ISSN 0105-8517

LEARNABILITY

Gudmund Frandsen

DAIMI PB - 199
October 1985

DATALOGISK AFDELING
Bygning 540 - Ny Munkegade - 8000 Aarhus C
tlf. (06) 12 83 55, telex 64767 aausci dk
Matematisk Institut   Aarhus Universitet

PB - 199   G. Frandsen: Learnability

TRYK: RECAU (06) 12 83 55

LEARNABILITY

by

Gudmund Frandsen

Computer Science Department

Aarhus University

Abstract

Les Valiant has recently conceived a remarkable mathematical model of learnability. The originality appears through several facets of the model. Objects belonging to a specific concept are given a measure of naturalness in the form of a probability distribution. The learning of a concept takes place by means of a protocol that among other tools allows the use of a source of natural examples. A concept is learnable if a recognition algorithm can be synthesized within a polynomial number of steps. The recognition algorithm is allowed to be incorrect for an adjustable fraction of inputs measured with respect to naturalness.

Technically the model is based on the propositional logic over a finite number of boolean variables. However, the underlying ideas are quite universal and can be realised by means of an almost arbitrary formal language, which we will demonstrate in this note. A single concept may include infinitely many objects within the formal language frame. Fortunately we can learn such concepts from finite sets of examples only. We shall prove a specific class of concepts to be learnable within the nontrivial formal language of predicate logic.

1. Introduction

Much work has been done in order to establish a mathematical theory of learning. Here we shall only recall a few results that are seminal for the work reported in this paper. For a broad survey of "inductive inference" one may consult [1].

Models of learnability for formal languages are described by Gold [3] and Wharton [12]. Quite recently Valiant [10,11] has come up with an interesting model of learnability for the (narrow) universe of propositional functions. The aim of this paper is to generalise Valiant's ideas to the universe of formal languages. We give a brief description of the development of ideas from Gold over Wharton to Valiant (Table 1 provides a schematic overview). It should be stressed that this "line of development" is crude in that important contributions have been ignored. In addition the "line" does not necessarily reflect an order of influence, e.g. Valiant seems not to know Wharton's work.

In Gold's model, learnability is a quality attributed to classes of languages rather than individual languages. The pupil must identify an unknown language belonging to a known class of possible languages using tools of two main types:

1) A source of positive examples, i.e. a complete enumeration of words belonging to the unknown language.

2) An oracle (informant), i.e., a tool that can decide whether a word belongs to the unknown language.

Gold's definition of learnability (identification in the limit) is the following: Assume time is quantised and has a finite starting time. At each time the pupil receives a unit of information (from some tool) and is to make a guess as to the identity of the unknown language on the basis of the information received so far. This process continues forever. A class of languages will be considered learnable with respect to the tool used if there is an algorithm that the learner can use to make his guesses, the algorithm having the following property: Given any language of the class, there is some finite time after which the guesses will all be the same and they will be correct.

According to this definition the pupil can not in general decide, whether his present guess is correct. Gold mentions other definitions that remedy this problem. Yet only the former definition is used to prove classes of languages to be or not to be learnable. The achieved results are quite strong (see [3]).

Many variations of Gold's model have been made. We consider a specific variation constructed by Wharton [12]. He introduces an interesting relaxation of the notion of learnability; exact identification is replaced by approximate identification. He gives every possible word a weight (the total sum of all weights is 1) and defines a metric on the set of languages by letting the distance between two languages equal the total weight of the symmetric set difference. Wharton proves that any language can be approximated arbitrarily well by a finite language with respect to this metric.

When learnability is based on approximate identification, the class of possible languages need not contain the unknown language to be learned, or a simple language may be learned as an approximation of a more complex one.

Wharton investigates approximate identification with respect to finite and fixed time identification apart from identification in the limit. When using finite identification the pupil recognises a correct guess as such, although he may not know how long time that has to pass before he makes a correct guess. In contrast fixed time identification allows to estimate the learning time in advance.

Wharton obtains some positive results that are considerably stronger than those Gold achieved with respect to exact identification, which should be no surprise.

