Monotonic Versus Non–monotonic Language Learning

Steffen Lange*
TH Leipzig
FB Mathematik und Informatik
PF 66
O–7030 Leipzig
steffen@informatik.th-leipzig.de

Thomas Zeugmann
TH Darmstadt
Institut für Theoretische Informatik
Alexanderstr. 10
W–6100 Darmstadt
zeugmann@iti.informatik.th-darmstadt.de

Abstract

In the present paper strong–monotonic, monotonic and weak–monotonic reasoning is studied in the context of algorithmic language learning theory from positive as well as from positive and negative data.

Strong–monotonicity describes the requirement to *only* produce better and better generalizations when more and more data are fed to the inference device. *Monotonic learning* reflects the eventual interplay between generalization and restriction during the process of inferring a language. However, it is demanded that for any two hypotheses the one output later has to be at least as good as the previously produced one with respect to the language to be learnt. *Weak–monotonicity* is the analogue of cumulativity in learning theory.

We relate all these notions one to the other as well as to previously studied modes of identification, thereby in particular obtaining a strong hierarchy.

1. Introduction

The process of hypothesizing a general rule from eventually incomplete data (e.g. examples, data obtained by performing experiments a.s.o.) is called inductive inference. In the philosophy of science inductive inference has attracted much attention during the last centuries. Some of the principles developed are very much alive in *algorithmic learning theory*, an emerging science starting with the seminal papers of Solomonoff (1964) and of Gold (1967). Computer scientists widely used their insight into the theory of computability to obtain a better and deeper understanding of processes performing inductive generalizations. In a beautiful paper Angluin and Smith (1987) survey the today state of the art in inductive inference.

*This research has been partially supported by the German Ministry for Research and Technology (BMFT) under grant no. 01 IW 101.

The present paper mainly deals with formal language learning, a field in which many interesting and sometimes surprising results have been elaborated within the last decades (cf. e.g. Osherson, Stob and Weinstein (1986), Case (1988), Fulk (1990), Jain and Sharma (1990)). One of the central questions studied so far is whether or not various restrictions on the behavior of a learner do limit the learning capabilities of machines. We shall continue along this line. Before explaining what requirements we want to deal with, let us recall the general situation investigated in language learning. The learner is provided with eventually incomplete information concerning the language to be inferred and has to produce, from time to time, a hypothesis about the phenomenon to be learnt. The information given may contain only *positive examples*, i.e., only strings from the language to be recognized, as well as both *positive and negative examples*, i.e., the learner is fed more and more strings over the underlying alphabet which are classified with respect to their containment to the unknown language. The space of hypotheses may vary from a particular set of acceptors or grammars to sets of characteristic functions. Moreover, the sequence of hypotheses is required to converge in some specified sense to a hypothesis correctly describing the object to be learnt. There are many possible requirements to the sequence of all created hypotheses. What we like to present in the sequel is an almost complete investigation of the influence of various monotonicity conditions originally introduced by Jantke (1991A) and Wiehagen (1990). The main underlying question can be posed as follows: Would it be possible to learn the unknown language in using more and more information about it and thereby producing *only* better and better generalizations? In its strongest interpretation that means we are required to infer a sequence of hypotheses describing an augmenting chain of languages, i.e., $L_i \subseteq L_j$ iff L_j is guessed later than L_i (cf. Definition 3 (A)).

This requirement finds its analogue in classical logics where an enlargement of the set of assumptions always leads to an eventually larger set of derivable theorems. However, as Jantke (1991A, 1991B) has shown, in the setting of recursive function learning, strong–monotonicity considerably restricts the inferring power. Starting from the observation that "better" has to be interpreted with respect to the goal of the learning process, i.e., with respect to the language L having to be learnt, Wiehagen (1990, 1991) proposed the following modification: Instead of demanding $L_i \subseteq L_j$ we only require $L_i \cap L \subseteq L_j \cap L$ iff L_j appears later in the sequence of created hypotheses than L_i does (cf. Definition 3 (B)). Intuitively speaking, now monotonicity means that the new hypothesis is never allowed to destroy something what a previously generated guess already *correctly* reflects.

The third version of monotonicity, which we call weak–monotonicity, is derived from non–monotonic logics and may be interpreted as the analogue of cumulativity. Consequently, we only require $L_i \subseteq L_j$ as long as the data obtained after having produced L_i do not contradict L_i (cf. Definition 3 (C)).

As it turns out, if the learning process is performed with *positive and negative examples* then weak–monotonicity does *not restrict* the inferring power as long as the space of hypotheses is suitable chosen (cf. Theorem 6). In case one learns from positive data alone, we show that weak–monotonically working learning devices are exactly as powerful as conservatively working ones. A learning device is said to be *conservative* if and only if it only performs justified mind changes, i.e., it changes its actual hypothesis only in case it is contradicted by new data (cf. Angluin (1980A)).

All other notions of monotonicity do immediately lead to a severe limitation of the learning power, as we shall show. Finally, in order to avoid confusion it should be mentioned that there is yet another notion of monotonic learning (cf. Osherson, Stob and Weinstein (1986), Fulk (1990)) which is slightly different from ours.

The paper is structured as follows. Section 2 presents preliminaries, i.e., notations, definitions and motivations. In section 3 we exclusively deal with identification from positive data. The monotonic inference with positive and negative data is studied in section 4. Section 5 is devoted to the question whether natural families of languages may be inferred monotonically. Finally, we give a summary and present open problems. All references are listed in section 7.

