Consistent and Conservative Iterative Learning

Sanjay Jain [a,1], Steffen Lange [b], and Sandra Zilles [c,*]

[a] *School of Computing, National University of Singapore, Singapore 117543*

[b] *Fachbereich Informatik, Hochschule Darmstadt, Haardtring 100, 64295 Darmstadt, Germany*

[c] *DFKI GmbH, Trippstadter Straße 122, 67663 Kaiserslautern, Germany*

Abstract

The present study aims at insights into the nature of incremental learning in the context of Gold's model of identification in the limit. With a focus on natural requirements such as consistency and conservativeness, incremental learning is analysed both for learning from positive examples and for learning from positive and negative examples. The results obtained illustrate in which way different consistency and conservativeness demands can affect the capabilities of incremental learners. These results may serve as a first step towards characterising the structure of typical classes learnable incrementally and thus towards elaborating uniform incremental learning methods.

1 Introduction

Considering data mining tasks, where specific knowledge has to be induced from a huge amount of more or less unstructured data, several approaches have been studied empirically in machine learning and formally in the field of learning theory. These approaches differ in terms of the form of interaction between the learning machine and its environment. For instance, scenarios have been analysed, where the learner receives instances of some target concept to be identified, see Gold [6] and Valiant [13] for two different approaches, or where

* corresponding author
Email addresses: sanjay@comp.nus.edu.sg (Sanjay Jain), s.lange@fbi.h-da.de (Steffen Lange), sandra.zilles@dfki.de (Sandra Zilles).

[1] Sanjay Jain was supported in part by NUS grant number R252-000-127-112 and R252-000-212-112.

the learner may pose queries concerning the target concept, see Angluin [2]. For learning from examples, one critical aspect is the limitation of a learning machine in terms of its memory capacity. In particular, if huge amounts of data have to be processed, it is conceivable that this capacity is too low to memorise all relevant information during the whole learning process. This has motivated the analysis of so-called *incremental learning*, as proposed by Wiehagen [14] and studied, e. g., by Case et al. [4], Gennari et al. [5], Kinber and Stephan [8], Lange and Grieser [9], Lange and Zeugmann [10], where in each step of the learning process, the learner has access only to a limited number of examples. Thus, in each step, its hypothesis can be built upon these examples and its former hypothesis, only. Other examples seen before have to be 'forgotten'.

It has been analysed how such constraints affect the capabilities of learning machines, thus revealing models in which certain classes of target concepts are learnable, but not learnable in an incremental manner. However, some quite natural constraints for successful learning have mainly been neglected in the corresponding studies. These constraints are (a) the requirement for *consistent* learning, i. e., the demand that none of the intermediate hypotheses a learner explicates should contradict the data processed so far, and (b) the requirement for *conservative* learning, i. e., the demand that each intermediate hypothesis should be maintained as long as it is consistent with the data seen.

The fact that there is no comprehensive analysis of how these demands affect the capabilities of incremental learners can be traced back to a lack of knowledge about the nature of incremental learning. In particular, there is no formal basis explaining typical or uniform ways for solving learning tasks in an incremental way. In terms of learning theory, incremental learning is one of the very few models, for which no characterisation of the typical structure of learnable classes is known. For other models of learning from examples, characterisations and uniform learning methods have often been the outcome of analysing the impact of consistency or conservativeness, see, e. g., Zeugmann and Lange [16]. Thus, also in the context of incremental learning, it is conceivable that studying these natural requirements may yield insights into typical learning methods. In other words, analysing consistency and conservativeness may be the key for a better understanding of the nature of incremental learning and may thus, in the long term, provide characterisations of learnable classes and uniform incremental learning methods.

The present study aims at insights into the nature of incremental learning in the context of Gold's model of learning in the limit from examples, see Gold [6]. For that purpose, we analyse Wiehagen's version of incremental learning, namely *iterative learning* [14] with a focus on consistent and conservative learners. In Gold's approach, learning is considered as an infinite process, where in each step the learner is presented an example e_n for the target concept and is supposed to return an intermediate hypothesis. In the limit, the hypotheses must stabilise on a correct representation of the target concept. Here, in step $n+1$ of the learning process, the learner has access to all examples $e_0, \ldots, e_n$ provided up to step n plus the current example e_{n+1}. In contrast, an iterative learner has no capacities for memorising any examples seen so far, i. e., its hypothesis h_{n+1} in step $n+1$ is built only upon the example e_{n+1} and its previous hypothesis h_n.

The present paper addresses consistency and conservativeness in the context of iterative learning. Here several possible ways to formalise the demands for consistency and conservativeness become apparent. Assume an iterative learner has processed the examples $e_0, \ldots, e_{n+1}$ for some target concept and returns some hypothesis h_{n+1} in step $n+1$. From a global perspective, one would define h_{n+1} consistent, if it agrees with the examples $e_0, \ldots, e_{n+1}$. But since the learner has not memorised $e_0, \ldots, e_n$, it might be considered natural to just demand that h_{n+1} agrees with the current example e_{n+1}. This is justified from a rather local perspective. Similarly, when defining conservativeness from a global point of view, one might demand that $h_{n+1} = h_n$ in case h_n does not contradict any of the examples $e_0, \ldots, e_{n+1}$, whereas a local variant of conservativeness would mean to require that $h_{n+1} = h_n$ in case h_n does not contradict the current example e_{n+1}. Note that local consistency is a weaker requirement than global consistency, whereas local conservativeness is stronger than global conservativeness.

In the present paper, we restrict our focus on recursive languages as target concepts. [2] In particular, the target classes are required to be indexable, i. e., there exist algorithms deciding the membership problem uniformly for all possible target languages. This restriction is motivated by the fact that many classes of target concepts relevant for typical learning tasks are indexable.

The paper is structured as follows. In Section 2, we provide the definitions and notations necessary for our formal analysis. Then Section 3 is concerned with a case study of iterative learning of regular erasing pattern languages – a quite natural and simple to define indexable class which has shown to be suitable for representing target concepts in many application scenarios. This case study shows how consistency and conservativeness may affect the learnability of such pattern languages in case quite natural hypothesis spaces are chosen for learning. Section 4 focuses on consistency in iterative learning. It has turned out, that iterative learners can be normalised to work in a locally consistent way, whereas global consistency is a constraint reducing the capabilities of iterative learners. Both results hold for learning from positive examples as well as for learning from both positive and negative examples. Section 5 then is concerned with conservativeness. Here we show that, in the scenario of

[2] See Angluin [1] and Zeugmann and Lange [16] for an overview on early results.

learning from only positive examples, the effects of global conservativeness demands and local conservativeness demands are equal, as far as the capabilities of iterative learners are concerned. In contrast to that there are classes which can be learned iteratively from positive and negative examples by a globally conservative learner, but not in a locally conservative manner. Concerning the effect of weak conservativeness demands (i. e., of global conservativeness), we can show that they strictly reduce the capabilities of iterative learners which are given both positive and negative examples as information. However, the corresponding comparison in the case of learning from only positive examples is still open. In our point of view, not only the mere results presented here, but in particular the proof constructions and separating classes give an impression of characteristic methods of iterative learning and characteristic properties of iteratively learnable classes, even though we cannot provide a formal characterisation yet. Section 6 contains a concluding discussion.

2 Preliminaries

Let Σ be a fixed finite alphabet, Σ^* the set of all finite strings over Σ, and Σ^+ its subset excluding the empty string. $|w|$ denotes the length of a string w. Any subset of Σ^* is called a *language*. For any language L, $co(L) = \Sigma^* \setminus L$. $\mathbb{N}$ is the set of all natural numbers. If L is a non-empty language, then any infinite sequence $t = (w_j)_{j \in \mathbb{N}}$ with $\{w_j \mid j \in \mathbb{N}\} = L$ is called a *text* for L. Moreover, any infinite sequence $i = ((w_j, b_j))_{j \in \mathbb{N}}$ over $\Sigma^* \times \{+, -\}$ such that $\{w_j \mid j \in \mathbb{N}\} = \Sigma^*$, $\{w_j \mid j \in \mathbb{N},\ b_j = +\} = L$, and $\{w_j \mid j \in \mathbb{N},\ b_j = -\} = co(L)$ is referred to as an *informant* for L. Now assume some fixed $t = (w_j)_{j \in \mathbb{N}}$ and $i = ((w_j, b_j))_{j \in \mathbb{N}}$, where $w_j \in \Sigma^*$ and $b_j \in \{+, -\}$ for all $j \in \mathbb{N}$. Then, for any $n \in \mathbb{N}$, $t[n]$ and $i[n]$ denote the initial segment of t and i of length $n+1$, while $t(n) = w_n$ and $i(n) = (w_n, b_n)$. Furthermore, $content(t[n]) = \{w_j \mid j \leq n\}$. Finally, $content(i[n])$, $content^+(i[n])$, and $content^-(i[n])$ denote the sets $\{(w_j, b_j) \mid j \leq n\}$, $\{w_j \mid j \leq n,\ b_j = +\}$, and $\{w_j \mid j \leq n,\ b_j = -\}$, respectively.

A family $(L_j)_{j \in \mathbb{N}}$ of languages is called an *indexing* for a class $\mathcal{C}$ of recursive languages, if $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$ and there is a recursive function f such that $L_j = \{w \in \Sigma^* \mid f(j, w) = 1\}$ for all $j \in \mathbb{N}$. $\mathcal{C}$ is called an *indexable class* (of recursive languages), if $\mathcal{C}$ possesses an indexing.

In our proofs, we will use a fixed Gödel numbering $(\varphi_j)_{j \in \mathbb{N}}$ of all (and only all) partial recursive functions over $\mathbb{N}$ as well as an associated complexity measure $(\Phi_j)_{j \in \mathbb{N}}$, see Blum [3]. Then, for $k, x \in \mathbb{N}$, φ_k is the partial recursive function computed by program k and we write $\varphi_k(x)\downarrow$ ($\varphi_k(x)\uparrow$), if $\varphi_k(x)$ is defined (undefined).

