Variants of iterative learning

Steffen Lange [a] and Gunter Grieser [b]

[a] *Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, Email: lange@dfki.de*

[b] *Technische Universität Darmstadt, FB Informatik, Alexanderstraße 10, 64283 Darmstadt, Germany, Email: grieser@informatik.tu-darmstadt.de*

Abstract

We investigate the principal learning capabilities of iterative learners in some more details. Thereby, we confine ourselves to study the learnability of indexable concept classes. The general scenario of iterative learning is as follows. An iterative learner successively takes as input one element of a text (an informant) for a target concept as well as its previously made hypothesis and outputs a new hypothesis about the target concept. The sequence of hypotheses has to converge to a hypothesis correctly describing the target concept.

We study two variants of this basic scenario and compare the learning capabilities of all resulting models of iterative learning to one another as well to the standard learning models finite inference, conservative identification, and learning in the limit.

First, we consider the case that an iterative learner has to learn from fat texts (fat informants), only. In this setting, it is guaranteed that relevant information is, in principle, accessible at any time in the learning process. Second, we study a variant of iterative learning, where an iterative learner is supposed to learn no matter which initial hypothesis is actually chosen. This variant is suited to describe scenarios that are typical for case-based reasoning.

1 Introduction

Induction constitutes an important feature of learning. The corresponding theory is called inductive inference. Inductive inference may be characterized as the study of systems that map evidence on a target concept into hypotheses about it. The investigation of scenarios in which the sequence of hypotheses stabilizes to an accurate and finite description of the target concept is of some particular interest. The precise definitions of the notions evidence, stabilization, and accuracy go back to Gold [9] who introduced the model of learning in the limit.

Preprint submitted to Elsevier Science *24 May 2006*

The general situation investigated in Gold's [9] model can be described as follows: Given more and more information concerning the concept to be learnt, the learning device has to produce hypotheses about the phenomenon to be inferred. The information sequence may contain only positive data, i.e., exactly all elements that constitute the concept to be recognized, as well as both positive and negative data, i.e., all elements of the underlying learning domain which are classified with respect to their containment in the unknown concept. Those information sequences are called text and informant, respectively. The sequence of hypotheses has to converge to a hypothesis correctly describing the object to be learnt. Consequently, the inference process is an ongoing one.

However, Gold's [9] model makes the unrealistic assumption that the learner has access to the whole initial segment of the information sequence provided so far. If huge data sets are around, no learning algorithm can use all the data or even large portions of it simultaneously for computing hypotheses about concepts represented by the data. Since each practical learning system has to deal with limitations of space, variants of the general approach restricting the accessibility of input data have been discussed in the computational learning theory community (cf., e.g., Wiehagen [29], Kinber and Stephan [13], Lange and Zeugmann [19], Jain *et al.* [11], Case *et al.* [4], Lange and Grieser [16], Lange [15]) as well as in the machine learning community (cf., e.g., Utgoff [26], Gennari *et al.* [7], Porat and Feldmann [22], Godin and Missaoui [8], Maloof and Michalski [21]). A prominent and intensively studied example is iterative learning. Here, the learning device (henceforth called iterative learner) is required to produce its actual hypothesis exclusively from its previous one and the next element in the information sequence.

Within the present paper, we investigate the principal learning capabilities of iterative learners in some more detail. Thereby, we confine ourselves to study the learnability of indexable concept classes (cf., e.g., Angluin [1], Zeugmann and Lange [30]). Our study draws its motivation from the rather simple observation that there is no learning *per se*. Learning is embedded into scenarios of a more comprehensive usage. Such an environment is usually putting constraints on the way information is accessible, requirements hypotheses have to meet, and so on.

For illustration, consider the following scenario which is typical for several approaches to case-based reasoning (cf., e.g., Kolodner [14]). A given case-based reasoning system is in use, i.e., some user is putting in repeatedly query cases and receives as the system's response proposals how to proceed with the query cases. If the proposals are satisfying, nothing has to be changed. If the outputs do not meet the user's expectations or the environmental needs, she is requested to provide data illustrating the system's misbehaviour. Based on this information, the system is supposed to change its state, and thereby to modify its behaviour appropriately. Thus, learning, in particular, some kind of iterative learning takes place. Learning succeeds, if the initial state is successfully transfered into a goal state (i.e., a state which meets all the users expectations) by processing only finitely many information units.

In order to gain a better understanding of the principal learning capabilities of those case-based reasoning systems, it seems to be reasonable to consider them as a certain kind of iterative learners. However, the basic model of iterative learning does not reflect all their specifics very well, and therefore some modifications are in order. Within the present paper we consider the following variants of the basic model of iterative learning:

First, in the learning scenario discussed above, it is highly desirable that every possible initial state of the system can be transformed into a goal state. The initial states of the system constitute treasures of experiences that have proved their usefulness in the past; so it is justified to keep these treasures if possible. Our notion of iterative learning with variable initial hypotheses (cf. Definition 5) reflects this intention. In contrast, in the basic model of iterative inference (cf. Definition 4), it is assumed that an iterative learner starts with an *a priori* fixed hypothesis. Since the initial hypothesis is here the same for all learning tasks, this hypothesis does not carry any message which in turn gives the learner the freedom to code, up to a certain extent, information about the progress made in the actual learning task directly into its intermediate hypotheses. In the modified model, such coding is meaningless, since every intermediate hypothesis may serve as initial hypothesis of different learning tasks, as well.

Second, in the basic model of iterative learning, a learner is supposed to learn on every possible information sequence. Thus, it may happen that relevant data items occur only once in the given information sequence. This may lead to situations in which relevant data items are overlooked, since they appear at the wrong time, and therefore learning may fail. However, this contradicts daily life experiences: if information is really important, it will not be presented only once. Fat information sequences have the property that every data item appears infinitely often, and therefore relevant data are, in principle, accessible at any time in the learning process. The corresponding learning model is called iterative learning from fat texts and informants, respectively (cf. Definition 6).

As we will see, iterative learners that are supposed to learn from fat information sequences, only, are much more powerful than those that have to be successful on every text and informant, respectively. On the one hand, when learning from positive data is concerned, iterative learning from fat information sequences is exactly as powerful as conservative inference which itself is less powerful than learning in the limit (cf. Corollary 4 and Proposition 1). On the other hand, iterative learning from fat informants is exactly as powerful as learning in the limit from informants (cf. Corollary 12). Consequently, if exclusively fat information sequences have to be processed, it is justified to use iterative learners instead of unconstrained ones.

Surprisingly, even iterative learning with variable initial hypotheses from fat informants turns out to be of the same learning power as learning in the limit (cf. Corollary 12). When learning from positive data is concerned, the situation changes. There are concept classes that are iteratively learnable from arbitrary texts and that cannot be iteratively learnt with variable initial hypotheses even in case that exclusively fat texts have to be processed (cf. Theorem 6).

As one may expect, the power of iterative learning with variable initial hypotheses from arbitrary texts and informants, respectively, is rather limited. In both cases, the corresponding learning model is incomparable to finite learning which itself is known to be very restrictive (cf. Theorems 9 and 16).

2 Preliminaries

$\mathbb{N} = \{0, 1, 2, ...\}$ is the set of all natural numbers. We set $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$. By $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote Cantor's pairing function. We write $A \# B$ to indicate that the sets A and B are incomparable, i.e., $A \setminus B \neq \emptyset$ and $B \setminus A \neq \emptyset$.

Let $(\varphi_j)_{j \in \mathbb{N}}$ denote any fixed acceptable programming system of all (and only all) partial recursive functions over $\mathbb{N}$ and let $(\Phi_j)_{j \in \mathbb{N}}$ be any associated complexity measure (cf. Blum [3]). Let $k, x \in \mathbb{N}$. Then, φ_k is the partial recursive function computed by program k in the programming system $(\varphi_j)_{j \in \mathbb{N}}$. Furthermore, if $\varphi_k(x)$ is defined (abbr. $\varphi_k(x) \downarrow$), then we also say that $\varphi_k(x)$ converges; otherwise, $\varphi_k(x)$ diverges (abbr. $\varphi_k(x) \uparrow$).

