Splitting of Learnable Classes

Hongyang Li[1] and Frank Stephan[2]

[1] Department of Mathematics and Department of Computer Science, National University of Singapore, Singapore 119076, Republic of Singapore.
hongyang@comp.nus.edu.sg

[2] Department of Mathematics and Department of Computer Science, National University of Singapore, Singapore 119076, Republic of Singapore.
fstephan@comp.nus.edu.sg

Abstract. A class $\mathcal{L}$ is called mitotic if it admits a splitting $\mathcal{L}_0, \mathcal{L}_1$ such that $\mathcal{L}, \mathcal{L}_0, \mathcal{L}_1$ are all equivalent with respect to a certain reducibility. Such a splitting might be called a symmetric splitting. In this paper we investigate the possibility of constructing a class which has a splitting and where any splitting of the class is a symmetric splitting. We call such a class a symmetric class. In particular we construct an incomplete symmetric BC-learnable class with respect to strong reducibility. We also introduce the notion of very strong reducibility and construct a complete symmetric BC-learnable class with respect to very strong reducibility. However, for EX-learnability, it is shown that there does not exist a symmetric class with respect to any weak, strong or very strong reducibility.

Keywords: inductive inference, mitotic classes, intrinsic complexity.

1 Introduction

Gold [7] initiated the study of inductive inference; he considered, besides various other models, in particular the learning of classes of recursively enumerable sets from positive data. The basic idea of this scenario is that the learner is presented with a list of all elements of some member set in the class in arbitrary order and has to find, in the limit, a program which enumerates the given language. The initial study was soon extended [2–4, 13, 14] and notions to compare the difficulties of classes were introduced, in particular notions which translate the sequence of data describing the language from the first class into a corresponding sequence of data for a language in the second class plus a reverse translation from any sequence of hypotheses for the image language back to a sequence of hypotheses for the first class; if such a reduction exists and the second class is learnable, so is the first. These reducibilities were introduced in order to measure the *intrinsic complexity* of learning [5, 8–10] and the field is quite well-studied within inductive inference. Based on this notion, Jain and Stephan [11] investigated whether mitoticity occurs in inductive inference. The notion of mitoticity stems from the study of recursively enumerable sets [15] and means that an r.e. set A is the union of two disjoint r.e. sets B, C such that A, B, C have all the same Turing degree [12]. This concept was also brought over to complexity theory [1, 6].

When studying it in inductive inference, Jain and Stephan [11] failed to solve the following related question (where the notions of splitting, intrinsic reducibility $\leqslant_r$ and complete class will be made more precise below).

Question 1. *Given an intrinsic reducibility $\leqslant_r$, is there a learnable class $\mathcal{L}$ such that $\mathcal{L}$ admits a splitting and every splitting of $\mathcal{L}$ is symmetric, that is, for every splitting of $\mathcal{L}$ into two halves $\mathcal{L}_0$ and $\mathcal{L}_1$ it holds that $\mathcal{L}_0 \leqslant_r \mathcal{L}_1$ and $\mathcal{L}_1 \leqslant_r \mathcal{L}_0$?*

Jain and Stephan [11] did not solve this question. However, they showed one result on the way to it: If a class $\mathcal{L}$ is BC-complete with respect to strong reducibility and if $\mathcal{L}_0, \mathcal{L}_1$ form a splitting of $\mathcal{L}$ then either $\mathcal{L}_0 \equiv_{strong} \mathcal{L}$ or $\mathcal{L}_1 \equiv_{strong} \mathcal{L}$. Note that splittings always exist in the case of a BC-complete class [11].

In the following, the underlying definitions and notions are explained formally.

- A general recursive operator Θ is a mapping from total functions to total functions such that there is a recursively enumerable set E of triples such that, for every total function f and every x, y, $\Theta(f)(x) = y$ iff there is an n such that $\langle f(0)f(1)\ldots f(n), x, y\rangle \in E$.
- A language L is a recursively enumerable subset of the natural numbers.
- A class $\mathcal{L}$ is a set of languages.
- A text T of a language L is an infinite sequence $T(0), T(1), T(2), \ldots$ such that every member of L is equal to some $T(m)$ and every $T(m)$ is either a member of L or a special symbol to denote a pause. The content of a text T, denoted by *content*(T), is the set containing all symbols that have appeared in T.
- A learner M is a general recursive operator that reads more and more elements of the text T and outputs a sequence $e_0, e_1, \ldots$ of conjectures. M explanatorily (EX) learns some language L iff there is an n such that $e_n = e_{n+1} = \ldots$ and $L = W_{e_n}$ where $W_0, W_1, \ldots$ is an underlying acceptable numbering of all r.e. sets which is used as a fixed hypothesis space for learning. M behaviourally correctly (BC) learns L iff $L = W_{e_n}$ for almost all n. Now M learns a class $\mathcal{L}$ iff M learns

every language $L \in \mathcal{L}$ from any text of L under the learning criterion considered (EX or BC, respectively).

- A classifier C is a general recursive operator that reads more and more elements of the text T and outputs a binary sequence $a_0, a_1, \ldots$ where each a_n is either 0 or 1. We say a classifier C classifies a class $\mathcal{L}$ if C converges to an element in $\{0,1\}$ in the limit on any text T of any language L in the class $\mathcal{L}$, and C converges to the same number on any text of the same language. For convenience we write $C(L)$ to denote the number that C converges on any text of L. It should be noted that a classifier of a class $\mathcal{L}$ is not required to converge on texts of languages outside the class $\mathcal{L}$.

