Uniform-Distribution Attribute Noise Learnability

Nader H. Bshouty
Technion
Haifa 32000, Israel
bshouty@cs.technion.ac.il

Jeffrey C. Jackson*
Duquesne University
Pittsburgh, PA 15282-1754, U.S.A.
jackson@mathcs.duq.edu

Christino Tamon
Clarkson University
Potsdam, NY 13699-5815, U.S.A.
tino@clarkson.edu

October 5, 2001

Abstract

We study the problem of PAC-learning Boolean functions with random attribute noise under the uniform distribution. We define a *noisy distance* measure for function classes and show that if this measure is small for a class $\mathcal{C}$ and an attribute noise distribution D then $\mathcal{C}$ is not learnable with respect to the uniform distribution in the presence of noise generated according to D. The noisy distance measure is then characterized in terms of Fourier properties of the function class. We use this characterization to show that the class of all parity functions is not learnable for any but very concentrated noise distributions D. On the other hand, we show that if $\mathcal{C}$ is learnable with respect to uniform using a standard Fourier-based learning technique, then $\mathcal{C}$ is learnable with time and sample complexity also determined by the noisy distance. In fact, we show that this style algorithm is the best possible for learning in the presence of attribute noise.

1 Introduction

The problem of attribute noise in PAC-learning was studied originally by Shackelford and Volper [10] for the case of k-DNF expressions. Their *uniform* attribute noise model consists of a Bernoulli process that will either flip or not flip each attribute value with a fixed probability $p \in [0, 1]$ that is the same for every attribute. While Shackelford and Volper assumed that the learner knows the noise rate p, Goldman and Sloan [6] proved that this assumption is not necessary in order to learn monomials.

In addition to uniform attribute noise, Goldman and Sloan also considered a *product* noise model in which there are n noise rates p_i, one for each distinct attribute x_i, $i \in [n]$. They showed that if the product noise rates p_i are unknown, then no PAC-learning algorithm exists that can tolerate a noise rate higher than 2ϵ, where ϵ is the required-accuracy parameter for PAC learning. Their proof uses the method of induced distribution of Kearns and Li [8]. Subsequently, Decatur and Gennaro [3] proved that if the different noise rates are *known* (or if some upper bound on them

[*] This material is based upon work supported by the National Science Foundation under Grants No. CCR-9800029 and CCR-9877079.

is given) then there exist efficient PAC-learning algorithms for simple classes such as monomials and k-DNF.

In this paper we consider a very general attribute noise model, but limit the distribution that will be used to generate examples and to evaluate the accuracy of the hypothesis generated by the learning algorithm. Specifically, we focus on the problem of PAC learning with respect to the uniform distribution over examples, with little or no constraint on the distribution used to generate attribute noise in the examples. We give both lower and upper bounds.

First, we define a measure of *noisy distance* for concept classes and show that the sample size required for PAC learning a class over the uniform distribution is inversely proportional to the noisy distance of the class. We also give a characterization of the noisy distance in terms of Fourier properties of the class. As an example of how this characterization can be used, we show that the class of all parity functions is not (even information theoretically) PAC learnable with respect to uniform unless the attribute noise distribution puts nonnegligible weight on one or more noise vectors. This holds even if the noise process is known. On the other hand, we observe as a corollary of a result of Blum, Burch, and Langford [1] that the class of monotone Boolean functions is weakly PAC-learnable even if the noise process if unknown.

We then turn to developing positive learnability results. Specifically, we show that any concept class that is PAC-learnable with respect to the uniform distribution using an algorithm in the style of Linial, Mansour, and Nisan [9] can be adapted to handle attribute noise, assuming the probability distribution of the noise process is known. However, the noisy distance of a class depends on the noise distribution, so the sample complexity of our algorithm is dependent on the noise process as well as the usual PAC factors. The dependence of the the sample complexity of our algorithm matches, to within polynomial factors, our lower bound for learning with attribute noise. We also give a simple argument showing that if the noise process is unknown then even extremely simple concept classes are not (strongly) learnable.

Our Fourier techniques share some commonalities with methods developed by Benjamini, Kalai, and Schramm [2] in their work that studied percolation and its relation to noise sensitivity of Boolean functions. Their techniques, like ours, were strongly motivated by the influential work of Kahn, Kalai, and Linial [7] on Fourier analysis of Boolean functions.

2 Definitions and Notation

The problem considered in this paper is *PAC learning* Boolean functions under some fixed distribution over instances when *attribute noise* is also applied to the instances. To a lesser extent, we also consider *classification noise*. We now define these concepts more precisely below. For simplicity, our definition suppresses some details of standard definitions (particularly the notion of *size* of functions) that are not critical to the results in this paper.