Any recursively enumerable set $\mathcal{X}$ is called a learning domain. By $\wp(\mathcal{X})$ we denote the power set of $\mathcal{X}$. Let $\mathcal{C} \subseteq \wp(\mathcal{X})$, and let $c \in \mathcal{C}$; then we refer to $\mathcal{C}$ and c as to a concept class and a concept, respectively. A concept class $\mathcal{C}$ is said to be inclusion-free iff $c \not\subseteq c'$ for all distinctive concepts $c, c' \in \mathcal{C}$.

In the sequel we deal with the learnability of indexable concept classes with uniformly decidable membership (cf. Angluin [1]). A class of non-empty concepts $\mathcal{C}$ is said to be an indexable class with uniformly decidable membership provided there are an effective enumeration $(c_j)_{j \in \mathbb{N}}$ of all and only the concepts in $\mathcal{C}$ and a recursive function f such that, for all $j \in \mathbb{N}$ and all $x \in \mathcal{X}$, the following holds:

$$f(j, x) = \begin{cases} 1, & \text{if} \quad x \in c_j, \\ 0, & \text{otherwise.} \end{cases}$$

In the following, we refer to indexable classes with uniformly decidable membership as to indexable classes, for short.

Next, we describe some well-known examples of indexable classes.

First, let Σ denote any fixed finite alphabet of symbols and let Σ^* be the free monoid over Σ. As usual, for all $a \in \Sigma$ and all $n \in \mathbb{N}$, we let $a^{n+1} = aa^n$, while, by convention, a^0 equals the empty string. Then, we let $\mathcal{X} = \Sigma^*$ be the learning domain. We refer to subsets $L \subseteq \Sigma^*$ as to languages (instead of concepts). For instance, the set of all context-sensitive languages, context-free languages, regular languages, and of all pattern languages $\mathcal{C}_{pat}$ (see also Section 4) form indexable classes (cf., e.g., Hopcroft and Ullman [10], Angluin [1]).

Second, let $X_n = \{0, 1\}^n$ be the set of all n-bit Boolean vectors. We consider $\mathcal{X} = \bigcup_{n \geq 1} X_n$ as learning domain. Then, the set of all concepts expressible as a monomial, a k-CNF, a k-DNF, and a k-decision list constitute indexable classes (cf., e.g., Valiant [27], Rivest [23]).

2.1 Gold-style learning from positive data

Let $\mathcal{X}$ be the underlying learning domain, let $c \subseteq \mathcal{X}$ be a concept, and let $t = (x_n)_{n \in \mathbb{N}}$ be an infinite sequence of elements from c such that $\{x_n \mid n \in \mathbb{N}\} = c$. Then, t is said to be a positive presentation or, synonymously, a text for c. By $\mathit{text}(c)$ we denote the set of all texts for c. As in Jain *et al.* [11], a text t is said to be fat provided that every element from c appears infinitely often, i.e., for all $x \in c$, there are infinitely many $n \in \mathbb{N}$ with $x_n = x$. By $\mathit{ftext}(c)$ we denote the set of all fat texts for c. (Note that, by definition, $\mathit{ftext}(c) \subseteq \mathit{text}(c)$.) Moreover, let t be a text and let y be a number. Then, t_y denotes the initial segment of t of length $y + 1$ and $t_y^+ = \{x_n \mid n \leq y\}$. Additionally, by $\sigma \diamond \tau$ we denote the concatenation of two finite sequences σ and τ.

As in Gold [9], we define an inductive inference machine (abbr. IIM) to be an algorithmic device working as follows: The IIM takes as its input larger and larger initial segments of a positive presentation. After processing an initial segment, the IIM either outputs a hypothesis, i.e., a number encoding a certain computer program, or it outputs '?,' a special symbol representing the case the machine outputs 'no conjecture'. More formally, an IIM maps finite sequences of elements from $\mathcal{X}$ into elements from $\mathbb{N} \cup \{?\}$.

The numbers output by an IIM are interpreted with respect to a suitably chosen hypothesis space $\mathcal{H}$. Since we exclusively deal with indexable classes $\mathcal{C}$, we always take as a hypothesis space an indexable class $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$. The indices are regarded as suitable finite encodings of the concepts described by the hypotheses. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing h_j. Clearly, $\mathcal{H}$ must be defined over some learning domain $\mathcal{X}$ which comprises the learning domain over which $\mathcal{C}$ is defined, and, moreover, $\mathcal{H}$ must comprise the target concept class $\mathcal{C}$. More formally speaking, we deal with class comprising learning (cf. Lange and Zeugmann [18]).

Let t be a positive presentation and let $y \in \mathbb{N}$. Then, we use $M(t_y)$ to denote the hypothesis produced by M when fed the initial segment t_y. The sequence $(M(t_y))_{y\in\mathbb{N}}$ is said to converge to the number j iff all but finitely many terms of the sequence $(M(t_y))_{y\in\mathbb{N}}$ are equal to j.

Next, we define some models of learning. We start with learning in the limit.

Definition 1 (Gold [9]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space. An IIM M $LimTxt_{\mathcal{H}}$–identifies c iff, for every $t \in text(c)$, there exists a $j \in \mathbb{N}$ such that $c = h_j$ and the sequence $(M(t_y))_{y\in\mathbb{N}}$ converges to j.*

Furthermore, M $LimTxt_{\mathcal{H}}$–identifies $\mathcal{C}$ iff, for each $c \in \mathcal{C}$, M $LimTxt_{\mathcal{H}}$–identifies c.

Finally, $LimTxt$ denotes the collection of all indexable classes $\mathcal{C}'$ for which there are an IIM M' and a hypothesis space $\mathcal{H}'$ such that M' $LimTxt_{\mathcal{H}'}$–identifies $\mathcal{C}'$.

In the above definition, *Lim* stands for “limit”. Suppose, an IIM identifies some concept c. That means, after having seen only finitely many data of c the IIM reaches its (unknown) point of convergence and it computes a correct and finite description of the target concept c. Hence, some form of learning must have taken place.

In general, it is not decidable whether or not an IIM M has already converged on a text t for a target concept c. Adding this requirement to Definition 1 results in finite learning.

Definition 2 (Gold [9]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space. An IIM M $FinTxt_{\mathcal{H}}$–identifies c iff, for every $t \in text(c)$, there exist $j, m \in \mathbb{N}$ such that $c = h_j$ as well as, for all $y \in \mathbb{N}$, $M(t_y) = ?$, if $y < m$, and $M(t_y) = j$, if $y \geq m$.*

Furthermore, M $FinTxt_{\mathcal{H}}$–identifies $\mathcal{C}$ iff, for each $c \in \mathcal{C}$, M $FinTxt_{\mathcal{H}}$–identifies c.

Finally, $FinTxt$ denotes the collection of all indexable classes $\mathcal{C}'$ for which there are an IIM M' and a hypothesis space $\mathcal{H}'$ such that M' $FinTxt_{\mathcal{H}'}$–identifies $\mathcal{C}'$.

Now, we define conservative IIMs. Intuitively, conservative IIMs maintain their actual hypothesis at least as long as they have not received data that “provably misclassify” it.

Definition 3 (Angluin [2]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ be a hypothesis space. An IIM M $ConsvTxt_{\mathcal{H}}$–identifies c iff M $LimTxt_{\mathcal{H}}$–identifies c, and, for every $t \in text(c)$ and for all $y, j \in \mathbb{N}$, condition (a) is fulfilled, where*

(a) if $j = M(t_y)$ and $M(t_y) \neq M(t_{y+1})$, then $t^+_{y+1} \not\subseteq h_j$.