Note that in the framework of inductive inference it does not matter how fast a learner M or a classifier C converges. The machine can be slowed down by starting with an arbitrary guess and later repeating conjectures. Similarly, if one translates one text of a language L into a text of a language H, it is not important how fast the symbols of H show up in the translated text; it is only important that they show up eventually. Therefore the translator can put into the translated text pause symbol, $\#$, until more data are available or certain simulated computations have terminated.

Therefore, learners, operators translating texts and classifiers can be made primitive recursive by the just mentioned delaying techniques. Thus one can have recursive enumerations $\Theta_0, \Theta_1, \Theta_2, \ldots$ of translators from texts to texts, $M_0, M_1, M_2, \ldots$ of learners and $C_0, C_1, C_2, \ldots$ of classifiers such that, for every given translator, learner or classifier, the corresponding list contains an equivalent one.

Definition 2. We say that a class $\mathcal{L}$ is weakly reducible to $\mathcal{H}$ (written $\mathcal{L} \leqslant_{weak} \mathcal{H}$) iff there are two general recursive operators Γ, Θ such that for every language $L \in \mathcal{L}$ and every text T of L there is a language $H \in \mathcal{H}$ satisfying that Γ translates T into a text of H and that whenever a sequence E converges to an index e_H such that $W_{e_H} = H$ then the sequence $\Theta(E)$ converges to an index e_L such that $W_{e_L} = L$.

We say that a general recursive operator Γ strongly maps texts of languages in $\mathcal{L}$ to texts of languages in $\mathcal{H}$ iff, whenever T, T' are texts of the same language in $\mathcal{L}$, $\Gamma(T), \Gamma(T')$ are texts of the same language in $\mathcal{H}$. Furthermore, $\mathcal{L}$ is strongly reducible to $\mathcal{H}$ (written $\mathcal{L} \leqslant_{strong} \mathcal{H}$) iff $\mathcal{L} \leqslant_{weak} \mathcal{H}$ via general recursive operators Γ, Θ such that Γ strongly maps texts of languages in $\mathcal{L}$ to texts of languages in $\mathcal{H}$.

A class $\mathcal{L}$ is very strongly reducible to $\mathcal{H}$ (written $\mathcal{L} \leqslant_{vs} \mathcal{H}$) iff there is a general recursive operator Γ and a recursive function f such that the following conditions hold:

- If T is a text of a language in $\mathcal{L}$ then $\Gamma(T)$ is a text of a language in $\mathcal{H}$.
- If T, T' are texts of the same language in $\mathcal{L}$ then $\Gamma(T), \Gamma(T')$ are texts of the same language in $\mathcal{H}$.
- If T is a text of a language $L \in \mathcal{L}$ and e is an index such that $W_e = \mathit{content}(\Gamma(T))$ then $W_{f(e)} = L$.

Definition 3. Given a learning criterion I and a reducibility $\leqslant_r$, a class $\mathcal{H}$ is I-complete with respect to $\leqslant_r$ if for any I-learnable class $\mathcal{L}$ we have $\mathcal{L} \leqslant_r \mathcal{H}$.

Ladner [12] introduced the recursion-theoretic version of mitoticity. He defined that an infinite recursively enumerable set A splits into infinite sets A_0, A_1 if there is a partial recursive function with domain A mapping the elements of A_a to a for all $a \in \{0, 1\}$. In learning theory, the corresponding role of the partial recursive function is played by a classifier C which classifies the class $\mathcal{L}$ into $\mathcal{L}_0, \mathcal{L}_1$.

Definition 4. A splitting of a class $\mathcal{L}$ is a pair of infinite sub-classes $\mathcal{L}_0, \mathcal{L}_1$ such that $\mathcal{L}_0 \cap \mathcal{L}_1 = \emptyset$, $\mathcal{L}_0 \cup \mathcal{L}_1 = \mathcal{L}$ and there exists a classifier C such that, for all $a \in \{0, 1\}$ and for all texts T with $\mathit{content}(T) \in \mathcal{L}_a$, C converges on T to a.

Definition 5. A class $\mathcal{L}$ is called a symmetric class with respect to a certain reducibility $\leqslant_r$ if $\mathcal{L}$ has a splitting, and for any splitting $\mathcal{L}_0, \mathcal{L}_1$ of $\mathcal{L}$, we have $\mathcal{L}_0 \equiv_r \mathcal{L}_1 \equiv_r \mathcal{L}$.

This paper contains two main results extending the theorems established in [11]. While Jain and Stephan has shown the existence of a complete BC-learnable class with only comparable splittings, in this paper both a complete BC-learnable symmetric class and an incomplete BC-learnable symmetric class have been constructed. Jain and Stephan has also shown in [11] that an EX-learnable class complete for $\leqslant_{weak}$ ($\leqslant_{strong}$) always has a symmetric splitting with respect to $\leqslant_{weak}$ ($\leqslant_{strong}$). However, in this paper it is shown that there exists no EX-learnable symmetric class for $\leqslant_{weak}$. Both results needed novel approaches which were not yet there in [11]. The other results in the present work are less involved and obtainable with known techniques.

2 Very Strong Reducibility

In this section we present several results related to the very strong reducibility. First we show in Theorem 6 that $\leqslant_{vs}$ is strictly stronger than $\leqslant_{strong}$. Then we show in Theorem 7 that various learnabilities are preserved under $\leqslant_{vs}$ so that it is reasonable to consider splittings of a learnable class with respect to $\leqslant_{vs}$.

