Types of Incremental Learning

Klaus P. Jantke*
FB Informatik
HTWK Leipzig (FH)
Postfach 66
7030 Leipzig
jantke@informatik.th-leipzig.de

Abstract

This paper is intended to introduce a closer look at incremental learning by developing the two concepts of ***informationally incremental learning*** and ***operationally incremental learning***. These concept are applied to the problem of learning containment decision lists for demonstrating its relevance.

1 Introduction

The intention of the present paper is to introduce two new notions in incremental learning which allow a classification of phenomena finer than known so far in the area. These concepts are denoted by the phrases **informationally incremental learning** and **operationally incremental learning**, respectively. Roughly spoken, informationally incremental algorithms are required to work incrementally as usual, i.e. they have no permission to look back at the whole history of information presented during the learning process. Operationally incremental learning algorithms may have permission to look back, but they are not allowed to use information of the past in some effective way. Obviously, the latter concept depends on some more detailed specification of how to process information presented.

It turns out that the two concepts introduced are provably different within some formal settings. For illustration, these concepts are formalized in the area of case-based learning of decision lists. The difference of informationally incremental learning and operationally incremental learning will be deduced formally and illustrated as well.

The author has been led to the the discovery of a difference between informationally incremental and operationally incremental behaviour by comparing his earlier work on several types of inductive inference strategies (cf. [JB81]) including incremental methods, which have been called iterative in the standard recursion-theoretic inductive inference literature (cf. [AS83], e.g.), to recent work on case-based inductive inference (cf. [Jan92]). Animated by ***Sakakibara's*** and ***Siromoney's*** recent investigation into PAC learnability of certain decision lists (cf. [SS92]), he has initiated some research on case-based inductive inference of certain decision lists. As a side effect of the endeavour to model case-based learning ideas in the domain of containment decision lists (This is ongoing work with YASUBUMI SAKAKIBARA.), there have been recognized effects different from those known in a recursion-theoretic framework. This will be informally described in the following paragraph.

In recursion-theoretic inductive inference, incremental learning methods are well-studied. Those methods are called iterative in [JB81], for example. The crucial problem for iterative methods is that they tend to *forget* information they have been fed in during the learning process. This is exactly the effect adopted by [PF91] when transforming a recursion-theoretic method to prove some (un)learnability result for automata. If learning algorithms are invented from an artificial intelligence point of view, they are usually assumed to work in some effective manner formalizing certain heuristics, intuition, or so. From a general theoretic viewpoint, this means to restrict the class of algorithms admitted. Such a way of restricting algorithms taken into account for problem solving frequently implies some methodology of processing information presented to a learning device. For instance, case-based learning algorithms are usually assumed to collect cases in some finite case base. On the contrary, they are usually not allowed to collect or encode anything else in their case base. Formally spoken, this restricts the class of algorithms competing for problem solving. Therefore, under certain formalizations, there may occur another type of incremental behaviour, if some algorithms has no effective way of

*This work has been partially supported by the German Ministry for Research and Technology (BMFT) under grant no. 413-4001-01-IW 101 A.

using certain information presented in the history. This is the case in the area of case-based learning of decision lists.

2 Incremental Learning of Recursive Functions

The present chapter is based on earlier work and intended to relate the new results on learnability in a case-based manner. It provides the background of the introduction of a finer classification of incremental learning.

2.1 Recursion-Theoretic Inductive Inference

In recursion-theoretic inductive inference as invented by *Gold* in his seminal paper [Gol67], total recursive functions are assumed as target objects to be learned from input/output examples only. [AS83] is an excellent survey in this regard. In the general approach, for any recursive function *f*, any complete presentation of its graph is admissible as information fed into some learning device. Arbitrary partial recursive functions **S** may be used for learning any function *f* stepwise presented by finite initial sequences of orderings of its graph. If some ordering is denoted by **X**, the corresponding initial segment of the first *n* input/output examples of *f* w.r.t. **X** is abbreviated by $f_{\mathbf{X}}[\mathbf{n}]$. If **X** is some ordering of arguments

$$x_1, x_2, x_3, x_4, x_5, \cdots$$

then $f_{\mathbf{X}}[\mathbf{n}]$ may be understood as the corresponding sequence

$$(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), \cdots$$

S is said to learn *f* in the limit, if and only if there is some **n** such that for all information covering the initial segment $f_{\mathbf{X}}[\mathbf{n}]$, the hypotheses generated by **S** are identical to each other and represent *f* correctly.

