On the Role of Update Constraints and Text-Types in Iterative Learning

Sanjay Jain[1,⋆], Timo Kötzing[2], Junqi Ma[1], and Frank Stephan[1,3,⋆⋆]

[1] Department of Computer Science, National University of Singapore,
Singapore 117417, Republic of Singapore
sanjay@comp.nus.edu.sg, ma.junqi@nus.edu.sg

[2] Friedrich-Schiller University, Jena, Germany
timo.koetzing@uni-jena.de

[3] Department of Mathematics, National University of Singapore,
Singapore 119076, Republic of Singapore
fstephan@comp.nus.edu.sg

Abstract. The present work investigates the relationship of iterative learning with other learning criteria such as decisiveness, caution, reliability, non-U-shapedness, monotonicity, strong monotonicity and conservativeness. Building on the result of Case and Moelius that iterative learners can be made non-U-shaped, we show that they also can be made cautious and decisive. Furthermore, we obtain various special results with respect to one-one texts, fat texts and one-one hypothesis spaces.

1 Introduction

Iterative learning is the most common variant of learning in the limit which addresses memory constraints: the memory of the learner on past data is just its current hypothesis. Due to the padding lemma, this memory is still not void, but finitely many data can be memorised in the hypothesis. However, one subfield of the study of iterative learning considers therefore the usage of class-preserving one-one hypothesis spaces which limit this type of coding during the learning process. Other ways to limit it is to control the amount and types of updates; such constraints also aim for other natural properties of the conjectures: For example, updates have to be motivated by inconsistent data observed (syntactic conservativeness), semantic updates have to be motivated by inconsistent data observed (semantic conservativeness), updates cannot repeat semantically abandoned conjectures (decisiveness), updates cannot go from correct to incorrect hypotheses (non-U-shapedness), conjectures cannot be proper supersets of the language to be learnt (cautiousness) or conjectures have to contain all the data observed so far (consistency). There is already a quite comprehensive body of work on how iterativeness relates with various combinations of these constraints [CK10, GL04, JMZ13, JORS99, Köt09, LG02, LG03, LZ96, LZZ08], however various important questions remained unsolved. A few years ago, Case and Moelius

[⋆] Supported by NUS grants C252-000-087-001 and R146-000-181-112.

[⋆⋆] Supported in part by NUS grant R146-000-181-112.

P. Auer et al. (Eds.): ALT 2014, LNAI 8776, pp. 55–69, 2014.

[CM08b] obtained a breakthrough result by showing that iterative learners can be made non-U-shaped. The present work improves this result by showing that they can also be made decisive — this stands in contrast to the case of the usual non-iterative framework where decisiveness is a real restriction in learning [BCMSW08]. Further results complete the picture and also include the role of hypothesis spaces and text-types in iterative learning.

We completely characterise the relationship of the iterative learning criteria with the different restrictions as given in the diagramme in Figure 1. A line indicates a previously known inclusion. A gray box around criteria indicates equality of these criteria, as found in this work.

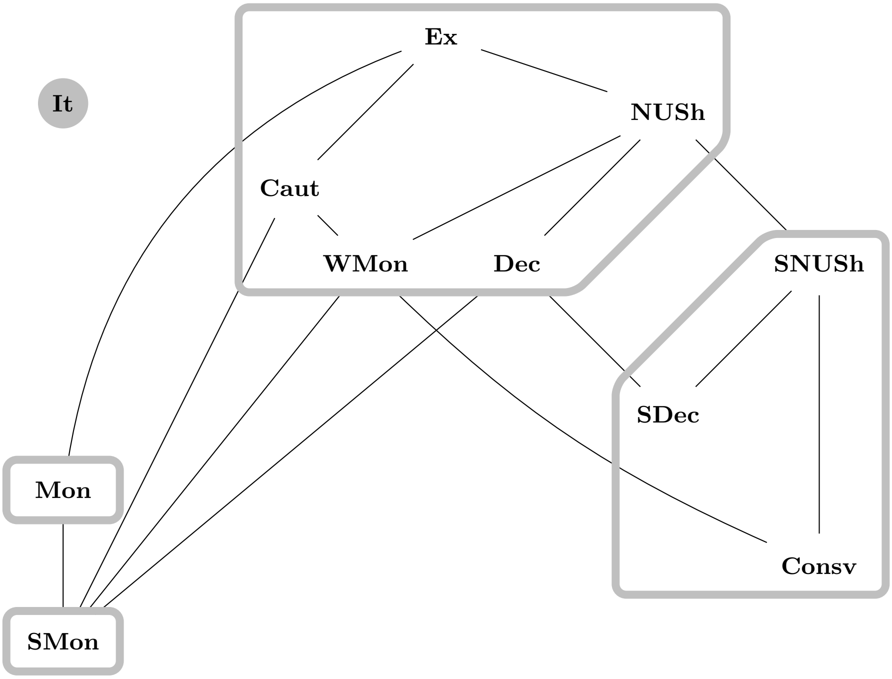


Fig. 1. Relation of criteria combined with iterative learning

The learning criteria investigated in the present work are quite natural. Conservativeness, consistency, cautiousness and decisiveness are natural constraints studied for a long time [Ang80, OSW86]; these criteria require that conjectures contain the data observed (consistency) or that mind changes are based on evidence that the prior hypothesis is incorrect (conservativeness); a lot of work has been undertaken using the assumption that learners are both, consistent and conservative. Monotonicity constraints play an important role in various fields like monotonic versus non-monotonic logic and this is reflected in inductive inference by considering the additional requirement that new hypotheses should be at least as general as the previous ones [Jan91, LZ93]. The fundamental notion of iterative learning is one of the first memory-constraints to be investigated in inductive inference and has been widely studied [LG02, LG03, LZ96, OSW86]; the beauty of this criterion is that the memory limitation comes rather indirectly, as for finitely many steps the memory can be enhanced by padding; after that, however, the learner has to converge and to ignore new data unless it gives enough evidence to undertake a mind change. Osherson, Stob and Weinstein [OSW82] formalised decisiveness as a notion where a learner never semantically returns to an abandoned hypothesis; they left it as an open problem whether the notion of decisiveness is restrictive; it took about two decades until the problem was solved [BCMSW08]. The search for this solution and also the parallels to developmental psychology motivated to study the related notion of non-U-shapedness where a non-U-shaped learner never abandons a correct hypothesis for an incorrect one and later (in a U-shaped way) returns to a correct hypothesis. The study of this field turned out to be quite fruitful and productive and we also consider decisive and non-U-shaped learning and its variants in this paper.