2. Preliminaries

By $N = \{1, 2, 3, ...\}$ we denote the set of all natural numbers. In the sequel we assume familarity with formal language theory (cf. e.g. Bucher and Maurer (1984)). By Σ we denote any fixed finite alphabet of symbols. Let Σ^* be the free monoid over Σ. The length of a string $w \in \Sigma^*$ is denoted by $|w|$. Any subset $L \subseteq \Sigma^*$ is called a language. By $co - L$ we denote the complement of L, i.e., $co - L = \Sigma^* \setminus L$. Let L be a language and $t = s_1, s_2, s_3, ...$ a sequence of strings from Σ^* such that $range(t) = \{s_k \mid k \in N\} = L$. Then t is said to be a *text* for L or, synonymously, a positive presentation. Furthermore, let $i = (s_1, b_1), (s_2, b_2), ...$ be a sequence of elements of $\Sigma^* \times \{+, -\}$ such that $range(i) = \{s_k \mid k \in N\} = \Sigma^*$, $i^+ = \{s_k \mid (s_k, b_k) = (s_k, +), k \in N\} = L$ and $i^- = \{s_k \mid (s_k, b_k) = (s_k, -), k \in N\} = co - L$. Then we refer to i as an *informant*. If L is classified via an informant then we also say that L is represented by positive and negative data. Moreover, let t, i be a text and an informant, respectively, and let x be a number. Then t_x, i_x denote the initial segment of t and i of length x, respectively, e.g., $i_3 = (s_1, b_1), (s_2, b_2), (s_3, b_3)$. Let t be a text and let $x \in N$. Then we set $t_x^+ = \{s_i \mid i \leq x\}$. Furthermore, by i_x^+ and i_x^- we denote the sets $\{s_k \mid (s_k, +) \in i, k \leq x\}$ and $\{s_k \mid (s_k, -) \in i, k \leq x\}$, respectively.

Following Angluin (1980A) we restrict ourselves to deal exclusively with indexed families of recursive languages defined as follows:
A sequence $L_1, L_2, L_3, ...$ is said to be an *indexed family* $\mathcal{L}$ of recursive languages provided all L_j are non–empty and there is a recursive function f such that for all numbers j and all strings $w \in \Sigma^*$ we have

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

As an example we consider the set $\mathcal{L}$ of all context–sensitive languages over Σ. Then $\mathcal{L}$ may be regarded as an indexed family of recursive languages (cf. [4]). In the sequel we often denote an indexed family and its range by the same symbol $\mathcal{L}$. What is meant will be clear from the context.

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 (an informant i) and it either requires the next input string, or it first outputs a hypothesis, i.e., a number encoding a certain computer program, and

then it requires the next input string (cf. e.g. Angluin (1980A)).

At this point we have to clarify what space of hypotheses we should choose, thereby also specifying the goal of the learning process. Gold (1967) and Wiehagen (1977) pointed out that there is a difference in what can be inferred in dependence on whether we want to synthesize in the limit grammars (i.e., procedures generating languages) or decision procedures, i.e., programs of characteristic functions. Case and Lynes (1982) investigated this phenomenon in detail. As it turns out, IIMs synthesizing grammars can be more powerful than those ones which are requested to output decision procedures. Moreover, several surprising results distinguishing language learning from inductive inference of recursive functions are originated in choosing grammars as space of hypotheses instead of characteristic functions. However, in the context of identification of indexed families both concepts are of equal power. Nevertheless, we decided to require the IIMs to output grammars. This decision has been caused by the fact that there is a big difference between the possible monotonicity requirements. A straightforward adaptation of the approaches made in inductive inference of recursive functions directly yields analogous requirements with respect to the correseponding characteristic functions of the languages to be inferred. On the other hand, it is only natural to interpret monotonicity with respect to the language to be learnt, i.e., to require containement of languages as described in the introduction. As it turned out, the latter approach increases considerably the power of monotonic language learning. Furthermore, since we exclusively deal with indexed families $\mathcal{L} = (L_j)_{j \in N}$ of recursive languages we always take as space of hypotheses an enumerable family of grammars $G_1, G_2, G_3, ...$ over the terminal alphabet Σ satisfying $\mathcal{L} = \{L(G_j) \mid j \in N\}$. Moreover, we always require that membership in $L(G_j)$ is uniformly decidable for all $j \in N$ and all strings $w \in \Sigma^*$. As it turns out, it is sometimes very important to choose the space of hypotheses appropriately in order to achieve the desired learning goal. Then the IIM outputs numbers j which we interpret as G_j.

A sequence $(j_x)_{x \in N}$ of numbers is said to be convergent in the limit if and only if there is a number j such that $j_x = j$ for almost all numbers x.

Definition 1, (Gold (1967)) *Let $\mathcal{L}$ be an indexed family of languages, $L \in \mathcal{L}$, and let $(G_j)_{j \in N}$ be a space of hypotheses. An IIM M $LIM-TXT$ ($LIM-INF$)-identifies L on a text t (an informant i) iff it almost always outputs a hypothesis and the sequence $(M(t_x))_{x \in N}$ ($M(i_x))_{x \in N}$) converges in the limit to a number j such that $L = L(G_j)$.*
Moreover, M $LIM-TXT$ ($LIM-INF$)-identifies L, iff M $LIM-TXT$ ($LIM-INF$)-identifies L on every text (informant) for L. We set:
$LIM-TXT(M) = \{L \in \mathcal{L} \mid M\ LIM-TXT-identifies\ L\}$ and define $LIM-INF(M)$ analogously.
Finally, let $LIM-TXT$ ($LIM-INF$) denote the collection of all families $\mathcal{L}$ of indexed families of recursive languages for which there is an IIM M such that $\mathcal{L} \subseteq LIM-TXT(M)$ ($\mathcal{L} \subseteq LIM-INF(M)$).

Definition 1 could be easily generalized to arbitrary families of recursively enumerable languages (cf. [18], [8]). Nevertheless, we exclusively consider the restricted case defined above, since our motivating examples are all families of recursive languages. Note that, in general, it is not decidable whether or not M has already inferred L. In case M produces only a *single* and *correct* guess after having been fed an initial segment of a

text t (or informant i) and *stops* then, we say that M *finitely* infers L on t (on i). M $FIN-TXT$ $(FIN-INF)$–infers L iff it finitely infers L on every text (informant). The resulting identification type is denoted by $FIN-TXT$ $(FIN-INF)$.

Next we want to formally define strong–monotonic, monotonic and weak–monotonic inference. In order to do this, first we have to explain what does it mean that an IIM *consistently* identifies a language. Consistent IIM have been introduced by Barzdin (1974). Intuitively speaking, consistency means that the IIM does correctly reflect the part of the language the IIM has already been fed with when it produces a guess.