Above, we have presented informally one of the basic definitions in recursion-theoretic inductive inference. The corresponding family of learning problems solvable according to this definition is called **LIM** in [JB81], for instance. In the early work of *Latvian* scientists, this so-called identification type is denoted by **GN**, whereas it is called **EX** in the recent *American* literature. Certain authors are used to call those partial recursive functions **S** solving some learning problem an inductive inference machine (**IIM**, for short). We adopt this notation in the sequel. The reader is directed to [AS83] for further details.

There is a large variety of additional requirements which may be considered to be natural properties of learning strategies (cf. [JB81]). One among them is the property to built hypotheses incrementally in using only the hypothesis before and the recent information on hand when constructing any new hypothesis.

2.2 Incremental Inductive Inference

The small number of concepts introduced above allows nicely to formalize incremental learning. Whereas an arbitrary learning strategy **S** may usually built any hypothesis from all the information $f_{\mathbf{X}}[\mathbf{n}]$ presented, an incremental one has to construct its sequence of hypotheses iteratively. A new hypothesis h_{n+1} is build in dependence of the hypothesis h_n before together with the recent piece of information provided as $(x_{n+1}, f(x_{n+1}))$. For an initially fixed standard information ordering $\mathbf{X_0}$, this led to an identification type **IT** (cf. [AS83], [JB81]) which will be called **INC** in the sequel. Note that the approach under consideration and all its versions require some initial hypothesis (the one preceding the first one constructed during learning), for technical reasons.

It is one of the crucial problems investigated in [JB81] to allow arbitrary orderings **X**. For generality, we are going to introduce the more general approach indicated by the upper index "arb":

Definition 1

A class of total recursive functions **U** is said to be *incrementally learnable in the limit* (notation: $\mathbf{U} \in \mathbf{INC^{arb}}$) **if and only if**
there exists some partial recursive inductive inference machine (**IIM**) named **S** which satisfies for all target functions $f \in \mathbf{U}$, for all information orderings **X**, and for natural numbers n the following conditions, where h_0 denotes some assumed initial hypothesis:

$$h_{n+1} = \mathbf{S}(h_n, (x_{n+1}, f(x_{n+1}))) \qquad \textit{is defined.} \tag{1}$$

$$\exists p (p = \lim_{n \to \infty} h_n \qquad \textit{exists and} \tag{2}$$

$$p \quad \textit{is a correct program for } f). \tag{3}$$

For a formally correct treatment, one usually assumes a priori some *Gödel*numbering φ. Thus, condition (3) above rewrites to

$$\varphi_p = f \tag{4}$$

This completes Definition 1.

The following concept formalizes the particular case of incremental learning over some a priori assumed information ordering.

Definition 2

Assume the standard ordering $\mathbf{X_0}$ of natural numbers. A class of total recursive functions **U** is said to be *incrementally learnable in the limit* (notation: $\mathbf{U} \in \mathbf{INC}$) **if and only if**
there exists some partial recursive inductive inference machine (**IIM**) named **S** which satisfies for all target functions $f \in \mathbf{U}$ and for natural numbers n the following conditions, where h_0 denotes some assumed initial

hypothesis as before:

$$h_{n+1} = \mathbf{S}(h_n, f(n+1)) \qquad \textit{is defined.} \tag{5}$$

$$\exists p(\ p = \lim_{n\to\infty} h_n \qquad \textit{exists and} \tag{6}$$

$$\varphi_p = f.\ ) \tag{7}$$

This completes Definition 2.

Note that it is one of the classical results of recursion-theoretic inductive inference that **LIM** equals $\mathbf{LIM}^{\mathbf{arb}}$. So far, there have been introduced here four collections of learning problems each of them comprising problem classes solvable uniformly in some sense. Two of them, namely $\mathbf{INC}^{\mathbf{arb}}$ and **INC**, formalize incremental learning over certain information sequences. Within the author's investigations presented here, this type of learning should be called **informationally incremental**. The proof of the following classical result (cf. [JB81], e.g.) exhibits the essence of the problem.

