On Learning Conjunctions with Malicious Noise*

Yishay Mansour Michal Parnas
Computer Science Dept.
Tel-Aviv University

Abstract

We show how to learn monomials in the presence of malicious noise, when the underlined distribution is a product distribution. We show that our results apply not only to product distributions but to a wide class of distributions.

1 Introduction

The Probably Approximately Correct (PAC) Learning model [Val84] has been the most widely studied model in Computational Learning Theory. From the beginning of the field, it has been considered of the utmost importance to develop algorithms that are tolerant to noise. A variety of noise models were studied. The most notable ones are the random classification noise, where the classification of the examples may be flipped randomly, and the malicious noise model, where there is no assumptions about the way in which the corrupted examples are generated. A general framework to handle noise is the Statistical Query model [K93], which was originally aimed at the random classification noise model, and later extended to the malicious noise model [D93]. (See [Slo88] for a discussion of various noise models).

In this paper we study the *malicious noise* model. The model is roughly the following: there is a noise parameter η, such that with probability $1-\eta$ the learner receives an example from the "true" distribution, which is labeled correctly, and with probability η the learner gets a "noisy" example, and in this case the labeling may be chosen adversarialy. It is not hard to see that from an information theoretic point of view, there is no "inherent" problem. In fact, the hypothesis that has the lowest observed error would be a good approximation of the target hypothesis [AL88]. As in many other learning tasks, the main open problem is how to perform such a task efficiently, namely to give a polynomial time algorithm. An interesting parameter is the noise rate that can be tolerated, so that learning is still possible in polynomial time.

Malicious noise has another motivation, other than the obvious motivations for noise. We can view the approximation of a target function by a function from the desired concept class, as a malicious noise. Consider the following scenario: suppose the target function is some fixed unknown function (say g). We have no guarantee that the target function does belong to the desired class (e.g. monomials), but we believe that it can be well approximated by a function from that class (say f). Now we can view the learning scenario as follows: the target function is f, and the examples drawn are labeled by g. However the difference (i.e., $f-g$) is attributed to noise. Overcoming malicious noise implies that we are able to approximate f, and thus approximate g.

One of the simplest concept classes is that of monomials, which are a conjunction of literals over n variables $\{z_1, z_2, ..., z_n\}$. This concept class has been widely studied in Computational Learning Theory, and both efficient and simple learning algorithms where developed for it in various learning models (e.g. PAC, online, Equivalence Queries). Unfortunately, it seems a much more involved task to develop noise tolerant learning algorithms. The problem of learning monomials with malicious noise was already discussed in one of the pioneering papers in computational learning [Val85], which gives a learning algorithm that tolerates a noise rate of $O(\epsilon/n)$. Later [KL88] improved the result and showed an algorithm that tolerates a noise rate of $O(\frac{\epsilon}{n}\ln\frac{n}{\epsilon})$ for monomials over n variables, and $O(\epsilon/\ell\log\frac{\ell\ln n}{\epsilon})$, when the target monomial has at most ℓ literals. The latter result was improved in [D93] to $O(\epsilon/\ell\log\frac{1}{\epsilon})$.

We develop an algorithm that is able to tolerate a noise rate of $O(\epsilon^{5/3}/n^{2/3})$, which is a substantial improvement over the previous results. Unfortunately, our algorithm requires stronger assumptions about the underlined distribution. We prove our algorithm for the case that the underlined distribution is a product distribution. We show how to extend it to a distribution in which the probability that a variable is zero, is at most α, and then we can tolerate a noise rate

*This research was supported in part by The Israel Science Foundation administered, by The Israel Academy of Science and Humanities, by a grant of the Israeli Ministry of Science and Technology, and by a grant from Tel-Aviv University.

of $O(\frac{\epsilon^3}{\alpha^2 n^2})$. (This is an improvement over previous results for α in the range ϵ/n to $\epsilon/\sqrt{n}$.) Also, we show that a wide class of distributions has the conditions required by our algorithm. (Loosely speaking, we require that if the probability that the function is one and the probability of a literal to be zero, are both substantial, then the probability of the joint event would be non-negligible.) We hope that the techniques developed might be of interest for the case of a general distribution as well.