For a natural number n, we consider classes of Boolean functions $f : \{0,1\}^n \to \{-1,+1\}$ and distributions over $\{0,1\}^n$. The uniform distribution on $\{0,1\}^n$ is denoted U, i.e., $U(x) = 2^{-n}$, for all $x \in \{0,1\}^n$. The *bitwise exclusive-or* of two n-bit vectors $a, b \in \{0,1\}^n$ is denoted $a \oplus b$. The *unit vector* $e_i \in \{0,1\}^n$ has its i-th bit set to one and all other bits set to zero. For $a \in \{0,1\}^n$, the parity function χ_a is defined as $\chi_a(x) = (-1)^{\sum_{i=1}^n a_i x_i}$. It is known that any Boolean function $f : \{0,1\}^n \to \{-1,+1\}$ can be represented as a weighted sum of parity functions (see [9])

$$f(x) = \sum_{a \in \{0,1\}^n} \hat{f}(a) \chi_a(x)$$

where $\hat{f}(a) = \mathbf{E}_U[f(x)\chi_a(x)]$ is the *Fourier coefficient* of f at a. This is called the Fourier representation of f and is a direct consequence of the fact that $\{\chi_a \mid a \in \{0,1\}^n\}$ forms an orthonormal basis for all Boolean (or even real-valued) functions over $\{0,1\}^n$, i.e., $\mathbf{E}_U[\chi_a(x)\chi_b(x)]$ is one if $a = b$ and zero otherwise.

The focus of the paper is on a learning model in which the instance distribution is uniform and the noise process is characterized by a pair of parameters (D, R). The noise process can be viewed as drawing a random vector a from the distribution D (representing the attribute noise process) and a random value b from the distribution R (representing classification noise), then returning the exclusive OR of a with the original example vector x and the exclusive OR of the label $f(x)$ with b. So the noise process changes an example $(x, f(x))$ to an example $(x \oplus a, f(x) \oplus b)$ (actually, because we consider functions mapping to $\{-1,+1\}$, we will assume that R produces values in $\{-1,+1\}$ and replace the latter $\oplus$ with multiplication). We will call this (D, R)-noise and denote the oracle that returns a (D, R)-noisy example for f with respect to the uniform distribution by $EX_{D,R}(f, U)$.

Definition 1 *Let C be a concept class containing functions $f : \{0,1\}^n \to \{-1,+1\}$. Then C is* PAC learnable under the uniform distribution with (D, R)-noise *if there is an algorithm A such that for any $\epsilon, \delta \in (0,1)$ and for any target $f \in C$, given the inputs ϵ, δ and access to a noisy example oracle $EX_{D,R}(f, U)$, the algorithm A outputs a hypothesis h such that $\Pr_U[h \neq f] < \epsilon$ with probability at least $1 - \delta$. The algorithm must make a number of oracle calls (have* sample complexity*) at most polynomial in n, $1/\epsilon$, and $1/\delta$. The* time complexity *of A is the number of computation steps taken by A. A PAC algorithm is* efficient *if its time complexity is also polynomial in n, $1/\epsilon$, and $1/\delta$.*

If the classification noise process R always returns 0, then (D, R)-noise is simply attribute noise and we refer to it as D-noise. Our lower bounds focus on this type of noise.

3 Model Transformation

Before developing our main results, it is useful to relate the (D, R)-noise model to another model where the example $(x, f(x))$ is changed to $(x, f(x \oplus a)b)$ for a random vector a drawn according to distribution D and $b \in \{-1,+1\}$ drawn according to distribution R.

Lemma 1 *Let $U = U_n$ be the uniform distribution over n-bit vectors and $(D, R) = (D_n, R_n)$ be any distribution over $\{0,1\}^n$ and $\{-1,+1\}$, respectively. Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function. If $X \in_U \{0,1\}^n$, $A \in_D \{0,1\}^n$ and $B \in_R \{-1,+1\}$ are random variables then the random variables $(X \oplus A, f(X)B)$ and $(X, f(X \oplus A)B)$ have identical distributions.*

Proof Consider the random variables $X_1 = (X, A, B)$ and $X_2 = (X \oplus A, A, B)$. Since X is uniformly distributed, X_1 and X_2 are identically distributed. Define $\varphi(x, y, z) = (x, f(x \oplus y)z)$. Then

$$(X \oplus A, f(X)B) = \varphi(X_2) = \varphi(X_1) = (X, f(X \oplus A)B),$$

completing the claim. □

This lemma is key to our subsequent results, as it allows us to consider the easier noise model of $(X, f(X \oplus A)B)$ instead of the random attribute noise model when learning is with respect to the uniform distribution.

4 Sample Complexity Lower Bound

In this section we give a lower bound for PAC-learning with D-noise. Because D-noise is a special case of (D, R)-noise, our lower bounds immediately generalize to this model as well.