Definition 2, Barzdin ((1974)) *An IIM M $CONS-TXT$ $(CONS-INF)$–identifies L on a text t (an informant i) iff*
(1) M $LIM-TXT$ $(LIM-INF)$*–identifies L on t (on i).*

(2) *Whenever M on t_x (i_x) produces a hypothesis j_x then $range(t_x) \subseteq L(G_{j_x})$ $(i_x^+ \subseteq L(G_{j_x})$ and $i_x^- \subseteq co-L(G_{j_x}))$.*

M $CONS-TXT$ $(CONS-INF)$–identifies L iff M $CONS-TXT$ $(CONS-INF)$–identifies L on every text t (informant i).
By $CONS-TXT(M)$ $(CONS-INF(M))$ we denote the set of all languages which M does $CONS-TXT$ $(CONS-INF)$–identify. $CONS-TXT$ and $CONS-INF$ are analogously defined as above.

Definition 3, Jantke ((1991A), Wiehagen (1991)) *An IIM M is said to identify a language L from text (informant)*
(A) *strong–monotonically*

(B) *monotonically*

(C) *weak–monotonically*

iff
M $LIM-TXT$ $(LIM-INF)$–identifies L and for any text t (informant i) 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} (i_x and i_{x+k}), for some $k \geq 1, k \in N$, 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 $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$ (if $i_{x+k}^+ \subseteq L(G_{j_x})$ and $i_{x+k}^- \subseteq co-L(G_{j_x})$ then $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$).*

Remark: (C) in particular means that M has to work strong–monotonically as long as its guess j is consistent with the data fed to M after M has output j.

By $SMON-TXT$, $SMON-INF$, $MON-TXT$, $MON-INF$, $WMON-TXT$, $WMON-INF$ we denote the family of all thoses sets $\mathcal{L}$ of languages for which there is an IIM inferring it strong–monotonically, monotonically, and weak–monotonically from text t or informant i, respectively.

This seems to be a good place to give an example showing what inferring power our actual choice of the space of hypotheses really implies. Let $L \subseteq \Sigma^*$ be any arbitrarily fixed *infinite* context–sensitive language. By $\mathcal{L}_{fin}$ we denote the set of all finite languages over Σ. Then we set $\mathcal{L} = \{L \cup L_{fin} \mid L_{fin} \in \mathcal{L}_{fin}\}$. In case we would require the IIMs to output

programs of characteristing functions one immediately obtains from Jantke (1991A) that, even on informant, $\mathcal{L}$ cannot be learnt strong–monotonically. On the other hand, in our setting $\mathcal{L}$ is strong–monotonically learnable, even on *text*.

The IIM M performing the inference process initially outputs a grammar G of L. Let j_x be $M's$ guess after having received t_x. If $t_x^+ \subset t_{x+1}^+$ then M tests whether the new string w belongs to $L(G_{j_x})$. In case it does, we set $j_{x+1} = j_x$. Otherwise M outputs a canonical number j_{x+1} of the grammar $G_{j_x} \cup \{\sigma \to w\}$ where σ denotes the distinguished non–terminal of G_{j_x} from which every derivation starts. Since for any L' in $\mathcal{L}$ there is a $L_{fin} \in \mathcal{L}_{fin}$ such that $L' = L \cup L_{fin}$ the "else" case can happen at most finitely often. Hence, M converges and by construction M does work strong–monotonically.

Finally in this section we define *conservatively* working IIMs.

Definition 4, (Angluin (1980A))
An IIM M CONSERVATIVE–TXT $(CONSERVATIVE–INF)$–identifies L on text t (on informant i), iff for every text t (informant i) the following conditions are satisfied:
(1) $L \in LIM - TXT(M)$ $(L \in LIM - INF(M))$

(2) If M on input t_x makes the guess j_x and then makes the guess $j_{x+k} \neq j_x$ at some subsequent step, then $L(G_{j_x})$ must fail to contain some string from t_{x+k} ($L(G_{j_x})$ must fail either to contain some string $w \in i_{x+k}^+$ or it generates some string $w \in i_{x+k}^-$).

$CONSERVATIVE–TXT(M)$, $CONSERVATIVE–INF(M)$ as well as $CONSERVATIVE–TXT$ and $CONSERVATIVE–INF$ are defined in an analogous manner as above.

Intuitively speaking, a conservatively working IIM performs *exclusively* justified mind changes.

Sometimes we want to combine some postulates, e.g. strong–monotonicity and consistency. That is denoted by e.g. $SMON - CONS - TXT$ or $SMON - CONS - INF$ and means that the particular IIM which has to perform the inferring process is required to work strong–monotonically *and* consistently on every text and informant, respectively.

In the next section we present results concerning text identification. Note that in all what follows $\subset$ denotes proper set inclusion and $\#$ denotes incomparability of sets.

3. Monotonic Inference on Text

Our first theorem actually shows that there is a strong hierarchy between the different notions of monotonicity.

Theorem 1 $SMON - TXT \subset MON - TXT \subset WMON - TXT$

Proof. $SMON - TXT \subseteq MON - TXT$ is an immediate consequence of Definition 3. The part $MON - TXT \setminus SMON - TXT \neq \emptyset$ is proved via the following indexed family of languages:
Let $(i_1, ..., i_k)$ be any collection of pairwise distinct numbers. We set $L_{i_1,...,i_k} = (\{a\}^* \setminus \{a^{i_1}, ..., a^{i_k}\}) \cup \{b^{i_1}, ..., b^{i_k}\}$ and define $\mathcal{L}$ to be the family of all languages $L_{i_1,...,i_k}$, where $(i_1, ..., i_k) \in N^k, k \in N, k \geq 1, i_j \neq i_l$ if $j \neq l$ for all $j, l \in \{1, ..., k\}$. We omit the details.

Next we show $MON - TXT \subseteq WMON - TXT$.