This paper is organized as follows. Section 2 describes the learning model. The learning algorithm appears in Section 3. The correctness of the algorithm for product distributions appears in Section 4. Extensions to other distributions are described in Section 5.

2 The Learning Model

Let $X = \{0,1\}^n$ be the *instance space*. A *concept* c over X is a boolean mapping $c : X \rightarrow \{0,1\}$, where $c(x) = 1$ indicates that x is a *positive example* of c, and $c(x) = 0$ indicates that x is a *negative example* of c. A collection C of concepts over X, will be called a *concept class*.

Let D be a probability distribution over X. Given a target concept $c \in C$ we extend D to a distribution over $X \times \{0,1\}$, such that it generates pairs $< x, c(x) >$, where $x \in X$ is generated according to D. Let D_N be a distribution over $X \times \{0,1\}$, which generates pairs $< x, b >$. D_N will be called the *noise distribution*. Note that we do not assume any relationship between x and b in the pairs $< x, b >$ that D_N generates.

Given a concept $c \in C$, we define a noisy oracle with noise rate $0 \leq \eta < 1/2$, that with probability $1-\eta$ draws an example x from D, and labels it by $c(x)$, and with probability η draws examples from D_N. That is, the oracle draws an example according to a probability distribution D_η, which is defined as follows:

$$D_\eta(< x, b >) = (1-\eta)D(< x, b >) + \eta D_N(< x, b >),$$

where $D(< x, b >) = D(x)$ if $b = c(x)$ and $D(< x, b >) = 0$ otherwise. We also refer to $D_\eta(x) = D_\eta(< x, 0 >) + D_\eta(< x, 1 >)$.

When the example $< x, b >$ is drawn from D, we are guaranteed that $b = c(x)$, where c is the (unknown) target function, but when the example is drawn from D_η, we do not have such a guarantee.

Definition 2.1 *Let C be a concept class over X, H be an hypothesis space over X, and let $\mathcal{D}$ be a class of distributions over X. The class C is PAC learnable with noise rate η, $0 \leq \eta < 1/2$, with respect to the class of distributions $\mathcal{D}$, if there exists an algorithm A with the following properties:*
For any concept $c \in C$, for any distribution $D \in \mathcal{D}$, for any noise distribution D_N over $X \times \{0,1\}$, and for any $0 < \epsilon, \delta < 1$, the algorithm A samples examples according to probability D_η, and outputs an hypothesis $h \in H$, such that with probability at least $1-\delta$,

$$error(h) = Pr_{D_\eta}[c(x) \neq h(x)] \leq \epsilon.$$

If A runs in time polynomial in n, $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ and $\frac{1}{1-2\eta}$, then C is efficiently PAC learnable with noise rate η with respect to the class of distributions $\mathcal{D}$.

Note that we made absolutely no assumptions about D_N, and thus the noise may generate examples in any fashion. The only parameter that controls the noise is the noise rate η, which is the probability that we sample from D_N.

Denote by C_n (C_n^+) the class of monomials (monotone monomials), i.e. the class of concepts over $X_n = \{0,1\}^n$, which represent boolean conjunctions of literals (variables) over the boolean variables $\{z_1, ..., z_n\}$. For example, $c = z_1 \wedge z_3$ is a monotone monomial over the set of variables $\{z_1, z_2, z_3\}$, and $x = (1,0,1) \in X_3$ is a positive example for c.

