Trading Monotonicity Demands versus Mind Changes

Steffen Lange
HTWK Leipzig
FB Mathematik und Informatik
PF 66
04275 Leipzig
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 present paper deals with with the learnability of indexed families $\mathcal{L}$ of uniformly recursive languages *from positive data.* We consider the influence of three monotonicity demands to the efficiency of the learning process. The efficiency of learning is measured in dependence on the number of mind changes a learning algorithm is allowed to perform. The three notions of monotonicity reflect different formalizations of the requirement that the learner has to produce better and better generalizations when fed more and more data on the target concept.

We distinguish between *exact* learnability ($\mathcal{L}$ has to be inferred with respect to $\mathcal{L}$), *class preserving* learning ($\mathcal{L}$ has to be inferred with respect to some suitable chosen enumeration of all the languages from $\mathcal{L}$), and *class comprising* inference ($\mathcal{L}$ has to be learned with respect to some suitable chosen enumeration of uniformly recursive languages containing at least all the languages from $\mathcal{L}$).

In particular, we prove that a relaxation of the relevant monotonicity requirement may result in an arbitrarily large speed-up.

1. Introduction

The present paper deals with inductive inference of formal languages. Looking at potential applications, Angluin (1980) started the systematic study of learning enumerable families of uniformly recursive languages, henceforth called *indexed families.* Recently, this topic has attracted much attention (cf., e.g., Shinohara (1990), Kapur and Bilardi (1992), Lange and Zeugmann (1993a), Mukouchi (1992), Wiehagen and Zeugmann (1994)).

Next we specify the information from which the target languages have to be learned. Throughout this paper we exclusively consider learning from *positive* *data*, or synonymously from text. A *text* of a language L is an infinite sequence of strings that eventually contains all strings of L.

An algorithmic learner, henceforth called *inductive inference machine* (abbr. IIM), takes as input initial segments of a text, and outputs, from time to time, a hypothesis about the target language. The set $\mathcal{G}$ of all admissible hypothesis is called *hypothesis space*. Furthermore, the sequence of hypotheses has to converge to a hypothesis correctly describing the language to be learned, i.e., after some point, the IIM stabilizes to an accurate hypothesis. If there is an IIM that learns a language L from all texts for it, then L is said to be *learnable from text in the limit* with respect to the hypothesis space $\mathcal{G}$. Consequently, when dealing with learning in the limit, we are faced with an ongoing inference process. If $d_0, ..., d_x$, $x = 0, 1, 2, ...$, denotes the sequence of data the IIM M is successively fed, then we use j_x to denote the last hypothesis output by M, if any, on successive input $d_0, ..., d_x$. We say that M changes its mind, or synonymously, M performs a mind change, iff $j_x \neq j_{x+1}$. The number of mind changes is a measure of efficiency and has been introduced by Barzdin and Freivalds (1972). Subsequently, this measure of efficiency has been intensively studied (cf., e.g., Barzdin, Kinber and Podnieks (1974), Case and Smith (1983), Wiehagen, Freivalds and Kinber (1984), Gasarch and Velauthapillai (1992)). However, all the mentioned papers considered the learnability of recursive functions. Hence, it is only natural to ask whether or not this measure of efficiency is of equal importance in the setting of language learning. This is indeed the case as recently obtained results show (cf., e.g., Mukouchi (1992), Lange and Zeugmann (1993b), Lange (1994)).

In this paper we study problems of higher granularity. In order to explain them we have to describe the monotonicity constraints we are going to deal with. The three notions of monotonicity reflect different formalizations of the requirement that the learner has to produce better and better generalizations when fed more and more data on the target concept (cf. Jantke (1991), Wiehagen (1991)). Interpreting generalization in its strongest sense yields that the learner is forced to produce an augmenting chain of languages, i.e., $L_i \subseteq L_j$ in case L_j is hypothesized later than L_i. This learning model is referred to as *strong monotonic* inference. Restricting "better generalization" to the language L to be learned results in demanding $L_i \cap L \subseteq L_j \cap L$ provided L_j is later guessed than L_i. Learning algorithms behaving thus are called *monotonic*.

Weakening the strong-monotonicity constraint in the same way as the monotonicity principle of classical logic is generalized to cumulativity yields *weak-monotonic* learning, i.e., now the learner is required to behave strong-monotonically as long as it does not receive data contradicting its actual guess (cf. Definition 3).

As Jantke (1991) pointed out, the monotonicity requirements described above reflect different degrees of *non-monotonic* reasoning that may be incorporated into the learning process. However, it is well imaginable that the use of non-monotonic reasoning does not only affect the learnability at all but also the *efficiency* of learning. Kinber (1994) first studied this problem for learning recursively enumerable languages. We continue along this line in the setting of uniformly recursive languages.

Clearly, this question is directly related to the problem of what a natural learning algorithm might look like. In particular, it is well imaginable that one may succeed in designing a learning algorithm that fulfills a desirable monotonicity demand. However, it seems to be interesting to know what price one might have to pay concerning the resulting efficiency. Therefore, we study the influence of different monotonicity constraints to the number of mind changes an IIM has to perform when inferring a target indexed family. Then, the right question to ask is whether a weakening of the monotonicity requirement may yield a speed-up. Therefore, we always start with a target indexed family inferable under some monotonicity constraint with an *a priori* fixed number of mind changes. Then we ask whether or not the least or some possible relaxation of the corresponding monotonicity requirement might help to uniformly reduce the number of mind changes. As we shall see, there is no unique answer to this problem.

