A Map of Update Constraints in Inductive Inference

Timo Kötzing and Raphaela Palenta*

Friedrich-Schiller-Universität Jena, Germany
{timo.koetzing,raphaela-julia.palenta}@uni-jena.de

Abstract. We investigate how different learning restrictions reduce learning power and how the different restrictions relate to one another. We give a complete map for nine different restrictions both for the cases of complete information learning and set-driven learning. This completes the picture for these well-studied *delayable* learning restrictions. A further insight is gained by different characterizations of *conservative* learning in terms of variants of *cautious* learning.

Our analyses greatly benefit from general theorems we give, for example showing that learners with exclusively delayable restrictions can always be assumed total.

1 Introduction

This paper is set in the framework of *inductive inference*, a branch of (algorithmic) learning theory. This branch analyzes the problem of algorithmically learning a description for a formal language (a computably enumerable subset of the set of natural numbers) when presented successively all and only the elements of that language. For example, a learner h might be presented more and more even numbers. After each new number, h outputs a description for a language as its conjecture. The learner h might decide to output a program for the set of all multiples of 4, as long as all numbers presented are divisible by 4. Later, when h sees an even number not divisible by 4, it might change this guess to a program for the set of all multiples of 2.

Many criteria for deciding whether a learner h is *successful* on a language L have been proposed in the literature. Gold, in his seminal paper [Gol67], gave a first, simple learning criterion, **TxtGEx**-*learning*[1], where a learner is *successful* iff, on every *text* for L (listing of all and only the elements of L) it eventually stops changing its conjectures, and its final conjecture is a correct description for the input sequence. Trivially, each single, describable language L has a suitable constant function as a **TxtGEx**-learner (this learner constantly outputs a description for L). Thus, we are interested in analyzing for which *classes of languages* $\mathcal{L}$ there is a *single learner* h learning *each* member of $\mathcal{L}$. This framework

* We would like to thank the reviewers for their very helpful comments.

[1] **Txt** stands for learning from a *text* of positive examples; **G** stands for Gold, who introduced this model, and is used to to indicate full-information learning; **Ex** stands for *explanatory*.

P. Auer et al. (Eds.): ALT 2014, LNAI 8776, pp. 40–54, 2014.

is also sometimes known as *language learning in the limit* and has been studied extensively, using a wide range of learning criteria similar to **TxtGEx**-learning (see, for example, the textbook [JORS99]).

A wealth of learning criteria can be derived from **TxtGEx**-learning by adding restrictions on the intermediate conjectures and how they should relate to each other and the data. For example, one could require that a conjecture which is consistent with the data must not be changed; this is known as *conservative* learning and known to restrict what classes of languages can be learned ([Ang80], we use **Conv** to denote the restriction of conservative learning). Additionally to conservative learning, the following learning restrictions are considered in this paper (see Section 2.1 for a formal definition of learning criteria including these learning restrictions).

In *cautious* learning (**Caut**, [OSW82]) the learner is not allowed to ever give a conjecture for a strict subset of a previously conjectured set. In *non-U-shaped* learning (**NU**, [BCM$^+$08]) a learner may never *semantically* abandon a correct conjecture; in *strongly non-U-shaped* learning (**SNU**, [CM11]) not even syntactic changes are allowed after giving a correct conjecture.

In *decisive* learning (**Dec**, [OSW82]), a learner may never (semantically) return to a *semantically* abandoned conjecture; in *strongly decisive* learning (**SDec**, [Köt14]) the learner may not even (semantically) return to *syntactically* abandoned conjectures. Finally, a number of monotonicity requirements are studied ([Jan91, Wie91, LZ93]): in *strongly monotone* learning (**SMon**) the conjectured sets may only grow; in *monotone* learning (**Mon**) only incorrect data may be removed; and in *weakly monotone* learning (**WMon**) the conjectured set may only grow while it is consistent.

The main question is now whether and how these different restrictions reduce learning power. For example, non-U-shaped learning is known not to restrict the learning power [BCM$^+$08], and the same for strongly non-U-shaped learning [CM11]; on the other hand, decisive learning *is* restrictive [BCM$^+$08]. The relations of the different monotone learning restriction were given in [LZ93]. Conservativeness is long known to restrict learning power [Ang80], but also known to be equivalent to weakly monotone learning [KS95, JS98].

Cautious learning was shown to be a restriction but not when added to conservativeness in [OSW82, OSW86], similarly the relationship between decisive and conservative learning was given. In Exercise 4.5.4B of [OSW86] it is claimed (without proof) that cautious learners cannot be made conservative; we claim the opposite in Theorem 13.