For any vector $x = (x_1, ..., x_n) \in X_n$, define random variables $Y_i(x)$, $1 \leq i \leq n$:

$$Y_i(x) = \begin{cases} 1 & \text{if } x_i = 0 \\ 0 & \text{if } x_i = 1 \end{cases}$$

Denote by $Z_S(x)$ the number of zeros in a vector $x \in X_n$, restricted to a set S of indices. That is,

$$Z_S(x) = \sum_{i \in S} Y_i(x).$$

3 An Algorithm for the Product Distribution

We show an algorithm for learning monotone monomials $c \in C_n^+$. This algorithm can be modified to learn arbitrary monomials $c \in C_n$, by using the following technique due to Littlestone [Lit88]. The algorithm for learning arbitrary monomials will find a positive example y, and modify each example $< x, b >$ to $< x \oplus \bar{y}, b >$. (We need that y is a "true" positive example. If there are many positive examples, then with high probability there is one "true" positive example. We run the algorithm in parallel with each positive example, and choose at the end the hypothesis with the lowest observed error rate.)

The algorithm for learning monotone monomials will proceed as follows. Let $\mathcal{D}$ be the class of product distributions, and let $D \in \mathcal{D}$. Then, since D is a product distribution, for any concept $c \in C_n^+$ and variable $z_i \notin c$,

$$Pr_D[c(x) = 1, x_i = 0] = Pr_D[c(x) = 1] Pr_D[x_i = 0].$$

Let,

$$\beta_i = Pr_{D_\eta}[x_i = 0].$$

For every variable z_i, estimate β_i, by drawing N examples according to the distribution D_η. Let $\hat{\beta}_i$ be this estimate. Partition the variables into three sets, according to the value of $\hat{\beta}_i$ (Note that the partition depends on the examples drawn, but once the sample is fixed, the partition is also fixed.)

$$S_1 = \{z_i | \hat{\beta}_i < \frac{\epsilon}{16n}\},$$

$$S_2 = \{z_i | \frac{\epsilon}{16n} \leq \hat{\beta}_i < 2\alpha\},$$

$$S_3 = \{z_i | \hat{\beta}_i \geq 2\alpha\}.$$

(The constant α would be fixed later to be $\Theta(\epsilon^{2/3}/n^{2/3})$.)

Let $T = \frac{24|S_2|\alpha}{\epsilon}$, and denote by $D_{\eta,T}$ the probability distribution D_η restricted to examples $x \in X_n$ with at most T zeros in the set S_2. That is,

$$D_{\eta,T}(< x, b >) = Pr_{D_\eta}[< x, b > | Z_{S_2}(x) \leq T].$$

Define $D_{N,T}$ in a similar way to be,

$$D_{N,T}(< x, b >) = Pr_{D_N}[< x, b > | Z_{S_2}(x) \leq T].$$

Define:

$$\gamma_i = Pr_{D_\eta}[x_i = 0, c(x) = 1],$$

$$\lambda_i = Pr_{D_{\eta,T}}[x_i = 0, c(x) = 1].$$

Again, estimate γ_i and λ_i, by drawing N examples according to the distribution D_η, and let $\hat{\gamma}_i$ and $\hat{\lambda}_i$ be these estimates.

Initially the hypothesis h includes all possible n variables, i.e. $h(z) = \wedge_{i=1}^n z_i$. We delete a variable z_i from h according to the following three rules:

1. If $z_i \in S_1$, then delete z_i from h.
2. If $z_i \in S_2$ and $\hat{\lambda}_i \geq \frac{\epsilon}{16n}$, then delete z_i from h.
3. If $z_i \in S_3$ and $\hat{\gamma}_i > \frac{\alpha\epsilon}{2}$, then delete z_i from h.

The hypothesis of the learning algorithm is the resulting h. In other words,

$$h(z) = \wedge_{z_i \in S_2 : \hat{\lambda}_i < \frac{\epsilon}{16n}} z_i \wedge_{z_j \in S_3 : \hat{\gamma}_j \leq \frac{\alpha\epsilon}{2}} z_j$$

In the analysis we regard h (and also the target concept c) as a set, this should be interpreted as the set of variables such that the hypothesis h (concept c) is their conjunction.

