Formal models of incremental learning and their analysis

Steffen Lange
Deutsches Forschungszentrum für
Künstliche Intelligenz, Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
Email: lange@dfki.de

Sandra Zilles
Universität Kaiserslautern
FB Informatik, Postfach 3049
67653 Kaiserslautern, Germany
Email: zilles@informatik.uni-kl.de

***Abstract*— We consider concept learning from examples. The learner receives – step by step – larger and larger initial segments of a sequence of examples describing an unknown target concept, processes these examples, and computes hypotheses. The learner is successful, if its hypotheses stabilize on a correct representation of the target concept. The underlying model is called *identification in the limit*.**

The present study concerns different versions of *incremental learning in the limit*. In contrast to the general case, now the learner has only limited access to the examples provided so far. In the special case of iterative learning, the learner builds its new hypotheses just on the basis of the current hypothesis and the next example, without having access to any of the other examples presented so far. In the case of bounded example-memory learning, the learner may in addition memorize up to an a priori fixed number of examples already presented.

Formal studies have shown that restricting the accessibility of the input data results in a loss of learning power, i. e. there are concept classes learnable in the limit, but not identifiable by any incremental learner at all. The present analysis aims at illustrating this phenomenon and giving insights into the structure of concept classes incremental learners can cope with. Examples of identifiable and non-identifiable classes are given; different learning models are compared to one another with respect to the competence of the corresponding learners.

I. Introduction

An often studied aspect of natural learning behaviour is the ability to learn from examples. There are different theoretical approaches of modelling and analyzing this aspect; some consider concept learning from only positive examples (i. e. examples matching / belonging to the concept), others consider concept learning from both positive and negative examples (counterexamples, i. e. examples not belonging to the concept). A learner is fed with these examples – one in each step of the learning process – and step by step returns hypotheses representing its guess concerning the unknown target concept. As soon as the learner stabilizes on a correct hypothesis, the concept is said to be identified from the given sequence of examples. A class of concepts is learned, if some learner identifies each concept in the class from each sequence of examples representing the whole concept. What makes learning so difficult in this perspective, is that the learner's guesses are always based on a finite amount of information, whereas in general infinitely many examples are needed to completely specify the concept.

Here we study learning of indexable classes of recursive concepts[1] based on Gold's [3] approach of identification in the limit.

In the initial perspective the learner is able to use all the information seen so far in each step of the learning process. We may understand the learner as a student writing down all the examples she reads on a notepad of infinite capacity. While the notes on her notepad may be changed in each learning step, her hypothesis on the unknown concept must stabilize on a fix correct guess. This approach of identification in the limit has been widely analyzed, cf. Gold [3] and Angluin [2], revealing universal learning methods. In particular, when learning from informant (that is, learning from positive and negative examples) is considered, each indexable class of recursive concepts is identifiable in the limit.

Now a quite natural question is whether this situation changes, if the resources are modified, such that the learner has to work in an *incremental* fashion. For illustration, imagine the student's notepad is limited in capacity, i. e. only a bounded number of input data can be memorized.

In the extreme case the student does not have any notepad at all; she only has her current hypothesis in mind. If a new example is presented, she may or may not revise it. Consequently, all information seen beforehand must be either reconstructable from the current hypothesis or it will simply be forgotten. We then speak of *iterative learning*, cf. Wiehagen [11]. Formally, the input of the learner no longer consists of all the information seen so far, but of the current hypothesis and a new piece of information. As it turns out, this approach is too restrictive to allow identification of all indexable classes of recursive concepts from informant, see Lange [6], in contrast to observations in the context of Gold's initial model.