We start with some intuition for the lower bound. Let C be the class being learned. Let f and g be two functions in the class C and suppose $\Pr_U[f \neq g] > \epsilon$. If for a fixed x and distribution D the expectation $\mathbf{E}_{a\sim D}[f(x \oplus a)]$ is very close to $\mathbf{E}_{a\sim D}[g(x \oplus a)]$, then we cannot notice the difference between $(x, f(x \oplus a_1))$ and $(x, g(x \oplus a_2))$. Now since the example oracle we consider chooses x according to the uniform distribution, we will look at $\mathbf{E}_x[|\mathbf{E}_a[f(x \oplus a) - g(x \oplus a)]|]$. This, we will show, is a good measure for learnability with noise. We now formalize the above.

Definition 2 *Let C be a concept class over $\{0, 1\}^n$ and let $f, g \in C$. Let D be any distribution over $\{0, 1\}^n$. Then the* noisy distance *between f and g under the distribution D is defined as*

$$\Delta_D(f, g) \equiv \frac{1}{2}\mathbf{E}_x[|\mathbf{E}_a[f(x \oplus a) - g(x \oplus a)]|],$$

where the expectation of x is taken over the uniform distribution over $\{0, 1\}^n$ and the expectation of a is taken with respect to D. For a concept class C let

$$\Delta^{\epsilon}_D(C) \equiv \min\{\Delta_D(f, g) \mid f, g \in C \text{ with } \Pr_U[f \neq g] > \epsilon\}.$$

The following theorem states an information-theoretic lower bound on the number of samples required by any PAC learning algorithm.

Theorem 2 *Let C be a concept class and, for fixed ϵ and D, represent $\Delta^{\epsilon}_D(C)$ by Δ. Then any PAC learning algorithm for C under a D-distribution noise that, with probability at least $1 - \delta$, outputs an ϵ-good hypothesis requires a sample complexity of $\Omega\left(\frac{1-\delta}{\Delta}\right)$.*

Proof Consider an algorithm that tries to distinguish whether a sample $S = \{\langle \vec{x}_i, b_i\rangle \mid i \in [m]\}$ is labeled by the function f or g, where $f, g \in C$. The claim is that no algorithm has a distinguishing probability greater than $m\Delta$.

Formally, let F and G be distributions over $\{0, 1\}^n \times \{-1, +1\}$ that produce $\langle x, f(x \oplus a)\rangle$ and $\langle x, g(x \oplus a)\rangle$, respectively, where x is drawn according to the uniform distribution and a is drawn according to the noise distribution D. Also let F^m and G^m be induced distributions on m independent samples drawn according to F and G, respectively. We must show that there exists no prediction algorithm A (that outputs $\{0, 1\}$) with the property that for all $f, g \in C$,

$$\left|\Pr_{S\sim F^m}[A(S) = 1] - \Pr_{S\sim G^m}[A(S) = 1]\right| > m\Delta.$$

Denote the above absolute difference of probabilities by $\delta_A(f, g)$.

Assume on the contrary that there exists an algorithm A such that $\delta_A(f, g) > m\Delta$. We will use the *hybrid method* or *probability walk* argument [5] to show that there is an algorithm B that can predict whether a labeled example $\langle x, b\rangle$ is drawn from F or from G with an advantage of $\Delta/2$ over random guessing, *i.e.*, the algorithm B satisfies

$$\Pr[B(x, b) \text{ predicts correctly}] > \frac{1}{2} + \frac{\Delta}{2}.$$

This is a contradiction by observing that the optimal Bayes predictor has a $\Delta/2$ advantage (see the appendix for details or [4]). Hence any algorithm needs $m = \Omega((1-\delta)/\Delta)$ samples to distinguish any pair of ϵ-apart functions with probability at least $1-\delta$.

We now elaborate on the hybrid method. Define a sequence of distributions $H_0, H_1, \ldots, H_{n-1}$ over $(\{0,1\}^n \times \{-1,+1\})^m$ where

$$H_i = (\langle x_1, f(x_1 \oplus a)\rangle, \ldots, \langle x_i, f(x_i \oplus a)\rangle, \langle x_{i+1}, g(x_{i+1} \oplus a)\rangle, \ldots, \langle x_m, g(x_m \oplus a)\rangle).$$

Note that $H_0 = G^m$ and $H_m = F^m$. Now let $p(H_i) = \Pr_{S \sim H_i}[A(S) = 1]$. Then

$$\begin{aligned} \delta_A(f,g) &= |p(H_0) - p(H_m)| = \left| \sum_{i=0}^{m-1} (p(H_i) - p(H_{i+1})) \right| \\ &\leq \sum_{i=0}^{m-1} |p(H_i) - p(H_{i+1})| \end{aligned}$$