In all variants of the learning model (incl. fixed time identification) the actual learning time may be quite unfeasable. This problem seems ignored until a recent paper by Valiant [10,11]. He connects learnability with approximate identification in probabilistic polynomial time. Valiant presents his model in terms of boolean functions (propositional logic) rather than formal language theory. He interprets classes of these functions as concepts, thereby letting his theory become one of "fast approximate concept identification". The weight associated with an individual object belonging to a concept is interpreted as a measure of naturalness for the object in question. In this way Valiant almost provides a mathematical model of Larsen's [4] fuzzy concepts organized around prototypical instances in that fuzziness corresponds to approximate identification whereas prototypes are instances with a relatively high weight.

We generalise Valiants model to languages in general, of which the language of Boolean functions is a specific instance. The main problem in our generalisation is the transition from a finite to a countable universe. However, there is a simple solution: Infinite languages (concepts) can be learned from a finite set of examples. Indeed, this seems to be the only type of solution that can satisfy requirements of polynomial time complexity.

Valiant's model uses fixed time identification, where the polynomial time complexity is measured with respect to several parameters, one of which is the size of the function to be learned. In our model, it is assumed that the pupil does not know the latter parameter, which appears to be a natural situation. Thus we obtain a model that uses finite time identification rather than fixed time identification.

Before presenting our model, we want to stress the philosophical basis for our use of probabilistic algorithms, which is importantly different from the traditional one. Probabilistic algorithms have mostly been applied to problems that had a welldefined unique solution, e.g., prime recognition, in which case a prime is correctly identified with probability at least $\frac{1}{2}$ based on a random number source. Existing implementations of such algorithms seem to work properly, although their correctness is an open problem, since they use pseudorandom number generators that are not truly random [2].

In contrast, we can prove a probabilistic algorithm to be correct with respect to our formal model of learnability, because assumptions about true randomness are built into the model, i.e., we suppose a concept to be fuzzy and probabilistic describable rather than sharply defined.

TABLE 1

	Gold	Wharton	Valiant	This paper
Universe of discourse:				
- formal languages	x	x		x
- propositional logic			x	
Learning tools:				
1) output from example:				
- complete enumeration	x	x		
- natural distribution			x	x
2) input to oracle:				
- single words	x	x		
- certain languages			x	x
Lernability Criterium:				
1) identification:				
- exact	x			
- approximate		x	x	x
2) success guarantee				
- in the limit	x	x		
- finite		x		x
- fixed time		x	x	
(- prob. pol. time)			x	x
3) what is learned:				
- recognizer	x		x	x
- generator	x	x		
Results:				
- learnable classes	x	x	x	x
- non-learnable classes	x			

2. Learnability of abstract concepts

Our model is abstracted from Valiant's model by ignoring the technical details of the boolean functions. In addition we generalize the model to incorporate concepts with infinite extension (infinite languages). In order to stress the concept-view on our model we will use Larsen's [4] notion of fuzzy concepts in the presentation. We start by sketching Larsen's paradigm and continue by an informal description of our model followed by a mathematical more precise definition.

According to classical logic a concept has an extension and an intention. The extension is composed of those objects or phenomena that are covered by the concept. The intension is those properties that are shared by all instances of the concept.

Larsen claims firstly that everyday concepts are fuzzy, e.g., it may be questioned, whether garlic is a vegetable. Secondly, membership of a concept is not a question of either-or. Some objects are more typical instances of a concept than others, e.g., carrot is one of the most typical vegetables, while garlic must be considered if at all vegetable very atypical.

In summary, the extension of a concept is characterized by fuzzy bounds and by varying typicalness of the individual instances.

Let us now turn towards our model of learnability. We start by introducing a context consisting of some objects, each of which bear a probability, measuring the relative typicalness of the bearer. A concept has an extension, which is a subset of these objects. The intension is a fast recognition algorithm for the extension. An individual is said to learn a specific concept if he in short time synthesizes a fast approximate recognition algorithm for the unknown concept. His only information consists of typical objects from the context and "oracle"-answers from another individual, who knows the concept.