2. Preliminaries

Let $\mathbb{N} = \{0, 1, 2, ...\}$ be the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Let $\varphi_0, \varphi_1, \varphi_2, ...$ denote any fixed ***acceptable programming system*** of all (and only all) partial recursive functions over $\mathbb{N}$, and let $\Phi_0, \Phi_1, \Phi_2, ...$ be any associated ***complexity measure*** (cf. Machtey and Young (1978)). Furthermore, let $k, x \in \mathbb{N}$. 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. $\varphi_k(x)\uparrow$). By $\langle\cdot,\cdot\rangle: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ we denote ***Cantor's pairing function***, i.e., $\langle x, y\rangle = ((x + y)^2 + 3x + y)/2$ for all $x, y \in \mathbb{N}$.

In the sequel we assume familiarity with formal language theory (cf., e.g., Hopcroft and Ullman (1969)). By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ, and let $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, where ε denotes the empty string. Any $L \subseteq \Sigma^*$ is called a language. Let L be a language and $t = s_0, s_1, s_2, ...$ an infinite sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in \mathbb{N}\} = L$. Then t is said to be a ***text*** for L or, synonymously, a ***positive presentation***. Let L be a language. By $text(L)$ we denote the set of all positive presentations of L. Moreover, let t be a text and let $x \in \mathbb{N}$. Then t_x denotes the initial segment of t of length $x + 1$, and t_x^+ its range, i.e., $t_x^+ = \{s_k \mid k \leq x\}$.

In this paper we deal with the learnability of indexed families defined as follows: A sequence $L_0, L_1, L_2, ...$ is said to be an ***indexed family*** provided all languages L_j are non-empty and membership in L_j is uniformly decidable for all $j \in \mathbb{N}$. Note that the definition of an indexed family includes both, a description for every language L_j, and a particular enumeration of all the languages.

As in Gold (1967), we define an ***inductive inference machine*** (abbr. IIM) to be an algorithmic device which works as follows: The IIM takes as its input larger and larger initial segments of a text t and it either requests the next input string, or it first outputs a hypothesis, i.e., a number, and then it requests the next input string.

At this point we have to clarify what space of hypotheses we should choose. Since we exclusively deal with the learnability of indexed families $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ we always take as hypothesis space an enumerable family of grammars $\mathcal{G} = G_0, G_1, G_2, \ldots$ over the terminal alphabet Σ satisfying $range(\mathcal{L}) \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$, and require that membership in $L(G_j)$ is uniformly decidable for all $j \in \mathbb{N}$ and all $s \in \Sigma^*$. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing the grammar G_j. Let t be a text, and $x \in \mathbb{N}$. Then we use $M(t_x)$ to denote the last hypothesis produced by M when successively fed t_x. The sequence $(M(t_x))_{x\in\mathbb{N}}$ is said to ***converge in the limit*** to the number j if and only if either $(M(t_x))_{x\in\mathbb{N}}$ is infinite and all but finitely many terms of it are equal to j, or $(M(t_x))_{x\in\mathbb{N}}$ is non-empty and finite, and its last term is j. Now we are ready to define learning in the limit from positive data.

Definition 1. (Gold, 1967) *Let $\mathcal{L}$ be an indexed family, let L be a language, and let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space.* ***An IIM M CLIM–identifies L from text with respect to $\mathcal{G}$*** *iff for every text t for L, there exists a $j \in \mathbb{N}$ such that the sequence $(M(t_x))_{x\in\mathbb{N}}$ converges in the limit to j and $L = L(G_j)$.*

Furthermore, M $CLIM$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in range(\mathcal{L})$, M $CLIM$–identifies L with respect to $\mathcal{G}$.

Finally, let $CLIM$ denote the collection of all indexed families $\mathcal{L}$ for which there are an IIM M and a hypothesis space $\mathcal{G}$ such that M $CLIM$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$.

In the above Definition LIM stands for "limit." Furthermore, the prefix C is used to indicate ***class comprising*** learning, i.e., the fact that $\mathcal{L}$ may be learned with respect to some hypothesis space comprising $range(\mathcal{L})$. The restriction of $CLIM$ to ***class preserving*** inference is denoted by LIM. That means LIM is the collection of all indexed families $\mathcal{L}$ that can be learned in the limit with respect to a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ such that $range(\mathcal{L}) = \{L(G_j) \mid j \in \mathbb{N}\}$. Moreover, if a target indexed family $\mathcal{L}$ has to be inferred with respect to the hypothesis space $\mathcal{L}$ itself, then we replace the prefix C by E, i.e., $ELIM$ is the collection of indexed families that can be ***exactly*** learned in the limit. Finally, we adopt this convention in defining all the learning types below.

By the definition of convergence, whenever an IIM identifies the language L, then it performs at most finitely many mind changes. However, the precise number of mind changes may well vary from text to text as well as for every language $L \in range(\mathcal{L})$. In particular, the number of allowed mind changes is *not* required to be *universally bounded* for all $L \in range(\mathcal{L})$. Within the next definition we consider the special case that the number of allowed mind changes is universally bounded by an *a priori* fixed number.

Definition 2. (Barzdin and Freivalds, 1972) *Let $\mathcal{L}$ be an indexed family, let L be a language, let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be a hypothesis space, and let $k \in \mathbb{N} \cup \{*\}$.* ***An IIM $CLIM_k$–identifies L from text with respect to $\mathcal{G}$*** *iff*

(1) *M $CLIM$–identifies L from text with respect to $\mathcal{G}$,*

(2) *for every text t for L the IIM M performs, when fed t, at most k ($k = *$ means at most finitely many) mind changes, i.e., $card(\{x \mid M(t_x) \neq M(t_{x+1})\}) \leq k$.*

Moreover, M $CLIM_k$–identifies $\mathcal{L}$ with respect to $\mathcal{G}$ if and only if, for each $L \in range(\mathcal{L})$, M $CLIM_k$–identifies L with respect to $\mathcal{G}$.

$CLIM_k$ is defined in the same way as above.