So if $\delta_A(f,g) > m\Delta$ then there exists $i_0 \in \{0, 1, \ldots, m-1\}$ such that

$$|p(H_{i_0}) - p(H_{i_0+1})| > \Delta.$$

Consider the following algorithm B: on input $\langle x, b\rangle$, run algorithm A on the input

$$Z = (\langle y_1, f(y_1 \oplus a)\rangle, \ldots, \langle y_{i_0}, f(y_{i_0} \oplus a)\rangle, \langle x, b\rangle, \langle y_{i_0+2}, g(y_{i_0+2} \oplus a)\rangle, \ldots, \langle y_m, g(y_m \oplus a)\rangle),$$

where $y_1, \ldots, y_{i_0}, y_{i_0+1}, \ldots, y_m$ are randomly drawn according to the uniform distribution. Assume without loss of generality that $p(H_{i_0}) < p(H_{i_0+1})$. So A is more likely to output 1 when the component (i_0+1) of its input is labeled by f than when it is labeled by g. Then B's prediction is given by

$$B(\langle x, b\rangle) = \begin{cases} f & \text{if } A(Z) = 1 \\ g & \text{otherwise} \end{cases}$$

Let $\mathcal{E}_B$ be the event that B makes a wrong prediction. The probability of event $\mathcal{E}_B$ is given by (assuming a uniform prior on f and g, i.e., chosen uniformly)

$$\begin{aligned} \Pr[\mathcal{E}_B] &= (\Pr[\mathcal{E}_B \mid \langle x, b\rangle \sim F] + \Pr[\mathcal{E}_B \mid \langle x, b\rangle \sim G]) \times \frac{1}{2} \\ &= \frac{1}{2} \times [(1 - p(H_{i_0+1})) + p(H_{i_0})] \\ &= \frac{1}{2} - \frac{1}{2}[p(H_{i_0+1}) - p(H_{i_0})]. \end{aligned}$$

So algorithm B has an advantage strictly greater than $\Delta/2$ in prediction. □

4.1 Near Tight Characterization

In the following we will use Fourier analysis to obtain a nearly tight characterization of the noisy distance quantity $\Delta_D(f,g)$.

Definition 3 *Let $f : \{0,1\}^n \to \{-1,+1\}$ be a Boolean function. Define the transformation T_α, $\alpha \in [0,1]^{\{0,1\}^n}$ on Boolean functions so that $T_\alpha(f) = \sum_{c \in \{0,1\}^n} \alpha_c \hat{f}(c) \chi_c$. Then the* α-attenuated power spectrum *of f is*

$$s_\alpha(f) = ||T_\alpha(f)||_2^2 = \sum_c \alpha_c^2 \hat{f}(c)^2.$$

As it turns out, $\Delta_D(f,g)$ is characterized by the α-attenuated power spectrum of $f-g$ with $\alpha_c = \mathbf{E}_{a\sim D}[\chi_c(a)]$. In particular, define $s_D(f)$ to be $s_\alpha(f)$ with α_c defined in this way. Then we have:

Theorem 3 *Let $f,g : \{0,1\}^n \to \{-1,+1\}$ be Boolean functions and D any probability distribution over $\{0,1\}^n$. Then*

$$s_D(f-g) \le \Delta_D(f,g) \le \sqrt{s_D(f-g)}. \tag{1}$$

Proof Using the fact that $\mathbf{E}[|X|] \le \sqrt{\mathbf{E}[X^2]}$, we get

$$\Delta_D(f,g) \le \frac{1}{2}\sqrt{\mathbf{E}_{x\sim U_n}[(\mathbf{E}_{a\sim D}[f(x\oplus a) - g(x\oplus a)])^2]}.$$

Let $h(x) = (f(x)-g(x))/2$. Then right hand side of the previous expression becomes

$$\sqrt{\mathbf{E}_x[\mathbf{E}_a^2[h(x\oplus a)]]}.$$

We now work with the inner expression $\mathbf{E}_x[\mathbf{E}_a^2[h(x\oplus a)]]$.

$$\begin{aligned}
\mathbf{E}_x[\mathbf{E}_a^2[h(x\oplus a)] &= \mathbf{E}_x[\mathbf{E}_a[h(x\oplus a)]\mathbf{E}_b[h(x\oplus b)] \\
&= \mathbf{E}_{a,b}[\mathbf{E}_x[\sum_{s,t}\hat{h}(s)\hat{h}(t)\chi_s(x\oplus a)\chi_t(x\oplus b)]] \\
&= \mathbf{E}_{a,b}[\sum_{s,t}\hat{h}(s)\hat{h}(t)\chi_s(a)\chi_t(b)\mathbf{E}_x[\chi_s(x)\chi_t(x)] \\
&= \sum_s \hat{h}(s)^2\mathbf{E}_a^2[\chi_s(a)] \\
&= s_D(h).
\end{aligned}$$

Hence we get

$$\Delta_D(f,g) \le \sqrt{s_D(f-g)}.$$

Next, we show a lower bound on $\Delta_D(f,g)$. We note that $0 \le |\mathbf{E}_a[h(x\oplus a)]| \le 1$, since $h \in \{-1,0,+1\}$. Thus

$$\mathbf{E}_x[|\mathbf{E}_a[h(x\oplus a)]|] \ge \mathbf{E}_x[\mathbf{E}_a^2[h(x\oplus a)]] = s_D(h)$$

using the same analysis as in the upper bound. This completes the theorem. □

Define

$$S_D^\epsilon(C) = \min\{s_D(f-g) \mid f,g \in C \textit{ with } \Pr_U[f \ne g] > \epsilon\}.$$

Using this definition with Theorem 3 we have the following inequalities.