Theorem 6. *There exist explanatorily learnable classes $\mathcal{L}$ and $\mathcal{H}$ such that $\mathcal{L} \leqslant_{strong} \mathcal{H}$ but $\mathcal{L} \nleqslant_{vs} \mathcal{H}$.*

Proof. Take the following classes which are clearly EX-learnable:

$$\mathcal{L} = \{\{n\} : n \in \mathbb{N}\} \tag{1}$$

and

$$\mathcal{H} = \{\{0, 1, \ldots, n\} : n \in \mathbb{N}\}. \tag{2}$$

Let Γ be the recursive operator which replaces in every text each occurrence of n by an occurrence of the sequence $0, 1, \ldots, n$ and therefore translates every text of $\{n\}$ into a text of a language of the form $\{0, 1, \ldots, n\}$. Then Γ strongly maps texts of languages in $\mathcal{L}$ to texts of languages in $\mathcal{H}$.

Let $E = (e_0, e_1, e_2, \ldots)$ be a sequence of indices. Define a recursive operator Θ such that whenever E converges syntactically to an index e and W_e has maximum n then $\Theta(E)$ is a sequence of indices which converges to some index m with $W_m = \{n\}$. The sequence output by the operator might either fail to converge or converge to an arbitrary index in the case that E does not converge or E converges to an index for a set which is not of the form $\{0, 1, \ldots, n\}$ for any n. Then Γ, Θ witness the strong reducibility from $\mathcal{L}$ to $\mathcal{H}$. Therefore the statement $\mathcal{L} \leqslant_{strong} \mathcal{H}$ holds.

Now we assume the contrary that $\mathcal{L} \leqslant_{vs} \mathcal{H}$. There is a recursive operator mapping every text of a set of the form $\{n\}$ to a set of the form $\{0, 1, \ldots, \gamma(n)\}$ for some K-recursive function γ and a recursive function f such that for all $n \in \mathbb{N}$ and for all $e \in \mathbb{N}$, $W_{f(e)} = \{n\}$ as long as $W_e = \{0, 1, \ldots, \gamma(n)\}$. Now choose m, n such that $\gamma(m) < \gamma(n)$ and let

$$W_{g(d)} = \begin{cases} \{0, 1, \ldots, \gamma(m)\} & \text{if } d \notin K; \\ \{0, 1, \ldots, \gamma(n)\} & \text{if } d \in K. \end{cases} \tag{3}$$

It follows that

$$m \in W_{f(g(d))} \Leftrightarrow d \notin K, n \in W_{f(g(d))} \Leftrightarrow d \in K \tag{4}$$

and therefore one could find out whether $d \in K$ by enumerating $W_{f(g(d))}$ until either m or n shows up in the enumeration. This would give that the halting problem is recursive, a contradiction. Therefore $\mathcal{L} \nleqslant_{vs} \mathcal{H}$. $\square$

Theorem 7. *Suppose $\mathcal{L}$ and $\mathcal{H}$ are classes of languages and suppose $\mathcal{L} \leqslant_{vs} \mathcal{H}$. Then the following statements hold:*

- *If $\mathcal{H}$ is EX-learnable, then $\mathcal{L}$ is also EX-learnable;*
- *If $\mathcal{H}$ is BC-learnable, then $\mathcal{L}$ is also BC-learnable;*

Proof. We prove for the case of EX-learnability. The proof for other cases are similar.

Let M be an EX-learner of $\mathcal{H}$. Assume the very strong reducibility is witnessed by (Γ, f). Define the new learner N for $\mathcal{L}$ as follows:

$$N(\sigma) = f(M(\Gamma(\sigma))). \tag{5}$$

Let T be a text of a language in $\mathcal{L}$. Since M EX-learns $\mathcal{H}$, M converges on $\Gamma(T)$ to an index e of $content(\Gamma(T))$. Then N converges on T to $f(e)$. By definition of very strong reducibility, $W_{f(e)} = content(T)$. Therefore, N converges on T to a right index and N EX-learns $\mathcal{L}$. $\square$

The next result investigates the question of whether there are complete classes with respect to very strong reducibility and it gives an affirmative answer for BC-learnable classes. Similarly one can also show the existence of complete EX-learnable classes with respect to very strong reducibility.

Theorem 8. *There exists a BC-learnable class which is complete for very strong reducibility.*

Proof. Fix an enumeration $M_0, M_1, M_2, \ldots$ of all learners. Consider the class

$$\mathcal{L} = \{\{x\} \oplus W_e \ : \ M_x \text{ BC-learns } W_e\}. \tag{6}$$

Now $\mathcal{L}$ is BC-learnable since there is a learner which waits until the element $2x$ has been seen in the input text (as $\{x\} \oplus W_e = \{2x\} \cup \{2y+1 : y \in W_e\}$); once $2x$ is known, the learner simulates M_x to learn W_e and translates every index d conjectured by M_x to an index for $\{2x\} \cup \{2y+1 : y \in W_d\}$.

For the converse direction, given an arbitrary BC-learnable class $\mathcal{H}$, let M_x be a learner for $\mathcal{H}$. We define a mapping Γ from languages in $\mathcal{H}$ to languages in $\mathcal{L}$ such that

$$\Gamma(W_e) = \{x\} \oplus W_e. \tag{7}$$

Note that there also exists a recursive function f such that

$$W_{f(e')} = \{y \;:\; 2y+1 \in W'_e\}. \tag{8}$$

Then the pair Γ, f shows that $\mathcal{H} \leqslant_{vs} \mathcal{L}$. Therefore the class $\mathcal{L}$ is complete for $\leqslant_{vs}$. □

3 Symmetric BC-Learnable Classes

This section contains the main results of the paper. We construct a BC-complete symmetric class with respect to $\leqslant_{vs}$ (which is automatically complete for $\leqslant_{strong}$) in Theorem 9 and a BC-incomplete symmetric class with respect to $\leqslant_{vs}$ in Theorem 10. In the end we construct a BC-complete class which is not symmetric in Theorem 11.

