PAC Learning from Positive Statistical Queries*

François Denis

Bât. M3, LIFL, Université de Lille I
59655 Villeneuve d'Ascq Cedex, France,
denis@lifl.fr

Abstract. Learning from positive examples occurs very frequently in natural learning. The PAC learning model of Valiant takes many features of natural learning into account, but in most cases it fails to describe such kind of learning. We show that in order to make the learning from positive data possible, extra-information about the underlying distribution must be provided to the learner. We define a PAC learning model from positive and unlabeled examples. We also define a PAC learning model from positive and unlabeled statistical queries. Relations with PAC model ([Val84]), statistical query model ([Kea93]) and constant-partition classification noise model ([Dec97]) are studied. We show that k-DNF and k-decision lists are learnable in both models, i.e. with far less information than it is assumed in previously used algorithms.

1 Introduction

The PAC learning model of Valiant ([Val84]) has become the reference model in computational learning theory. However, in spite of the importance of learning from positive examples in natural learning, extending the PAC model in order to modelize this kind of learning seems difficult. The reason for it is that it does not exist any good way to define the learning error. Suppose for example that f is the target concept, that h is a hypothesis and let μ be the underlying distribution. If the error is measured relatively to the positive examples of f, i.e. if $error(h) = \mu_f(f \Delta h)$, then over-generalization seems unavoidable: the "full" concept (Σ^* for languages, function $\mathbf{1}$ for boolean functions) is always a good answer. But if the error is measured over all the examples, i.e. if $error(h) = \mu(f \Delta h)$, the learner cannot differentiate between different distributions whose restrictions on the positive examples of f are equal. Consequently, the output concept must always be included into the target concept and the learning boils down to learning with one-sided error ([Nat87], [Nat91]). But since the underlying distribution can be equal to 0 on some positive examples of the target, a learning algorithm will not be able to use *missing* examples to infer *negative* ones. As a result, it is often impossible to be sure that a hypothesis is included into the target concept. To sum up, in most cases, positive examples provide not enough information

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

to learn in the PAC framework ([Shv90]). The above discussion is detailed in section 3.

However, there exist classes of concepts satisfying the following property: there exists a (polynomial) collection of sets $(E_i)_{i\in I}$ such that for every concept f and g and for every distribution μ, if for every index i, $\mu_f(E_i) \simeq \mu_g(E_i)$ then $\mu(f\Delta g) \simeq 0$. This property does not mean that relative frequencies measured on the positive examples suffice to determine the target, but that the target is determined **relatively to the underlying distribution**. In other words, extra-information about the underlying distribution suffice to make the learning possible. These considerations lead us to define a PAC model of learning from positive examples where information about the distribution are given by unlabeled examples. Note that there are many situations in which it is natural to suppose that the learner is given positive and unlabeled data: for example, in marketing analysis context, if we want to know which customers are liable to ask for some specific service, we have at our disposal a population of customers who have already asked for these services (positive data) and the global population (unlabeled data). In medical context, a physician knows the patients who have developed a given disease (positive data) among his whole practice (unlabeled data).

A similar approach was taken in [BDL97] where a model of concept learning from unlabeled examples only is defined: the information about the target concept come through a dependence of the generating distribution upon this target.

We also define a model of learning from positive statistical queries where information about the distribution are given by unlabeled queries. Relations with PAC model ([Val84]), statistical query model ([Kea93]) and constant-partition classification noise model ([Dec97]) are studied in section 4. We show in section 5 that the classes of k-DNF and k-decision lists are learnable from positive statistical queries, i.e. with far less information than what is supposed in previously known algorithms ([Val84], [Kea93], [Riv87]).

A lot of work have been done on learning from positive examples only in Gold's model of learning in the limit ([Gol67], [Ang80], [Ber86], [Shi90], [ZL95]). The problems encountered in Gold framework, as over-generalizations, are clearly related to the questions studied here. But a systematic comparison between the two frameworks is out of the scope of this paper.

2 Preliminaries

Let $\mathcal{B}_n$ be the set of boolean functions from $X_n = \{0,1\}^n$ into $\{0,1\}$. Let $X = \cup_{n\geq 1} X_n$ and $\mathcal{B} = \cup_{n\geq 1}\mathcal{B}_n$. A *concept class* $\mathcal{C}$ over X is a subset of $\mathcal{B}$. We note $\mathcal{C}_n = \mathcal{C} \cap \mathcal{B}_n$.

A *representation scheme* for a concept class $\mathcal{C}$ is a function $R : \mathcal{C} \to 2^{\Sigma^*}$ where Σ is a finite alphabet and such that for each f and f' in $\mathcal{F}$, $R(f)$ is not empty and if $f \neq f'$, $R(f) \cap R(f') = \emptyset$. The *size* of a concept f is $size(f) = min\{|c||c \in R(f)\}$. 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.

An *example* of a concept f is a pair $(x, f(x))$, where x is in the domain of f. An example $(x, f(x))$ is *positive* if $f(x) = 1$ and *negative* otherwise. We denote by $pos(f)$ (resp. $neg(f)$) the set of all x such that $f(x) = 1$ (resp . $f(x) = 0$). If μ is a probability distribution on X_n and if f is a boolean function defined on X_n, $\mu(f)$ denotes $\mu(pos(f))$. If $\mu(f) \neq 0$, let μ_f be the restriction of μ to $pos(f)$ defined as follows: $\mu_f(x) = \mu(x)/\mu(f)$ if $x \in pos(f)$ and 0 otherwise.

A *statistical query* over X_n is a mapping $\chi : X_n \times \{0,1\} \to \{0,1\}$. If $f \in \mathcal{B}_n$, the query χ_f denotes the mapping defined by $\chi_f(x, y) = 1$ iff $y = f(x)$.