In a more relaxed case the student actually has a notepad, but its capacity is restricted to store only a fixed number of examples. If a new example is presented, and the capacity of the notepad is not exhausted yet, she may memorize the current example. But if there is no space left, she may memorize the current example only in case she removes another example from the notepad. As in the usual scenario, hypotheses can be changed in each step. In addition to the new example, both the information on the notepad and the

[1] An indexable class of recursive concepts may be seen as an effective enumeration of concepts, such that there is a uniform procedure which can decide, for any index n and any element x of a fixed domain ("learning domain"), whether or not x belongs to n-th concept in the enumeration, cf. Angluin [2].

current hypothesis can be used to construct the new hypothesis. The corresponding learning model is called *bounded example-memory learning*, cf. Lange and Zeugmann [8]. Indeed, as soon as one example can be stored, more concept classes become learnable. Whether a further increase of the space bound results in an increase of learning power, depends on the type of data presented: for learning from positive examples only, every add-on in the capacity of the notepad yields an add-on in the learning power. In contrast to that, when both positive and negative examples are presented, learners using any bounded example-memory can be simulated by learners using an example-memory with a capacity suitable for storing one example. So in this case any further extension of the example-memory will definitely not increase the learning power.

All these results will be illustrated below. Moreover, we compare iterative learning from informant to the initial approach of learning "with notepads of infinite capacity" from text (i. e. from positive examples only), presenting classes learnable in one of these models, but not in the other. The corresponding observations give insights into the structure of concept classes learnable "without any notepad at all" or "with notepads of restricted capacity", i. e., using the results from Lange and Zeugmann [8] and Lange [6], we try to bring about an idea of what incrementally learnable classes as well as suitable learning methods look like.

II. Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ be the set of all natural numbers. If A is any set, then $card(A)$ denotes the cardinality of A. Let σ be any finite sequence and let τ be any possibly infinite sequence. Then $\sigma \diamond \tau$ denotes the concatenation of σ and τ.

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 $c \in \mathcal{C}$. $\mathcal{C}$ is called a *concept class* and c a *concept*. By *co-c* we denote the complement of c, i. e. *co-c* $= \mathcal{X} \setminus c$. Sometimes we will identify a concept c with its characteristic function, i. e. we let $c(x) = +$, if $x \in c$, and $c(x) = -$, otherwise.

We deal with the learnability of indexable concept classes with uniformly decidable membership defined as follows (cf. Angluin [2]). A class of non-empty concepts $\mathcal{C}$ is said to be an *indexable concept class with uniformly decidable membership* if 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 $x \in \mathcal{X}$, it holds $f(j, x) = +$, if $x \in c_j$, and $f(j, x) = -$, otherwise. We refer to indexable concept classes with uniformly decidable membership by the phrase *indexable classes*, for short.

For illustration 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. Then, for all $a \in \Sigma$ and for all $n \in \mathbb{N}$, $a^{n+1} = aa^n$, while, by convention, a^0 equals the empty string. Moreover, we let $\mathcal{X} = \Sigma^*$ be the learning domain. Subsets $L \subseteq \Sigma^*$ are also called languages (instead of concepts). Then the set of all context-sensitive languages, context-free languages, regular languages, and of all pattern languages form indexable classes (cf. Hopcroft and Ullman [4], 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 the 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. Valiant [10], Rivest [9]).

A. Gold-style language learning

Next, we provide notions and notations that are fundamental for Gold's [3] model of *identification in the limit*.

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 *text* for c. By $Text(c)$ and $TextSeg(c)$ we denote the set of all texts for c and of all initial segments of texts for c, respectively. Alternatively, let $i = ((x_n, b_n))_{n \in \mathbb{N}}$ be an infinite sequence of elements from $\mathcal{X} \times \{+, -\}$ such that $\{x_n \mid n \in \mathbb{N}\} = \mathcal{X}$, $\{x_n \mid n \in \mathbb{N}, b_n = +\} = c$, and $\{x_n \mid n \in \mathbb{N},\ b_n = -\} =$ *co-c*. Then we refer to i as an *informant* for c. By $Info(c)$ and $InfoSeg(c)$ we denote the set of all informants for c and of all initial segments of informants for c, respectively. Moreover, let t be a text, let i be an informant, and let y be a number. Then t_y and i_y denote the initial segments of t and i of length $y + 1$, respectively.