Obviously, $\lambda LIM_* = \lambda LIM$ for all $\lambda \in \{E, \varepsilon, C\}$. Moreover, λLIM_0 is also referred to as *finite learning*, $\lambda \in \{E, \varepsilon, C\}$, since the IIM is only allowed to produce a single guess that cannot be changed later. Note that the learning types λLIM_k do heavily depend on $\lambda \in \{E, \varepsilon, C\}$ (cf. Lange and Zeugmann (1993b), Lange (1994)).

Next, we want to formally define strong-monotonic, monotonic and weak-monotonic inference.

Definition 3. (Jantke, 1991; Wiehagen, 1991) *Let L be a language, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be a hypothesis space.* ***An IIM M is said to identify the language L from text with respect to $\mathcal{G}$***

(A) ***strong-monotonically***

(B) ***monotonically***

(C) ***weak-monotonically***

iff

M $CLIM$–identifies L with respect to $\mathcal{G}$ and for every text t of L as well as for any two consecutive hypotheses j_x, j_{x+k} which M has produced when fed t_x and t_{x+k}, where $k \geq 1, k \in \mathbb{N}$, the following conditions are satisfied:

(A) $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$

(B) $L(G_{j_x}) \cap L \subseteq L(G_{j_{x+k}}) \cap L$

(C) *if $t^+_{x+k} \subseteq L(G_{j_x})$, then $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$.*

We denote by $CSMON$, $CMON$, $CWMON$ the collection of all those indexed families $\mathcal{L}$ for which there are a hypothesis space $\mathcal{G}$ and an IIM inferring them strong-monotonically, monotonically, and weak-monotonically from text with respect to $\mathcal{G}$. Note that the learning types $\lambda SMON$, λMON, and $\lambda WMON$ do heavily depend on $\lambda \in \{E, \varepsilon, C\}$ (cf. Lange and Zeugmann (1993c)).

Finally, we use $CSMON_k$, $CMON_k$, $CWMON_k$, where $k \in \mathbb{N}$, to denote the collections of all those indexed families $\mathcal{L}$ for which there are a hypothesis space $\mathcal{G}$ and an IIM inferring them strong-monotonically, monotonically, and weak-monotonically from text *with at most k mind changes* with respect to $\mathcal{G}$.

3. Results

In this section we study the problem whether or not any of the monotonicity constraints defined above may be traded versus the efficiency of learning. Since each monotonicity demand has its peculiarities, we handle each of them separately in a special subsection. Moreover, in the following we exclusively consider the case where at least one mind change is mandatory, since otherwise finite learning is compared with some type of monotonic learning.

3.1. Strong-Monotonic Inference

We start our investigations with the strongest possible monotonicity constraint, i.e., with $SMON$ and its variations.

Theorem 1. *Let $\mathcal{L}$ be an indexed family. Then, for every $n \in \mathbb{N}^+$ we have:*

(1) $\mathcal{L} \in ESMON_{n+1} \setminus ESMON_n$ *implies* $\mathcal{L} \notin CLIM_n$,

(2) $\mathcal{L} \in SMON_{n+1} \setminus SMON_n$ *implies* $\mathcal{L} \notin CLIM_n$.

Proof. The proof is based on the following observations. Let $\hat{M}$ be any strong-monotonic IIM, and let $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ be any hypothesis space such that $\hat{M}$ witnesses $\mathcal{L} \in SMON_{n+1}$ with respect to $\mathcal{G}$. Then the IIM $\hat{M}$ can be simulated by an IIM M such that for all texts $t \in \bigcup_{L \in range(\mathcal{L})} text(L)$ and all $x \in \mathbb{N}$

(A) if M on input t_x makes an output j_x then $t_x^+ \subseteq L(G_{j_x})$, i.e., M is consistent, and if $M(t_x) \neq M(t_{x+1})$, then $t_{x+1}^+ \not\subseteq L(G_{j_x})$, i.e., M is conservative,

(B) M witnesses $\mathcal{L} \in SMON_{n+1}$ with respect to $\mathcal{G}$, i.e., M performs at most as many mind changes as $\hat{M}$ does (cf. Lange and Zeugmann (1993a)).

Let $\mathcal{L}$ be any indexed family with $\mathcal{L} \in SMON_{n+1} \setminus SMON_n$. Furthermore, let $\mathcal{L} \in SMON_{n+1}$ be witnessed by M, where M is chosen in accordance with (A) and (B). Since $\mathcal{L} \notin SMON_n$, there have to be an $L \in range(\mathcal{L})$ and a text t for L such that M changes its mind exactly $n+1$ times when fed t. Let $j_0, \dots, j_{n+1}$ denote the finite sequence of M's mind changes produced on t. Since M is strong-monotonic, consistent and conservative, we directly obtain that $L(G_{j_0}) \subset \cdots \subset L(G_{j_{n+1}}) = L$.

Now, $\mathcal{L} \notin CLIM_n$ is a direct consequence of Proposition 3.7 by Mukouchi (1994). This proves Assertion (2). Finally, the same arguments apply in order to prove Assertion (1). q.e.d.

The latter theorem allows the following interpretation. Relaxing the requirement to learn exactly (class preservingly) strong-monotonically as much as possible does not increase the efficiency. This is even true, if we are allowed to choose an arbitrary class comprising hypothesis space provided that the target indexed family is inferable in the sense of $ESMON_{n+1}$ ($SMON_{n+1}$), but cannot be exactly (class preservingly) and strong-monotonically learned with at most n mind changes for some $n \in \mathbb{N}$. Hence, in the case considered in Theorem 1 the possible efficiency of learning is completely determined by the topology of the target indexed family.

Next we consider the class comprising case. Interestingly enough, now the situation considerably changes. The following theorem shows that a suitable choice of the hypothesis space may increase the efficiency of learning, even under the strong-monotonicity constraint.