Definition 1. *Let $\mathcal{C}$ be a concept class over X. Let $f \in \mathcal{C}_n$ and μ be a distribution over X_n.*

- *The oracle $EX(f, \mu)$ is a procedure that returns at each call an example $(x, f(x))$ drawn randomly according to μ.*
- *The oracle $UNL(\mu)$ is a procedure that returns at each call an unlabeled example x drawn randomly according to μ.*
- *The oracle $STAT(f, \mu)$ is a procedure that, for every statistical query χ and every $\tau \in (0, 1]$, with input (χ, τ) returns an approximation of $\mu(\{x|\chi(x, f(x)) = 1\})$ with an accuracy at least τ.*
- *The noisy oracle $EX^{\eta_+,\eta_-}(f, \mu)$ is a procedure which at each call draws an element x of X_n according to μ and returns (i) $(x, 1)$ with probability $1 - \eta_+$ and $(x, 0)$ with probability η_+ if $x \in pos(f)$, (ii) $(x, 0)$ with probability $1 - \eta_-$ and $(x, 1)$ with probability η_- if $x \in neg(f)$*

All these oracles run in unit time.

A *k-monomial* on the variables $x_1, \ldots, x_n$ is a conjunction of exactly k literals. When there is no ambiguity on the set of variables, we note k-MON the set of all k-monomials and for every boolean function f, we note $M_k(f)$ the set of all k-monomials m such that $m(x) = 1 \Rightarrow f(x) = 1$. The number of k-monomials over n variables is at most $(2n)^k$. A k-DNF is a disjunction of k-monomials. A k-decision list (k-DL) is an ordered sequence $f = (m_1, b_1), \ldots, (m_l, b_l)$ in which each m_i is a k-monomial, each $b_i \in \{0,1\}$ and $m_l = 1$. If $u \in X_n$, the value $f(u)$ is defined to be b_j, where j is the smallest index satisfying $m_j(u) = 1$. We choose representation schemes such that the size of a k-DNF or a k-DL over n variables is bounded by a polynomial in n. We note $\mathbf{1}$ the boolean function such that $\mathbf{1}(u) = 1$ for every u.

We take the two basic following definitions in [KV94].

Definition 2. *Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is **PAC learnable** if there exist a learning algorithm L and a polynomial $p(.,.,.,.)$ with the following property: for any $f \in \mathcal{C}$, for any distribution μ on X, and for any $0 < \epsilon < 1$ and $0 < \delta < 1$, if L is given access to $EX(f, \mu)$ and to inputs ϵ and δ, then with probability at least $1 - \delta$, L outputs a hypothesis concept $h \in \mathcal{C}$ satisfying $\mu(f \Delta h) \leq \epsilon$ in time bounded by $p(1/\epsilon, 1/\delta, size(f), n)$.*

Definition 3. *Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is* ***learnable from statistical queries*** *if there exist a learning algorithm L and polynomials $p(.,.,.), q(.,.,.)$ and $r(.,.,.)$ with the following property: for any $f \in \mathcal{C}$, for any distribution μ over X, and for any $0 < \epsilon < 1$, if L is given access to $STAT(f,\mu)$ and to input ϵ, then*

- *For every query (χ, τ) made by L, the predicate χ can be evaluated in time $q(1/\epsilon, n, size(f))$, and $1/\tau$ is bounded by $r(1/\epsilon, n, size(f))$.*
- *L halts in time bounded by $p(1/\epsilon, n, size(f))$.*
- *L outputs a hypothesis $h \in \mathcal{C}$ that satisfies $\mu(f \Delta h) \leq \epsilon$.*

The standard classification noise model is defined in [AL88]. It is generalized by the constant-partition classification noise (CPCN) model defined in [Dec97]. We give below a restricted variant of the CPCN model.

Definition 4. *Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is* ***CPCN learnable*** *if there exist a learning algorithm L and a polynomial $p(.,.,.,.,.)$ with the following property: for any $f \in \mathcal{C}$, for any distribution μ on X, and for any $0 \leq \eta_+, \eta_- < 1/2$ and $0 < \epsilon, \delta < 1$, if L is given access to $EX^{\eta_+,\eta_-}(f,\mu)$ and to inputs ϵ and δ, then with probability at least $1-\delta$, L outputs a hypothesis concept $h \in \mathcal{C}$ satisfying $\mu(f \Delta h) \leq \epsilon$ in time bounded by $p(1/\epsilon, 1/\delta, 1/\gamma, size(f), n)$ where $\gamma = min\{1/2 - \eta_+, 1/2 - \eta_-\}$.*

3 Is it possible to learn with positive examples only?

Let f be a target over X_n, let μ be the underlying distribution (such that $\mu(f) \neq 0$) and suppose that the only oracle available to the learner is $EX(f,\mu_f)$. Before saying whether he is able to learn, we have to define how the error will be evaluated.

The first idea could be to measure the error of a hypothesis h on the positive examples only. But if we do so, over-generalization will be unavoidable: $\mathbf{1}$ is a correct answer whatever the target is.

Then, it seems necessary to take negative examples into account. But if the error is measured in the standard way, taking $error(h) = \mu(f \Delta h)$, another problem appears: two distributions μ and μ' can have the same restriction on the positive examples of the target while they are very different on the negative examples. More precisely, let $x_0 \in X_n \backslash f$ and let $\alpha \in [0,1)$ such that $|\alpha - \mu(x_0)| \geq 1/2$. Define

$$\mu'(x) = \begin{cases} \alpha & \text{if } x = x_0 \\ \frac{\mu(x)}{\mu(f)}(1-\alpha) & \text{if } x \in f \\ 0 & otherwise \end{cases} \quad (1)$$

We have $\mu_f = \mu'_f$ and $|\mu(x) - \mu'(x_0)| \geq 1/2$. Therefore, as it is impossible to differentiate μ and μ' with the help of the oracle $EX(f,\mu_f)$, x_0 must not belong

to the output hypothesis. That is, learning from positive examples requires the output hypothesis to be included into the target concept. But only for very constrained classes, as k-CNF or lattices, it will be possible to ensure that the output concept is included into the target concept. See ([Nat87], [Shv90]) for characterizations of such classes.