Theorem 1

$$\mathbf{INC}^{\mathbf{arb}} \quad \subset \quad \mathbf{INC} \quad \subset \quad \mathbf{LIM}^{\mathbf{arb}} \quad = \quad \mathbf{LIM}$$

The proofs of the two proper inclusions are based on the problem that **informationally incremental IIM** can't store all the information seen during a learning process. There is the problem of forgetting (cf. [AS83], [JB81], and [PF91], for details).

2.3 Classical Results

In the early days of inductive inference, learnability seemed to depend of effective enumerability. In [Gol67], *Gold* invented the principle of **identification by enumeration**. The family of all classes of total recursive functions which can be embedded into an effective enumeration of only total functions is denoted by **NUM**. It is among the classical results of inductive inference that learnability requires certain generalizations of the **identification by enumeration** principle. Furthermore, there must be other powerful learning principles as indicated by the result that there are learnable problem classes not belonging to **NUM**. Incorporating **NUM**, the first theorem extends to the following one which may be found in [JB81] and [Jan92], for example.

Theorem 2

$$\mathbf{NUM} \quad \subset \quad \mathbf{INC} \quad \subset \quad \mathbf{LIM}^{\mathbf{arb}} \quad = \quad \mathbf{LIM}$$

$$\mathbf{NUM} \quad \# \quad \mathbf{INC}^{\mathbf{arb}}$$

Here, # denotes set-theoretic incomparability.

There have been studied several combinations of seemingly natural properties of **IIM**. It turns out that for some of those properties, it further restricts the power of **IIMs** of this type, if one forces an **IIM** to work incrementally. This typically holds for consistent **IIMs** on arbitrary information sequences.

An **IIM** is said to work consistently, if every hypothesis generated reflects all the information it has been built upon.

Definition 3

A class of total recursive functions **U** is said to be *consistently learnable in the limit* (notation: $\mathbf{U} \in \mathbf{CONS}^{\mathbf{arb}}$)
if and only if
there exists some partial recursive inductive inference machine (**IIM**) named **S** which satisfies for all target functions $f \in \mathbf{U}$, for all information orderings **X**, and for natural numbers n the following conditions:

$$h_{n+1} = \mathbf{S}(f_{\mathbf{X}}[\mathbf{n}]) \qquad \textit{is defined.} \tag{8}$$

$$\forall m(\ m \le n \Rightarrow \varphi_{h_n}(x_m) = f(x_m)\ ) \tag{9}$$

$$\exists p(\ p = \lim_{n\to\infty} h_n \qquad \textit{exists and} \tag{10}$$

$$\varphi_p = f\ ). \tag{11}$$

Additionally, if one requires an **IIM** to be totally defined, this is indicated by the prefix "**R**-". This leads to the identification type $\mathbf{R}-\mathbf{CONS}^{\mathbf{arb}}$ analogously to the definition above.

This completes Definition 3.

As already mentioned above, it may be desirable to combine several postulates of naturalness. This kind of modelling is motivated by artificial intelligence research, where algorithms are usually designed in a way that they meet several conditions simultaneously. Frequently, authors of those algorithms do not consider the problem whether or not a combination as proposed has any impact on the amount of solvable problems.

Definition 4

By combining postulates used in the preceding definitions, one defines further identification types as follows.

$\mathbf{U} \in \mathbf{CONS} - \mathbf{INC}^{\mathbf{arb}}$

if and only if

$$\exists \mathbf{S}(\ \mathbf{S}\ \textit{is}\ \textbf{partial}\ \textit{recursive and satisfies}\ (1),(2),(4),(9)\ ) \tag{12}$$

$\mathbf{U} \in \mathbf{R} - \mathbf{CONS} - \mathbf{INC}^{\mathbf{arb}}$

if and only if

$$\exists \mathbf{S}(\ \mathbf{S}\ \textit{is}\ \textbf{total}\ \textit{recursive and satisfies}\ (1),(2),(4),(9)\ ) \tag{13}$$

This completes Definition 4.