Theorem 4 *For any class C and any ϵ we have*

$$S_D^\epsilon(C) \le \Delta_D^\epsilon(C) \le \sqrt{S_D^\epsilon(C)}.$$

Then by Theorem 3 we have the following lower bound.

Theorem 5 *Let C be a concept class with $S_D^\epsilon(C) \le S$. Then any PAC learning algorithm for C under D-distribution attribute noise that outputs an ϵ-good hypothesis with probability at least $1-\delta$ requires a sample complexity of $\Omega\left(\frac{1-\delta}{\sqrt{S}}\right)$.*

We now show that the class of parity functions is not PAC learnable under the uniform distribution with D-noise for almost every noise distribution D.

Theorem 6 *Let D be a distribution such that $\max_x D(x)$ is superpolynomially small (or $1/\omega(poly(n))$). Then the set of parity functions is not PAC-learnable under D-distribution noise.*

Proof Notice that for any two distinct parity functions f and g we have $\Pr[f \neq g] = 1/2$. Since f and g are parity functions, $s_D(f - g) = s_D(f) + s_D(g)$, and it is enough to find two distinct parity functions f and g with superpolynomially small $s_D(f)$ and $s_D(g)$.

Consider $\mathbf{E}_c[s_D(\chi_c)]$ where c is over the uniform distribution. We have

$$\begin{aligned}
\mathbf{E}_c[s_D(\chi_c)] &= \mathbf{E}_c[\alpha_c^2] \\
&= \mathbf{E}_c[\mathbf{E}_{a\sim D}[\chi_c(a)]\mathbf{E}_{b\sim D}[\chi_c(b)]] \\
&= \mathbf{E}_c\mathbf{E}_{a,b}[\chi_c(a \oplus b)] \\
&= \mathbf{E}_{a,b}\mathbf{E}_c[\chi_{a\oplus b}(c)] \\
&= \mathbf{E}_{a,b} I[a = b] \\
&\leq \max_x D(x).
\end{aligned}$$

where $I[a = b] = 1$ if $a = b$ and 0 otherwise. Therefore, because $s_D(f)$ is nonnegative for all D and Boolean f, only a superpolynomially small fraction of parity functions χ_c can be inverse polynomially large if $D(x)$ is superpolynomially small for all x. So there are at least two parity functions f and g for which both $s_D(f)$ and $s_D(g)$ are superpolynomially small. □

Finally, it should be noted that Theorem 5 is only a hardness result for strong PAC learnability. As an example of a class that can be weakly learned in spite of arbitrary and unknown random attribute noise, consider monotone Boolean functions. Blum, Burch, and Langford [1] have shown that every monotone Boolean function is weakly approximated with respect to the uniform distribution by either one of the two constant functions or by the majority function. Since random attribute noise alone does not change the label of a function, it is easy to test noisy examples to see if one of the constants functions is a weak approximator to a montone function f; if not, majority is.

5 Upper Bounds

In this section we consider a certain type of Fourier-based learning algorithm which we will call *LMN-style*. The LMN-style algorithm was introduced by Linial, Mansour, and Nisan [9], who showed that the class AC^0 of polynomial-size, constant depth circuits is PAC learnable with respect to the uniform distribution in quasipolynomial (roughly $n^{\text{polylog}(n)}$) time. The key to their result was analyzing the Fourier properties of AC^0 to show that for every AC^0 function f, the sum of the squares of the Fourier coefficients of degree polylog(n) or less is nearly 1. They then showed that the function

$$h(x) = \text{sign}\left(\sum_{|a|\leq \text{polylog}(n)} \hat{f}(a)\chi_a(x)\right)$$

is a good approximator to the target function f. Finally, it follows from standard Chernoff bounds that all of these Fourier coefficients can be closely approximated by sampling from a uniform-distribution example oracle, with sample size and running time dominated by $n^{\text{polylog}(n)}$.

An LMN-style algorithm, then, given $\epsilon > 0$, consists of estimating —for every n-bit index in a set T_ϵ—Fourier coefficients, with the guarantee that the sum of the squares of these coefficients is nearly 1. How near 1 this sum must be, and therefore how large the set T_ϵ must be, depends on the value of the PAC accuracy parameter ϵ, which is why ϵ subscripts T. For example, in the case of Linial *et al.*'s algorithm for AC^0, the Hamming weight of the Fourier indices grows as ϵ approaches 0. The hypothesis resulting from an LMN-style algorithm will be of the form

$$h(x) = \text{sign}\left(\sum_{a \in T_\epsilon} \tilde{f}(a)\chi_a(x)\right),$$

where $\tilde{f}(a)$ represents an estimate of the Fourier coefficient $\hat{f}(a)$.