Note that the models of learning from text considered below are concerned with learning a target language L from positive examples presented in the form of a text for L. For this reason we assume from now on that all languages considered as target objects for learning are non-empty.

2.1 Learning from text

Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ any indexing of some $\mathcal{C}' \supseteq \mathcal{C}$ (called *hypothesis space*), and $L \in \mathcal{C}$. An *inductive inference machine* (*IIM* for short) M is an algorithmic device that reads longer and longer initial segments σ of a text and outputs numbers $M(\sigma)$ as its hypotheses. An IIM M returning some j is construed to hypothesize the language L_j. The following definition of learning from positive data is based on Gold [6].

Definition 1 (Gold [6]) *Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ an indexing of some $\mathcal{C}' \supseteq \mathcal{C}$, and $L \in \mathcal{C}$. Let t be a text for L, M an inductive inference machine.*

(1) M learns L from t with respect to $\mathcal{H}$, if
  (a) the sequence $(M(t[n]))_{n\in\mathbb{N}}$ stabilises on a number j (i. e., past some point M always outputs the hypothesis j *) and*
  (b) this number j fulfils $L_j = L$.
(2) M learns L in the limit from text with respect to $\mathcal{H}$, if M learns L from every text for L with respect to $\mathcal{H}$.
(3) M learns $\mathcal{C}$ in the limit from text with respect to $\mathcal{H}$, if M learns every language in $\mathcal{C}$ from text with respect to $\mathcal{H}$.

Correspondingly, a class $\mathcal{C}$ is said to be *learnable in the limit from text*, if there is some hypothesis space $\mathcal{H}$, i. e., an indexing, and some inductive inference machine M, such that M learns $\mathcal{C}$ in the limit from text with respect to $\mathcal{H}$. *LimTxt* denotes the collection of all classes learnable in the limit from text.

Having a closer look at learning algorithms from an application-oriented point of view, it is rather unlikely that the general case of inductive inference machines—as specified in Gold's model—will turn out satisfactory. This might have several reasons, because the model does not include any constraints concerning

- consistency,
- conservativeness,
- memory bounds.

Consistency is the quite natural property that a learner only generates hypotheses which are consistent with the data seen so far, i. e., in the case of learning in the limit from text, which represent languages containing all the examples provided as input.

Definition 2 (Gold [6]) *Let* $\mathcal{C}$ *be an indexable class,* $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ *a hypothesis space, and* M *an IIM.* M *is consistent for* $\mathcal{C}$ *iff* $content(t[n]) \subseteq L_{M(t[n])}$ *for every text segment* $t[n]$ *for some* $L \in \mathcal{C}$*.*

Cons Txt denotes the collection of all indexable classes $\mathcal{C}'$ for which there is a hypothesis space $\mathcal{H}'$ and an IIM which is consistent for $\mathcal{C}'$ and learns $\mathcal{C}'$ in the limit from text with respect to $\mathcal{H}'$.

As it turns out, this demand does not really restrict the capabilities of IIMs, i. e., IIMs can be normalised to work in a consistent manner.

Proposition 3 (Angluin [1]) *Cons Txt* $=$ *Lim Txt.*

With conservativeness, it is a little different. Conservative IIMs do not change their hypotheses, if they are consistent with all data provided so far. This demand is very important when analysing the possible reasons for learners to change their hypotheses during the learning process.

Definition 4 (Angluin [1], Zeugmann and Lange [16]) *Let* $\mathcal{C}$ *be an indexable class,* $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ *be a hypothesis space, and* M *an IIM.* M *is conservative for* $\mathcal{C}$ *iff, for every text segment* $t[n+1]$ *for some* $L \in \mathcal{C}$*,* $M(t[n+1]) \neq M(t[n])$ *implies* $content(t[n+1]) \not\subseteq L_{M(t[n])}$*.*

Correspondingly, *Conv Txt* denotes the collection of all indexable classes $\mathcal{C}'$ for which there is a hypothesis space $\mathcal{H}'$ and an IIM which is conservative for $\mathcal{C}'$ and learns $\mathcal{C}'$ from text with respect to $\mathcal{H}'$.

A phenomenon which might seem astonishing at first glance is that conservativeness really restricts the capabilities of Gold-style inductive inference machines. The reason is that there are classes in *Lim Txt*, for which a successful IIM sometimes has to return hypotheses which overgeneralise the target language.

Proposition 5 (Zeugmann and Lange [16]) *Conv Txt* $\subset$ *Lim Txt.*

Note that originally Angluin [1] has proven a weaker result, showing that *Lim Txt*-learners for an indexable class $\mathcal{C}$ can in general not be made conservative, if it is required that all the intermediate hypotheses they return represent languages in $\mathcal{C}$—that is to say if they work in a so-called class-preserving manner.

Finally, let us consider a third important aspect not addressed in Definition 1, namely bounds on the example memory. Note that an IIM, when learning in the limit, processes gradually growing finite sequences of examples, where it is assumed that the amount of data the IIM can store and process in each step is not bounded a priori. This rather unrealistic assumption is suspended in the approach of incremental learning, particularly in iterative learning.

An *iterative inductive inference machines* is only allowed to use its previous hypothesis and the current string in a text for computing its current hypothesis. More formally, an *iterative* IIM M is an algorithmic device that maps elements from $\mathbb{N} \cup \{init\} \times \Sigma^*$ into $\mathbb{N}$, where *init* denotes a fixed initial 'hypothesis' (not a natural number) which the IIM may never output. Let $t = (w_n)_{n \in \mathbb{N}}$ be any text for some language $L \subseteq \Sigma^*$. Then we denote by $(M[init, t[n]])_{n \in \mathbb{N}}$ the sequence of hypotheses generated by M when processing t, i. e., $M[init, w_0] = M(init, w_0)$ and, for all $n \in \mathbb{N}$, $M[init, t[n+1]] = M(M[init, t[n]], w_{n+1})$.

Definition 6 (Wiehagen [14]) *Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space, and $L \in \mathcal{C}$. Let M be an iterative IIM.*

(1) M learns L from text with respect to $\mathcal{H}$ iff, for any text $t = (w_n)_{n \in \mathbb{N}}$ for L, the sequence $(M[init, t[n]])_{n \in \mathbb{N}}$ stabilises on a number j with $L_j = L$.
(2) M learns $\mathcal{C}$ from text with respect to $\mathcal{H}$, if it learns every $L' \in \mathcal{C}$ from text with respect to $\mathcal{H}$.

Finally, *ItTxt* denotes the collection of all indexable classes $\mathcal{C}'$ for which there is a hypothesis space $\mathcal{H}'$ and an iterative IIM learning $\mathcal{C}'$ from text with respect to $\mathcal{H}'$.

Obviously, each class learnable iteratively from text is learnable in the limit from text—having a closer look: even conservatively. However, there are classes in *ConvTxt*, which cannot be identified iteratively from text.

Proposition 7 (Lange and Zeugmann [10]) *$ItTxt \subset ConvTxt$.*

The model of iterative learning is one instantiation of the idea of incremental learning and is the main focus of the formal study below, in particular in combination with consistency and conservativeness demands.

In the definition of consistent learning above, a hypothesis of a learner is said to be consistent, if it reflects the data it was built upon correctly. Since an iterative IIM M, when processing some text t, is only allowed to use its previous hypothesis, say $L_{j'}$, and the current string v in t for computing its current hypothesis L_j, it is quite natural to distinguish two variants of consistent learning. In the first case, it is demanded that L_j contains all elements of t seen so far, while, in the second case, it is only required that L_j contains the string v.

Definition 8 *Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space, and*

M an iterative IIM. M is globally (locally) consistent for $\mathcal{C}$ iff $content(t[n]) \subseteq L_{M[init,t[n]]}$ ($t(n) \in L_{M[init,t[n]]}$) for every text segment $t[n]$ for some $L \in \mathcal{C}$.

Moreover, *ItGConsTxt* (*ItLConsTxt*) denotes the collection of all indexable classes $\mathcal{C}'$ for which there is a hypothesis space $\mathcal{H}'$ and an iterative IIM which is globally (locally) consistent for $\mathcal{C}'$ and learns $\mathcal{C}'$ from text with respect to $\mathcal{H}'$.

Finally we consider conservative iterative IIMs. Informally speaking, a conservative learner maintains its current hypothesis as long as the latter does not contradict any data seen. Hence, whenever a conservative IIM changes its recent hypothesis, this must be justified by data having occurred which prove an inconsistency of its recent hypothesis. Similarly to the case of consistent iterative learning, it is quite natural to distinguish two variants of conservativeness in the context of iterative learning.

Definition 9 *Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ be a hypothesis space, and M be an iterative IIM. M is globally (locally) conservative for $\mathcal{C}$ iff, for every text segment $t[n+1]$ for some $L \in \mathcal{C}$, $M[init, t[n+1]] \neq M[init, t[n]]$ implies $content(t[n+1]) \not\subseteq L_{M[init,t[n]]}$ (implies $t(n+1) \notin L_{M[init,t[n]]}$).*

In parallel to the notions defined above, *ItGConvTxt* (*ItLConvTxt*) denotes the collection of all indexable classes $\mathcal{C}'$ for which there is a hypothesis space $\mathcal{H}'$ and an iterative IIM which is globally (locally) conservative for $\mathcal{C}'$ and learns $\mathcal{C}'$ from text with respect to $\mathcal{H}'$.

Note that we allow a mind change from *init* after the first input data is received.

2.2 Learning from informant

For all variants of *ItTxt* considered so far we define corresponding models capturing the case of learning from informant. Now an iterative IIM M maps $\mathbb{N} \times (\Sigma^* \times \{+, -\})$ into $\mathbb{N}$. Let $i = (w_n, b_n)_{n\in\mathbb{N}}$ be any informant for some language L, and let *init* be a fixed initial hypothesis. Then $(M[init, i[n]])_{n\in\mathbb{N}}$ is the sequence of hypotheses returned by M when processing the informant i, i.e., $M[init, (w_0, b_0)] = M(init, (w_0, b_0))$ and, for all $n \in \mathbb{N}$, $M[init, i[n+1]] = M(M[init, i[n]], (w_{n+1}, b_{n+1}))$.