The problem we have to deal with is caused by the fact that any monotonically working IIM may eventually output an overgeneralized hypothesis j, i.e., $L \subset L(G_j)$, while a weakly-monotonically working IIM may not. The proof is done using the following claim:

Claim: Let $\mathcal{L}$ be any indexed family, and let M be any monotonically working IIM inferring $\mathcal{L}$. Let $L \in \mathcal{L}$ and let t be any fixed text for L. If M on t produces a guess j such that $L \subset L(G_j)$, then $L(G_j) \notin \mathcal{L}$.

Suppose the converse, i.e., there is an index x such that $M(t_x) = j$, $L \subset L(G_j)$ and $L(G_j) \in \mathcal{L}$. Then t_x is also an initial segment of a text for $L(G_j)$. On the other hand, t_x possesses an enlargement t_{x+k} such that $t^+_{x+k} \subseteq L$ and $j_{x+k} := M(t_{x+k})$ satisfies $L(G_{j_{x+k}}) = L$. Again, t_{x+k} is an initial segment of a text for $L(G_j)$. Consequently, there should be an enlargement of t_{x+k}, say t_{x+k+z}, with strings from $L(G_j)$ such that $j_{x+k+z} := M(t_{x+k+z})$ and $L(G_{j_{x+k+z}}) = L(G_j)$.
Since

$$L(G_j) \supset L = L(G_{j_{x+k}}) \cap L(G_j) \subset L(G_{j_{x+k+z}}) \cap L(G_j) = L(G_{j_{x+k+z}})$$

we have found a text t for $L(G_j)$ on which M does not identify $L(G_j)$ monotonically, a contradiction. This proves the claim.

On the other hand, since we have required $\mathcal{L} = \{L(G_j) \mid j \in N\}$ for any space of hypotheses $(G_j)_{j \in N}$ the wanted inclusion follows immediately by the claim made above.

The remaining part, i.e., $WMON - TXT \setminus MON - TXT \neq \emptyset$ can be proved using the following indexed family $\mathcal{L}_{wmon}$:
Let $L_1 := \{a\}^*$ and for $i > 1$ set $L_i := \{a^z \mid z < i\} \cup \{b^z \mid z \geq i\}$ as well as $L_{i,j} := \{a^z \mid z < i\} \cup \{b^z \mid i \leq z < j\} \cup \{a^z \mid z \geq j\}$ for all $i, j \in N$ with $i < j$, and $i, j > 1$. $\mathcal{L}_{wmon}$ is defined to be the collection of all $L_i, L_{i,j}$. For the demonstration of $\mathcal{L}_{wmon} \in WMON - TXT \setminus MON - TXT$ the reader is referred to the proof of Theorem 8, claim C and D, where the slightly stronger result $\mathcal{L}_{wmon} \in WMON - TXT \setminus MON - INF$ is shown.

q.e.d.

Moreover, $SMON - TXT$ and $MON - TXT$ may already be separated on sets of regular languages. At this point it is only natural to ask what are lower and upper bounds of this hierarchy. The answer is given by the next theorem.

Theorem 2

(1) $FIN - TXT \subset SMON - TXT$ *and*

(2) $WMON - TXT = CONSERVATIVE\text{–}TXT$

Proof. By our definition, if $\mathcal{L} \in FIN - TXT(M)$, then M outputs a single guess. Hence $FIN - TXT \subseteq SMON - TXT$.

The remaining part $SMON - TXT \setminus FIN - TXT \neq \emptyset$ can be directly obtained by proving $\mathcal{L} \notin FIN - TXT$, where $\mathcal{L} = \{L \cup L_{fin} \mid L_{fin} \in \mathcal{L}_{fin}\}$, i.e., the family defined before presenting Definition 4.

Looking at assertion (2) we directly see that $CONSERVATIVE\text{–}TXT \subseteq WMON - TXT$, since a conservatively working IIM only performs justified mind changes.
Now suppose $\mathcal{L} \in WMON - TXT(M')$. We have to construct an IIM M such that $\mathcal{L} \in CONSERVATIVE\text{–}TXT(M)$. M is easily obtained from M' by simply adding a

consistency test, i.e., let $j_x := M(t_x)$ and $w \in t^+_{x+k-1}$ for some $k = 1, 2, ...$ Then test whether or not $w \in L(G_{j_x})$ for all w. In case it is, M repeats j_x as its hypothesis. Otherwise, it outputs $M'(t_{x+k})$.

q.e.d.

As an immediate consequence of Angluin's (1980A) Theorem 4 one directly obtains:

Corollary 3 $WMON - TXT \subset LIM - TXT$

Next we want to deal with the combination of monotonic and consistent inference. Considering a particular family of languages, the so–called pattern languages (cf. section 5) Jantke(1991B) has shown that weak–monotonicity and consistency may be combined without any limitations concerning the inference power, while strong–monotonicity and consistency cannot. However, the negative result is mainly based on the particular choice of the space of hypotheses, i.e., the set of all patterns, as a careful analysis of his proof shows. Nevertheless, we have been surprised to obtain the following theorem:

Theorem 4

(1) $WMON - CONS - TXT = WMON - TXT$

(2) $MON - CONS - TXT = MON - TXT$

(3) $SMON - CONS - TXT = SMON - TXT$

Our result in particular shows that the choice of the space of hypotheses may not only influence the inferring power at all but may or may not allow the combination of postulates of naturalness.

4. Monotonic Inference on Informant

The first theorem in the section again establishes a hierarchy between the different types of monotonicity.

Theorem 5 $SMON - INF \subset MON - INF \subset WMON - INF$

Moreover, while the proper set inclusions remain valid if one considers only families of regular languages, the next theorem shows the power of weak–monotonic IIM. However, the power of weak–monotonic IIMs again may be characterized as to coincide with the inferring capabilities of conservative ones.

Theorem 6

(1) $FIN - INF \subset SMON - INF$ *and*

(2) $WMON - INF = CONSERVATIVE\text{–}INF = LIM - INF$

Moreover, in our setting, i.e., considering only indexed families of recursive languages, we even obtain that $LIM - INF = CONS - INF$.
The next theorem shows that we again may combine monotonicity and consistency without limiting the learning power.