Theorem 2. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that*

$$\mathcal{L} \in (CSMON_{n+1} \cap ELIM_n) \setminus CSMON_n.$$

Proof. Due to the lack of space, we only sketch the main proof ideas. Consider the $n = 1$ case. The first idea is to incorporate a non-recursive but recursively enumerable problem in the definition of the target indexed family. Note that this incorporation has to be done in a way such that membership in the enumerated languages remains uniformly decidable. For that purpose, we used the halting problem. Without loss of generality, we may assume that $\Phi_j(j) \geq 1$ for all $j \in \mathbb{N}$.

The desired indexed family is defined as follows. Let $k, j \in \mathbb{N}$. We set $L_{3\langle k,j\rangle} = \{a^k b^z \mid z \in \mathbb{N}^+\}$. The remaining languages will be defined as follows.

Case 1. $\neg\Phi_k(k) \leq j$.

Then we set $L_{3\langle k,j\rangle+1} = L_{3\langle k,j\rangle+2} = L_{3\langle k,0\rangle}$.

Case 2. $\Phi_k(k) \leq j$.

Let $n = \Phi_k(k)$. Now we set: $L_{3\langle k,j\rangle+1} = \{a^k b^z \mid 1 \leq z \leq n\} \cup \{a^k c^n\}$, and $L_{3\langle k,j\rangle+2} = L_{3\langle k,0\rangle} \cup \{a^k d^n\}$.

It is easy to see that $\mathcal{L} = (L_z)_{z\in\mathbb{N}}$ is an indexed family. Whenever $\Phi_k(k)\downarrow$, the main problem for any strong-monotonic IIM consists in learning the finite language $L_{3\langle k,\Phi_k(k)\rangle+1}$ with at most one mind change. Hence, for proving $\mathcal{L} \in CSMON_2$, another ingredient is required, i.e., a suitable choice of a hypothesis space. A suitable hypothesis space $\tilde{\mathcal{L}} = (\tilde{L}_i)_{i\in\mathbb{N}}$ can be defined as follows: For all $k, j \in \mathbb{N}$ and $z \in \{0, 1, 2\}$, we set:

$$\tilde{L}_{3\langle k,j\rangle+z} = \begin{cases} \bigcap_{n\in\mathbb{N}} L_{3\langle k,n\rangle+z}, & \text{if } j = 0, \\ L_{3\langle k,j\rangle+z}, & \text{otherwise.} \end{cases}$$

Now, it is not hard to define an IIM which $CSMON_2$–learns $\mathcal{L}$ with respect to $\tilde{\mathcal{L}}$. Moreover, the following IIM M $ELIM_1$–learns $\mathcal{L}$. Let $L \in range(\mathcal{L})$, $t \in text(L)$, and $x \in \mathbb{N}$.

IIM *M*: "On input t_x do the following: Determine the unique k such that $a^k b^m \in t_x^+$ for some $m \in \mathbb{N}$. Test whether or not $t_x^+ \subseteq L_{3\langle k,0\rangle}$. In case it is, output $3\langle k, 0\rangle$, and request the next input.
Otherwise, goto (A).

(A) Compute $n = \Phi_k(k)$. In case that $a^k c^n \in t_x^+$, output $3\langle k,n\rangle + 1$, and request the next input.
Otherwise, output $3\langle k,n\rangle + 2$, and request the next input."

The harder part is to show that $\mathcal{L} \notin CSMON_1$. As long as only class preserving hypothesis spaces are allowed, it is intuitively obvious that any IIM M strong-monotonically learning $\mathcal{L}$ has to solve the halting problem. However, we have additionally to show that none of the possible choices of the hypothesis space may prevent M to recursively handle the halting problem. Suppose, there are a class comprising hypothesis space $\mathcal{G}$ for $\mathcal{L}$, and an IIM M witnessing $\mathcal{L} \in CSMON_1$ with respect to $\mathcal{G}$. Then, we can define the following effective procedure solving the halting problem.

"Let $k \in \mathbb{N}$, and let t be the lexicographically ordered text for $L_{3\langle k,0\rangle}$. For $x = 0,1,...$, compute $M(t_x)$ until the minimal index z is found such that M, on successive input t_z outputs its first guess, say j. Then test whether $\Phi_k(k) \leq z+1$. If it is, output $\varphi_k(k)\downarrow$. Otherwise, output $\varphi_k(k)\uparrow$."

It remains to show that the procedure defined above correctly works. Obviously, if the output is $\varphi_k(k)\downarrow$, then $\varphi_k(k)$ is indeed defined. Suppose, the procedure outputs $\varphi_k(k)\uparrow$ but $\varphi_k(k)$ is defined. Hence, $\Phi_k(k)$ is defined, too. Let $y = \Phi_k(k)$. By construction, $y > z+1$. Since M is a strong-monotonic IIM, one easily verifies that $L(G_j) \notin range(\mathcal{L})$. Furthermore, M has to infer $L_{3\langle k,0\rangle}$ from its lexicographically ordered text. Hence, there has to be an $m > z$ such that $M(t_m) = r$ and $L(G_r) = L_{3\langle k,0\rangle}$. Therefore, M performs at least one mind change when seeing t_m. Finally, due to our construction, there is a language $L' \in range(\mathcal{L})$ such that $t_m^+ \subseteq L'$ and $L' \neq L_{3\langle k,0\rangle}$, namely $L' = L_{3\langle k,0\rangle} \cup \{a^k d^y\}$. Consequently, t_m may be extended to a text for L' on which M has to perform an additional mind change, a contradiction.

The cases $n > 1$ may be proved using the same "lifting" technique as in Lange and Zeugmann (1993b). q.e.d.

At this point it is only natural to ask whether the latter theorem generalizes to all indexed families from $CSMON_{n+1} \setminus CSMON_n$ not belonging to $SMON$. The negative answer is provided by our next theorem.

