Refined Incremental Learning

Steffen Lange*
HTWK Leipzig
FB Mathematik und Informatik
PF 66
04521 Leipzig, Germany
steffen@informatik.th-leipzig.de

Thomas Zeugmann
Research Institute of
Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
thomas@rifis.kyushu-u.ac.jp

Abstract

The paper provides a systematic study of incremental learning algorithms. The general scenario is as follows. Let c be any concept; then every infinite sequence of elements exhausting c is called *positive presentation* of c. An algorithmic learner takes as input one element of a positive presentation and its previously made hypothesis at a time, and outputs a new hypothesis about the target concept. The sequence of hypotheses has to converge to a hypothesis correctly describing the target concept. This scenario is referred to as *iterative learning*.

We refine this scenario by defining and investigating *bounded example memory inference* and *feed-back* identification. Bounded example memory and feed-back learning generalizes iterative inference by allowing to store an *a priori* bounded number of carefully chosen examples and asking whether or not a particular element did already appear in the data provided so far, respectively.

We provide a sufficient condition for iterative learning allowing *non-enumerative* learning. The learning power of our models is related to one another, and an infinite hierarchy of bounded example memory inference is established. These results nicely contrast previously made attempts to enlarge the learning capabilities of iterative learners (cf. [8]). In particular, they provide strong evidence that incremental learning is the art of knowing what to overlook. Finally, feed-back learning is more powerful than iterative inference, and its learning power is incomparable to that of bounded example memory inference. Hence, there is no unique way to design superior incremental learning algorithms.

Key words: Algorithmic Learning Theory, Computational issues in A.I., Concept learning

1. Introduction

We consider general systems that map evidence on a concept into hypotheses about it. We deal with scenarios in which the sequence of hypotheses *stabilizes* to an *accurate* and *finite* description of the target concept. Thus, after having seen only finitely many data of the possibly infinite target, the algorithm performing the mapping of the data to hypotheses reaches its (generally unknown) point of convergence to a correct and finite

*This work has been supported by the Japanese International Information Science Foundation under Grant No. 94.3.3.543

description of the target concept. Clearly, then some form of learning must have taken place. Formalizing the notions "evidence," "stabilization," and "accuracy" results in the model of learning in the limit (cf. [3]). During the last three decades much has been learned about the classes of formal languages and recursive functions that can be learned within Gold's [3] model and variations thereof (cf., e.g., [8, 12, 13, 16]). We continue along these lines of research, i.e., we investigate the principal learning capabilities of learners which perform *incremental* learning. Next, we introduce some notations.

A *positive presentation* of a concept c is an infinite sequence of elements that eventually exhausts all and only the elements of c. An algorithmic learner called *inductive inference machine* (abbr. IIM), takes as input initial segments of a positive presentation, and outputs, from time to time, a hypothesis about the target concept. The set $\mathcal{H}$ of all admissible hypotheses is called *hypothesis space*. The sequence of hypotheses has to converge to a hypothesis correctly describing the target concept. If there is an IIM that learns a concept c from all positive presentations for it, then c is said to be *learnable in the limit* with respect to $\mathcal{H}$ (cf. Definition 1).

However, this model makes the unrealistic assumption that the learner has access to the whole initial segment of a positive presentation provided so far. Clearly, each practical learning system has to deal with the limitations of space. Thus, we formally define and investigate variations of learning in the limit restricting the accessibility of input data. We deal with *iterative* learning, *bounded example memory* inference, and *feed-back* identification (cf. Definitions 3, 4, 5). All these models formalize *incremental learning*, a topic attracting considerable attention in the machine learning community (cf., e.g. [2, 10]). An iterative learner is required to produce its actual guesses exclusively from its previous one and the next element in the positive presentation. Results concerning this learning model can be found in [5, 6, 8, 13]. Osherson et al. [8] also considered the variant that the learners has access to the *last* k elements, where k is *a priori* fixed. Interestingly, the latter approach does *not* increase the learning power. Alternatively, we study learners that are allowed to store k *carefully chosen* examples, where k is *a priori* fixed (bounded example memory inference). We obtain an *infinite* hierarchy of more and more powerful learners (cf. Theorem 7). This result provides strong evidence that learning is the art of *knowing what to overlook*. Finally, we study feed-back identification. In this setting, the iterative learner is additionally allowed to ask whether or not a particular element did already appear in the data provided so far. Again, the learning power considerably increases but the supplementary learning power is incomparable to those of bounded example memory inference (cf. Theorem 11). The latter result provides strong evidence that there is no unique way to design superior space efficient inference procedures.

