathcal{L}$, we will define $F_e = \bigcup_s F_e^s$, where $F_e \subseteq \{\langle e, 4x\rangle \mid x \in \mathbb{N}\}$. F_e^s would satisfy the following constraints:

(PA) $F_e^0 = \{\langle e, 0\rangle\}$
(PB) F_e^s can be obtained effectively from e and s,
(PC) $F_e^s \subseteq F_e^{s+1}$,
(PD) $F_e^s \subseteq \{\langle e, 4x\rangle \mid x \leq s\}$,
(PE) $F_e \cap \{\langle e, 4x\rangle \mid x \leq s\} \setminus F_e^s$ has at most one element.

We will later define $i_e \in \{1, 3\}$ based on the behaviour of $C_e((\langle e, 0\rangle), \tau)$ for larger and larger τ. Let $S_e^s = F_e^s \cup \{\langle e, 4s + i_e\rangle\}$.

Let $\mathcal{L} = \{\emptyset\} \cup \{S_e^s \mid e, s \in \mathbb{N}\} \cup \{F_e \mid e \in \mathbb{N} \text{ and } F_e \text{ is finite}\}$.

For classifying $\mathcal{L}$, $C(\sigma, \tau)$ is defined as follows.

(1) If $\mathrm{cnt}(\sigma) = \emptyset$, then $C(\sigma, \tau)$ is the sequence of truth values of τ for $\emptyset$.

(2) Otherwise, let $e \in \mathbb{N}$ be such that $\mathrm{cnt}(\sigma) \subseteq \{\langle e, x\rangle \mid x \in \mathbb{N}\}$ (if there is no such e, then clearly, σ is not for $\mathcal{L}$, and thus the output of C is irrelevant).
(3) Let m be maximal such that $\in_{\langle e,m\rangle}$ is in τ.
(4) If $\mathrm{cnt}(\sigma)$ contains $\langle e, 4s+i\rangle$ for some $s \in \mathbb{N}$ and $i \in \{1, 3\}$, then $C(\sigma, \tau)$ is the sequence of truth values of τ for $F_e^s \cup \{\langle e, 4s+i\rangle\}$ where s, i are such that $\langle e, 4s+i\rangle \in \mathrm{cnt}(\sigma)$ (note that such an s and i are unique if σ is for $\mathcal{L}$).
(5) Otherwise, $C(\sigma, \tau)$ is the sequence of truth values of τ for $F_e^m \cup \mathrm{cnt}(\sigma)$.

Note that the mind change in (4) and (5) are conservative for $\mathcal{L}$, as in (4) there is unique language in $\mathcal{L}$ which contains $\langle e, 4s+i\rangle$, and any change in classification due to (5) is clearly conservative. Also, for σ for $\mathcal{L}$, the classification as in (4) and (5) is clearly consistent. Furthermore, C classifies $\mathcal{L}$, as for any text t for $L \in \mathcal{L}$, for large enough initial segment σ of t such that $\mathrm{cnt}(\sigma) = L$, if $L = S_e^s$ then (4) will succeed and if $L = F_e$, then the answer as in (5) is correct.

To consider the number of mind changes made by the classifier we argue as follows. For each $L \in \mathcal{L}$, for some e, one of the following holds:

(i) $L = F_e \supseteq F_e^m$ and $F_e \setminus F_e^m$ contains at most one element from $\{\langle e, x\rangle \mid x \leq m\}$.

(ii) $L = S_e^s$, for some $s \leq m$, and $(S_e^s \setminus F_e^m) \cap \{\langle e, x\rangle \mid x \leq m\} \subseteq \{\langle e, 4s+i_e\rangle\}$.

(iii) $L = S_e^s \supseteq F_e^m$, for some $s > m$, and $(S_e^s \setminus F_e^m) \cap \{\langle e, x\rangle \mid x \leq m\} \subseteq F_e \setminus F_e^m$, which contains at most one element from $\{\langle e, x\rangle \mid x \leq m\}$.

(i) above holds by property (PE) for F_e. For (ii), note that $F_e^s \subseteq F_e^m$. Thus, the only element in $S_e^s \setminus F_e^m$ is $\langle e, 4s+i_e\rangle$.

For (iii), note that $S_e^s \setminus F_e^m \subseteq (F_e \setminus F_e^m) \cup \{\langle e, 4s+i_e\rangle\}$. As $4s+i_e > m$ and $F_e \setminus F_e^m$ has at most one element from $\{\langle e, x\rangle \mid x \leq m\}$ (by property (PE) for F_e), (iii) holds.