Assume any **IIM** working consistently in the sense of $\mathbf{CONS^{arb}}$ or $\mathbf{R-CONS^{arb}}$. If one additionally wants the **IIM** to work incrementally, this may restrict its learning power properly. This discovery is mainly based on the problem of forgetting information seen during the process of learning. Therefore, we are faced to the difficulty of achieving **informationally incremental** behaviour.

Theorem 3

$$\mathbf{R-CONS-INC^{arb}} \subset \mathbf{R-CONS^{arb}}$$

$$\mathbf{CONS-INC^{arb}} \subset \mathbf{CONS^{arb}}$$

Orthogonally, if some incrementally learning **IIM** is forced to meet further conditions, this may restrict its learning power essentially.

Theorem 4

$$\mathbf{R-CONS-INC^{arb}} \subset \mathbf{CONS-INC^{arb}}$$

$$\mathbf{CONS-INC^{arb}} \subset \mathbf{INC^{arb}}$$

For the proofs, wich exhibit the character of these statements and point to the problem of forgetting, the reader is directed to [JB81].

To sum up the recursion-theoretic results presented above, there is a considerable amount of theoretical work illustrating the impact of incremental behaviour in learning. A careful inspection of the corresponding proofs exhibits the nature of the results which is related to the phenomenon of forgetting information seen earlier. This seems also to be the crux in more applied approaches. However, in the author's opinion, this view at incremental learning is much too restrictive. When investigating the problems above, we have only been faced to phenomena of **informationally incremental** learning. But there is another phenomenon being quite different in character. This is related to **operationally incremental** learning investigated in the sequel.

3 Incremental Learning of Containment Decision Lists

3.1 Motivation of the Domain

There are different motivations for the current work on inductive inference of containment decision lists. Some of them are sketched below.

First, there are some recent applications of learning decision trees and decision lists of several types wich are interesting both because of the simplicity of decisions list providing some practical success and because of the relevance of the application (cf. [AKM+92]) addressed. Therefore, it seems promising to explore more deeply further approaches to learning containment decision lists.

Second, there is recent work (cf. [SS92]) about the learnability of containment decision lists within the PAC framework. On the one hand, the deduced positive results provide some hope for further learnability results. On the other hand, they are providing the background for comparisions of PAC learnability.

Third, containment decision lists seem particularly suited for case-based learning approaches. This needs some more detailed explanations.

Case-based reasoning is a recently booming subarea of artificial intelligence. One important reason is that human experts tend to use knowledge in the form of particular cases or episodes rather frequently than generalized knowledge as described by rules, e.g. Therefore, there is some hope that case-based reasoning may help to widen the bottleneck of knowledge acquisition. The reader is directed to [Kol92] for a recent introduction in and survey of case-based reasoning. Within case-based reasoning, case-based learning is a rather natural way of designing learning procedures. Recent formalizations (cf. [Jan92]) have exhibited the remarkable power of case-based learning algorithms. In the particular setting of learning total recursive functions, which covers the problem of learning effective classifiers in formalized areas, everything learnable inductively turns out to be learnable in a case-based manner. This may be understood as a normal form result for inductive inference algorithms.

This general result raised the question how to be interpreted in particular settings where there may be or not a natural concept of cases. In some areas, cases seem to be conceptually far from the target objects to be learned. For example, in the area of learning number-theoretic functions from input/ output examples, those input/output examples may be considered as cases specifying the intended target behaviour. In despite of any paricular choice of an underlying programming language, there is usually a considerable syntactical difference between programs and examples of their corresponding behaviours. Usually, there is no position in a program where some input/output examples occur syntactically. There is a minor class of exceptions including textual dialogue components.

On the contrary, the area of containment decision lists looks quite promising. If containment decision lists are understood to accept formal languages, and if cases are formalized as labelled words, those decision lists are obviously constructed directly from the best cases describing the language accepted. This yields an immediate syntactic correspondence between the information to be processed and the hypotheses to be generated within an inductive learning process. Because of these extraordinary formal assumption, one might expect further insights into the nature of case-based learning when investigating both the power and the limitations of case-based learning applied to containment decision lists.