Theorem 7

(1) $SMON - CONS - INF = SMON - INF$

(2) $MON - CONS - INF = MON - INF$

(3) $WMON - CONS - INF = WMON - INF$

Now we want to compare monotonic inference from positive data with monotonic inference from both, positive and negative data. In his seminal paper Gold (1967) has shown that the inference from positive data alone is much weaker than inference on informant. Consequently, one might well expect that monotonic inference on text is less powerful than learning monotonically on informant. However, the more interesting question is whether one can strengthen the monotonicity requirement in case one changes from positive data to positive and negative data. The opposite direction of this problem is wether the weakening of the notion of monotonicity allows the inference of sets of languages on text which may only be inferred on informant in the stronger monotonic case. Our next theorem shows that all these things are almost always different ones, i.e., in general one cannot trade monotonicity versus information presentation.

Theorem 8

(1) $MON - INF \# WMON - TXT$

(2) $MON - INF \# LIM - TXT$

(3) $SMON - INF \# MON - TXT$

(4) $FIN - INF \# SMON - TXT$

Proof. The only part we prove here is assertion (2); the rest will be presented in the full version of the paper.

We set $L_1 = \{a\}^*$, $L_k = L_1 \setminus \{a^{k-1}\}$ for $k > 1$, and define $\mathcal{L} = L_1, L_2, ...$ Obviously, $\mathcal{L}$ is an indexed family.

Claim A: $\mathcal{L} \in MON - INF$

The wanted IIM M works as follows: Initially, it outputs a canonical grammar j_1 for L_1. As long as some $(a^k, -)$ does not appear in the informant, the machine M outputs j_1. In case it does, M performs a mind change and outputs a canonical grammar for L_k. This hypothesis is then repeated in any subsequent step. Since $L_1 \cap L_k = L_k$ for all $k \geq 1$, the machine M works monotonically.

Claim B: $\mathcal{L} \notin LIM - TXT$

Due to Theorem 1 of Angluin(1980A) it suffices to show that there is no finite tell–tale for L_1. Suppose the converse, i.e., there is a recursively enumerable finite set T_1 satisfying:

(1) $T_1 \subset L_1$

(2) for all $j \geq 1$, if $T_1 \subseteq L_j$, then $L_j \not\subset L_1$

Let $z = max\{|w| \mid w \in T_1\}$. Then, in accordance with the definition of the languages L_k we immediately get $L_{z+2} \supseteq T_1$. Moreover, $L_{z+2} \subset L_1$, yielding a contradiction to condition (2). This proves $MON - INF \setminus LIM - TXT \neq \emptyset$.

In order to prove the opposite direction, i.e, $LIM - TXT \setminus MON - INF \neq \emptyset$ we use the indexed family $\mathcal{L}_{wmon}$ mentioned in the proof of Theorem 1.

Claim C: $\mathcal{L}_{wmon} \in WMON - TXT$

The wanted IIM M is informally defined as follows: Let $L \in \mathcal{L}_{wmon}$, and let t be any text for L. M initially outputs a canonical grammar for the languaguge $\{a\}^*$. In case the first string b^z appears in t the machine M outputs a grammar for L_z. This guess remains unchanged until

(α) there appears a sting b^k in the text t with $k < z$ or
(β) there appears a string a^m in the text t with $z < m$.
If (α) happens first, then M outputs a grammar for L_k, in case (β) happens first $M's$ new output is $L_{z,m}$, and finally, if (α) happens after (β) has already happened, then M produces a canonical grammar for $L_{k,m}$.
Looking at the definition of M we immediately observe that M changes its mind only in case if its current guess is inconsistent with the new data. Consequently, $\mathcal{L}_{wmon} \in WMON - TXT(M)$
Finally we have to prove the following claim:

Claim D: $\mathcal{L}_{wmon} \notin MON - INF$
Suppose the converse, i.e., assume $\mathcal{L}_{wmon} \in MON - INF(M)$ for some IIM M. Let i be any informant for $\{a\}^*$. Since $L_1 \in \mathcal{L}_{wmon}$ there must be an x such that $j_x = M(i_x)$ and $L(G_{j_x}) = L_1$. Next we successively enlarge i_x by $(b^z, +)$, where $z \geq y = max\{|w| \mid w \in i_x^+ \cup i_x^-\} + 1$. Consequently, all i_{x+k} are initial segments of an informant for L_y. Hence there must be a number k such that M on i_{x+k} outputs a grammar j_{x+k} being correct for L_y. But now we may enlarge i_{x+k} in a canonical manner to an informant i_{fool} for $L_{y,m}$ where $m = max\{|w| \mid w \in i_{x+k}^+ \cup i_{x+k}^-\} + 1$. It is easy to see that M either does not work monotonically on i_{fool} or it does not infer $L_{y,m}$. This proves the claim, and so we have proved assertion (2).

q.e.d.

Next we ask whether assertion (2) of the latter theorem can be sharpened to $SMON - INF \setminus LIM - TXT \neq \emptyset$. The answer to this question put the hardness of strong–monotonic inference in the right perspective.

Theorem 9 $SMON - INF \subset WMON - TXT$

Proof. It suffices to show $SMON - INF \subseteq WMON - TXT$ since by Theorem 8, assertion (1), one directly obtains $WMON - TXT \setminus SMON - INF \neq \emptyset$. Let $\mathcal{L} \in SMON - INF(M)$, for some IIM M. Without loss of generality we may assume M to be total, i.e., to be defined on any finite sequence of strings from $\Sigma^* \times \{+, -\}$. This can be seen as follows.

Let $\hat{M}$ be any IIM inferring $\mathcal{L}$ strong-monotonically on informant. Moreover, let $i_x = (s_1, b_1), ..., (s_x, b_x)$ be any finite sequence. Then the IIM M first checks wether or not there is a language $L_k \in \mathcal{L}$ such that $k \leq x$, $i_x^+ \subseteq L_k$, and $i_x^- \subseteq co - L_k$. In case such L_k does not exists, M requests the next input. Otherwise, M on input i_x behaves exactly as $\hat{M}$ does. Note that in the latter case $\hat{M}$ has to be defined, since otherwise it would fail on a possible initial segment of some informant for L_k. Moreover, M infers $\mathcal{L}$ strong-monotonically. Let $L \in \mathcal{L}$, $y = \mu z[L_z = L]$, and i any informant of L. Consequently, on input i_x, for any $x \geq y$, M finds at least one language L_k such that $k \leq x$, $i_x^+ \subseteq L_k$, and $i_x^- \subseteq co - L_k$, i.e., it works exactly as $\hat{M}$ does. Furthermore, on input i_x, for any $x < y$, it either behaves as $\hat{M}$ does or it possibly ships over some hypotheses $\hat{M}$ possibly might have computed. Since set inclusion is a transitive relation, M works strong monotonically.

