Title	Learning DNF by Approximating Inclusion-Exclusion Formulae (Models of Computation and Algorithms)
Author(s)	Tarui, Jun; Tsukiji, Tatsuie
Citation	数理解析研究所講究録 (1999), 1093: 130-135
Issue Date	1999-04
URL	http://hdl.handle.net/2433/62960
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
1093 巻 1999 年 130-135

Learning DNF by Approximating Inclusion-Exclusion Formulae

Jun Tarui (垂井 淳) *　　Tatsuie Tsukiji (築地 立家) †

1. Introduction

Probably Approximately Correct learning algorithms generalize a small number of examples about an unknown concept into a function that can predict a future observation. More formally, let X and Y be the instance and outcome spaces, respectively. Then a PAC algorithm observes *randomly drawn examples* $(x, f(x))$ about an unknown concept $f : X \to Y$. These examples are independently and identically distributed random variables governed by an *arbitrary and unknown* distribution over X. With and only with these training examples, the algorithm aims to find a *hypothesis* $h : X \to Y$ that approximates the target concept f with respect to the same distribution. Hence it measures "goodness" of the hypothesis h by the probability $\mathrm{acc}(h) = \mathrm{Prob}_{x \in X}\{h(x) = f(x)\}$ called the *prediction accuracy*.

Valiant introduced the PAC model in a series of papers [12, 13], which is currently one of the most standard platforms for invention of polynomial-time learning algorithms. The PAC theory aims to learn as much general concept classes as possible, beginning from simple structures, e.g. depth-one or depth-two Boolean circuits. Valiant proved that Boolean conjunctions are polynomial-time learnable, and left the learning problem of the class DNF = {polynomial-size Disjunctive Normal Form formulae} for the future research. Here, as usual, a DNF formula is a disjunction of a amily of conjunctions of Boolean literals. These Boolean conjunctions are commonly called the terms of the DNF formula. The size of a DNF formula is the number of its (distinct) terms. Since then, a lot of literatures have proved learnability of subclasses of DNF by specifying either structural parameters of formulae or the distribution for the training examples ([1] provides a list of literatures). However, in spite of much effort, Valiant's original problem still remains unresolved.

Recently, Bshouty obtained a $2^{O(\sqrt{n}(\log n)^2)}$ time PAC algorithm for learning DNF. More strongly, he proved a similar upper bound in the exact non-proper learning model using equivalence queries. [1] The current paper is devoted to give a further understanding to this learning time in the PAC model: roughly speaking, we show that Bshouty's leaning time $2^{\tilde{O}(\sqrt{n})}$ is possible and best possible to be attained [2], if learning algorithms search hypotheses in the order of their "succinctness".

In more detail, we analyze upper and lower bounds on size of Boolean conjunctions necessary and sufficient to approximate a given DNF formula by accuracy slightly better than 1/2 (here we define the size of a Boolean conjunction as the number of distinct variables on which it depends.) Such an analysis determines the performance of a naive search algorithm that exhausts Boolean conjunctions in the order of their sizes. In fact, our analysis does not depend on kinds of symmetric functions to be exhausted: instead of conjunctions, counting either disjunctions, parity functions, majority functions, or even general symmetric functions, derives the same learning results from similar analyses.

Naive search algorithms find only weakly accurate hypotheses, so they need to be boosted on accuracy to complete the PAC learning process. Schapire [11] and later on Freund [4] invented efficient algorithms that boost the accuracy of a given weak algorithm (in sense of [10]) under a given distribution, which we refer to as *general boosters*. In this paper, we study the performance of naive search algorithms in the frameworks of general boosters. On the other hand, if the target distribution is specified as the uniform distribution, naive search algorithms have been widely applied to learn DNF cooperating with specific boosters. In fact, Verbeurgt [14] showed that naively searching Boolean conjunctions learns the class DNF in quasi-polynomial time. Linial, Mansor and Nisan [8] showed,

*Department of Communications and Systems, University of Electro-communications,Chogugaoka, Chofu-shi, Tokyo 182, Japan.
E-mail address: jun@sw.cas.uec.ac.jp
†School of Informatics and Sciences, Nagoya University, Nagoya 464-01, Japan.
E-mail address: tsukiji@info.human.nagoya-u.ac.jp

[1] Angluin's proof [2] provides a similar lower bound on the number of *proper* equivalence queries.