4 Proof of Correctness

Partition the variables in the concept c and in the hypothesis h into 3 disjoint concepts, c_1, c_2, c_3, and h_1, h_2, h_3, respectively, such that $c_i = S_i \cap c$ and $h_i = S_i \cap h$, for $i = 1, 2, 3$, and:

$$c = c_1 \wedge c_2 \wedge c_3, \quad h = h_1 \wedge h_2 \wedge h_3,$$

Since $h_1 = \emptyset$, we regard h_1 as being the constant function 1, that is, $h_1(x) = 1$ for all x. We bound the error by,

$$\begin{aligned}
&Pr_{D_\eta}[c(x) \neq h(x)] \leq \\
&\quad Pr_{D_\eta}[c_1(x) \neq h_1(x)] + Pr_{D_\eta}[c(x) = 1, h_2(x) = 0] + \\
&\quad Pr_{D_\eta}[c_2(x) = 0, h(x) = 1] + Pr_{D_\eta}[c_3(x) \neq h_3(x)],
\end{aligned}$$

therefore, it is sufficient to bound each one of the terms in this sum. Note that rather then using $Pr_{D_\eta}[c_2(x) \neq h_2(x)]$, we do a more refine bound, comparing $h_2(x) = 0$ to $c(x) = 1$ and $c_2(x) = 0$ to $h(x) = 1$. However, note that the sum of the four probabilities bounds the error between c and h in a correct way.

Lemma 4.1 (Chernoff Bounds) *Let $X_1, X_2, ..., X_N$ be N independent Bernoulli trials, each with probability of success $p = E(X_i)$. Let $\hat{p} = \sum_i X_i/N$. If $N \geq \frac{1}{\epsilon^2} ln \frac{1}{\delta}$, then with probability $1 - \delta$,*

$$Pr[|\hat{p} - p| \geq \epsilon] \leq \delta.$$

We start with the following technical lemma, which shows that with high probability, our approximations of β_i, λ_i and γ_i are accurate. The following follows from Lemma 4.1.

Lemma 4.2 *Given a sample of size $N = \Omega(\frac{n^2}{\alpha^2\epsilon^2} ln \frac{n}{\delta})$, then with probability $1 - \delta$, for every i,*

$$|\beta_i - \hat{\beta}_i| \leq \frac{\alpha\epsilon}{16n}, \quad |\lambda_i - \hat{\lambda}_i| \leq \frac{\epsilon}{32n}, \quad \textit{and} \quad |\gamma_i - \hat{\gamma}_i| \leq \frac{\alpha\epsilon}{16n}$$

First we bound the error due to variables in S_1.

Lemma 4.3 *With probability $1 - \delta$, $Pr_{D_\eta}[c_1(x) \neq h_1(x)] < \frac{\epsilon}{8}$.*

Proof: By the first step of the algorithm, $h_1 = \emptyset$, and as stated we regard h_1 as being equal to the constant function 1. In order to bound $Pr_{D_\eta}[c_1(x) \neq h_1(x)]$, we bound $Pr_{D_\eta}[c_1(x) = 0]$, since $h_1(x) \equiv 1$.

The event $c_1(x) = 0$ can happen only if there exists a variable z_i, such that $z_i \in c_1$ and $x_i = 0$. Since $z_i \in S_1$ then $\hat{\beta}_i < \frac{\epsilon}{16n}$. Therefore, by Lemma 4.2, with probability $1 - \delta$,

$$Pr_{D_\eta}[x_i = 0] = \beta_i \leq \hat{\beta}_i + \frac{\alpha\epsilon}{16n} < \frac{\epsilon}{8n}.$$

We simply sum over all such variables z_i:

$$Pr_{D_\eta}[c_1(x)=0] \le \sum_{z_i \in S_1} Pr_{D_\eta}[x_i=0] < n\frac{\epsilon}{8n} = \frac{\epsilon}{8}.$$

Hence, with probability $1-\delta$, $Pr_{D_\eta}[c_1(x)=0, h_1(x)=1] < \frac{\epsilon}{8}$. □

The following lemma shows that when we restrict the distribution to examples x with $Z_{S_2}(x) \le T$, then not many variables can have a high probability of being set to zero.