Definition 10 (Wiehagen [14]) *Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ a hypothesis space, and $L \in \mathcal{C}$. An iterative IIM M learns L from informant with respect to $\mathcal{H}$, iff for every informant i for L, the sequence $(M[init, i[n]])_{n\in\mathbb{N}}$ stabilises on a number j with $L_j = L$. Moreover, M learns $\mathcal{C}$ from informant with respect to $\mathcal{H}$, if M learns every $L' \in \mathcal{C}$ from informant with respect to $\mathcal{H}$.*

The notion *ItInf* is defined similarly to the text case. Now also the consistency and conservativeness demands can be formalised. For instance, for consistency, let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ a hypothesis space, and M an iterative IIM. M is globally (locally) consistent for $\mathcal{C}$ iff $\mathit{content}^+(i[n]) \subseteq L_{M[\mathit{init},i[n]]}$ and $\mathit{content}^-(i[n]) \subseteq co(L_{M[\mathit{init},i[n]]})$ ($b = +$ for $w \in L_{M[\mathit{init},i[n]]}$ and $b = -$ for $w \notin L_{M[\mathit{init},i[n]]}$) for every informant segment $i[n]$ for some $L \in \mathcal{C}$, where $i(n) = (w, b)$. Finally, the definitions of *ItGConsInf*, *ItLConsInf*, *ItGConvInf*, *ItLConvInf* can be adapted from the text case to the informant case.

3 A case study: The regular erasing pattern languages

Let Σ be any fixed finite alphabet. Let $X = \{x_1, x_2, \dots\}$ be an infinite set of variables, disjoint with Σ. A *regular pattern* α is a string from $(\Sigma \cup X)^+$ which contains every variable at most once. Let α be a regular pattern. Then $L_\varepsilon(\alpha)$, the *regular erasing pattern language* generated by α, contains all strings in Σ^* that can be obtained by replacing the variables in α by strings from Σ^*, see, e. g., Shinohara [12]. Note that $L_\varepsilon(\alpha)$ constitutes a regular language. Subsequently, let $\mathcal{C}_{\mathrm{rp}}$ denote the collection of all regular erasing pattern languages.

Our first result, stating that the regular erasing pattern languages can be be learned by an iterative IIM which is both globally consistent and locally conservative, can be achieved by adapting a standard idea, see, e. g., Case et al. [4]. For its proof the following folklore lemma is required.

Lemma 11 *Let $(D_j)_{j\in\mathbb{N}}$ be the canonical enumeration of all finite subsets of $\mathbb{N}$ and $(\alpha_j)_{j\in\mathbb{N}}$ a recursively enumerable family of regular patterns such that $(L_\varepsilon(\alpha_j))_{j\in\mathbb{N}}$ is an effective, repetition-free indexing of $\mathcal{C}_{\mathrm{rp}}$. There is an algorithm A which, given any string $w \in \Sigma^+$ as input, outputs an index j such that $D_j = \{z \in \mathbb{N} \mid w \in L_\varepsilon(\alpha_z)\}$.*

Theorem 12 *There is a learner witnessing both $\mathcal{C}_{\mathrm{rp}} \in \mathit{ItGConsTxt}$ and $\mathcal{C}_{\mathrm{rp}} \in \mathit{ItLConvTxt}$.*

Sketch of the proof. Let $(D_j)_{j\in\mathbb{N}}$ and $(\alpha_j)_{j\in\mathbb{N}}$ be chosen as in Lemma 11. Moreover let $L'_j = \bigcap_{z\in D_j} L_\varepsilon(\alpha_z)$. Hence $(L'_j)_{j\in\mathbb{N}}$ is an indexing comprising the class $\mathcal{C}_{\mathrm{rp}}$. The proof is essentially based on Lemma 11, using the algorithm A claimed there.

A learner M witnessing $\mathcal{C}_{\mathrm{rp}} \in \mathit{ItGConsTxt}$ and $\mathcal{C}_{\mathrm{rp}} \in \mathit{ItLConvTxt}$ with respect to $(L')_{j\in\mathbb{N}}$ may simply work as follows:

Initially, if the first string w appears, M starts its subroutine A according to Lemma 11, determines $j = A(w)$, and guesses the language L'_j, i.e., $M(\mathit{init}, w) = j$. Next M, when receiving a new string v, refines its recent hypothesis, say j', as follows. M determines the canonical index j of the set $\{z \mid z \in D_{j'}, v \in L_\varepsilon(\alpha_z)\} \subseteq D_{j'}$ and guesses the language L'_j, i.e., $M(j', v) = j$.

It is not hard to see that M learns as required. □

Although the iterative learner M used in this proof is locally conservative and globally consistent, M has the disadvantage of guessing languages not contained in the class of all regular erasing pattern languages. At first glance, it might seem that this weakness can easily be compensated, since the final guess returned by M is always a regular erasing pattern language and, moreover, one can effectively determine whether or not the recent guess of M equals a regular erasing pattern language. Surprisingly, even under this quite 'perfect' circumstances, it is impossible to replace M by an iterative, locally conservative, and globally consistent learner for $\mathcal{C}_{\mathrm{rp}}$ that hypothesizes languages in $\mathcal{C}_{\mathrm{rp}}$, exclusively.

Theorem 13 *Let* $\mathit{card}(\Sigma) \geq 2$. *Let* $(L_j)_{j\in\mathbb{N}}$ *be any indexing of* $\mathcal{C}_{\mathrm{rp}}$. *Then there is no learner* M *witnessing both* $\mathcal{C}_{\mathrm{rp}} \in \mathit{ItGConsTxt}$ *and* $\mathcal{C}_{\mathrm{rp}} \in \mathit{ItLConvTxt}$ *with respect to* $(L_j)_{j\in\mathbb{N}}$.

Proof. Let $\{a, b\} \subseteq \Sigma$. Assume to the contrary that there is an iterative learner M which learns $\mathcal{C}_{\mathrm{rp}}$ locally conservatively and globally consistently, hypothesising only regular erasing pattern languages. Consider M for any text of some $L \in \mathcal{C}_{\mathrm{rp}}$ with the initial segment $\sigma = (aba, aab)$. Since M must avoid overgeneralisations, only minimally general hypotheses are returned. There are only two possible semantically different hypotheses which are globally consistent with σ and minimally general with that property, namely x_1abx_2 and ax_1ax_2. Distinguish two cases:

Case (a). $L_{M[\mathit{init},\sigma]} = L_\varepsilon(x_1abx_2)$.

Consider M when processing the two sequences $\sigma_1 = (aba, aab, ab, aa)$ and $\sigma_2 = (aba, aab, aa)$. Since $ab \in L_\varepsilon(x_1abx_2)$ and M is locally conservative for $\mathcal{C}_{\mathrm{rp}}$, we obtain $M[\mathit{init}, (aba, aab, ab)] = M[\mathit{init}, (aba, aab)] = M[\mathit{init}, \sigma]$. For reasons of global consistency, $L_{M[\mathit{init},\sigma_1]} = L_\varepsilon(ax_1)$.

Now, since $M[\mathit{init}, (aba, aab, ab)] = M[\mathit{init}, \sigma]$, this yields $L_{M[\mathit{init},\sigma_2]} = L_\varepsilon(ax_1)$. However, σ_2 can be extended to a text for $L_\varepsilon(ax_1ax_2)$, on which M will fail to learn locally conservatively, since $M[\mathit{init}, \sigma_2]$ overgeneralises the target. This contradicts the assumptions on M.

Case (b). $L_{M[\mathit{init},\sigma]} = L_\varepsilon(ax_1ax_2)$.

Here a similar contradiction can be obtained for M processing the sequences $\sigma_1 = (aba, aab, aa, ab)$ and $\sigma_2 = (aba, aab, ab)$.

Both cases yield a contradiction and thus the theorem is verified. □

However, as Theorems 15 and 16 show, each of our natural requirements, in its stronger formulation, can be achieved separately, if an appropriate indexing of the regular erasing pattern languages is used as a hypothesis space. To prove this the following folklore lemma, which can be verified with standard methods, is needed.

Lemma 14 *Let $(D_j)_{j\in\mathbb{N}}$ be the canonical enumeration of all finite subsets of $\mathbb{N}$ and $(\alpha_j)_{j\in\mathbb{N}}$ a recursively enumerable family of regular patterns such that $(L_\varepsilon(\alpha_j))_{j\in\mathbb{N}}$ is an effective, repetition-free indexing of $\mathcal{C}_{\mathrm{rp}}$. There is an algorithm A' which, given any index j as input, outputs an index k with $L_\varepsilon(\alpha_k) = \bigcap_{z\in D_j} L_\varepsilon(\alpha_z)$, if such an index exists, and 'no', otherwise.*

Proof idea. Since every regular erasing pattern language is a regular language and both the inclusion problem as well as the equivalence problem for regular languages are decidable, such an algorithm A' exists. □

Theorem 15 *There is an indexing $(L^*_j)_{j\in\mathbb{N}}$ of $\mathcal{C}_{\mathrm{rp}}$ and a learner M witnessing $\mathcal{C}_{\mathrm{rp}} \in ItLConvTxt$ with respect to $(L^*_j)_{j\in\mathbb{N}}$.*

Proof. Let $(D_j)_{j\in\mathbb{N}}$ and $(\alpha_j)_{j\in\mathbb{N}}$ be chosen as in Lemma 14. Moreover let $L'_j = \bigcap_{z\in D_j} L_\varepsilon(\alpha_z)$ for all $j \in \mathbb{N}$. Hence $(L'_j)_{j\in\mathbb{N}}$ is an indexing comprising the class $\mathcal{C}_{\mathrm{rp}}$.