It immediately follows that C, for input texts for non-empty languages from $\mathcal{L}$ makes at most two mind changes. One from outputing a classification for $\emptyset$ to outputing classification for F_e^m, and then maybe to the final correct answer.

We are now ready to define F_e^s, $e, s \in \mathbb{N}$. Let $(C_e)_{e\in\mathbb{N}}$ denote an effective enumeration of all partial computable In-classifiers. For each e, we will define F_e^s, $s \in \mathbb{N}$ along with $i_e \in \{1, 3\}$ so as to make sure that C_e does not strong consistently classifies $\mathcal{L}$.

Let $F_e^0 = \{\langle e, 0\rangle\}$. Let τ_r denote $(\in_0, \in_1, \ldots, \in_r)$. Inductively, if we have already defined F_e^s but not F_e^{s+1}, then we define F_e^{s+1} as follows.

(S1) If there exists an odd number q and s'' such that $\langle e, q\rangle \leq s'' \leq s$ and within s computation steps, $C_e(\langle e, 0\rangle, \tau_{s''})$ converges and gives truth value 1 to $\in_{\langle e,q\rangle}$, Then
 - for all $s' > s$, let $F_e^{s'} = F_e^s$, and
 - let $i_e = 1$, if $q \mod 4 = 3$; otherwise $i_e = 3$.
(S2) If there is no such odd number q, then let p be the largest $s' \leq s$ such that $F_e^{s'} \neq F_e^{s'-1}$ (where we take $F_e^{-1} = \emptyset$).
If there exists a s'' such that

– $\langle e, 4p+4\rangle \leq s'' \leq s$,
– $C_e(\langle e, 0\rangle, \tau_{s''})$ halts within s steps, and
– $C_e(\langle e, 0\rangle, \tau_{s''})$ gives truth value 0 to $\in_{\langle e,4p+4\rangle}$ and truth value 1 to $\in_x$ for all $x \in F_e^s$

then set $F_e^{s+1} = F_e^s \cup \{\langle e, 4p+4\rangle\}$; otherwise, set $F_e^{s+1} = F_e^s$.

If i_e is not already defined during the process of defining F_e^s (due to some step (S1) as above), then let $i_e = 1$.

It is easy to verify that the F_e^s as defined above satisfy the properties (PA) to (PD). For property (PE) note that if $F_e^{s+1} \neq F_e^s$, then it is due to step (S2), where we add $\langle e, 4p+4\rangle$ to F_e^{s+1}; but then, for any future additions to F_e, we will use only $p \geq s+1$ in step (S2); thus property (PE) holds.

We now show that C_e does not strong consistently classify $\mathcal{L}$. Suppose otherwise. Consider the first case below which applies.

Case 1: In the construction of $(F_e^s)_{s\in\mathbb{N}}$ above, the condition in (S1) succeeds at some point.
In this case C_e is not consistent, as for some s'' and odd number q, $C_e(\langle e, 0\rangle, \tau_{s''})$ gave truth value 1 to $\in_{\langle e,q\rangle}$ even though $\langle e, q\rangle$ is not in any language in $\mathcal{L}$.

Case 2: Note case 1, and in the construction of $(F_e^s)_{s\in\mathbb{N}}$, condition in (S2) succeeds for infinitely many s.
Then F_e is infinite, and by step (S2), for all s' such that $F_e^{s'} \neq F_e^{s'+1}$, there exist $s''' \geq s'' \geq \langle e, 4(s'+1)\rangle$ such that $C_e(\langle e, 0\rangle, \tau_{s''})$ gives truth value 0 to $\in_{\langle e,4(s'+1)+4\rangle}$ and $C_e(\langle e, 0\rangle, \tau_{s'''})$ gives truth value 1 to $\in_{\langle e,4(s'+1)+4\rangle}$. Thus C_e is not strong consistent.

Case 3: Not Cases 1 and 2.
In this case, F_e is finite. Let s be such that $F_e = F_e^s$. Suppose X is the language such that, for large enough s'', $C_e(\langle e, 0\rangle, \tau_{s''})$ is the sequence of truth values of $\tau_{s''}$ for X. As C_e is supposed to be consistent, X must contain $\langle e, 0\rangle$.

Case 3a: X contains $\langle e, q\rangle$, for some odd $q \in \mathbb{N}$.
In this case (S1) would eventually succeed and Case 1 would apply.