[2] $\tilde{O}(t(n)) = \cup_{k \geq 0} O(t(n) \log^k(t(n)))$.

moreover, that even so does the class AC^0 by searching short parity functions.

In 1989, Linial and Nisan [9] showed positive and negative results about approximating the *inclusion-exclusion formula* by a linear combination of the sizes of short intersections. Our analysis is built on some of their results.

Our positive results for approximating and learning DNF are:

Theorem 1.1. For any s-term DNF formula f and any distribution over the instance space X there exists a size-$O(\sqrt{n}\log s)$ conjunction h that satisfies

$$|\mathrm{Prob}_{x\in X}\{f(x) = h(x)\} - 1/2| \;=\; 2^{-O(\sqrt{n}\log n\log s)}.$$

Theorem 1.2. Any naive search algorithm, cooperating with general boosters, PAC learns the class DNF in $2^{O(\sqrt{n}(\log n)^2)}$ time with respect to any distribution.

Note that if we adopt Freund's booster [4], then for a given DNF formula and a given distribution, the algorithm outputs as a highly accurate hypothesis a majority-vote of Boolean conjunctions of size at most $O(\sqrt{n}\log n\log s)$.

Our negative result is:

Theorem 1.3. For any Boolean conjunction f of length $\Theta(n)$ and any constant $0 \le \varepsilon \le 1/2$ there exist $k = \Omega(\sqrt{n\varepsilon})$ and a joint-distribution over $X \times Y$ such that we have $\mathrm{Prob}_{(x,y)\in X\times Y}\{f(x) = y\} \ge 1-\varepsilon$ and $\mathrm{Prob}_{(x,y)\in X\times Y}\{h(x) = y\} = 1/2$ for any Boolean function $h : \{0,1\}^n \to \{0,1\}$ that depends on at most k variables.

Therefore, under such a joint-distribution, a naive search algorithm must enumerate at least $2^{\Omega(\sqrt{n\varepsilon}\log n)}$ number of symmetric functions until finding a decision-rule that is better than guessing at random. Note that in the standard PAC model, where $\mathrm{Prob}_{(x,y)\in X\times Y}\{f(x) = y\} = 1$, any Boolean conjunction can be learned in $O(n\log n)$ time by searching only literals [12].

2. Learning Frameworks

A formal definition for a PAC algorithm requires two real parameters $0 \le \varepsilon, \delta \le 1$, called the *accuracy* and *confidence* parameters of the algorithm, respectively. Let C, H be classes of functions from the instance space X to the outcome space Y.

Definition 2.1. A randomized algorithm is called a PAC algorithm that learns the target class C by the hypothesis class H within accuracy $1-\varepsilon$ and confidence $1-\delta$ if for any $f \in C$ the algorithm, given access to random examples $(x, f(x))$, outputs $h \in H$ that achieves $\mathrm{acc}(h) \ge 1-\varepsilon$ with probability at least $1-\delta$.

The PAC model is an abstraction of practical situations where machines learn from their own random experiences. Most of all, it assumes that the training examples are perfectly consistent with the target concept. Empirical datum, however, involve inevitable errors (i.e. inconsistency) due to inaccuracy in measurement systems, intervention by malicious adversaries, uncertainty of the nature, and so on. Under such practical situations, the training examples $(x, y) \in X \times Y$ are no more consistent with the target concept f. Kearns and Schapire [7], and Haussler [5], assumed that they are governed by an *unknown joint-distribution* over the *observation space* $X \times Y$. Thus a random example $(x, y) \in X \times Y$ may differ from $(x, f(x))$ with probability $\mathrm{Prob}_{x,y}\{f(x) \ne y\}$.

Angluin and Laird [3] introduced so-called the *white noise* into the target distribution. The joint-distribution has the white noise of rate η, $0 \le \eta \le 1/2$, if it is the production over $x \in X$ of the identical distributions over Y that satisfies $\mathrm{Prob}_{y\in Y}\{f(x) \ne y\} = \eta$.

Definition 2.2. We say that a PAC algorithm tolerates the white noise of rate η if it PAC learns under any distribution having the white noise of rate η.