From now on, let $(w_n)_{n \in \mathbb{N}}$ be any fixed, repetition-free, effective enumeration of all elements in $\mathcal{X}$, for example the lexicographically ordered enumeration.

Let $\mathcal{C}$ be an indexable class. As in Gold [3], we define an *inductive inference machine* for $\mathcal{C}$ (*IIM* for $\mathcal{C}$, for short)[2] to be a total algorithmic mapping from $TextSeg(\mathcal{C})$ [$InfoSeg(\mathcal{C})$] to $\mathbb{N}$. Thus, an IIM, when processing an initial segment of a text [an informant] for some $c \in \mathcal{C}$, always returns a hypothesis, i. e. a number encoding a certain computer program.

The numbers output by an IIM are interpreted with respect to a suitably chosen *hypothesis space* $\mathcal{H} = (h_j)_{j \in \mathbb{N}}$. Since we exclusively deal with indexable classes $\mathcal{C}$, we always assume that $\mathcal{H}$ is also an indexing of some possibly larger indexable class of non-empty concepts. Hence, membership is uniformly decidable in $\mathcal{H}$, too. Formally speaking, we deal with class comprising learning (cf. Zeugmann and Lange [12]). An IIM returning some number j is construed to hypothesize h_j.

In the sequel a data sequence $\sigma = (d_n)_{n \in \mathbb{N}}$ for a target concept c is either a text $t = (x_n)_{n \in \mathbb{N}}$ or an informant $i = ((x_n, b_n))_{n \in \mathbb{N}}$ for c. By convention, for all $y \in \mathbb{N}$, σ_y denotes the initial segment t_y or i_y. For any finite initial segment σ_y, let $|\sigma_y|$ denote its length, i. e. $|\sigma_y| = y + 1$.

We define convergence of IIMs as usual. Let σ be given and let M be an IIM. The sequence $(M(\sigma_y))_{y \in \mathbb{N}}$ of M's hypotheses *converges* to a number j iff all but finitely many of its terms are equal to j.

Now we are ready to define *learning in the limit*.

[2] Whenever the target indexable class $\mathcal{C}$ is clear from the context, we suppress the term "for $\mathcal{C}$".

Definition 1 (Gold [3]) *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 $Lim\,Txt_{\mathcal{H}}$ $[LimInf_{\mathcal{H}}]$–identifies c iff, for every data sequence σ with $\sigma \in Text(c)$ $[\sigma \in Info(c)]$, there is some $j \in \mathbb{N}$ with $h_j = c$ such that the sequence $(M(\sigma_y))_{y\in\mathbb{N}}$ converges to j.*

Then M $Lim\,Txt_{\mathcal{H}}$ $[LimInf_{\mathcal{H}}]$–identifies $\mathcal{C}$ iff, for all $c' \in \mathcal{C}$, M $Lim\,Txt_{\mathcal{H}}$ $[LimInf_{\mathcal{H}}]$–identifies c'.

Finally, $Lim\,Txt$ $[LimInf]$ denotes the collection of all indexable classes $\mathcal{C}'$ for which there are a hypothesis space $\mathcal{H}' = (h'_j)_{j\in\mathbb{N}}$ and an IIM M' such that M' $Lim\,Txt_{\mathcal{H}'}$ $[LimInf_{\mathcal{H}'}]$–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 about c the IIM reaches its (unknown) point of convergence and it computes a correct and finite description of the target concept. This may be understood as a process of learning.

As the learner in the definition of $Lim\,Txt$ and $LimInf$ may always use all the information about the target concept known in the current learning step, these identification types correspond to the perspective of learning with notepads of infinite size.

B. Incremental Learning

Now, we formally define the different models of incremental learning.

An ordinary IIM M always has access to the whole history of the learning process, i. e. it computes its current guess on the basis of all the input data seen so far. In contrast, in an incremental learning process, the access to the history of provided examples is limited.