Taking this into account, we believe that the criteria investigated are natural and deserve to be studied; the restrictions on texts which we investigated are motivated from the fact that in the case of memory limitations (like enforced by iterativeness), the learners cannot keep track of which information has been presented before and therefore certain properties of the text (like every datum appearing exactly once or every datum appearing infinitely often) can be exploited by the learner during the learning process. In some cases these exploitations only matter when the restrictions on the hypothesis space make the iterativeness-constraint stricter, as they might rule out padding. Such a restriction is quite natural, as padding is a way to permit finite calculations to go into the update process and thereby bypass the basic idea behind the notion of iterativeness; this is reflected in the finding that the relations between the learning criteria differ for iterative learning in general and iterative learning using a class-preserving one-one hypothesis space.

Due to space restrictions some proofs are omitted. The full paper is available as Technical Report TRA7/14, School of Computing, National University of Singapore.

2 Mathematical Preliminaries

Unintroduced notation follows the textbook of Rogers [Rog67] on recursion theory. The set of natural numbers is denoted by $\mathbb{N} = \{0, 1, 2, \ldots\}$. The symbols $\subseteq$, $\subset$, $\supseteq$, $\supset$ respectively denote the subset, proper subset, superset and proper superset relation between sets. The symbol $\emptyset$ denotes both the empty set and the empty sequence.

With dom and range we denote, respectively, domain and range of a given function. We sometimes denote a partial function f of $n > 0$ arguments $x_1, \ldots, x_n$ in lambda notation (as in Lisp) as $\lambda x_1, \ldots, x_n . f(x_1, \ldots, x_n)$. For example, with $c \in \mathbb{N}$, $\lambda x . c$ is the constantly c function of one argument.

We let $\langle x, y\rangle = \frac{(x+y)(x+y+1)}{2} + x$ be Cantor's Pairing function which is an invertible, order-preserving function from $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Whenever we consider tuples of natural numbers as input to a function, it is understood that the general coding function $\langle\cdot,\cdot\rangle$ is used to code the tuples into a single natural number. We similarly fix a coding for finite sets and sequences, so that we can use those as input as well.

If a function f is not defined for some argument x, then we denote this fact by $f(x)\uparrow$ and we say that f on x *diverges*; the opposite is denoted by $f(x)\downarrow$ and we say that f on x *converges*. If f on x converges to p, then we denote this fact by $f(x)\downarrow = p$.

$\mathcal{P}$ and $\mathcal{R}$ denote, respectively, the set of all partial recursive and the set of all recursive functions (mapping $\mathbb{N} \to \mathbb{N}$). We let φ be any fixed acceptable numbering for $\mathcal{P}$ (an acceptable numbering could, for example, be based on a natural programming language such as C or Java). Further, we let φ_p denote the partial-recursive function computed by the φ-program with code number p. A set $L \subseteq \mathbb{N}$ is *recursively enumerable (r.e.)* iff it is the domain of a partial recursive function. We let $\mathcal{E}$ denote the set of all r.e. sets. We let W be the mapping such that $\forall e : W_e = \mathrm{dom}(\varphi_e)$. W is, then, a mapping from $\mathbb{N}$ *onto* $\mathcal{E}$. We say that e is an index, or program, (in W) for W_e. Let $W_{e,s}$ denote W_e enumerated in s steps in some uniform way to enumerate all the W_e's. We let pad be a 1–1 padding function such that for all e and finite sets D, $W_{\mathrm{pad}(e,D)} = W_e$.

The special symbol ? is used as a possible hypothesis (meaning "no change of hypothesis"). The symbol # stands for a pause, that is, for "no new input data in the text". For each (possibly infinite) sequence q with its range contained in $\mathbb{N} \cup \{\#\}$, let $\mathrm{content}(q) = (\mathrm{range}(q) \setminus \{\#\})$. By using an appropriate coding, we assume that ? and # can be handled by recursive functions.

For any function f and all i, we use $f[i]$ to denote the sequence $f(0), \ldots, f(i-1)$ (the empty sequence if $i = 0$ and undefined, if one of these values is undefined).

3 Learning Criteria

In this section we formally introduce our setting of learning in the limit and associated learning criteria. We follow [Köt09] in its "building-blocks" approach for defining learning criteria.

A *learner* is a partial function from $\mathbb{N}$ to $\mathbb{N} \cup \{?\}$. A *language* is a r.e. set $L \subseteq \mathbb{N}$. Any total function $T : \mathbb{N} \to \mathbb{N} \cup \{\#\}$ is called a *text*. For any given language L, a *text for L* is a text T such that $\mathrm{content}(T) = L$. Initial parts of this kind of text is what learners usually get as information. We let σ and τ range over initial segments of texts. Concatenation of two initial segments σ and τ is denoted by $\sigma \diamond \tau$. For a given set of texts F, we let $\mathbf{Txt}^F(L)$ denote the set of all texts in F for L.

An *interaction operator* is an operator β taking as arguments a function M (the learner) and a text T, and that outputs a function p. We call p the *learning sequence* (or *sequence of hypotheses*) of M given T. Intuitively, β defines how a learner can interact with a given text to produce a sequence of conjectures.