Thus our model incorporates the varying typicalness of individual instances, whereas fuzziness appears only indirectly as "approximate identification" in connection with learning. Technically, we let the set of objects be words over some finite alphabet. A concept is then a language, which may be described by its extension (a set of words) or by its intention (a recognizer). The precise notions of concepts and learnability are now introduced via a row of definitions.

A context (R,D) consists of a countable set of objects R and a probability distribution D defined on R. We assume a simple structure on R in that R should be a recursive language over a finite alphabet. A standard enumeration of R is given $R = \{r_1, r_2, \ldots\}$ such that $\mathrm{size}(r_i) \leq \mathrm{size}(r_{i+1})$, where the size-function counts the number of symbols in the argument word. The distribution $D\colon R \to [0,1]$ fulfils $\sum_{r\in R} D(r) = 1$. D measures the typicalness or relative frequency of the individual objects. The prototypes are those objects that have associated a high relative frequency. Such objects are few (finite in number) and they are represented by "short" words. Most of the words are quite atypical, and it may happen that $D(r) = 0$ for an object r. To verify these observations define

$$l_h = \min\{l \in \mathbb{N} \mid \sum_{\mathrm{size}(r)\leq l} D(r) \geq 1-h^{-1}\}.$$

l_h gives an upperbound on the length of words that represent the most typical objects (as measured by the parameter h). In a learning situation the context can only be accessed through a routine example. When stimulated the routine outputs an object. The probability that a specific object r is output on any single call with input h is

$$D(r) / \sum_{\mathrm{size}(r')\leq l_h} D(r')$$

for $\mathrm{size}(r) \leq l_h$ and 0 otherwise. If we do not take precautions to bound the size of output from example, we can not bound the time-complexity of an algorithm that uses this routine. There should exist a polynomium p that is an upperbound of l_h. This restriction is trivially fulfilled when R is finite. In general we say that the context (R,D) is bounded by a polynomium p, when l_h is bounded by p.

To sum up, a context consists of a countable set of objects R, on which a probability distribution D, measuring typicalness is defined. This context can only be accessed through the routine example, which returns typical objects randomly according to the probability distribution D. We proceed to define the notion of concept.

A concept is given by its intension, i. The intension is some procedure, which recursively distinguishes those objects that are instances of the concept in question from those that are not. This procedure should have polynomial time complexity. The extension includes precisely those objects, which belong to the concept according to the intension. The extension is denoted by ext(i). In a learning situation a concept may only be accessed through the routine oracle. This routine outputs 1 or 0 given an arbitrary intension i' as input according to whether $ext(i') \subseteq ext(i)$ or $ext(i') \nsubseteq ext(i)$. Oracles task is truly intractable, since oracle can not in general be based on a recursive procedure (To see this, choose i to be a recognizer for some appropriate contextfree language).

In summary, a concept is defined by its intension, a fast recognition algorithm. From the intension, we form the extension and the oracle routine. External information about the concept can only be extracted through the oracle.

We proceed to define fuzziness followed by learnability. The notion of fuzziness is connected to learnability. When a concept is learned only a fuzzy or approximative description of it is obtained. Let a context (R,D), a concept (i) and an approximation parameter $h > 1$ be given. Another concept (i') is is said to be an h-approximation for (i), when

$$\sum_{r \in \Delta ext(i,i')} D(r) \leq h^{-1},$$

where

$$\Delta ext(i,i') = (ext(i) \setminus ext(i')) \cup (ext(i') \setminus ext(i)).$$

This definition may be interpreted in the way that one concept(i') is an approximation of another concept(i), if their respective recognition algorithms do agree on a majority (quantified by $1-h^{-1}$) of the most typical objects.

We are now ready to define learnability: Given a set of objects R and a polynomium p, a class of concepts C is <u>learnable</u> if there exists an algorithm A that for any context (R,D) bounded by p, any concept (i) $\in$ C and any parameter $h > 1$ outputs a fuzzy description of (i) fast almost always, i.e., A deduces an h-approximation for (i) with probability at least $(1-h^{-1})$ and A has polynomial time complexity with respect to

1) The degree of approximation as measured by h.

