PAC Learning under Helpful Distributions*

François Denis, Rémi Gilleron
LIFL, URA 369 CNRS, Université de Lille I†
and Université Charles de Gaulle, Lille 3

Abstract

A PAC model under helpful distributions is introduced. A teacher associates a teaching set with each target concept and we only consider distributions such that each example in the teaching set has a non-zero weight. The performance of a learning algorithm depends on the probabilities of the examples in this teaching set. In this model, an Occam's razor theorem and its converse are proved. The class of decision lists is proved PAC learnable under helpful distributions. A PAC learning model with simple teacher (simplicity is based on program-size complexity) is also defined and the model is compared with other models of teaching.

1 Introduction

It seems that many concept classes are not polynomially learnable in the basic PAC learning model of Valiant([20]). One reason is the distribution free requirement (the learning must work with respect to an arbitrary distribution) even if providing examples to the learner and evaluating the output hypothesis is performed with the same distribution.

In practical learning situations, the examples given are likely to be chosen so that they are "representative" of the target concept rather than random or arbitrary. Therefore, it seems reasonable to investigate learning methods that assume that the source of examples is "helpful".

There are several ways to assume that examples are not arbitrary: the learner may ask queries (see [2] for an overview); the class of distributions used to draw examples can be restricted, thus the learning algorithm knows something about the underlying distribution ([3], [11], [15], [13]); a teaching set may be designed

*This research was partially supported by "Motricité et Cognition : Contrat par objectifs région Nord/Pas-de-Calais

†LIFL, Bat. M3, USTL, 59655, Villeneuve d'Ascq Cedex, France, e-mail: {denis,gilleron}@lifl.fr

in order to help the learner ([7], [19], [8]). For instance, Goldman and Mathias assume that the teacher builds a teaching set related to the target concept and that an adversary adds new examples to this set in order to prevent collusion between the learner and the teacher. Then, the learner must identify exactly the target from this set of examples.

In the present paper, a new learning model with a helpful source of examples is proposed. It is supposed that the teacher knows a representation of the target concept and uses this representation to define a teaching set. The criterion of success is the PAC criterion. But, we only consider helpful distributions i.e. such that examples in the teaching set have a non zero probability to be drawn. Moreover the time requirement depends on the least probability of an example in the teaching set to be drawn according to the probability distribution. The fact that a learning algorithm must learn under *all* distributions not null on the teaching set prevents, in some sense, collusion between the learner and the teacher while the restriction on the class of distributions suffices to make statistical inferences feasible.

In PAC learning theory, the Occam's Razor Theorem of Blumer et al. [4] is one of the most important results. Let us recall that an algorithm is an Occam algorithm if it finds a short hypothesis consistent with the observed data. The Occam's razor theorem states that any efficient Occam algorithm is also a PAC learning algorithm. This theorem provides a formal justification of the Occam principle. We prove an Occam's razor theorem in the PAC learning model under helpful distributions. The main difference is that we use multisets of examples and that the size condition depends on the frequencies of examples of the teaching set in the multisample. Using this theorem, we prove that decision lists are PAC learnable in our model. Our Occam algorithm for decision lists supposes that examples in the teaching set are frequent in the multisample. Consequently, examples are examined by decreasing multiplicity. This corresponds to a usual heuristic in practical learning algorithms. Like in the usual PAC setting (see [17]), we prove a converse of the Occam' razor theorem for many natural classes.

Finally, we define a simple PAC learning model and compare it with the model of Li and Vitányi ([13]). A teacher is simple if for each concept, the examples of the teaching set are of low conditional Kolmogorov complexity relatively to the target. We consider PAC learnability under helpful distributions for a simple teacher. We can use our Occam's Razor Theorem to find new proofs of simple learnability results in the sense of Li and Vitányi (even those which did not seemed possible to be stemed from an Occam algorithm [5]).