In this section we show that if there is an LMN-style algorithm for learning a class of functions C, then C is PAC-learnable under any (D, R)-noise in time polynomial in $|T|$, $1/(1-2\eta)$, and $1/\Delta$, where η is the expectation of the noise rate in the label. Since $1/\Delta$ is a lower bound for PAC-learning with D-distribution noise and $1/(1-2\eta)$ is a lower bound for learning with label noise [11], our result is tight (up to polynomial factors). Before we formally state the result, we recall the following version of Chernoff bounds.

Lemma 7 *(Chernoff bounds) Let X_i, $1 \le i \le m$, be independent, identically distributed random variables, where $\mathbf{E}[X_i] = \mu$ and $|X_i| \le B$. Then*

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mu\right| > \gamma\right] \le \delta,$$

whenever $m \ge (2B^2/\gamma^2)\ln(2/\delta)$.

Theorem 8 *Let C be a class of Boolean functions and suppose that C is learnable with respect to the uniform distribution by an LMN-style algorithm using index set T_ϵ. Then for every ϵ such that the set of parity functions indexed by T_ϵ is a subset of C, C is learnable with respect to the uniform distribution and with any known (D, R)-noise in time polynomial in $1/\epsilon, 1/\delta, 1/(1-2\eta), |T_{O(\epsilon)}|$, and $1/\Delta_D^\epsilon$, where η is the expectation of the classification noise rate.*

Proof Let $\Delta = \Delta_D^\epsilon(C)$ and $T = T_\epsilon$. First we show that there is at most one $c \in T$ so that $|\alpha_c| < \Delta/2$. Suppose that there are two α_{c_1} and α_{c_2} such that $|\alpha_{c_1}| < \Delta/2$, $|\alpha_{c_2}| < \Delta/2$. Then

$$\Delta^2 \le s_D(\chi_{c_1} - \chi_{c_2}) = s_D(\chi_{c_1}) + s_D(\chi_{c_2}) = \alpha_{c_1}^2 + \alpha_{c_2}^2 \le \Delta^2/2,$$

which is a contradiction. Let $c_0 \in T$, if it exists, be such that $|\alpha_{c_0}| < \Delta/2$.

Now to find the coefficient of $c \in T$ we take a sample $S = \{(x^i \oplus a^i, f(x^i)b^i) \mid 1 \le i \le m\}$, m to be determined later, (since the function f is $\{+1, -1\}$-valued, we choose $b^i \in \{-1, +1\}$, so XOR becomes multiplication) and estimate the expectation $\mu_c = \mathbf{E}_{x,a,b}[f(x)b\chi_c(x \oplus a)]$. So, for a fixed $c \in T$, let $\beta_c = \frac{1}{m}\sum_{i=1}^{m} \chi_c(x^i \oplus a^i)f(x^i)b^i$ be the estimate for the above expectation. Note that

$$\mu_c = \mathbf{E}_{x,a,b}[f(x)b\chi_c(x \oplus a)] = \mathbf{E}_b[b]\mathbf{E}_x[f(x)\chi_c(x)]\mathbf{E}_a[\chi_c(a)] = (1-2\eta)\hat{f}(c)\alpha_c.$$

Because we are assuming that D and R are known, the factors of $(1-2\eta)$ and α_c are known[1] and can easily be eliminated for those c's such that $\alpha_c \ge \Delta/2$. Thus, for such c's, a good estimate of

[1] We assume that η and the α_c's are exactly known. More tedious error analysis could be done to eliminate this assumption.

the above expectation gives a good estimate of the Fourier coefficient $\hat{f}(c)$. Using Chernoff bounds (c.f. Lemma 7), we can estimate this expectation with a sample size (and time complexity, with polynomial blowup) of

$$m = \frac{32|T|}{\epsilon(1-2\eta)^2\Delta^2} \ln \frac{4|T|}{\delta}$$

(i.e., letting $B = 1$, $\gamma = \sqrt{\epsilon/(2|T|)}(1-2\eta)\Delta/2$, and using $\delta/(2|T|)$ as the confidence). This will guarantee that with probability at least $1 - \delta/2$, $|\beta_c - \mu_c| < \sqrt{\frac{\epsilon}{2|T|}}(1-2\eta)|\alpha_c|$ holds, for all $c \in T$ (except maybe c_0); which in turn, implies that $|\hat{\beta}_c - \hat{f}(c)| < \sqrt{\epsilon/(2|T|)}$, where

$$\hat{\beta}_c = \frac{\beta_c}{(1-2\eta)\alpha_c}$$

is the estimate for $\hat{f}(c)$. This shows that the set L of all of the relevant coefficients indexed by T, except maybe c_0, can be estimated in time polynomial in $|T|$, $1/\Delta$, and $1/\epsilon$.