Lemma 4.4 *For any* S_2,

$$\sum_{i \in S_2} Pr_{D_{N,T}}[x_i=0] \le T.$$

Proof: Note that $\sum_x D_{N,T}(x) \le 1$, and that if $D_{N,T}(x) > 0$ then $Z_{S_2}(x) \le T$. Therefore,

$$\begin{aligned}\sum_{z_i \in S_2} Pr_{D_{N,T}}[x_i=0] &= \sum_{z_i \in S_2}\sum_x Y_i(x) D_{N,T}(x)\\ &= \sum_x \sum_{z_i \in S_2} Y_i(x) D_{N,T}(x)\\ &= \sum_x Z_{S_2}(x) D_{N,T}(x) \le T,\end{aligned}$$

which completes the proof of the lemma. □

When considering a variable that appears in the target monomial, it is clearly the case that when the variable is zero the monomial is also zero. The following lemma bounds the probability that the malicious noise changes this situation.

Lemma 4.5 *Let* $z_i \in c$. *Then,* $Pr_{D_{\eta,T}}[c(x)=1, x_i=0] \le \eta Pr_{D_{N,T}}[x_i=0]$.

Proof: By the definition of $D_{\eta,T}$,

$$\begin{aligned}&Pr_{D_{\eta,T}}[c(x)=1, x_i=0] =\\ &\quad \eta Pr_{D_{N,T}}[x_i=0] Pr_{D_{N,T}}[c(x)=1|x_i=0] +\\ &\quad (1-\eta) Pr_{D_T}[x_i=0] Pr_{D_T}[c(x)=1|x_i=0].\end{aligned}$$

Since $Pr_{D_T}[c(x)=1|x_i=0]=0$ when $z_i \in c$, we have that,

$$\begin{aligned}&Pr_{D_{\eta,T}}[c(x)=1, x_i=0] =\\ &\quad \eta Pr_{D_{N,T}}[x_i=0] Pr_{D_{N,T}}[c(x)=1|x_i=0]\\ &\quad \le \eta Pr_{D_{N,T}}[x_i=0].\end{aligned}$$

□

The following lemma establishes the bound on the false-negative error due to h_2. Note that we compare h_2 to c and not to c_2.

Lemma 4.6 *With probability* $1-\delta$,

$$Pr_{D_\eta}[c(x)=1, h_2(x)=0] \le \frac{\epsilon}{4}.$$

Proof: Assume that $c(x)=1$ and $h_2(x)=0$. There must exist a variable z_i, such that $z_i \notin c$, $z_i \in h_2$, and $x_i=0$. However, if $z_i \in h_2$, then $\hat{\lambda}_i < \frac{\epsilon}{16n}$. Therefore, by Lemma 4.2, with probability $1-\delta$,

$$Pr_{D_{\eta,T}}[x_i=0, c(x)=1] = \lambda_i \le \hat{\lambda}_i + \frac{\epsilon}{32n} < \frac{\epsilon}{8n}.$$

Therefore, if we sum over all variables z_i, then with probability $1-\delta$,

$$\begin{aligned}q_1 &= Pr_{D_{\eta,T}}[c(x)=1, h_2(x)=0]\\ &\le \sum_{i:\hat{\lambda}_i < \frac{\epsilon}{16n}} Pr_{D_{\eta,T}}[c(x)=1, x_i=0]\\ &< n\frac{\epsilon}{8n} = \frac{\epsilon}{8}.\end{aligned}$$

Also, since $z_i \in h_2$, then $\hat{\beta}_i < 2\alpha$. Therefore, by Lemma 4.2, with probability $1-\delta$,

$$\beta_i = Pr_{D_\eta}[x_i=0] \le \hat{\beta}_i + \frac{\alpha\epsilon}{16n} < 3\alpha.$$