We define the sequence generating operators **G** and **It** (corresponding to the learning criteria discussed in the introduction) as follows. For all learners M, texts T and all i,

$$\mathbf{G}(M,T)(i) = M(T[i]);$$
$$\mathbf{It}(M,T)(i) = \begin{cases} M(\emptyset), & \text{if } i = 0; \\ M(\mathbf{It}(M,T)(i-1), T(i-1)), & \text{otherwise;} \end{cases}$$

where $M(\emptyset)$ denotes the *initial conjecture* made by M. Thus, in iterative learning, the learner has access to the previous conjecture, but not to all previous data as in **G**-learning. With any iterative learner M we associate a learner M^* such that

$$M^*(\emptyset) = M(\emptyset) \text{ and}$$
$$\forall \sigma, x : M^*(\sigma \diamond x) = M(M^*(\sigma), x).$$

Intuitively, M^* on a sequence σ returns the hypothesis which M makes after being fed the sequence σ in order. Note that, for all texts T, $\mathbf{G}(M^*, T) = \mathbf{It}(M, T)$. We let $M(T)$ (respectively $M^*(T)$) denote $\lim_{n\to\infty} M(T[n])$ (respectively, $\lim_{n\to\infty} M^*(T[n])$) if it exists.

Successful learning requires the learner to observe certain restrictions, for example convergence to a correct index. These restrictions are formalised in our next definition.

A *learning restriction* is a predicate δ on a learning sequence and a text. We give the important example of explanatory learning (**Ex**, [Gol67]) and that of vacillatory learning (**Fex**, [CL82, OW82, Cas99]) defined such that, for all sequences of hypotheses p and all texts T,

$$\mathbf{Ex}(p,T) \Leftrightarrow [\exists n_0 \forall n \geq n_0 : p(n) = p(n_0) \wedge W_{p(n_0)} = \text{content}(T)];$$
$$\mathbf{Fex}(p,T) \Leftrightarrow [\exists n_0 \exists \text{ finite } D \subset \mathbb{N}$$
$$\forall n \geq n_0 : p(n) \in D \wedge \forall e \in D : W_e = \text{content}(T)].$$

Furthemore, we formally define the restrictions discussed in Section 1 in Figure 2. We combine any two sequence acceptance criteria δ and δ' by intersecting them; we denote this by juxtaposition (for example, all the restrictions given in Figure 2 are meant to be always used together with **Ex**).

For any set of texts F, interaction operator β and any (combination of) learning restrictions δ, $\mathbf{Txt}^F\beta\delta$ is a *learning criterion.* A learner M $\mathbf{Txt}^F\beta\delta$*-learns* all languages in the class

$$\mathbf{Txt}^F\beta\delta(M) = \{L \in \mathcal{E} \mid \forall T \in \mathbf{Txt}(L) \cap F : \delta(\beta(M,T),T)\}$$

and we use $\mathbf{Txt}\beta\delta$ to denote the set of all $\mathbf{Txt}\beta\delta$-learnable classes (learnable by some learner). Note that we omit the superscript F whenever F is the set of all texts.

In some cases, we consider learning using an explicitly given particular hypothesis space $(H_e)_{e\in\mathbb{N}}$ instead of the usual acceptable numbering $(W_e)_{e\in\mathbb{N}}$. For this, one replaces W_e by H_e in the respective definitions of learning as above.

$$\begin{aligned}
\mathbf{Consv}(p,T) &\Leftrightarrow [\forall i : \text{content}(T[i+1]) \subseteq W_{p(i)} \Rightarrow p(i) = p(i+1)];\\
\mathbf{Caut}(p,T) &\Leftrightarrow [\forall i,j : W_{p(i)} \subset W_{p(j)} \Rightarrow i < j];\\
\mathbf{NUSh}(p,T) &\Leftrightarrow [\forall i,j,k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} = \text{content}(T) \Rightarrow W_{p(j)} = W_{p(i)}];\\
\mathbf{Dec}(p,T) &\Leftrightarrow [\forall i,j,k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} \Rightarrow W_{p(j)} = W_{p(i)}];\\
\mathbf{SNUSh}(p,T) &\Leftrightarrow [\forall i,j,k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} = \text{content}(T) \Rightarrow p(j) = p(i)];\\
\mathbf{SDec}(p,T) &\Leftrightarrow [\forall i,j,k : i \leq j \leq k \ \wedge\ W_{p(i)} = W_{p(k)} \Rightarrow p(j) = p(i)];\\
\mathbf{SMon}(p,T) &\Leftrightarrow [\forall i,j : i < j \Rightarrow W_{p(i)} \subseteq W_{p(j)}];\\
\mathbf{Mon}(p,T) &\Leftrightarrow [\forall i,j : i < j \Rightarrow W_{p(i)} \cap \text{content}(T) \subseteq W_{p(j)} \cap \text{content}(T)];\\
\mathbf{WMon}(p,T) &\Leftrightarrow [\forall i,j : i < j \wedge \text{content}(T[j]) \subseteq W_{p(i)} \Rightarrow W_{p(i)} \subseteq W_{p(j)}].
\end{aligned}$$

Fig. 2. Definitions of learning restrictions

4 Plain-Text Learning

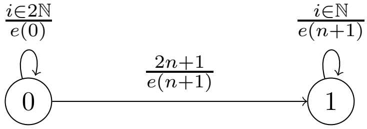
In this section we first show that, for iterative learning, the convergence restrictions **Ex** and **Fex** allow for learning the same sets of languages. After that we give the necessary theorems establishing the diagramme given in Figure 1.

Theorem 1. **TxtItFex** = **TxtItEx**.

Next we give separating theorems for monotone learning and first show that there is a class which can be learnt iteratively by a learner which is strongly decisive, conservative, monotone and cautious while on the other hand, there is no learner which, even non-iteratively, learns the same class strongly monotonically.

Theorem 2. **TxtItSDecConsvMonCautEx** $\not\subseteq$ **TxtGSMonEx**.

Proof. Let $L_0 = \{0, 2, 4, \ldots\}$ and for all n, $L_{n+1} = \{2m \mid m \leq n\} \cup \{2n+1\}$. Let $\mathcal{L} = \{L_n : n \in \mathbb{N}\}$. Let e be a recursive function computing an r.e. index for L_n: $W_{e(n)} = L_n$. Let $M \in \mathcal{P}$ be the iterative learner which memorises a single state in its conjecture (using padding) and has the following state transition diagramme (an edge labeled $\frac{x}{e}$ means that the edge indicates a state transition on input x with conjecture output e).