This list of previously known results leaves a number of relations between the learning criteria open, even when adding trivial inclusion results (we call an inclusion trivial iff it follows straight from the definition of the restriction without considering the learning model, for example strongly decisive learning is included in decisive learning; formally, trivial inclusion is inclusion on the level of learning restrictions as predicates, see Section 2.1). With this paper we now give the complete picture of these learning restrictions. The result is shown as a map in Figure 1. A solid black line indicates a trivial inclusion (the lower

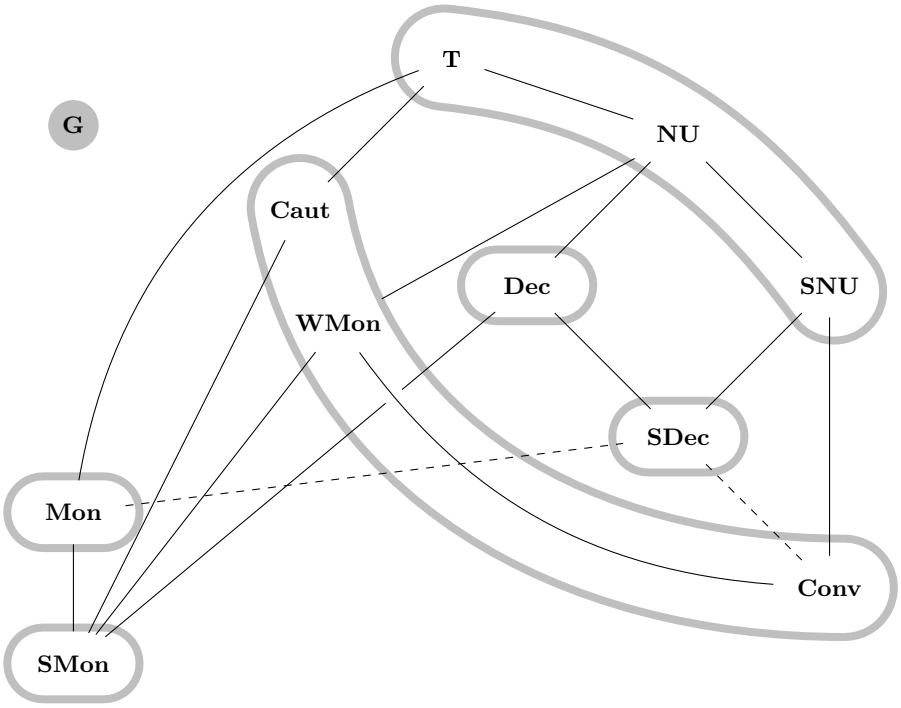


Fig. 1. Relation of criteria

criterion is included in the higher); a dashed black line indicates inclusion (which is not trivial). A gray box around criteria indicates equality of (learning of) these criteria.

A different way of depicting the same results is given in Figure 2 (where solid lines indicate any kind of inclusion). Results involving monotone learning can be found in Section 6, all others in Section 4.

For the important restriction of conservative learning we give the characterization of being equivalent to cautious learning. Furthermore, we show that even two weak versions of cautiousness are equivalent to conservative learning. Recall that cautiousness forbids to return to a strict subset of a previously conjectured set. If we now weaken this restriction to forbid to return to *finite* subsets of a previously conjectured set we get a restriction still equivalent to conservative learning. If we forbid to go down to a correct conjecture, effectively forbidding to ever conjecture a superset of the target language, we also obtain a restriction equivalent to conservative learning. On the other hand, if we weaken it so as to only forbid going to *infinite* subsets of previously conjectured sets, we obtain a restriction equivalent to no restriction. These results can be found in Section 4.

In *set-driven* learning [WC80] the learner does not get the full information about what data has been presented in what order and multiplicity; instead, the learner only gets the set of data presented so far. For this learning model it is

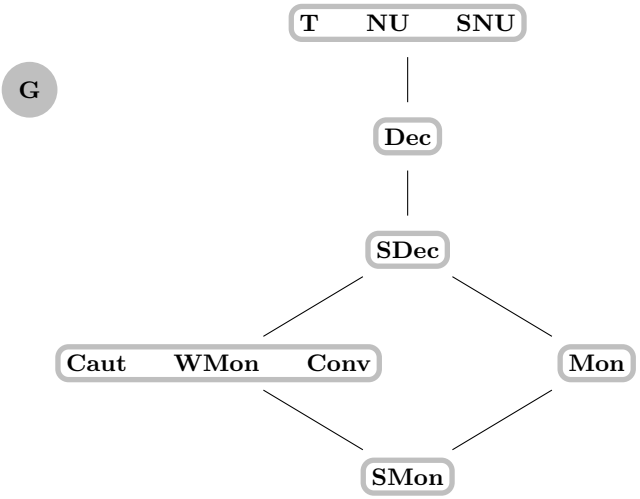


Fig. 2. Partial order of delayable learning restrictions in Gold-style learning

known that, surprisingly, conservative learning is no restriction [KS95]! We complete the picture for set driven learning by showing that set-driven learners can always be assumed conservative, strongly decisive and cautious, and by showing that the hierarchy of monotone and strongly monotone learning also holds for set-driven learning. The situation is depicted in Figure 3. These results can be found in Section 5.