Furthermore, M $ConsvTxt_{\mathcal{H}}$–identifies $\mathcal{C}$ iff, for each $c \in \mathcal{C}$, M $ConsvTxt_{\mathcal{H}}$–identifies c.

Finally, ConsvTxt denotes the collection of all indexable classes $\mathcal{C}'$ for which there are an IIM M' and a hypothesis space $\mathcal{H}'$ such that M' $ConsvTxt_{\mathcal{H}'}$–identifies $\mathcal{C}'$.

As it turned out, for proving some of the results below, it is conceptually simpler to use the characterization of conservative learning equating it with set-driven inference (cf. Lange and Zeugmann [20]). Set-drivenness has been introduced by Wexler and Culicover [28] and describes the requirement that the output of an IIM is only allowed to depend on the range of its input. More formally, an IIM M is said to be set-driven with respect to $\mathcal{C}$ iff, for all $y, y' \in \mathbb{N}$ and all texts $t, \hat{t}$ for concepts in $\mathcal{C}$, $t^+_y = \hat{t}^+_{y'}$ implies $M(t_y) = M(\hat{t}_{y'})$. By *s-LimTxt* we denote the collection of all indexable classes $\mathcal{C}'$ for which there are a hypothesis space $\mathcal{H}'$ and a set-driven IIM M' that $LimTxt_{\mathcal{H}'}$–identifies $\mathcal{C}'$.

2.2 Formalizing variants of iterative learning from positive data

Looking at the above definitions, we see that an IIM M has always access to the whole history of the learning process, i.e., in order to compute its actual guess, M is fed all examples seen so far. In contrast to that, next we define iterative inductive inference machines. An iterative IIM is only allowed to use its last guess and the next element in the positive presentation of the target concept for computing its actual guess.

More formally, let $\mathcal{X}$ be the underlying learning domain. Then, an iterative IIM M is an algorithmic device that maps elements from $\mathbb{N} \times \mathcal{X}$ into $\mathbb{N}$. Let $t = (x_n)_{n \in \mathbb{N}}$ be any text for some concept $c \subseteq \mathcal{X}$, and let k be M's initial hypothesis. Then, we denote by $(M_n(k, t))_{n \in \mathbb{N}}$ the sequence of hypotheses generated by M when successively fed t, i.e., $M_0(k, t) = M(k, x_0)$ and, for all $n \in \mathbb{N}$, $M_{n+1}(k, t) = M(M_n(k, t), x_{n+1})$. In the next definition, it is assumed that M's initial hypothesis is *a priori* fixed in that it equals 0.

Definition 4 (Wiehagen [29]). *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space. An iterative IIM M $ItTxt_{\mathcal{H}}$–identifies c iff, for every $t \in text(c)$, there exists a $j \in \mathbb{N}$ such that $c = h_j$ and the sequence $(M_n(0,t))_{n\in\mathbb{N}}$ converges to j.*

Finally, M $ItTxt_{\mathcal{H}}$–identifies $\mathcal{C}$ iff, for each $c \in \mathcal{C}$, M $ItTxt_{\mathcal{H}}$–identifies c.

The resulting learning type *ItTxt* is defined analogously to Definitions 1 to 3.

Subsequently, we use the following convention. Let σ be any finite sequence of elements over the relevant learning domain. Then, we denote by $M_*(k,\sigma)$ the last hypothesis output by M when successively fed σ (as above, k denotes M's initial hypothesis).

In the following definition, we consider a variant of iterative learning, where an iterative IIM has to learn successfully no matter which initial hypothesis has been selected.

Definition 5. *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space. An iterative IIM M $It^vTxt_{\mathcal{H}}$–identifies c iff, for every $t \in text(c)$ and every initial hypothesis $k \in \mathbb{N}$, there exists a $j \in \mathbb{N}$ such that $c = h_j$ and the sequence $(M_n(k,t))_{n\in\mathbb{N}}$ converges to j.*

Finally, M $It^vTxt_{\mathcal{H}}$–identifies $\mathcal{C}$ iff, for each $c \in \mathcal{C}$, M $It^vTxt_{\mathcal{H}}$–identifies c.

The resulting learning type *It*^{v}*Txt* is defined analogously to Definitions 1 to 3.

Finally, we define versions of the models of iterative learning introduced above in which it is sufficient that an iterative learner is successful on the subset of all fat texts. More formally:

Definition 6. *Let $\mathcal{C}$ be an indexable class, let c be a concept, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space. An iterative IIM M $ItFTxt_{\mathcal{H}}$ $[It^vFTxt_{\mathcal{H}}]$–identifies c iff, for every fat text $t \in ftext(c)$ [and every initial hypothesis $k \in \mathbb{N}$], there exists a $j \in \mathbb{N}$ such that $c = h_j$ and the sequence $(M_n(0,t))_{n\in\mathbb{N}}$ $[(M_n(k,t))_{n\in\mathbb{N}}]$ converges to j.*

Finally, M $ItFTxt_{\mathcal{H}}$ $[It^vFTxt_{\mathcal{H}}]$–identifies $\mathcal{C}$ iff, for each $c \in \mathcal{C}$, M $ItFTxt_{\mathcal{H}}$ $[It^vFTxt_{\mathcal{H}}]$–identifies c.

The resulting learning types *ItFTxt* and *It*^{v}*FTxt* are defined analogously to Definitions 1 to 3.

At the end of this subsection, we define the following notion.

Definition 7. *Let c be a concept, let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space, let M be an iterative IIM, and let $k \in \mathbb{N}$. Then, k is a locking hypothesis of M*

for c iff (i) $h_k = c$, (ii), for all $x \in c$, $M(k, x) = k$, and (iii) there is text t for c on which M eventually outputs k.

The following simple observation shows the importance of this notion concerning iterative learning.

Observation 1. *Let c be a concept, let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space, and let M be an iterative IIM that $LimTxt_{\mathcal{H}}$–identifies c. Then, there is a $k \in \mathbb{N}$ that constitutes a locking hypothesis k of M for c.*

Proof. Since M $LimTxt_{\mathcal{H}}$–identifies c, M, in particular, learns c on every fat text for it. So, let $t = (x_n)_{n\in\mathbb{N}}$ be a fat text for c and let $(M_n(0, t))_{n\in\mathbb{N}}$ be the sequence of hypotheses generated by M when successively fed t. Since M learns c on t, there are $m, k \in \mathbb{N}$ such that $h_k = c$ and, for all $r \geq 1$, $k = M_m(0, t) = M_{m+r}(0, t)$. Now, since M is an iterative IIM, we may conclude that, for all $r \geq 1$, $M(k, x_{m+r}) = k$. Hence, (i) and (ii) are fulfilled. Since t constitutes a text for c, we are done. Finally, notice that there are also non-fat texts for c on which M outputs k, namely on every text t' with $t'_m = t_m$. ∎

3 Iterative learning from positive data

In this section, we compare the learning capabilities of all models of iterative learning from positive data to one another as well as to finite inference, learning in the limit, and conservative identification from text.

First, we summarize the previously known results (cf. Lange and Zeugmann [17–20]).