A related problem come from the following fact: as it is impossible to differentiate a negative example from a positive one on which the distribution is equal to 0, the learner cannot use missing examples to infer negative information.

Example 1. Consider the class of 1-DNF on two variables x_1 and x_2. Let $f = x_1$, $g = x_2$, μ and μ' such that $\mu(11) = \mu(01) = 1/2$ and $\mu'(11) = \mu'(10) = 1/2$. Whatever the pair (*target, distribution*) is among (f, μ) and (g, μ'), the sample will be $S = \{11\}$. Is 01 a negative example or a positive example on which the distribution is null? What must be learned?

Now, in order to make the learning possible, we could demand that each used distribution points out only one target concept. That is, we could demand the target to be the minimal concept consistent with a sufficiently large sample.

For example, if the target is $x_1 + x_2$, we should have $\mu(01), \mu(10)$ and $\mu(11)$ not too small. But, in addition to the fact that this restriction seems artificial, the simplest classes of concepts remain not learnable. We have shown (see [Den98]) that the problem of finding a minimal 1-DNF consistent with a *positive* sample is not polynomial (under the assumption $P \neq LOGSNP$).

So, isn't anything possible ? To our knowledge, the analysis of PAC learning from positive examples only usually stopped here. And yet, it is possible to go further. The following result shows that, with regard to k-DNF, the possible outputs are somehow determined by positive data.

Proposition 1. *For every $\epsilon \in [0, 1]$, for every integer n, for every k-DNF f and g over X_n and for every distribution μ over X_n such that $\mu(f) \neq 0$ and $\mu(g) \neq 0$, if for every k-monomial m,*

$$|\mu_f(m) - \mu_g(m)| < \alpha = \frac{\epsilon}{N(N+1)}$$

where $N = (2n)^k$, then

$$\mu(f \Delta g) < \epsilon$$

Proof. Let $m \in M_k(f)$ such that $\mu(m) \geq \mu(f)/N$. Such a monomial exists since the number of k-monomials is bounded by N.

We have

$$\mu_f(m) = \mu(m)/\mu(f) \leq \mu_g(m) + \alpha = \mu(m \cap g)/\mu(g) + \alpha \leq \mu(m)/\mu(g) + \alpha$$

that is,

$$\mu(g) \leq \mu(f) + \alpha\mu(f)\mu(g)/\mu(m) \leq \mu(f) + \alpha N$$

Symmetrically, we can get

$$\mu(f) \leq \mu(g) + \alpha N$$

and therefore

$$|\mu(g) - \mu(f)| \leq \alpha N$$

Now we have

$$\begin{aligned}
\mu(f \setminus g) &\leq \sum_{m \in M_k(f) \setminus M_k(g)} \mu(m \setminus g) = \sum_{m \in M_k(f) \setminus M_k(g)} [\mu(m \cap f) - \mu(m \cap g)] \\
&= \sum_{m \in M_k(f) \setminus M_k(g)} [\mu(f)\mu_f(m) - \mu(g)\mu_g(m)] \\
&\leq \sum_{m \in M_k(f) \setminus M_k(g)} [\mu(f)|\mu_f(m) - \mu_g(m)| + \mu_g(m)|\mu(f) - \mu(g)|] \\
&\leq \sum_{m \in M_k(f) \setminus M_k(g)} [|\mu_f(m) - \mu_g(m)| + |\mu(g) - \mu(f)|]
\end{aligned}$$

Getting a similar bound for $\mu(g \setminus f)$ we get

$$\mu(f \Delta g) \leq \sum_{m \in M_k(f) \Delta M_k(g)} [|\mu_f(m) - \mu_g(m)| + |\mu(g) - \mu(f)|] \leq N\alpha(N+1) = \epsilon$$

□

This result may seem quite paradoxical. Example 1 shows that it is impossible to differentiate $f = x_1$ from $g = x_2$ if the only available data is 11 and the previous proposition says that the target is determined by the frequencies on positive data. In fact, what is determined is not the target but **the target when the underlying distribution is known**. On the previous example, proposition 1 says that if the distribution is μ then the correct hypothesis must be f while if the distribution is μ', it must be g.

We think that this is the best we can expect from positive examples in the PAC framework: a learning algorithm has to return an approximation of the target concept as soon as extra-information about the underlying distribution are given.

4 Learning from positive examples

In the following definitions, the "positive" information about the target will be given by the oracles $EX(f, \mu_f)$ or $STAT(f, \mu_f)$ while the extra-information about the distribution will be given by $UNL(\mu)$ or $STAT(1, \mu)$.

Definition 5. *Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is **PAC learnable from positive examples** if there exist a learning algorithm L and a polynomial $p(.,.,.,.)$ with the following property: for any integer n, for any $f \in \mathcal{C}_n$, for any distribution μ on X_n, and for any $0 < \epsilon < 1$ and $0 < \delta < 1$, if L is given access to $EX(f, \mu_f), UNL(\mu)$ and to inputs ϵ and δ, then with probability at least $1 - \delta$, L outputs a hypothesis concept $h \in \mathcal{C}_n$ satisfying $\mu(f \Delta h) \leq \epsilon$ in time bounded by $p(1/\epsilon, 1/\delta, size(f), n)$.*

Remark that if a concept class $\mathcal{C}$ is PAC learnable and if there exists a learning algorithm for $\mathcal{C}$ which does not use negative examples of the target, then $\mathcal{C}$ is PAC learnable from positive examples. Therefore, k-CNF ([Val84]) and integer lattices ([HSW92]) are learnable from positive examples.