$\frac{i \in 2\mathbb{N}}{e(0)}$
$\frac{i \in \mathbb{N}}{e(n+1)}$
$\frac{2n+1}{e(n+1)}$
0
1

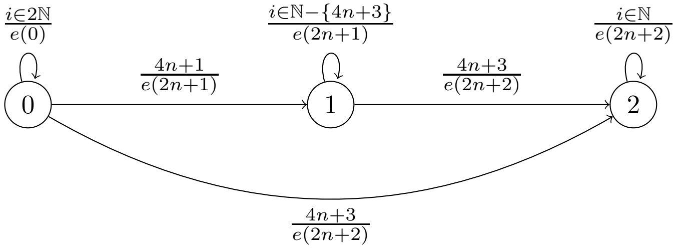
Clearly, M is a **TxtItSDecConsvMonCautEx**-learner for $\mathcal{L}$. It is known that $\mathcal{L}$ is not strongly monotonically learnable. □

Note that one can modify this protocol such that M only memorises the state; however, M then abstains from repeating correct conjectures and one has to modify the learnability criterion such that outputting a special symbol for repeating the last (correct) conjecture is allowed. The next result shows that there is a class of languages which can be learnt by an iterative learner which is strongly decisive, conservative and cautious; on the other hand, there is no learner, even non-iterative one, that learns the class monotonically.

Theorem 3. TxtItSDecConsvCautEx $\not\subseteq$ **TxtGMonEx**.

Proof. We consider $L_0 = \{0, 2, 4, \ldots\}$ and, for all n, $L_{2n+1} = \{2m \mid m \leq n\} \cup \{4n+1\}$ and $L_{2n+2} = \{2m \mid m \leq n+1\} \cup \{4n+1, 4n+3\}$. We let $\mathcal{L} = \{L_n \mid n \in \mathbb{N}\}$.

Let e be a recursive function such that, for all n, $W_{e(n)} = L_n$. Let $M \in \mathcal{P}$ be the iterative learner using state transitions as given by the following diagramme.


$\frac{i \in 2\mathbb{N}}{e(0)}$
$\frac{i \in \mathbb{N} - \{4n+3\}}{e(2n+1)}$
$\frac{i \in \mathbb{N}}{e(2n+2)}$
$\frac{4n+1}{e(2n+1)}$
$\frac{4n+3}{e(2n+2)}$
0
1
2
$\frac{4n+3}{e(2n+2)}$

Clearly, M fulfills all the desired requirements for **TxtItSDecConsvCautEx**-learning $\mathcal{L}$. One can show that every learner of $\mathcal{L}$ outputs on some text for some L_{2n+2} hypotheses for L_0, L_{2n+1} and L_{2n+2} (in that order, with possibly other hypotheses in between) and is therefore not learning monotonically. □

The next result shows that there is a class of languages which is simultaneously iteratively, monotonically, decisively, weakly monotonically and cautiously learnable, but not iteratively strongly non-U-shapedly learnable.

Theorem 4. TxtItMonDecWMonCautEx $\not\subseteq$ **TxtItSNUShEx**.

The next result shows that there is an iteratively and strongly monotonically learnable class which does not have any iterative learner which is strongly non-U-shaped, that is, which never revises a correct hypothesis. The proof uses the notion of a join which is defined as $A \oplus B = \{2x : x \in A\} \cup \{2x+1 : x \in B\}$.

Theorem 5. TxtItSMonEx $\not\subseteq$ **TxtItSNUShEx**.

Proof. Let $M_0, M_1, \ldots$ denote a recursive listing of all partial recursive iterative learning machines. Consider a class $\mathcal{L}$ consisting of the following sets for each $e \in \mathbb{N}$ (where $F(\cdot)$, $G(\cdot)$ are recursively enumerable sets in the parameters described later):

- $\{2e\} \oplus F(e)$
- $\{2e, 2d+1\} \oplus G(e, d)$

– $\{2e, 2d + 1\} \oplus \mathbb{N}$

where,

(a) If there exists an s such that $M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2s + 1) = M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2s + 1 \diamond \# \diamond 2s' + 1)$, for all $s' > s$, then $F(e) = \{0, 1, 2, \ldots, s\}$, else $F(e) = \mathbb{N}$.

(b) If $F(e) = \mathbb{N}$ or $\max(F(e)) > d$, then $G(e, d) = \mathbb{N}$. Otherwise, if there exists a $k > d$ such that $M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2\max(F(e)) + 1 \diamond \# \diamond 4d + 2 \diamond \#^r) = M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \diamond \ldots \diamond 2max(F(e)) + 1 \diamond \# \diamond 4d + 2 \diamond \#^r \diamond \#) \neq M_e^*(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond 5 \diamond \# \ldots \diamond 2max(F(e)) + 1 \diamond \# \diamond 4d + 2 \diamond \#^r \diamond \# \diamond 2k + 1)$ then $G(e, d) = F(e) \cup \{k\}$ for first such k found in some algorithmic search, else $G(e, d) = F(e)$.

Now, the above class is **TxtItSMonEx** learnable, as the learner can remember seeing $4e, 4d + 2$ in the input text, if any:

- Having seen only $4e$, the learner outputs a grammar for $\{2e\} \oplus F(e)$;
- Having seen $4e, 4d+2$, the learner outputs a grammar for $\{2e, 2d+1\} \oplus G(e, d)$ until it sees, (after having seen $4e, 4d + 2$), two more odd elements bigger than $2d$ in the input, at which point the learner switches to outputting a grammar for $\{2e, 2d + 1\} \oplus \mathbb{N}$.

It is easy to verify that the above learner will **TxtItSMon** learn $\mathcal{L}$.