Therefore, the expected number of zeros in an example x, restricted to indices in the set S_2, according to the distribution D_η, is at most $3\alpha|S_2|$. Thus, by the Markov inequality, the probability that the number of zeros in an example x, restricted to S_2, will deviate by much from the expected number of zeros, is:

$$q_2 = Pr_{D_\eta}[Z_{S_2}(x) > T] = Pr_{D_\eta}[Z_{S_2}(x) > \frac{24\alpha|S_2|}{\epsilon}] < \frac{\epsilon}{8}.$$

Therefore,

$$Pr_{D_\eta}[c(x)=1, h_2(x)=0] \le q_1 + q_2 < \frac{\epsilon}{8} + \frac{\epsilon}{8} = \frac{\epsilon}{4}.$$

□

The next lemma bounds false-positive errors due to variables in S_2.

Lemma 4.7 *If the noise rate is* $\eta = \frac{\epsilon^2}{384\alpha n T}$, *then with probability* $1-\delta$,

$$Pr_{D_\eta}[c_2(x)=0, h(x)=1] \le \frac{\epsilon}{4}.$$

Proof: Assume that $c_2(x)=0$ and $h(x)=1$. There must exist a variable z_i, such that $z_i \in c_2$, $z_i \notin h$, and $x_i=0$. Since $z_i \in S_2$ but $z_i \notin h$, then $\hat{\lambda}_i \ge \frac{\epsilon}{16n}$. Therefore, with probability $1-\delta$,

$$Pr_{D_{\eta,T}}[c(x)=1, x_i=0] = \lambda_i \ge \hat{\lambda}_i - \frac{\epsilon}{32n} \ge \frac{\epsilon}{32n}$$

Let L be the number of variables z_i, for which $z_i \in S_2$ and $\hat{\lambda}_i \ge \frac{\epsilon}{16n}$, and assume w.l.o.g that these variables are

$z_1, z_2, \ldots, z_L$. Then, with probability $1-\delta$,

$$\begin{aligned} L\frac{\epsilon}{32n} &\leq \sum_{i=1}^{L} Pr_{D_{\eta,T}}[c(x)=1, x_i=0] \\ &\leq \eta\sum_{i=1}^{L} Pr_{D_{N,T}}[x_i=0] \leq \eta T, \end{aligned}$$

where the second inequality follows from Lemma 4.5 and the last inequality follows from Lemma 4.4. Therefore, with probability $1-\delta$,

$$L \leq \eta\frac{32nT}{\epsilon}.$$

But for each one of these L literals, also $\hat{\beta}_i < 2\alpha$, because they all belong to S_2. Therefore, by Lemma 4.2, for each of these L literals, with probability $1-\delta$,

$$\beta_i = Pr_{D_\eta}[x_i=0] \leq \hat{\beta}_i + \frac{\alpha\epsilon}{16n} < 3\alpha.$$

Thus, if we sum over these L variables, then with probability $1-\delta$,

$$\begin{aligned} Pr_{D_\eta}[c_2(x)=0, h(x)=1] &\leq \sum_{i=1}^{L} Pr_{D_\eta}[x_i=0] \\ &< 3\alpha L \leq \eta\frac{96\alpha nT}{\epsilon} < \frac{\epsilon}{4}, \end{aligned}$$

where the last inequality follows by the choice of η and T. $\square$

The following lemma states that with high probability $h_3 = c_3$ on the support of D_η.

Lemma 4.8 *If the noise rate is $\eta < \frac{\alpha\epsilon}{4}$ then with probability $1-\delta$, $c_3 \equiv h_3$, i.e.,*
$Pr_{D_\eta}[c_3(x) \neq h_3(x)] = 0.$

Proof: We bound $Pr_{D_\eta}[c_3(x)=0, h_3(x)=1]$ and $Pr_{D_\eta}[c_3(x)=1, h_3(x)=0]$ separately.