A similar approach has been taken in [BDL97]. A model of unsupervised learning is defined in which the task of the learner is to identify a probability distribution or more precisely, its high probability-density areas, from unlabeled examples. Then, a learning Without A Teacher model is proposed, in which it is assumed that "for points outside the target the distribution density is lower that a certain threshold α, while inside the target the density exceeds some value $\beta > \alpha$". A characterization of learnability is given, from an information-theoretic point of view; but the computational complexity of learning inside specific hypothesis spaces is not studied.

Definition 6. *Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$* ***is learnable from positive statistical queries*** *if there exist a learning algorithm L and polynomials $p(.,.,.), q(.,.,.)$ and $r(.,.,.)$ with the following property: for any integer n, for any $f \in \mathcal{C}_n$, for any distribution μ over X_n, and for any $0 < \epsilon < 1$, if L is given access to $STAT(f, \mu_f)$ and $STAT(1, \mu)$ and to input ϵ, then*

- *For every query (χ, τ) made by L, the predicate χ can be evaluated in time $q(1/\epsilon, n, size(f))$, and $1/\tau$ is bounded by $r(1/\epsilon, n, size(f))$.*
- *L will halt in time bounded by $p(1/\epsilon, n, size(f))$.*
- *L will output a hypothesis $h \in \mathcal{C}_n$ that satisfies $\mu(f \Delta h) \leq \epsilon$.*

Proposition 2. *Let us note $POSQ$ (resp. Q, $CPCN$, $POSEX$, PAC) the set of classes learnable with* positive statistical queries *(resp.* statistical queries, constant partition classification noise, positive examples, positive and negative examples*). Following relations hold:*

$$POSQ \subseteq Q \subseteq CPCN \subseteq POSEX \subseteq PAC$$

Proof. (sketch)

$POSQ \subseteq Q$**:** Note first that if χ_1 is the query defined by $\chi(x, y) = 1$ iff $y = 1$, we have $\mu(\{x | \chi_1(x, f(x)) = 1\} = \mu(f)$.

Let χ be a query and define

- $\chi^1(x, y) = \chi(x, 1)$. We have $\mu(\{x | \chi^1(x, f(x)) = 1\} = \mu(\{x | \chi(x, 1) = 1\}$.
- $\chi'(x, y) = 1$ iff $(\chi(x, 1) = 1$ and $y = 1)$. We have $\mu(\{x | \chi'(x, f(x)) = 1\}) = \mu_f(\{x | \chi(x, f(x)) = 1\})\mu(f)$.

Let L be a learning algorithm from positive statistical positive queries and let L' be the following algorithm:

Learning from statistical queries
Input: ϵ, n
Begin
{Compute a first approximation of $\mu(f)$}

$\hat{\mu}_f = STAT(f, \mu, \chi_1, \epsilon/4)$
If $\hat{\mu}_f \leq 3\epsilon/4$ **then** $h = \emptyset$
else {we know that $\mu(f) \geq \epsilon/2$}
run algorithm $L(\epsilon, n)$
at each call of $STAT(1, \mu, \chi, \tau)$, form the query χ^1
and return $STAT(f, \mu, \chi^1, \tau)$ to L
at each call of $STAT(f, \mu_f, \chi, \tau)$, let $\tau' = \frac{\tau\epsilon^2}{16}$
form the query χ' and return $\frac{STAT(f,\mu,\chi',\tau')}{STAT(f,\mu,\chi_1,\tau')}$ to L
let h be the hypothesis output by L
End
Output: h

We have $\mu_f(\{x|\chi(x, f(x)) = 1\} = \frac{\mu(\{x|\chi'(x,f(x))=1\}}{\mu(f)}$.
If $|a - \hat{a}| \leq \tau', |b - \hat{b}| \leq \tau', b \geq \epsilon/2, \hat{b} \geq \epsilon/4$, we have $|a/b - \hat{a}/\hat{b}| \leq \tau$.
This proves that $\frac{STAT(f,\mu,\chi',\tau')}{STAT(f,\mu,\chi_1,\tau')}$ is an approximation of $\mu_f(\{x|\chi(x, f(x)) = 1\}$ with an accuracy at least τ.
It is now easy to verify that L' is a learning algorithm from statistical queries.

$Q \subseteq CPCN$**:** This result is proved in [Dec97].

$CPCN \subseteq POSEX$**:** (with the help of an anonymous referee). Let $\mathcal{C}$ be a concept class in CPCN, f be a concept of $\mathcal{C}_n$, μ be a distribution over X_n such that $\mu(f) \neq 0$, $0 < \epsilon < 1$ and $0 < \delta < 1$. Let ν be the distribution defined by:

$$\nu(x) = \begin{cases} 2\mu_f(x)/3 + \mu(x)/3 & \text{if } x \in pos(f) \\ \mu(x)/3 & otherwise. \end{cases}$$

We can easily verify that the noisy oracle $EX^{\eta_+,\eta_-}(f, \nu)$ with $\eta_- = 0$ and $\eta_+ = \frac{\mu(f)}{2+\mu(f)}$ can be simulated this way: with probability $2/3$, get an example from $EX(f, \mu_f)$ and label it $+$, and with probability $1/3$, get an example from $UNL(\mu)$ and label it -. A negative example of f is always labelled -: a positive example of f is labelled - with probability $\mu(f)/3$.
Note that $1/2 - \eta_+ \geq 1/6$ and that for every subset A of X_n, $\nu(A) \geq \mu(A)/3$. Therefore, in order to learn $\mathcal{C}$ from positive examples with accuracy parameter ϵ, run the CPCN algorithm with accuracy parameter $\epsilon/3$ and at each call of $EX^{\eta_+,\eta_-}(f, \mu)$, call $EX(f, \mu_f)$ with probability $2/3$ and $UNL(\mu)$ with probability $1/3$ and return the result according to the labelling defined above.