All of the discovered noise-tolerable algorithms depend on some statistical information over the training examples, rather than the point-wise information of them. Kearns [6] bound these algorithms into the notion of *Statistical Query* learning algorithms. An SQ-algorithm asks to the query of the probability $\mathrm{Prob}_{x,y}\{\chi(x,y) = 1\}$ for a predicate χ defined over the observation space to the SQ-oracle. The SQ-oracle then returns an estimation $\mathrm{est}(\chi)$ of it within error τ, hence it satisfies

$$|\mathrm{Prob}_{x,y}\{\chi(x,y) = 1\} - \mathrm{est}(\chi)| \le \tau.$$

Where $0 \le \tau \le 1$ is a constant parameter associated with the SQ-oracle called the tolerance of the SQ-oracle.

Definition 2.3. An SQ-algorithm is defined as as in Definition 2.1 by given access to the SQ-oracle instead of the random training examples.

Theorem 2.4. (Kearns [6]) If a class C is SQ learnable by a class H within accuracy $1-\varepsilon$ and confidence

$1-\delta$ from the oracle of tolerance τ in $O(t)$ time, then C is PAC learnable by H within accuracy $1-\varepsilon$ and confidence $1-\delta$ under the white noise of rate η in $O\big(t\tau^{-2}(1/2-\eta)^{-2}\log(1/\delta)\big)$ time

The current paper learns Boolean concepts. Thus for each dimension $n = 0, 1, 2, \ldots$ the instance space is the n-dimensional Boolean cube $X = \{0,1\}^n$ and the outcome space is $Y = \{0,1\}$, so target concepts and hypotheses are Boolean functions. Learning algorithms are assumed to know the parameters so far appeared; the dimension n of the instance space, the accuracy and confidence parameters, the rate of the whilte noise, and the tolerance of the SQ-oracle.

3. Approximating Inclusion-Exclusion Formulae

Our positive and negative results for learning DNF are built on positive and negative results for approximating the inclusion-exclusion formula, respectively, established by Linial and Nisan [9]. For a given family of n sets $\{A_1, \ldots, A_n\}$, the inclusion-exclusion formula on them measures the union size by the sizes of the intersections of all the subfamilies as

$$\begin{aligned}\|A_1 \cup A_2 \cup \cdots \cup A_n\| \;=\; & \\ & \sum_i \|A_i\| - \sum_{i<j} \|A_i \cap A_j\| \\ & + \sum_{i<j<k} \|A_i \cap A_j \cap A_k\| \\ & - \cdots + (-1)^n \|A_1 \cap A_2 \cap \cdots \cap A_n\|.\end{aligned}$$

Linial and Nisan asked to approximate the union size by using only initial terms of them, in other words, by only the sizes of the "short" intersections. They translated the problem into a problem of approximating a certain discrete delta function by low degree polynomials, where the sizes of the intersections correspond to the degrees of the polynomials. We state two of their results in a general probability space $(X, \Sigma, \mathcal{D})$ where X is any set, Σ is a σ-field over X and $\mathcal{D}$ is a probability measure over (X, Σ). We denote the weight of a set $A \subseteq X$ by $\|A\|_{\mathcal{D}} = \mathrm{Prob}_{x \in X}\{x \in A\}$.

Theorem 3.1. (Linial and Nisan [9]) For any integers $n \geq 1$ and $k \geq c_1\sqrt{n}$, where $c_1 > 0$ is a certain constant, there exist constants $\alpha_1^{k,n}, \alpha_2^{k,n}, \ldots, \alpha_k^{k,n}$ such that for every probability space $(X, \Sigma, \mathcal{D})$ and every collection of sets $A_1, \ldots, A_n$ we have

$$\Big(1 \pm O\big(e^{-2k/\sqrt{n}}\big)\Big)\left\|\bigcup_{i=1}^n A_i\right\|_{\mathcal{D}}$$

$$= \sum_{0<|S|\leq k} \alpha_{|S|}^{k,n} \left\|\bigcap_{i\in S} A_i\right\|_{\mathcal{D}} \qquad (1)$$

Linial and Nisan described these constants $\alpha_i^{k,n}$ explicitly in terms of the coefficients of the Chebyshev polynomial of degree k. In special, their description derives $\left|\alpha_i^{k,n}\right| < 2^k$.