Our model is defined in Section 2. The Occam's razor theorem is given and proved in Section 3. Learnability of decision lists is proved in Section 4. Our simple PAC learning model is defined in Section 5.

2 PAC Learning under Helpful Distributions

2.1 Definitions and Notations

Let $\mathcal{B}_n$ be the set of boolean functions from $X_n = \{0,1\}^n$ into $\{0,1\}$. Let $\mathcal{B} = \cup_{n\geq 1}\mathcal{B}_n$. A class $\mathcal{F}$ of boolean functions is a subset of $\mathcal{B}$. A *representation scheme* for a class of boolean functions $\mathcal{F}$ is a function $R : \mathcal{F} \to 2^{\Sigma^*}$ where Σ is a finite alphabet and such that for each f and f' in $\mathcal{F}$, $R(f)$ is not empty and if $f \neq f'$, $R(f) \cap R(f') = \emptyset$. We suppose that R is computable in polynomial-time, that is, there exists a polynomial-time deterministic algorithm which takes as input a pair of strings x and c and outputs 1 if $f(x) = 1$ with $c \in R(f)$, and 0 otherwise. A *concept class* C is defined by $C = \cup_{f\in\mathcal{F}} R(f)$. We will identify a concept c in C and the function f which is represented by c. We define the size of a concept c as its length $|c|$ and we suppose as usual that $|c| \geq n$.

An *example* of a concept c is a pair $(x, c(x))$, where x is in the domain of c. An example $(x, c(x))$ is *positive* if $c(x) = 1$ and *negative* otherwise. We denote by $EX(c)$ (respectively $POS(c)$, $NEG(c)$) the set of all examples (respectively positive examples, negative examples) of a concept c. A *sample* of c is a subset of $EX(c)$. A *multisample* of c is a multiset of examples of c. Let S_c be a sample of c and S be a multisample of c, $S_c \subseteq S$ if each example in S_c occurs at least once in S. Let c be a target concept over X_n and let P be any fixed probability distribution over X_n. Let $EX(c, P)$ be a procedure that runs in unit time and that at each call returns an example $(x, c(x))$, where x is drawn randomly and independently according to P. If c' is any concept in C over X_n, we define $error(c') = P(\{x \in X_n \mid c(x) \neq c'(x)\})$.

2.2 Definition of our model

Definition 1. Let C be a concept class. A *teaching set* for $c \in C$ is a sample of c. A *teacher for* C is a mapping $\mathcal{T}$ which associates with each concept c, a teaching set $\mathcal{T}(c)$. A teacher is *polynomial* if there is a constant k such that for every concept c, $Card(\mathcal{T}(c)) \leq |c|^k$. A teacher is *computable* if there exists an algorithm which takes as input a concept c in C, and produces as output the teaching set $\mathcal{T}(c)$.

Definition 2. Let C be a concept class and let $\mathcal{T}$ be a teacher for C. Let c be a target concept over X_n and let P be any fixed probability distribution over X_n. Let us define

$$P_{min}(c) = \begin{cases} min\{P(x) \mid (x, c(x)) \in \mathcal{T}(c)\} & \text{if } \mathcal{T}(c) \neq \emptyset, \\ 1 & \text{otherwise.} \end{cases}$$

A distribution P is *helpful* w.r.t. c and $\mathcal{T}$ if $P_{min}(c) \neq 0$.

We now define PAC learnability under helpful distributions.

Definition 3. Let C be a concept class and let $\mathcal{T}$ be a teacher for C.

- An algorithm $\mathcal{A}$ is a *PAC learning algorithm for C under helpful distributions* if $\mathcal{A}$ takes as input $\epsilon \in (0,1]$, $\delta \in (0,1]$, an integer l, and for all concepts c in C with $|c| \leq l$ and all helpful probability distributions P, $\mathcal{A}$ is given access to $EX(c,P)$ and $\mathcal{A}$ outputs some c' in C, such that with probability at least $1-\delta$, $error(c') \leq \epsilon$.