Let us continue. We have to construct an IIM M' identifying $\mathcal{L}$ weak-monotonically on text. This is done as follows: Let $L \in \mathcal{L}$ and let $t = (s_j)_{j \in N}$ be an arbitrarily fixed text for L. Then the wanted machine M' is defined as follows:

$M'(t_x) =_{df}$ "rearrange the strings contained in t_x in lexicographical order without repititions. Let $t'_k = w_1, ..., w_k$, $k \leq x$ the sequence obtained.
Compute $A_x := \{(w_1, +), ..., (w_k, +)\} \cup \{(v, -) \mid v \in \Sigma^* \setminus t_x^+, \quad |v| \leq max\{x, |w_k|\}\}$
Determine i_z by rearranging the elements of A_x in lexicographical order with respect to the first component.
(* Note that i_z is not necessarily an initial segment of an informant of L, since it may contain some $(v, -)$ such that $v \in L$. However, if x is large enough, then there is an $l \in N$ such that i_m is an initial segment of an informant for L for any $m \leq l$ *)
Compute $M(i_0)$, $M(i_1)$, ... until M outputs the first hypothesis j_r, where $r \leq z$, satisfying $i_z^+ \subseteq L(G_{j_r})$ and $i_r^- \subseteq co - L(G_{j_r})$.
(*Observe that every string $(w, +) \in i_z^+$ belongs indeed to L by our construction*)
In case j_r has been found, output it. Otherwise, output nothing, and request t_{x+1}."

It remains to show that $L \in WMON - TXT(M')$. We have to prove that

(1) M' converges to a number j such that $L = L(G_j)$, and

(2) M' works weak–monotonically

In order to show (1) remember that M works strong–monotonically on every informant for L. Consequently, M works strong–monotonically on the lexicographical ordered informant i. Hence, there is a minimal index m such that $M(i_m) = j_m$ and $L(G_{j_m}) = L$. Moreover, since t is a text for L there is an index x satisfying $i_m^+ \subseteq t_{x+k}^+$ and $i_m^- \subseteq range(i_{z+k})$ for all $k \geq 1$, where i_{z+k} is the initial segment of an informant obtained from t_{x+k} as described above. Since $L(G_{j_m}) = L$ we get:

$$i_{z+k}^+ \subseteq L(G_{j_m}) \; and \; i_m^- \subseteq co - L(G_{j_m}) \; for \; every \; k \geq 1.$$

Thus, $M'(t_x) = M'(t_{x+k})$ for all $k \geq 1$, i.e., M' converges.

We proceed in showing (2). Since M works strong–monotonically on i_m we have: Whenever M on i_l, $l \leq m$ outputs a guess j_l, then $L(G_{j_l}) \subseteq L$.
Now it suffices to prove that every mind change of M' is forced by an inconsistency with the text t. Let j_r be guess produced by M', i.e., in simulating M an initial segment i_r has been found such that $M(i_r) = j_r$ and $i_z^+ \subseteq L(G_{j_r})$ as well as $i_r^- \subseteq co - L(G_{j_r})$. We distinguish the following two cases:
Case (α): i_r is an initial segment of i_m

Then $L(G_{j_r}) \subseteq L$. Consequently, if j_r is not the hypothesis M' converges to, then there has to be a k such that $i_{z+k}^+ \not\subseteq L(G_{j_r})$. But that means the text t has to contain some word $w \notin L(G_{j_r})$. Hence, this mind change is forced by an inconsistency.
Case (β): i_r is not an initial segment of i_m

Then i_r must contain a string $(v, -)$ such that $v \in L$. Since $i_r^- \subseteq co - L(G_{j_r})$ it follows that $v \notin L(G_{j_r})$. Consequently, in this case j_r is changed, if either a string s appears in t such that $(s, -) \in i_r^-$ or $s \in i_z^+$ and $s \notin L(G_{j_r})$. In both cases the mind change is due to inconsistency.

q.e.d.

Summarizing these results with those ones obtained previously, we get the following

figure.

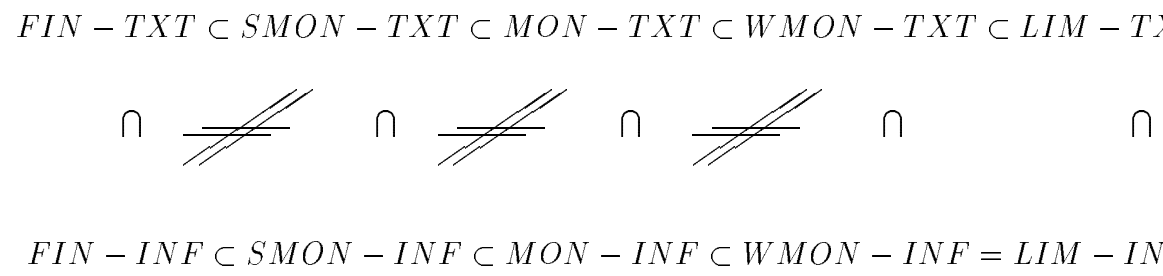

Figure 1

After having clarified the principal relations between the different notions of monotonicity we want to deal with the question what "natural" sets of languages may or may not be inferred monotonically. This is done in the next section.

5. Monotonic Learning of Natural Families of Languages

In this section we mainly deal with the question whether sets of pattern languages may be learnt strong–monotonically or monotonically on text or on informant. Pattern languages have been introduced by Angluin (1980B), thereby proving that the whole class of pattern languages can be inferred in the limit ¿from positive data. Subsequently, Shinohara (1982) dealt with polynomial time learnability of subclasses of pattern languages. Nix (1983) outlined interesting applications of pattern inference algorithms. Recently, Kearns and Pitt (1989) as well as Ko et al. (1990) studied intensively pattern inference from positive and negative data in the PAC–learning model.