In the present paper, the key reason to consider case-based learning of containment decision lists is that the domain is suitable to illustrate and prove the importance of the concept of **operationally incremental** learning.

3.2 Syntax and Semantics of Containment Decision Lists

We assume any finite, non-empty alphabet $\mathbf{A}$. For technical reasons, $\mathbf{A}$ should contain at least two different elements. Words over $\mathbf{A}$ are build as usual. $\mathbf{A}^*$ denotes the set of all words over $\mathbf{A}$, whereas $\mathbf{A}^+$ is the set of all non-empty words over $\mathbf{A}$. The empty word is written ε.

Containment decision list are finite lists of labelled words. As labels, we consider only $\mathbf{0}$ and $\mathbf{1}$. The empty list is denoted by **nil**. For the formal definition, we assume any alphabet $\mathbf{A}$. There is a basic relation used in the sequel and denoted by $\preceq$. The definition is as follows.

Definition 5

$\forall u, v \in \mathbf{A}^*$

$$u \preceq v \quad \textbf{if and only if} \quad \exists w_1, w_2 (w_1 u w_2 = v) \qquad (14)$$

Definition 6

Base of Induction

nil is a containment decision list over $\mathbf{A}$.

Step of Induction

If $\mathbf{X}$ is any containment decision list over $\mathbf{A}$, w is any word from $\mathbf{A}$, and d is either $\mathbf{0}$ or $\mathbf{1}$, then $[\mathbf{X}|(w,d)]$ is also a containment decision list over $\mathbf{A}$.

The class $\mathcal{DL}$ is the smallest class w.r.t. the *Base of Induction* and the *Step of Induction* above. This completes Definition 6.

Those decision lists are called containment decision lists, as their nodes of the form (w,d) are understood as tests whether or not some processed input word z contains w. In case it does, the computation process described terminates with the output d. Therefore, one may consider containment decision lists as devices accepting formal languages. If some word z is fed into a list $\mathbf{X}$, there will be performed this test subsequently at every node until there is a first positive result or, alternatively, the whole list has been passed. If there is some first node (w,d) where w turns out to be a substring of z, d determines whether or not z is accepted. One usually interprets $\mathbf{1}$ as the positive answer. If there is no node (w,d) in the list satisfying $w \preceq z$, and if there is some final node is (w',d'), the decision is $\mathbf{1}-d'$. This will be defined formally below. Besides defining the language accepted by some containment decision list, there will be defined the language passing through some list, for technical purposes. The empty decision list accepts the empty language. This is dropped below.

Definition 7

Base of Induction

$$\mathbf{L}([(w,\mathbf{1})]) = \mathbf{A}^*\{w\}\mathbf{A}^*$$

$$\mathbf{L}([(w,\mathbf{0})]) = \mathbf{A}^*\backslash\mathbf{A}^*\{w\}\mathbf{A}^*$$

$$\mathbf{Lp}([(w,d)]) = \mathbf{A}^*\backslash\mathbf{A}^*\{w\}\mathbf{A}^* \qquad (\textit{for } d = \mathbf{0},\mathbf{1})$$

Step of Induction

$$\mathbf{L}([[\mathbf{X}|(v,\mathbf{1})]|(w,\mathbf{1})]) =$$
$$\mathbf{L}([\mathbf{X}|(v,\mathbf{1})]) \cup (\mathbf{Lp}([\mathbf{X}|(v,\mathbf{1})]) \cap \mathbf{A}^*\{w\}\mathbf{A}^*)$$

$$\mathbf{L}([[\mathbf{X}|(v,\mathbf{1})]|(w,\mathbf{0})]) =$$
$$\mathbf{L}([\mathbf{X}|(v,\mathbf{1})]) \cup (\mathbf{Lp}([\mathbf{X}|(v,\mathbf{1})]) \cap (\mathbf{A}^*\backslash\mathbf{A}^*\{w\}\mathbf{A}^*))$$