To estimate the coefficient of $\hat{f}(c_0)$ (only an approximation of this coefficient is required) we can use the fact that the sum of all of the relevant coefficients indexed by T should be nearly 1. If after we have found estimates for all the coefficients for indices other than c_0 the resulting sum of squares is noticeably less than 1, then we know that $\hat{f}^2(c_0)$ must be approximately the difference. Thus we can estimate $\hat{f}(c_0)$ as well, although perhaps not quite as accurately as the other coefficients. However, given the form of the LMN-style hypothesis h, it can be argued that this estimate can be made sufficiently close to ensure that h is an ϵ-approximator. More formally, let $\tau = \sum_{c\in L} \hat{\beta}_c^2$ be the estimate of $\sum_{c\in L} \hat{f}(c)^2$. With probability at least $1 - \delta/2$, we know that $\sum_{c\in L}(\beta_c - \hat{f}(c))^2 < \epsilon/2$. Thus

$$1 - \epsilon \le \sum_{c\in T} \hat{f}(c)^2 \le \hat{f}(c_0)^2 + \tau + \frac{\epsilon}{2},$$

and hence $\tau + \hat{f}(c_0)^2 \ge 1 - 3\epsilon/2$. If $\tau \ge 1 - 2\epsilon$ then we can ignore the contribution of $\hat{f}(c_0)$ and we set $\hat{\beta}_{c_0} = 0$.

If $\tau < 1 - 2\epsilon$, then $\hat{f}(c_0)^2 \ge \epsilon/2$ (or $|\hat{f}(c_0)| \ge \sqrt{\epsilon/2}$). To discover the sign of $\hat{f}(c_0)$, we could estimate this using a Chernoff sample of size $m = \frac{8}{\epsilon\alpha_{c_0}^2(1-2\eta)^2}\ln(\frac{4}{\delta})$ which would guarantee that with probability at least $1 - \delta/2$, the estimate for $\mathbf{E}_{x,a,b}[\chi_{c_0}(x \oplus a)f(x)b]$ is within $\frac{1}{2}\sqrt{\epsilon/2}|\alpha_{c_0}|(1-2\eta)$ from the true mean; this implies that an estimate for $\hat{f}(c_0)$ is within $\frac{1}{2}\sqrt{\epsilon/2}$ of the true value. So we set $\hat{\beta}_{c_0}$ accordingly. So the final hypothesis is $h(x) = sign(\sum_{c\in T}\hat{\beta}_c\chi_c(x))$. By the standard LMN analysis, we get

Flawed: m may not be poly in $1/\Delta$!

$$\begin{aligned}
\Pr_x[sign(h(x)) \ne f(x)] &\le \frac{1}{4}\mathbf{E}_x[(f(x) - h(x))^2] = \frac{1}{4}\sum_c(\hat{f}(c) - \hat{h}(c))^2 \\
&= \frac{1}{4}\left[\sum_{c\notin T_\epsilon}\hat{f}(c)^2 + \sum_{c\in T_\epsilon}(\hat{f}(c) - \hat{\beta}_c)^2\right] \\
&< \frac{\epsilon}{4} + \frac{\epsilon}{8} + \frac{1}{4}(\hat{f}(c_0) - \hat{\beta}_{c_0})^2.
\end{aligned}$$

So the true error rate (without attribute and classification noise) depends on the accuracy of our estimate of $\hat{f}(c_0)$. □

For the LMN-style algorithm for AC^0, as long as $1/\epsilon = O(n^{\mathrm{polylog}(n)})$, the parity functions indexed by T_ϵ are in AC^0 (by results in Linial *et al.* [9] and the fact that parity on polylogarithmic bits can be computed in AC^0). This gives us immediately the following result.

Theorem 9 *For $1/\epsilon = O(n^{polylog(n)})$, the class AC^0 of constant depth, polynomial size circuits is learnable under the uniform distribution with any known (D, R)-noise in time dominated by*

$$n^{poly(\log n)} poly(1/\Delta_D^\epsilon).$$

As a specific example of the application of this theorem, we claim that if the attribute noise rate on all attributes is independent with rate $O(1/\mathrm{polylog}(n))$ for each attribute (but possibly different values for each) then there is a learning algorithm for AC^0 with time dominated by $n^{\mathrm{polylog}(n)}$. To see this, recall that the hypothesis in an LMN-style algorithm is formed using only (estimates of) coefficients indexed by T_ϵ, and that for AC^0 all of these indices have polylogarithmic Hamming weight. Furthermore, based on results of Linial *et al.* [9], if f and g are Boolean functions such that