Now we show that $\mathcal{L}$ is not **TxtItSNUShEx**-learnable. Suppose by way of contradiction that M_e **TxtItSNUShEx**-learns $\mathcal{L}$. Then the following statements hold:

– There exists an s as described in the definition of $F(e)$ above and thus $F(e)$ is finite, as otherwise M_e does not learn $2e \oplus F(e) = 2e \oplus \mathbb{N}$;
– For $d > \max(F(e))$, there exists a $k > d$ as described in the definition of $G(e, d)$, as otherwise M_e does not learn at least one of $\{2e, 2d + 1\} \oplus G(e, d)$ and $\{2e, 2d + 1\} \oplus \mathbb{N}$;
– Now the learner M_e has two different hypotheses on the segments $(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond \ldots \diamond 2F(e) + 1 \diamond \# \diamond 2k + 1 \diamond \# \diamond 4d + 2 \diamond \#^r)$ and $(4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond \ldots \diamond 2F(e) + 1 \diamond \# \diamond 2k + 1 \diamond \# \diamond 4d + 2 \diamond \#^r \diamond 2k + 1)$ and first of them must be correct hypothesis for $\{2e, 2d + 1\} \oplus G(e, d)$, as otherwise the learner M_e does not learn it from the text — $4e \diamond 1 \diamond \# \diamond 3 \diamond \# \diamond \ldots \diamond 2F(e) + 1 \diamond \# \diamond 2k + 1 \diamond \# \diamond 4d + 2 \diamond \#^r \diamond \#^\infty$ — see part (b) in the definition of $G(e, d)$, whereas second is a mind change, after the correct hypothesis by M_e on $\{2e, 2d + 1\} \oplus G(e, d)$.

Thus, M_e does not **TxtItSNUShEx**-learn $\mathcal{L}$. □

For our following proofs we will require the notion of a *canny* learner [CM08b].

Definition 6 (Case and Moelius [CM08b]). For all iterative learners M, we say that M is *canny* iff

1. M never outputs ?,

2. for all e, $M(e, \#) = e$ and
3. for all x, τ and σ, if $M^*(\sigma \diamond x) \neq M^*(\sigma)$ then $M^*(\sigma \diamond x \diamond \tau \diamond x) = M^*(\sigma \diamond x \diamond \tau)$.

Case and Moelius [CM08b] showed that, for **TxtItEx**-learning, learners can be assumed to be canny.

Lemma 7 (Case and Moelius [CM08b]). *For all $\mathcal{L} \in$ **TxtItEx** there exists canny iterative learner M such that $\mathcal{L} \subseteq$ **TxtItEx**(M).*

The term "sink-locking" means that on any text for a language to be learnt the learner converges to a *sink*, a correct hypothesis which is not abandoned on any continuation of the text. The following result does not only hold for the case where all texts are allowed but also for the case where only fat texts are allowed (see Section 5).

Theorem 8. *Let $\mathcal{L}$ be sink-lockingly* **TxtItEx**-*learnable. Then $\mathcal{L}$ is cautiously, conservatively, strongly decisively and weakly monotonically* **TxtItEx**-*learnable.*

The previous theorem gives us the following immediate corollary which states that a class is iteratively strongly decisive learnable from text iff it is iteratively conservatively learnable from text iff it is iteratively strongly non-U-shaped learnable from text.

Corollary 9. *We have that*

$$\mathbf{TxtItSDecEx} \;=\; \mathbf{TxtItConsvEx} \;=\; \mathbf{TxtItSNUShEx}.$$

Proof. We have that strongly decisive or conservative (iterative) learnability trivially implies strongly non-U-shaped learnability. Using Theorem 8 it remains to show that strongly non-U-shaped learnability implies sink-locking learnability. But this is trivial, as the learner can never converge to a correct conjecture that might possibly be abandoned on the given language, as this would contradict strong non-U-shapedness. □

Case and Moelius [CM08b] showed that **TxtItNUShEx** = **TxtItEx**; we finally show that this proof can be extended to also cover decisiveness, weak monotonicity and caution.

Theorem 10. *We have that*

$$\mathbf{TxtItEx} \;=\; \mathbf{TxtItDecEx} \;=\; \mathbf{TxtItWMonEx} \;= \mathbf{TxtItCautEx}.$$

Proof. Suppose M is a canny iterative learner which learns a class $\mathcal{L}$. Below we will construct an iterative learner N which is weakly monotonic and learns $\mathcal{L}$. Let

$$\begin{aligned}
C_M(\sigma) &= \{x \in \mathbb{N} \cup \{\#\} : M^*(\sigma \diamond x)\!\downarrow = M^*(\sigma)\!\downarrow\};\\
B_M(\sigma) &= \{x \in \mathbb{N} \cup \{\#\} : M^*(\sigma \diamond x)\!\downarrow \neq M^*(\sigma)\!\downarrow\};\\
B_M^{\cap}(\sigma) &= \bigcap_{0 \leq i \leq |\sigma|} B_M(\sigma[i]);\\
CB_M(\sigma) &= \bigcup_{0 \leq i < |\sigma|} C_M(\sigma[i]) \cap B_M(\sigma).
\end{aligned}$$

Let P be such that for all σ and m and $x \in \mathbb{N} \cup \{\#\}$, $P(\sigma, m, x)$ iff (i) $x \neq \#$ and (ii) $(\exists w)[M^*(\sigma \diamond w)$ converges in x steps, $W_{M^*(\sigma)}$ enumerates w in x steps, $w \in CB_M(\sigma)$ and $m < w \leq x]$.

Let N be such that $N(\emptyset) = f(\emptyset, 0, \emptyset)$, and for all inputs x, and previous conjecture $f(\sigma, m, \alpha)$, N outputs as follows:

$$\begin{cases} \uparrow, & \text{(i) if } M^*(\tau)\uparrow \text{ for some } \tau \in \{\sigma, \sigma \diamond \alpha, \sigma \diamond x, \sigma \diamond \alpha \diamond x\}; \\ f(\sigma \diamond \alpha \diamond x, 0, \emptyset), & \text{(ii) if } \neg \text{ (i) and } (x \in B^{\cap}_M(\sigma) \text{ or } (x \in CB_M(\sigma) \text{ and } x > m)); \\ f(\sigma, m, \alpha \diamond x), & \text{(iii) if } \neg \text{ ((i) or (ii)) and} \\ & x \in CB_M(\sigma \diamond \alpha) \\ f(\sigma, x, \emptyset), & \text{(iv) if } \neg \text{ ((i) or (ii)) and} \\ & x \in C_M(\sigma \diamond \alpha) \text{ and } P(\sigma, m, x) \text{ and } \alpha = \emptyset; \\ f(\sigma \diamond \alpha \diamond x, 0, \emptyset), & \text{(v) if } \neg \text{ ((i) or (ii)) and} \\ & x \in C_M(\sigma \diamond \alpha) \text{ and } P(\sigma, m, x) \text{ and } \alpha \neq \emptyset; \\ f(\sigma, m, \alpha), & \text{(vi) if } \neg \text{ ((i) or (ii)) and} \\ & x \in C_M(\sigma \diamond \alpha) \text{ and } \neg P(\sigma, m, x). \end{cases}$$

Here $W_{f(\sigma,m,\alpha)}$ is defined as follows.