2. Formalizing Incremental Learning

By $\mathbb{N} = \{0, 1, 2, ...\}$ we denote the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. By $\langle ., . \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote ***Cantor's pairing function***. By π_1 and π_2 we denote the ***projection functions*** over $\mathbb{N} \times \mathbb{N}$ to the first and second component, respectively.

Any 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. Let c be a concept, and let $t = x_0, x_1, x_2, ...$ an infinite sequence of elements from c such that $range(t) = \{x_k \mid k \in \mathbb{N}\} = c$. Then t is said to be a ***positive***

presentation for c. By $pos(c)$ we denote the set of all positive presentations of c. Let t be a positive presentation, and let y be a number. Then, t_y denotes the initial segment of t of length $y+1$, and $t_y^+ =_{df} \{x_k \mid k \leq y\}$.

We deal with the learnability of indexable concept classes with uniformly decidable membership defined as follows (cf. [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 $x \in \mathcal{X}$ we have $f(j,x) = 1$, if $x \in c_j$, and $f(j,x) = 0$ otherwise.

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. We set $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, where ε denotes the empty string. Then $\mathcal{X} = \Sigma^+$ serves as the learning domain. We refer to subsets $L \subseteq \Sigma^+$ as to 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. [4, 1]).

Next, 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 form indexable classes (cf. [11, 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 t and it either requests the next input element, or it first outputs a hypothesis, i.e., a number, and then it requests the next input element (cf. [3]).

The indices 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 $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing h_j. Note that $\mathcal{H}$ must be defined over some learning domain $\mathcal{Z}$ comprising the learning domain $\mathcal{X}$ over which $\mathcal{C}$ is defined, and, moreover, $\mathcal{H}$ must comprise the target concept class $\mathcal{C}$.

Let t be a positive presentation, and let $y \in \mathbb{N}$. By $M(t_y)$ we denote the last hypothesis produced by M when successively fed t_y. The sequence $(M(t_y))_{y\in\mathbb{N}}$ is said to ***converge in the limit*** to the number j iff either $(M(t_y))_{y\in\mathbb{N}}$ is infinite and all but finitely many terms of it are equal to j, or $(M(t_y))_{y\in\mathbb{N}}$ is non-empty and finite, and its last term is j. Now we define some models of learning. We start with learning in the limit.

Definition 1 (cf. [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-learns c w.r.t. $\mathcal{H}$*** *iff for every $t \in pos(c)$, there exists a $j \in \mathbb{N}$ such that the sequence $(M(t_y))_{y\in\mathbb{N}}$ converges in the limit to j and $c = h_j$.*

M LIM-learns $\mathcal{C}$ w.r.t. $\mathcal{H}$ iff, for all $c \in \mathcal{C}$, M LIM-learns c w.r.t. $\mathcal{H}$.

Finally, let LIM denote the collection of all indexable classes $\mathcal{C}$ for which there are an IIM M and a hypothesis space $\mathcal{H}$ such that M LIM-learns $\mathcal{C}$ w.r.t. $\mathcal{H}$.

Next we consider the restriction that the IIM is not allowed to output guesses describing proper supersets of the target concept. IIMs behaving thus are called ***conservative***.