Theorem 9. *There is a BC-learnable class $\mathcal{J}$ which is complete for $\leqslant_{vs}$ such that, for any splitting $\mathcal{J}_0, \mathcal{J}_1$ of $\mathcal{J}$, we have $\mathcal{J}_0 \equiv_{vs} \mathcal{J}_1 \equiv_{vs} \mathcal{J}$.*

Proof. Let $\mathcal{L}$ be a BC-learnable class which is complete for $\leqslant_{vs}$. Fix a numbering of r.e. sets such that $W_0 = \emptyset$. Define

$$f(n) = \max\{0, \varphi_e(m)\downarrow : e \le n, m \le n\}. \tag{9}$$

Define for any recursive function g the language

$$J_g = \{\langle x, y, z\rangle \;:\; (y < f(x)) \vee (y = f(x) \wedge z \in W_{g(x)})\} \tag{10}$$

and define the class

$$\mathcal{J} = \{J_g \;:\; (\forall^\infty x[g(x) = 0]) \wedge (\forall x[g(x) \neq 0 \Rightarrow W_{g(x)} \in \mathcal{L}])\}. \tag{11}$$

First we show that $\mathcal{J}$ is BC-learnable. Given any input string σ, let

$$\sigma_{x,y}(n) = \begin{cases} z & \text{if } \sigma(n) = \langle x, y, z\rangle; \\ \# & \text{otherwise.} \end{cases} \tag{12}$$

Let M be a learner for $\mathcal{L}$ such that $\forall n[M(\#^n) = 0]$. We define the learner N such that

$$W_{N(\sigma)} = \{\langle x, y, z\rangle \;:\; \exists s[(y < f_s(x)) \vee (y = f_s(x) \wedge z \in W_{M(\sigma_{x,y})})]\} \tag{13}$$

where $f_s(x)$ is the maximum of 0 and all values $\varphi_e(m)$ where $e \le x$, $m \le x$ and the corresponding computation terminates within s computation steps. Note that the sequence $f_0, f_1, \ldots$ recursively approximates f from below.

Let J_g be any language in $\mathcal{J}$. To see that N learns J_g, first note that given any x, we always have $f_s(x) = f(x)$ for sufficiently large s. Then we have $\forall x \forall y[y < f(x) \Rightarrow \exists s[y < f_s(x)]]$. Therefore, all tuples $\langle x, y, z\rangle \in J_g$ with $y < f(x)$ will eventually go into the set $W_{N(\sigma)}$ for any σ. Also note that $\forall s \forall x[f_s(x) \leq f(x)]$. Therefore, the condition $y < f_s(x)$ will not put any tuple $\langle x, y, z\rangle$ with $y \geq f(x)$ into $W_{N(\sigma)}$.

Since there are only finitely many x with $g(x) \neq 0$, there exists an s such that $f_s(x) = f(x)$ for all x with $g(x) \neq 0$. By definition, if $g(x) \neq 0$, then $W_{g(x)} \in \mathcal{L}$. Since M BC-learns $\mathcal{L}$, we have $W_{M(\sigma_{x,f(x)})} = W_{g(x)}$ in the limit, namely, given any text T of J_g, for sufficiently long initial segment σ of T, we have for all x that $W_{M(\sigma_{x,f(x)})} = W_{g(x)}$. It follows that $W_{N(\sigma)}$ enumerates a tuple of the form $\langle x, f(x), z\rangle$ if and only if $g(x) \neq 0$ and $z \in W_{g(x)}$. This justifies that N learns J_g. Since J_g is chosen arbitrarily, we claim that N BC-learns $\mathcal{J}$.

Next we show that $\mathcal{L} \leqslant_{vs} \mathcal{J}_0$ for any splitting $\mathcal{J}_0, \mathcal{J}_1$ of $\mathcal{J}$. Then analogously $\mathcal{L} \leqslant_{vs} \mathcal{J}_1$ and the theorem follows automatically from the transitivity of $\leqslant_{vs}$ and the completeness of $\mathcal{L}$.

Fix some $J_h \in \mathcal{J}_0$. Let C be an arbitrary classifier. Without loss of generality assume $C(J_h) = 0$. Let ϵ be the locking sequence of C on J_h. Define

$$\varphi_e(n) = \max\{y : \exists x, z[\tau_n \text{ is defined and } \langle x, y, z\rangle \in \tau_n]\} \tag{14}$$

where τ_n is the first string found such that

1. $C(\epsilon\tau_n) \neq C(\epsilon)$;
2. $\langle u, v, z\rangle \in \tau_n \Rightarrow u > n$ or $\langle u, v, z\rangle \in J_h$.

If such a τ_n cannot be found, then $\varphi_e(n)$ is undefined.

Note that τ_n must contain some $\langle x_{\tau_n}, y_{\tau_n}, z_{\tau_n}\rangle \notin J_h$; otherwise the condition $C(\epsilon\tau_n) \neq C(\epsilon)$ would contradict the fact that ϵ is a locking sequence. Then we have $y_{\tau_n} \geq f(x_{\tau_n})$. Since $\forall x \leq n[\langle x, y, z\rangle \in \tau_n \Rightarrow \langle x, y, z\rangle \in J_h]$, we have $x_{\tau_n} > n$. Therefore we have $\varphi_e(n) \geq y_{\tau_n} \geq f(x_{\tau_n}) \geq f(n)$.