$$\mathbf{L}([[\mathbf{X}|(v,\mathbf{0})]|(w,\mathbf{1})]) =$$
$$\mathbf{L}([\mathbf{X}|(v,\mathbf{0})]) \setminus (\mathbf{Lp}([\mathbf{X}|(v,\mathbf{0})]) \cap (\mathbf{A}^*\backslash\mathbf{A}^*\{w\}\mathbf{A}^*))$$

$$\mathbf{L}([[\mathbf{X}|(v,\mathbf{0})]|(w,\mathbf{0})]) =$$
$$\mathbf{L}([\mathbf{X}|(v,\mathbf{0})]) \setminus (\mathbf{Lp}([\mathbf{X}|(v,\mathbf{0})]) \cap \mathbf{A}^*\{w\}\mathbf{A}^*)$$

$$\mathbf{Lp}([\mathbf{X}|(w,d)]) = \mathbf{Lp}([\mathbf{X}]) \setminus \mathbf{A}^*\{w\}\mathbf{A}^*$$
$$(\textit{for } d = \mathbf{0},\mathbf{1})$$

The class $\mathcal{LDL}$ is the class of all languages accepted by any containment decision list over some given alphabet $\mathbf{A}$.

Obviously, $\mathcal{LDL}$ is a subclass of the class of all regular languages over the corresponding alphabet. It is distinguished by a number of interesting properties including particular pumping phenomena. (There is no space to go into details, here.)

There are known some normal form results. Instead of a theoretical investigation, we are providing an example, only. For this example, assume $\mathbf{A} = \{a,b\}$. An infinite sequence of containment decision lists $\mathbf{X_0}, \mathbf{X_1}, \mathbf{X_2}, \cdots$ is defined by

$$\mathbf{X_0} = \mathbf{nil}$$

$$\mathbf{X_{n+1}} = [[\mathbf{X_n}|(ba^{n+1}b,\mathbf{1})]|(ab^{n+1}a,\mathbf{0})]$$

Without technical investigations, it is quite clear that no containment decision list $\mathbf{X_n}$ can be reduced without changing the language accepted. One may also easily recognize that the sequence of languages defined satisfies the following property:

$$\mathbf{L}(\mathbf{X_1}) \supset \mathbf{L}(\mathbf{X_2}) \supset \mathbf{L}(\mathbf{X_3}) \supset \cdots$$

3.3 Case-Based Learning of Containment Decision Lists

Labelled words are understood as cases describing some language to be learned. Containment decision lists seem particularly tailored to case-based learning, as the nodes of a list are obviously distinguished cases describing its behaviour. The reader may consult the example above in this regard.

Containment decision lists may be easily transformed into a pair consisting of a case-base **CB** and a similiraty measure σ defining the same language. This requires some specification of the semantics of pairs $(\mathbf{CB},\sigma)$. The most natural one is as follows.

Definition 8

Assume any finite set $\mathbf{CB} \subset (\mathbf{A}^+ \times \{\mathbf{0},\mathbf{1}\})$ and any computable mapping σ from the set $(\mathbf{A}^+ \times \{\mathbf{0},\mathbf{1}\}) \times (\mathbf{A}^+ \times \{\mathbf{0},\mathbf{1}\})$ into $[\mathbf{0},\mathbf{1}]$. This defines a formal language $\mathbf{L}(\mathbf{CB},\sigma)$ by

$$z \in \mathbf{L}(\mathbf{CB},\sigma) \quad \textbf{if and only if}$$

$$\exists (w,1) \in \mathbf{CB} \; (\; \sigma(w,z) > 0 \quad \textbf{and} \tag{15}$$

$$\forall (v,0) \in \mathbf{CB} \; (\; \sigma(v,z) < \sigma(w,z) \;) \;) \tag{16}$$

A simple class Σ of similarity measures σ is sufficient for expressing any language generated by a containment decision list. The reader may easily check this by means of the example above. The target class of similarity measures is defined as follows.