Theorem 3.2. (Linial and Nisan [9]) For any integers $n \geq 1$ and $k \leq c_2\sqrt{n}$, where c_2 is a certain constant, there exist a probability space $(X, \Sigma, \mathcal{D})$ and two collections of sets $A_1, \ldots, A_n$ and $B_1, \ldots, B_n$ such that we have both

$$\frac{\|\bigcup_{i=1}^n A_i\|_{\mathcal{D}}}{\|\bigcup_{i=1}^n B_i\|_{\mathcal{D}}} = \Omega\Big(\frac{n}{k^2}\Big) \qquad (2)$$

and

$$\left\|\bigcap_{i\in S} A_i\right\|_{\mathcal{D}} = \left\|\bigcap_{i\in S} B_i\right\|_{\mathcal{D}} \qquad (3)$$

for all the nonempty sets $S \subseteq \{1, \ldots, n\}$ with $0 < |S| \leq k$.

4. Learnability of DNF

In this section we show how to derive learning DNF in subexponential time from Theorem 3.1.

Without loss of generality, we may assume that the constant function 0 is always contained in the terms of a given DNF formula. Then it is easy to see that any DNF formula can be weakly approximated by some terms.

Lemma 4.1. For any size-s DNF formula f and any distribution over the instance space X there exists a term g of f that satisfies

$$|\mathrm{Prob}_{x\in X}\{f(x) = g(x)\} - 1/2| \;>\; 1/10s. \qquad (4)$$

Proof. We suppose that there is no such term of f, and will derive a contradiction. Inverting (4) for 0 derives

$$|\mathrm{Prob}_x\{f(x) = 0\} - 1/2| \;\leq\; 1/10s$$

or equivalently

$$|\mathrm{Prob}_x\{f(x) = 1\} - 1/2| \;\leq\; 1/10s. \qquad (5)$$

For any term g of f, $g(x) = 1$ forces $f(x) = 1$, hence

$$\begin{aligned}\mathrm{Prob}_x\{g(x) = 1\} \;=\; & \\ & \mathrm{Prob}_x\{f(x) = 1\} - \mathrm{Prob}_x\{f(x) \neq g(x)\} \\ & \leq\; 1/2 + 1/10s - (1/2 - 1/10s) \\ & \leq\; 1/5s.\end{aligned}$$

However, $f(x) = 1$ forces $g(x) = 1$ for some term g of f, so we obtain

$$\begin{aligned}\mathrm{Prob}_x\{f(x)=1\} &\le \sum_g \mathrm{Prob}_x\{g(x)=1\} \\ &\le \sum_g 1/5s \;=\; 1/5.\end{aligned}$$

This contracts (5). □

Due to Theorem 3.1, a conjunction that approximates the target DNF formula provides a much shorter conjunction that still approximates the target to some extent.

Definition 4.2. For any functions $f, F : X \to Y$ and any distribution over X, the bias of F to f (with respect to the distribution) is

$$\begin{aligned}\mathrm{bias}_f(F) &= \mathrm{Prob}_x\{F(x)=1 \wedge f(x)=1\} \\ &\quad -\mathrm{Prob}_x\{F(x)=1 \wedge f(x)=0\}\end{aligned}$$

and the correlation of f and F is

$$\mathrm{cor}(f,F) \;=\; \mathrm{Prob}_{x\in X}\{F(x)=f(x)\} - 1/2.$$

Lemma 4.3.

$$\mathrm{bias}_f(F) \;=\; \mathrm{cor}(f,F) - \mathrm{cor}(f,0).$$

Proof.

$$\begin{aligned}\mathrm{bias}_f(F) &= \mathrm{Prob}_x\{F(x)=f(x)\} \\ &\quad -\mathrm{Prob}_x\{f(x)=0\} \\ &= (\mathrm{Prob}_x\{F(x)=f(x)\} - 1/2) \\ &\quad -(\mathrm{Prob}_x\{f(x)=0\} - 1/2) \\ &= \mathrm{cor}(f,F) - \mathrm{cor}(f,0).\end{aligned}$$

□

Now we prove Theorem 1.1.

