# Learning Sat- $k$-DNF formulas from membership queries 

F. Bergadano*, D. Catalano ${ }^{+}$and S. Varricchio ${ }^{\circ}$

*Università di Torino<br>+Università di Catania<br>${ }^{\circ}$ Università di L' Aquila


#### Abstract

In this paper we study the problem of learning Sat-k-DNF formulas from membership queries. We show that Sat-kDNF are PAC learnable with membership queries by proving that $k$-ambiguous automata are PAC learnable with membership queries and by establishing a PAC reduction that preserves membership queries between these two classes of concepts. We also give a positive answer in the direction of learning two way finite automata. We show that $k$-reversal bounded two-way automata (i.e. two-way automata that change head direction at most $k$ times) are PAC learnable with membership queries. As a corollary of Sat-1-DNF learnability one easily derives that decision trees are PAC learnable with membership queries. All these results are valid for every distribution of probability.


## 1 Introduction

Several techniques have been used in literature for PAC learning (or exactly learning) decision trees. Kushilevitz and Mansour [10] devised a technique for learning decision trees under the uniform distribution via the Fourier Spectrum. Schapire and Sellie gave in [14] a lattice based algorithm for learning multivariate polynomials under an arbitrary distribution. By this result, decision trees became PAC learnable in terms of multivariate polynomials. Bshouty and Mansour [5] developed a new approach: learning decision trees and multivariate polynomials via multivariate interpolation. This algebraic technique yields learning algorithms for decision trees and multivariate polynomials over fields under any constant bounded product distribution.

The problem, in the distribution-free model, was solved by Bshouty in [4], where the following was proved:

1. any boolean function is learnable with membership queries in time polynomial in its minimal DNF size, its minimal CNF size and the number of variables $n$,

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.
STOC'96, Philadelphia PA,USA

- 1996 ACM 0-89791-785-5/96/05..\$3.50

2. decision trees are learnable with membership queries.

In this paper, we give a positive answer to the open problem of learning Sat- $k$-DNF with membership queries, i.e. DNF formulas where, for every assignment, at most $k$ terms are satisfied. The learnability of decision trees with membership queries is obtained as a corollary. The results are obtained with techniques derived from previous work of Bergadano and Varricchio [2] on learning multiplicity automata. In this paper, we generalize the results of [2] on learning of unambiguous automata, proving that $k$-ambiguous automata are PAC-learnable from membership queries. Related results for two-way automata are also given. We then show that learning Sat- $k$-DNF is easily PAC-reducible to learning $k$-ambiguous automata. Our results are distribution-free. Under the uniform distribution, the whole class of DNF formulae was proved to be PAC-learnable [9]. Sat-k-DNF formulae were shown to be PAC-learnable for any constant bounded product distribution [5].

## 2 Preliminaries

The problem of learning automata from queries and examples has been extensively studied in the past. Bergadano and Varricchio [2] proved that the behavior of an unknown automaton with multiplicity in a field $K$ ( $K$-automaton) is exactly identifiable when multiplicity and equivalence queries are allowed. Therefore, $K$-automata are PAC-learnable from multiplicity queries under any distribution. A corollary of this result is that regular languages are PAC learnable using membership queries with respect to the representation of unambiguous non-deterministic automata. They introduce the notion of multiplicity query. In the case of a non deterministic automaton a multiplicity query returns the number of accepting paths for a given string. In the case of unambiguous non deterministic automata, multiplicity queries return either 0 or 1 , and then reduce to membership queries.

The general case is when the automaton is a multiplicity automaton. Automata with multiplicity (also called multiplicity automata) are the most important generalization of the automata theory. Let $M$ be a non deterministic finite automaton. We can consider the so called behavior of $M$ which is the map that associates to each string the number of its different accepting paths. More generally we can assign a number (multiplicity) to each initial state, to each final state and to each edge of the automaton. In this
context one can construct a very general theory in which classical and probabilistic automata are considered as particular cases. Now we recall some definitions and notations about the multiplicity automata theory. More details are in [3, 7, 13].