- $c_3(x)=0, h_3(x)=1$**:** Therefore, there exists a variable z_i, such that $z_i \in c_3$, $z_i \notin h_3$, and $x_i=0$. Since $z_i \in c$, then without noise, $Pr_D[c(x)=1, x_i=0] = 0$. With noise the probability of this event is bounded by,

  $$\gamma_i = Pr_{D_\eta}[c(x)=1, x_i=0] \leq \eta < \frac{\alpha\epsilon}{4}.$$

  Therefore, by Lemma 4.2, with probability $1-\delta$,

  $$\hat{\gamma}_i \leq \gamma_i + \frac{\alpha\epsilon}{16n} < \frac{\alpha\epsilon}{2}.$$

  Hence, we will not delete z_i from the hypothesis h, and thus $z_i \in h_3$. Therefore, $Pr_{D_\eta}[c_3(x)=0, h_3(x)=1] = 0$.

- $c_3(x)=1, h_3(x)=0$**:** Therefore, there exists a variable z_i, such that $z_i \notin c_3$, $z_i \in h_3$, and $x_i=0$. Since $z_i \in h_3$, then $\hat{\beta}_i \geq 2\alpha$. Therefore, by Lemma 4.2, with probability $1-\delta$,

  $$\beta_i = Pr_{D_\eta}[x_i=0] \geq \hat{\beta}_i - \frac{\alpha\epsilon}{16n} > 2\alpha - \frac{\alpha\epsilon}{16n}.$$

  Hence, with probability $1-\delta$,

  $$Pr_D[x_i=0] > 2\alpha - \frac{\alpha\epsilon}{16n} - \eta > \frac{3\alpha}{2}.$$

  Assume without loss of generality that $Pr_D[c(x)=1] > \epsilon$. Therefore, without noise, (for the product distribution),

  $$\begin{aligned} Pr_D[c=1, z_i=0] &= Pr_D[c=1]Pr_D[z_i=0] \\ &> \frac{3\alpha\epsilon}{2}. \end{aligned} \qquad (1)$$

  Note that (1) is the *only* place in the proof that we use the assumption that D is a product distribution. Hence, with probability $1-\delta$,

  $$\gamma_i = Pr_{D_\eta}[c(x)=1, x_i=0] > \frac{3\alpha\epsilon}{2} - \eta > \frac{5\alpha\epsilon}{4},$$

  where the last inequality follows from the choice of η. Therefore, by Lemma 4.2, with probability $1-\delta$,

  $$\hat{\gamma}_i \geq \gamma_i - \frac{\alpha\epsilon}{4n} \geq \frac{\alpha\epsilon}{2}.$$

  Thus, with probability $1-\delta$, the variable z_i will be deleted from the hypothesis h. Hence, $Pr_{D_\eta}[c_3(x)=1, h_3(x)=0] = 0$.

Combining the two cases we have that with probability $1-\delta$, $c_3 \equiv h_3$. $\square$

Theorem 4.1 *If the noise rate is $\eta \leq \frac{\epsilon^{5/3}}{56n^{2/3}}$ then with probability $1-\delta$, $error(h) \leq \epsilon$.*

Proof: By Lemma 4.3, Lemma 4.6, Lemma 4.7 and Lemma 4.8, if $\eta \leq min\{\frac{\epsilon^2}{384\alpha nT}, \frac{\alpha\epsilon}{4}\}$, then with probability $1-\delta$,

$$\begin{aligned} &error(h) = Pr_{D_\eta}[c(x) \neq h(x)] \leq \\ &\quad Pr_{D_\eta}[c_1(x) \neq h_1(x)] + Pr_{D_\eta}[c(x)=1, h_2(x)=0] + \\ &\quad Pr_{D_\eta}[c_2(x)=0, h(x)=1] + Pr_{D_\eta}[c_3(x) \neq h_3(x)] \\ &\quad \leq \frac{\epsilon}{8} + \frac{\epsilon}{4} + \frac{\epsilon}{4} < \epsilon. \end{aligned}$$