- C is *PAC learnable under helpful distributions $\mathcal{T}$* if there is a PAC learning algorithm $\mathcal{A}$ for C under helpful distributions which runs in time polynomial in $1/\epsilon$, $1/\delta$, l, and $1/P_{min}(c)$.

Our model generalizes the PAC learning model because a concept class C is PAC learnable if and only if C is PAC learnable with the empty teacher (for each concept c, the teaching set is empty). Goldman and Mathias have defined a teaching model ([8]) in which for each concept c, a teacher chooses a set of examples $T(c)$. The size of $T(c)$ must be polynomial but the teacher can have unbounded computation time. The learner gets the examples in the teaching set along with a set chosen by the adversary. The learner must output a hypothesis logically equivalent to c in polynomial time. Now let us consider a concept class C which is learnable in this model with learner L and teacher T. We consider the teacher $\mathcal{T} = T$. It is now easy to define a PAC learning algorithm $\mathcal{A}$ for C under helpful distributions : draw a large enough sample S (such that $Pr(\mathcal{T}(c) \subseteq S) \geq 1-\delta$) and output $L(S)$. The number of drawn examples is polynomial in $1/\delta$ and $1/P_{min}(c)$ and Algorithm $\mathcal{A}$ is probably exact.

We will also need PAC learnability in usually polynomial time.

Definition 4. Let C be a concept class and let $\mathcal{T}$ be a teacher. C is *PAC learnable under helpful distributions in usually polynomial time* if there is a PAC learning algorithm $\mathcal{A}$ for C under helpful distributions such that, with probability at least $1-\delta$, $\mathcal{A}$ halts in time polynomial in $1/\epsilon$, $1/\delta$, l, and $1/P_{min}(c)$.

As in Valiant's framework, if $\mathcal{A}(\epsilon, \delta, l)$ is a usually polynomial time PAC learning algorithm for C under helpful distributions, there exists a usually polynomial time PAC learning algorithm $\mathcal{A}'$ which only takes as input ϵ and δ.

3 Occam's Razor Theorem

3.1 Occam Learning

In this Section we give an Occam's Razor Theorem for our learning model.

Definition 5. Let C be a concept class and $\mathcal{T}$ be a teacher for C. Let S be a non empty multisample of some concept c in C. Let us define

$$f_{min}(S,c) = \begin{cases} min\{\frac{S(x,c(x))}{Card(S)} \mid (x,c(x)) \in \mathcal{T}(c)\} & \text{if } \mathcal{T}(c) \neq \emptyset, \\ 1 & \text{otherwise.} \end{cases}$$

where $S(x,c(x))$ denotes the number of occurences of $(x,c(x))$ in S.

Note that if P is the uniform distribution on S, we have $P_{min}(c) = f_{min}(S,c)$. Note also that if $\mathcal{T}(c) \subseteq S$, then $1/f_{min}(S,c) \leq Card(S)$.

Definition 6. Let C be a concept class and $\mathcal{T}$ be a teacher for C.

B is an Occam algorithm for $C, \mathcal{T}$ if there exists constants $a \geq 0$, $b \geq 0$ and $0 \leq \alpha < 1$ such that with a multisample S of c in X_n such that $\mathcal{T}(c) \subseteq S$ on input, B outputs a hypothesis concept c' such that:

- c' is consistent with S,
- $|c'| \leq a(|c|/f_{min}(S,c))^b (Card(S))^\alpha$,
- B runs in polynomial time in $|c|$, $Card(S)$.

Let us point out the differences between the definition of an Occam algorithm in Valiant's PAC learning model and the definition of an Occam algorithm in our model. First, an Occam algorithm in our model only has to work properly if the multisample S contains the teaching set $\mathcal{T}(c)$ of the target concept c. Second, c' need to be short only if $f_{min}(S,c)$ is not too small (polynomial in $1/|c|$), i.e. if S is drawn according to a helpful probability distribution P and if S is sufficiently large, $P_{min}(c)$ is not too small. In our model, given an Occam algorithm for $C, \mathcal{T}$, a PAC learning algorithm $\mathcal{A}$ for C under helpful distributions will iterate over larger guesses for $1/P_{min}(c)$.