The required iterative learner uses the algorithm A' claimed in Lemma 14 and the iterative learner M from the demonstration of Theorem 12 as its subroutines. Let $(L^*_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ be an indexing of $\mathcal{C}_{\mathrm{rp}}$ with $L^*_{\langle k,j\rangle} = L_\varepsilon(\alpha_k)$ for all $k, j \in \mathbb{N}$. We define an iterative learner M' for $\mathcal{C}_{\mathrm{rp}}$ that uses the hypothesis space $(L^*_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$.

Initially, if the first string w appears, M' determines the canonical index k of the regular erasing pattern language $L_\varepsilon(w)$ as well as $j = M(init, w)$, and outputs the hypothesis $\langle k, j\rangle$, i. e., $M'(init, w) = \langle k, j\rangle$. Next M', when receiving a string v, refines its recent hypothesis, say $\langle k', j'\rangle$, as follows. First, if $v \in L^*_{\langle k',j'\rangle}$, M' repeats its recent hypothesis, i. e., $M'(\langle k', j'\rangle, v) = \langle k', j'\rangle$. (* Note that $j' = M(j', v)$, too. *) Second, if $v \notin L^*_{\langle k',j'\rangle}$, M' determines $j = M(j', v)$ and runs A' on input j. If A' returns some $k \in \mathbb{N}$, M' returns $\langle k, j\rangle$, i. e., $M'(\langle k', j'\rangle, v) = \langle k, j\rangle$. If A' returns 'no', M' determines the canonical index k of the regular erasing pattern language $L_\varepsilon(v)$ and returns $\langle k, j\rangle$, i. e., $M'(\langle k', j'\rangle, v) = \langle k, j\rangle$.

By definition, M' is an iterative and locally conservative learner. Let t be any text for any $L \in \mathcal{C}_{\text{rp}}$. Since M learns L, there is some n such that $M[init, t[n]] = j$ with $L'_j = L$. By definition, for $\langle k, j \rangle = M'[init, t[n]]$, we have $L_\varepsilon(\alpha_k) = L'_j$. Thus $L^*_{\langle k,j \rangle} = L_\varepsilon(\alpha_k)$. Since M' is a locally conservative learner, M' learns L, too. □

Theorem 16 *There is an indexing* $(L_j)_{j \in \mathbb{N}}$ *of* $\mathcal{C}_{\text{rp}}$ *and a learner* M *witnessing* $\mathcal{C}_{\text{rp}} \in ItGConsTxt$ *with respect to* $(L_j)_{j \in \mathbb{N}}$.

Proof. The proof proceeds similarly to that of Theorem 15. Hence, define $(D_j)_{j \in \mathbb{N}}$, $(\alpha_j)_{j \in \mathbb{N}}$, $(L'_j)_{j \in \mathbb{N}}$ analogously. Note that $(L'_j)_{j \in \mathbb{N}}$ is an indexing comprising the class $\mathcal{C}_{\text{rp}}$.

The proof is again based on Lemma 14, which says that there is an algorithm A' which, given any index j as input, outputs an index k with $L_\varepsilon(\alpha_k) = L'_j$, if such an index exists, and 'no', otherwise.

The required iterative learner uses the algorithm A' and the iterative learner M from the demonstration of Theorem 12 as its subroutines. Let $(L^*_{\langle k,j \rangle})_{k,j \in \mathbb{N}}$ be an indexing of $\mathcal{C}_{\text{rp}}$ with $L^*_{\langle k,j \rangle} = L_\varepsilon(\alpha_k)$ for all $k, j \in \mathbb{N}$. We define an iterative learner M'' for $\mathcal{C}_{\text{rp}}$ that uses the hypothesis space $(L^*_{\langle k,j \rangle})_{k,j \in \mathbb{N}}$.

Initially, if the first string w appears, M'' determines the canonical index k of the regular erasing pattern language $L_\varepsilon(w)$ as well as $j = M(init, w)$, and outputs the hypothesis $\langle k, j \rangle$. Next M'', when receiving a string v, refines its recent hypothesis, say $\langle k', j' \rangle$, as follows.

- Let c be the canonical index of the regular erasing pattern language $L_\varepsilon(x_1)$
- First, if $L_\varepsilon(\alpha_{k'}) = \{v\}$, M'' repeats its recent hypothesis, that is to say, $M''(\langle k', j' \rangle, v) = \langle k', j' \rangle$. (* Note that $j' = M(j', v)$, too. *)
- Second, if $L_\varepsilon(\alpha_{k'}) \neq \{v\}$, M'' determines $j = M(j', v)$ and runs A' on input j. If A' returns some $k \in \mathbb{N}$, M'' returns $\langle k, j \rangle$, i. e., $M''(\langle k', j' \rangle, v) = \langle k, j \rangle$. If A' returns 'no', M'' returns $\langle c, j \rangle$, i. e., $M''(\langle k', j' \rangle, v) = \langle c, j \rangle$.

Since $L_\varepsilon(x_1) = \Sigma^*$, M'' is an iterative and globally consistent learner. Moreover, the same arguments as in the proof of Theorem 15 can be used to verify that M'' learns every $L \in \mathcal{C}_{\text{rp}}$. □

This case study shows that the necessity of auxiliary hypotheses representing languages outside the target class may depend on whether both global consistency and local conservativeness or only one of these properties is required. In what follows, we analyse the impact of consistency and conservativeness separately in a more general context, assuming that auxiliary hypotheses are allowed.

4 Incremental learning and consistency

This section is concerned with the impact of consistency demands in iterative learning. In the case of learning from text, the weaker consistency demand, namely local consistency, does not restrict the capabilities of iterative learners.

Theorem 17 $ItLConsTxt = ItTxt$.

Proof. By definition, $ItLConsTxt \subseteq ItTxt$. To prove $ItTxt \subseteq ItLConsTxt$, fix an indexable class $\mathcal{C} \in ItTxt$. Let $(L_j)_{j\in\mathbb{N}}$ be an indexing comprising $\mathcal{C}$ and M an iterative learner for $\mathcal{C}$ with respect to $(L_j)_{j\in\mathbb{N}}$.

The required learner M' uses the indexing $(L'_{\langle j,w\rangle})_{j\in\mathbb{N},w\in\Sigma^*}$, where $L'_{\langle j,w\rangle} = L_j \cup \{w\}$ for all $j \in \mathbb{N}$, $w \in \Sigma^*$. Initially, $M'(init, w) = \langle j, w\rangle$ for $j = M(init, w)$. Next M', upon a string v, refines its recent hypothesis, say $\langle j', w'\rangle$, as follows. First, M' determines $j = M(j', v)$. Second, if $v \in L_j$, M returns $\langle j, w'\rangle$; otherwise, it returns $\langle j, v\rangle$. Obviously, M' witnesses $\mathcal{C} \in ItLConsTxt$. □

In contrast to that, requiring global consistency results in a loss of learning potential, as the following theorem shows.

Theorem 18 $ItGConsTxt \subset ItTxt$.

Proof. By definition, $ItGConsTxt \subseteq ItTxt$. It remains to provide a separating class $\mathcal{C}$ that witnesses $ItTxt \setminus ItGConsTxt \neq \emptyset$.

Let $\Sigma = \{a, b\}$ and let $(A_j)_{j\in\mathbb{N}}$ be the canonical enumeration of all finite subsets of $\{a\}^+$. Now $\mathcal{C}$ contains the language $L = \{a\}^+$ and, for all $j \in \mathbb{N}$, the finite language $L_j = A_j \cup \{b^z \mid z \leq j\}$.

Claim 19 $\mathcal{C} \in ItTxt$.

The required iterative learner M may work as follows. As long as exclusively strings from $\{a\}^+$ appear, M just guesses L. If a string of form b^j appears for the first time, M guesses L_j. Past that point, M, when receiving a string v, refines its recent guess, say L_k, as follows. If $v \in L$ or $v = b^z$ for some $z \leq k$, M repeats its guess L_k. If $v = b^z$ for some $z > k$, M guesses L_z.

It is not hard to verify that M is an iterative learner that learns $\mathcal{C}$ as required.

Claim 20 $\mathcal{C} \notin ItGConsTxt$.

Suppose to the contrary that there is an indexing $(L'_j)_{j\in\mathbb{N}}$ comprising $\mathcal{C}$ and a learner M witnessing $\mathcal{C} \in ItGConsTxt$ with respect to $(L'_j)_{j\in\mathbb{N}}$.

Consider M when processing the text $t = a^1, a^2, \ldots$ for L. Since M is a learner for $\mathcal{C}$, there has to be some n such that $M[init, t[n]] = M[init, t[n+m]]$ for all $m \geq 1$. (* Note that $M[init, t[n]] = M[init, t[n]a^z]$ for all $z > n + 1$. *)

Now let j be fixed such that $A_j = content(t[n]) = \{a^1, \ldots, a^{n+1}\}$. Consider M when processing any text $\hat{t}$ for L_j with $\hat{t}[n] = t[n]$. Since M is a learner for $\mathcal{C}$, there is some $n' > n$ such that $content(\hat{t}[n']) = L_j$ as well as $L'_k = L_j$ for $k = M[init, \hat{t}[n']]$. Fix a finite sequence σ with $\hat{t}[n'] = t[n]\sigma$. (* Note that such a sequence σ exists. *)

Next let $j' > j$ be fixed such that $A_j \subset A_{j'}$. Moreover fix any string a^z in $A_{j'} \setminus A_j$. (* Note that $z > n+1$ and $a^z \notin L_j$. *) Consider M when processing any text $\tilde{t}$ for the language $L_{j'}$ having the initial segment $\tilde{t}[n'+1] = t[n]a^z\sigma$. Since $M[init, t[n]] = M[init, t[n]a^z]$, one obtains $M[init, \tilde{t}[n+1]] = M[init, \hat{t}[n]]$. Finally since M is an iterative learner, $\hat{t}[n'] = \hat{t}[n]\sigma$, and $\tilde{t}[n'+1] = \tilde{t}[n+1]\sigma$, one can conclude that $M[init, \tilde{t}[n'+1]] = M[init, \hat{t}[n']] = k$. But $L'_k = L_j$, and therefore $a^z \notin L'_k$. The latter implies $content(\tilde{t}[n'+1]) \not\subseteq L'_k$, contradicting the assumption that M is an iterative and globally consistent learner for $\mathcal{C}$. □

In the case of learning from informant, the results obtained are parallel to those in the text case. Theorem 21 can be verified similarly to Theorem 17.