Let $K$ be a field and $A^{*}$ be the free monoid over the finite alphabet $A$, we consider the set $K\langle(A\rangle\rangle$ of all the applications

$$
S: A^{*} \rightarrow K
$$

An element $S$ of $K\langle\langle A\rangle\rangle$ is called a $K$-subset of $A^{*}$ or also a $K$-set. For any $S \in K\langle\langle A\rangle\rangle$ and $u \in A^{*}$ we will denote $S(u)$ as ( $S, u$ ). Let $K^{n \times n}$ be the set of all square $n \times n$ matrices with entries in $K$. Suppose that $K^{n \times n}$ is equipped with the row by column product. A map

$$
\mu: A^{*} \rightarrow K^{n \times n}
$$

is called a morphism if

$$
\mu(\epsilon)=I d
$$

(where $I d$ is the identity matrix), and for any $w=a_{1} \ldots a_{n}$, $a_{\mathrm{i}} \in A$,

$$
\mu(w)=\mu\left(a_{1}\right) \ldots \mu\left(a_{n}\right)
$$

A $K$-set $S$ is called recognizable (or also representable) if there exists a positive integer $n$, and $\lambda, \mu \in K^{n}$, and a morphism $\mu: A^{*} \rightarrow K^{n \times n}$ such that, for any $w \in A^{*}$

$$
(S, w)=\lambda \mu(w) \gamma
$$

where $\lambda$ and $\mu$ are to be considered a row-vector and a column-vector respectively. The triplet $(\lambda, \mu, \gamma)$ is called a linear representation of $S$ of dimension $n$. The linear representation is also called a $K$-automaton for $S$. As a matter of fact a $K$-automaton is a 5 -tuple

$$
M=(Q, A, E, I, F)
$$

where $A$ is a finite alphabet, $Q$ is a finite set of states, $E: Q \times A \times Q \rightarrow K$ is a map that associates to each edge a multiplicity, $I, F: Q \rightarrow K$ are maps that associate to each state the multiplicity as initial and final state respectively. To such an automaton one can associate a linear representation. In fact, let $Q=\{1, \ldots, n\}$ and let $\lambda, \gamma \in K^{n}$ be the characteristic vectors of $I$ and $F$, respectively. Let $\mu$ the morphism

$$
\mu: A^{*} \rightarrow K^{n \times n}
$$

defined by $\mu(a)_{\imath j}=E(i, a, j)$. The behavior of $M$ is the recognizable $K$-set $S_{M}$, defined by

$$
\left(S_{M}, w\right)=\lambda \mu(w) \gamma
$$

Any non-deterministic finite automaton $M$ can be represented as a $K$-automaton, since initial states, final states and edges of the automaton can be represented by their characteristic functions. In this case one can easily prove that for any $w \in A^{*},\left(S_{M}, w\right)$ is the number of different successful paths on the input $w$.

If $M$ is an unambiguous non-deterministic automaton ( i.e. any word has at most one successful path), then $S_{M}$ is the characteristic function of $L(M)$, the language accepted by $M$.

## 3 PAC reducibility

Pitt and Warmuth [12] introduced a notion of PAC reducibility: let $C$ and $C^{\prime}$ be two concept classes, if $C$ is PAC reducible to $C^{\prime}$ and if $C^{\prime}$ is PAC learnable then also $C$ is PAC learnable. Define $X_{n}$ as the instances of length at most $n$, and $C_{n}$ as the concepts having positive examples in $X_{n}$. In general, we say that the concept class $C$ over the domain $X$ reduces to the concept class $C^{\prime}$ over $X^{\prime}$ if the following two conditions are met:

1. (Efficient Instance Transformation) There should exist a map

$$
G: X \rightarrow X^{\prime}
$$

and a polynomial $p(\cdot)$ such that for any $n G\left(X_{n}\right) \subseteq$ $X_{p(n)}^{\prime}$ and G is polynomial time computable.
2. (Existence of Image Concept) There must exist a polynomial $q(\cdot)$ such that for every concept $c \in C_{n}$ there should exist a concept $c^{\prime} \in C_{p(n)}^{\prime}$ such that size $\left(c^{\prime}\right) \leq$ $q(s i z e(c))$, and, for all $x \in X_{n}, c(x)=1$ if and only if $c^{\prime}(G(x))=1$.
However, this scheme is not generalizable to the case where $C^{\prime}$ is PAC learnable with membership queries. The problem is that we use the same learning algorithm both for $C$ and for $C^{\prime}$. The reduction map then adapts to $C$ the work done for $C^{\prime}$. In this sense the presence of membership queries is a heavy impediment, because every time we need a query for $C$ we first need an answer to the query

$$
y \in C^{\prime}
$$

and then we must find an $x \in C$ such that

$$
G(x)=y
$$

where $G$ is the reduction map. Such an $x$ could not exists or it could not be computable in polynomial time. Then, if we want to extend the notion of PAC reducibility to the general case with membership queries, we must suppose the reduction map $G$ to be surjective and always equipped with a polynomial time computable counterimage.