Proof of Theorem 1.1. Lemma 4.1 provides a conjunction g that is a term of f and correlated with f as

$$|\mathrm{cor}(f,g)| \;>\; 1/10s. \qquad (6)$$

We may assume without loss of generality that $g(x) = \bar{x}_1 \wedge \bar{x}_2 \wedge \cdots \wedge \bar{x}_n$.

Set $\gamma = 2^{-c_3\sqrt{n}\log n \log s}$ for a positive constant c_3. We may assume that 0 is not correlated with f as $|\mathrm{cor}(f,0)| < \gamma$. Lemma 4.3 then yields

$$\begin{aligned}|\mathrm{bias}_f(\bar{g}) - \mathrm{cor}(f,\bar{g})| &= |\mathrm{cor}(f,0)| \\ &< \gamma,\end{aligned}$$

so from (6) and $|\mathrm{cor}(f,g)| = |\mathrm{cor}(f,\bar{g})|$ we obtain

$$|\mathrm{bias}_f(\bar{g})| \;>\; 1/10s - \gamma. \qquad (7)$$

Let $A_i = \{((a_1,\ldots,a_n),1) \in X \times Y : a_i = 1\}$ and $B_i = \{((a_1,\ldots,a_n),0) \in X \times Y : a_i = 1\}$. Let $k = c_4\sqrt{n}\log s$ for a constant $c_4 > 0$ and D be the given target distribution over X. Theorem 3.1 then writes $\mathrm{bias}_f(\bar{g})$ as a linear combination over the biases of short conjunctions $h_S = \bigwedge_{i\in S} x_i$:

$$\begin{aligned}\mathrm{bias}_f(\bar{g}) &= \|\cup_{i=1}^n A_i\|_D - \|\cup_{i=1}^n B_i\|_D \\ &= \sum_{0<|S|\le k} \alpha_{|S|}^{k,n} \left(\|\cap_{i\in S} A_i\|_D \right. \\ &\quad \left. - \|\cap_{i\in S} B_i\|_D\right) \pm O\left(e^{-2k/\sqrt{n}}\right) \\ &= \sum_{0<|S|\le k} \alpha_{|S|}^{k,n} \mathrm{bias}_f(h_S) \\ &\quad \pm O\left(e^{-2k/\sqrt{n}}\right).\end{aligned}$$

Therefore if we choose c_4 large enough then from (7) we obtain

$$(1 \pm o(1))\mathrm{bias}_f(\bar{g}) = \sum_{0<|S|\le k} \alpha_{|S|}^{k,n} \mathrm{bias}_f(h_S).$$

Thus (7) indicates that one of these conjunction h_S satisfies

$$\begin{aligned}|\mathrm{bias}_f(h_S)| &\ge (1 \pm o(1))\binom{n}{k}^{-1} 4^{-k} |\mathrm{bias}_f(\bar{g})| \\ &= 2^{-O(\sqrt{n}\log n \log s)},\end{aligned}$$

so if we choose c_3 large enough then Lemma 4.3 derives the required lower bound on the corelation of f and h_S:

$$\begin{aligned}|\mathrm{cor}(f,h_S)| &\ge 2^{-O(\sqrt{n}\log n \log s)} - \gamma \\ &= 2^{-O(\sqrt{n}\log n \log s)}.\end{aligned}$$

□ Theorem 1.1

Corollary 4.4. Any size-s DNF formula f is SQ learnable by either a Boolean conjunction or a disjunction of size-$O(\sqrt{n}\log n \log s)$ within accuracy $1/2 + 2^{-O(\sqrt{n}\log n \log s)}$ and confidence 1 from the SQ-oracle of tolerance $2^{-O(\sqrt{n}\log n \log s)}$ in $2^{O(\sqrt{n}\log n \log s)}$ time.

Proof. For $k = O(\sqrt{n}\log s)$, Theorem 1.1 guarantees a size-k conjunction h that satisfies

$$|\mathrm{cor}(f,h)| \;=\; 2^{-O(\sqrt{n}\log n \log s)}.$$

Hence choosing the tolerance $\tau = 2^{-O(\sqrt{n}\log n \log s)}$ sufficiently small, the naive search algorithm finds a conjunction h of size at most k after at most $\binom{n}{k}2^k$ SQ-queries that satisfies

$$|\mathrm{est}(f = h) - 1/2| \;=\; 2^{-O(\sqrt{n}\log n \log s)},$$

so

$$|\mathrm{cor}(f, h)| \;=\; 2^{-O(\sqrt{n}\log n \log s)}.$$

If $f = h$ happens more certainly than $f \neq h$ then the naive search algorithm outputs h itself, otherwise its negation $\bar{h}$ that is represented by a disjunction due to the De-Morgan rule. □

Now we let a general booster improve the accuracy of the naive search algorithm.