Therefore, if we set

$$\alpha \leq \frac{\epsilon^{2/3}}{14n^{2/3}},$$

then $error(h) < \epsilon$, for any noise rate $\eta \leq \frac{\epsilon^{5/3}}{56n^{2/3}}$. $\square$

Corollary 4.1 *The class C_n of boolean conjunctions is efficiently PAC learnable with noise rate $\eta = O(\frac{\epsilon^{5/3}}{n^{2/3}})$, with respect to product distributions.*

5 Extensions

We can extend the proof beyond product distributions. Notice that the requirement that D is a product distribution is used only once, in the proof of Lemma 4.8, in Equation (1). If for each variable z_i,

$$Pr_D[x_i = 0] < \alpha$$

then the analysis of the third phase of the algorithm is not needed (i.e. we can set $S_3 = \emptyset$.) Therefore, in this case the algorithm can learn the target function with respect to *any* distribution that has this property. Since we do not have case three, we do not need that $\eta \leq \alpha\epsilon/2$. This implies that we can set leave α as a parameter that defines the bound that a variable is zero. Given α, we can tolerate a noise rate of $O(\frac{\epsilon^3}{\alpha^2 n^2})$. Note that for $\alpha_0 = \epsilon/n$ we have $\eta = O(\epsilon)$, and for $\alpha_1 = \epsilon/\sqrt{n}$, we have $\eta = O(\epsilon/n)$. For all the range between α_0 and α_1 we get an improvement over the previous results. More formally,

Theorem 5.1 *Let $\mathcal{D}_2$ be the class of distributions, such that for each variable z_i, $Pr_D[x_i = 0] < \alpha$. The class C_n is efficiently PAC learnable with noise rate $\eta = O(\frac{\epsilon^3}{\alpha^2 n^2})$ with respect to distributions in $\mathcal{D}_2$.*

In the analysis of the third phase of the algorithm, in Equation (1), we use that fact that D is a product distribution, and that $Pr_D[c = 1]$ and $Pr_D[z_i = 0]$ are both substantial, in order to show that $Pr_D[c = 1, z_i = 0]$ is large. However, it is sufficient that if $Pr_D[c(x) = 1]$ and $Pr_D[x_i = 0]$ are both substantial then $Pr_D[c(x) = 1, x_i = 0]$ is non-negligible.

Theorem 5.2 *Let $\mathcal{D}_3$ be the class of distributions, such that if $Pr_D[c(x) = 1] > \epsilon$, then for each variable z_i for which $Pr_D[x_i = 0] > \alpha$, also $P_D[c(x) = 1, x_i = 0] = \Omega(\alpha\epsilon)$. The class C_n is efficiently PAC learnable with noise rate $\eta = O(\frac{\epsilon^{5/3}}{n^{2/3}})$, with respect to distributions in $\mathcal{D}_3$.*

Note that all the conditions are on the underlined distribution and not on the noise distribution. We can tolerate *any* noise distribution, combined with an underlined distribution from one of the above classes of distributions.

Acknowledgment

We would like to thank Mike Kearns and Dana Ron for helpful discussions.

References

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

[D93] Scott Decatur Statistical querries and faulty PAC oracles. In *Proc. 6th Annu. Workshop on Comput. Learning Theory*, pages 262–268. ACM Press, 1993.

[K93] Michael Kearns Efficient noise tolerant learning from statistical querries. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, May 1993.

[KL88] Michael Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, Chicago, Illinois, May 1988.

[KSS92] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 341–352. ACM Press, 1992.

[Lit88] Nick Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[Slo88] Robert H. Sloan. Types of noise in data for concept learning. In David Haussler and Leonard Pitt, editors, *First Workshop on Computational Learning Theory*, pages 91–96. Morgan Kaufmann, August 1988.

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

[Val85] Leslie G. Valiant. Learning disjunctions of conjunctions. In *Proceedings IJCAI-85*, pages 560–566. International Joint Committee for Artificial Intelligence, Morgan Kaufmann, August 1985.