Definition 2 (cf. [1]). *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 CONSV-learns c w.r.t. $\mathcal{H}$*** *iff*

(1) *M LIM-learns c w.r.t.* $\mathcal{H}$,

(2) *for every* $t \in pos(c)$ *and for all* $y, k \in \mathbb{N}$, *if* $M(t_y) \neq M(t_{y+k})$ *then* $t^+_{y+k} \not\subseteq h_{M(t_y)}$.

Finally, M CONSV-learns $\mathcal{C}$ *w.r.t.* $\mathcal{H}$ *iff, for each* $c \in \mathcal{C}$, *M CONSV-learns c w.r.t.* $\mathcal{H}$. *By CONSV we denote the collection of all indexable classes* $\mathcal{C}$ *for which there are an IIM M and a hypothesis space* $\mathcal{H}$ *such that M CONSV-learns* $\mathcal{C}$ *w.r.t.* $\mathcal{H}$.

Looking at the definitions above, we see that, in order to compute its actual guess, M is fed all examples seen so far. In contrast to that, next we define ***iterative IIMs*** and a natural generalization of them called ***bounded example memory IIMs***. 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. Conceptionally, an iterative IIM M defines a sequence $(M_n)_{n\in N}$ of machines each of which takes as its input the output of its predecessor. Hence, the IIM M has always to produce a hypothesis.

Definition 3 (cf. [14]). *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-learns c w.r.t.*** $\mathcal{H}$ *iff for every* $t = (x_j)_{j\in\mathbb{N}} \in pos(c)$ *the following conditions are satisfied:*

(1) *for all* $n \in \mathbb{N}$, $M_n(t)$ *is defined, where* $M_0(t) =_{df} M(x_0)$ *and for all* $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), x_{n+1})$,

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

Finally, M IT-learns $\mathcal{C}$ *w.r.t.* $\mathcal{H}$ *iff, for each* $c \in \mathcal{C}$, *M IT-learns c w.r.t.* $\mathcal{H}$.

The resulting learning type IT is analogously defined as above.

Next, we introduce a natural relaxation of iterative learning. Now, an IIM M is allowed to memorize an *a priori* bounded number of the examples it already has had access to during the learning process. Again, M defines a sequence $(M_n)_{n\in\mathbb{N}}$ of machines each of which takes as input the output of its predecessor. Thus, a bounded example memory IIM has to output a hypothesis as well as a subset of the set of examples seen so far.

Definition 4. *Let* $k \in \mathbb{N} \cup \{*\}$, *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*** BEM_k***-learns c w.r.t.*** $\mathcal{H}$ *iff for every* $t = (x_j)_{j\in\mathbb{N}} \in pos(c)$ *the following conditions are satisfied:*

(1) *for all* $n \in \mathbb{N}$, $M_n(t)$ *is defined, where* $M_0(t) =_{df} M(x_0) = \langle j_0, S_0\rangle$ *such that* $S_0 \subseteq t^+_0$ *and* $\mathrm{card}(S_0) \leq k$, *and for all* $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), x_{n+1}) = \langle j_{n+1}, S_{n+1}\rangle$ *such that* $S_{n+1} \subseteq S_n \cup \{x_{n+1}\}$ *and* $\mathrm{card}(S_{n+1}) \leq k$, *(*$k = *$ *means finitely many.)*

(2) *the sequence* $(\pi_1\langle j_n, S_n\rangle)_{n\in\mathbb{N}}$ *converges in the limit to j such that* $c = h_j$.

Finally, M BEM_k*-learns* $\mathcal{C}$ *w.r.t.* $\mathcal{H}$ *iff, for each* $c \in \mathcal{C}$, *M* BEM_k*-learns c w.r.t.* $\mathcal{H}$.

For every $k \in \mathbb{N} \cup \{*\}$, the resulting learning type BEM_k is analogously defined as above. By definition, $IT = BEM_0$ as well as $BEM_* = LIM$.