## 4 k-ambiguous automata

We recall that $k$-ambiguous automata ( $k$ is a positive integer) are a generalization of unambiguous automata, in the sense that, for each word $w$, there are at most $k$ different accepting paths. We prove that the class of $k$ ambiguous automata is PAC learnable with membership queries.

First we recall an important operation over $K$-sets.
Definition 1 Let $S, T \in K\langle\langle A\rangle\rangle(K$ is a field).
The Hadamard product of $S$ and $T$ is denoted by the $K$-set $S \odot T$, defined by

$$
(S \odot T, w)=(S, w) \cdot(T, w)
$$

It is well known that recognizable $K$-sets are closed under Hadamard product [3]. We give a constructive proof of this fact since we need an upper bound to the dimension of a linear representation of the Hadamard product of two given recognizable $K$-sets.

We prove that

Lemma 1 The class of recognizable $K$-sets is closed under the Hadamard product.

Proof. Let $S$ and $T$ be two recognizable $K$-sets, and $A_{1}, A_{2}$ two $K$-automata associated to $S$ and $T$, respectively.

$$
\begin{gathered}
A_{1}=\left(Q^{\prime}, A, E^{\prime}, I^{\prime}, F^{\prime}\right) \\
A_{2}=\left(Q^{\prime \prime}, A, E^{\prime \prime}, I^{\prime \prime}, F^{\prime \prime}\right)
\end{gathered}
$$

Let $n$ and $m$ be the number of states of $A_{1}$ and $A_{2}$ respectively. We consider the automaton:

$$
M=(Q, A, E, I, F)
$$

where

$$
\begin{aligned}
& \text { - } Q=Q^{\prime} \times Q^{\prime \prime} \\
& \text { - } E\left(\left(q_{i}, p_{l}\right), a,\left(q_{3}, a, p_{k}\right)\right)=E^{\prime}\left(q_{i}, a, q_{j}\right) E^{\prime \prime}\left(p_{l}, a, p_{k}\right) \\
& \text { - } I\left(q_{1}, p_{\jmath}\right)=I^{\prime}\left(q_{\imath}\right) I^{\prime \prime}\left(p_{j}\right) \\
& \text { - } F\left(q_{i}, p_{j}\right)=F^{\prime}\left(q_{\imath}\right) F^{\prime \prime}\left(p_{j}\right) .
\end{aligned}
$$

This automaton has $n m$ states. Let $(\lambda, \mu, \gamma)$ be the linear representation corresponding to $M$. Clearly the dimension of $(\lambda, \mu, \gamma)$ is $n m$. Let $\left(\lambda^{\prime}, \mu^{\prime}, \gamma^{\prime}\right)$ and ( $\left.\lambda^{\prime \prime}, \mu^{\prime \prime}, \gamma^{\prime \prime}\right)$ be the linear representations corresponding to $A_{1}$ and $A_{2}$. Then by construction one has

$$
\lambda_{(2, j)}=\lambda_{i}^{\prime} \lambda_{j}^{\prime \prime}, \quad 1 \leq i \leq n \text { and } 1 \leq j \leq m .
$$

We use ( $i, j$ ) as an index for $\lambda$, but $\lambda$ is still an unidimensional vector. We use this notation only for convenience. Similarly one has

$$
\gamma_{(t, j)}=\gamma_{1}^{\prime} \gamma_{j}^{\prime \prime}, \quad 1 \leq i \leq n \text { and } 1 \leq j \leq m
$$

and, $\forall a \in A$, the $(n m) \times(n m)$ matrix $\mu$ is such that
$\mu(a)_{(r, j),(h, k)}=\mu^{\prime}(a)_{\imath, h} \mu^{\prime \prime}(a)_{j, k} \quad 1 \leq i, h \leq n \quad 1 \leq j, k \leq m$.
By an easy computation one proves that

$$
\begin{gathered}
\lambda \mu(w) \gamma=\left(\lambda^{\prime} \mu^{\prime}(w) \gamma^{\prime}\right)\left(\lambda^{\prime \prime} \mu^{\prime \prime}(w) \gamma^{\prime \prime}\right)= \\
=(S, w) \cdot(T, w)=(S \odot T, w) .
\end{gathered}
$$

Then $S \odot T$ is a recognizable $K$-set and has a linear representation of dimension $n m$.