Proposition 1. *$FinTxt \subset ItTxt \subset ConsvTxt = s\text{-}LimTxt \subset LimTxt$.*

The following example should help to illustrate the principal weakness of iterative learners. Consider the following indexable class $\mathcal{C}_{ex}$. Let $\mathcal{C}_{ex}$ be the collection of all concepts $c_j = \{a\}^+ \backslash \{a^{j+1}\}$. It is folklore that $\mathcal{C}_{ex} \in ConsvTxt$. Moreover, it is also well-known that $\mathcal{C}_{ex} \notin ItTxt$ (cf. Lange and Zeugmann [19]). To see the latter, suppose the converse, i.e., there are a hypothesis space $\mathcal{H}$ and an iterative IIM M that $ItTxt_{\mathcal{H}}$–identifies $\mathcal{C}_{ex}$. The basic idea is easily explained. M cannot successfully handle the following situation. Let k be a locking hypothesis of M for c_0. By Observation 1, such a hypothesis must exist. Moreover, let σ be an initial segment of a text for c_0 on which M outputs k. Now, after reading σ, M cannot encode any additional information in its actual hypothesis until the element $a \notin c_0$ possibly appears in the input data sequence. Consequently, M can be forced to forget some relevant information. If this relevant information will not be repeated, M will fail to learn some concept c_j with $c_j \neq c_0$.

In case that it is guaranteed that the relevant information appears infinitely often in a text, any conservative learner can be simulated by an iterative IIM that has the same learning power. Note that this gives us, in particular, $\mathcal{C}_{ex} \in \mathit{ItFTxt}$. More formally:

Theorem 2. *$\mathit{ConsvTxt} \subseteq \mathit{ItFTxt}$.*

Proof. Let $\mathcal{X}$ be the relevant learning domain over which $\mathcal{C}$ is defined. Assume $\mathcal{C} \in \mathit{ConsvTxt}$. Applying the characterization of *ConsvTxt* from Lange and Zeugmann [18], we know that there are a hypothesis space $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ and a computable function T that assigns a finite telltale set T_j to every hypothesis h_j. More formally, on every input $j \in \mathbb{N}$, T enumerates a finite set T_j and stops (i.e., all sets T_j are finite and recursively generable). Furthermore, for all $j \in \mathbb{N}$, T_j meets conditions (1) and (2), where

(1) $T_j \subseteq h_j$.
(2) for all $k \in \mathbb{N}$, $T_j \subseteq h_k$ implies $h_k \not\subset h_j$.

Without loss of generality, we may assume that, for all $j, k \in \mathbb{N}$, $h_j = h_k$ implies $T_j = T_k$.[1]

Let $\mathcal{F} = (F_j)_{j \in \mathbb{N}}$ denote any repetition free enumeration of all finite subsets of $\mathcal{X}$, where $F_0 = \emptyset$. Furthermore, we assume an effective procedure computing, for every finite set $F \subseteq \mathcal{X}$, its uniquely determined index $\#(F)$ in $\mathcal{F}$. Let f be any total recursive function such that, for all $n \in \mathbb{N}$, there are infinitely many $j \in \mathbb{N}$ with $f(j) = n$. To show that $\mathcal{C} \in \mathit{ItFTxt}$, we select a hypothesis space $\mathcal{H}' = (h'_j)_{j \in \mathbb{N}}$ that meets, for all $j, k \in \mathbb{N}$, $h'_{\langle j,k \rangle} = h_{f(j)}$.

Note that, by definition of Cantor's pairing function, $\langle 0, 0 \rangle = 0$ which is, by definition, M's initial hypothesis.

IIM M: "On input $\langle j, k \rangle$ and x do the following:

Set $F' = F_k \cup \{x\}$. If $T_{f(j)} \subseteq F' \subseteq h_{f(j)}$ then goto (A). Otherwise, goto (B).

(A) Set $S = \bigcup_{z \leq j} T_{f(z)}$ and test whether or not $x \in S$. In case it is, set $F'' = F'$. Otherwise, set $F'' = F_k$. Output $\langle j, \#(F'') \rangle$ and goto Stage $n + 1$.
(B) Output $\langle j + 1, \#(F') \rangle$ and goto Stage $n + 1$.

By definition and since all telltale sets T_j are finite and recursively generable, M is indeed an iterative IIM. We claim that M learns as required.

[1] The appropriateness of this assumption is based on the following fact: Given any enumeration $(c_j)_{j \in \mathbb{N}}$ of any indexable class $\mathcal{C}$, one can effectively construct an enumeration $(c'_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ and a total recursive function f such that (i) the set $\{(j,k) \mid c'_j = c'_k\}$ is recursive and (ii), for all $j \in \mathbb{N}$, $c'_j = c_{f(j)}$ (cf., e.g., Eršov [5]).

So, let $c \in \mathcal{C}$, let $t = (x_n)_{n \in \mathbb{N}}$ be a fat text for c, and let $(\langle j_n, k_n \rangle)_{n \in \mathbb{N}}$ be the sequence of hypotheses generated by M when successively fed t. Furthermore, let $(j_n)_{n \in \mathbb{N}}$ and $(k_n)_{n \in \mathbb{N}}$ be the sequence of the projections to the first and second components of M's hypotheses, respectively. Next, we show that M $\mathit{ItFTxt}_{\mathcal{H}'}$–identifies c. The verification is based on the following claims.

Claim 1. *If the sequence $(j_n)_{n \in \mathbb{N}}$ converges, say to j, then $h_{f(j)} = c$.*

Suppose to the contrary that $h_{f(j)} \neq c$. Let y be the least index such that, for all $n \in \mathbb{N}$, $j_{y+n} = j$. By M's definition, $T_{f(j)} \subseteq F_{k_{y+1}}$. Moreover, $F_{k_{y+1}} \subseteq t^+_{y+1} \subseteq c$, and therefore $T_{f(j)} \subseteq c$. Next, since M converges to j and since t is a fat text for c, we may conclude that, by M's definition, $c \subseteq h_{f(j)}$. However, by assumption, $h_{f(j)} \neq c$, and therefore $T_{f(j)} \subseteq c$ and $c \subset h_{f(j)}$, contradicting Property (2) of the telltale set $T_{f(j)}$. This proves Claim 1.

Claim 2. *If the sequence $(j_n)_{n \in \mathbb{N}}$ converges, say to j, then the sequence $(k_n)_{n \in \mathbb{N}}$ converges, too.*

Let $y \in \mathbb{N}$ be fixed such that, for all $n \in \mathbb{N}$, $j_{y+n} = j$. By Claim 1, $h_{f(j)} = c$, and thus, for all $n \in \mathbb{N}$, $F_{k_{y+n}} \subseteq F_{k_y} \cup (\bigcup_{z \leq j} T_{f(z)})$. Now, since, by M's definition, the sequence $(F_{k_n})_{n \in \mathbb{N}}$ is monotonically increasing (with respect to set inclusion) and since every telltale set is finite, Claim 2 is shown.

Claim 3. *The sequence $(j_n)_{n \in \mathbb{N}}$ converges.*

Let j' be the least number such that $h_{f(j')} = c$, and let $y \in \mathbb{N}$ be the least index such that $j_y = j'$. By Claim 1, such a y must exist, since, by M's definition, $j_n \leq j_{n+1} \leq j_n + 1$ for all $n \in \mathbb{N}$. Furthermore, since t is a fat text for c and since $T_{f(j')} \subseteq c$, there has to be a least $m \in \mathbb{N}$ such that $T_{f(j')} \subseteq \{x_r \mid y \leq r \leq y+m\}$. Therefore, by M's definition, $T_{f(j')} \subseteq F_{k_{y+m}}$.

Next, let j be the least index such that $j \geq j_{y+m}$ and $h_{f(j)} = c$. We claim that the sequence $(j_n)_{n \in \mathbb{N}}$ converges to j. We distinguish the following cases.

Case 3.1: $j_{y+m} = j$.

Recall that $T_{f(j)} = T_{f(j')}$. Now, since t is a fat text for $c = h_{f(j)}$ and since, by M's definition, the sequence $(F_{k_n})_{n \in \mathbb{N}}$ is monotonically increasing (with respect to set inclusion), we may conclude that the sequence $(j_n)_{n \in \mathbb{N}}$ converges to j.

Case 3.2: $j_{y+m} \neq j$.