Theorem 3. *For all $n \in \mathbb{N}$, there exists an indexed family $\mathcal{L}$ such that*

(1) $\mathcal{L} \in CSMON_{n+1} \setminus SMON$,

(2) $\mathcal{L} \notin ELIM_n$.

Theorem 3 directly yields the problem whether or not Theorem 2 can be strengthened, i.e., whether or not the number of mind changes that can be traded versus the strong-monotonicity constraint is bounded by one. The answer is provided by our next theorem.

Theorem 4. *For every $n \in \mathbb{N}^+$ there exists an indexed family $\mathcal{L}$ such that*

$$\mathcal{L} \in (CSMON_{n+1} \cap EMON_1) \setminus CSMON_n.$$

Proof. Again, we only sketch the proof using the $n = 2$ case, thereby explaining the *proof technique* developed. The main idea is to suitably iterate the proof technique presented in the demonstration of Theorem 2. Therefore, we incorporate one more halting problem into the definition of the indexed family $\mathcal{L}$ witnessing $\mathcal{L} \in CSMON_3 \setminus CSMON_2$, and $\mathcal{L} \in EMON_1$. This is done as follows. Without loss of generality, we may assume that $\Phi_j(j) \geq 1$ for all $j \in \mathbb{N}$. We set $L_{4\langle k_1,k_2,j\rangle} = \{a^{\langle k_1,k_2\rangle} b^z \mid z \in \mathbb{N}^+\}$ for all $k_1, k_2, j \in \mathbb{N}$. In order to define the remaining languages of $\mathcal{L}$ we distinguish the following cases.

Case 1. $\neg\Phi_{k_1}(k_1) \leq j$.

Then we set $L_{4\langle k_1,k_2,j\rangle+1} = L_{4\langle k_1,k_2,j\rangle+2} = L_{4\langle k_1,k_2,j\rangle+3} = L_{4\langle k_1,k_2,0\rangle}$.

Case 2. $\Phi_{k_1}(k_1) \leq j$.

Let $n = \Phi_{k_1}(k_1)$. We set $L_{4\langle k_1,k_2,j\rangle+1} = \{a^{\langle k_1,k_2\rangle} b^z \mid 1 \leq z \leq n\} \cup \{a^{\langle k_1,k_2\rangle} c^n\}$.

Furthermore, we distinguish the following subcases.

Subcase 2.1. $\neg\Phi_{k_2}(k_2) \leq j$.

Then let $L_{4\langle k_1,k_2,j\rangle+2} = L_{4\langle k_1,k_2,j\rangle+3} = L_{4\langle k_1,k_2,0\rangle}$.

Subcase 2.2. $\Phi_{k_2}(k_2) \leq j$.

Let $m = \Phi_{k_2}(k_2)$, and $r = n + m$. We set:

$L_{4\langle k_1,k_2,j\rangle+2} = \{a^{\langle k_1,k_2\rangle} b^z \mid 1 \leq z \leq r\} \cup \{a^{\langle k_1,k_2\rangle} d^r\}$, and

$L_{4\langle k_1,k_2,j\rangle+3} = L_{4\langle k_1,k_2,0\rangle} \cup \{a^{\langle k_1,k_2\rangle} e^r\}$.

Now, it is easy to see that $\mathcal{L} = (L_z)_{z\in\mathbb{N}}$ constitutes an indexed family. It remains to show that $\mathcal{L}$ fulfills the stated requirements. As in the proof of Theorem 2 one proves *mutatis mutandis* that $\mathcal{L} \in EMON_1$, and $\mathcal{L} \in CSMON_3$.

The remaining part, i.e., $\mathcal{L} \notin CSMON_2$, is much harder to prove. For that purpose we need some additional insight into the behavior of every IIM that learns $\mathcal{L}$. In particular, we are mainly interested in knowing how every IIM inferring $\mathcal{L}$ behaves when successively fed the lexicographically ordered text for $L_{4\langle k_1,k_2,0\rangle}$. The desired information is provided by the following lemma.

***Lemma* 1**. *Let $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ be any class comprising hypothesis space for $\mathcal{L}$ and let M be any IIM witnessing $\mathcal{L} \in CLIM$ with respect to $\mathcal{G}$. Then we have: For all k_2 there are numbers $k_1, x, j \in \mathbb{N}$ such that $M(t_x) = j$, and $\Phi_{k_1}(k_1) > x + 1$ and $\varphi_{k_1}(k_1)\downarrow$, where t is the lexicographically ordered text of $L_{4\langle k_1,k_2,0\rangle}$.*

Suppose the converse. Then there is a k_2 such that for all k_1, x, j we have: $M(t_x) = j$ implies $\Phi_{k_1}(k_1) \leq x + 1$ or $\Phi_{k_1}(k_1)\uparrow$.

Assuming the latter statement we have the following claim.

Claim. Provided the latter statement is true, any program for M may be used to obtain *non-effectively* an algorithm deciding "$\varphi_{k_1}(k_1)\downarrow$."

By assumption, there is a k_2 such that for all k_1, x, j: If $M(t_x) = j$, then either $\Phi_{k_1}(k_1) \leq x + 1$ or $\Phi_{k_1}(k_1)\uparrow$. Using this k_2 we can define the following algorithm $\mathcal{A}$ deciding the halting problem for all $k_1 \in \mathbb{N}$.

Algorithm $\mathcal{A}$: "On input k_1 execute (A1) and (A2).

(A1) Generate successively the lexicographically ordered text t of $L_{4\langle k_1,k_2,0\rangle}$ and simulate M until the first hypothesis j is produced.
Let x_0 be the least x such that $M(t_x) = j$.

(A2) Test whether $\Phi_{k_1}(k_1) \leq x_0 + 1$.
In case it is, output "$\varphi_{k_1}(k_1)\downarrow$."
Otherwise, output "$\varphi_{k_1}(k_1)\uparrow$" and stop."