We claim that there exists some m such that $\varphi_e(n)$ is undefined for all $n > m$. Consider the function

$$\varphi_{e'}(n) = \begin{cases} 1 + \varphi_e(n) & \text{if } \varphi_e(n)\downarrow, \\ \uparrow & \text{if } \varphi_e(n)\uparrow. \end{cases} \tag{15}$$

Let $m = \max\{e', e\}$. Assume that $n > m$ and $\varphi_e(n)$ is defined. Then $\varphi_{e'}(n)$ is also defined. By definition of f we have $f(n) \geq \varphi_{e'}(n) > \varphi_e(n)$, contradicting the fact that $\varphi_e(n) \geq f(n)$.

Since h is 0 almost everywhere and since $W_0 = \emptyset$, there exists some $d > m$ such that

$$\forall x \geq d\ \forall y, z[\langle x, y, z\rangle \in J_h \iff y < f(x)]. \tag{16}$$

Since $d > m$, $\varphi_e(d)$ is undefined. Therefore any string τ must violate at least one of the conditions in the definition of τ_n where $n = d$. Let $B = \{\langle x, y, z\rangle : x \leq d - 1\}$. Now consider any superset J_g of J_h such that $J_g \cap B = J_h \cap B$, and let τ be any string from the language J_g. Since J_g preserves J_h up to $x = d - 1$, τ always satisfies the second condition in the definition of τ_n for $n = d - 1$. However τ_n is not defined for $n = d - 1$ since $d - 1 \geq m$. Therefore one must conclude that τ violates the first condition in the definition of τ_n, which implies that $C(\epsilon) = C(\epsilon\tau)$. Since τ is chosen arbitrarily, we have the following conclusion:

> Let $B = \{\langle x, y, z\rangle : x \leq d - 1\}$. For any language $J_g \in \mathcal{J}$, if $J_h \subseteq J_g$ and $J_g \cap B = J_h \cap B$, then $C(J_g) = C(J_h) = 0$.

Now consider a mapping Γ from texts of languages in $\mathcal{L}$ to texts of languages in $\mathcal{J}_0$ such that

$$\Gamma(W_e) = J_h \cup \{\langle d, f(d), z\rangle \ :\ z \in W_e\}. \tag{17}$$

Since d is a fixed finite number, we can assume that Γ knows the value of d and $f(d)$. Then we can construct such a Γ by putting into the resulting text all tuples $\langle x, y, z\rangle \in J_h$ and $\langle d, f(d), w\rangle$ for all w in the input text. It is clear that $C(\Gamma(W_e)) = C(J_h) = 0$ since $\Gamma(W_e) \cap B = J_h \cap B$ and $\Gamma(W_e)$ is a superset of J_h.

For the other direction of the reduction, define a recursive function $\tilde{f}$ such that for any index e'

$$W_{\tilde{f}(e')} = \{z \ :\ \langle d, f(d), z\rangle \in W_{e'}\}. \tag{18}$$

Note that $J_h \cap \{\langle d, f(d), z\rangle \ :\ z \in W_e\} = \emptyset$ by the choice of d. Then the pair $\Gamma, \tilde{f}$ shows that $\mathcal{L} \leqslant_{vs} \mathcal{J}_0$. Due to the transitivity of $\leqslant_{vs}$, we have $\mathcal{J} \leqslant_{vs} \mathcal{L} \leqslant \mathcal{J}_0$, which implies that $\mathcal{J} \equiv_{vs} \mathcal{J}_0$. Since J_h and the classifier C are arbitrary, the theorem is proven. □

By a similar proof, one can show there is also a BC-incomplete symmetric class.

Theorem 10. *There is a BC-learnable class $\mathcal{J}$ which is incomplete for $\leqslant_{strong}$ (and thus incomplete for $\leqslant_{vs}$) such that, for any splitting $\mathcal{J}_0, \mathcal{J}_1$ of $\mathcal{J}$, we have $\mathcal{J}_0 \equiv_{vs} \mathcal{J}_1 \equiv_{vs} \mathcal{J}$.*

Proof. Fix any acceptable numbering of r.e. sets such that $W_0 = \emptyset$. Define

$$f(n) = \max\{0, \varphi_e(m)\downarrow : e \leq n, m \leq n\}. \tag{19}$$

Define for any recursive function g the language

$$J_g = \{\langle x, y, z\rangle \;:\; (y < f(x)) \vee (y = f(x) \wedge z \in D_{g(x)})\} \tag{20}$$

where D_e is the finite set with canonical index e. That is, $D_e = E$ iff E is finite and $\sum_{d\in E} 2^d = e$. Note that $D_0 = \emptyset$. Let

$$\mathcal{J} = \{J_g : \forall^\infty x[g(x) = 0]\}. \tag{21}$$

Similar to the proof of theorem 9 one could show that $\mathcal{J}$ is BC-learnable and symmetric.

To show that the class $\mathcal{J}$ is not BC-complete, consider the class

$$\mathcal{H} = \{H_n : x \in H_n \iff x \geq n\}. \tag{22}$$

Clearly $\mathcal{H}$ is BC-learnable. Note that $\mathcal{H}$ contains an infinite descending chain starting with H_0. However, for any $J_g \in \mathcal{J}$, there does not exist an infinite descending chain in $\mathcal{J}$ starting with J_g due to the fact that any $J_{\tilde{g}} \in \mathcal{J}$ is a finite-variant superset of $\{\langle x, y, z\rangle : y < f(x)\}$. Since a strong reduction preserves the proper subset relationship among the members of the class [10], it also preserves infinite descending chains with respect to the subset relation and so $\mathcal{H} \nleqslant_{strong} \mathcal{J}$.