For example, an *iterative* IIM is only allowed to use its latest guess and the next data element in the data sequence σ. Conceptionally, an iterative IIM M defines a sequence $(M_n)_{n\in\mathbb{N}}$ of machines each of which takes as its input the output of its predecessor.

Definition 2 (Wiehagen [11]) *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 $It\,Txt_{\mathcal{H}}$ $[ItInf_{\mathcal{H}}]$–identifies c iff, for every data sequence $\sigma = (d_n)_{n\in\mathbb{N}}$ with $\sigma \in Text(c)$ $[\sigma \in Info(c)]$, the following conditions are fulfilled:*

(1) *for all $n \in \mathbb{N}$, $M_n(\sigma)$ is defined, where*
   (i) $M_0(\sigma) = M(-1, d_0)$[3],
   (ii) $M_{n+1}(\sigma) = M(M_n(\sigma), d_{n+1})$.

(2) *the sequence $(M_n(\sigma))_{n\in\mathbb{N}}$ converges to a number j with $h_j = c$.*

Furthermore, M $It\,Txt_{\mathcal{H}}$ $[ItInf_{\mathcal{H}}]$–identifies $\mathcal{C}$ iff, for each $c' \in \mathcal{C}$, M $It\,Txt_{\mathcal{H}}$ $[ItInf_{\mathcal{H}}]$–identifies c'.

The learning types $It\,Txt$ and $ItInf$ are defined analogously to Definition 1, where It is short for *iterative learning*. The idea of iterative learning, as defined here, agrees with our initial conception of a student learning from examples without using any notepad at all.

For the sake of simplicity, we introduce the following shorthand: for any data sequence σ and any $y \in \mathbb{N}$, we define $M_*(\sigma_y) = M_y(\sigma)$. That is, $M_*(\sigma_y)$ denotes the last hypothesis generated by M when processing the initial segment σ_y.

Next, we consider a natural relaxation of iterative learning, named *k-bounded example-memory* inference. Now, an IIM M is allowed to memorize at most k of the data elements seen so far in the learning process, where $k \in \mathbb{N}$ is fixed *a priori*. Again, M defines a sequence $(M_n)_{n\in\mathbb{N}}$ of machines each of which takes as input the output of its predecessor. A k-bounded example-memory IIM outputs a hypothesis along with the set of memorized data elements.

Definition 3 (Lange and Zeugmann [8]) *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. Moreover, let $k \in \mathbb{N}$. An IIM M $Bem_k\,Txt_{\mathcal{H}}$ $[Bem_k Inf_{\mathcal{H}}]$–identifies c iff, for every data sequence $\sigma = (d_n)_{n\in\mathbb{N}}$ with $\sigma \in Text(c)$ $[\sigma \in Info(c)]$, the following conditions are satisfied:*

(1) *for all $n \in \mathbb{N}$, $M_n(\sigma)$ is defined, where*
   (i) $M_0(\sigma) = M((-1, \emptyset), d_0) = (j_0, S_0)$ *with* $S_0 \subseteq \{d_0\}$ *and* $card(S_0) \leq k$,
   (ii) $M_{n+1}(\sigma) = M(M_n(\sigma), d_{n+1}) = (j_{n+1}, S_{n+1})$ *with* $S_{n+1} \subseteq S_n \cup \{d_{n+1}\}$ *and* $card(S_{n+1}) \leq k$.

(2) *for $(M_n(\sigma))_{n\in\mathbb{N}} = ((j_n, S_n))_{n\in\mathbb{N}}$ the sequence $(j_n)_{n\in\mathbb{N}}$ converges to a number j with $h_j = c$.*

Furthermore, M $Bem_k\,Txt_{\mathcal{H}}$ $[Bem_k Inf_{\mathcal{H}}]$–identifies $\mathcal{C}$ iff, for each $c' \in \mathcal{C}$, M $Bem_k\,Txt_{\mathcal{H}}$ $[Bem_k Inf_{\mathcal{H}}]$–identifies c'.