Finally, we define learning by feed-back IIMs. The idea of feed-back learning goes back to Wiehagen [14] who considered it in the setting of inductive inference of recursive functions. However, his definition cannot be directly applied to concept learning. A feed-back IIM M is an iterative IIM that is additionally allowed to ask a particular type of questions. In each learning stage $n+1$, M has access to the actual input x_{n+1}, and its previous guess j_n, and M is additionally allowed to compute a query from x_{n+1} and j_n. The query concerns the history of the learning process. That is, an element x and a "YES/NO" answer A are computed such that $A = 1$ iff $x \in t^+_n$ and $A = 0$, otherwise.

Thus, M can just ask whether or not a particular string has already been presented in previous learning stages.

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. Moreover, let $Q: \mathcal{X} \times \mathbb{N} \to \mathcal{X}$, and $A: \mathcal{X} \to \{0,1\}$ be computable total mappings.* ***An IIM M FB-learns c w.r.t. $\mathcal{H}$*** *iff for every $t = (x_j)_{j\in\mathbb{N}} \in pos(c)$ the following conditions are satisfied:*

(1) *for all $n \in \mathbb{N}$, $M_n(t)$ is defined, where $M_0(t) =_{df} M(x_0)$ and for all $n \geq 0$: $M_{n+1}(t) =_{df} M(M_n(t), A(Q(M_n(t), x_{n+1})), x_{n+1})$,*

(2) *the sequence $(M_n(t))_{n\in\mathbb{N}}$ converges in the limit to a number j such that $c = h_j$ provided that A truthfully answers the questions computed by Q.*

Finally, M FB-learns $\mathcal{C}$ w.r.t. $\mathcal{H}$ iff there are computable mappings Q and A as described above such that, for each $c \in \mathcal{C}$, M FB-learns c w.r.t. $\mathcal{H}$.

3. Results

In this section we relate the learning power of all the models introduced to one another. Moreover, we provide results showing that rich concepts classes are incrementally learnable. Due to the lack of space, no proofs are included. The interested reader is referred to Lange and Zeugmann [7] for a full version of this extended abstract.

3.1. Iterative Learning

There are several well-known criteria that ensure learnability in the limit of indexable classes from positive data, i.e., finite thickness and finite elasticity. Both conditions are sufficient but not necessary. Hence, it is natural to ask whether or not these conditions guarantee iterative learning, too. Unfortunately, the general answer is negative. However, a natural sharpening of finite thickness directly yields a sufficient condition for iterative learning.

Definition 6. *Let $\mathcal{C}$ be an indexable class. $\mathcal{C}$ has* ***finite thickness*** *if and only if for every $x \in \mathcal{X}$ there are at most finitely many $c \in \mathcal{C}$ satisfying $x \in c$.*

Theorem 1. *There is an indexable class $\mathcal{C} \notin IT$ which has finite thickness.*

Next, we define recursive finite thickness. Let $\mathcal{X}$ be any recursively enumerable learning domain, and let $x_0,\ x_1,\ x_2,\ \ldots$ be any effective enumeration of all elements in $\mathcal{X}$. Furthermore, assume an effective enumeration $N_0,\ N_1,\ N_2,\ \ldots$ of all finite subsets of $\mathbb{N}$.

Definition 7. *Let $\mathcal{C}$ be an indexable class. $\mathcal{C}$ has* ***recursive finite thickness*** *provided there are an indexing $c_0,\ c_1,\ c_2,\ \ldots$ of $\mathcal{C}$ and a total recursive function g such that, for all $m,\ k \in \mathbb{N}$, $x_m \in c_k$ if and only if $k \in N_{g(m)}$ or there is a $j \in N_{g(m)}$ with $c_j = c_k$.*

It is easy to verify that the class of all concepts describable by a monomial, a k-CNF, a k-DNF, a k-decision list, respectively, have recursive finite thickness. The pattern languages provide another interesting example of a concept class having recursive finite thickness. The following theorem establishes the iterative learnability of all these concept classes.