1.1 Techniques

A major emphasis of this paper is on the techniques used to get our results. These techniques include specific techniques for specific problems, as well as general theorems which are applicable in many different settings. The general

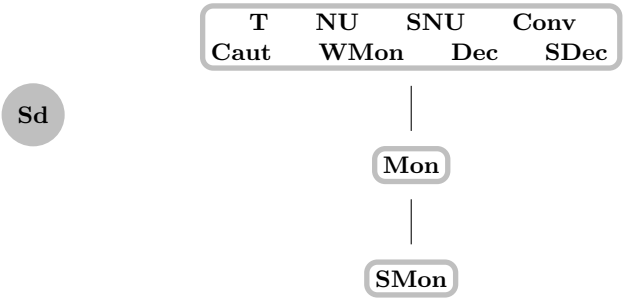


Fig. 3. Hierarchy of delayable learning restrictions in set-driven learning

techniques are given in Section 3, one main general result is as follows. It is well-known that any **TxtGEx**-learner h learning a language L has a *locking sequence*, a sequence σ of data from L such that, for any further data from L, the conjecture does not change and is correct. However, there might be texts such that no initial sequence of the text is a locking sequence. We call a learner such that any text for a target language contains a locking sequence *strongly locking*, a property which is very handy to have in many proofs. Fulk [Ful90] showed that, without loss of generality, a **TxtGEx**-learner can be assumed strongly locking, as well as having many other useful properties (we call this the *Fulk normal form*, see Definition 8). For many learning criteria considered in this paper it might be too much to hope for that all of them allow for learning by a learner in Fulk normal form. However, we show in Corollary 7 that we can always assume our learners to be strongly locking, total, and what we call *syntactically decisive*, never *syntactically* returning to syntactically abandoned hypotheses.

The main technique we use to show that something is decisively learnable, for example in Theorem 22, is what we call *poisoning* of conjectures. In the proof of Theorem 22 we show that a class of languages is decisively learnable by simulating a given monotone learner h, but changing conjectures as follows. Given a conjecture e made by h, if there is no mind change in the future with data from conjecture e, the new conjecture is equivalent to e; otherwise it is suitably changed, *poisoned*, to make sure that the resulting learner is decisive. This technique was also used in [CK10] to show strongly non-U-shaped learnability.

Finally, for showing classes of languages to be not (strongly) decisively learnable, we adapt a technique known in computability theory as a "priority argument" (note, though, that we do not deal with oracle computations). We use this technique to reprove that decisiveness is a restriction to **TxtGEx**-learning (as shown in [BCM$^+$08]), and then use a variation of the proof to show that strongly decisive learning is a restriction to decisive learning.

Due to space constraints, we cannot give all proofs in this version of the paper. The full version of the paper can be found at `http://arxiv.org/abs/1404.7527`.

2 Mathematical Preliminaries

Unintroduced notation follows [Rog67], a textbook on computability theory.

$\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. The symbols $\subseteq$, $\subset$, $\supseteq$, $\supset$ respectively denote the subset, proper subset, superset and proper superset relation between sets; $\setminus$ denotes set difference. $\emptyset$ and λ denote the empty set and the empty sequence, respectively. The quantifier $\forall^\infty x$ means "for all but finitely many x". With dom and range we denote, respectively, domain and range of a given function.

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. For finite sequences, we suppose that

for any $\sigma \subseteq \tau$ we have that the code number of σ is at most the code number of τ. We let $\mathbb{S}\mathrm{eq}$ denote the set of all (finite) sequences, and $\mathbb{S}\mathrm{eq}_{\leq t}$ the (finite) set of all sequences of length at most t using only elements $\leq t$.

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 $f(x)\downarrow = p$. We let $\mathfrak{P}$ denote the set of all partial functions $\mathbb{N} \to \mathbb{N}$ and $\mathfrak{R}$ the set of all total such functions.

$\mathcal{P}$ and $\mathcal{R}$ denote, respectively, the set of all partial computable and the set of all total computable functions (mapping $\mathbb{N} \to \mathbb{N}$).

We let φ be any fixed acceptable programming system for $\mathcal{P}$ (an acceptable programming system could, for example, be based on a natural programming language such as C or Java, or on Turing machines). Further, we let φ_p denote the partial computable function computed by the φ-program with code number p. A set $L \subseteq \mathbb{N}$ is *computably enumerable (*ce*)* iff it is the domain of a computable function. Let $\mathcal{E}$ denote the set of all ce sets. We let W be the mapping such that $\forall e : W(e) = \mathrm{dom}(\varphi_e)$. For each e, we write W_e instead of $W(e)$. 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.

We let Φ be a Blum complexity measure associated with φ (for example, for each e and x, $\Phi_e(x)$ could denote the number of steps that program e takes on input x before terminating). For all e and t we let $W_e^t = \{x \leq t \mid \Phi_e(x) \leq t\}$ (note that a complete description for the finite set W_e^t is computable from e and t). The symbol # is pronounced *pause* and is used to symbolize "no new input data" in a 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 computable functions. For any function T and all i, we use $T[i]$ to denote the sequence $T(0), \ldots, T(i-1)$ (the empty sequence if $i = 0$ and undefined, if any of these values is undefined).