First we observe that M has to infer $L_{4\langle k_1,k_2,0\rangle}$ from its lexicographically ordered text t. Hence, M should eventually output a hypothesis j when fed t. Furthermore, Instruction (A2) can be effectively accomplished, too. Hence, $\mathcal{A}$ is an algorithm and the execution of (A1) and (A2) must eventually terminate. Finally, by assumption we immediately obtain the correctness of $\mathcal{A}$'s output. This proves the claim. Since the halting problem is algorithmically undecidable, the lemma follows.

***Lemma* 2.** $\mathcal{L} \notin CSMON_2$.

Suppose the converse, i.e., there exist a hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and an IIM M that $CSMON_2$–learns $\mathcal{L}$ with respect to $\mathcal{G}$. Then we can prove the following lemma.

***Lemma* 3.** *Given any hypothesis space $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ and any program for M witnessing $\mathcal{L} \in CSMON_2$, one can effectively construct an algorithm deciding the halting problem.*

Let $K = \{k \mid \varphi_k(k)\downarrow\}$ and let $j_0, j_1, j_2, \ldots$ be any fixed effective enumeration of K. We define an algorithm $\mathcal{B}$ as follows.

Algorithm $\mathcal{B}$: "On input k_2 execute (B1) and (B2).

(B1) For $z = 0,\ 1,\ 2, \ldots$ successively compute the lexicographically ordered texts $t^{j_0},\ t^{j_1},\ t^{j_2}, \ldots$ for $L_{4\langle j_0,k_2,0\rangle}$, $L_{4\langle j_1,k_2,0\rangle}, \ldots, L_{4\langle j_z,k_2,0\rangle}$ of length $z+1$, respectively. Then, dovetail the simulation of M on successive input of each of these initial segments until the first initial segment $t_x^{j_r}$ $(r, x \leq z)$ and the first hypothesis j are found such that

(α1) $M(t_x^{j_r}) = j$,

(α2) $\Phi_{j_r}(j_r) > x + 1$.

(* By Lemma 1, the execution of (B1) has to terminate *)

(B2) Let $f =_{df} \langle j_r, k_2\rangle$ and $n = \Phi_{j_r}(j_r)$. Furthermore, we define $\hat{t}_{n+y}$ as follows:

$$\hat{t}_{n+y} = \underbrace{a^f b,\ \ldots,\ a^f b^{x+1},\ \ldots,\ a^f b^n}_{=t_{n-1}^{j_r}},\ a^f b^n,\ \underbrace{a^f b^{n+1},\ \ldots,\ a^f b^{n+y}}_{y-\text{strings}}$$

For $y = 0,\ 1,\ 2,\ \ldots$ execute in parallel (β1) and (β2) until (β3) or (β4) happens.

(β1) Test whether $\Phi_{k_2}(k_2) \leq n + y$.

($\beta 2$) Compute $j_{n+y} = M(\hat{t}_{n+y})$.

($\beta 3$) $\Phi_{k_2}(k_2) \leq n+y$ is verified. Then output "$\varphi_{k_2}(k_2)\downarrow$."

($\beta 4$) In ($\beta 2$) a hypothesis $j_{n+y} = M(\hat{t}_{n+y})$ is computed such that $a^f b^{n+1} \in L(G_{j_{n+y}})$. Then output "$\varphi_{k_2}(k_2)\uparrow$" and stop."

We omit the proof of $\mathcal{B}$'s termination and correctness. q.e.d.

Note that the proof of the latter theorem directly allows the following corollary.

Corollary 5. $EMON_1 \setminus SMON \neq \emptyset$.

3.2. Monotonic Inference

This subsection deals with monotonic inference, and possible relaxations of the monotonicity requirement. But there is a peculiarity which we point out with the following theorem.

Theorem 6. $\lambda LIM_1 = \lambda MON_1$ *for all* $\lambda \in \{E, \varepsilon, C\}$,

Proof. Let $\mathcal{L}$ be any indexed family such that $\mathcal{L} \in \lambda LIM_1$, where $\lambda \in \{E, \varepsilon, C\}$. Hence, there are a hypothesis space $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ and an IIM M that λLIM_1–infers $\mathcal{L}$ with respect to $\mathcal{G}$. Consequently, when fed any text of any language $L \in range(\mathcal{L})$ the IIM M performs at most one mind change. Suppose, first M outputs k, and then it changes its mind to j. Hence, j has to be a correct guess for L, i.e., we have $L = L(G_j)$. Therefore, we directly obtain $L(G_k) \cap L \subseteq L(G_j) \cap L = L$. Hence, M monotonically infers $\mathcal{L}$. q.e.d.

Next we show that the monotonicity constraint can be traded versus efficiency. This is even true, if the relaxation is as weak as possible, i.e., if the requirement to learn monotonically is relaxed to weak-monotonic inference.

Theorem 7. *For every* $n \in \mathbb{N}$, $n \geq 2$ *there exists an indexed family such that*

$$\mathcal{L} \in (EMON_{n+1} \cap EWMON_n) \setminus CMON_n.$$

Proof. For the sake of presentation, we consider the case $n = 2$. The extension to all $n \geq 3$ may be easily obtained by applying the lifting technique of Lange and Zeugmann (1993b). The desired indexed family is defined as follows. Initially, we set $L_0 = \{a\}^+$. For all $k \in \mathbb{N}$, we set $L_{3k+1} = L_0 \cup \{a^k b\}$. By convention, a^0 equals the empty string. In order to define the remaining languages we distinguish the following cases:

Case 1. $\Phi_k(k)\uparrow$.

We set $L_{3k+3} = L_{3k+2} = L_{3k+1}$.

Case 2. $\Phi_k(k)\downarrow$.