Note that $h_{f(j_{y+m})} \neq c$. First, by M's definition, if $T_{f(j_{y+m})} \not\subseteq c$ then $j_{y+m+1} = j_{y+m} + 1$. Second, let $T_{f(j_{y+m})} \subseteq c$. Then, by Property (2) of the telltale sets and since t is a fat text for c, there has to be an $n \in \mathbb{N}$ such that $x_{y+m+n} \notin$

$h_{f(j_{y+m})}$, and thus, by definition of M, $j_{y+m+n} \neq j_{y+m}$. By simply iterating this argumentation and since, by M's definition, $j_n \leq j_{n+1} \leq j_n + 1$ for all $n \in \mathbb{N}$, one easily sees that there is some $n' \in \mathbb{N}$ with $j_{y+m+n'} = j$. Hence, we are back in Case 3.1, and thus Claim 3 follows.

Combining Claims 1 to 3, one directly sees that M converges. Moreover, by the properties of $\mathcal{H}'$, M converges to a correct hypothesis for c, and thus we are done. ∎

Interestingly, iterative IIMs cannot outperform conservative learners, even in case that the iterative IIMs have to learn from fat texts, only. Thus, the principal weakness of iterative learners (compared to the capabilities of unconstrained IIMs) cannot be compensated, although each relevant data item appears infinitely often in the input data sequence. This result points to one of the peculiarities of learning indexable classes, in particular, and of learning machines, in general. In Jain *et al.* [11], it has been shown that, when learning from fat text is considered, non-computable iterative learners [2] are exactly as powerful as non-computable unconstrained learners. In order to elaborate the result mentioned above we heavily exploit the fact that conservative learners are exactly as powerful as set-driven IIMs (cf. Proposition 1). Thereby, we adapt an idea from Kinber and Stephan [13] and Lange and Zeugmann [19] who proposed a general method of how to simulate iterative learners by set-driven IIMs.

Theorem 3. *ItFTxt* $\subseteq$ *s-LimTxt*.

Proof. Let $\mathcal{X}$ be the relevant learning domain over which $\mathcal{C}$ is defined, and assume $\mathcal{C} \in$ *ItFTxt*. Then, there are an iterative IIM M and a hypothesis space $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ such that M $ItFTxt_{\mathcal{H}}$–identifies $\mathcal{C}$. For proving $\mathcal{C} \in$ *s-LimTxt*, we construct a suitable hypothesis space $\mathcal{H}' = (h'_j)_{j \in \mathbb{N}}$ as follows. Let $\mathcal{F} = (F_j)_{j \in \mathbb{N}}$ and $\#(F)$ be defined as in the demonstration of Theorem 2 above. Then, we define $h'_{2j} = h_j$ and $h'_{2j+1} = F_j$ for every $j \in \mathbb{N}$.

Subsequently, we use the following shorthands. Let S be any non-empty finite set $S \subseteq \mathcal{X}$ with $card(S) = n + 1$. We define $ref(S) = x_0, x_1, \ldots, x_n$ to be the repetition free enumeration of all the elements of S in lexicographical order. Furthermore, if $card(S) = 1$, we set $exh(S) = ref(S)$. Otherwise, we set $exh(S) = exh(S') \diamond ref(S)$, where $S' = S \setminus \{x\}$ and x is the lexicographically last element in S.

The desired set-driven IIM M' is defined as follows. Let $c \in \mathcal{C}$, let $t \in text(c)$, and let $n \in \mathbb{N}$.

IIM M': "On input t_n do the following:

[2] Note that, in Jain *et al.* [11], iterative learners are called memory-limited learners.

Determine $S = t_n^+$ and $exh(S)$. For all $x \in S$, test whether or not it is the case that $M_*(0, exh(S)) = M_*(0, exh(S) \diamond x)$.
In case it is, determine $j = M_*(0, exh(S))$, output $2j$, and request the next input. Otherwise, determine $z = \#(S)$, output $2z + 1$, and request the next input."

By definition, M' is set-driven. For showing that M' $LimTxt_{\mathcal{H}'}$–infers c when fed t, we distinguish the following cases.

Case 1. c is finite.

Then, there exists an $n \in \mathbb{N}$ with $t_n^+ = c$. It suffices to show that $c = h'_{M'(c)}$. If $M'(c) = 2z + 1$ with $z = \#(c)$, we are done, by construction. Otherwise, for all $x \in c$, we have $M_*(0, exh(c)) = M_*(0, exh(c) \diamond x)$. Let $j = M_*(0, exh(c))$. Hence, M converges to j when fed the fat text $exh(c) \diamond ref(c) \diamond ref(c) \diamond \cdots$ for c. Since M learns c, we are done.

Case 2. c is infinite.

Let $t^c = (x_j)_{j \in \mathbb{N}}$ be the lexicographically ordered text for c. Thus, $t^{exh} = x_0 \diamond x_0, x_1 \diamond x_0, x_1, x_2 \diamond \cdots$ is a fat text for c. Since M $ItFTxt_{\mathcal{H}}$–learns c from t^{exh}, there are $n_0, k \in \mathbb{N}$ such that $M_*(0, t_{n_0}^{exh}) = k$ and k is a locking hypothesis of M for c (cf. the verification of Observation 1). Now, let $\sigma = t_{n_0}^{exh}$. Finally, since $t \in text(c)$, there is an index m_0 such that $\sigma^+ \subseteq t_{m_0}^+$. Thus, σ constitutes a prefix of $exh(t_{m_0}^+)$, and hence $M'(t_m^+) = 2k$ for all $m \geq m_0$. Since, by definition, $h'_{2k} = h_k = c$, we are done. ∎

Furthermore, taking into consideration that $ItTxt \subset ConsvTxt$ (cf. Proposition 1), we may easily conclude:

Corollary 4.

(a) *$ItFTxt = ConsvTxt$.*
(b) *$ItTxt \subset ItFTxt$.*

Next, we show that intermediate hypotheses have to be used to reflect the progress made in the learning process. Without this option, iterative learners fail to exploit the additional information that is provided within fat texts.

In order to achieve the announced result, we start with a theorem that illuminates the structural properties of those concept classes that are It^vFTxt–learnable.

Theorem 5. *For all indexable classes $\mathcal{C}$: $\mathcal{C} \in It^vFTxt$ iff $\mathcal{C}$ is inclusion-free.*

Proof. First, suppose that an inclusion-free indexable class $\mathcal{C} = (c_j)_{j \in \mathbb{N}}$ is given. Select the hypothesis space $\mathcal{H} = (h_{\langle j,n \rangle})_{j,n \in \mathbb{N}}$ that meets, for all $j, n \in \mathbb{N}$,

$h_{\langle j,n\rangle} = c_j$. Then, the following iterative IIM M $\mathit{It}^v\mathit{FTxt}_{\mathcal{H}}$–identifies $\mathcal{C}$: For all $k \in \mathbb{N}$ and all possible input data x, let $M(k,x) = \mathit{min}\{j \mid j \geq k,\, x \in h_j\}$. To see this, note that, by the properties of $\mathcal{H}$ and $\mathcal{C}$, M can never output an overgeneral hypothesis, i.e., a hypothesis k' with $c \subset h_{k'}$. Since, by definition, M never rejects a correct hypothesis, one immediately sees that M converges to the least $k' \geq k$ with $c_{k'} = c$, where k is M's initial hypothesis.