1. Enumerate content(σ)
   In the following, if the needed $M^*(\cdot)$ (to compute various parameters), is not defined, then do not enumerate any more.
2. Go to stage 0.
   Stage s:
   Let $A_s = \text{content}(\sigma) \cup W_{M^*(\sigma),s}$
   (a) If there exists an $x \in A_s$ such that $x \in B^{\cap}_M(\sigma)$, then no more elements are enumerated.
   (b) If there exists an $x \in A_s$ such that $x > m$, and $[x \in CB_M(\sigma)$ or $P(\sigma, m, x)]$, then:
   If for all τ with content(τ) $\subseteq A_s$ and $|\tau| \leq |A_s| + 1$, τ not containing # and τ starting with a y in $CB_M(\sigma)$: $A_s \subseteq W_{f(\sigma \diamond \tau, 0, \emptyset)}$,
   then enumerate A_s and go to stage $s + 1$;
   otherwise, no more elements are enumerated.
   (basically, this is testing if x satisfies clauses ii, iv or v in the defn of M)
   (c) If both (a) and (b) fail, then enumerate A_s, and go to stage $s + 1$.
   End stage s

It can be easily shown by induction on the length of ρ, that for all input ρ, if $N^*(\rho) = f(\sigma, m, \alpha)$, then $M^*(\rho) = M^*(\sigma \diamond \alpha)$.

Now, for finite languages L iteratively learnt by M, if content(σ) $\subseteq L$ and $L \cap B^{\cap}_M(\sigma) = \emptyset$, then $W_{M^*(\sigma)} = L$. To see this note that if we construct a sequence τ from σ, by inserting elements of $L -$ content(σ) after the initial segment σ' of σ such that $x \in C_M(\sigma')$, then $M^*(\sigma) = M^*(\tau)$, and content(τ) $= L$; thus, $M^*(\sigma) = M^*(\sigma \#^\infty) = M^*(\tau \#^\infty)$, which must be a grammar for L. Thus for such σ, for content$(\alpha) \subseteq L$, using the fact that M is canny and using reverse induction on the number of mind changes made by M on σ (which is bounded by card(L) due to M being canny), it is easy to verify that $W_{f(\sigma,m,\alpha)}$ would be L.

Given an infinite languages $L \in \mathcal{L}$ and a text T for L, consider the output $f(\sigma_n, m_n, \alpha_n)$ of $N^*(T[n])$. As $M^*(T)$ converges, it holds that $\sigma = \lim_{n\to\infty} \sigma_n$ and $\lim_{n\to\infty} \alpha_n$ would converge. For this paragraph fix this σ and α. If $\alpha \neq \emptyset$, then clearly $m = \lim_{n\to\infty} m_n$ also converges, and as $B_M^{\cap}(\sigma) \cap L = \emptyset$, we also have $W_{M^*(\sigma)} = L$. If $\alpha = \emptyset$, then as $M^*(T) = M^*(\sigma)$, we have that $W_{M^*(\sigma)} = L$ and all but finitely many of the elements of L do not belong to $B_M(\sigma)$. Thus, in this case also $m = \lim_{n\to\infty} m_n$ converges. In both cases, m bounds all the elements of L which are in $B_M(\sigma)$. Thus, $f(\sigma, m, \alpha)$ would be a grammar for L.

We show the weak monotonicity of N. Note that, for all σ, α, m, $W_{f(\sigma,m,\alpha)} \subseteq$ content$(\sigma) \cup W_{M^*(\sigma)}$.

Also, note that $W_{f(\sigma,m,\alpha)} \subseteq W_{f(\sigma,m+1,\alpha')}$ for all $m, \alpha, \sigma, \alpha'$ — (P1).

Now suppose N on input $\rho \diamond x$ and previous conjecture (on input ρ) being $f(\sigma, m, \alpha)$ outputs $f(\sigma \diamond \alpha \diamond x, 0, \emptyset)$. This implies that, $x \in B_M^{\cap}(\sigma)$ or $x > m$ and $(CB_M(\sigma)$ or $P(\sigma, m, x))$ hold.

Case 1: content$(\alpha \diamond x)$ is not contained in $W_{f(\sigma,m,\alpha)}$.

In this case clearly content$(\rho \diamond x) \supseteq$ content$(\sigma \diamond \alpha \diamond x)$ and thus, content$(\rho \diamond x)$ is not contained in $W_{f(\sigma,m,\alpha)}$, so mind change is safe.

Case 2: content$(\alpha \diamond x)$ is contained in $W_{f(\sigma,m,\alpha)}$ and thus in content$(\sigma) \cup W_{M^*(\sigma)}$.

Let s be least such that content$(\alpha \diamond x)$ is contained in A_s as in stage s. Then, the definition of $W_{f(\sigma,m,\alpha)}$ ensures that $W_{f(\sigma,m,\alpha)}$ enumerates $A_t, t \geq s$, only if A_t is contained in $W_{f(\sigma\diamond\alpha\diamond x,0,\emptyset)}$ (note that the case of $A_t =$ content(σ), already satisfies $A_t \subseteq W_{f(\sigma\diamond\alpha\diamond x,0,\emptyset)}$).

It follows from the above analysis that either the new input is not contained in the previous conjecture of N, or the previous conjecture is contained in the new conjecture. Thus, N is weakly monotonic.