Then, let $n = \Phi_k(k)$, and let $\hat{L}_k = \{a^z \mid 1 \leq z \leq n\} \cup \{a^k b\}$. We set:

$L_{3k+2} = \hat{L}_k \cup \{a^k c^n\}$, $L_{3k+3} = \hat{L}_k \cup \{a^k c^n,\ a^k d^n\}$.

After a bit of reflection it is not hard to see that $\mathcal{L} = (L_z)_{z\in\mathbb{N}}$ is an indexed family that is $EMON_3$ as well as $EWMON_2$–learnable.

It remains to show that $\mathcal{L} \notin CMON_2$. Suppose there are a hypothesis space $\mathcal{G}$ and an IIM M such that M $CMON_2$–learns $\mathcal{L}$ with respect to $\mathcal{G}$. By assumption M, in particular, infers the language L_0 from its text $t = a,\ a^2,\ a^3,\ \ldots$ Thus, there has to be a least index z such that $M(t_z) = j$ and $L(G_j) = L_0$. Given this index z the following recursive predicate ψ solves the halting problem.

Let $k \in \mathbb{N}$; the desired predicate ψ is defined as follows.

$\psi(k) =$ "Execute Instructions (A) and (B).

(A) For $m = 1,\ 2,\ \ldots$ simulate M, when fed $\hat{t}_{z+1+m} = \underbrace{a,\ \ldots,\ a^{z+1}}_{=t_z},\ a^k b,\ a,\ \ldots,\ a^m$, until the first y is found such that $j_{z+1+y} = M(\hat{t}_{z+1+y})$ and $a^k b \in L(G_{j_{z+1+y}})$.

(B) Test whether or not $\Phi_k(k) \leq z+1+y$. In case it is, output 1. Otherwise, output 0."

Obviously, if m tends to infinite then the limit $\hat{t}$ of $\hat{t}_{z+1+m}$ constitutes a text for L_{3k+1}. Since M has to infer the language L_{3k+1} from $\hat{t}$, it is easy to verify that the procedure defined above terminates for every $k \in \mathbb{N}$. Hence, ψ is recursive. It remains to show that $\varphi_k(k)$ is undefined, if $\psi(k) = 0$. Suppose the converse, i.e., $\psi(k) = 0$ and $\varphi_k(k)$ is defined. Therefore, $\Phi_k(k) = n > z+1+y$.

Recall that M has already performed at least one mind change when fed $\hat{t}_{z+1+y}$, namely from j to j_{z+1+y}. Since M monotonically infers L_{3k+1} from $\hat{t}$ and $a^k b \in L(G_{j_{z+1+y}})$, we obtain $L(G_{j_{z+1+y}}) \supseteq L_{3k+1}$. Otherwise, M violates the monotonicity constraint when inferring L_{3k+1} from its text $\hat{t}$. Consequently, $L(G_{j_{z+1+y}}) \neq L_{3k+2}$. Now, taking $\mathcal{L}$'s definition into account, it follows that $\hat{t}_{z+1+y}$ may also serve as an initial segment of a text for the language L_{3k+2} because $\Phi_k(k) = n > z+1+y$. Finally, since $L_{3k+2} \subset L_{3k+3}$, it is easy to verify that $\hat{t}_{z+1+y}$ can be extended to a text for L_{3k+3} such that M has to perform at least two additional mind changes in order to infer L_{3k+3} from this particular text. This contradicts our assumption that M monotonically infers $\mathcal{L}$ with at most two mind changes. Therefore, $\varphi_k(k)$ is undefined, if $\psi(k) = 0$. Hence, the predicate ψ solves the halting problem for the φ-system. q.e.d.

Refining *mutatis mutandis* the latter proof analogously as the demonstration of Theorem 2 has been extended to show Theorem 4, one obtains the following result.

Theorem 8. *For every $n \geq 2$ there exists an indexed family such that*

$\mathcal{L} \in (EMON_{n+1} \cap EWMON_2) \setminus CMON_n$.

The latter theorems allow the following interpretation. Removing the constraint to learn monotonically may considerably increase the efficiency of the learning process.

3.3. Weak-Monotonic Learning

Finally, we consider weak-monotonic learning. Possible relaxations include learning in the limit. We start with the following results which shed considerable light on the power of learning with at most one mind change.

Theorem 9.

(1) $MON_1 \setminus EWMON \neq \emptyset$,

(2) $ELIM_2 \setminus WMON \neq \emptyset$,

(3) $CMON_1 \setminus WMON \neq \emptyset$.

Proof. Lange and Zeugmann (1993b) proved $LIM_1 \setminus EWMON \neq \emptyset$, and recently Lange (1994) shows $CLIM_1 \setminus WMON \neq \emptyset$. Combining these results with Theorem 6 we directly get Assertion (1) and (3). Finally, for a proof of Assertion (2) we refer the reader to Lange (1994). q.e.d.

Consequently, relaxing the weak-monotonicity constraint may considerably increase the inference capabilities. However, the latter theorem dealt with indexed families that are themselves not weak-monotonically learnable. Therefore, it is only natural to ask whether or not there are indexed families that can be weak-monotonically inferred within an *a priori* bounded number of mind changes and that are learnable in the limit with less mind changes. The affirmative answer is provided by our next theorem. In particular, we show that unconstrained IIMs may be much more efficient than weak-monotonic machines. In Kinber (1994) a similar result concerning the learnability of classes of recursively enumerable languages has been shown. Modifying the construction underlying Kinber's proof the following result can be achieved.

Theorem 10. *For every $n \in \mathbb{N}$, $n \geq 2$, there exists an indexed family $\mathcal{L}$ such that*

$$\mathcal{L} \in (ELIM_2 \cap CWMON_{n+1}) \setminus CWMON_n .$$

For a detailed proof of the above theorem the interested reader is referred to Lange and Zeugmann (1994) which is a substantially revised version of the present paper. Up to now, it remains open whether or not a similar speed-up can be achieved in the exact and class preserving case, too.