Theorem 21 *ItLConsInf* = *ItConsInf*.

Considering the stronger consistency requirement, there are even classes learnable iteratively from text, but not globally consistently from informant.

Theorem 22 $ItTxt \setminus ItGConsInf \neq \emptyset$.

Proof. A class $\mathcal{C} \in ItTxt \setminus ItGConsInf$ can be defined as follows:

Let $\Sigma = \{a, b\}$ and let $(A_j)_{j \in \mathbb{N}}$ be the canonical enumeration of all finite subsets of $\{a\}^+$. Now $\mathcal{C}$ contains the language $L = \{a\}^+$ and, for all $j, k \in \mathbb{N}$, the finite language $L_{\langle j,k \rangle} = A_j \cup A_k \cup \{b^j, b^k\}$.

Claim 23 $\mathcal{C} \in ItTxt$.

The required iterative learner M may work as follows. As long as only strings from $\{a\}^+$ appear, M guesses L. If a string of form b^z appears for the first time, M guesses $L_{\langle z,z \rangle}$. Past that point, M refines its recent guess, say $L_{\langle j',k' \rangle}$, when receiving a string v as follows. If $j' = k'$ and $v = b^z$ with $z \neq j'$, M guesses $L_{\langle j',z \rangle}$. In all other cases, M repeats its guess $L_{\langle j',k' \rangle}$.

It is not hard to verify that M is an iterative learner that learns $\mathcal{C}$ as required.

Claim 24 $\mathcal{C} \notin ItGConsInf$.

Suppose to the contrary that there is an indexing $(L'_j)_{j\in\mathbb{N}}$ comprising $\mathcal{C}$ and a learner M witnessing $\mathcal{C} \in \mathit{ItGConsInf}$ with respect to $(L'_j)_{j\in\mathbb{N}}$.

Consider a fixed informant $i = ((w_n, b_n)_{n\in\mathbb{N}})$ for L. Since M is a learner for $\mathcal{C}$, there has to be some n such that $M[\mathit{init}, i[n]] = M[\mathit{init}, i[n+m]]$ for all $m \geq 1$.

Let j be fixed such that $\mathit{content}^+(i[n]) \subseteq A_j$ and $b^j \notin \mathit{content}^-(i[n])$. Now consider M when processing an informant $\hat{\imath}$ for $L_{\langle j,j\rangle}$ with $\hat{\imath}[n] = i[n]$. Since M is a learner for $\mathcal{C}$, there has to be some $n' > n$ such that $\mathit{content}^+(\hat{\imath}[n']) = L_{\langle j,j\rangle}$ and $L'_k = L_{\langle j,j\rangle}$ for $k = M[\mathit{init}, \hat{\imath}[n']]$. Fix a finite sequence σ such that $\hat{\imath}[n'] = i[n]\sigma$. (* Note that such a sequence σ exists. *)

Now let $k' > j$ be fixed such that $A_j \subset A_{k'}$, $\mathit{content}^-(\hat{\imath}[n]) \cap A_{k'} = \emptyset$, and $b^{k'} \notin \mathit{content}^-(\hat{\imath}[n])$. Let a^z be any string in $A_{k'} \setminus A_j$. (* Note that $a^z \notin L_{\langle j,j\rangle}$. *) Consider M when processing any informant $\tilde{\imath}$ for the language $L_{\langle j,k'\rangle}$ with $\tilde{\imath}[n'+1] = i[n](a^z,+)\sigma$. Since $M[\mathit{init}, i[n]] = M[\mathit{init}, i[n](a^z,+)]$, one obtains $M[\mathit{init}, \tilde{\imath}[n+1]] = M[\mathit{init}, \hat{\imath}[n]]$. Finally since M is an iterative learner, $\hat{\imath}[n'] = \hat{\imath}[n]\sigma$, and $\tilde{\imath}[n'+1] = \tilde{\imath}[n+1]\sigma$, one may conclude that $M[\mathit{init}, \tilde{\imath}[n'+1]] = M[\mathit{init}, \hat{\imath}[n']] = k$. But $L'_k = L_{\langle j,j\rangle}$, and therefore $a^z \notin L'_k$. The latter implies $\mathit{content}^+(\tilde{\imath}[n'+1]) \not\subseteq L'_k$, contradicting the assumption that M is an iterative and globally consistent learner for $\mathcal{C}$. □

Obviously $\mathit{ItTxt} \subseteq \mathit{ItInf}$, and thus we obtain the following corollary.

Corollary 25 $\mathit{ItGConsInf} \subset \mathit{ItInf}$.

5 Incremental learning and conservativeness

This section deals with conservativeness in the context of iterative learning. Here the results for learning from text differ from those for the informant case.

5.1 The case of learning from text

Let us first discuss the different conservativeness definitions in the context of learning from positive examples only. By definition, local conservativeness is a stronger demand, since the learner is required to maintain a hypothesis if it is consistent with the most recent piece of information, even if it contradicts some previously processed examples. However, it turns out that this demand does not have any negative effect on the capabilities of iterative learners. Intuitively, a globally conservative learner may change its mind depending on inconsistency with only a limited set of examples, which can be coded within the hypothesis.

Theorem 26 *$ItGConvTxt = ItLConvTxt$.*

Proof. By definition, $ItLConvTxt \subseteq ItGConvTxt$. Fix an indexable class $\mathcal{C} \in ItGConvTxt$; let $(L_j)_{j\in\mathbb{N}}$ be an indexing and M an iterative IIM identifying $\mathcal{C}$ globally conservatively with respect to $(L_j)_{j\in\mathbb{N}}$. It remains to prove $\mathcal{C} \in ItLConvTxt$. For that purpose, we need the following notion and technical claim.

Notion. For any text t and any $n \in \mathbb{N}$, let $mc(t[n], M)$ denote the set $\{t(0)\} \cup \{t(m) \mid 1 \leq m \leq n \text{ and } M[init, t[m-1]] \neq M[init, t[m]]\}$ of all strings in $content(t[n])$, which force M to change its mind when processing $t[n]$.

Claim 27 *Let $L \in \mathcal{C}$, t a text for L, and $n \in \mathbb{N}$. Let $j = M[init, t[n]]$. If $t(n+1) \cup mc(t[n], M) \subseteq L_j$, then $M[init, t[n+1]] = M[init, t[n]]$.*

Proof. Let $W = content(t[n+1]) \setminus L_j$. As $t(n+1) \cup mc(t[n], M) \subseteq L_j$, then $M[init, t[m+1]] = M[init, t[m]]$ for all $m < n$ with $t(m+1) \in W$. Now let τ be the subsequence of $t[n]$ obtained by deleting all $w \in W$ from $t[n]$. Obviously, $M[init, \tau] = M[init, t[n]]$ and $mc(t[n], M) \subseteq content(\tau) \subseteq L_j$. This implies

$$M[init, t[n+1]] = M[init, \tau t(n+1)] = M[init, \tau] = M[init, t[n]]\,,$$

because M is globally conservative for L. (QED, Claim 27).

Define an indexing $(L'_j)_{j\in\mathbb{N}}$ by $L'_{2\langle j,k\rangle} = L_j$ and $L'_{2\langle j,k\rangle+1} = \emptyset$ for all $j, k \in \mathbb{N}$. (* Note that all languages in the target class are required to be non-empty. However, since the hypothesis space in the model considered may in general strictly comprise the target class, here the use of the empty language as represented by an intermediate hypothesis is allowed. *)

We now define an IIM M' (witnessing $\mathcal{C} \in ItLConvTxt$ using $(L'_j)_{j\in\mathbb{N}}$), such that, on any finite text segment σ for some $L \in \mathcal{C}$, the following invariant holds:

$M'[init, \sigma] = 2\langle M[init, \sigma], k\rangle + y$ for some $k \in \mathbb{N}$, $y \in \{0, 1\}$, such that

- $D_k = mc(\sigma, M)$ (* and thus $D_k \subseteq content(\sigma)$ *).
- If $y = 0$, then $D_k \subseteq L_{M[init,\sigma]}$.

The reader may check that this invariant holds, if M' is defined as follows:

Definition of $M'(init, w)$, for $w \in \Sigma^*$: Let $j = M(init, w)$.

- If $w \in L_j$, let $M'(init, w) = 2\langle j, k\rangle$, where $D_k = \{w\}$.
- If $w \notin L_j$, let $M'(init, w) = 2\langle j, k\rangle + 1$, where $D_k = \{w\}$.

Definition of $M'(2\langle j,k\rangle + 1, w)$, for $w \in \Sigma^*$, $j,k \in \mathbb{N}$: Let $j' = M(j,w)$.

- If $j = j'$ and $D_k \subseteq L_j$, let $M'(2\langle j,k\rangle + 1, w) = 2\langle j,k\rangle$.
- If $j = j'$ and $D_k \not\subseteq L_j$, let $M'(2\langle j,k\rangle + 1, w) = 2\langle j,k\rangle + 1$.
- If $j \neq j'$, let $M'(2\langle j,k\rangle + 1, w) = 2\langle j',k'\rangle + 1$, where $D_{k'} = D_k \cup \{w\}$.

Definition of $M'(2\langle j,k\rangle, w)$, for $w \in \Sigma^*$, $j,k \in \mathbb{N}$: Let $j' = M(j,w)$.