2.1 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 computable function $h \in \mathcal{P}$. A *language* is a ce 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.

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

We define the interaction operators **G**, **Psd** (partially set-driven learning, [SR84]) and **Sd** (set-driven learning, [WC80]) as follows. For all learners h, texts T and all i,

$$\begin{aligned}\mathbf{G}(h,T)(i) &= h(T[i]);\\ \mathbf{Psd}(h,T)(i) &= h(\text{content}(T[i]),i);\\ \mathbf{Sd}(h,T)(i) &= h(\text{content}(T[i])).\end{aligned}$$

Thus, in set-driven learning, the learner has access to the set of all previous data, but not to the sequence as in **G**-learning. In partially set-driven learning, the learner has the set of data and the current iteration number.

Successful learning requires the learner to observe certain restrictions, for example convergence to a correct index. These restrictions are formalized 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]) defined such that, for all sequences of hypotheses p and all texts T,

$$\mathbf{Ex}(p,T) \Leftrightarrow p \text{ total } \wedge [\exists n_0 \forall n \geq n_0 : p(n) = p(n_0) \wedge W_{p(n_0)} = \text{content}(T)].$$

Furthermore, we formally define the restrictions discussed in Section 1 in Figure 4 (where we implicitly require the learning sequence p to be total, as in **Ex**-learning; note that this is a technicality without major importance).

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

Fig. 4. Definitions of learning restrictions

A variant on decisiveness is *syntactic decisiveness*, **SynDec**, a technically useful property defined as follows.

$$\mathbf{SynDec}(p,T) \Leftrightarrow [\forall i,j,k : i \leq j \leq k \ \wedge \ p(i) = p(k) \Rightarrow p(j) = p(i)].$$

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 4 are meant to be always used together with **Ex**). With **T** we denote the always true sequence acceptance criterion (no restriction on learning).

A *learning criterion* is a tuple $(\mathcal{C}, \beta, \delta)$, where $\mathcal{C}$ is a set of learners (the admissible learners), β is an interaction operator and δ is a learning restriction;

we usually write $\mathcal{C}\mathbf{Txt}\beta\delta$ to denote the learning criterion, omitting $\mathcal{C}$ in case of $\mathcal{C} = \mathcal{P}$. We say that a learner $h \in \mathcal{C}$ $\mathcal{C}\mathbf{Txt}\beta\delta$-*learns* a language L iff, for all texts T for L, $\delta(\beta(h, T), T)$. The set of languages $\mathcal{C}\mathbf{Txt}\beta\delta$-learned by $h \in \mathcal{C}$ is denoted by $\mathcal{C}\mathbf{Txt}\beta\delta(h)$. We write $[\mathcal{C}\mathbf{Txt}\beta\delta]$ to denote the set of all $\mathcal{C}\mathbf{Txt}\beta\delta$-learnable classes (learnable by some learner in $\mathcal{C}$).

3 Delayable Learning Restrictions

In this section we present technically useful results which show that learners can always be assumed to be in some normal form. We will later always assume our learners to be in the normal form established by Corollary 7, the main result of this section. We start with the definition of *delayable*. Intuitively, a learning criterion δ is delayable iff the output of a hypothesis can be arbitrarily (but not indefinitely) delayed.

Definition 1. Let $\vec{R}$ be the set of all non-decreasing $r : \mathbb{N} \to \mathbb{N}$ with infinite limit inferior, i.e. for all m we have $\forall^\infty n : r(n) \geq m$.

A learning restriction δ is *delayable* iff, for all texts T and T' with $\text{content}(T) = \text{content}(T')$, all p and all $r \in \vec{R}$, if $(p, T) \in \delta$ and $\forall n : \text{content}(T[r(n)]) \subseteq \text{content}(T'[n])$, then $(p \circ r, T') \in \delta$. Intuitively, as long as the learner has at least as much data as was used for a given conjecture, then the conjecture is permissible. Note that this condition holds for $T = T'$ if $\forall n : r(n) \leq n$.

Note that the intersection of two delayable learning criteria is again delayable and that *all* learning restrictions considered in this paper are delayable.

As the name suggests, we can apply *delaying tricks* (tricks which delay updates of the conjecture) in order to achieve fast computation times in each iteration (but of course in the limit we still spend an infinite amount of time). This gives us equally powerful but total learners, as shown in the next theorem. While it is well-known that, for many learning criteria, the learner can be assumed total, this theorem explicitly formalizes conditions under which totality can be assumed (note that there are also natural learning criteria where totality cannot be assumed, such as consistent learning [JORS99]).

Theorem 2. For any delayable learning restriction δ, we have $[\mathbf{Txt}\mathbf{G}\delta] = [\mathcal{R}\mathbf{Txt}\mathbf{G}\delta]$.

Next we define another useful property, which can always be assumed for delayable learning restrictions.