Theorem 2. *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ has recursive finite thickness, then $\mathcal{C} \in IT$.*

The proof of the latter theorem has some interesting features we want to point to. First of all, the learning algorithm produces its hypotheses in a rather *constructive* manner. This nicely contrasts the enumerative character of many inference procedures often provided in abstract studies within Gold's [3] model (cf., e.g., [3, 8]). In contrast, our general learning algorithm immediately produces a finite subspace of hypotheses from which it computes its actual guess. Subsequently, it *deletes* all nonrelevant hypotheses from this subspace. Moreover, the algorithm learns by *generalization*, i.e., the sequence of its guesses constitutes an augmenting chain of concepts. As a matter of fact, the converse is also true. Whenever the learning process can be exclusively performed by generalization, then one can learn iteratively, too (cf. [6]). However, the generality of the result above does not always yield the most effective iterative learning algorithm. For example, a straightforward application of Valiant's [11] proof technique directly yields iterative learning algorithms for the class of all concepts describable by a k-CNF and k-DNF, respectively, that are much more efficient. Another example are the pattern languages. In this case, Lange and Wiehagen's [5] iterative learning algorithm is the better choice.

Nevertheless, the proof given above allows some further general insight. That is, every indexable class possessing recursive finite thickness can be identified by a conservative IIM. Moreover, Lange and Zeugmann [6] have shown that $CONSV \setminus IT \neq \emptyset$. Hence, one may conjecture that $IT \subset CONSV$. On the other hand, iterative learning is not requested to realize the *subset principle* (cf. [12]). However, since conservative IIMs have access to the whole initial segment of a positive presentation provided so far, they are able to compensate this additional strength of iterative learning.

Theorem 3. $IT \subset CONSV$.

Note that the above theorem heavily depends on our assumption that an IIM may select a hypothesis space that comprises the target class.

As our next result states, recursive finite thickness is only a sufficient criteria that ensures learnability by iterative IIMs.

Theorem 4. *There is an indexable class $\mathcal{C} \in IT$ not having recursive finite thickness.*

Next, we consider finite elasticity introduced by Wright [15].

Definition 8. *Let $\mathcal{C}$ be an indexable class. $\mathcal{C}$ has **infinite elasticity** if and only if there are an infinite sequence of elements $x_0,\ x_1,\ x_2,\ \ldots$ and an infinite sequence of concepts $c_0,\ c_1,\ c_2,\ \ldots$ each in $\mathcal{C}$ such that, for all $n \in \mathbb{N}$, $\{x_0, \ldots, x_{n-1}\} \subseteq c_n$ but $x_n \notin c_n$. $\mathcal{C}$ has **finite elasticity** provided that $\mathcal{C}$ does not have infinite elasticity.*

Obviously, finite thickness implies finite elasticity. Therefore, we may easily conclude:

Corollary 5. *There is an indexable class $\mathcal{C} \notin IT$ which has finite elasticity.*

On the other hand, the indexable class $\mathcal{C}$ used in the demonstration of Theorem 4 does not have finite elasticity as well. Consequently:

Corollary 6. *There is an indexable class $\mathcal{C} \in IT$ which does not have finite elasticity.*

3.2. Bounded Example Memory Inference

As it turns out, both IIMs with an *a priori* bounded example memory and feed-back IIMs are more powerful than iterative ones. Interestingly enough, even the ability to store exactly one distinguished example seriously increases the learning capabilities of iterative IIMs.

Theorem 7.

(1) $IT \subset BEM_1$,

(2) $BEM_k \subset BEM_{k+1}$ *for all* $k \in \mathbb{N}$.

The additional learning power of IIMs with an *a priori* bounded example memory mainly comes from two sources. First of all, carefully chosen examples can be memorized. A different version goes back to Osherson *et al.* [8] who associated with a learning device a window allowing the IIM to inspect in every stage the last k examples presented. However, this approach does *not* enlarge the learning capabilities of iterative IIMs. Second, the sequence of the stored examples is not required to converge. If it would, again, the resulting learning power equates that of iterative machines.