We can now give our main theorem:

Theorem 1. *Let C be a concept class and let $\mathcal{T}$ be a polynomial teacher for C. If there is an Occam algorithm for $C, \mathcal{T}$, then C is PAC learnable under helpful distributions in usually polynomial time.*

3.2 Proof of the Occam's Razor Theorem

The proof is merely sketched. Let C be a concept class and let $\mathcal{T}$ be a polynomial teacher for C. Let k be a constant such that for every concept c, $Card(\mathcal{T}(c)) \leq (|c|)^k$. Let B be an Occam algorithm for $C, \mathcal{T}$ with constants (a,b,α). Let q denote the polynomial such that B has time complexity $q(|c|, Card(S))$. Let c be a concept. Whenever we consider the oracle $EX(c,P)$, we suppose that the probability distribution P is helpful w.r.t. c and $\mathcal{T}$. Let l be an integer such that $|c| \leq l$. We first prove two technical lemmas.

Lemma 1. *Let p be an integer such that $p \geq 1/P_{min}(c)$. Make $N_1(\delta, l, p) = \lceil 8p \log(l^k/\delta) \rceil$ calls to $EX(c, P)$. This defines a multisample S. Then the probability that $f_{min}(S, c) \geq P_{min}(c)/2$ is at least $1 - \delta$.*

Proof. The proof is based on the classical Chernoff bounds (see for instance [12], 190-192). □

Lemma 2. *Make $N_2(\epsilon, \delta, l, p) = \lceil ((\log(1/\delta) + a(2pl)^b + 1)/\epsilon)^{\frac{1}{1-\alpha}} \rceil$ calls to $EX(c, P)$. This defines a multisample S. Suppose that $f_{min}(S, c) \geq 1/2p$. Let c' be the output of the Occam algorithm B on input S. Then the probability that $error(c') > \epsilon$ is at most δ.*

Proof. Let S be a multisample of cardinality N and let us consider the set

$$H_c = \{c' \in C \mid error(c') > \epsilon \text{ and } |c'| \leq a(2pl)^b N^\alpha\}.$$

Let $c' \in H_c$. The probability that a call to $EX(c, P)$ returns an example consistent with c' is less than $1 - \epsilon$. Therefore the probability that N calls to $EX(c, P)$ return a multisample S consistent with c' is less than $(1 - \epsilon)^N$. Moreover, the cardinality of H_c is less than $2^{a(2pl)^b N^\alpha + 1}$. Consequently the probability that the multisample S of cardinality N is consistent with a concept c' in H_c is less than $2^{a(2pl)^b N^\alpha + 1}(1 - \epsilon)^N$. Verify that $2^{aN^\alpha (2pl)^b + 1}(1 - \epsilon)^N \leq \delta$ whenever $N \geq N_2(\epsilon, \delta, l, p)$.

Now let us suppose that $N \geq N_2(\epsilon, \delta, l, p)$ and $f_{min}(S, c) \geq 1/2p$. $f_{min}(S, c) > 0$, thus $\mathcal{T}(c) \subseteq S$. On input S the algorithm B outputs a hypothesis concept c' consistent with S such that

$$|c'| \leq a(|c|/f_{min}(S, c))^b (Card(S))^\alpha \leq a(2pl)^b N^\alpha.$$

Finally if $error(c') > \epsilon$, $c' \in H_c$. □

Now let us consider the hypothesis testing algorithm $TEST$ with parameters ϵ, δ, i, and c' (see [10]). The algorithm makes $\lceil (32/\epsilon)(i \ln 2 + \ln 2/\delta) \rceil$ calls to $EX(c, P)$ to test hypothesis c'. It accepts the hypothesis if the hypothesis is wrong on no more than a fraction of $\frac{3}{4}\epsilon$ of the examples returned by the oracle, and rejects it otherwise. $TEST(\epsilon, \delta, i, c')$ is polynomial in $1/\epsilon, 1/\delta, i, |c'|$. And we have:

Lemma 3. *(Haussler et al. [10]) The test $TEST(\epsilon, \delta, i, c')$ has the property that: when $error(c') \geq \epsilon$, the probability is at most $\delta/2^{i+1}$ that the test will accept c', when $error(c') \leq \epsilon/2$, the probability is at most $\delta/2^{i+1}$ that the test will reject c'.*

PAC Learning Algorithm $\mathcal{A}$ for $C, \mathcal{T}$
input: ϵ, δ, l
begin

Set S to $\emptyset$ - - S is the current multisample
Set p to 1 - - p is the current guess for $1/P_{min}(c)$
loop
Set N to $sup\{N_1(\delta/3, l, p), N_2(\epsilon/2, \delta/3, l, p)\}$
Call $EX(c, P)$ until $Card(S) = N$
Run at most $q(l, N, 2p)$ steps of B on input S
if B outputs some c'
and c' consistent with S **and** $|c'| \leq a(pl)^b N^\alpha$
and $TEST(\epsilon, \delta/3, p, c')$
then output c' and halt
endif
Set p to p+1
endloop
end

We now prove that $\mathcal{A}$ is a PAC learning algorithm for C under helpful distributions. When the learning algorithm $\mathcal{A}$ halts at some step p, the probability that $error(c') > \epsilon$ is at most $\delta/(3 \times 2^{p+1})$. This is due to the halting condition $TEST(\epsilon, \delta/3, p, c')$, and to Lemma 3. Therefore, when the learning algorithm $\mathcal{A}$ halts, the probability that $error(c') > \epsilon$ is at most $\delta/3$.

It remains to prove that, with probability at least $1-\delta$, $\mathcal{A}$ halts in polynomial time in $1/\epsilon$, $1/\delta$, l, and $1/P_{min}(c)$. Let $p = \lceil 1/P_{min}(c) \rceil$ and $N \geq N_1(\delta/3, l, p)$. Then, the probability that $f_{min}(S, c) \geq P_{min}(c)/2$ is at least $1 - \delta/3$ (Lemma 1). Suppose $f_{min}(S, c) \geq P_{min}(c)/2$ and $N \geq N_2(\epsilon/2, \delta/3, l, p)$. Then, the probability that $error(c') > \epsilon/2$ is at most $\delta/3$ (Lemma 2). Suppose now that $error(c') \leq \epsilon/2$, the probability that the test $TEST(\epsilon, \delta, p, c')$ will reject c' is at most $\delta/(3 \times 2^{p+1})$ (Lemma 3). Therefore, the probability that the algorithm $\mathcal{A}$ does not halt at step $p = \lceil 1/P_{min}(c) \rceil$ is at most δ and the probability that the learning algorithm $\mathcal{A}$ does not halt before $p = \lceil 1/P_{min}(c) \rceil$ is at most δ. It is now easy to verify that if the algorithm $\mathcal{A}$ halts before $p = \lceil 1/P_{min}(c) \rceil$, then the running time is bounded by a polynomial in $1/\epsilon$, $1/\delta$, l, and $1/P_{min}(c)$. This ends the proof.