2) The difficulty of the concept to be learned as measured by the size of the intention i.

The algorithm has full knowledge of R and p, but may only access D, the measure of typicalness through "example". Correspondingly only "oracle" provides information about the unknown concept(i). So the algorithm has no knowledge of the actual number of symbols in i.

After these definitions of concept and learnability, we present a simple learning algorithm (adapted to our model from Valiants paper [10]), but first a combinatorial definition:

Given a real number $h > 1$ and a positive integer S, let <u>L(h,S)</u> denote the least integer so the following is true: In L(h,S) independent Bernoulli-trials each with probability at least

h^{-1} of success, the probability of having fewer than S successes is less than h^{-1}.

The text of the learning algorithm is as follows:

<u>Algorithm 1</u>:

```
begin
    i_1 ≔ i_0           (: ext(i_0)=∅ :)
    T ≔ 0               (: no trials have been made yet :)
    S ≔ 0               (: no successes have been registered :)
    repeat
        r ≔ example(2h); T ≔ T + 1
        if oracle(r) and r ∉ ext(i_1) then
            i_1 ≔ improve(i_1,r);  S ≔ S + 1
    until T ≥ L(2h,S+1)
end
```

In algorithm 1, we start by an empty initial approximation, which is stepwise improved by means of typical objects that are instances of the concept to be learned, although they are not accounted for by the present hypothesis. The "improve"-function may use oracle, but need not be specified in detail to reason about the partial correctness of algorithm 1:

<u>Theorem 1</u>: Let the following quantities be given:

i) A set of objects and a polynomium p

ii) A context (R,D) bounded by p and a concept (i), accessible through respectively example and oracle

iii) An approximation parameter h

iv) An initial approximation (the concept i_0)

v) A detail-specification of the "improve"-function of algorithm 1 such that this function, when being input an approximation (i_1) and an object r such that $r \in \Delta ext(i,i_1)$ outputs an improved approximation (i_2) such that $r \notin \Delta ext(i,i_2) \subsetneq \Delta ext(i,i_1)$.

Then we may conclude: If algorithm 1 halts then i_1 is an h-approximation for (i) with probability at least $(1-h^{-1})$.

<u>Proof</u>: Suppose the algorithm halts without finding an h-approximation. In this situation we have made $L(2h,S+1)$. Bernoulli-trials (i.e., calls of example) each with probability at least $(2h)^{-1}$ (lower bounded by $h^{-1} - \sum_{size(r)>l_h} D(r)$) of success (i.e., finding a "small" object r that is not explained by the present approximation i_1: $r \in \Delta ext(i,i_1)$) and obtained fewer than $S+1$ successes (namely S successes each resulting in a call of the improve-function). The probability that this situation may occur is (by def. of $L(h,S)$) less than $(2h)^{-1} (\leq h^{-1})$.

q.e.d.

Theorem 1 establishes a paradigm for learning algorithms. To fulfil the requirements of polynomial time complexity, we can use an equality obtained by Valiant [10]:

<u>Theorem 2</u>: $L(h,S) \leqq 2h(S+\log h)$ for $h > 1$, $S \geqq 1$.

Thus to obtain a concrete result of learnability, we need only specify the improve function of algorithm 1 in such a way that at most S_0 successive calls are needed to obtain a perfect identification of the unknown object, where S_0 is polynomially dependent on h and the number of symbols in the intension of the unknown concept. In addition each call of "improve" should have polynomial time complexity.

We have described our general model of learnability. As a specific instance we might consider the language of Boolean expressions and obtain a specific model similar to Valiant's model albeit a bit different. Valiant proves 3 different classes of concepts to be learnable, but only one of those may be transferred into our setting, namely the result on monotone DNF-expressions ([10],Th. B). Valiant allows polynomial time complexity with respect to a third parameter, viz. the number of propositional variables. This enables him to prove the class of k-CNF-expressions to be learnable for any k.