Case 3b: X contains some $y \notin F_e$, where $y \neq \langle e, q\rangle$ for any odd q.
Then, for some large enough s'', $C_e(\langle e, 0\rangle, \tau_{s''})$ gives truth value 1 for $\langle e, 0\rangle$ and y, even though $\{\langle e, 0\rangle, y\}$ is not contained in any language in $\mathcal{L}$.

Case 3c: $X = F_e$.
Then, for large enough $s' > s$, in step (S2) $F_e^{s'}$ would be made proper superset of F_e^s, a contradiction to $F_e = F_e^s$.

Case 3d: $X \subset F_e$.
Let $s' > \langle e, 4s+3\rangle$ be large enough such that $C_e(\langle e, 0\rangle, \tau_{s'})$ is the sequence of truth values of $\tau_{s'}$ for X. But then C_e is not consistent as there is no language L' in $\mathcal{L}$ such that $C_e(\langle e, 0\rangle, \tau_{s'})$ is the sequence of truth values of $\tau_{s'}$ for L'.

From the above cases it follows that C_e is not strong consistent on $\mathcal{L}$. □

Theorem 19. *Let $k > 1$ be given. There exists a class $\mathcal{L}$ of languages such that some strongly consistent within class and conservative* In*-classifier classifies $\mathcal{L}$ making at most k mind changes, but no* In*-classifier classifies $\mathcal{L}$ making at most $k-1$ mind changes.*

Proof. Let $\mathcal{L}$ consist of all sets of the form $\{0, 1, 2, \ldots, i\}$ with $i \leq k$. It is easily verified that some strongly consistent within class and conservative In-classifier classifies $\mathcal{L}$ making at most k mind changes.

Let C be an In-classifier that classifies $\mathcal{L}$. Set $\tau = (\in_0, \in_1, \in_2, \ldots, \in_k)$. Then a text for $\{0, 1, \ldots, k\}$ with initial segments $\sigma_0, \ldots, \sigma_k$ can be constructed in such a way that for all $i \leq k$, $\mathrm{cnt}(\sigma_i) = \{0, 1, 2, \ldots, i\}$ and $C(\sigma_i, \tau)$ is the sequence of truth values of τ for $\{0, 1, \ldots, i\}$. So C makes at least k mind changes on the classification task τ. $\square$

If conservative classifiers were required to output an initial hypothesis different to ? (in line with what is required of consistent classifiers), then the number of mind changes for conservative classifiers could be one more than the number of mind changes needed by strongly consistent classifiers. Indeed, consider the class of languages defined as in Theorem 19 augmented with $\{k+1\}$ and $\{0, k+1\}$. If C was a conservative In-classifier in that alternative sense, then $C\big((), (0, 1, \ldots, k)\big)$ could not be $(1, 0, \ldots, 0)$, as C would otherwise fail to classify $\{k+1\}$. Hence $k+1$ mind changes would be needed for C to classify that class of languages, whereas k mind changes are sufficient for a strongly consistent In-classifier.

Similarly, if conservative learners could not output ?, then the number of mind changes by the conservative learner in the next theorem would increase by one. This result should be compared with Theorem 14, as it also shows that the behaviour of a classifier can be improved "for free". Here k rather than at most one mind change is allowed, and conservativeness is the behaviour of the classifier which can be guaranteed.

Theorem 20. *Let $k \in \mathbb{N}$ and a class of languages $\mathcal{L}$ be given. If some* In*-classifier classifies $\mathcal{L}$ making at most k mind changes, then some conservative* In*-classifier classifies $\mathcal{L}$ making at most k mind changes.*

Proof. Let C be an In-classifier that classifies $\mathcal{L}$ making at most k mind changes. Without loss of generality assume that $C(\sigma, \tau)$ is defined for all σ and τ. Let an In-classifier C' be defined as follows. Let $\tau \in \mathrm{In}^\star$ be given. Set $C'\big((), \tau\big) =?$. For all $\sigma \in \mathrm{SEQ}$ and $x \in \mathbb{N} \cup \{\#\}$, if $C(\sigma \diamond x, \tau) \neq ?$ and $C(\sigma \diamond x, \tau)$ is the sequence of truth values of τ for $\mathrm{cnt}(\sigma \diamond x)$, then let $C'(\sigma \diamond x, \tau) = C(\sigma \diamond x, \tau)$; otherwise, $C'(\sigma \diamond x, \tau) = C'(\sigma, \tau)$. Note that C' essentially only copies those $C(\sigma, \tau)$ which are exactly for σ. Thus, it can immediately be seen that C' is conservative and classifies $\mathcal{L}$ making at most k mind changes. $\square$

The last result of the paper deals with the hierarchy of the number of mind changes that might be needed as a function of the number of membership predicates to decide. It shows that for any strictly increasing total function f, for some class of languages $\mathcal{L}$, it is impossible for any consistent classifier for $\mathcal{L}$ to decide n predicates using at most $f(n)$ mind changes, though for some particular choices of f, some classifier might do so with $f(n+1)$ mind changes at most.