Definition 3. A *locking sequence for a learner h on a language L* is any finite sequence σ of elements from L such that $h(\sigma)$ is a correct hypothesis for L and, for sequences τ with elements from L, $h(\sigma \diamond \tau) = h(\sigma)$[BB75]. It is well known that every learner h learning a language L has a locking sequence on L. We say that a learning criterion I *allows for strongly locking learning* iff, for each I-learnable class of languages $\mathcal{L}$ there is a learner h such that h I-learns $\mathcal{L}$ and, for each $L \in \mathcal{L}$ and any text T for L, there is an n such that $T[n]$ is a locking sequence of h on L (we call such a learner h *strongly locking*).

With this definition we can give the following theorem.

Theorem 4. Let δ be a delayable learning criterion. Then $\mathcal{R}\mathbf{Txt}\mathbf{G}\delta\mathbf{Ex}$ allows for strongly locking learning.

Next we define semantic and pseudo-semantic restrictions introduced in [Köt14]. Intuitively, semantic restrictions allow for replacing hypotheses by equivalent ones; pseudo-sematic restrictions allow the same, as long as no new mind changes are introduced.

Definition 5. For all total functions $p \in \mathfrak{P}$, we let

$$\begin{aligned}\mathrm{Sem}(p) &= \{p' \in \mathfrak{P} \mid \forall i : W_{p(i)} = W_{p'(i)}\};\\ \mathrm{Mc}(p) &= \{p' \in \mathfrak{P} \mid \forall i : p'(i) \neq p'(i+1) \Rightarrow p(i) \neq p(i+1)\}.\end{aligned}$$

A sequence acceptance criterion δ is said to be a *semantic restriction* iff, for all $(p, q) \in \delta$ and $p' \in \mathrm{Sem}(p)$, $(p', q) \in \delta$.

A sequence acceptance criterion δ is said to be a *pseudo-semantic restriction* iff, for all $(p, q) \in \delta$ and $p' \in \mathrm{Sem}(p) \cap \mathrm{Mc}(p)$, $(p', q) \in \delta$.

We note that the intersection of two (pseudo-) semantic learning restrictions is again (pseudo-) semantic. All learning restrictions considered in this paper are pseudo-semantic, and all except **Conv**, **SNU**, **SDec** and **Ex** are semantic.

The next lemma shows that, for every pseudo-semantic learning restriction, learning can be done syntactically decisively.

Lemma 6. Let δ be a pseudo-semantic learning criterion. Then we have

$$[\mathcal{R}\mathbf{Txt}\mathbf{G}\delta] = [\mathcal{R}\mathbf{Txt}\mathbf{G}\mathbf{SynDec}\delta].$$

As **SynDec** is a delayable learning criterion, we get the following corollary by taking Theorems 2 and 4 and Lemma 6 together. We will always assume our learners to be in this normal form in this paper.

Corollary 7. Let δ be pseudo-semantic and delayable. Then $\mathbf{Txt}\mathbf{G}\delta\mathbf{Ex}$ allows for strongly locking learning by a syntactically decisive total learner.

Fulk showed that any $\mathbf{Txt}\mathbf{G}\mathbf{Ex}$-learner can be (effectively) turned into an equivalent learner with many useful properties, including strongly locking learning [Ful90]. One of the properties is called *order-independence*, meaning that on any two texts for a target language the learner converges to the same hypothesis. Another property is called *rearrangement-independence*, where a learner h is rearrangement-independent if there is a function f such that, for all sequences σ, $h(\sigma) = f(\mathrm{content}(\sigma), |\sigma|)$ (intuitively, rearrangement independence is equivalent to the existence of a partially set-driven learner for the same language). We define the collection of all the properties which Fulk showed a learner can have to be the *Fulk normal form* as follows.

Definition 8. We say a **TxtGEx**-learner h is in *Fulk normal form* if (1) – (5) hold.

(1) h is order-independent.
(2) h is rearrangement-independent.
(3) If h **TxtGEx**-learns a language L from some text, then h **TxtGEx**-learns L.
(4) If there is a locking sequence of h for some L, then h **TxtGEx**-learns L.
(5) For all $\mathcal{L} \in \mathbf{TxtGEx}(h)$, h is strongly locking on $\mathcal{L}$.

The following theorem is somewhat weaker than what Fulk states himself.

Theorem 9 ([Ful90, Theorem 13]). Every **TxtGEx**-learnable set of languages has a **TxtGEx**-learner in Fulk normal form.

4 Full-Information Learning

In this section we consider various versions of cautious learning and show that all of our variants are either no restriction to learning, or equivalent to conservative learning as is shown in Figure 5.

Additionally, we will show that every cautious **TxtGEx**-learnable language is conservative **TxtGEx**-learnable which implies that [**TxtGConvEx**], [**TxtGWMonEx**] and [**TxtGCautEx**] are equivalent. Last, we will separate these three learning criteria from strongly decisive **TxtGEx**-learning and show that [**TxtGSDecEx**] is a proper superset.