Our models inability to support the latter result is not necessarily a weekness. One may dislike the underlying learning algorithm, since the learner has to know k. Previously, we transformed Valiant's basic fixed-time learning algorithm into our finite time learning algorithm (Alg. 1), and hence allowed the learner to be ignorant of the size of the intension of a concept to be learned. Such a trick is not applicable to k, however, since the time complexity of Valiant's learning algorithm is exponential in k.

In addition Valiant considers more sophisticated oracles, as a means of access to the unknown concept, and this leads to another class of learnable expressions.

To sum up, our model has less expressive power than Valiant's model. In return it is defined from more general notions. Whether our model has a "good" structure can only be evaluated by using it. First order logic is usually considered as <u>the</u> language of formal (mathematical) reasoning, i.e., it is commonly believed that all apparantly extra logical assumptions can be made explicit within first order logic. Therefore it would appear a reasonable "test" for our model to find a learnability result within predicate logic. This is the goal of the next section.

3. Learnability within predicate logic

A specialisation of our model to predicate logic could possibly be done in several ways. In our approach a concept is defined by some properly restricted set of axioms (the intension) and the extension consists of all ground atomic formulae provable from this axiom system. We present the necessary formal definitions:

Context: To fill our model, we need only define the set of objects as some language over a finite alphabet. In order to obtain countably infinite sets of function, predicate and variable symbols (F, P and V respectively), we choose the following alphabet {F,P,V,0,1} (ex.: the 11th function symbol (f_{11}) is represented by the sting "F1011"). We assume that each function and predicate symbol has a specific arity. The set of groundterms T_0 is defined inductively:

i) Basis: if $f \in F$ has arity 0 then $f \in T_0$.

ii) Induction step: if $f \in F$ has arity $k \geq 1$ and $t_1,\ldots,t_k \in T_0$ then $f(t_1,\ldots,t_k) \in T_0$.

The set of objects, which coincide with the set of ground atomic formulae is $R = \{p(t_1,\ldots,t_n) \mid p \in P$ has arity n and $t_1,\ldots,t_n \in T_0\}$.

Concept: We need only specify the set of possible intensions. An intension is some axiomatization of the objects included in the concept. In addition we must insist that such axiomatization can form the basis of a recognition algorithm with polynomial time complexity.

We have now specified a model of learnability for predicate logic. Within this model, we shall prove a class of concepts to be learnable. This class consists of those concepts that have an intension in the form of a finite set of unit classes. In order to describe this class we need a definition of atomic formula, i.e., the above ground atomic formulae with variables allowed. The set of terms is defined inductively:

i) Basis: every variable $v \in V$ belong to T, and if $f \in F$ has arity 0 then $f \in T$.

ii) Induction step: If $f \in F$ has arity $k \geq 1$ and $t_1,\ldots,t_k \in T$ then $f(t_1,\ldots,t_k) \in T$.

The set of atomic formulae is now AF = $\{p(t_1,\dots,t_n) \mid p \in P$ has arity n and $t_1,\dots,t_n \in T\}$. There is a natural quasiorder on AF: $a_1 \sqsubseteq a_2 \Leftrightarrow \exists\, \theta \in$ substitutions. $a_1\,\theta = a_2$. This order reflects specialisation/generalisation: $p(x_1,x_2) \sqsubsetneq p(x_1,x_1)$ means that $p(x_1,x_2)$ is a proper generalisation of $p(x_1,x_1)$. In predicate logic the individual undefined values are labelled. This precaution enables us to express relation between undefined values (e.g., equivalence: $p(x_1,x_1)$). This is the only reason for naming undefined values. Hence the individual names of variables may be chosen arbitrarily.

Let us define a positive unit clause as a universally quantified atomic formula: UC = $\{\forall x_1,\dots,x_k a \mid a \in AF,\ \{x_1,\dots,x_k\}$ is precisely those variables that occur in $a\}$. We form the class of unit clause concepts: $C_1 = \{U \mid U$ is a finite set of unit clauses: $U \subseteq UC\}$.