Let $M$ be a $k$-ambiguous automaton and let $S$ be the behavior of $M$. We have that $S \in \mathcal{Z}\langle\langle A\rangle\rangle$ and

$$
\forall w \in A^{*} \quad 0 \leq(S, w) \leq k
$$

Let $L$ be the regular language accepted by $M$. It is clear that

$$
\forall w \in A^{*} \quad w \in L \Leftrightarrow 1 \leq(S, w) \leq k
$$

Also note that $S$ is recognizable. Let $(\lambda, \mu, \gamma)$ be the linear representation of $S$, corresponding to $M$, and $n$ its dimension. We observe that

$$
\lambda, \gamma \in \mathcal{Z}^{n}
$$

and

$$
\forall a \in A^{*} \quad \mu(a) \in \mathcal{Z}^{n \times n}
$$

Let $p$ be a prime number strictly greater than $k$. One can consider the set $\mathcal{Z}_{p}=\mathcal{Z} / \equiv_{p}$ and the canonical epimorphism $\phi: \mathcal{Z} \rightarrow \mathcal{Z}_{p}$, defined by $\phi(n)=[n]_{\Xi_{p}}$. Define $\bar{\lambda}, \bar{\mu}$ and $\bar{\gamma}$ as the projections under $\phi$ of $\lambda, \mu$ and $\gamma$. Let $\bar{S}$ be the $\mathcal{Z}_{p}$-set defined by

$$
(\bar{S}, w)=\phi((S, w))
$$

We observe that $\bar{S}$ is recognizable, since a linear representation is given by $(\bar{\lambda}, \bar{\mu}, \bar{\gamma})$. Note that $\mathcal{Z}_{p}$ is a field with respect to the sum and the product. Moreover, by Fermat's little theorem, we have

$$
\forall a \in \mathcal{Z}_{p}, a \neq 0, \quad a^{p-1}=1
$$

We can consider the $\mathcal{Z}_{p}$-set $T$ defined by

$$
(T, w)=\underbrace{(\bar{S}, w) \odot \ldots \odot(\bar{S}, w)}_{p-1 \text { ttmes }} \quad \forall w \in A^{*} .
$$

From the little Fermat theorem we have

$$
\begin{aligned}
& T(w)=0 \Leftrightarrow \bar{S}(w)=0 \Leftrightarrow S(w)=0 \\
& T(w)=1 \Leftrightarrow \bar{S}(w) \neq 0 \Leftrightarrow S(w) \geq 1 .
\end{aligned}
$$

By Lemma 1, $T$ is a recognizable $\mathcal{Z}_{p}$-set and has a linear representation of dimension $n^{p-1}$, if $n$ is the dimension of the linear representation of $S$. By the result of Bergadano and Varricchio we know that a $\mathcal{Z}_{p}$-set like $T$ is PAC learnable with multiplicity queries, in polynomial time with respect to the dimension of its linear representation.

However, since $T(w)$ is either 0 or 1 for every $w \in A^{*}$, every multiplicity query simply reduces to a membership query for $L(M)$. In fact

$$
\begin{aligned}
& T(w)=1 \Leftrightarrow w \in L(M) \\
& T(w)=0 \Leftrightarrow w \notin L(M) .
\end{aligned}
$$

Then $T$ is PAC learnable with membership queries for $L(M)$. We have established a characterization of $L$ by $T$ that allows us to learn $L$ in terms of $T$. Hence we proved the following:

Theorem 1 The class of $k$-ambiguous automata is PAC learnable with membership queries.

## 5 Two way finite automata

An interesting extension of classical automata is to allow the tape head the ability to move left as well as right. Such a finite automaton is called a two way finite automaton (2DFA). We prove that $k$-reversal-bounded 2DFA are PAC learnable with membership queries We recall that a two-way automaton is $k$-reversal-bounded if its tape head can change direction at most $k$ times: this implies that the tape head cannot visit each tape square more than $k+1$ times

An useful picture of a computation of a 2DFA consists of the input, the path of the tape head and the state in which the automaton is each time the boundary between two adjacent tape square is crossed. The list of states below each boundary between two consecutive squares is called a crossing sequence.