$POSEX \subseteq PAC$**:** the oracles $UNL(\mu)$ and $EX(f, \mu_f)$ can easily be simulated using the oracle $EX(f, \mu)$.

Remark that the class of parity functions can be learned in PAC model using positive examples uniquely ([HSW92], [Kea93]). It is proved in [Kea93] that it is not learnable with statistical queries. Therefore, the class of parity functions is in $POSEX$ but not in Q.

We can't prove that $POSEX$ (resp. $POSQ$) is strictly included into PAC (resp. Q). We conjecture that the class composed of complementary sets of lattices is not learnable from positive examples (while it is PAC learnable). □

As a corollary, the previous proposition proves that k-DNF and k-DL are learnable from positive examples since they are learnable from statistical queries [Kea93].

Moreover, if the learner knows the underlying distribution and can simulate it within polynomial time, he can learn any class in Q from positive examples only. For example,

Corollary 1. *The classes of k-DNF and k-DL are learnable from $EX(f, u_f)$ only under the uniform distribution u.*

Proof. The oracle $EX(1, u)$ can be simulated by tossing a coin. □

A concept class learnable from statistical queries can be not learnable from positive statistical queries with the same space of queries. For example, let $\mathcal{C} = \{f, g\} \subset 2^{\{a,b\}}$ where $f = \{a, b\}$ and $g = \{a\}$ and let $\chi(x, y) = 1$ if $y = 1$ and $\chi(x, y) = 0$ otherwise. We have $STAT(f, \mu)(\chi, \tau) \simeq 1$ and $STAT(g, \mu)(\chi, \tau) \simeq \mu(a)$ while $STAT(f, \mu_f)(\chi, \tau) \simeq 1$ and $STAT(g, \mu_g)(\chi, \tau) \simeq 1$. Therefore, C is learnable using statistical query χ but it is not learnable using positive (restriction of) statistical query χ.

We prove in the next section that k-DNF and k-DL remains learnable from positive statistical queries.

5 Learning from positive statistical queries

Definition 7. *Let $\mathcal{C}$ be a concept class over X. We say that the weight of concepts of $\mathcal{C}$ can be estimated from positive statistical queries if there exist an algorithm W and a polynomial $p(.,.,.)$ with the following property: for any integer n, for any $f \in \mathcal{C}_n$, for any distribution μ over X_n, and for any $0 < \epsilon < 1$, if W is given access to statistical queries oracles $STAT(f, \mu_f)$ and $STAT(1, \mu)$ and to input ϵ, then W outputs a number $\hat{\mu}(f)$ such that $|\hat{\mu}(f) - \mu(f)| \leq \epsilon$ and W halts in time bounded by $p(1/\epsilon, n, size(f))$.*

Theorem 1. *Let $\mathcal{C}$ be a concept class over X learnable from statistical queries. If the weight of concepts of $\mathcal{C}$ can be estimated from positive statistical queries then $\mathcal{C}$ is learnable from positive statistical queries.*

Proof. Let L be the learning algorithm from statistical queries and let W be the algorithm which evaluates the weight of concepts of $\mathcal{C}$. The following algorithm learns $\mathcal{C}$ from positive statistical queries.

Learning algorithm L'
Input: ϵ, n

Begin

Run algorithm L

Each time algorithm L asks the oracle $STAT(f, \mu)$

in order to evaluate the query (χ, τ)

Run $W(\tau/4)$ and let $\hat{\mu}(f)$ be the result

Let χ^0 be the query defined by $\chi^0(x, y) = \chi(x, 0)$

Let χ^1 be the query defined by $\chi^1(x, y) = \chi(x, 1)$

Let $\hat{\mu}_{\chi^0} = STAT(1, \mu, \chi^0, \tau/4)$

Let $\hat{\mu}^+_{\chi^0} = STAT(f, \mu_f, \chi^0, \tau/4)$

Let $\hat{\mu}^+_{\chi^1} = STAT(f, \mu_f, \chi^1, \tau/4)$

Return $\hat{\mu}_{\chi^0} + (\hat{\mu}^+_{\chi^1} - \hat{\mu}^+_{\chi^0})\hat{\mu}(f)$ to algorithm L

End

Output: the output of algorithm L

It is easy to verify that

$$\begin{aligned}
&\mu(\{x|\chi(x, f(x)) = 1\}) \\
&= \mu(\{x|\chi(x, 1) = 1 \wedge f(x) = 1\}) + \mu(\{x|\chi(x, 0) = 1 \wedge f(x) = 0\}) \\
&= \mu_f(\{x|\chi(x, 1) = 1\})\mu(f) + (\mu(\{x|\chi(x, 0) = 1\}) - \mu(\{x|\chi(x, 0) = 1 \wedge f(x) = 1\})) \\
&= \mu(\{x|\chi(x, 0) = 1\}) + (\mu_f(\{x|\chi(x, 1) = 1\}) - \mu_f(\{x|\chi(x, 0) = 1\}))\mu(f)
\end{aligned}$$

The proposition follows. □

We now apply this result to k-DNF and k-DL.