Note that $\mathcal{H}$ from the previous proof is EX-learnable and therefore $\mathcal{L}$ is not hard for all EX-learnable classes with respect to strong and very strong reducibility.

Theorem 11. *There is BC-complete class $\mathcal{H}$ which has a splitting $\mathcal{H}_0, \mathcal{H}_1$ such that $\mathcal{H} \nleqslant_{strong} \mathcal{H}_1$ and $\mathcal{H}_0 \nleqslant_{strong} \mathcal{H}_1$.*

Proof. Let $\mathcal{L}$ be an arbitrary BC-complete class. Define

$$\mathcal{H}_0 = \{\{0, 1\} \cup \{x + 2 : x \in L\} \;:\; L \in \mathcal{L}\}. \tag{23}$$

Then $\mathcal{H}_0$ is also BC-complete. Now let $\mathcal{H}_1 = \{\{x\} : x \in \mathbb{N}\}$. Note that $\mathcal{H}_1$ is EX-learnable.

Then $\mathcal{H} = \mathcal{H}_0 \cup \mathcal{H}_1$ is also BC-complete. It is easy to see that there is a classifier that splits $\mathcal{H}$ into $\mathcal{H}_0$ and $\mathcal{H}_1$. However, we cannot have $\mathcal{H} \leqslant_{strong} \mathcal{H}_1$ and $\mathcal{H}_0 \leqslant_{strong} \mathcal{H}_1$, which would imply that every BC-learnable class is EX-learnable. □

4 Asymmetric EX-Learnable Classes

We have shown that there is a BC-learnable symmetric class with respect to $\leqslant_{vs}$. It is natural to ask whether it is possible to construct a symmetric class for EX-learnability. As we will show in this section, there is no EX-learnable symmetric classes.

Definition 12. We say that a general recursive operator Γ mapping texts of languages in $\mathcal{L}$ to texts of languages in $\mathcal{H}$ is 1-1 if, whenever T_1 and T_2 are texts of different languages in $\mathcal{L}$, $\Gamma(T_1)$ and $\Gamma(T_2)$ are texts of different languages in $\mathcal{H}$.

Note that the general recursive operator Γ in the definition of weak, strong and very-strong reducibility must be 1-1; otherwise it is impossible for the operator Θ to translate a sequence of indices converging to an index of $content(\Gamma(T))$ to a sequence of indices converging to an index of $content(T)$.

Theorem 13. *There is no EX-learnable class $\mathcal{L}$ such that $\mathcal{L}$ has a splitting and for every splitting $\mathcal{L}_0, \mathcal{L}_1$ of $\mathcal{L}$, $\mathcal{L}_0 \equiv_{weak} \mathcal{L}_1$.*

Proof. Let M be an EX-learner for $\mathcal{L}$. Without loss of generality we could assume that M converges to the same index on all texts of the same language in $\mathcal{L}$ [4]. Let C be an arbitrary classifier that classifies $\mathcal{L}$ and let $\mathcal{L}_0, \mathcal{L}_1$ be the splitting of $\mathcal{L}$ produced C. Fix some language $L \in \mathcal{L}_0$. Then $C(L) = 0$. Suppose M converges on all texts of L to an index e_L. Let σ_L be the minimum locking sequence of M on L. Define another classifier C' such that

$$C'(\tau) = \begin{cases} C(\tau) & \text{if } content(\sigma_L) \nsubseteq content(\tau) \\ & \text{or } M(\sigma_L \tau) \neq M(\sigma_L); \\ 1 & \text{otherwise.} \end{cases} \tag{24}$$

We claim that C' produces a splitting $\mathcal{L}_0 - \{L\}, \mathcal{L}_1 \cup \{L\}$. To see this, note that for any $L' \neq L$, if $\sigma_L \nsubseteq L'$, then the condition $content(\sigma_L) \nsubseteq content(\tau)$ will always hold, and C' will preserve the classification made by C on L'. If σ_L is contained in L', then σ_L cannot be a locking sequence for L' since $L' \neq L$. Then there exists some $\tau \subseteq L'$ such that $M(\sigma_L \tau) \neq M(\sigma_L)$, and C' will also preserve the classification made by C on L'. It is clear that $C'(L) = 1$ while $C(L) = 0$. Therefore, C' moves exactly L from $\mathcal{L}_0$ to $\mathcal{L}_1$ and preserves all the rest classifications made by C.

Now assume the contrary that every splitting of $\mathcal{L}$ produces two subclasses of the same complexity, then we must have $\mathcal{L}_0 \equiv_{weak} \mathcal{L}_1$ and $\mathcal{L}_0 - \{L\} \equiv_{weak} \mathcal{L}_1 \cup \{L\}$. Then there exists a 1-1 general recursive operator Γ' which maps texts of languages in $\mathcal{L}_0$ to texts of languages in $\mathcal{L}_1$ and a 1-1 general recursive operator Γ'' which maps texts of languages in $\mathcal{L}_1 \cup \{L\}$ to texts of languages in $\mathcal{L}_0 - \{L\}$. Therefore $\Gamma = \Gamma'' \circ \Gamma'$ is a 1-1 general recursive operator which maps texts of languages in $\mathcal{L}_0$ to texts of languages in $\mathcal{L}_0 - \{L\}$.