- If $w \notin L_j$ and $j = j'$, let $M'(2\langle j,k\rangle, w) = 2\langle j,k\rangle + 1$.
- If $w \notin L_j$ and $j \neq j'$, let $M'(2\langle j,k\rangle, w) = 2\langle j',k'\rangle + 1$, where $D_{k'} = D_k \cup \{w\}$.
- If $w \in L_j$ (* by the invariant, there is some text segment σ with $M[\mathit{init},\sigma] = j$ and $D_k = mc(\sigma, M) \subseteq L_j$; hence $D_k \cup \{w\} \subseteq L_j$ and $j = j'$ by claim 27 *), let $M'(2\langle j,k\rangle, w) = 2\langle j,k\rangle$.

By definition, M' is locally conservative with respect to $(L'_j)_{j\in\mathbb{N}}$. Since M is globally conservative for $\mathcal{C}$ with respect to $(L_j)_{j\in\mathbb{N}}$ and because of the invariant, it is not hard to verify that M' learns $\mathcal{C}$ iteratively. Thus $\mathcal{C} \in \mathit{ItLConvTxt}$. □

So local and global conservativeness are equal constraints for iterative text learners. Whether they reduce the capabilities of iterative text learners in general, i. e., whether *ItGConvTxt* and *ItTxt* coincide, remains an open question.

5.2 The case of learning from informant

First, comparing the two versions of conservativeness, the informant case yields results different from those in the text case, namely that globally conservative iterative learners cannot be normalised to being locally conservative. In particular, the property that globally conservative learners can code all previously seen examples, for which their current hypothesis is inconsistent, no longer holds in the informant case.

Theorem 28 *ItLConvInf* $\subset$ *ItGConvInf*.

Proof. By definition, *ItLConvInf* $\subseteq$ *ItGConvInf*. Thus it remains to provide a separating class $\mathcal{C}$ that witnesses *ItGConvInf* $\setminus$ *ItLConvInf* $\neq \emptyset$.

Let $\Sigma = \{a\}$ and $(D_j)_{j\in\mathbb{N}}$ the canonical enumeration of all finite subsets of $\{a\}^+$. Assume $D_0 = \emptyset$. For all $j \in \mathbb{N}$, set $L_j = \{a^0\} \cup D_j$ and $L'_j = \{a\}^+ \setminus D_j$. Let $\mathcal{C}$ be the collection of all finite languages L_j and all co-finite languages L'_j.

Claim 29 $\mathcal{C} \in$ *ItGConvInf*.

For all $j, k, z \in \mathbb{N}$, let $H_{2\langle j,k,z\rangle} = \{a\}^+ \setminus \{a^z\}$ and $H_{2\langle j,k,z\rangle+1} = \{a^z\}$. Now the required iterative learner M, processing an informant $i = ((w_n, b_n))_{n\in\mathbb{N}}$ for some $L \in \mathcal{C}$ may work as follows.

(i) As long as neither $(a^0, +)$ nor $(a^0, -)$ appear, M guesses — depending on whether or not $(w_0, b_0) = (a^z, +)$ or $(w_0, b_0) = (a^z, -)$ — in the first case $H_{2\langle j,k,z\rangle}$, in the second case $H_{2\langle j,k,z\rangle+1}$, where $D_j = \mathit{content}^+(i[n])$ and $D_k = \mathit{content}^-(i[n])$ (* The recent guess of M is inconsistent, so M can change its mind without violating the global conservativeness demand. *)
(ii) If $(a^0, +)$ or $(a^0, -)$ appears for the first time, the following cases will be distinguished. If $w_0 = a^0$ and $b_0 = +$, M guesses L_0. If $w_0 = a^0$ and $b_0 = -$, M guesses L'_0. Otherwise, let $j' = 2\langle j, k, z\rangle + y$, $y \in \{0, 1\}$, denote the recent guess of M. If $(a^0, +)$ appears, M' guesses the finite language L_j. If $(a^0, -)$ appears, M' guesses the co-finite language L'_k.
(iii) Then M refines its recent guess as follows. If a positive example $(a^z, +)$ appears, the recent guess of M is $L_{j'}$, and $a^z \notin L_{j'}$, M guesses $L_j = L_{j'} \cup \{a^z\}$. If a negative example $(a^z, -)$ appears, the recent guess of M is $L'_{k'}$, and $a^z \in L'_{k'}$, M guesses $L'_k = L'_{k'} \setminus \{a^z\}$. Else M repeats its recent guess.

It is not hard to verify that M is an iterative learner that learns $\mathcal{C}$ as required.

Claim 30 $\mathcal{C} \notin \mathit{ItLConvInf}$.

Suppose to the contrary that there is an indexing $(L^*_j)_{j\in\mathbb{N}}$ comprising $\mathcal{C}$ and a learner M which locally conservatively identifies $\mathcal{C}$ with respect to $(L^*_j)_{j\in\mathbb{N}}$.

Let $j = M(\mathit{init}, (a, +))$. We distinguish the following cases:

Case 1. $L^*_j \cap \{a\}^+$ is infinite.

Choose $a^r \in L^*_j$ with $r > 1$ and $L = \{a^0, a^1, a^r\}$. Consider M on the informant $i = (a, +), (a^r, +), (a^0, +), (a^2, -), \ldots, (a^{r-1}, -), (a^{r+1}, -), (a^{r+2}, -), \ldots$ for L. As M learns $\mathcal{C}$, there is an $n \geq 2$ with $M[\mathit{init}, i[n]] = M[\mathit{init}, i[n + m]]$ for all $m \geq 1$. (* $M[\mathit{init}, i[n](a^s, -)] = M[\mathit{init}, i[n]]$ for all a^s with $a^s \notin (\mathit{content}^+(i[n]) \cup \mathit{content}^-(i[n]))$. *) Let a^s be any string in L^*_j with $s > r + 1$, $a^s \notin (\mathit{content}^+(i[n]) \cup \mathit{content}^-(i[n]))$. As $L_j \cap \{a\}^+$ is infinite, such a^s exists. Fix some σ with $i = (a, +), (a^r, +)\sigma(a^{s-1}, -), (a^s, -), (a^{s+1}, -), \ldots$

Next let $\hat{\imath} = (a^1, +), (a^r, +), (a^s, +)\sigma(a^{s-1}, -), (a^{s+1}, -), (a^{s+2}, -), \ldots$ Consider M when processing the informant $\hat{\imath}$ for $L' = \{a^0, a^1, a^r, a^s\}$. Since M is locally conservative and $a^s \in L^*_j$, $M[\mathit{init}, \hat{\imath}[2]] = M[\mathit{init}, i[1]]$. As M is an iterative learner, $M[\mathit{init}, \hat{\imath}[n+1]] = M[\mathit{init}, i[n]]$. Past step $n+1$, M receives only negative examples $(a^z, -)$ with $a^z \notin (\mathit{content}^+(i[n]) \cup \mathit{content}^-(i[n]))$. Hence M converges on $\hat{\imath}$ to the same hypothesis j as on i, namely to $j = M[\mathit{init}, i[n]]$. Finally because $L \neq L'$, M cannot learn both finite languages L and L'.

Case 2. $L_j^* \cap \{a\}^+$ is finite.

An argumentation similar to that used in Case 1 shows that M must fail to learn some co-finite language in $\mathcal{C}$. We omit the relevant details. □

The observed difference in the above theorem can now even be extended to a proper hierarchy of iterative learning from informant; globally conservative learners in general outperform locally conservative ones, but are not capable of solving all the learning tasks a general iterative learner can cope with. So there are classes in *ItInf* which cannot be learned by any iterative, globally conservative learner.

Theorem 31 $\mathit{ItGConvInf} \subset \mathit{ItInf}$.

Proof. By definition, $\mathit{ItGConvInf} \subseteq \mathit{ItInf}$. Thus it remains to provide a separating class $\mathcal{C}$ that witnesses $\mathit{ItInf} \setminus \mathit{ItGConvInf} \neq \emptyset$.

Let $(D_j)_{j\in\mathbb{N}}$ be the canonical enumeration of all finite subsets of $\mathbb{N}$.

Let $\mathcal{C} = \bigcup_{k\in\mathbb{N}} \mathcal{C}_k$, where $\mathcal{C}_k$ is defined below based on the following cases.

Case (a). If $\varphi_k(k)\uparrow$, then $\mathcal{C}_k$ contains just one language, namely $L_k = \{a^k\}$.

Case (b). If $\varphi_k(k)\downarrow$, then $\mathcal{C}_k$ contains infinitely many languages. Let $s = \Phi_k(k)$. For all $j \in \mathbb{N}$, $\mathcal{C}_k$ contains the language $L_{\langle k,j\rangle} = \{a^k\} \cup \{c^s\} \cup \{d^{s+z} \mid z \in D_j\}$ as well as the language $L'_{\langle k,j\rangle} = \{a^k\} \cup \{d^{s+z} \mid z \notin D_j\}$. (* Note that $L_{\langle k,j\rangle}$ contains a finite subset of $\{d\}^*$, whereas $L'_{\langle k,j\rangle}$ contains a co-finite subset of $\{d\}^*$. *)

It is not hard to verify that $\mathcal{C}$ constitutes an indexable class.

Claim 32 $\mathcal{C} \in \mathit{ItInf}$.

Let $i = ((w_n, b_n))_{n\in\mathbb{N}}$ be an informant for some $L \in \mathcal{C}$. A corresponding iterative learner M' may be informally defined as follows:

(i) As long as no positive example $(a^k, +)$ appears, M' encodes in its guess all examples seen so far.
(ii) If some positive example $(a^k, +)$ appears, M' tests whether or not $\Phi_k(k) \leq |w|$, where w is the longest string seen so far. In case that $\varphi_k(k)\downarrow$ has been verified, M' guesses L_k, where in its hypothesis all examples seen so far are encoded. Subsequently, M' behaves according to *(iv)*. In case that $\Phi_k(k) > |w|$, M' guesses L_k, where the encoded examples can be simply ignored. Afterwards, M' behaves according to *(iii)*.
(iii) As long as M' guesses L_k, M' uses the recent example (w_n, b_n) to check whether or not $\Phi_k(k) \leq |w_n|$. In the positive case, M' behaves as in *(iv)*.