3.3 Converse of the Occam's Razor Theorem

As in the usual PAC setting, a converse of the Occam theorem holds for concept classes which are strongly closed under exception ([17], see also [16]):

Theorem 2. *If $\mathcal{C}$ is PAC learnable under helpful distributions in usually polynomial time and if $\mathcal{C}$ is strongly closed under exception then there exists a randomized algorithm $\mathcal{B}$ such that with δ and a multisample S of c such that $\mathcal{T}(c) \subseteq S$ on input, then with probability at least $1 - \delta$, $\mathcal{B}$ outputs a hypothesis concept c' such that:*

- *c' is consistent with S,*
- *there exists constants $a \geq 0, b \geq 0$ and $\alpha < 1$ which do not depend on S and c such that $|c'| \leq a(|c|/f_{min}(S,c))^b |S|^\alpha$,*
- *$\mathcal{B}$ runs in polynomial time in $\log(1/\delta)$, $Card(S)$ and $|c|$.*

Proof. We just remark here that the uniform distribution on S is helpful and that it can be simulated by a randomized algorithm. The proof is similar to the one for the classical PAC model. □

4 Learning Decision Lists

A *decision list* over $x_1, \ldots, x_n$ is an ordered sequence

$$L = (m_1, b_1), \ldots, (m_p, b_p)$$

of terms, in which each m_i is a monomial over $x_1, \ldots, x_n$, and each $b_i \in \{0,1\}$. The last monomial is always $m_p = 1$. For any input $a \in \{0,1\}^n$, the value $L(a)$ is defined as b_i, where i is the smallest index satisfying $m_i(a) = 1$. We consider polynomial-time binary representations of decision lists such that if c represents a decision list of p terms over n variables, then $|c| = O(np)$. The class of k-decision lists (each monomial contains at most k litterals) are PAC learnable ([18]). Decision lists are not known to be PAC learnable, but lower bounds on learning decision lists are given in [9]. The class of k-decision lists is learnable in the teaching model of [8]. Therefore it is probably exactly learnable under helpful distributions. We prove in this Section that the concept class of decision lists is PAC learnable under helpful distributions. Note that decision lists are a superset of DNF formulas and thus DNF formulas are learnable using decision lists. It can be proved, using a greedy heuristic, that DNF formulas are learnable under helpful distributions.

First, let us define the teacher $\mathcal{T}$. Let m be a monomial over $x_1, \ldots, x_n$. Let $0_m \in X_n$ be defined by $m(0_m) = 1$ and vector 0_m has the value 0 for variables not in m. Let $1_m \in X_n$ be defined by $m(1_m) = 1$ and vector 1_m has the value 1 for variables not in m ([13]). Let c be a representation of a decision list $L = (m_1, b_1), \ldots, (m_p, b_p)$, we define

$$\mathcal{T}(c) = \{(0_{m_i}, c(0_{m_i})) \mid 1 \leq i \leq p\} \cup \{(1_{m_i}, c(1_{m_i})) \mid 1 \leq i \leq p\}.$$

The purpose of the teaching set for a target decision list c is to allow the construction of the monomials of c. Indeed, let x, x' in X_n, we define $x \diamond x'$ as the monomial over $x_1, \ldots, x_n$ which contains x_i if x and x' have '1' in position i, contains $\overline{x_i}$ if x and x' have '0' in position i, and does not contain variable x_i otherwise ($1 \leq i \leq n$). Given the teaching set $\mathcal{T}(c)$ of some decision list c, the set $\{m = x \diamond x' \mid (x, c(x)), (x', c(x')) \in \mathcal{T}(c)\}$ contains the monomials of c.

Proposition 1. *Decision lists are PAC learnable under helpful distributions.*

Proof. We denote by D the class of decision lists. We give an Occam algorithm for D, $\mathcal{T}$ which is based on the Rivest's algorithm for k-decision lists, and then we use our Occam's razor theorem. Our algorithm uses examples in S by decreasing number of occurences until we are ensured that all examples in the teaching set have been used.