Let T_0 be a text of L and $T_n = \Gamma(T_{n-1})$ for all $n > 0$. Now define $H_n = \mathit{content}(T_n)$ for all $n \in \mathbb{N}$. Note that $i \neq j \Rightarrow H_i \neq H_j$ since Γ is a 1-1 general recursive operator. Fix an enumeration of all primitive-recursive operators $\{\Gamma_0, \Gamma_1, \Gamma_2, \ldots\}$ translating texts. Define

$$O_{d,e} = \mathit{content}(\Gamma_d(T_e)). \tag{25}$$

Then there exists a K-recursive function $f(b, d, e)$ such that

$$\forall c \leq b \; [c = f(b, d, e) \text{ or } H_c \neq O_{d,e}]. \tag{26}$$

To see the existence of such a function f, first note that $H_i \neq H_j$ whenever $i \neq j$. Therefore, given any b, d, e, there can be at most one c such that $c \leq b$ and $H_c = O_{d,e}$. Moreover, for any i, j with $i \neq j$, there must be some x such that $H_i(x) \neq H_j(x)$. Then it follows that for any i, j with $i < j \leq c$, there must be some x such that either $H_i(x) \neq O_{d,e}(x)$ or $H_j(x) \neq O_{d,e}(x)$. Therefore between any two different languages H_i and H_j, we can always identify one of them that is not identical to $O_{d,e}$ using oracle K. Moreover, we can repeat this process for b times to identify, among all the $b+1$ languages H_i with $i \leq b$, b languages which are not identical to $O_{d,e}$. Then we set $f(b, d, e) = c$ where H_c is the only language with index less than or equal to b that has not been identified as different from $O_{d,e}$.

Now we define a K-recursive sequence $\{a_n\}$, where $a_0 = 0$. To define a_{n+1}, let $b = (n+1)(a_n + 2n + 5) + (n+2) + (a_n + 2n + 5) + 1$ and let a_{n+1} be the least c found which satisfies all the following conditions:

– $\forall d \leq n \forall e \leq a_n + 2n + 4 \; [c \neq f(b, d, e)]$;
– $M(T_c) > n + 1$;
– $c > a_n + 2n + 4$.

Note that there can be at most $(n+1)(a_n+2n+5)$ different choices of c that violate the first condition, at most $n+2$ choices of c with $M(T_c) \leq n+1$ that violate the second condition, and at most $a_n + 2n + 5$ choices of c

with $c \le a_n + 2n + 4$ that violate the third condition. Therefore, given our definition of b, it is guaranteed that there exists a $c \le b$ which satisfies all the three conditions.

The set $E = \{M(T_{a_n}) : n \in \mathbb{N}\} = \{e : \exists n \le e[M(T_{a_n}) = e]\}$ is K-recursive. Let E_s be the s-th recursive approximation to E. Now define

$$C''(\sigma) = E_{|\sigma|}(M(\sigma)) \tag{27}$$

and note that this is a classifier splitting $\mathcal{L}$ into two halves $\mathcal{H}_0$ and $\mathcal{H}_1$ where $\mathcal{H}_1 = \{H_{a_n} : n \in \mathbb{N}\}$ and $\mathcal{H}_0$ contains all other members of $\mathcal{L}$. To see this, consider any text T. On T, M converges to a value e and C'' converges to the value $E(e)$. If T is a text of a language in $\mathcal{H}_1$ then M converges on T to some value of the form $M(T_{a_n})$ and hence M converges to a value in E; if T is a text of a language in $\mathcal{H}_0$ then M converges on T to an index e for a language outside $\mathcal{H}_1$, hence $e \notin E$ and C'' converges on T to 0.

We show that $\mathcal{H}_0 \not\equiv_{weak} \mathcal{H}_1$ by showing it is impossible for any 1-1 general recursive operator to map texts of languages in $\mathcal{H}_0$ to texts of languages in $\mathcal{H}_1$.

Assume the contrary that Γ_d is a 1-1 general recursive operator which maps texts of languages in $\mathcal{H}_0$ to texts of languages in $\mathcal{H}_1$. Choose some $n > d$. Note that since $a_{n+1} > a_n + 2n + 4$, there are at most $n + 1$ numbers $e \le a_n + 2n + 4$ such that $H_e \in \mathcal{H}_1$. It follows that there exist more than $n+2$ numbers $e \le a_n + 2n + 4$ such that $H_e \in \mathcal{H}_0$. Then there must be some $m > n$ and some $e \le a_n + 2n + 4$ such that $O_{d,e} = H_{a_m}$, which contradicts our definition of the array $\{a_n\}$.

Since Γ_d is chosen arbitrarily, we conclude that it is not possible to find a 1-1 general recursive operator which maps texts of languages in $\mathcal{H}_0$ to texts of languages in $\mathcal{H}_1$. Therefore $\mathcal{H}_0 \not\equiv_{weak} \mathcal{H}_1$. □

Although there is no EX-learnable symmetric class, one could construct an EX-learnable class which has and only has comparable splittings as shown in the next theorem:

Theorem 14. *There is an EX-learnable class $\mathcal{L}$ such that $\mathcal{L}$ has a splitting, and for every splitting $\mathcal{L}_0, \mathcal{L}_1$ of $\mathcal{L}$, either $\mathcal{L}_0 \leqslant_{vs} \mathcal{L}_1$ or $\mathcal{L}_1 \leqslant_{vs} \mathcal{L}_0$.*

Proof. Let A be a K-cohesive set, that is, A satisfies that whenever a K-r.e. set contains infinitely many elements of A then this set contains all but finitely many elements of A. Now define the class $\mathcal{L}$ such that $\mathcal{L} = \{\{2x\}, \{2x+1\} : x \in A\}$. It is clear that $\mathcal{L}$ is EX-learnable since every language in $\mathcal{L}$ is a singleton set.