So let us define what are a pattern and a pattern language. Let Σ be any non–empty finite alphabet, and let $X = \{x_1, x_2, ...\}$ a set of variables. *Patterns* are non–empty strings of variables and constants from Σ, e.g. x_1x_2, $ax_1x_7bx_3$ are patterns. $L(p)$, the language generated by pattern p is the set of strings which can be obtained by substituting non–null strings from Σ^* for the variables of the pattern p. PAT denotes the set of all pattern languages. Finally, let DPAT be the family of all languages L for which there are patterns p_{i_1}, p_{i_2}, ..., p_{i_k} such that $L = \bigcap_{j=1}^{n} L(p_{i_j})$.

Using the techniques developed in Lange and Wiehagen (1990) one gets the following theorem:

Theorem 10

(1) $PAT \in FIN-INF$

(2) $PAT \notin FIN-TXT$

Proof. In order to prove assertion (1) let $L \in PAT$ and let i be any informant for L. The wanted IIM M is informally defined as follows:

M requests more and more inputs i_x until it finds the $k \in N$ such that the following conditions are satisfied:
(α) Every $w \in \Sigma^*$ with $|w| < k$ belongs to i_x^-.
(β) There is a $w \in \Sigma^*$ of length k that is contained in i_x^+, and all $w \in \Sigma^*$ of length k are classified in i_x.

Having found this k the machine M starts the procedure described in Lange/ Wiehagen (1990) with all strings $w \in L$ of length k and generates the uniquely determined canonical pattern p such that $L = L(p)$. Finally, it outputs p and stops.

The proof of assertion (2) can be directly obtained by observing that every text for any pattern language L is also a text for $L(x_1)$. Hence, $L(x_1)$ cannot be finitely inferred on text. We omit the details.

q.e.d.

On the other hand, Jantke (1991B) pointed out that $PAT \notin SMON-CONS-TXT$. However, this result is mainly based on the requirement that in his setting any IIM is only allowed to output patterns. If the space of hypotheses is restricted to PAT, then one can reduce the question whether at least $PAT \in SMON - TXT$ to the decidability of inclusion of pattern languages, i.e., to decide whether or not $L(p) \subseteq L(q)$ for any given patterns p and q. Unfortunately, the latter problem, first stated in Angluin (1980B), is open.

Therefore it is only natural to ask whether an appropriate choice of the space of hypotheses may help to overcome the difficulties in designing an IIM that infers PAT strong–monotonically. Note that the next theorem is mainly based on an idea communicated to us by H.R. Beick (1991).

Theorem 11 *There is a space of hypotheses such that $DPAT \in SMON - TXT$.*

Proof. The space of hypotheses is a subset of all context–sensitive grammars obtained as follows: It is not hard to see that there is an effective procedure translating every pattern p into a context–sensitive grammar G_p such that $L(p) = L(G_p)$. Moreover, the set of all context–sensitive grammars is closed under intersection (cf. [6]), i.e., given two context–sensitive grammars G_1 and G_2 one can effectively construct a grammar $G_{1,2}$ such that $L(G_{1,2}) = L(G_1) \cap L(G_2)$. Consequently, one can effectively construct a recursively enumerable family $(G_j)_{j \in N}$ of grammars such that $DPAT = \{L(G_j) \mid j \in N\}$.

We define an IIM M inferring $DPAT$ strong–monotonically on text as follows: Let $L \in DPAT$ and let t be any text for L. For any $x \in N$ we set:
$M(t_x) =$ "Compute $z = min\{|w| \mid w \in t_x^+\}$ and generate all canonical patterns $p_1, ..., p_k$ with $|p_i| \leq z$. Test for all these patterns if $t_x^+ \subseteq L(p_i)$ for $i = 1, ...k$ and let I_x be the set of indices fulfilling the test.
Output a canonical grammar j_x for $\bigcap_{i \in I_x} L(p_i)$."

It remains to show that $DPAT \in SMON - TXT(M)$. Let $L \in DPAT$, i.e., there are patterns $q_1, ..., q_l$ such that $L = \bigcap_{j=1}^{l} L(q_j)$. Let $m = min\{|w| \mid w \in L\}$. Consequently, $|q_j| \leq m$ for all $j = 1, ...l$. Moreover, since t is a text for L we directly obtain that $m \leq z$ for all initial segments t_x of t and that $t_x^+ \subseteq L(q_j)$ for all $j = 1, ...l$. Thus $\{1, ..., l\} \subseteq I_x$ for every $x \in N$. Hence, every $j_x = M(t_x)$ satisfies $L(G_{j_x}) \subseteq L$. Furthermore, since $I_{x+k} \subseteq I_x$

for every $k \in N$ we additionally get $L(G_{j_x}) \subseteq L(G_{j_{x+k}}) \subseteq L$ for all $k \in N$. Consequently, M works strong–monotonically. Finally, M obviously converges since $I_0 \setminus \{1, ...l\}$ is finite. Let j_{fin} be the limit of the sequence of hypotheses. Suppose $L(G_{fin}) \subset L$. Then there is at least a string $s \in L \setminus L(G_{fin})$. On the other hand, there must be an $x \in N$ such that $s \in t_x^+$, since t is a text for L. Due to the construction it must hold that $t_x^+ \subseteq L(p_i)$ for every $i \in I_x$, in particular $s \in L(p_i)$ for all $i \in I_x$. Hence $s \in L(G_{fin})$. Consequently, M converges to a correct grammar.

q.e.d.

On the other hand, at first glance it might seem that the decidability of $L(G) \subseteq L(G')$ may be generally helpful in proving that families of languages are monotonically learnable. However, decidability of inclusion does *not* guarantee monotonic learnability, even *on informant.*

Theorem 12 *The set of all regular languages cannot be monotonically inferred on informant.*

Let us finish this section with the remark that several problems remained open. We shall discuss them in the next section.