If the 2DFA accepts its input no crossing sequence can have a repeated state with the head moving in the same direction. Otherwise the 2DFA, being deterministic, could not reach the right end of the tape. Similarly, if the input is accepted no crossing sequence can have even length. A
crossing sequence is said to be valid if it has odd length and no two odd nor two even numbered states are identical. A 2DFA with $n$ states can have valid crossing sequences of length at most $2 n$. Two way finite automata are no more powerful than classical finite automata. One can prove [8] the following

Theorem 2 If $L$ is accepted by a 2DFA then $L$ is a regular set.

The general strategy of the proof is to construct a NFA equivalent to the 2DFA whose states are the valid crossing sequences of the 2DFA. Intuitively the NFA puts together parts of the computation of the 2DFA. This is done by guessing successive crossing sequences.

Let us briefly examine the relationship existing between adjacent crossing sequences. Suppose we are given an isolated tape square holding the symbol $a$ and let $q_{1}, \ldots, q_{k}$ and $p_{1}, \ldots, p_{l}$ the valid crossing sequences at the left and at the right boundary of the square. This scheme could not correspond to any concrete scenario, but we can test the local compatibility of two sequences as follows. If the tape head moves left from the square holding $a$ in state $q_{t}$, restart the automaton on the square holding $a$ in state $q_{t+1}$. Similarly, if the tape head moves right from the square containing $a$ in state $p_{3}$, restart the automaton on the square holding $a$ in state $p_{1+1}$. In this way we can test the local consistency of the two crossing sequences. More precisely let us define right matching and left matching for pairs of crossing sequences as follows: we say that $q_{1}, \ldots, q_{k}$ right matches $p_{1}, \ldots, p_{l}$ on $a$ if these sequences are consistent assuming we initially reach a moving right in state $q_{1}$. Similarly $q_{1}, \ldots, q_{k}$ left matches $p_{1}, \ldots, p_{l}$ on $a$ if these sequences are consistent assuming we initially reach $a$ in state $p_{1}$ moving left. More details about the construction of the NFA can be found in [8].

Then by Theorem 2 we have that each 2DFA $M$ is reducible to a NFA $M^{\prime}$ whose states are the valid crossing sequences of $M$. It can be easily seen that the reduction map is the identity function. Also note that $M^{\prime}$ is unambiguous. Otherwise we will have in $M^{\prime}$ two different accepting paths for a string $w$. Let $\pi$ and $\pi^{\prime}$ such paths. These paths corresponds to different computations for $w$ in $M$. This is absurd, because $M$ is deterministic.
Let $Q$ be the set of states of $M$ and let $Q^{\prime}$ the set of states of $M^{\prime}$. It is clear that if $|Q|=n$,

$$
\left|Q^{\prime}\right| \leq|Q|^{2 n}
$$

Then the number of states of $M^{\prime}$ is exponential in $n$. Nevertheless, for $k$-reversal-bounded 2DFA, this bound is too large because we know that the tape head can visit every tape square at most $k+1$ times. Then the tape head can cross every boundary between adjacent squares at most $k+1$ times. Then we have that

$$
\left|Q^{\prime}\right| \leq|Q|^{k+1}
$$

and $\left|Q^{\prime}\right|$ becomes polynomial in $n$. By the result of Bergadano and Varricchio we know that an automaton like $M^{\prime}$ is PAC learnable with membership queries. Then we can conclude that

Theorem 3 The class of $k$-reversal-bounded two way finite automata is PAC learnable with membership queries.


Figure 1: NFA associated to the DNF $\overline{x_{2}} \overline{x_{3}} \overline{x_{1}}+\overline{x_{2}} x_{3} x_{4}+$ $x_{2} x_{1} \overline{x_{4}}+x_{2} \overline{x_{1}}$.

## 6 Sat-k-DNF

Sat-k-DNF are particular DNF where, for any assignment, at most $k$ terms are satisfied. In this paper we prove that the class of Sat- $k$-DNF is PAC learnable with membership queries. We show this by reducing the class of Sat- $k$-DNF to the class of $k$-ambiguous automata.

Let $S=\left\{x_{1}, \ldots, x_{n}\right\}$ a set of variables, we can associate to each term of a Sat- $k$-DNF, defined over $S$, a deterministic automaton. We want that a string $w \in\{0,1\}^{n}$ is accepted by the automaton if and only if the corresponding term is satisfied by $w$.