Definition 9

Assume any finite, non-empty alphabet **A**.

$$\sigma \in \Sigma \quad \textbf{if and only if}$$

$$\sigma \textit{ is computable and} \tag{17}$$

$$\forall v,w \in \mathbf{A}^* \; (\; \sigma(v,w) > 0 \iff v \preceq w \;) \tag{18}$$

The formal approach to case-based learning adopted here has been directly derived from algorithms in the area published recently (cf. [Aha91] and [AKA91]). In [Jan92], there has been taken a similar approach based on the references mentioned. The following formalization is directly reflecting the crucial properties of case-based resp. instance-based algorithms. *First*, these algorithms work **incrementally** in stepwise collecting a case-base starting with the empty case-base $\mathbf{CB_0}$ and eventually additing cases on hand to the case-base generated so far. This type of work should be called **operationally incremental** as it is seriously distinguished from the type of incremental learning investigated before. Its limitations do not stem from a lack of information, but from certain operational restrictions. *Second*, these algorithms may learn some similarity measures of Σ starting with any naive measure σ_0.

One may easily invoke *Gold's* seminal work (cf. [Gol67]) on learning finite deterministic automata in the limit. *Gold's* results may be adopted to prove that there is no universal algorithm for learning arbitrary containment decision lists in the limit from only positive cases. But his results imply directly the learnability of arbitrary containment decision lists from complete sequences of both positive and negative cases in the limit. These results do not refer to learning in a base-based manner. In particular, incremental learning ist not explicitly attacked. Our formalization of case-based learning reflects the idea of **operationally incremental learning**.

Definition 10

A class $\mathcal{C} \in \mathcal{DL}$ of containment decision lists is learnable in the limit in a case-based manner
if and only if
there is an **IIM** basically consisting of two computable procedures $\mathbf{M_1}$ and $\mathbf{M_2}$ such that for all $\mathbf{X} \in \mathcal{C}$ and for all sequences $(w_1,d_1),(w_2,d_2),(w_3,d_3),\cdots$ of cases completely describing $\mathbf{L}(\mathbf{X})$ in the limit, it holds

$$\forall n \in \mathcal{N} (\, \mathbf{CB_{n+1}} =_{def} \mathbf{M_1}(\mathbf{CB_n},(w_{n+1},d_{n+1})) \, \textit{with}$$

$$\mathbf{CB_n} \subseteq \mathbf{CB_{n+1}} \subseteq \mathbf{CB_n} \cup (w_{n+1},d_{n+1}) \,) \tag{19}$$

$$\forall n \in \mathcal{N} \; (\; \sigma_{n+1} =_{def} \mathbf{M_2}(\sigma_n,(w_{n+1},d_{n+1})) \in \Sigma) \tag{20}$$

$$\mathbf{CB}^* =_{def} \lim_{x\to\infty} \mathbf{CB}_n \textit{ exists.} \tag{21}$$

$$\sigma^* =_{def} \lim_{x\to\infty} \sigma_n \textit{ exists.} \tag{22}$$

$$\mathbf{L}(\mathbf{CB}^*,\sigma^*) \; = \; \mathbf{L}(\mathbf{X}) \tag{23}$$

The conditions (19) and (20) formalize the intended case-based behaviour. In particular, (19) specifies **operationally incremental learning**. The following conditions (21), (22), and (23) reflect the basic properties of learning in the limit as seen above in Definition 1, e.g.

It turns out that only very restricted classes of containment decision list are learnable in the sense introduced here. The key reason is the restrictedness of **operationally incremental learning**.

Theorem 5

The class $\mathcal{DL}$ ist not learnable in the limit in a case-based manner.

Proof: For every **IIM** intended to learn containment decision lists in a case-based manner, it suffices to find a subclass of $\mathcal{DL}$. Fortunately, already a slight modification of the simple example discussed above will do. Because of the lack of space, we will only sketch the key idea. For any increasing sequence of exponents $k_1,k_2,\cdots,k_n$, one defines a containment decision list.

$$\mathbf{X}_{k_1,k_2,\cdots,k_n} \quad = \quad [(ba^{k_1}b,1),(ab^{k_1}a,0),\cdots,(ba^{k_n}b,1),(ab^{k_n}a,0)]$$