Our next result states that bounded example memory IIMs are *not* able to capture the whole learning power of conservative IIMs.

Theorem 8.

(1) $CONSV \setminus \bigcup_{k \in \mathbb{N}} BEM_k \neq 0$.

(2) $\bigcup_{k \in \mathbb{N}} BEM_k \subset LIM$.

3.3. Feed-Back Learning

Now we study to what extend, if ever, feed-back learning enlarges the learning capabilities of iterative and bounded example memory IIMs, respectively. As the next theorem shows, feed-back learning is more powerful than iterative inference, i.e., the ability to ask whether or not a particular example did already appear seriously increases the learning capabilities of iterative IIMs, too.

Theorem 9. $IT \subset FB$

Next, we compare feed-back inference with conservative inference and learning in the limit.

Theorem 10.

(1) $CONSV \setminus FB \neq \emptyset$.

(2) $FB \subset LIM$.

Finally, the increase in the learning power obtained by bounded examples memories and feed-back questions is incomparable. Consequently, there is no unique way to design superior learning algorithms when space limitations are a serious concern.

Theorem 11.

(1) $FB \setminus \bigcup_{k \in \mathbb{N}} BEM_k \neq \emptyset$,

(2) $BEM_1 \setminus FB \neq \emptyset$.

Corollary 12. *$BEM_k \# FB$ for all $k \in \mathbb{N}^+$.*

Parts of the latter theorem and corollary are obtained by comparing the learning power of finite inference from *positive and negative data* (abbr. $FIN{-}INF$) with those of bounded example memory learning and feed-back inference. Finite inference is similarly defined as learning in the limit but the learner is restricted to a single output that must be correct. As it turned out, feed-back learning from positive data can simulate finite inference from *positive and negative data* while bounded example memory learning cannot (cf. [7] for details). This is interesting, since it addresses the issue whether information presentation can be traded versus memory limitations. The only known result in this regard established $FIN-INF \subset CONSV$ (cf. [16]).

4. References

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

[2] A. Cornuéjols, Getting order independence in incremental learning, *Proc. European Conference on Machine Learning 1993*, (P.B. Brazdil, Ed.), Lecture Notes in Artificial Intelligence 667, Springer-Verlag, Berlin, 1993, pp. 196 – 212.

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

[4] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata*, Addison-Wesley, Reading, Mass., 1969.

[5] S. Lange and R. Wiehagen, Polynomial-time inference of arbitrary pattern languages, *New Generation Computing*, **8** (1991), 361 – 370.

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

[7] S. Lange and T. Zeugmann, Modeling incremental learning from positive data, RIFIS Technical Report RIFIS-TR-CS-117, Kyushu University 33, August 1995.

[8] D. Osherson, M. Stob and S. Weinstein, *Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists*, MIT Press, Cambridge, Mass., 1986.

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

[10] L. Torgo, Controlled redundancy in incremental rule learning, *Proc. European Conference on Machine Learning 1993*, (P.B. Brazdil, Ed.), Lecture Notes in Artificial Intelligence 667, Springer-Verlag, Berlin, 1993, pp. 185 – 195.

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

[12] K. Wexler, The subset principle is an intensional principle, In *Knowledge and Language: Issues in Representation and Acquisition* (E. Reuland and W. Abrahamson, Eds.), Kluwer Academic Publishers, 1992.

[13] K. Wexler and P. Culicover, *Formal Principles of Language Acquisition*, MIT Press, Cambridge, Mass., 1980.

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

[15] K. Wright, Identification of unions of languages drawn from an identifiable class, *Proc. 2nd Annual ACM Workshop on Computational Learning Theory*, Morgan Kaufmann, San Mateo, 1989, pp. 328 – 333.

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