For example consider the set $S=\left\{x_{1}, \ldots, x_{4}\right\}$ and the DNF

$$
\overline{x_{2}} \overline{x_{3}} \overline{x_{1}}+\overline{x_{2}} x_{3} x_{4}+x_{2} \overline{x_{1}} \overline{x_{4}}+x_{2} \overline{x_{1}} .
$$

To this formula we can associate the automata in figure 1.
All the automata are deterministic, but they can be seen, altogether, as a unique non deterministic automaton. Since, for any string of assignments over $S$, at most $k$ terms of the formula can be satisfied, we have that this automaton is also $k$-ambiguous. Then we have established a reduction between the class of Sat- $k$-DNF and the class of $k$-ambiguous automata. Note that the reduction map is the identity function.

Then, by the results of previous sections we can conclude that

Theorem 4 The class of Sat-k-DNF is PAC learnable with membership queries.

## 7 Decision trees

As a corollary of the result shown in the previous section we can give another proof of the PAC learnability of decision trees. We reach this goal by showing a reduction between the class of decision trees and the class of Sat-1-DNF. One can associate to every accepting path of the tree a term constituted by all the literals met in each node with the correct sign.


Figure 2: Example of decision tree

For example, to the tree in figure 2 we associate the DNF

$$
\overline{x_{2}} \overline{x_{3}} \overline{x_{1}}+\overline{x_{2}} x_{3} x_{4}+x_{2} x_{1} \overline{x_{4}}+x_{2} \overline{x_{1}} .
$$

Since every string $w$ can determine at most one accepting path in the tree, we have that the formula is Sat-1-DNF. We have found a reduction between the class of decision trees and the class of Sat-1-DNF, where the reduction map is the identity function. We can conclude that

Theorem 5 The class of decision trees is PAC learnable with membership queries.

AKNOWLEDGEMENTS: the first author was supported by the European Union under IV Framework grant "ILP2". The third author was supported by the Italian MURST $40 \%$ project "Algoritmi, Modelli di Calcolo e Strutture Informative".

## References

[1] D. Angluin. Learning regular sets from queries and counterexamples. Information and Computation, 75:87106, 1987.
[2] F. Bergadano and S. Varricchio. Learning Behaviors of Automata from Multiplicity and Equivalence Queries. SIAM J. Computing, to appear. Abstract available in LNCS 778, pages 54-62, 1994.
[3] J. Berstel and C. Reutenauer. Rational series and their languages. Springer-Verlag, Berlin, 1988.
[4] N. H. Bshouty. Exact Learning Boolean Functions via the Monotone Theory. In Proc. Symp. on the Foundations of Computer Science (FOCS), 1993.
[5] N. H. Bshouty and Y. Mansour. Simple Learning Algorithms for Decision Trees and Multivariate Polynomials.
[6] A. Ehrenfeucht and D. Haussler. Learning decision trees from random examples. Information and computation, 82:231-246, 1989.
[7] S. Eilenberg. Automata Languages and Machines, Vol. A, Academic Press, New York, 1974.
[8] J. E. Hopcroft and J. D. Ullman. Introduction to Automata Theory Languages and Computation, AddisonWesley, Reading, MA, 1979.
[9] J. Jackson. An efficient membership query algorithm for learning DNF with respect to the uniform distribution. In Proc. Symp. on the Foundations of Computer Science (FOCS), 1994.
[10] E. Kushilevitz and Y. Mansour. Learning Decision Trees using the Fourier Spectrum. SIAM J. Computing, 22(6):1331-1348, 1993.
[11] B. K. Natarajan. Machine Learning. Morgan Kaufmann Publishers, inc.
[12] L. Pitt and M. K. Warmuth. Prediction-preserving reducibility. Journal of Computer and System Science, 41(3):430-467, 1990.
[13] A. Salomaa and M. Soittola. Automata theoretic aspects of formal power series. Springer-Verlag, New York, 1978.
[14] R. E. Shapire, L. M. Sellie. Learning sparse multivariate polynomials over a field with queries and counterexamples. In Proceedings of the Sixth Annual ACM Workshop on Computational Learning Theory. July 1993.
[15] L. G. Valiant. A theory of learnable. Communications of the $A C M, 27(11): 1134-1142,1984$.