Proposition 3. *The class of k-DNF formulas is learnable from positive statistical queries.*

Proof. Let f be a k-DNF over n variables and m be a k-monomial over X_n. Let μ be a distribution over X_n such that $\mu(f) \neq 0$. We have

$$\mu_f(m) = \mu(f \cap m)/\mu(f) \leq \mu(m)/\mu(f)$$

i.e. for every $m \in k$-MON such that $\mu_f(m) \neq 0$,

$$\mu(f) = \mu(f \cap m)/\mu_f(m) \leq \mu(m)/\mu_f(m)$$

and if m is in $M_k(f)$, i.e. if $m \Rightarrow f$,

$$\mu(f) = \mu(m)/\mu_f(m)$$

Therefore, we get

$$\mu(f) = min\{\frac{\mu(m)}{\mu_f(m)}|m \in k\text{-MON}, \mu_f(m) \neq 0\}$$

and since there exists a monomial m in $M_k(f)$ such that $\mu_f(m) \geq 1/N$ (where $N = (2n)^k$), we have

$$\mu(f) = min\{\frac{\mu(m)}{\mu_f(m)} | m \in k\text{-MON}, \mu_f(m) \geq 1/N\}$$

The following algorithm computes an estimation of $\mu(f)$.

Learning the weight of a k-DNF
Input: ϵ, n
Begin
Let $\tau = \frac{\epsilon}{(8N^2)}$
For all k-monomial m
compute $\hat{\mu}_f(m) = STAT(f, \mu_f, \chi_m, \tau)$
$\{\chi_m(x, y) = 1$ iff $y = m(x)\}$
compute $\hat{\mu}(m) = STAT(1, \mu, \chi_m, \tau)$
EndFor
Let $\hat{\mu}(f) = min\{\hat{\mu}(m)/\hat{\mu}_f(m) | m \in k\text{-MON}, \hat{\mu}_f(m) \geq 1/N - \tau\}$
End
Output: $\hat{\mu}(f)$

We have $\mu(f) = min\{\mu(m)/\mu_f(m) | m \in k\text{-MON}, \hat{\mu}_f(m) \geq 1/N - \tau\}$.
Verify that if $\hat{\mu}_f(m) \geq 1/N - \tau$,

$$|\hat{\mu}(m)/\hat{\mu}_f(m) - \mu(m)/\mu_f(m)| \leq 2\tau/(\hat{\mu}_f(m)\mu_f(m)) \leq 2\tau/[(1/N - \tau)(1/N - 2\tau)]$$

and since $1/N - \tau \geq 1/N - 2\tau \geq 1/(2N)$ we have

$$|\hat{\mu}(f) - \mu(f)| \leq 2\tau 4N^2 = \epsilon$$

We can now apply theorem 1. □

We now prove an analogous result for k-decision lists. The proof is trickier in this case.

Theorem 2. *The class of k-decision lists is learnable from positive statistical queries.*

As in previous proposition, we just have to prove that the weight of k-decision lists can be estimated from positive statistical queries.

Let f be a k-DL over n variables and let μ be a distribution over X_n such that $\mu(f) \neq 0$.

Let

$$M_f^\mu = \{x \in X_n | \forall m \in k\text{-MON}, m(x) \Rightarrow \mu_f(m) \neq 0\}$$

Let $\overline{M_f^\mu}$ be the complementary set of M_f^μ. We have

$$\overline{M_f^\mu} = \bigcup \{m \in k\text{-MON} | \mu_f(m) = 0\}$$

We show below some properties of M_f^μ.

Lemma 1. *1.* $\mu(f \setminus M_f^\mu) = 0$
2. for every subset A *of* X_n, $\mu_f(A) \leq \mu(A \cap M_f^\mu)/\mu(f)$.
3. if $(m, 1)$ *is the first (positive) term of* f *such that* $\mu_f(m) \neq 0$, *then* $\mu_f(m) = \mu(m \cap M_f^\mu)/\mu(f)$
4. $\mu(f) = min\{\mu(m \cap M_f^\mu)/\mu_f(m) | m \in k\text{-}MON, \mu_f(m) \neq 0\}$

Proof. 1. let $x \in f \setminus M_f^\mu$, and let $m \in k$-MON such that $m(x) = 1$ and $\mu_f(m) = 0$. As $x \in f$, we have $\mu(x) = 0$.
2. $\mu_f(A) = \mu(A \cap f)/\mu(f) \leq [\mu(A \cap (f \setminus M_f^\mu)) + \mu(A \cap M_f^\mu)]/\mu(f) \leq \mu(A \cap M_f^\mu)/\mu(f)$.
3. let $x \in m \cap M_f^\mu$ such that $\mu(x) \neq 0$. For every term (m', b) preceding $(m, 1)$ in f, $\mu_f(m') = 0$ and since $x \in M_f^\mu$, we have $m'(x) = 0$. Therefore $x \in f$ and $\mu(m \cap M_f^\mu) = \mu(m \cap f)$.
4. apply the two previous points.

□

The last relation is much less robust than the analogous one for k-DNF. This is because $\mu_f(m) = \mu(m \cap M_f^\mu)/\mu(f)$ can be true for only one monomial m, and moreover, the weight of m under μ can be very small. In the following learning algorithm, we build a distribution ν, close to μ and such that the first positive term $(m, 1)$ of f such that $\nu_f(m) \neq 0$ has not too small a weight under ν.