For any given list $\mathbf{X}_{k_1,k_2,\cdots,k_n}$, one may construct a sequence of cases describing the corresponding language such that there are repeatedly presented positive cases of the form $(ba^m b, 1)$, where m exceeds k_n. Because of the stabilization conditions (21) and (22) above, there must be a first case $(ba^m b, 1)$ ignored by the **IIM**, i.e. there is some step l in which $(ba^m b, 1)$ is presented and it holds

$$\mathbf{CB_{l+1}} =_{def} \mathbf{M_1}(\mathbf{CB_l}, (ba^m b, 1)) = \mathbf{CB_l}.$$

As the case $(ba^m b, 1)$ may be never presented again, the **IIM** turns out to be unable to learn $\mathbf{X}_{k_1,k_2,\cdots,k_n,m}$ as specified in Definition 10, even if the learning device had permission to look back at all the information seen before. This completes the proof.

4 Conclusions

There has been proposed a finer classification of **incremental learning** consisting in a distinction of **informationally incremental learning** and **operationally incremental learning**. Both types of learning have a restricted power compared to unconstrained learning. The algorithmic learning theory is providing firm results in this regard. Within certain formal settings, **operationally incremental learning** is particularly restrictive. Especially, **operationally incremental learning** seems to be the crux of the limitations of strong case-based learning approaches. This may be understood as a hint how to relax requirements for enlargement of the scope of case-based learning devices.

In particular, one may argue that the best way to overcome the limitations exhibited above is to derive the missing cases effectively from the current cases presented. But this exceeds the standard approach. It leads to the problem of generalizing or abstracting cases. Trivially, a sufficiently general approach will allow to learn arbitrary containment decision lists in the limit, as the class of all regular languages is known to be learnable from informant (cf. [Gol67]).

A reasonable generalization should be still in the spirit of case-based reasoning. The results presented above suggest to consider certain cases as distinguished as they are more typical than others. This may be understood as the necessity to introduce formal approaches to the concept of *prototypes*. Cognitive sciences provide a rich background to guide appropriate formalizations (cf. [Ros75], e.g.).

Note that the problem of prototypes is somehow related to the concept of *good examples* which is an important issue of recent inductive inference research.

Furthermore, nonstandard approaches to similarity may allow two widen the scope of case-based reasoning methods. It is one of the most important nonstandard ideas to lift similarity from a local property to a global one which allows to express the similarity of more than two objects uniformly.

This paper is intended to be a first step in its area.

References

[Aha91] David W. Aha. Case-based learning algorithms. In *DARPA Workshop on Case Based Reasoning*, pages 147–157. Morgan Kaufmann, 1991.

[AKA91] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[AKM+92] Setsuo Arikawa, Satoru Kuhara, Satoru Miyano, Yasuhito Mukouchi, Ayumi Shinohara, and Takeshi Shinohara. A machine discovery from amino acid sequences by decision trees over regular patterns. In *Intern. Conference on Fifth Generation Computer Systems, June 1-5, 1992*, volume 2, pages 618–625. Institute for New Generation Computer Technology (ICOT), Tokyo, Japan, 1992.

[AS83] Dana Angluin and Carl H. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–269, 1983.

[Gol67] E Mark Gold. Language identification in the limit. *Information and Control*, 14:447–474, 1967.

[Jan92] Klaus P. Jantke. Case based learning in inductive inference. In *Proc. of the 5th ACM Workshop on Computational Learning Theory, (COLT'92), July 27-29, 1992, Pittsburgh, PA, USA*, pages 218–223. ACM Press, 1992.

[JB81] Klaus P. Jantke and Hans-Rainer Beick. Combining postulates of naturalness in inductive inference. *EIK*, 17(8/9):465–484, 1981.

[Kol92] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6:3–34, 1992.

[PF91] Sandra Porat and Jerome A. Feldman. Learning automata from ordered examples. *Machine Learning*, 7:109–138, 1991.

[Ros75] Eleanor Rosch. Cognitive reference points. *Cognitive Psychology*, 7:531–547, 1975.

[SS92] Yasubumi Sakakibara and Rani Siromoney. A noise model on learning sets of strings. In *Proc. of the 5th ACM Workshop on Computational Learning Theory, (COLT'92), July 27-29, 1992, Pittsburgh, PA, USA*, pages 295–302. ACM Press, 1992.