Theorem 21. *For every strictly increasing total function f from $\mathbb{N}$ into $\mathbb{N}$, there exists a class $\mathcal{L}$ of languages which satisfies the following:*

1. *For all consistent* In*-classifiers C for $\mathcal{L}$ and $n > 0$, there exists $\tau \in \mathrm{In}^n$, $L \in \mathcal{L}$ and a text t for L such that $\{i \in \mathbb{N} \mid C(t_{|i+1}, \tau) \neq C(t_{|i}, \tau)\}$ has cardinality greater than $f(n)$.*
2. *There exists a consistent* In*-classifier C for $\mathcal{L}$ such that for all $n \in \mathbb{N}$, for all $\tau \in \mathrm{In}^n$, for all $L \in \mathcal{L}$ and for all texts t for L, the set $\{i \in \mathbb{N} \mid C(t_{|i+1}, \tau) \neq C(t_{|i}, \tau)\}$ has cardinality at most $3(f(n)+2)$.*

Proof. Let $(C_e)_{e\in\mathbb{N}}$ denote an effective enumeration of all partial computable In-classifiers. Let $e \in \mathbb{N}$ and $n > 0$ be given. Let $\tau_{e,n}$ denote the sequence $(\in_{\langle e,n,0\rangle}, \dots, \in_{\langle e,n,n-1\rangle})$. If there exists $p \in \mathbb{N}$ such that $C_e(\#^p, \tau_{e,n})$ is defined and is a sequence of nothing but 0's, then let $g(e,n)$ denote such a p; otherwise, let $g(e,n)$ denote 0. Given $e \in \mathbb{N}$, $n > 0$ and $r \in \{n, \dots, n+f(n)+1\}$, let $L^r_{e,n}$ denote $\{\langle e,n,x\rangle \mid n \le x \le r\}$, $\sigma^r_{e,n}$ the concatenation of $\#^{g(e,n)}$ with $(\langle e,n,n\rangle, \dots, \langle e,n,r\rangle)$, and for all $i < n$, $L^{r,i}_{e,n}$ the set $L^r_{e,n} \cup \{\langle e,n,i\rangle\}$.

Let $\mathcal{L}$ consist of $\varnothing$ and, for $e \in \mathbb{N}$, $n > 0$ and $r \in \{n, \dots, n+f(n)+1\}$, sets of the form $L^r_{e,n}$ or $L^{r,i}_{e,n}$ such that the following holds.

- Suppose that $n \le r \le n+f(n)$ and $r-n$ is even. Then for all $i < n$, $\mathcal{L}$ contains $L^{r,i}_{e,n}$ iff $\mathcal{L}$ contains $L^s_{e,n}$ and $L^{s,i}_{e,n}$ for all $s \in \{n, \dots, r-1\}$ and $i < n$. Moreover, $\mathcal{L}$ contains $L^r_{e,n}$ iff (i) $\mathcal{L}$ contains $L^{r,i}_{e,n}$ for all $i < n$ and (ii) $C_e(\sigma^r_{e,n}, \tau_{e,n})$ is defined and is a sequence where 1 occurs once and only once.
- Suppose that $n \le r \le n+f(n)$ and $r-n$ is odd. Then $\mathcal{L}$ contains $L^r_{e,n}$ iff $\mathcal{L}$ contains $L^s_{e,n}$ and $L^{s,i}_{e,n}$ for all $s \in \{n, \dots, r-1\}$ and $i < n$. Moreover, for all $i < n$, $\mathcal{L}$ contains $L^{r,i}_{e,n}$ iff (i) $\mathcal{L}$ contains $L^r_{e,n}$ and (ii) $C_e(\sigma^r_{e,n}, \tau_{e,n})$ is defined and is a sequence of nothing but 0's.
- Suppose that $r = n+f(n)+1$. Then $\mathcal{L}$ contains $L^r_{e,r}$ and $L^{n,i}_{e,r}$ for all $i < n$ iff $\mathcal{L}$ contains $L^s_{e,n}$ and $L^{s,i}_{e,n}$ for all $s \in \{n, \dots, r-1\}$ and $i < n$.