Theorem 10. We have that any conservative learner can be assumed cautious and strongly decisive, i.e.

$$[\mathbf{TxtGConvEx}] = [\mathbf{TxtGConvSDecCautEx}].$$

Proof. Let $h \in \mathcal{R}$ and $\mathcal{L}$ be such that h **TxtGConvEx**-learns $\mathcal{L}$. We define, for all σ, a set $M(\sigma)$ as follows

$$M(\sigma) = \{\tau \mid \tau \subseteq \sigma \ \wedge \ \forall x \in \mathrm{content}(\tau) : \Phi_{h(\tau)}(x) \leq |\sigma|\}.$$

We let

$$\forall \sigma : h'(\sigma) = h(\max(M(\sigma))).$$

Let T be a text for a language $L \in \mathcal{L}$. We first show that h' **TxtGEx**-learns L from the text T. As h **TxtGConvEx**-learns L, there are n and e such that $\forall n' \geq n : h(T[n]) = h(T[n']) = e$ and $W_e = L$. Thus, there is $m \geq n$ such that $\forall x \in \mathrm{content}(T[n]) : \Phi_{h(T[n])}(x) \leq m$ and therefore $\forall m' \geq m : h'(T[m]) = h'(T[m']) = e$.

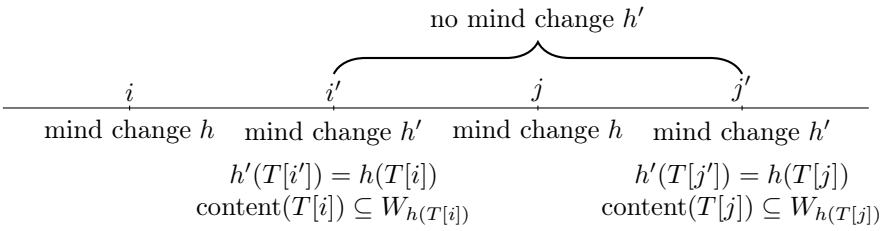
Next we show that h' is strongly decisive and conservative; for that we show that, with every mind change, there is a new element of the target included in the conjecture which is currently not included but is included in all future conjectures; it is easy to see that this property implies both caution and strong decisiveness. Let i and i' be such that $\max(M(T[i'])) = T[i]$. This implies that

$$\mathrm{content}(T[i]) \subseteq W_{h'(T[i'])}.$$

Let $j' > i'$ such that $h'(T[i']) \neq h'(T[j'])$. Then there is $j > i$ such that $\max(M(T[j'])) = T[j]$ and therefore

$$\text{content}(T[j]) \subseteq W_{h'(T[j'])}.$$

Note that in the following diagram j could also be between i and i'.



As h is conservative and $\text{content}(T[i]) \subseteq W_{h(T[i])}$, there exists ℓ such that $i < \ell < j$ and $T(\ell) \notin W_{h(T[i])}$. Then we have $\forall n \geq j' : T(\ell) \in W_{h'(T[n])}$ as $T(\ell) \in W_{h'(T[j'])}$.

Obviously h' is conservative as it only outputs (delayed) hypotheses of h (and maybe skip some) and h is conservative. □

In the following we consider three new learning restrictions. The learning restriction $\mathbf{Caut_{Fin}}$ means that the learner never returns a hypothesis for a finite set that is a proper subset of a previous hypothesis. $\mathbf{Caut}_\infty$ is the same restriction for infinite hypotheses. With $\mathbf{Caut_{Tar}}$ the learner is not allowed to ever output a hypothesis that is a proper superset of the target language that is learned.

Definition 11.

$$\begin{aligned}
\mathbf{Caut_{Fin}}(p,T) &\Leftrightarrow [\forall i < j : W_{p(j)} \subset W_{p(i)} \Rightarrow W_{p(j)} \text{ is infinite}] \\
\mathbf{Caut}_\infty(p,T) &\Leftrightarrow [\forall i < j : W_{p(j)} \subset W_{p(i)} \Rightarrow W_{p(j)} \text{ is finite}] \\
\mathbf{Caut_{Tar}}(p,T) &\Leftrightarrow [\forall i : \neg(\text{content}(T) \subset W_{p(i)})]
\end{aligned}$$

The proof of the following theorem is essentially the same as given in [OSW86] to show that cautious learning is a proper restriction of **TxtGEx**-learning, we now extend it to strongly decisive learning. Note that a different extension was given in [BCM+08] (with an elegant proof exploiting the undecidability of the halting problem), pertaining to *behaviorally correct* learning. The proof in [BCM+08] as well as our proof would also carry over to the combination of these two extensions.

Theorem 12. There is a class of languages that is **TxtGSDecMonEx**-learnable, but not **TxtGCautEx**-learnable.

The following theorem contradicts a theorem given as an exercise in [OSW86] (Exercise 4.5.4B).