Occam Algorithm B
input: a multisample S of a target concept c
begin
  Set i to 1; set hypothesis concept c' to the empty list
  while $S \neq \emptyset$
    Set S_i to $\{(x, c(x)) \in S \mid S(x, c(x)) \geq \frac{Card(S)}{i}\}$
    Set M_i to $\{m = x \diamond x' \mid (x, c(x)) \in S_i, (x', c(x')) \in S_i\}$
    while there is a monomial m in M_i satisfied by an example (y, b) in S and by no example $(z, \overline{b})$ in S
      Set c' to $c' + (m, b)$
      Set S to $S - \{(y, b) \in S \mid m(y) = 1\}$
    endwhile
    Set i to $i + 1$
  endwhile
  output hypothesis concept c'
end

It is now easy to prove that B is an Occam algorithm for D, $\mathcal{T}$. Let c be the target decision list. Let us suppose that $\mathcal{T}(c) \subseteq S$, and let $j = \lceil 1/f_{min}(S, c) \rceil$. B halts at most at step $i = j$ because the set M_j contains all the monomials of c. The hypothesis concept c' is consistent with S and the length of c' is bounded by $Card(M_j) = (\lceil 1/f_{min}(S, c) \rceil)^2$. As $1/f_{min}(S, c) \leq Card(S)$, it is easy to prove that B runs in polynomial time in $|c|$, $Card(S)$. Now, Theorem 1 can be applied. □

5 Simple PAC Learning Models

The reader may refer to [14] for complete definitions, proofs and guide-lines in the litterature on Kolmogorov complexity.

The definition of a *simple PAC learning algorithm* (Li and Vitányi [13]) is the same as the definition of a PAC learning algorithm except that the class of probability distributions is restricted to the universal Solomonoff-Levin distribution $\mathbf{m}$ which is defined w.r.t. a reference universal prefix Turing machine U.

Some classes were shown simple PAC learnable in [13]. Castro and Balcàzar have proved that log-n decision lists (where each term is of Kolmogorov complexity $O(\log n)$) are simple PAC learnable in [5]. It is not clear whether the sets of simple PAC learnable classes depend on the reference prefix Turing machine but one can easily prove that the simple PAC learnable classes of ([13],[5]) are simple PAC learnable for all reference prefix Turing machines. An Occam theorem can be proved in the simple PAC learning framework but it seems doubtful to obtain a converse of this result. Another drawback is that $\mathbf{m}$ is not computable.

Definition 7. Let C be a concept class. $\mathcal{T}$ is a *simple teacher* if there exists a constant $k > 0$ satisfying:

$$\forall c \in C \ \forall (x, c(x)) \in \mathcal{T}(c) \ K(x|c) \leq k \log(|c|).$$

Recall that $K(x|c)$ is the conditional complexity of x w.r.t. c, i.e. the length of a least self delimited program which computes x from c. We note that the simplicity of a teacher does not depend on the reference prefix Turing machine U.

Proposition 2. *A simple teacher is polynomial.*

Proof. The cardinality of the set of strings of length lower than $k \log(|c|)$ is bounded by $|c|^{k+1}$. □

Proposition 3. *Let C be a concept class. Let $\mathcal{T}$ be a polynomial and computable teacher. Then $\mathcal{T}$ is a simple teacher.*

Proof. Let c be a concept in C. Let $(x, c(x))$ be an example in $\mathcal{T}(c)$. We have: $K(x|c) \leq K(x|\mathcal{T}(c)) + K(\mathcal{T}(c)|c) + O(1)$. Since $\mathcal{T}$ is polynomial, $Card(S_c) \leq |c|^k$, therefore we get $K(x|\mathcal{T}(c)) \leq k \log(|c|) + O(1)$. Moreover, since $\mathcal{T}$ is computable, $K(\mathcal{T}(c)|c) \leq O(1)$. □

For instance, the teacher we have defined for the class of decision lists is polynomial and computable therefore it is a simple teacher. We now define our simple PAC learning model:

Definition 8. Let C be a concept class. C is *PAC learnable with simple teacher* if C is PAC learnable under helpful distributions for some simple teacher.