Learning the weight of a k-decision list (WDL)
Input: ϵ, n
Begin
Let $N = (2n)^k$, $\alpha = \frac{\epsilon}{25N}$, $\tau_1 = \alpha/4$, $\tau_2 = (\epsilon\alpha^2)/64$
{We now build a set M such that for every k-monomial m, $\mu_f(m \cap M)$ is null or not too small}
{$\overline{M}$ is the complementary set of M}
$\overline{M} = \emptyset$
$MON_\alpha = k$-MON
Loop
For all k-monomials $m \in MON_\alpha$
ask $\hat{\mu}_f(m \cap M) = STAT(f, \mu_f, \chi_{m \cap M}, \tau_1)$
EndFor
If $\forall m \in MON_\alpha, \hat{\mu}_f(m \cap M) \geq \alpha$ **then**
ExitLoop
EndIf
$AUX \leftarrow \{m \in MON_\alpha | \hat{\mu}_f(m \cap M) < \alpha\}$

$\overline{M} \leftarrow \overline{M} \cup \bigcup\{m \in AUX\}$
$MON_\alpha \leftarrow MON_\alpha \setminus AUX$
{Note that $\overline{M}$ is a k-DNF and that
the queries $\chi_{m\cap M}$ can be evaluated in polynomial time}
EndLoop
For all k-monomials m in MON_α **do**
ask $\hat{\mu}(m \cap M) = STAT(1, \mu, \chi_{m\cap M}, \tau_2)$
ask $\hat{\mu}_f(m \cap M) = STAT(f, \mu_f, \chi_{m\cap M}, \tau_2)$
EndFor
compute $\hat{\mu}(f) = min\{\frac{\hat{\mu}(m\cap M)}{\hat{\mu}_f(m\cap M)} | m \in MON_\alpha\}$
End
Output: $\hat{\mu}(f)$

Lemma 2. *The previous algorithm runs in polynomial time and outputs* $\hat{\mu}(f)$ *such that* $|\hat{\mu}(f) - \mu(f)| \leq \epsilon$.

The proof, a bit technical, relies on several lemmas.
Suppose in all the following that we have run the algorithm WDL.

Lemma 3. $\mu(\overline{M} \cap f) \leq N(\alpha + \tau_1)\mu(f) < 1$.

Proof. Each time a monomial m is added to $\overline{M}$ in the previous algorithm, this is because $\hat{\mu}_f(M \cap m) < \alpha$ which implies $\mu_f(M \cap m) < \alpha + \tau_1$. The quantity added to $\overline{M} \cap f$ is $\mu(M \cap m \cap f) < (\alpha + \tau_1)\mu(f)$. And because the number of k-monomials is less than N, we get the result. □

Let ν be the distribution over X_n defined by :

$\nu(x) = 0$ if $x \in \overline{M} \cap f$ and $\nu(x) = \mu(x)/\mu(M \cup \overline{f})$ otherwise.

We prove some facts about ν which show that $\nu(f)$ is close to $\mu(f)$:

Lemma 4. *1. We have* $M = M_f^\nu$.
2. for every subset A *of* X_n, *we have* $|\nu(A) - \mu(A)| \leq 2N(\alpha + \tau_1)\mu(f)$.
3. we have $1 - 2N(\alpha + \tau_1) \leq \nu(f)/\mu(f) \leq 1 + 2N(\alpha + \tau_1)$.

Proof. 1. – Let $x \in \overline{M}$. There exists $m \in k$-MON such that $m(x) = 1$ and $m \subseteq \overline{M}$. Then, $\nu(m \cap f) = \nu_f(m) = 0$ and $x \in \overline{M_f^\nu}$.
– Let $x \in \overline{M_f^\nu}$. There exists $m \in k$-MON such that $m(x) = 1$ and $\nu_f(m) = \nu(m \cap f) = 0$. Then $\mu_f(m \cap M) = \mu(m \cap M \cap f)/\mu(f) = \mu(M \cup \overline{f})\nu(m \cap M \cap f)/\mu(f) = 0$. Therefore, m cannot be in MON_α since $\tau_1 < \alpha$. We have $x \in m \subseteq \overline{M}$.

2. we have

$$|\nu(A) - \mu(A)| \leq \sum_{x\in X_n} |\nu(x) - \mu(x)| \leq \sum_{x\in\overline{M}\cap f} |\nu(x) - \mu(x)| + \sum_{x\in M\cup\overline{f}} |\nu(x) - \mu(x)|$$

$$\leq \mu(\overline{M} \cap f) + \sum_{x\in M\cup\overline{f}} \mu(x)(1/\mu(M \cup \overline{f}) - 1)$$

$$\leq \mu(\overline{M} \cap f) + 1 - \mu(M \cup \overline{f}) \leq 2\mu(\overline{M} \cap f) \leq 2N(\alpha + \tau_1)\mu(f)$$

3. applying the last point, we get :

$$-2N(\alpha+\tau_1)\mu(f) \leq -\mu(f)+\nu(f) \leq 2N(\alpha+\tau_1)\mu(f)$$

that is

$$1-2N(\alpha+\tau_1) \leq \nu(f)/\mu(f) \leq 1+2N(\alpha+\tau_1)$$

□

We can now prove the lemma 2.

Proof. First note that the algorithm runs in polynomial time.
The only thing to prove is that $|\mu(f)-\hat{\mu}(f)| \leq \epsilon$.

- From lemma 4, we have $|\frac{\nu(f)}{\mu(f)}-1| \leq 2N(\alpha+\tau_1)$
- Let $m \in MON_\alpha$. We have

$$\begin{aligned}
|\frac{\mu(m\cap M)}{\mu_f(m\cap M)}-\frac{\hat{\mu}(m\cap M)}{\hat{\mu}_f(m\cap M)}| &\leq \frac{|\mu(m\cap M)\hat{\mu}_f(m\cap M)-\mu_f(m\cap M)\hat{\mu}(m\cap M)|}{\mu_f(m\cap M)\hat{\mu}_f(m\cap M)} \\
&\leq \frac{2\tau_2}{\mu_f(m\cap M)\hat{\mu}_f(m\cap M)} \\
&\leq \frac{2\tau_2}{(\alpha-\tau_1)(\alpha-\tau_1-\tau_2)} \leq \frac{8\tau_2}{\alpha^2}
\end{aligned}$$

since $\tau_1 < \alpha/4$ and $\tau_2 < \alpha/4$

- We also have

$$\begin{aligned}
\frac{\nu(m\cap M)}{\nu_f(m)} &= \frac{\mu(m\cap M)}{\mu(M\cup\overline{f})}\frac{\nu(f)}{\nu(m\cap f)} \\
&= \frac{\mu(m\cap M)}{\mu(M\cup\overline{f})}\frac{\mu(M\cup\overline{f})}{\mu(m\cap f\cap M)}\nu(f) = \frac{\mu(m\cap M)}{\mu(m\cap f\cap M)}\nu(f) \\
&= \frac{\mu(m\cap M)}{\mu_f(m\cap M)}\frac{\nu(f)}{\mu(f)}
\end{aligned}$$