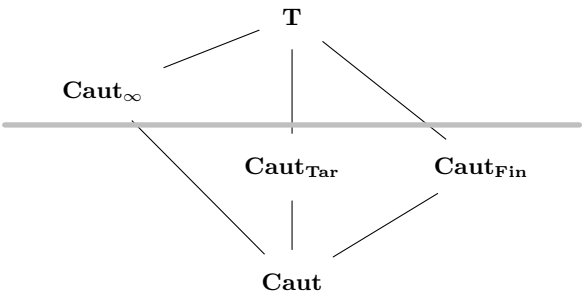


Fig. 5. Relation of different variants of cautious learning. A black line indicates inclusion (bottom to top); all and only the black lines meeting the gray line are proper inclusions.

Theorem 13. For $\delta \in \{\mathbf{Caut}, \mathbf{Caut_{Tar}}, \mathbf{Caut_{Fin}}\}$ we have

$$[\mathbf{TxtG}\delta\mathbf{Ex}] = [\mathbf{TxtGConvEx}].$$

From the definitions of the learning criteria we have [**TxtGConvEx**] $\subseteq$ [**TxtGWMonEx**]. Using Theorem 13 and the equivalence of weakly monotone and conservative learning (using **G**) [KS95, JS98], we get the following.

Corollary 14. We have

$$[\mathbf{TxtGConvEx}] = [\mathbf{TxtGWMonEx}] = [\mathbf{TxtGCautEx}].$$

Using Corollary 14 and Theorem 10 we get that weakly monotone **TxtGEx**-learning is included in strongly decisive **TxtGEx**-learning. Theorem 12 shows that this inclusion is proper.

Corollary 15. We have

$$[\mathbf{TxtGWMonEx}] \subset [\mathbf{TxtGSDecEx}].$$

The next theorem is the last theorem of this section and shows that forbidding to go down to strict *infinite* subsets of previously conjectures sets is no restriction.

Theorem 16. We have

$$[\mathbf{TxtGCaut_{\infty}Ex}] = [\mathbf{TxtGEx}].$$

Proof. Obviously we have [**TxtGCaut$_\infty$Ex**] $\subseteq$ [**TxtGEx**]. Thus, we have to show that [**TxtGEx**] $\subseteq$ [**TxtGCaut$_\infty$Ex**]. Let $\mathcal{L}$ be a set of languages and h be a learner such that h **TxtGEx**-learns $\mathcal{L}$ and h is strongly locking on $\mathcal{L}$ (see Corollary 7). We define, for all σ and t, the set M_σ^t such that

$$M_\sigma^t = \{\tau \mid \tau \in \mathbb{S}\mathrm{eq}(W_{h(\sigma)}^t \cup \mathrm{content}(\sigma)) \ \wedge \ |\tau \diamond \sigma| \leq t\}.$$

Using the S-m-n Theorem we get a function $p \in \mathcal{R}$ such that

$$\forall \sigma : W_{p(\sigma)} = \text{content}(\sigma) \bigcup_{t \in \mathbb{N}} \begin{cases} W^t_{h(\sigma)}, & \text{if } \forall \rho \in M^t_\sigma : h(\sigma \diamond \rho) = h(\sigma); \\ \emptyset, & \text{otherwise.} \end{cases}$$

We define a learner h' as

$$\forall \sigma : h'(\sigma) = \begin{cases} p(\sigma), & \text{if } h(\sigma) \neq h(\sigma^-); \\ h'(\sigma^-), & \text{otherwise.} \end{cases}$$

We will show now that the learner h' $\mathbf{TxtGCaut}_\infty\mathbf{Ex}$-learns $\mathcal{L}$. Let an $L \in \mathcal{L}$ and a text T for L be given. As h is strongly locking there is n_0 such that for all $\tau \in \mathbb{S}\text{eq}(L)$, $h(T[n_0] \diamond \tau) = h(T[n_0])$ and $W_{h(T[n_0])} = L$. Thus we have, for all $n \geq n_0$, $h'(T[n]) = h'(T[n_0])$ and $W_{h'(T[n_0])} = W_{p(T[n_0])} = W_{h(T[n_0])} = L$. To show that the learning restriction $\mathbf{Caut}_\infty$ holds, we assume that there are $i < j$ such that $W_{h'(T[j])} \subset W_{h'(T[i])}$ and $W_{h'(T[j])}$ is infinite. W.l.o.g. j is the first time that h' returns the hypothesis $W_{h'(T[j])}$. Let τ be such that $T[i] \diamond \tau = T[j]$. From the definition of the function p we get that $\text{content}(T[j]) \subseteq W_{h'(T[j])} \subseteq W_{h'(T[i])}$. Thus, $\text{content}(\tau) \subseteq W_{h'(T[i])} = W_{p(T[i])}$ and therefore $W_{p(T[i])}$ is finite, a contradiction to the assumption that $W_{h'(T[j])}$ is infinite. □

The following theorem can be shown with a priority argument. The detailed proof is about eight pages long, following some ideas given in [BCM+08] for proving the second inequality, but adapted to the priority argument.

Theorem 17. We have

$$[\mathbf{TxtGSDecEx}] \subset [\mathbf{TxtGDecEx}] \subset [\mathbf{TxtGEx}].$$

5 Set-Driven Learning

In this section we give theorems regarding set-driven learning. For this we build on the result that set-driven learning can always be done conservatively [KS95].