Else M' repeats its recent guess, without encoding any further example.

(iv) Let $s = \Phi_k(k)$. As long as $(c^s, +)$ and $(c^s, -)$ neither appear nor belong to the examples encoded in the recent guess, M' adds the new example into the encoding of examples in the recent guess. If $(c^s, +)$ (or $(c^s, -)$) appears or is encoded, M' guesses a language $L_{\langle k,j\rangle}$ (or $L'_{\langle k,j\rangle}$, respectively), where j is chosen such that D_j is the set of all z for which $(d^{s+z}, +)$ (or $(d^{s+z}, -)$, respectively) is encoded in the previous hypothesis or as the current example. M' can then identify the target language by explicitly coding any further positive/negative examples of $\{d\}^*$ occurring—this is done in a way similar to the proof of Claim 29.

It is not hard to see that M' is an iterative learner for $\mathcal{C}$.

Claim 33 $\mathcal{C} \notin \mathit{ItGConvInf}$.

Suppose the converse. That is, there is an indexing $(L^*_j)_{j\in\mathbb{N}}$ comprising $\mathcal{C}$ and an iterative learner M which globally conservatively identifies $\mathcal{C}$ with respect to $(L^*_j)_{j\in\mathbb{N}}$. We shall show that M can be utilised to solve the halting problem.

Algorithm $\mathcal{A}$: Let k be given. Let $i = (w_n, b_n)_{n\in\mathbb{N}}$ be a repetition-free informant for L_k with $w_0 = a^k$ and $b_0 = +$ such that, for all $n \in \mathbb{N}$, $w_m = c^n$ implies $m > n$. For $m = 0, 1, 2, \ldots$ test in parallel whether $(\alpha 1)$ or $(\alpha 2)$ happens.

$(\alpha 1)$ $\Phi_k(k) \le m$.

$(\alpha 2)$ An index $j_m = M(\mathit{init}, i[m])$ is output such that $\mathit{content}^+(i[m]) \subseteq L^*_{j_m}$ and $\mathit{content}^-(i[m]) \cap L^*_{j_m} = \emptyset$.

If $(\alpha 1)$ happens first, output '$\varphi_k(k)\downarrow$.' Otherwise, i.e., $(\alpha 2)$ happens first, output '$\varphi_k(k)\uparrow$.'

We next show:

(1) On every input k, algorithm $\mathcal{A}$ terminates.
(2) Algorithm $\mathcal{A}$ decides the halting problem.

ad (1). It suffices to show that either $(\alpha 1)$ or $(\alpha 2)$ happens. Suppose, $(\alpha 1)$ does not happen, and thus $\varphi_k(k)\uparrow$. Hence, $L_k \in \mathcal{C}_k \subseteq \mathcal{C}$. Consequently, M, when processing the informant i for L_k, eventually returns a hypothesis $j_m = M(\mathit{init}, i[m])$ such that $L^*_{j_m} = L_k$. Thus, $(\alpha 2)$ must happen.

ad (2). Obviously, if $(\alpha 1)$ happens then $\varphi_k(k)$ is indeed defined. Suppose $(\alpha 2)$ happens. We have to show that $\varphi_k(k)\uparrow$. Assume $\varphi_k(k)\downarrow$. Then, $\Phi_k(k) = s$ for some $s \in \mathbb{N}$. Since $(\alpha 2)$ happens, there is an $m < s$ such that $j_m = M(\mathit{init}, i[m])$ as well as $\mathit{content}^+(i[m]) \subseteq L^*_{j_m}$ and $\mathit{content}^-(i[m]) \cap L^*_{j_m} = \emptyset$. (* Note that neither $(c^s, +)$ nor $(c^s, -)$ appears in the initial segment $i[m]$. *)

Now, similarly to the proof of Claim 30 one has to distinguish two cases: (i) $L^*_{j_m}$ contains infinitely many strings from $\{d\}^*$ and (ii) $L^*_{j_m}$ contains only finitely many strings of from $\{d\}^*$. In both cases, an argumentation similar to that used in the proof of Claim 30 can be utilised to show that M fails to learn at least one language in $\mathcal{C}_k$ which contain a finite (co-finite) subset of $\{d\}^*$. We omit the relevant details. Since M is supposed to learn $\mathcal{C}$, the latter contradicts our assumption that $\varphi_k(k)\downarrow$, and thus Assertion (2) follows.

Since the halting problem is undecidable, $\mathcal{C} \notin \mathit{ItGConvInf}$. □

6 Discussion

6.1 Versions of consistency and conservativeness

We have studied iterative learning with two versions of consistency and conservativeness. In fact, a third sensible version is conceivable. Note that an iterative learner M may use a redundant hypothesis space for coding in its current hypothesis all examples, upon which M has previously changed its guess. So one may think of mind changes as 'memorising examples' and repeating hypotheses as 'forgetting examples'. One might call a hypothesis consistent with the examples seen, if it does not contradict the 'memorised' examples, i. e., those upon which M has changed its hypothesis. Similarly, M may be considered conservative, if M sticks to its recent hypothesis, as long as it agrees with the 'memorised' examples.

Obviously, this version of consistency is equivalent to local consistency – the proof is essentially the same as for Theorem 17 and the fact is not surprising.

However, the third version of conservativeness is worth considering a little closer. For iterative learning from text Theorem 26 immediately implies that this notion is equivalent to both global and local conservativeness. The idea is quite simple: a conservative learner really has to 'know' that it is allowed to change its hypothesis! Thus being inconsistent with forgotten positive examples doesn't help at all, because the learner cannot memorise the forgotten examples and thus not justify its mind change. In this sense, 'forgotten' examples are really examples without any relevance for the learner on the given text. This intuition is already reflected in Claim 27 used in the proof of Theorem 26.

Many similar insights may be taken from the proofs above to obtain further results. For instance, the separating classes provided in the proofs of Theorems 18 and 22, additionally lift our results to a more general case of incremental learning, where the learner has a k-bounded memory, i. e., the capacity for memorising up to k examples during the learning process, cf. Lange and Zeugmann [10].

6.2 Characterisations of iterative learning

One important step towards the understanding of learning according to a proposed formal model is usually achieved by characterising the corresponding learnable classes, for instance, in terms of structural properties. Such characterisations often have two advantages:

(1) they provide necessary and sufficient criteria for deciding whether a special class of languages is learnable in the model of interest;
(2) they often provide uniform algorithms for learning according to the specified model.

Thus such characterisation results convey an understanding for the typical structure of learnable classes and an understanding for a typical learning behaviour of a successful inference machine.

Very prominent examples in this context are the so-called 'telltale' characterisations for the classes learnable in the limit from text, see Angluin [1], as well as for the classes learnable in the limit from text by a conservative inductive inference machine, see Zeugmann and Lange [16]. These characterisations are built on families $(T_j)_{j\in\mathbb{N}}$ of finite sets T_j (telltales), which are uniformly recursively enumerable. The latter means that there is a two-place recursive function F, such that $\{F(j,n) \mid n \in \mathbb{N}\} = T_j$ for all $j \in \mathbb{N}$. If, in addition, it is decidable, given j, n_0, whether or not $\{F(j,n) \mid n < n_0\} = T_j$, then the family $(T_j)_{j\in\mathbb{N}}$ is called recursively generable. [3]

Theorem 34 (Angluin [1], Zeugmann and Lange [16]) *Let $\mathcal{C}$ be an indexable class of recursive languages.*

(1) $\mathcal{C} \in \mathit{LimTxt}$ iff there is an indexing $(L_j)_{j\in\mathbb{N}}$ of $\mathcal{C}$ and a uniformly recursively enumerable family $(T_j)_{j\in\mathbb{N}}$ of finite sets, such that for all $j, k \in \mathbb{N}$:
(a) $T_j \subseteq L_j$,
(b) $T_j \subseteq L_k \subseteq L_j$ implies $L_k = L_j$.
(2) $\mathcal{C} \in \mathit{ConvTxt}$ iff there is an indexing $(L_j)_{j\in\mathbb{N}}$ with $\mathcal{C} \subseteq \{L_j \mid j \in \mathbb{N}\}$ and a recursively generable family $(T_j)_{j\in\mathbb{N}}$ of finite sets, such that for all $j, k \in \mathbb{N}$:

[3] In fact, recursively generable families are families of finite sets which can be uniformly listed by a partial-recursive enumeration procedure, which stops if it has listed all elements in the respective set in the family.

(a) $T_j \subseteq L_j$*,*
(b) $T_j \subseteq L_k \subseteq L_j$ *implies* $L_k = L_j$*.*

The families $(T_j)_{j\in\mathbb{N}}$ are called telltale families, since the finite set T_j conveys a crucial piece of information about the language L_j with respect to the indexing $(L_j)_{j\in\mathbb{N}}$: every other language containing T_j in the indexing must contain at least one string not contained in L_j.

This theorem illustrates that special conditions on the algorithmic structure of a language class (here concerning telltale families) can be proven both necessary and sufficient for learnability with respect to some learning model. Moreover, the proof of the sufficiency part—in the case of Theorem 34 as well as for many similar characterisations—is constructive in the sense that a uniform learning method is provided. For instance, concerning Theorem 34, the learning procedures provided in the proofs work uniformly since they depend only on the particular indexing and telltale family. Consider as an example the uniform learning method for conservative inference from text, given an indexing $(L_j)_{j\in\mathbb{N}}$ comprising the target class as well as some recursively generable family $(T_j)_{j\in\mathbb{N}}$ fulfilling the conditions of Theorem 34.