Theorem 4.5. (Schapre [11], Feund [4]) Given an SQ-algorithm that learns C by H within accuracy $1 - \varepsilon_0 > 1/2$ and confidence $1 - \delta_0$ from the SQ-oracle of tolerance τ in $O(t)$ time. Then for any accuracy and confidence parameters ε and δ, respectively, one can design an SQ-algorithm that learns C by H from the SQ-oracle of torelance τ within time polynomial in t, n, $(1/2 - \varepsilon_0)^{-1}$,$1/\delta_0$, $1/\varepsilon$ and $\log(1/\delta)$.

We apply this boosting theorem to Theorem 4.4 and obtain:

Corollary 4.6. s-term DNF is SQ learnable from the SQ-oracle of tolerance $2^{-O(\sqrt{n}\log n \log s)}$ in time polynomial in $2^{O(\sqrt{n}\log n \log s)}$, $1/\varepsilon$ and $\log(1/\delta)$.

Applying Theorem 2.4 then derives:

Corollary 4.7. s-term DNF is PAC learnable under the white noise of rate η in time polynomial in $2^{O(\sqrt{n}\log n \log s)}$, $(1/2 - \eta)^{-1}$, $1/\varepsilon$ and $\log(1/\delta)$.

Theorem 1.2 is now obtained by putting $s = \mathrm{poly}(n)$, $\eta = 0$ and $1/\varepsilon = 1/\delta = O(1)$.

5. Unlearnability of DNF

Finally, we show that Theorem 3.2 derives Theorem 1.3.

Proof of Theorem 1.3. We may assume without loss of generality that $f(x) = \bar{x}_1 \wedge \bar{x}_2 \wedge \cdots \wedge \bar{x}_{n_0}$ for $n_0 = \Omega(n)$. Theorem 3.2 gives a distribution $\mathcal{E}$ and two collections of sets $A_1, \ldots, A_{n_0}$ and $B_1, \ldots, B_{n_0}$ that satisfy both (2) and (3). Without loss of generality, we may assume existence of disjoint universal sets U and V that contain all of A_i and B_i, respectively, and $\mathcal{E}$ is defined over $U \cup V$. We can moreover choose U as

$$\bigcup_{i=1}^{n_0} A_i \;=\; U \tag{8}$$

and adjust V and $\mathcal{E}$ so to satisfy

$$||U||_{\mathcal{E}} \;=\; ||V||_{\mathcal{E}}. \tag{9}$$

We denote for any set $A \subseteq U$ that $A^1 = A$ and $A^0 = U - A$. Similarly, for $B \subseteq V$, $B^1 = B$ and $B^0 = V - B$. From (9) and the inclusion-exclusion formula, we can extend (3) to

$$\left\| \bigcap_{i \in S} A_i^{a_i} \right\|_{\mathcal{E}} = \left\| \bigcap_{i \in S} B_i^{a_i} \right\|_{\mathcal{E}} \tag{10}$$

for any nonempty set $S \subseteq \{1, \ldots, n_0\}$ with $0 < |S| \leq k$ and any $a_i \in \{0, 1\}$ with $i \in S$.

Now we define a joint-distribution over $X \times Y$ by

$$\begin{aligned}\mathrm{Prob}_{x,y}\{(x_1, \ldots, x_{n_0}, y) = (a_1, \ldots, a_{n_0}, 0)\} \\ = \; \alpha||\cap_{i=1}^{n_0} A_i^{a_i}||_{\mathcal{E}}\end{aligned}$$

and

$$\begin{aligned}\mathrm{Prob}_{x,y}\{(x_1, \ldots, x_{n_0}, y) = (a_1, \ldots, a_{n_0}, 1)\} \\ = \; \alpha||\cap_{i=1}^{n_0} B_i^{a_i}||_{\mathcal{E}}\end{aligned}$$

for any $a = (a_1, a_2, \ldots, a_{n_0}) \in \{0, 1\}^{n_0}$. Where α is the constant for normalizing $\mathcal{E}$, hence $\alpha(||U|| + ||V||) = 1$.