Next, we show that $\mathit{It}^v\mathit{FTxt}$-identifiable classes must be inclusion-free. To see this assume, for a moment, that there is an indexable class $\mathcal{C} \in \mathit{It}^v\mathit{FTxt}$ that is not inclusion-free. Hence, there are an iterative IIM M and a hypothesis space $\mathcal{H}$ such that M $\mathit{It}^v\mathit{FTxt}_{\mathcal{H}}$–identifies $\mathcal{C}$. Let $c, c' \in \mathcal{C}$ with $c' \subset c$. By Observation 1, there is some locking hypothesis k of M for c. Now, let t' be any fat text for c'. Since $c' \subset c$ and since k is locking hypothesis of M for c, $M_*(k, t'_n) = k$ for all $n \in \mathbb{N}$, and therefore M fails to learn c' on t', if the initial hypothesis equals k. ∎

Theorem 6. *$\mathit{ItTxt} \# \mathit{It}^v\mathit{FTxt}$.*

Proof. Consider the class of all finite concepts $\mathcal{C}_{\mathit{fin}}$ over the given learning domain $\mathcal{X}$. Clearly, $\mathcal{C}_{\mathit{fin}} \in \mathit{ItTxt}$, but $\mathcal{C}_{\mathit{fin}}$ is not inclusion-free, and therefore, by Theorem 5, $\mathcal{C}_{\mathit{fin}} \notin \mathit{It}^v\mathit{FTxt}$. On the other hand, recall the definition of the indexable class $\mathcal{C}_{ex}$. That is, $\mathcal{C}_{ex}$ is the collection of all concepts $c_j = \{a\}^+ \setminus \{a^{j+1}\}$. Clearly, $\mathcal{C}_{ex}$ is inclusion-free, and thus, by Theorem 5, $\mathcal{C}_{ex} \in \mathit{It}^v\mathit{FTxt}$. Since $\mathcal{C}_{ex} \notin \mathit{ItTxt}$ (cf. the discussion at the beginning of Section 3), we are done. ∎

Furthermore, since, by definition, $\mathit{It}^v\mathit{Txt} \subseteq \mathit{It}^v\mathit{FTxt}$ and $\mathit{It}^v\mathit{Txt} \subseteq \mathit{ItTxt}$, we directly obtain:

Corollary 7.

(a) *$\mathit{It}^v\mathit{Txt} \subset \mathit{It}^v\mathit{FTxt}$.*
(b) *$\mathit{It}^v\mathit{Txt} \subset \mathit{ItTxt}$.*

Our next result puts the weakness of the learning type $\mathit{It}^v\mathit{Txt}$ into the right perspective.

Theorem 8. *$\mathit{FinTxt} \setminus \mathit{It}^v\mathit{Txt} \neq \emptyset$.*

Proof. Let $\mathcal{C}$ be the indexable class that contains exactly all $c \subseteq \{a\}^*$ with $\mathit{card}(c) = 2$. Obviously, $\mathcal{C} \in \mathit{FinTxt}$. On the other hand, even the simple subclass $\mathcal{C}'$ that contains the concepts $\{a, a^2\}$, $\{a, a^3\}$, and $\{a^2, a^3\}$ does not belong to $\mathit{It}^v\mathit{Txt}$. To see this, suppose that there are an iterative IIM M and a hypothesis space $\mathcal{H}$ such that M $\mathit{It}^v\mathit{Txt}_{\mathcal{H}}$–identifies $\mathcal{C}'$. Let k be some locking hypothesis of M for $\{a^2, a^3\}$. By Observation 1, such k must exist. Thus, M, when starting with the initial hypothesis k, outputs exactly the same sequence of hypotheses when fed the text $t = a^2, a, a, \ldots$ for $\{a, a^2\}$ and the text $t' = a^3, a, a, \ldots$ for $\{a, a^3\}$. Thus, M must fail to learn at least one of both concepts, a contradiction. ∎

However, *It*^{v}*Txt* may outperform *FinTxt*, as well.

Theorem 9. *FinTxt* $\#$ *It*^{v}*Txt*.

Proof. By Theorem 8, it remains to show that *It*^{v}*Txt*$\setminus$*FinTxt* $\neq \emptyset$. A separating class $\mathcal{C}$ will be defined as follows.

We let $\mathcal{X} = \{a, b\}^+$ be the learning domain. Let $j \in \mathbb{N}$. If $\varphi_j(j) \uparrow$, we set $c_{2j} = c_{2j+1} = \{a^j b\}$. If $\varphi_j(j) \downarrow$, there is a $y \in \mathbb{N}$ with $y = \Phi_j(j)$ and we set $c_{2j} = \{a^j b, a^j b^{y+100}\}$ and $c_{2j+1} = \{a^j b, a^j b^{y+200}\}$. Finally, let $\mathcal{C}$ be the collection of all concepts c_{2j} and c_{2j+1}.

Clearly, $\mathcal{C}$ constitutes an indexable class. Moreover, the following IIM M obviously *It*^{v}*Txt*$_{\mathcal{H}}$–identifies $\mathcal{C}$, where $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ with $h_j = c_j$ for all $j \in \mathbb{N}$. For all $k \in \mathbb{N}$ and all $x \in \mathcal{X}$, $M(k, x) = k$, if $x \in c_k$, and $M(k, x) = min\{j \mid x \in h_j\}$, otherwise.

Next, we verify that $\mathcal{C} \notin$ *FinTxt*. Suppose to the contrary that there are a hypothesis space $\mathcal{H}$ and an IIM M that *FinTxt*$_{\mathcal{H}}$–identifies $\mathcal{C}$. Based on M, we define a decision procedure P that solves the halting problem.

Procedure P: "On input $j \in \mathbb{N}$ proceed as follows:
Set $z = 0$ and execute instruction (A).
(A) Test whether or not (i) $\Phi_j(j) \leq z$ or (ii) M on input $t_z = \underbrace{a^j b, \ldots, a^j b}_{(z+1)-\text{times}}$, outputs a hypothesis $k \in \mathbb{N}$.
If (i) happens, output '$\varphi_j(j) \downarrow$.' If (ii) happens, output '$\varphi_j(j) \uparrow$.' Otherwise, i.e., neither (i) nor (ii) happens, set $z = z + 1$ and execute instruction (A)."

It remains to show that P decides the halting problem. Let $j \in \mathbb{N}$. We distinguish the following cases.

Case 1. $\varphi_j(j) \uparrow$.

Then, $t = a^j b, a^j b, \ldots$ constitutes a text for c_{2j}. Since, by assumption, M learns c_{2j} on t, (ii) eventually happens, and thus P outputs '$\varphi_j(j) \uparrow$.'

Case 2. $\varphi_j(j) \downarrow$.

Hence, there is a $y \in \mathbb{N}$ such that $y = \Phi_j(j)$, and therefore P must terminate. Now, suppose that P outputs '$\varphi_j(j) \uparrow$.' Hence, (ii) happened. Because of $a^j b \in c_{2j} \cap c_{2j+1}$, one can easily construct a text for c_{2j} and a text for c_{2j+1} on which M converges to the same final hypothesis. Since $c_{2j} \neq c_{2j+1}$, this would contradict our assumption that M learns both concepts. Hence, (ii) cannot happen, and thus P's output must be correct. ∎

It is quite obvious that FinTxt cannot contain any indexable concept class $\mathcal{C}$ that contains two distinctive concept c, c' with $c \subset c'$. Hence, we may conclude:

Corollary 10. *$\mathit{FinTxt} \subset \mathit{It}^v\mathit{FTxt}$.*

Figure 1 displays the achieved separations and coincidences of the considered learning types. Each learning type is represented as a vertex in a directed graph. A directed edge (or path) from vertex A to vertex B indicates that A is a proper subset of B, and no edge (or path) between these vertices imply that A and B are incomparable.

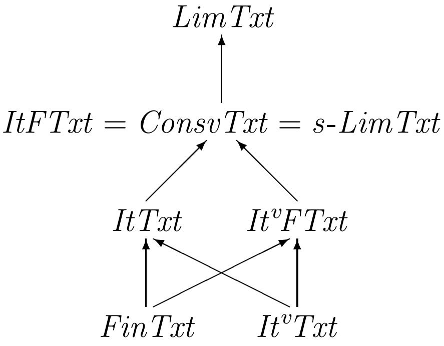

Fig. 1. The relations of iterative learning from positive data

4 Iterative learning from positive and negative data

Next, we study iterative learning from positive and negative data. Thus, we have to introduce some more notations and definitions.