Let $e \in \mathbb{N}$ be such that C_e is a consistent In-classifier that classifies $\mathcal{L}$. Let $n > 0$ be given. Since C_e classifies $\varnothing$, the definition of $g(e,n)$ implies that $C_e(\#^{g(e,n)}, \tau_{e,n})$ is a sequence of nothing but 0's. Moreover, for all members r of $\{n, \dots, n+f(n)\}$, either $r-n$ is even and $C(\sigma^r_{e,n}, \tau_{e,n})$ is defined and is a sequence where 1 occurs once and only once, or $r-n$ is odd and $C(\sigma^r_{e,n}, \tau_{e,n})$ is defined and is a sequence of nothing but 0's. Indeed, suppose otherwise for a contradiction. Let $r \in \{n, \dots, n+f(n)\}$ be least such that the previous condition does not hold. Assume that $r-n$ is even. (The case where $r-n$ is odd is similar.) According to its definition, $\mathcal{L}$ contains $L^{r,i}_{e,n}$ for all $i < n$, but it contains no $L^s_{e,n}$ with $s \in \{r, \dots, n+f(n)+1\}$. Since C_e classifies $\mathcal{L}$ and is consistent, we infer that $C_e(\sigma^r_{e,n}, \tau_{e,n})$ is defined and is a sequence where 1 occurs once and only once. This is a contradiction. Hence any text t for $L^{n+f(n)+1}_{e,n}$ having

$\sigma^r_{e,n}$ as initial segment for all $r \in \{n, \ldots, n + f(n) + 1\}$ is such that the cardinality of the set $\{i \in \mathbb{N} \mid C_e\big(t_{|i+1}, \tau_{e,n}\big) \neq C_e\big(t_{|i}, \tau_{e,n}\big)\}$ is greater than $f(n)$. This completes the proof of part 1.

For part 2, define an In-classifier C as follows. For all $\tau \in \mathrm{In}^\star$, and σ such that $\mathrm{cnt}(\sigma) \neq \emptyset$, let $C\big(\sigma, \tau\big)$ consists of nothing but 0's. Now, let $n \in \mathbb{N}$, $\tau \in \mathrm{In}^n$ and σ be for $\mathcal{L}$. Let natural numbers e and m be such that (all of) the members of $\mathrm{cnt}(\sigma)$ are of the form $\langle e, m, x\rangle$. Let p be the largest integer such that $\langle e, m, p\rangle$ belongs to $\mathrm{cnt}(\sigma)$.

- Suppose that $p < m$. Then $C(\sigma, \tau)$ is the sequence of truth values of τ for $L^{m,p}_{e,m}$.
- Suppose that $p \geq m$ and, for some $p' < m$, $\in_{\langle e,m,p'\rangle}$ is not in τ. If there exists $i < m$ such that $\mathrm{cnt}(\sigma)$ contains $\langle e, m, i\rangle$ then $C(\sigma, \tau)$ is the sequence of truth values of τ for $L^{p,i}_{e,m}$. Otherwise, $C(\sigma, \tau)$ is the sequence of truth values of τ in $L^p_{e,m}$.
- Suppose that $p \geq m$ and $\in_{\langle e,m,p'\rangle}$ is in τ for all $p' < m$. If there exists $i < m$ such that $\mathrm{cnt}(\sigma)$ contains $\langle e, m, i\rangle$ then $C(\sigma, \tau)$ is the sequence of truth values of τ for $L^{p,i}_{e,m}$. Suppose otherwise.
  - If $p - m$ is odd then $C(\sigma, \tau)$ is the sequence of truth values of τ for $L^p_{e,m}$.
  - If $p - m$ is even then $C(\sigma, \tau)$ is the sequence of truth values of τ for $L^{p,0}_{e,m}$, unless $C_e(\sigma, \tau)$ is found out within $\mathrm{len}(\sigma)$ steps to be defined and be a sequence where 1 occurs once and only once, in which case $C(\sigma, \tau)$ is the sequence of truth values of τ for $L^p_{e,m}$.