Now we check that this distribution satisfies

$$\mathrm{Prob}_{(x,y) \in X \times Y}\{f(x) = y\} > 1 - \varepsilon, \tag{11}$$

and

$$\mathrm{Prob}_{(x,y) \in X \times Y}\{h(x) = y\} = 1/2 \tag{12}$$

for any Boolean function $h : \{0, 1\}^n \to \{0, 1\}$ that depends on at most k variables. We begin with establishing (12) for the special case that $h = 0$. From (8) and (9)

$$\begin{aligned}&\mathrm{Prob}_{x,y}\{y = 0\} \\ &\quad = \; ||\cup_{i=1}^{n_0} A_i^{a_i}||_{\mathcal{E}} \\ &\quad = \; 1/2\end{aligned} \tag{13}$$

If we choose k sufficiently small then (2) implies

$$\begin{aligned}&\frac{\mathrm{Prob}_{x,y}\{f(x) = 0 \wedge y = 0\}}{\mathrm{Prob}_{x,y}\{f(x) = 0 \wedge y = 1\}} \\ &\quad = \; \frac{\alpha||\cup_{i=1}^{n_0} A_i||_{\mathcal{E}}}{\alpha||\cup_{i=1}^{n_0} B_i||_{\mathcal{E}}} \\ &\quad > \; \Omega(n/k^2) \\ &\quad > \; 1/\varepsilon.\end{aligned}$$

From (8) $f(x) = 1$ implies $y = 1$, hence

$$\begin{aligned} &\mathrm{Prob}_{x,y}\{f(x) \neq y\} \\ &\quad = \mathrm{Prob}_{x,y}\{f(x) = 0 \wedge y = 1\} \\ &\quad < \varepsilon. \end{aligned}$$

Finally we check (12). To any Boolean function h we associate its bias as

$$\begin{aligned} \mathrm{bias}(h) \;=\; &\mathrm{Prob}_{x,y}\{h(x) = 1, y = 1\} \\ &-\mathrm{Prob}_{x,y}\{h(x) = 1, y = 0\}. \end{aligned}$$

If h is a Boolean function that depends on a set S_0 of at most k variables then letting $h_A = \prod_{x_i \in S_0} x_i^{A_i}$ for $A \in \{0,1\}^k$ where $x_i^1 = x_i$ and $x_i^0 = \bar{x}_i$, h can be written as $h = \sum_A c_A h_A$ for some integers c_A, so the linearlity of bias and (10) derive

$$\mathrm{bias}(h) \;=\; \sum_A c_A \mathrm{bias}(h_A) \;=\; 0,$$

hence combining it with (13) yields that

$$\begin{aligned} \mathrm{Prob}_{\mathcal{D}}\{h(x) = y\} \;&=\; \mathrm{bias}(h) + \mathrm{Prob}_{\mathcal{D}}\{y = 0\} \\ &=\; 1/2. \end{aligned}$$

□ Theorem 1.3

Acknowledgments

We would like to thank to Professor Osam Watanabe for his helpful discussions in the early stage of this research.

References

[1] H. Aizenstein and L. Pitt. On the learnability of disjunctive normal from formulas. *Machine Learning*, 19:183–208, 1995.

[2] D. Angluin. Negative results for equivalence queries. *Machine Learning*, 5:121–150, 1990.

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

[4] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256, 1995.

[5] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

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

[7] M. Kearns and R. Schapire. Efficient distribution-free learning of probabilistic concepts. *Proceedings of the 31th IEEE Symposium on Foundations of Computer Science*, pages 382–391, 1990.

[8] N. Linial, Y. Mansor, and N. Nisan. Constant depth circuits, fourier transforms and learnability. *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 574–579, 1989.

[9] N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10:349–365, 1990.

[10] L. Pitt and L. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35:965, 1988.

[11] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2), 1990.

[12] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[13] L. Valiant. Learning disjunctions of conjunctions. *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 560–566, 1985.

[14] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.