Let $\mathcal{X}$ be the underlying learning domain, let $c \subseteq \mathcal{X}$ be a concept, and let $i = ((x_n, b_n))_{n \in \mathbb{N}}$ be any sequence of elements of $\mathcal{X} \times \{+, -\}$ such that $\{x_n \mid n \in \mathbb{N}\} = \mathcal{X}$, $\{x_n \mid n \in \mathbb{N}, b_n = +\} = c$ and $\{x_n \mid n \in \mathbb{N}, b_n = -\} = \mathcal{X} \setminus c = \overline{c}$. Then, we refer to i as an informant. By $\mathit{info}(c)$ we denote the set of all informants for c and by $\mathit{finfo}(c)$ the set of all fat informants for c, i.e., informants having the property that, for all $x \in \mathcal{X}$, there are infinitely many $n \in \mathbb{N}$ with $x_n = x$ (cf. Jain *et al.* [11]). (Note that, by definition, $\mathit{finfo}(c) \subseteq \mathit{info}(c)$.) We use i_y to denote the initial segment of i of length $y+1$, and define $i_y^+ = \{x_n \mid n \leq y,\, b_n = +\}$ and $i_y^- = \{x_n \mid n \leq y,\, b_n = -\}$.

Furthermore, let $c \subseteq \mathcal{X}$, and let $(x, b) \in \mathcal{X} \times \{+, -\}$. Then, c is said to be consistent with (x, b), which we denote by $\mathit{cons}(c, (x, b))$, provided that $x \in c$, if $b = +$, and $x \notin c$, otherwise.

The learning models *LimInf* and *FinInf* are defined analogously as their text counterparts by replacing text by informant. Finally, we extend the definitions of all variants of iterative learning in the same way, and denote the resulting learning types by *ItInf*, *It*^{v}*Inf*, *ItFInf*, and *It*^{v}*FInf*, respectively.

As in the previous section, we first summarize the known results (cf. Lange and Zeugmann [17]).

Proposition 2. *FinInf* $\subset$ *ItInf* $\subset$ *LimInf*.

In contrast to the text case, iterative learning from fat positive and negative data is at least as powerful as learning in the limit from informant. This add-on in learning power can also be observed, if iterative learners have to be successful no matter which initial hypothesis has been selected.

Theorem 11. *For all indexable classes* $\mathcal{C}$*:* $\mathcal{C} \in$ *It*^{v}*FInf*.

Proof. Let $\mathcal{C} = (c_j)_{j\in\mathbb{N}}$ be an indexable concept class. Select the hypothesis space $\mathcal{H} = (h_{\langle j,n\rangle})_{j,n\in\mathbb{N}}$ that meets, for all $j, n \in \mathbb{N}$, $h_{\langle j,n\rangle} = c_j$. The required iterative IIM M is defined as follows. For all $k \in \mathbb{N}$ and all input data $(x, b) \in \mathcal{X} \times \{+, -\}$, $M(k, (x, b)) = min\{j |\ j \geq k,\ cons(h_j, (x, b))\}$.

Since M implements the identification by enumeration principle (cf. Gold [9]), one directly sees that M, when fed any fat informant for some $c \in \mathcal{C}$, converges to the least $j \geq k$ that meets $h_j = c$, where k is M's initial hypothesis. Hence, M *It*^{v}*FInf*–identifies $\mathcal{C}$. ∎

Finally, since, by definition, *It*^{v}*FInf* $\subseteq$ *ItFInf* and since every indexable concept class belongs to *LimInf* (cf. Gold [9]), we can conclude:

Corollary 12. *It*^{v}*FInf* $=$ *ItFInf* $=$ *LimInf*.

The picture changes drastically, if iterative learning from arbitrary informants is considered. However, in contrast to the text case, *It*^{v}*Inf* contains relatively rich concept classes.

Observation 13. $\mathcal{C}_{\mathit{fin}} \in$ *It*^{v}*Inf*.

Proof. As in the proof of Theorem 2, let $\mathcal{F} = (F_j)_{j\in\mathbb{N}}$ denote any repetition free enumeration of all finite subsets of the learning domain $\mathcal{X}$ and assume any effective procedure computing, for every finite set $F \subseteq \mathcal{X}$, its uniquely determined index $\#(F)$ in $\mathcal{F}$. We choose $\mathcal{F}$ as hypothesis space and define the needed iterative learner M as follows. Let $k \in \mathbb{N}$ and (x, b) be given. Then, we let $M(k, (x, b)) = \#(F_k \cup \{x\})$, if $b = +$, and $M(k, (x, b)) = \#(F_k \setminus \{x\})$, if $b = -$.

We next verify that M learns as required. So, let $c \in \mathcal{C}$, let i be an informant for c, and let k be M's initial hypothesis. Now, let $S = (F_k \setminus c) \cup (c \setminus F_k)$. Clearly, S is finite. Now, by definition, if M receives an element from S, it performs a mind change. Moreover, every of M's mind changes reduces the cardinality of S, and therefore M converges to a correct hypothesis for c. ∎

If the target concept class contains finite and infinite concepts, it might be inevitable to select the initial hypothesis appropriately. To see this, let $\mathcal{C}_s$ be the indexable class that contains the concept $c = \{a\}^+$ and all singleton concepts $c_j = \{a^{j+1}\}$ over the learning domain $\mathcal{X} = \{a\}^+$.

Observation 14. *$\mathcal{C}_s \notin It^v Inf$.*

Proof. Suppose to the contrary that there are an iterative learner M and a hypothesis space $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$ such that M $It^v Inf_{\mathcal{H}}$–identifies $\mathcal{C}_s$. Since M, in particular, learns c, there has to be some locking hypothesis k of M for c, and thus, for all $j \in \mathbb{N}$, $M(k, (a^j, +)) = k$. (Note that, in the informant case, the analogue of Observation 1 holds, too.) Next, consider the sequence of hypotheses $(M_n(k, i))_{n \in \mathbb{N}}$ generated by M when successively processing the lexicographically ordered informant $i = (a, +), (a^2, -), (a^3, -), \ldots$ of the concept c_0. Since M has to infer c_0, there have to be $j, z \in \mathbb{N}$ such that $h_j = c_0$, $M_z(k, i) = j$, and $M(j, (a^m, -)) = j$ for all $m \geq z$. Since k is locking hypothesis of M for c, z is greater than 0. Now, fix any $m \geq z + 1$, set $\hat{\imath} = (a^m, +) \diamond (a^2, -), \ldots, (a^{m-1}, -) \diamond (a^{m+1}, -) \diamond (a, -) \diamond (a^{m+2}, -), (a^{m+3}, -), \ldots$ and $\tilde{\imath} = (a^{m+1}, +) \diamond (a^2, -), \ldots, (a^{m-1}, -) \diamond (a^m, -) \diamond (a, -) \diamond (a^{m+2}, -), (a^{m+3}, -), \ldots$ By definition, $\hat{\imath} \in info(c_{m-1})$ and $\tilde{\imath} \in info(c_m)$. By the properties of k and by the choice of $\hat{\imath}$ and $\tilde{\imath}$, one immediately sees that, for all $n \in \mathbb{N}$, $M_n(k, \hat{\imath}) = M_n(k, \tilde{\imath})$, and thus M fails to infer at least one of both concepts, a contradiction. ∎

The proof idea presented above can easily be adapted to show that $\mathcal{C}_{pat}$ [3], the well-known claas of all pattern languages, does not belong to $It^v Inf$. Pattern languages, as introduced in Angluin [1], are of particular interest, since pattern language learning algorithms have found interesting applications in different areas including molecular biology (cf., e.g., Shinohara and Arikawa [24]).