We conclude this section with some remarks which may orient further investigations. As our results show a relaxation of the corresponding monotonicity demands may sometimes yield a *significant speed-up* of the learning process. Hence, it seems highly desirable to investigate necessary and sufficient conditions $\mathcal{C}_{csmon}$, $\mathcal{C}_{mon}$, and $\mathcal{C}_{wmon}$ allowing assertions of the following type.

Let LT as well as LT' be any learning type, and let $\mathcal{L} \in LT$. Then one may learn $\mathcal{L}$ more efficiently in the sense of LT' if and only if $\mathcal{C}_{lt'}$ is satisfied but $\mathcal{C}_{lt}$ is not.

Moreover, it would be very interesting to relate possible relaxations of our monotonicity requirements to problems studied in complexity theory. Recently, such an approach has been undertaken concerning consistent and inconsistent learning resulting in a proof for the superiority of an inconsistent learning algorithm (cf. Wiehagen and Zeugmann, 1994). We will see what the future brings concerning these problems.

4. References

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

Barzdin, Ya.M., and Freivalds, R.V. (1972), On the prediction of general recursive functions, *Sov. Math. Dokl.* **13**, 1224 – 1228.

Barzdin, Ya.M., Kinber, E.B., and Podnieks, K.M. (1974), Об ускорении синтеза и прогнозирования функций, *in* "Теория Алгоритмов и Программ," Vol. 1 (Ya.M. Barzdin, Ed.) Latvian State University, Riga, pp. 117 – 128.

Case, J., and Smith, C.H. (1983), Comparison of identification criteria for machine inductive inference, *Theoretical Computer Science* **25**, 193 - 220.

Gasarch, W.I., and Velauthapillai, M. (1992), Asking questions versus verifiability, *in* "Proceedings 3rd International Workshop on Analogical and Inductive Inference," (K.P. Jantke, ed.) Lecture Notes in Artificial Intelligence Vol. 642, pp. 197 – 213, Springer-Verlag, Berlin.

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

Hopcroft, J.E., and Ullman, J.D. (1969), "Formal Languages and their Relation to Automata," Addison-Wesley, Reading, Massachusetts.

Jantke, K.P. (1991), Monotonic and non-monotonic inductive inference, *New Generation Computing* **8**, 349 – 360.

Kapur, S., and Bilardi, G. (1992), Language learning without overgeneralization, *in* "Proceedings 9th Annual Symposium on Theoretical Aspects of Computer Science," (A. Finkel and M. Jantzen, Eds.), Lecture Notes in Computer Science Vol. 577, pp. 245 – 256, Springer-Verlag, Berlin.

Kinber, E.B. (1994), Monotonicity versus efficiency for learning languages from texts, *in* "Proceedings 5th Workshop on Algorithmic Learning Theory," (S. Arikawa and K.P. Jantke, Eds.), Lecture Notes in Artificial Intelligence Vol. 872, pp. 395 – 406.

LANGE, S. (1994), The representation of recursive languages and its impact on the efficiency of learning, *in* "Proceedings 7th Annual ACM Conference on Computational Learning Theory, New Brunswick, July 1994," pp. 256 – 267, ACM Press, New York.

LANGE, S., AND ZEUGMANN, T. (1993a), Monotonic versus non-monotonic language learning, *in* "Proceedings 2nd International Workshop on Nonmonotonic and Inductive Logic, December 1991, Reinhardsbrunn," (G. Brewka, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 659, pp. 254 – 269, Springer-Verlag, Berlin.

LANGE, S., AND ZEUGMANN, T. (1993b), Learning recursive languages with bounded mind changes, *International Journal of Foundations of Computer Science* **4**, 157 – 178.

LANGE, S., AND ZEUGMANN, T. (1993c), Language learning in dependence on the space of hypotheses, *in* "Proceedings 6th Annual ACM Conference on Computational Learning Theory," pp. 127 – 136, ACM Press, New York.

LANGE, S., AND ZEUGMANN, T. (1994), Trading monotonicity demands versus efficiency, *Bulletin of Informatics and Cybernetics*, to appear.

LANGE, S., ZEUGMANN, T., AND KAPUR, S. (1994), Monotonic and dual-monotonic language learning, *Theoretical Computer Science*, to appear.

MACHTEY, M., AND YOUNG, P. (1978), "An Introduction to the General Theory of Algorithms," North-Holland, New York.

MUKOUCHI, Y. (1992), Inductive inference with bounded mind changes, *in* "Proceedings 3rd Workshop on Algorithmic Learning Theory," Tokyo, Japan, JSAI, pp. 125 – 134.

MUKOUCHI, Y. (1994), Inductive inference of recursive concepts, Ph.D. Thesis, RIFIS, Kyushu University 33, RIFIS-TR-CS-82, March 25th.

OSHERSON, D., STOB, M., AND WEINSTEIN, S. (1986), "Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists," MIT-Press, Cambridge, Massachusetts.

SHINOHARA, T. (1990), Inductive Inference from Positive Data is Powerful, *in* "Proc. 3rd Annual Workshop on Computational Learning Theory," pp. 97 - 110, Morgan Kaufmann Publishers Inc.

WIEHAGEN, R. (1991), A thesis in inductive inference, *in* "Proceedings First International Workshop on Nonmonotonic and Inductive Logic," (J. Dix, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence 543, pp. 184 – 207, Springer-Verlag, Berlin.

WIEHAGEN, R., AND ZEUGMANN, T. (1994), Ignoring data may be the only way to learn efficiently, *Journal Experimental and Theoretical Artificial Intelligence* **6**, 131 – 144.