- Using this relation, we get

$$\begin{aligned}
&|\frac{\nu(m\cap M)}{\nu_f(m)}-\frac{\hat{\mu}(m\cap M)}{\hat{\mu}_f(m\cap M)}| \\
&\leq \frac{\nu(f)}{\mu(f)}|\frac{\mu(m\cap M)}{\mu_f(m\cap M)}-\frac{\hat{\mu}(m\cap M)}{\hat{\mu}_f(m\cap M)}| + \frac{\hat{\mu}(m\cap M)}{\hat{\mu}_f(m\cap M)}|\frac{\nu(f)}{\mu(f)}-1| \\
&\leq 2|\frac{\mu(m\cap M)}{\mu_f(m\cap M)}-\frac{\hat{\mu}(m\cap M)}{\hat{\mu}_f(m\cap M)}| + 2|\frac{\nu(f)}{\mu(f)}-1| \\
&\leq \frac{16\tau_2}{\alpha^2}+4N(\alpha+\tau_1)
\end{aligned}$$

for every $m \in MON_\alpha$

– Now, let $m_0 \in MON_\alpha$ such that $\nu(f) = \frac{\nu(m_0 \cap M)}{\nu_f(m_0)}$ and $m_1 \in MON_\alpha$ such that $\hat{\mu}(f) = \frac{\hat{\mu}(m_1 \cap M)}{\hat{\mu}_f(m_1 \cap M)}$.

$$
\begin{aligned}
|\nu(f) - \hat{\mu}(f)| &= |\frac{\nu(m_0 \cap M)}{\nu_f(m_0)} - \frac{\hat{\mu}(m_1 \cap M)}{\hat{\mu}_f(m_1 \cap M)}| \\
&\leq 2Max\{|\frac{\nu(m \cap M)}{\nu_f(m)} - \frac{\hat{\mu}(m \cap M)}{\hat{\mu}_f(m \cap M)}||m \in MON_\alpha\} \\
&\leq \frac{32\tau_2}{\alpha^2} + 8N(\alpha + \tau_1)
\end{aligned}
$$

– To end the proof,

$$|\mu(f) - \hat{\mu}(f)| \leq |\mu(f) - \nu(f)| + |\nu(f) - \hat{\mu}(f)|$$

and since $|\mu(f) - \nu(f)| \leq 2N(\alpha + \tau_1)$ from lemma 4,

$$|\mu(f) - \hat{\mu}(f)| \leq \frac{32\tau_2}{\alpha^2} + 10N(\alpha + \tau_1) \leq \epsilon$$

□

As in corollary 1, if the learner knows the underlying distribution and can compute it within polynomial time, he can learn k-DNF and k-DL from positive queries only.

6 Conclusion

The models defined in this paper show that it is possible to describe learning from positive data in the PAC learning framework, as soon as information are given on the underlying distribution. Moreover, learning from positive and unlabeled data seems natural in many contexts. Lastly, these results show that many classes learnable in the PAC model are eventually learnable with much more severe constraints: positive and unlabeled queries provide far less information than positive and negative examples. In other words, classes which are learnable in the PAC framework are so not only because they meet the PAC model requirements but also others more restricting.

References

[AL88] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[Ang80] D. Angluin. Inductive inference of formal languages from positive data. *Inform. Control*, 45(2):117–135, May 1980.

[BDL97] Shai Ben-David and Michael Lindenbaum. Learning distributions by their density levels: A paradigm for learning without a teacher. *Journal of Computer and System Sciences*, 55(1):171–182, August 1997.

[Ber86] R. Berwick. Learning from positive-only examples. In *Machine Learning, Vol. II*, pages 625–645. Morgan Kaufmann, 1986.

[Dec97] S. E. Decatur. Pac learning with constant-partition classification noise and applications to decision tree induction. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

[Den98] F. Denis. Pac learning from positive statistical queries. Technical report, L.I.F.L., 1998. full version: `http:\\www.lifl.fr\~denis`.

[Gol67] E.M. Gold. Language identification in the limit. *Inform. Control*, 10:447–474, 1967.

[HSW92] D. Helmbold, R. Sloan, and M. K. Warmuth. Learning integer lattices. *SIAM J. COMPUT.*, 21(2):240–266, 1992.

[Kea93] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 392–401. ACM Press, New York, NY, 1993.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[Nat87] B. K. Natarajan. On learning boolean functions. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 296–304. ACM Press, 1987.

[Nat91] B. K. Natarajan. Probably approximate learning of sets and functions. *SIAM J. COMPUT.*, 20(2):328–351, 1991.

[Riv87] R.L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[Shi90] Takeshi Shinohara. Inductive inference from positive data is powerful. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 97–110, Rochester, New York, 6–8 August 1990. ACM Press.

[Shv90] Haim Shvayster. A necessary condition for learning from positive examples. *Machine Learning*, 5:101–113, 1990.

[Val84] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.

[ZL95] T. Zeugmann and S. Lange. A guided tour across the boudaries of learning recursive languages. In Lectures Notes in Artificial Intelligence, editor, *Algorithmic learning for knowledge-based systems*, volume 961, pages 190–258. 1995.