For every $k \in \mathbb{N}$, the learning types $Bem_k\,Txt$ and $Bem_k Inf$ are defined analogously with the customs above. By definition, $Bem_0\,Txt = It\,Txt$ and $Bem_0 Inf = ItInf$.

This perception coincides with our idea of a student using a notepad of restricted size. Of course, using a notepad of size 0, that is, a notepad on which the student cannot memorize anything, has the same effect as working without any notepad at all.

The following proposition is an immediate consequence of the definitions above.

Proposition 1 *Let $k \geq 1$. Then*

(1) $It\,Txt \subseteq Bem_k\,Txt \subseteq Bem_{k+1}\,Txt \subseteq Lim\,Txt$,
(2) $ItInf \subseteq Bem_k Inf \subseteq Bem_{k+1} Inf \subseteq LimInf$.

Which of these inclusions are proper inclusions, will be discussed below.

[3]The term -1 denotes an *a priori* fixed initial hypothesis. This hypothesis is used for technical reasons, only. We adopt this convention to the other definitions below.

III. Results

We concentrate on incremental learning of indexable classes, the corresponding learnable concept classes, and appropriate learning methods.

First note that any indexable class is learnable in the limit, i. e. "with a notepad of infinite capacity", provided both positive and negative examples are presented. There is even a universal learning method in this context: assume an indexable class $\mathcal{C}$, given by the enumeration $(c_j)_{j\in\mathbb{N}}$, must be identified. Any new example will first be added to the list of examples stored in the notepad. Then the learner returns as its current hypothesis the least index j, such that c_j agrees with the information memorized in the notepad. This is possible, since membership is uniformly decidable. Then the sequence of hypotheses will converge to the minimal index of the unknown concept in the enumeration $(c_j)_{j\in\mathbb{N}}$. This learning method has been introduced by Gold [3] and is known as *identification by enumeration*.

Now what happens, if the learner does not have any notepad at all? A natural approach to adapt the method of identification by enumeration is the following: given the current hypothesis k and the new example (x, b), return the minimal index $j \geq k$, such that c_j agrees with (x, b). It is not hard to verify that a corresponding sequence of hypotheses converges, because the learner never returns an index larger than the minimal index of the target concept in the given enumeration. The trouble is, that finitely many errors may occur in the final hypothesis. So this variant of identification by enumeration is in general not suitable for iterative learning from informant.

Now it is conceivable, that the reason for the deficiency of this method is that it is just too simple, i. e. maybe for iterative learning of indexable classes in general more complex methods are needed. But, as we will see in Theorem 1 below, there is an indexable class which cannot be identified iteratively from informant. That means, no matter which iterative learner is chosen, it fails for at least one concept in the class. Hence, the deficiency of the principle of identification by enumeration for iterative learning from positive and negative examples cannot be compensated by any other learning strategy.

In order to verify this, we need the following simple, but very important observation: imagine an IIM M learns a concept c iteratively from informant. Moreover, let τ be any initial segment of some informant for c. Then there must be a finite extension σ, such that (i) $\tau \diamond \sigma$ forms an initial segment of some informant for c, (ii) $h = M_*(\tau \diamond \sigma)$ is a correct hypothesis for c, and (iii) $M(h, (x, c(x)))$ equals h for all x in the learning domain! Otherwise it would be possible to construct an informant for c on which M does not converge; we omit the details. Such a finite segment $\tau \diamond \sigma$ is called a *locking sequence* for M and c.

Note that, as soon as a locking sequence for a target concept c has been processed, a stage is reached, in which the learner has no chance to determine, for all but finitely many examples $(x, c(x))$, whether or not they have been presented yet.