It follows from the above construction that N is also decisive and cautious. To see this, note that whenever mind change of N falls in Case 1 above, all future conjectures of N (beyond input $\rho \diamond x$) contain content$(\alpha \diamond x)$; thus, N never returns to the conjecture $W_{f(\sigma,m,\alpha)}$, which does not contain content$(\alpha \diamond x)$. On the other hand, the mind changes due to Case 2 or mind changes due to N outputting $f(\sigma, m', \alpha')$ after outputting $f(\sigma, m, \alpha)$, are strongly monotonic (see the discussion in Case 2, as well as property (P1) mentioned above). The theorem follows. □

5 Learning from Fat-Texts and other Texts

In this section we deal with special kinds of texts. A text is called *fat* iff every datum appears infinitely often in that text. A text T is called *one-one* iff for all $x \in$ content(T), there exists a unique n such that $T(n) = x$. We let fat denote the set of all fat texts and one − one the set of all one-one texts. Standard techniques can be used to show the following result.

Theorem 11. $\mathbf{TxtItEx} \subset \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx} \subset \mathbf{TxtGEx}$.

The above result shows that iterative learners have not only information-theoretic limitations in that they forget past data and cannot recover them (on normal text), but also computational limitations which cannot be compensated by having fat text. Next we show that fat text always allows for learning conservatively (as well as cautiously and strongly decisively).

Theorem 12. $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItEx} = \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItConsvEx} = \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItSDecEx}$.

Proposition 13. *(a) There exists a class of languages which is* **TxtItMonEx***,* **TxtItSDecEx***,* **TxtItConsvEx***-learnable but not* $\mathbf{Txt}^{\mathrm{fat}}\mathbf{SMonEx}$*-learnable.*

(b) There is a class which is **TxtItSDecEx***-learnable (and therefore also* **TxtItConsvEx***-learnable) but not* $\mathbf{Txt}^{\mathrm{fat}}\mathbf{ItMonEx}$ *or* $\mathbf{Txt}^{\mathrm{one-one}}\mathbf{ItMonEx}$*-learnable.*

Theorem 14. $\mathbf{TxtItSMonEx} \not\subseteq \mathbf{Txt}^{\mathrm{fat}}\mathbf{ItSNUShEx}$.

We next show that learning from one-one texts is equivalent to learning from arbitrary text.

Theorem 15. $\mathbf{Txt}^{\mathrm{one-one}}\mathbf{ItEx} = \mathbf{TxtItEx}$.

Theorem 16. *There exists a class $\mathcal{L}$ which is* $\mathbf{Txt}^{\mathrm{one-one}}\mathbf{ItFex}$*-learnable but not* $\mathbf{Txt}^{\mathrm{one-one}}\mathbf{Ex}$*-learnable. Therefore $\mathcal{L}$ is not* **TxtItEx***-learnable (and hence not* **TxtItFex***-learnable).*

Proof. Let $\mathcal{L}$ consist of the languages $L_{e,z}$, $z \leq e$, $e, z \in \mathbb{N}$, where $L_{e,z} = \{(e,x,y) : x = z \text{ or } x + y < |W_e|\}$.

The learner on seeing any input element (e,x,y), outputs a grammar (obtained effectively from (e,x)) for $L_{e,\min(\{e,x\})}$.

If W_e is infinite, then $L_{e,e} = L_{e,z}$ for all $z \leq e$, and thus all the (finitely many) grammars output by the learner are for $L_{e,e}$.

If W_e is finite, then $L_{e,z}$ contains only finitely many elements which are not of the form $(e,z,\cdot)$, and thus on any one-one text for $L_{e,z}$, the learner converges to a grammar for $L_{e,z}$.

We now show that $\mathcal{L}$ is not **TxtEx**-learnable. Suppose otherwise that some learner **TxtEx**-learns $\mathcal{L}$. Then, for $e \geq 2$, W_e is infinite iff the learner has a stabilising sequence [BB75, Ful90] τ on the set $\{(e,x,y) : x, y \in \mathbb{N}\}$ and the largest sum $x + y$ for some (e,x,y) occurring in τ is below $|W_e|$. Thus it would be a Σ_2 condition to check whether W_e is infinite in contradiction to the fact that checking whether W_e is infinite is Π_2 complete. Thus such a learner does not exist. □

Theorem 17. *There exists a class of languages which is iteratively learnable using texts where every element which is maximal so far is marked, but is not* **TxtItEx***-learnable.*

6 Class Preserving Hypotheses Spaces

A one-one hypothesis space might be considered in order to prevent that an iterative learner cheats by storing information in the hypothesis. A hypothesis space $(H_e)_{e\in\mathbb{N}}$ is called class preserving (for learning $\mathcal{L}$) iff $\{H_e : e \in \mathbb{N}\} = \mathcal{L}$. A learner is class preserving, if the hypothesis space used by it is class preserving. The first result shows that the usage of one-one texts increases the learning power of those iterative learners which are forced to use one-one hypothesis spaces, that is, which cannot store information in the hypothesis during the learning process.

Theorem 18. *There exists a class $\mathcal{L}$ having a one-one class preserving hypothesis space such that the following conditions hold:*

(a) $\mathcal{L}$ can be $\mathbf{Txt}^{\text{one}-\text{one}}\mathbf{ItEx}$*-learnt using any fixed one-one class preserving hypothesis space for $\mathcal{L}$;*

(b) $\mathcal{L}$ cannot be **TxtItEx***-learnt using any fixed one-one class preserving hypothesis space for $\mathcal{L}$.*

In general, the hierarchy **SMonEx** $\subseteq$ **MonEx** $\subseteq$ **WMonEx** holds. The following result shows that this hierarchy is proper and that one can get the separations even in the case that the more general criterion is made stricter by enforcing the use of a one-one hypothesis space.

Theorem 19. *(a)* **TxtItWMonEx** $\not\subseteq$ **TxtItMonEx***;*
(b) **TxtItMonEx** $\not\subseteq$ **TxtItSMonEx***.*
Here the positive sides can be shown using a one-one class preserving hypothesis space.

Theorem 2 and Theorem 3 show the above result and also provide conservatively learnable families for these separations. We now consider learning by *reliable* learners. A learner is *reliable* if it is total and for any text T, if the learner converges on T to a hypothesis e, then e is a correct grammar for content(T). We denote the reliability constraint on the learner by using **Rel** in the criterion name. For the following result, we assume (by definition) that if a learner converges to ? on a text, then it is not reliable. The next result shows that there is exactly one class which has a reliable iterative learner using a one-one class preserving hypothesis space and this is the class FIN $= \{L : L \text{ is finite}\}$.