Let C be any classifier that classifies $\mathcal{L}$ and let $\mathcal{L}_0, \mathcal{L}_1$ be the splitting of $\mathcal{L}$ produced by C. Without loss of generality assume C puts $\{2x\}$ into $\mathcal{L}_0$ for infinitely many $x \in A$. Then, by definition of K-cohesive sets, C must put $\{2x\}$ into $\mathcal{L}_0$ for all but finitely many $x \in A$ and must put $\{2x+1\}$ into $\mathcal{L}_1$ for all but finitely many $x \in A$. Hence there exists a number m such that

$$\forall x \geq m\ [x \in A \Rightarrow (C(\{2x\}) = 0 \text{ and } C(\{2x+1\}) = 1)]. \tag{28}$$

Let $\mathcal{L}'_0 = \mathcal{L}_0 \cap \{\{y\} : y < 2m\}$ and $\mathcal{L}'_1 = \mathcal{L}_1 \cap \{\{y\} : y < 2m\}$. Again we may assume that $|\mathcal{L}'_0| \leq |\mathcal{L}'_1|$. Note that both $\mathcal{L}'_0$ and $\mathcal{L}'_1$ are finite. It is clear that there exists a 1-1 mapping from languages in $\mathcal{L}'_0$ to languages in $\mathcal{L}'_1$, and that there exists a 1-1 general recursive operator Γ' which maps texts of languages in $\mathcal{L}'_0$ to texts of languages in $\mathcal{L}'_1$.

Now define another recursive operator Γ such that

$$\Gamma(\{y\}) = \begin{cases} \Gamma'(\{y\}) & \text{if } y < 2m; \\ \{y+1\} & \text{otherwise.} \end{cases} \tag{29}$$

It is then easy to verify that Γ 1-1 strongly maps texts of languages in $\mathcal{L}_0$ to texts of languages in $\mathcal{L}_1$. The reverse mapping is done by a recursive function f which maps an index e to an index $f(e)$ such that $W_{f(e)} = \{y\}$ for the first pair x, y found with $x \in W_e \wedge \Gamma(\{y\}) = \{x\}$; if such x, y do not exist then $W_{f(e)} = \emptyset$. Note that only the search-algorithm for the x, y is coded into the index $f(e)$ but not the values x, y themselves, therefore f can be chosen to be a total-recursive function. It follows that $\mathcal{L}_0 \leqslant_{vs} \mathcal{L}_1$. □

5 Conclusion

In this paper we have investigated the existence of symmetric classes with respect to various reducibilities and learning criteria. In particular we have shown the existence of BC-learnable symmetric classes with respect to very strong reducibility $\leqslant_{vs}$. We have also shown that there exists no EX-learnable symmetric classes even for weak reducibility $\leqslant_{weak}$.

Note that a symmetric class requires each sub-class in any splitting to have the same complexity as the original class. While the existence of a BC-learnable symmetric class has been shown, it is not yet clear whether there is a BC-learnable class $\mathcal{L}$ which has a splitting such that for any splitting $\mathcal{L}_0, \mathcal{L}_1$ of $\mathcal{L}$, $\mathcal{L}_0 \equiv_r \mathcal{L}_1 <_r \mathcal{L}$. This could also be an interesting question to investigate.

References

1. Klaus Ambos-Spies. P-mitotic sets. *Logic and Machines*. Springer LNCS 177:1–23, 1984.
2. Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
3. Janis Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs*, Volume 1, Latvian State University, Riga, 210:82–88, 1974.
4. Lenore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. Information and Control **28**(2) (1975) 125 – 155
5. Rusins Freivalds, Efim Kinber and Carl Smith. On the intrinsic complexity of learning. *Information and Computation*, 123:64–71, 1995.
6. Christian Glaßer, Mitsunori Ogihara, Aduri Pavan, Alan Selman and Liyu Zhang. Autoreducibility, mitoticity and immunity. *Mathematical Foundations of Computer Science*: Thirtieth International Symposium, MFCS 2005, Gdansk, Poland, August 29 - September 2, 2005, Proceedings. Springer LNCS, 3618:387–398, 2005.
7. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
8. Sanjay Jain, Efim Kinber and Rolf Wiehagen. Language learning from texts: degrees of intrinsic complexity and their characterizations. *Journal of Computer and System Sciences*, 63:305–354, 2001.
9. Sanjay Jain and Arun Sharma. The intrinsic complexity of language identification. *Journal of Computer and System Sciences*, 52:393–402, 1996.
10. Sanjay Jain and Arun Sharma. The Structure of Intrinsic Complexity of Learning. *The Journal of Symbolic Logic*, 62:1187–1201, 1997.
11. Sanjay Jain and Frank Stephan. Mitotic classes in inductive inference. *SIAM Journal on Computing*, 38:1283–1299, 2008.
12. Richard Ladner. Mitotic recursively enumerable sets. *The Journal of Symbolic Logic*, 38:199–211, 1973.
13. Daniel N. Osherson, Michael Stob and Scott Weinstein. *Systems That Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists.* Bradford — The MIT Press, Cambridge, Massachusetts, 1986.
14. Daniel N. Osherson and Scott Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.
15. Emil Post. Recursively enumerable sets of positive integers and their decision problems, *Bulletin of the American Mathematical Society*, 50:284–316, 1944.