Theorem 1 *Let $\mathcal{C}$ be a concept class containing all finite concepts and at least one infinite concept. Then $\mathcal{C} \notin ItInf$.*

Proof. We only sketch the proof for the special case of a singleton alphabet $\{a\}$. Imagine the concept class $\mathcal{C}_{\text{weak}}$ consists of the infinite concept $\{a\}^*$ of all strings over our alphabet as well as all finite subsets of $\{a\}^*$. Assume some IIM M learns $\mathcal{C}_{\text{weak}}$ iteratively from informant. Let τ be a locking sequence for M and $\{a\}^*$. τ contains only positive examples and even only finitely many, say $(a^{p_0}, +), (a^{p_1}, +), \ldots, (a^{p_k}, +)$. Now there is also a locking sequence $\tau \diamond \sigma$ for M and the concept $c = \{a^{p_0}, a^{p_1}, \ldots, a^{p_k}\}$ (note that c is finite and thus it belongs to $\mathcal{C}_{\text{weak}}$). σ contains only positive examples from c and finitely many negative examples, say $(a^{n_0}, -), (a^{n_1}, -), \ldots, (a^{n_l}, -)$. Now let $m > max\,(p_0, \ldots, p_k, n_0, \ldots, n_l)$, i. e. neither the positive example $(a^m, +)$ nor the negative example $(a^m, -)$ occurs in $\tau \diamond \sigma$. Then $M_*(\tau \diamond (a^m, +)) = M_*(\tau)$ and thus $M_*(\tau \diamond (a^m, +) \diamond \sigma) = M_*(\tau \diamond \sigma)$ by choice of τ. As $\tau \diamond \sigma$ is a locking sequence for M and c, there is an informant $\tau \diamond (a^m, +) \diamond \sigma \diamond \sigma'$ for $c' = c \cup \{a^m\}$, on which M converges to a hypothesis for c. So M does not learn c' from informant, although c' belongs to $\mathcal{C}_{\text{weak}}$, a contradiction, and thus we are done. □

As all indexable classes are learnable in the limit from informant, this justifies the following corollary.

Corollary 2 $ItInf \subset LimInf$.

How can we overcome the weakness of iterative learners? Let us analyze this problem for the concept class $\mathcal{C}_{\text{weak}}$ in the proof of Theorem 1: assume the learner was allowed to use extra memory to store the maximal data element $(x_{\max}, +)$ seen so far. In other words, we consider a student using a notepad with restricted space sufficient for memorizing just one example. This kind of extra information now helps the learner to adapt its hypothesis appropriately.

Suppose the learner M is fed with an informant for any target concept in our concept class $\mathcal{C}_{\text{weak}}$. As long as only positive examples are shown, let M return an index of the concept $\{a\}^*$; in each step the currently memorized example $(x_{\max}, +)$ is adjusted, whenever a longer example is presented. If the first negative example $(x, -)$ appears, let M output an index for $\{a^z \mid z \leq |x_{\max}|\} \setminus \{x\}$. Obviously, this hypothesis corresponds to a finite variant of the target concept. Hence, at most finitely many corrections are needed to come up with a correct final guess. Fortunately, all the relevant data eventually appear in subsequent steps: if some a^z, $z \leq |x_{\max}|$ does *not* belong to the target concept, the corresponding example still has to appear in the current informant. So a 1-bounded example-memory suffices to identify our concept class from informant.

As the concept class $\mathcal{C}_{\text{weak}}$ is not identifiable iteratively from informant, this verifies the corollary below.

Corollary 3 $ItInf \subset Bem_1Inf$.

Thus, having the ability to memorize one data element increases the learning power. But what happens, if the incremental learner is allowed to store two, three, or even four

data elements? Will this result in a further add-on in learning power? Surprisingly, not. As it turned out, an example-memory of size one is already sufficient for identification of *any* indexable class, if both positive and negative examples are available, i. e. Bem_1Inf and $LimInf$ coincide.

Theorem 4 $Bem_1Inf = LimInf$.