Theorem 20. *If $\mathcal{L}$ is* **TxtItRelEx***-learnable using a one-one class preserving hypothesis space then $\mathcal{L}$ must be* FIN.

Theorem 21. *There exists a subclass of* FIN *which is not* **TxtItEx***-learnable using a one-one class preserving hypothesis space.*

Note that in learning theory without loss of generality one assumes that classes are not empty. The next theorem characterises when a class can be iteratively and reliably learnt using a class preserving hypothesis space: it is the case if and only if the set of canonical indices of the languages in the class is recursively enumerable. Note that the hypothesis space considered here is not one-one and that padding is a natural ingredient of the (omitted) learning algorithm.

Theorem 22. *A class $\mathcal{L}$ has a class-preserving iterative and reliable learner iff it does not contain infinite languages and the set $\{e : D_e \in \mathcal{L}\}$ of its canonical indices is recursively enumerable.*

7 Syntactic versus Semantic Conservativeness

A learner is called *semantically conservative* iff whenever it outputs two indices i, j such that $W_i \neq W_j$ and i is output before j then the hypothesis j is based on some observed data not contained in W_i. This notion coincides with syntactic conservative learning in the case of standard explanatory learning; however, in the special case of iterative learning, it is more powerful than the usual notion of conservative learning.

Theorem 23. *There is a class $\mathcal{L}$ which can be learnt iteratively and strongly monotonically and semantically conservatively but which does not have an iterative and syntactically conservative learner.*

References

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

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

[BCMSW08] Baliga, G., Case, J., Merkle, W., Stephan, F., Wiehagen, R.: When unlearning helps. Information and Computation 206, 694–709 (2008)

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

[Cas94] Case, J.: Infinitary self-reference in learning theory. Journal of Experimental and Theoretical Artificial Intelligence 6, 3–16 (1994)

[Cas99] Case, J.: The power of vacillation in language learning. SIAM Journal on Computing 28, 1941–1969 (1999)

[CK10] Case, J., Kötzing, T.: Strongly non-U-shaped learning results by general techniques. In: Proceedings of COLT (Conference on Computational Learning Theory), pp. 181–193 (2010)

[CL82] Case, J., Lynes, C.: Machine inductive inference and language identification. In: Proceedings of ICALP (International Colloquium on Automata, Languages and Programming), pp. 107–115 (1982)

[CM08a] Case, J., Moelius III, S.E.: Optimal language learning. In: Freund, Y., Györfi, L., Turán, G., Zeugmann, T. (eds.) ALT 2008. LNCS (LNAI), vol. 5254, pp. 419–433. Springer, Heidelberg (2008)

[CM08b] Case, J., Moelius, S.E.: U-shaped, iterative, and iterative-with-counter learning. Machine Learning 72, 63–88 (2008)

[Ful90] Fulk, M.: Prudence and other conditions on formal language learning. Information and Computation 85, 1–11 (1990)

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

[GL04] Grieser, G., Lange, S.: Incremental learning of approximations from positive data. Information Processing Letters 89, 37–42 (2004)

[Jan91] Jantke, K.-P.: Monotonic and non-monotonic inductive inference of functions and patterns. In: Dix, J., Schmitt, P.H., Jantke, K.P. (eds.) NIL 1990. LNCS, vol. 543, pp. 161–177. Springer, Heidelberg (1991)

[JMZ13] Jain, S., Moelius, S.E., Zilles, S.: Learning without coding. Theoretical Computer Science 473, 124–148 (2013)

[JORS99] Jain, S., Osherson, D., Royer, J., Sharma, A.: Systems that Learn: An Introduction to Learning Theory, 2nd edn. MIT Press, Cambridge (1999)

[Köt09] Kötzing, T.: Abstraction and Complexity in Computational Learning in the Limit. PhD thesis, University of Delaware (2009), `http://pqdtopen.proquest.com/#viewpdf?dispub=3373055`

[Köt14] Kötzing, T.: A Solution to Wiehagen's Thesis. In: Symposium on Theoretical Aspects of Computer Science (STACS 2014), pp. 494–505 (2014)

[LG02] Lange, S., Grieser, G.: On the power of incremental learning. Theoretical Computer Science 288, 277–307 (2002)

[LG03] Lange, S., Grieser, G.: Variants of iterative learning. Theoretical Computer Science 292, 359–376 (2003)

[LZ93] Lange, S., Zeugmann, T.: Monotonic versus non-monotonic language learning. In: Brewka, G., Jantke, K.P., Schmitt, P.H. (eds.) NIL 1991. LNCS, vol. 659, pp. 254–269. Springer, Heidelberg (1993)

[LZ96] Lange, S., Zeugmann, T.: Incremental learning from positive data. Journal of Computer and System Sciences 53, 88–103 (1996)

[LZZ08] Lange, S., Zeugmann, T., Zilles, S.: Learning indexed families of recursive languages from positive data: a survey. Theoretical Computer Science 397, 194–232 (2008)

[OSW82] Osherson, D., Stob, M., Weinstein, S.: Learning strategies. Information and Control 53, 32–51 (1982)

[OSW86] Osherson, D., Stob, M., Weinstein, S.: Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists. MIT Press, Cambridge (1986)

[OW82] Osherson, D., Weinstein, S.: Criteria of language learning. Information and Control 52, 123–138 (1982)

[RC94] Royer, J., Case, J.: Subrecursive Programming Systems: Complexity and Succinctness. Research monograph in Progress in Theoretical Computer Science. Birkhäuser, Basel (1994)

[Rog67] Rogers, H.: Theory of Recursive Functions and Effective Computability. McGraw Hill, New York (1967); Reprinted by MIT Press, Cambridge (1987)

[Wie91] Wiehagen, R.: A thesis in inductive inference. *Nonmonotonic and Inductive Logic*. In: Dix, J., Schmitt, P.H., Jantke, K.P. (eds.) NIL 1990. LNCS, vol. 543, pp. 184–207. Springer, Heidelberg (1991)