Let us first see that a recognition algorithm exists for any member of C_1 and thereby verify that C_1 is a class of concepts. Observe that an object r belong to the set of theorems derivable from the axiomset given by $i = \{\forall x_1 \dots x_{k_1} a_1,\dots,\forall y_1,\dots,y_{k_n} a_n\}$ if and only if $a_i \sqsubseteq r$ for some $1 \leq i \leq n$ (This follows from the resolution theorem [7]). Furthermore a recognition algorithm for ext(i) can be based on Robinson's resolution principle [7] and consequently is of polynomial time complexity.

We may now state

<u>Theorem 3</u>: Given the above set of objects R and a polynomium p, then C_1 is learnable via algorithm 1.

Proof: We are going to specify the improve-function of algorithm 1 such that

a) we obtain a perfect identification of the unknown concept within at most $S_0 = d \cdot l$ successive calls of improve, where d is the number of unit clauses in the intension of the concept to be learned and $l = p(2h)$.

b) each call of improve fulfil the requirements of polynomial time complexity.

According to Th. 1 and 2, a) and b) will suffice for a proof. We represent a unit clause by the corresponding atomic formula and consequently we represent an intension (and an approximation) by a set of atomic formulae. The initial approximation is the empty set. Every approximation is represented without redundancy because we obey the following rule:

(*) Every pair of atomic formulae r_1, r_2 in an approximation fulfil: No common generalization of r_1 and r_2 is included in the extension of the concept to be learned.

Improve works as follows: A new typical object (ground atomic formula) is added to the present approximation, if (*) is not violated. Otherwise some atomic formula in the present approximation is replaced by the least general generalisation (lgg) of this atomic formula and the new typical object. The lgg of two objects is another name for the greatest lower bound (glb) of these with respect to the $\sqsubseteq$-order that we have defined on the set of atomic formulae. Since $\sqsubseteq$ is a quasiorder, glb is only defined modulo the equivalence ($\sqsubseteq \cap \sqsupseteq$). This amounts to saying that the individual names of variables in an atomic formulae may be chosen arbitrarily. The lgg is computed by a simple algorithm, which was discovered independently by Reynolds [6] and Plotkin [5].

The details of improve are the following:

```
function improve (in i_1: approximation; r: object out: i_2 approximation)
    begin
       improved := false
       for r_1 ∈ i_1 do
          ┌─────────────────────────────────────────────────────┐
          │ if not improved and "r and r_1 begin with the same  │
          │    predicate symbol" then                            │
          │    r_2 := r_1 ⊓ r;                                   │
          │    if oracle (r_2) then                              │
          │       i_2 := (i_1 \ {r_1}) ∪ {r_2}                   │
          │       improved := true                               │
          └─────────────────────────────────────────────────────┘
       if not improved then
          i_2 := i_1 ∪ {r}
    end
```

Let us estimate the maximal number of successive calls of "improve" that are needed to obtain a perfect approximation. There are two sorts of improvements:

(1) An atomic formula a belonging to the approximation is generalised. Suppose a can be generalised k times, i.e., there exists a sequence of atomic formulae $a_1, \ldots, a_k$ such that $a_1 \sqsubsetneq a_2 \sqsubsetneq \ldots \sqsubsetneq a_k = a$. Reynolds [6] has proved that $k \leq \text{length}(a) - \#\{x \in \mathcal{V} \mid x \text{ occur in } a\}$. Here $\text{length}(a) \leq \text{size}(a)$ denotes the number of symbols in a with respect to the infinite alphabet $\mathcal{F} \cup \mathcal{P} \cup \mathcal{V}$. We know size(a) is bounded by $l = p(2h)$. Hence the maximal number of successive generalisations of a single atomic formula in an approximation is also bounded by l.

(2) An object is added to the approximation. Observe that every atomic formula a_1 in an approximation must be a specialisation instance of some atomic formula a_2 in the intension of the concept to be learned, $a_2 \sqsubseteq a_1$ (This follows from the resolution theorem [7], since the approximation is a sound although possibly incomplete axiomatization of the extension of the concept to be learned).