Proof. We only sketch the idea of the corresponding proof: it is similar to the idea explained for our example below Theorem 1. In each step, the learner uses its example-memory in order to store from its input the particular element which has a maximal index in our fixed enumeration $(w_m)_{m\in\mathbb{N}}$. If the new information presented agrees with the latest hypothesis, M will hypothesize the same concept again. Otherwise, let (w_m, b) be the element stored in the example-memory. Then let M look for the minimal index j agreeing with the new information as well as with the latest hypothesis respecting the first $m+1$ elements $w_0, \ldots, w_m$. Since it might happen that such an index does not exist, the search must be bounded, say by m. Thus, if this bounded search is not successful, M simply returns some auxiliary hypothesis representing the least finite concept matching the demands j is supposed to meet. In case the bounded search *is* successful, M returns the index j.

Verifying the correctness of M is a little bit more involved. Therefore the relevant details are omitted. □

Still, analyzing the impact of a k-bounded example-memory in general makes sense, because, in contrast, when learning from only positive data is concerned, an infinite hierarchy of more and more powerful bounded-example-memory learners has been revealed. The proof is omitted, see Lange and Zeugmann [8] for details.

Theorem 5 *For all* $k \geq 1$*:* $ItTxt \subset Bem_kTxt \subset Bem_{k+1}Txt \subset LimTxt$.

Finally, we concentrate on gaining a better understanding of the real learning power of iterative machines.

Interestingly, the indexable class used to illustrate the statement of Theorem 1 is neither identifiable iteratively from informant nor identifiable in the limit from text. Even more: it seems that the structural complexity making this class so hard to learn meets a specific conceptional deficiency which $ItInf$-learners and $LimTxt$-learners have in common. So one might conjecture that iterative learners are under no circumstances able to exploit the additional information provided by the negative examples. The following theorem states, that this is not the case. Indeed there are indexable classes learnable iteratively from informant, but not learnable in the limit from text.

Theorem 6 $ItInf \setminus LimTxt \neq \emptyset$.

Proof. This is witnessed by a quite simple concept class over the singleton alphabet $\{a\}$, namely the class consisting of $c = \{a\}^*$ and all concepts $c_k = c \setminus \{a^k\}$ for $k \in \mathbb{N}$. An easy argument using the idea of a locking sequence in the context of text-learning shows that this concept class does not belong to $LimTxt$. To verify that the concept class is learnable iteratively from informant, let an IIM M guess c, as long as only positive examples are presented. The first negative example $(a^k, -)$ makes M hypothesize the concept c_k, a hypothesis that will never be changed afterwards. Obviously, this hypothesis must be correct. □

Still, this does not imply, that the profit resulting from negative data always outperforms the use of a notepad. Maybe iterative learners cannot always exploit the additional information which negative data provide. Actually, there are concept classes, which are on the one hand appropriate for identification in the limit from positive examples only, but on the other hand too complex for iterative identification, even if negative data are presented. That means, it is not always possible to simulate a $LimTxt$-learner without using any notepad at all, even if the information in the infinite learning process is completed by adding the formerly missing negative examples. Consequently, $LimTxt$ is not a subset of $ItInf$, and thus both identification models are incomparable.

Theorem 7 $LimTxt \setminus ItInf \neq \emptyset$.

Proof. We consider the alphabet $\{a, b\}$ and the concept class $\mathcal{C}$ consisting of the infinite concept $\{a\}^*$ as well as all finite concepts containing exactly one element from $\{b\}^*$ plus finitely many elements of $\{a\}^*$.

Firstly, note that $\mathcal{C}$ is identifiable in the limit from text: an appropriate IIM only has to guess the concept $\{a\}^*$, until an element from $\{b\}^*$ appears in the text. Afterwards M always hypothesizes the finite concept consisting of exactly all examples seen so far. As the target concept in the latter case must be finite, this method is successful, i. e. the sequence of hypotheses converges to a correct guess.