- Assume an adequate hypothesis space containing all L_j, $j \in \mathbb{N}$, as well as all finite languages.
- Given the empty sequence, return a hypothesis representing the empty set.
- Given a text segment $\sigma = w_0 \dots w_n$ of length $n+1$, first test whether the hypothesis returned for $w_0 \dots w_{n+1}$ is still consistent. If it is, then return the same hypothesis again.
- Else look for some index $j \leq n$, such that $T_j \subseteq \{w_0, \dots, w_n\} \subseteq L_j$ (* note that this test is recursive because of the properties of recursively generable families *).
  If such an index j exists, then return a hypothesis representing L_j.
  If no such index j exists, then return a hypothesis representing the finite language $\{w_0, \dots, w_n\}$.

This illustrates the idea indicated already above: such a uniform learning method provided by a characterisation theorem helps to get a better understanding of how typical learning processes look like.

Now it is a would of course be an important, if not the essential step on the way to understanding the principles of incremental learning (or in particular iterative learning) to find a similar result for *ItTxt* or *ItInf*, i. e., a characterisation of the structure of the classes learnable iteratively. Unfortunately, among the results presented above, there is no such characterisation. However, we are still convinced that the questions discussed above are essential in the context of finding such characterisations. This concerns particularly the discussion of conservative iterative learning.

In the case of learning indexable classes of recursive languages from text, previous studies have shown an interesting (maybe coincidental?) relation between the utility of a characterisation and conservative learnability. This concerns inference types studied so far, for which the collection of learnable indexable classes is a subset of *ConvTxt*.

As it seems from previous results, the following relation might hold: Assume a text learning inference type *TTxt* is defined in a way similar to Definitions 1, 2, 6, such that $TTxt \subseteq ConvTxt$ holds for the corresponding collection of indexable classes. Then there is an accessible and well usable characterisation of the inference type, if and only if each indexable class in *TTxt* can be identified by an IIM which satisfies the constraints of the definition of *TTxt* and is conservative in parallel. [4]

Such a statement must be considered very vague, since of course there is no adequate definition of the term 'accessible and well usable characterisation'. However, this can be illustrated by two example inference types in the context of monotonic learning. The approach of monotonic learning deals with the reasonable constraint that, during the learning process, a learner should not 'unlearn' any part of the target concept which has once been reflected in its hypothesis, i. e., the part of the hypothesis which correctly reflects the target language, should be monotonically non-decreasing when processing a text. In a stronger version of monotonicity, one even requires that the sequence of languages hypothesized when processing the text is monotonically non-decreasing with respect to inclusion.

Definition 35 (Jantke [7], Wiehagen [15]) *Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ be a hypothesis space, and M an IIM.*

(1) M is monotonic for $\mathcal{C}$ iff, for every text t for some $L \in \mathcal{C}$ and any $n \in \mathbb{N}$, $L \cap L_{M(t[n])} \subseteq L \cap L_{M(t[n+1])}$.
(2) M is strongly monotonic for $\mathcal{C}$ iff, for every text t for some $L \in \mathcal{C}$ and any $n \in \mathbb{N}$, $L_{M(t[n])} \subseteq L_{M(t[n+1])}$.

Correspondingly, *MonTxt* and *SMonTxt* denote the collections of all indexable classes $\mathcal{C}'$ for which there is a hypothesis space $\mathcal{H}'$ and an IIM which learns $\mathcal{C}'$ from text with respect to $\mathcal{H}'$ and which is monotonic for $\mathcal{C}'$ or strongly monotonic for $\mathcal{C}'$, respectively.

[4] Note that $TTxt \subseteq ConvTxt$ trivially implies that each indexable class in *TTxt* can be identified by an IIM conservative on all relevant input sequences. However, this does not in general mean that conservative learning can be achieved by an IIM which fulfils the constraints of the definition of the inference type *TTxt* at the same time.

The analysis of monotonic inference has shown a proper hierarchy below the collection of all classes learnable conservatively in the limit from text:

Theorem 36 (Lange et al. [11]) *SMonTxt* $\subset$ *MonTxt* $\subset$ *ConvTxt.*

An interesting difference between a monotonic and a strongly monotonic behaviour is that IIMs working strongly monotonically can be normalised to strongly monotonic and conservative IIMs without loss of generality, whereas this does not hold for monotonic learning accordingly.

Theorem 37 (Lange et al. [11])

(1) There is an indexable class $\mathcal{C} \in$ *MonTxt such that there does not exist any IIM* M *witnessing* $\mathcal{C} \in$ *MonTxt which is conservative for* $\mathcal{C}$*.*
(2) For each indexable class $\mathcal{C} \in$ *SMonTxt there exists an IIM* M *witnessing* $\mathcal{C} \in$ *SMonTxt which is conservative for* $\mathcal{C}$*.*

Now, when looking at the currently known characterisations for *MonTxt* and *SMonTxt*, one finds significant differences concerning the intuition, readability, and thus utility.

Theorem 38 (Zeugmann et al. [17]) *Let* $\mathcal{C}$ *be an indexable class. Then* $\mathcal{C} \in$ *SMonTxt iff there is an indexing* $(L_j)_{j\in\mathbb{N}}$ *and a recursively generable family* $(T_j)_{j\in\mathbb{N}}$ *of finite sets such that*

(1) $\mathcal{C} \subseteq \{L_j \mid j \in \mathbb{N}\}$,
(2) $T_j \subseteq L_j$ *for all* $j \in \mathbb{N}$,
(3) $L_j \subseteq L$ *for all* $L \in \mathcal{C}$ *and all* $j \in \mathbb{N}$ *with* $T_j \subseteq L$.

Theorem 39 (Zeugmann et al. [17]) *Let* $\mathcal{C}$ *be an indexable class. Then* $\mathcal{C} \in$ *MonTxt iff there is an indexing* $(L_j)_{j\in\mathbb{N}}$*, a computable relation* R *over* $\mathbb{N}$*, and a recursively generable family* $(T_j)_{j\in\mathbb{N}}$ *of finite sets such that*

(1) $\mathcal{C} \subseteq \{L_j \mid j \in \mathbb{N}\}$,
(2) $T_j \subseteq L_j$ *for all* $j \in \mathbb{N}$,
(3) for all $L \in \mathcal{C}$*, all* $k \in \mathbb{N}$*, and all finite sets* $A \subseteq L$*, if* $T_k \subseteq L$ *and* $L_k \neq L$*, then there is a* j *such that* kRj *and* $A \subseteq T_j \subseteq L_j = L$,
(4) for all $L \in \mathcal{C}$*, there is no infinite sequence* $(k_r)_{r\in\mathbb{N}}$ *such that, for all* $r \in \mathbb{N}$, $k_r R k_{r+1}$ *and* $\bigcup_{r\in\mathbb{N}} T_{k_r} = L$,
(5) for all $L \in \mathcal{C}$ *and all* $k, j \in \mathbb{N}$*, if* kRj *and* $T_j \subseteq L$*, then* $L_k \cap L \subseteq L_j \cap L$.

The characterisation given here for *MonTxt* is mathematically not as pure and simple as the one for *SMonTxt*. Moreover, it does not reveal some interesting and essential structural property of the classes learnable in the corresponding model.

Having a closer look at the *SMonTxt*-characterisation, one can see that the special property of the telltale family easily yields an inference machine, which can learn strongly monotonically and simultaneously be conservative for the target class. Since this is in general impossible for monotonic learning, see Theorem 37.(1), it is not so astonishing that it is not so easy to find a nice characterisation for *MonTxt*.

And the clue why no nice characterisations for iterative learning are known so far, might be exactly that conservativeness might be a proper restriction for iterative IIMs when learning from text, i. e., that *ItTxt* might not be equal to *ItLConvTxt*. So studying conservativeness in the case of iterative learning can be considered a very promising approach towards a better understanding of iterative learning and incremental learning in general. In particular, the open question whether or not *ItTxt* equals *ItLConvTxt* could play a central role in further research in this context.

References

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

[2] Angluin, D., Queries and concept learning, *Machine Learning* **2**, 319–342, 1988.

[3] Blum, M., A machine independent theory of the complexity of recursive functions, *Journal of the ACM* **14**, 322–336, 1967.

[4] Case, J., Jain, S., Lange, S., and Zeugmann, T., Incremental concept learning for bounded data mining, *Information and Computation* **152**, 74–110, 1999.

[5] Gennari, J.H., Langley, P., and Fisher, D., Models of incremental concept formation, *Artificial Intelligence* **40**, 11–61, 1989.

[6] Gold, E.M., Language identification in the limit, *Information and Control* **10**, 447–474, 1967.

[7] Jantke, K.P., Monotonic and Non-monotonic Inductive Inference, *New Generation Computing* **8**, 349–360, 1991.

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

[9] Lange, S. and Grieser, G., On the power of incremental learning, *Theoretical Computer Science* **288**, 277-307, 2002.

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

[11] Lange, S. and Zeugmann, T., and Kapur, S., Monotonic and dual-monotonic language learning, *Theoretical Computer Science* **155**, 365–410, 1996.

[12] Shinohara, T., Polynomial time inference of extended regular pattern languages, *in: Proc. RIMS Symposium on Software Science and Engineering*, Lecture Notes in Computer Science, Vol. 147, pp. 115–127, Springer-Verlag, 1983.

[13] Valiant, L.G., A theory of the learnable, *Communications of the ACM* **27**, 1134–1142, 1984.

[14] Wiehagen, R., Limes-Erkennung rekursiver Funktionen durch spezielle Strategien, *Journal of Information Processing and Cybernetics (EIK)* **12** , 93–99, 1976.

[15] Wiehagen, R., A thesis in inductive inference, *in: Proc. International Workshop on Nonmonotonic and Inductive Logic*, Lecture Notes in Artificial Intelligence, Vol. 543, pp. 184–207, Springer-Verlag, 1991.

[16] Zeugmann, T. and Lange, S., A guided tour across the boundaries of learning recursive languages, *in: Algorithmic Learning for Knowledge-Based Systems*, Lecture Notes in Artificial Intelligence, Vol. 961, pp. 190–258, Springer-Verlag, 1995.

[17] Zeugmann, T., Lange, S., and Kapur, S., Characterizations of monotonic and dual monotonic language learning, *Information and Computation*, **120** 155–173, 1995.