Furthermore, two different atomic formulae belonging to a single approximation can not be specialisation instances of a single atomic formula in the intension of the concept to be learned (this is assured by (*)). The number of atomic formulae in this intension is d. Hence the maximal number of additions of an object to a specific approximation is bounded by d.

In summary a perfect identification is obtained within $S_0 = d \cdot l$ successive calls of improve.

Finally, we must verify the polynomial time complexity of "improve". lgg(Π) is the only potentially time consuming operation. By examining the algorithms of Reynolds and Plotkin, we obtain a quadratic upperbound on the time complexity of this procedure: $O(l^2) \leq O(p^2(h))$.

q.e.d.

4. Further work

It might be interesting to take some existing learning algorithm and describe its power in terms of our model, i.e., we should prove a specific class of concepts to be learnable by means of one such algorithm. Shapiro [8] and Summers [9] have described algorithms for inferring PROLOG-programs and LISP-programs respectively. However, our naturally distributed examples may be a too week learning tool for these algorithms: Shapiro mentions some results that he has achieved by running his algorithm. One gets the impression that these results are due to intelligently chosen (by Shapiro) examples, since he does not describe what examples the algorithm needs in order to succeed fastly. In contrast Summers' algorithm needs a clearly described set of examples in order to succeed. Yet, this latter set of examples is not directly describable by the notion of typicalness provided by our model. In summary, there seems to be no simple relation between our model of learnability and the assumptions underlying Shapiro's and Summers' algorithms. Still it might be worth the effort to investigate the nature of a possible relation, since especially Summers' results are quite elegant.

We have formed one class of learnable concepts and the above discussion indicates a possible way to find more classes. This is, however, a week result that does not tell us much about the limits of learnability according to our model. It would be interesting to achieve a completeness result, i.e., we should give a simple characterisation of all learnable classes of concepts. Yet, this could be a difficult task, since a non-trivial result would allow us to exhibit an unlearnable class of concepts, which implies a lowerbound on the probabilistic time complexity of any learning algorithm for this particular unlearnable class of concepts. Traditionally, it has been difficult to establish non trivial lowerbounds on the time complexity of specific (natural) problems [2]. Hence, a completeness result may be difficult to obtain.

References

[1] Angluin, D. and Smyth, C. H.: "Inductive inference: Theory and methods". Computing Surveys, Vol. 15, 1983, pp. 237-269.

[2] Cook, S. A.: "An overview of computational complexity". Communications of the ACM, Vol. 26, 1983, pp. 401-408.

[3] Gold, E. M.: "Language identification in the limit". Information and Control, Vol. 10, 1967, pp. 447-474.

[4] Larsen, S. F.: "Egocentric speech, concept structure, and semantic development" (in danish). Nordisk Psykologi, Vol. 32, 1980, pp. 55-73.

[5] Plotkin, G. D.: "A note on inductive generalization". In Machine Intelligence 5 (eds.: Meltzer, B. and Michie, D., Edinburgh University Press), 1970, pp. 153-163.

[6] Reynolds J. C.: "Transformational Systems and the algebraic structure of atomic formulas". In Machine Intelligence 5 (eds.: Meltzer, B. and Michie, D., Edinburgh University Press), 1970, pp. 135-151.

[7] Robinson, J. A.: "A machine-Oriented logic based on the resolution principle". Journal of the ACM, Vol. 12, 1965, pp. 23-41.

[8] Shapiro, E. Y.: "An algorithm that infers theories from facts". In Proceedings of the 7th IJCAI (Vancouver, Canada), 1981, pp. 446-451.

[9] Summers, P. D.: "A methodology for LISP-program construction from examples". Journal of the ACM, Vol. 24, 1977, pp. 161-175.

[10] Valiant, L. G.: "A theory of the learnable", Communications of the ACM, Vol. 27, 1984, pp. 1134-1142.

[11] Valiant, L. G.: "Deductive learning". In "Mathematical logic and programming languages" (eds.: Hoare and Shepherdson, Prentice Hall Inc.), 1985, pp. 107-112.

[12] Wharton, R. M.: "Approximate language identification". Information and control, Vol. 26, 1974, pp. 236-255.