Note that the definition of PAC learnability with simple teacher does not depend on the reference prefix Turing machine U. As a corollary of Proposition 3, we get:

Proposition 4. *Let C be a concept class. If C is PAC learnable under helpful distributions for a polynomial and computable teacher, then C is PAC learnable with simple teacher.*

For instance, we have proved that decision lists are PAC learnable under helpful distributions for a polynomial and computable teacher. Therefore decision lists are PAC learnable with simple teacher.

In order to prove that simple classes of concepts are simple PAC learnable in the sense of [13], we can often show that a more general class is learnable with simple teacher and then use the next proposition.

Proposition 5. *Let C be a concept class. Suppose C is PAC learnable with a simple teacher $\mathcal{T}$. Let k be an integer and let us define the concept class*

$$C_k = \{c \in C \mid \forall (x, c(x)) \in \mathcal{T}(c) \;\; K(x) \leq k \log(|c|)\}.$$

For each k, the concept class C_k is simple PAC learnable.

For instance, since decision lists are PAC learnable with simple teacher, the previous result provides a new proof of the simple PAC learnability of $\log n$-decision lists. Note that this proof uses an Occam algorithm contrary to what was announced in [5].

References

[1] D.Angluin, Learning regular sets from Queries and Counterexamples, *Inform. and Computation* **75** (1987) 87-106.

[2] D.Angluin, Queries and Concept Learning, *Machine Learning* **2** (1988) 319-342.

[3] G.Benedek and A.Itai, Learnability by fixed ditribution, *Proc. 1st ACM Workshop on Computational Learning Theory*, (1988) 80-90.

[4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth, Occam's razor, *Inform. Proc. Lett.* **24** (1987) 377-380.

[5] J. Castro and J. L. Balcazar, simple PAC learning of simple decision lists, *Proc. 6th International Workshop on Algorithmic Learning Theory*, (1995).

[6] F. Denis, C. D'Halluin and R. Gilleron, PAC Learning with Simple Examples, *Proc. 13th Symposium on Theoretical Aspects of Computer Science*, (1996) 231-242.

[7] S. Goldman and M. Kearns, On the Complexity of Teaching, *J. of Computer and System Sciences* **50**, (1995) 20-31.

[8] S. A. Goldman and H. D. Mathias, Teaching a Smarter Learner, *J. of Computer and System Sciences* **52**, (1996) 255-267.

[9] T. Hancock, T. Jiang, M. Li and J. Tromp, Lower Bounds on Learning Decision Lists and Trees, *Inform. and Computation* **126** (1996) 114-122.

[10] D.Haussler, M.Kearns, N.Littlestone and M.Warmuth, Equivalence of models for polynomial learnability, *Proc. 1st ACM Workshop on Computational Learning Theory*, (1988) 42-55.

[11] M.Kearns, M.Li, L.Pitt and L.G.Valiant, On the learnability of boolean formulae, *Proc. 19th ACM Symposium on Theory of Computing*, (1987) 285-295.

[12] M. Kearns and V. Vazirani, An Introduction to Computational Learning Theory, MIT Press, (1994).

[13] M.Li and P.Vitányi, Learning simple concepts under simple distributions, *SIAM J. Comput.* **20** (1991) 911-935.

[14] M.Li and P.Vitányi, An introduction to Kolmogorov complexity and its applications, *Texts and Monographs in Computer Science* , Springer Verlag, (1993).

[15] B.K.Natarajan, On learning boolean functions, *Proc. 19th ACM Symposium on Theory of Computing*, (1987) 296-304.

[16] B.K.Natarajan, Machine Learning: a theoretical approach, Morgan Kaufman, (1991).

[17] On the necessity of Occam algorithms, *Proc. 22th ACM Symposium on Theory of Computing*, (1990) 54-63.

[18] R.Rivest, Learning Decision Lists, *Machine Learning* **2**, (1987) 229-246.

[19] A. Shinohara and S. Miyano, Teachability in computational learning, *New Generation Computing* **8** (1991) 337-347.

[20] L.G.Valiant, A theory of the learnable, *Comm. ACM.*, (1984) 1134-1142.