$$\Pr[f \neq g] > \epsilon = \Omega(1/n^{\mathrm{polylog}(n)})$$

then the difference $\hat{f}(c) - \hat{g}(c)$ must be at least $1/n^{\mathrm{polylog}(n)}$ large for at least one of the coefficients indexed by T_ϵ. But then $s_D(f-g) = \sum_c \alpha_c^2 (\widehat{f-g})^2(c) = \sum_c \alpha_c^2 (\hat{f}(c) - \hat{g}(c))^2$ (the final equality follows by linearity of the Fourier transform) will be inverse quasipolynomially large as along as $\alpha_c = \mathbf{E}_{a\sim D}[\chi_c(a)]$ is inverse quasipolynomial for all c in T_ϵ. But then a simple probabilistic analysis shows that in fact all of these α_c will be sufficiently large as long as $|c|$ is polylogarithmic in n. In particular,

$$\begin{aligned}
\alpha_c &= \mathbf{E}_{a\sim D}[(-1)^{\sum_{i=1}^n a_i c_i}] \\
&= \prod_{i=1}^n \mathbf{E}_{a_i \sim D_i}[(-1)^{a_i c_i}], \text{ } D \text{ is a product distribution} \\
&= \prod_{i\in c}(1-2p_i) \\
&> (1 - 1/poly(\log n))^{|c|}, \text{ since } (\forall i)\ p_i < 1/\mathrm{polylog}(n) \\
&> 1/n^{\mathrm{polylog}(n)}, \quad \text{since } |c| \leq \mathrm{polylog}(n)
\end{aligned}$$

Therefore, Δ_D^ϵ is inverse quasipolynomially large, and our claim follows by the theorem.

However, we note that learning even very simple classes of functions can be hard if the attribute noise D is unknown. Consider the problem of learning the monomials x_i and $\overline{x_i}$ under the no-noise (noise rate equals 0) and full-noise (noise rate equals 1) distributions.

References

[1] Avrim Blum, Carl Burch, John Langford. On Learning Monotone Boolean Functions. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998.

[2] Itai Benjamini, Gil Kalai, Oded Schramm. Noise Sensitivity of Boolean Functions and Applications to Percolation. *Inst. Hautes Études Sci. Publ. Math.*, **90**:5-43, 1999.

[3] Scott Decatur, Rossario Gennaro. On Learning from Noisy and Incomplete Examples. In *Proceedings of the 8th Annual ACM Conference on Computational Learning Theory*, 353–360, 1995.

[4] Luc Devroye, László Györfi, Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.

[5] Shafi Goldwasser, Silvio Micali. Probabilistic Encryption. In *Journal of Computer and System Sciences*, 28(2):270-299, 1984.

[6] Sally Goldman, Robert Sloan. Can PAC Learning Algorithms Tolerate Random Attribute Noise? In *Algorithmica*, 14(1):70-84, 1995.

[7] Jeff Kahn, Gil Kalai, Nathan Linial. *The Influence of Variables on Boolean Functions.* In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, 68-80, 1988.

[8] M. Kearns, M. Li. Learning in the Presence of Malicious Errors. In *SIAM Journal on Computing*, **22**(4):807–837, 1993.

[9] Nathan Linial, Yishay Mansour, Noam Nisan. Constant Depth Circuits, AC^0 Circuits, and Learnability. In *Journal of the ACM*, **40**(3):607-620, 1993.

[10] George Shackelford, Dennis Volper. Learning k-DNF with Noise in the Attributes. In *Proceedings of the 1988 Workshop on Computational Learning Theory*, 97-103, 1988.

[11] Hans Ulrich Simon. General Bounds on the Number of Examples Needed for Learning Probabilistic Concepts. In *Proceedings of the 6th Annual ACM Workshop on Computational Learning Theory*, pages 402-411, 1993.

Appendix A

We prove the connection between $\Delta_D(f,g)$ and the optimal Bayes' loss. Consider a fixed x and a predictor that is given a bit b that is either $f(x \oplus a)$ or $g(x \oplus a)$, for a randomly chosen a from a simple product distribution with rate p, and must predict whether b is labeled according to f or g. For $b \in \{-1,+1\}$, let $p_f^b(x) = \Pr[f(x \oplus a) = b]$ and $p_g^b(x) = \Pr[g(x \oplus a) = b]$. Then, by standard Bayes theory, the Bayes predictor B, whose decision is given by

$$B(\langle x, b\rangle) = \begin{cases} f & \text{if } p_f^b(x) > p_g^b(x) \\ g & \text{otherwise} \end{cases}$$

achieves a discrete loss of $1/2 - |p_f^b(x) - p_g^b(x)|/2$, for the given fixed x, and this quantity is the smallest achievable loss over all predictors. Here we are assuming uniform priors on f and g. Note that $|p_f^b(x) - p_g^b(x)| = |\mathbf{E}_a[(f-g)(x \oplus a)]|/2$. Now allowing x to vary according to some distribution D, the expected loss of the Bayes predictor is

$$L^* = \frac{1}{2} - \frac{1}{2}\mathbf{E}_x[|p_f^b(x) - p_g^b(x)|] = \frac{1}{2} - \frac{1}{2}\Delta_D(f,g).$$

By the same token, there is no predictor that has smaller expected loss than L^*.