First we show that any conservative set-driven learner can be assumed to be cautious and syntactically decisive, an important technical lemma.

Lemma 18. We have

$$[\mathbf{TxtSdEx}] = [\mathbf{TxtSdConvSynDecEx}].$$

In other words, every set-driven learner can be assumed syntactically decisive.

The following Theorem is the main result of this section, showing that set-driven learning can be done not just conservatively, but also strongly decisively and cautiously *at the same time*.

Theorem 19. We have

$$[\mathbf{TxtSdEx}] = [\mathbf{TxtSdConvSDecCautEx}].$$

6 Monotone Learning

In this section we show the hierarchies regarding monotone and strongly monotone learning, simultaneously for the settings of **G** and **Sd** in Theorems 20 and 21. With Theorems 22 and 23 we establish that monotone learnabilty implies strongly decisive learnability.

Theorem 20. There is a language $\mathcal{L}$ that is **TxtSdMonWMonEx**-learnable but not **TxtGSMonEx**-learnable, i.e.

$$[\mathbf{TxtSdMonWMonEx}] \backslash [\mathbf{TxtGSMonEx}] \neq \emptyset.$$

Theorem 21. There is $\mathcal{L}$ such that $\mathcal{L}$ is **TxtSdWMonEx**-learnable but not **TxtGMonEx**-learnable.

The following theorem is an extension of a theorem from [BCM$^+$08], where the theorem has been shown for decisive learning instead of strongly decisive learning.

Theorem 22. Let $\mathbb{N} \in \mathcal{L}$ and $\mathcal{L}$ be **TxtGEx**-learnable. Then, we have $\mathcal{L}$ is **TxtGSDecEx**-learnable.

Theorem 23. We have that any monotone **TxtGEx**-learnable class of languages is strongly decisive learnable, while the converse does not hold, i.e.

$$[\mathbf{TxtGMonEx}] \subset [\mathbf{TxtGSDecEx}].$$

Proof. Let $h \in \mathcal{R}$ be a learner and $\mathcal{L} = \mathbf{TxtGMonEx}(h)$. We distinguish the following two cases. We call $\mathcal{L}$ *dense* iff it contains a superset of every finite set.

Case 1: $\mathcal{L}$ is dense. We will show now that h **TxtGSMonEx**-learns the class $\mathcal{L}$. Let $L \in \mathcal{L}$ and T be a text for L. Suppose there are i and j with $i < j$ such that $W_{h(T[i])} \not\subseteq W_{h(T[j])}$. Thus, we have $W_{h(T[i])} \backslash W_{h(T[j])} \neq \emptyset$. Let $x \in W_{h(T[i])} \backslash W_{h(T[j])}$. As $\mathcal{L}$ is dense there is a language $L' \in \mathcal{L}$ such that $\mathrm{content}(T[j]) \cup \{x\} \in L'$. Let T' be a text for L' and T'' be such that $T'' = T[j] \diamond T'$. Obviously, T'' is a text for L'. We have that $x \in W_{h(T''[i])}$ but $x \notin W_{h(T''[j])}$ which is a contradiction as h is monotone. Thus, h **TxtGSMonEx**-learns $\mathcal{L}$, which implies that h **TxtGWMonEx**-learns $\mathcal{L}$. Using Corollary 15 we get that $\mathcal{L}$ is **TxtGSDecEx**-learnable.

Case 2: $\mathcal{L}$ is not dense. Thus, $\mathcal{L}' = \mathcal{L} \cup \mathbb{N}$ is **TxtGEx**-learnable. Using Theorem 22 $\mathcal{L}'$ is **TxtGSDecEx**-learnable and therefore so is $\mathcal{L}$.

Note that $[\mathbf{TxtGSDecEx}] \subseteq [\mathbf{TxtGMonEx}]$ does not hold as in *Case 1* with Corollary 15 a proper subset relation is used.

□

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)

[BCM+08] Baliga, G., Case, J., Merkle, W., Stephan, F., Wiehagen, W.: When unlearning helps. Information and Computation 206, 694–709 (2008)

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

[CM11] Case, J., Moelius, S.: Optimal language learning from positive data. Information and Computation 209, 1293–1311 (2011)

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

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

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

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

[JS98] Jain, S., Sharma, A.: Generalization and specialization strategies for learning r.e. languages. Annals of Mathematics and Artificial Intelligence 23, 1–26 (1998)

[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: Proc. of STACS (Symposium on Theoretical Aspects of Computer Science), pp. 494–505 (2014)

[KS95] Kinber, E., Stephan, F.: Language learning from texts: Mind changes, limited memory and monotonicity. Information and Computation 123, 224–241 (1995)

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

[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)

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

[SR84] Schäfer-Richter, G.: Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien. PhD thesis, RWTH Aachen (1984)

[WC80] Wexler, K., Culicover, P.: Formal Principles of Language Acquisition. MIT Press, Cambridge (1980)

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