6. Conclusions and Open Problems

We have studied the power of strong–monotonic, monotonic and weak–monotonic IIMs in the setting of algorithmic language learning. All these notions have been related one to the other and two new hierarchies could been proven. Although all these notions seem to be quite natural ones, in general, strong–monotonicity and monotonicity lead to a severe restriction of the inference power. This actually shows that learning processes have to be performed, at least to some extent, non–monotonically or at least weak–monotonincly. Moreover, as our results show, if one deals with monotonically working IIMs the choice of the space of hypotheses is of great influence to what actually can be inferred. If that space is appropriately chosen, then weak–monotonic inference can be characterized to be as powerful as conservative learning.

However, several problems remained open. First, it would be very interesting to know whether or not the requirement that the range of the indexed family to be learnt has to coincide with the range of the space of hypotheses is essential for proving $MON-TXT \subseteq WMON-TXT$

Moreover, it seems to be challenging to combine monotonic language learning with other postulates of naturalness, e.g., with iterative inference introduced by Wiehagen (1976).

In the context of inference of natural families of languages the most interesting open question is whether $PAT \in SMON-TXT$, if the space of hypotheses is PAT. Finally, it would be desirable to prove sufficient and necessary conditions for monotonic inference.

Acknowledgement

The authors heartily thank Rolf Wiehagen for many inspiring discussions, and especially, for pointing out an error in an earlier version of this paper.

7. References

[1] Angluin, D., (1980A), Inductive Inference of Formal Languagues from Positive Data, Information and Control 45, 117 - 135.

[2] Angluin, D., (1980B), Finding Patterns Common to a Set of Strings, J. Computer and System Sciences 21, 46 - 62.

[3] Angluin, D. and C.H. Smith, (1987), Formal Inductive Inference, In Encyclopedia of Artificial Intelligence, St.C. Shapiro (Ed.), Vol. 1, pp. 409 - 418, Wiley–Interscience Publication, New York.

[4] Barzdin, Ya.M., (1974), Inductive Inference of Automata, Functions and Programs, Proc. Int. Congress of Math., Vancouver, pp. 455 - 460.

[5] Beick, H.R., (1991), Personal Communication.

[6] Bucher, W. and H. Maurer, (1984), Theoretische Grundlagen der Programmiersprachen, Automaten und Sprachen, Bibliographisches Institut AG, Wissenschaftsverlag, Zürich.

[7] Case, J., (1988), The Power of Vacillation, In Proc. 1st Workshop on Computational Learning Theory, D. Haussler and L. Pitt (Eds.), pp. 196 -205, Morgan Kaufmann Publishers Inc.

[8] Case, J. and C. Lynes, (1982), Machine Inductive Inference and Language Identification, Proc. Automata, Languages and Programming, Ninth Colloquim, Aarhus, Denmark, M.Nielsen and E.M. Schmidt (Eds.), Lecture Notes in Computer Science 140, pp. 107 -115, Springer–Verlag.

[9] Fulk, M.,(1990), Prudence and other Restrictions in Formal Language Learning, Information and Computation 85, 1 - 11.

[10] Gold, M.E., (1967), Language Identification in the Limit, Information and Control 10, 447 - 474.

[11] Kearns, M. and L. Pitt, (1989), A Polynomial-time Algorithm for Learning k-variable Pattern Languages from Examples, In Proc. 2nd Workshop on Computational Learning Theory, R. Rivest, D. Haussler, and M.K. Warmuth (Eds.), pp. 57 - 70, Morgan Kaufmann Publishers Inc.

[12] Jain, S. and A. Sharma, (1990), Language Learning by a "Team", Proc. Automata, Languages and Programming, 17th International Colloquium, Warwick University, England, M.S. Paterson (Ed.), Lecture Notes in Computer Science 443, pp. 153 - 166, Springer–Verlag.

[13] Jantke, K.P., (1991A), Monotonic and Non–monotonic Inductive Inference, New Generation Computing 8, 349 - 360.

[14] Jantke, K.P., (1991B), Monotonic and Non–monotonic Inference of Functions and Patterns, in Proc. First International Workshop on Nonmonotonic and Inductive Logics, December 1990, Karlsruhe, J. Dix, K.P. Jantke and P.H. Schmitt (Eds.), Lecture Notes in Artificial Intelligence 543, pp. 161 - 177, Springer–Verlag.

[15] Ko, K., Marron, A. and W.G. Tzeng, (1990), Learning String Patterns and Tree Patterns From Examples, Proc. 7th Conference on Machine Learning, pp. 384 - 391.

[16] Lange, S. and R. Wiehagen, (1990), Polynomial–Time Inference of Pattern Languages, Proc. Algorithmic Learning Theory 1990, pp. 289 - 301, Tokyo, Ohmsha Ltd.

[17] Nix, R.P., (1983), Editing by Examples, Yale University, Dept. Computer Science, Technical Report 280.

[18] Osherson, D., Stob, M. and S. Weinstein, (1986), Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists, MIT–Press, Cambridge, Massachusetts.

[19] Shinohara, T., (1982), Polynomial Time Inference of Extended Regular Pattern Languages, RIMS Symposia on Software Science and Engineering, Kyoto, Lecture Notes in Computer Science 147, pp. 115 - 127, Springer–Verlag.

[20] Solomonoff, R., (1964), A Formal Theory of Inductive Inference, Information and Control 7, 1 - 22, 234 - 254.

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

[22] Wiehagen, R., (1977), Identification of Formal Languages, Proc. Mathematical Foundations of Computer Science, Tatranska Lomnica, J. Gruska (Ed.), Lecture Notes in Computer Science 53, pp. 571 - 579, Springer–Verlag.

[23] Wiehagen, R., (1990), Personal Communication

[24] Wiehagen, R., (1991), A Thesis in Inductive Inference, in Proc. First International Workshop on Nonmonotonic and Inductive Logics, December 1990, Karlsruhe, J. Dix, K.P. Jantke and P.H. Schmitt (Eds.), Lecture Notes in Artificial Intelligence 543, pp. 184 - 207, Springer–Verlag.