It is easy to verify that C classifies $\mathcal{L}$ and is consistent. Furthermore, for all texts t for $\mathcal{L}$, $n \in \mathbb{N}$ and $\tau \in \mathrm{In}^n$, $\{i \in \mathbb{N} \mid C\big(t_{|i+1}, \tau\big) \neq C\big(t_{|i}, \tau\big)\}$ has cardinality at most equal to $3 \times (f(n) + 2)$. $\square$

6 Conclusion

Our aim was to close the gap between classification and learning, using a notion of multiclassification where arbitrarily large finite sets of membership queries have to be dealt with synchronously. Learning implicitly permits to multiclassify the membership of all numbers in the limit and therefore our results that in many cases learnability is more difficult to achieve than classification is not unexpected. More precisely, we considered BC-learnability and showed that with at most two mind changes, classifiers enjoying good properties in terms of consistency and conservativeness could succeed where BC-learners would fail. Then we considered Ex-learnability and showed that it could be easier than classification in terms of mind changes, and we established deeper relationships between classification and learning by exhibiting how an Ex-learner could be constructed from a classifier constrained in the maximum number of mind changes it is allowed to make.

We have also shown that multiclassification is interesting in its own right. In particular, we combined it with conservativeness and various variants of consistency which resulted in a complex and interesting picture. More precisely, we considered conservativeness together with one form of consistency, and possibly with restrictions on the maximal number of mind changes, and we showed that it allows classifiers to be sometimes more powerful than classifiers not required to be conservative, but subjected to more stringent conditions on consistency or on mind changes.

Acknowledgements

We thank the anonymous referees for several helpful comments which improved the presentation of the paper.

References

[Ang80] Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[Bār74a] Janis Bārzdiņš. Inductive inference of automata, functions and programs. In *Proceedings of the 20th International Congress of Mathematicians, Vancouver*, pages 455–560, 1974. In Russian. English translation in American Mathematical Society Translations: Series 2, 109:107-112, 1977.

[Bār74b] Janis Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 82–88. Latvian State University, 1974. In Russian.

[BB75] Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28(2):125–155, 1975.

[Cas74] John Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8(1):15–32, 1974.

[CKSS04] John Case, Efim Kinber, Arun Sharma and Frank Stephan. On the classification of recursive languages. *Information and Computation*, 192:15–40, 2004.

[CL82] John Case and Chris Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 107–115. Springer-Verlag, 1982.

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

[FH94] Rusins Freivalds and Achim Hoffmann. An inductive inference approach to classification. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:63–72, 1994.

[GM98] Bill Gasarch and Georgia Martin. Bounded Queries in Recursion Theory. Birkhäuser 1998.

[GSPM98] William Gasarch, Frank Stephan, Mark G. Pleszkoch and Mahendran Velauthapillai. Classification using information. *Annals of Mathematics and Artificial Intelligence*, 23(1-2), 147–168, 1998.

[Gol67] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[Jai01] Sanjay Jain. On an open problem in classification of languages. *Journal of Experimental and Theoretical Artificial Intelligence*, 13(2):113–118, 2001.

[JB81] Klaus Peter Jantke and Hans-Rainer Beick. Combining postulates of naturalness in inductive inference. *Journal of Information Processing and Cybernetics (EIK)*, 17:465–484, 1981.

[JMS08] Sanjay Jain, Eric Martin and Frank Stephan. Absolute versus probabilistic classification in a logical setting. *Theoretical Computer Science A*, 397(1–3):114–128, 2008. Special Issue on Forty Years of Inductive Inference. Dedicated to the 60th Birthday of Rolf Wiehagen.

[JORS99] Sanjay Jain, Daniel Osherson, James Royer and Arun Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, second edition, 1999.

[OW82] Daniel Osherson and Scott Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.

[Rog67] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted by MIT Press in 1987.

[Ste01] Frank Stephan. On one-sided versus two-sided classification. *Archive for Mathematical Logic*, 40:489–513, 2001.

[SWZ97] Carl H. Smith, Rolf Wiehagen and Thomas Zeugmann. Classifying predicates and languages. *International Journal of Foundations of Computer Science*, 8:15–41, 1997.

[WL76] Rolf Wiehagen and Walter Liepe. Charakteristische Eigenschaften von erkennbaren Klassen rekursiver Funktionen. *Journal of Information Processing and Cybernetics (EIK)*, 12:421–438, 1976.

[WS95] Rolf Wiehagen and Carl H. Smith. Generalization versus classification. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:163–174, 1995.

[ZZ08] Thomas Zeugmann and Sandra Zilles. Learning recursive functions: A survey. *Theoretical Computer Science A*, 397(1–3):4–56, 2008. Special Issue on Forty Years of Inductive Inference. Dedicated to the 60th Birthday of Rolf Wiehagen.