Corollary 15. *$\mathcal{C}_{pat} \notin It^v Inf$.*

Furthermore, it is well-known that $\mathcal{C}_{pat} \in FinInf$ as well as $\mathcal{C}_{fin} \notin FinInf$ (cf., e.g., Zeugmann and Lange [30]). Hence, we may conclude:

[3] Let Σ be a non-empty finite alphabet of symbols and let X be an infinite set of variables such that $\Sigma \cap X = \emptyset$. Then, every non-empty word in $(\Sigma \cup X)^*$ constitutes a pattern. The language $L(p)$ defined by a pattern p is the set of all strings that can be obtained by replacing all variables in p by non-empty strings from Σ^*. Thereby, each occurrence of a variable has to be replaced by the same string. Now, $\mathcal{C}_{pat}$ is the set of all languages L for which there is a pattern p such that $L = L(p)$. Note that $\mathcal{C}_{pat}$ contains all singleton languages as well as Σ^+.

Theorem 16. *$\mathit{FinInf} \;\#\; \mathit{It}^v\mathit{Inf}$.*

Since $\mathit{FinInf} \subset \mathit{ItInf}$ (cf. Proposition 2) and $\mathit{It}^v\mathit{Inf} \subseteq \mathit{ItInf}$, we obtain the missing part in the picture for the informant case.

Corollary 17. *$\mathit{It}^v\mathit{Inf} \subset \mathit{ItInf}$.*

Figure 2 summarizes the established relations of the considered learning types for the informant case. The semantics is analogous to that of Figure 1.

$$\begin{tikzcd}
& \mathit{LimInf} = \mathit{ItFInf} = \mathit{It}^v\mathit{FInf} & \\
& \mathit{ItInf} \arrow[u] & \\
\mathit{FinInf} \arrow[ur] & & \mathit{It}^v\mathit{Inf} \arrow[ul]
\end{tikzcd}$$

Fig. 2. The relations of iterative learning from positive and negative data

5 Conclusions

Gold's [9] model of concept learning in the limit relies on the assumption that, at every learning stage, the learner has access to all input data about a target concept seen so far. Since each practical learning system has to deal with space limitations, it is unrealistic to assume that an algorithmic learner processes samples of growing size. Models of incremental learning refine Gold's [9] model in that they considerably restrict the accessibility of the input data. Incremental learning has formally been studied by several authors including Wiehagen [29], Jantke and Beick [12], Fulk *et al.* [6], Kinber and Stephan [13], Lange and Zeugmann [19], Case *et al.* [4], and Jain *et al.* [11]. Their studies rigorously proved that, in general, limitations in the accessibility of the input data result in a remarkable loss of learning power.

In order to model learning scenarios that are typical for several approaches to case-based reasoning (cf., e.g., Kolodner [14]), we studied two new models of incremental learning – called iterative learning from fat information sequences and iterative learning with arbitrary initial hypotheses. The theoretical results obtained allow for the following interpretation.

Limitations in the accessibility of the input data are not that relevant, if it is *a priori* known that an iterative learner will receive every relevant data item infinitely often. When learning from positive data is concerned, this *a priori* knowledge enables iterative learners to become exactly as powerful as conservative IIMs which themselves are less powerful than unconstrained learners. In case that positive and negative data are available, now the learning capabilities of iterative learners and unconstrained IIMs coincide.

Moreover, the strength of iterative learners heavily depend on their ability to encode additional information in their intermediate hypotheses. Iterative learner that do not have this option are extremely weak. Even finite learners, which are themselves very restrictive, may outperform iterative learners that are supposed to learn no matter which initial hypothesis is actually chosen. This results is valid in case that positive data or positive and negative data are available. However, in the latter case, the situation changes completely if learning from fat information sequences is considered. If it is *a priori* known that an iterative learner will receive every positive and every negative example infinitely often, there is no need to encode any additional information in its intermediate hypotheses.

Recently, the problem of how iterative learners are able to cope with noise in the input data sequence has systematically been investigated (cf. Lange and Grieser [16]). It turned out that an indexable class can be iteratively identified from noisy text[4] if and only if it is inclusion-free. Comparing this equivalence with Theorem 5, one arives at the following insight: On the one hand, iterative learners which can successfully handle noise in the input data sequence do not need to encode any additional information in their intermediate hypotheses. On the other hand, if an iterative learner performs well no matter which initial hypothesis is actually chosen, it can successfully handle noise in the input data.

References

[1] Angluin, D., Finding patterns common to a set of strings, *Journal of Computer and System Sciences* **21**, 46–62, 1980.

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

[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] Ers̃ov, Yu.L., “Theory of Numberings,” Nauka, Moscow, 1977 (in Russian).

[6] Fulk, M., Jain, S., and Osherson, D.N., Open problems in systems that learn, *Journal of Computer and System Sciences* **49**, 589–604, 1994.

[4] As in Stephan [25], an infinite sequence t of elements from the learning domain $\mathcal{X}$ is said to be a *noisy text* for a concept $c \subseteq \mathcal{X}$ iff t is a fat text for c that may, in addition, contain finitely many elements not belonging to c.

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

[8] Godin, R., and Missaoui, R., An incremental concept formation approach for learning from databases, *Theoretical Computer Science* **133**, 387–419, 1994.

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

[10] Hopcroft, J.E., and Ullman, J.D., "Formal Languages and their Relation to Automata," Addison-Wesley, 1969.

[11] Jain S., Osherson, D.N., Royer, J., and Sharma, A., "Systems that Learn - An Introduction to Learning Theory, 2nd Edition," MIT Press, 1999.

[12] Jantke, K.P., and Beick, H.R., Combining postulates of naturalness in inductive inference, *Journal of Information Processing and Cybernetics (EIK)* **17**, 465–484, 1981.

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

[14] Kolodner, J.K., An introduction to case-based reasoning, *Artificial Intelligence Review* **6**, 3–34, 1992.

[15] Lange, S., "Algorithmic Learning of Recursive Languages," Mensch & Buch Verlag Berlin, 2000.

[16] Lange, S., and Grieser, G., On the strength of incremental learning, *in* "Proceedings 10th International Workshop on Algorithmic Learning Theory," Lecture Notes in Artificial Intelligence 1720, pp. 118–131, Springer-Verlag, 1999.

[17] Lange, S., and Zeugmann, T., Types of monotonic language learning and their characterization, *in* "Proceedings 5th Annual ACM Workshop on Computational Learning Theory," pp. 377–390, ACM Press, 1992.

[18] Lange, S., and Zeugmann, T., Language learning in dependence on the space of hypotheses, *in* "Proceedings 6th Annual ACM Conference on Computational Learning Theory," pp. 127–136, ACM Press, 1993.

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

[20] Lange, S., and Zeugmann, T., Set-driven and rearrangement-independent learning of recursive languages, *Mathematical Systems Theory* **29**, 599–634, 1996.

[21] Maloof, M.A., and Michalski, R.S., Selecting examples for partial memory learning, *Machine Learning* **41**, 27–52, 2000.

[22] Porat, S., and Feldman, J.A., Learning automata from ordered examples, *Machine Learning* **7**, 109–138, 1991.

[23] Rivest, R., Learning decision lists, *Machine Learning* **2**, 229–246, 1987.

[24] Shinohara, T., and Arikawa, S., Pattern inference, *in* "Algorithmic Learning for Knowledge-Based Systems," Lecture Notes in Artificial Intelligence, Vol. 961, pp. 259–291, Springer-Verlag, 1995.

[25] Stephan, F., Noisy inference and oracles, *Theoretical Computer Science* **185**, 129–157, 1997.

[26] Utgoff, P.E., Incremental induction of decision trees, *Machine Learning* **4**, 161–186, 1989.

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

[28] Wexler, K., and Culicover, P., "Formal Principles of Language Acquisition," MIT Press, 1980.

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

[30] 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.