Secondly, we have to verify that $\mathcal{C}$ is not $ItInf$-identifiable. For that purpose assume some IIM M learns $\mathcal{C}$ iteratively from informant. Let τ be a locking sequence for M and $\{a\}^*$. τ contains only finitely many examples: negative examples from $\{b\}^*$ and positive examples from $\{a\}^*$, say $(a^{p_0}, +), (a^{p_1}, +), \ldots, (a^{p_k}, +)$. Let $(b^z, -)$ be a negative example, which does not appear in τ. Now there is also a locking sequence $\tau \diamond (b^z, +) \diamond \sigma$ for M and the concept $c = \{a^{p_0}, a^{p_1}, \ldots, a^{p_k}\} \cup \{b^z\}$ (note that c is finite and thus it belongs to $\mathcal{C}$). σ contains only positive examples from c and finitely many negative examples, say $(a^{n_0}, -), (a^{n_1}, -), \ldots, (a^{n_l}, -)$. Now let $m > max\,(p_0, \ldots, p_k, n_0, \ldots, n_l)$, i. e. neither the positive example $(a^m, +)$ nor the negative example $(a^m, -)$ occurs in $\tau \diamond (b^z, +) \diamond \sigma$. Then $M_*(\tau \diamond (a^m, +)) = M_*(\tau)$ and thus $M_*(\tau \diamond (a^m, +) \diamond (b^z, +) \diamond \sigma) = M_*(\tau \diamond (b^z, +) \diamond \sigma)$ by choice of τ. As $\tau \diamond (b^z, +) \diamond \sigma$ is a locking sequence for M and c, there is an informant $\tau \diamond (a^m, +) \diamond (b^z, +) \diamond \sigma \diamond \sigma'$ for $c' = c \cup \{a^m\}$, on which M converges to a hypothesis for c. So M does not learn c' from informant, although c' belongs to $\mathcal{C}$. This contradiction proves the claim. □

So we know that the lack of notepads of infinite size cannot be compensated by negative information additionally presented to the learner. Still the question remains, whether

negative examples provide enough additional information to make any *bounded* notepad superfluous. Even this question must be answered in the negative: there are indexable classes, which cannot be learned iteratively from informant, but a 1-bounded "notepad" even suffices to identify these classes in the limit from text, i. e. from positive examples only.

Theorem 8 $Bem_1\, Txt \setminus ItInf \neq \emptyset$.

Proof. We consider the alphabet $\{a, b\}$ and the concept class $\mathcal{C}$ consisting of the infinite concept $\{a\}^*$ as well as all finite concepts containing exactly one element b^z from $\{b\}^*$ together with all elements a^m with $m \leq z$ and exactly one element a^n with $n > m$.

Applying similar locking sequence arguments as above, one easily sees that there is no iterative learner able to identify all $c \in \mathcal{C}$ from informant. On the other hand, there is a 1-bounded example-memory learner M that identifies all $c \in \mathcal{C}$ from text. M just memorizes the longest element from $\{a\}^*$ presented so far. M guesses the infinite concept $\{a\}^*$ until a string b^{z+1} is presented. Past this point, M always guesses the concept c that contains b^z, all elements a^m with $m \leq z$ as well as the longest element from $\{a\}^*$ seen so far. □

References

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

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

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

[4] J. E. Hopcroft, J. D. Ullman, *Formal Languages and their Relation to Automata* (1969) Addison, Reading, MA.

[5] S. Jain, D. Osherson, J. Royer, A. Sharma, *Systems that Learn – 2nd Edition, An Introduction to Learning Theory*, (1999) MIT Press, Cambridge, MA.

[6] S. Lange, *Algorithmic Learning of Recursive Languages*, (2000) Mensch & Buch Verlag, Berlin.

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

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

[9] R. Rivest, Learning decision lists, *Machine Learning* 2 (1988) 229–246.

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

[11] R. Wiehagen, Limes-Erkennung rekursiver Funktionen durch spezielle Strategien, *Elektronische Informationsverarbeitung und Kybernetik* 12 (1976) 93–99.

[12] A guided tour across the boundaries of learning recursive languages, in: K. P. Jantke, S. Lange (Eds.), *Algorithmic Learning for Knowledge-Based Systems*, Lecture Notes in Artificial Intelligence, Vol. 961 (1995) 190–258.