Hardness of Learning Parity Functions Under Different Frameworks

Oliver Vipond

April 10, 2020

Contents

1 Introduction

The problem of Learning Parity functions is well studied in Computational Learning Theory, and is closely related to cryptographic and coding problems [BKW03] [Pie12]. Of particular interest in this project is whether one can learn the concept class $\mathsf{PARITIES}$ in the presence of noise. This concept class is efficiently learnable in the PAC-Learning framework but it can be shown not to be efficiently learnable in the Statistical Query framework when endowed with the uniform distribution. In this project we shall introduce the various learning frameworks, how these frameworks are related, and the learnability of the $\mathsf{PARITIES}$ concept class in each framework. Finally we shall introduce results from [BKW03] and comment on their significance to the relationship between the learning models. Let us formally define the parity function class.

Definition 1.1. Let $X_n = \{0,1\}^n$, then for any $S \subseteq X$ let $f_S(x) = \bigoplus_{i \in S} x_i$ where $\oplus$ denotes XOR or equivalently addition in $\mathbb{F}_2$. Then $\mathsf{PARITIES}_n = \{f_S : S \subseteq X_n\}$ and $\mathsf{PARITIES} = \bigcup_n \mathsf{PARITIES}_n$ is a concept class on $X = \bigcup_n X_n$.

2 Learning Frameworks

Let us introduce some preliminary definitions which provide the foundations to our learning models.

Definition 2.1. [KV94] Let X denote the *instance space* thought of as the objects or instances over which a concept is defined. Let $c \subset X$ or equivalently $c : X \to \{0, 1\}$ denote a particular *concept* over X for which $c(x) = 1$ embodies that the instance x is a positive example of the concept c. Then we denote a collection of concepts over X as $\mathcal{C}$ and this is known as our concept class. Let $\mathcal{D}$ denote an arbitrary probability distribution on our instance space X.

As part of our learning framework we will want to encode the complexity of a concept, in order that we allow the flexibility for the complexity of our algorithms to depend on the difficulty of the concept being learned.

Definition 2.2. [KV94] Let Σ denote an alphabet and $\mathcal{R} : \Sigma^* \to \mathcal{C}$ a surjective map known also as a *representation scheme*. Let size : $\Sigma^* \to \mathbb{N}$ define a natural notion of size on words in the alphabet Σ. Then we define the size of a concept under this representation scheme to be $\text{size}(c) = \min_{\mathcal{R}(\sigma)=c}\{\text{size}(\sigma)\}$.

It is common to grade both the instance space and concept class by size $X = \bigcup_n X_n$, $\mathcal{C} = \bigcup_n \mathcal{C}_n$, and this grading will usually be intuitive e.g. $\{0, 1\}^* = \bigcup_n \{0, 1\}^n$, with $\mathcal{C}_n$ a collection of concepts on the instance subspace X_n.

If we wish to evaluate the performance of a learning algorithm we also need to define a measure of how well the output of an algorithm matches the desired concept. The error of an output is naturally defined as follows:

Definition 2.3. [KV94] Let $h : X \to \{0, 1\}$ be the output of an algorithm trying to learn the concept c over X with distribution $\mathcal{D}$. The error is defined as:

$$\text{error}(h) = \mathbb{P}_{x \sim \mathcal{D}}(c(x) \neq h(x))$$

The main difference between the learning frameworks we consider will be the richness of the information about the concept c that the algorithm is assumed to have access to. A slightly more subtle change one can investigate is the set from which the output hypothesis of an algorithm h, is permitted to be drawn.

2.1 PAC-Learning

The first central model in Computational Learning Theory is the Probably Approximately Correct Learning model (PAC-Learning). Our initial model will assume access to perfect samples from the concept class being learned.

Definition 2.4. [KV94] Suppose c is a concept over X with distribution $\mathcal{D}$. The example oracle $\mathsf{EX}(c, \mathcal{D})$ is a resource that at unit cost draws an example $(x, c(x))$ where the example is drawn according to the distribution on X.

Definition 2.5. A *hypothesis class* on instance space X, $\mathcal{H} = \bigcup_n \mathcal{H}_n$, is merely a concept class. We say the hypothesis class is polynomially evaluable if there is an algorithm that evaluates $h(x)$ for any $h \in \mathcal{H}, x \in X$ in time polynomial in n, size(h) where $h \in \mathcal{H}_n$.

Definition 2.6. (PAC-Learnable)

Let $\mathcal{C}, \mathcal{H}$ be concept classes over X. We say that $\mathcal{C}$ is PAC learnable using hypothesis class $\mathcal{H}$ if there is an algorithm $\mathcal{L}$ such that: for any $c \in \mathcal{C}_n$, for any distribution $\mathcal{D}$ on X, given tolerance constants $\varepsilon, \delta \in (0, \frac{1}{2})$, the constant size$(c)$, and access to the example oracle $\mathsf{EX}(c, \mathcal{D})$ then $\mathcal{L}$ outputs a concept $h \in \mathcal{H}$ such that with probability at least $1 - \delta$ we have error$(h) \leq \varepsilon$.

Definition 2.7. (Efficiently PAC-Learnable) We say that $\mathcal{C}$ is *efficiently* PAC learnable using hypothesis class $\mathcal{H}$ if $\mathcal{C}$ is PAC learnable using $\mathcal{H}$, $\mathcal{H}$ is polynomially evaluable and for $c \in \mathcal{C}_n$ the algorithm $\mathcal{L}$ runs in time polynomial in $\{\text{size}(c), n, \frac{1}{\varepsilon}, \frac{1}{\delta}\}$.

Let us show that in this noiseless model the class PARITIES is efficiently PAC-learnable using hypothesis class PARITIES.

Proposition 2.1. The concept class PARITIES is efficiently PAC-Learnable over PARITIES.

Proof. Let us recast learning our parity function into a problem of performing Gaussian elimination. Consider a parity function f_S for $S \subseteq X_n$ then for all $x \in X_n$ we note $f_S(x) = x^T \mathbf{1}_S$ where we consider $x \in \mathbb{F}_2^n$ and $\mathbf{1}_S$ is the vector with a 1 in position i for $i \in S$.

Consider taking m samples from the example oracle and assembling an $m \times n$ matrix A with each row corresponding to a sample and constructing a vector $b \in \mathbb{F}_2^m$ of the corresponding concept value on each sample. Using Gaussian elimination we can find a subset $S \subset X_n$ such that $A\mathbf{1}_S = b$ which corresponds to a parity function consistent with our samples in time polynomial in m, n.

Let $T = \{h \in \mathsf{PARITIES}_n : \text{error}(h) > \varepsilon\}$ and let E_h denote the event that hypothesis h agrees with m independent samples produced by the oracle. For $h \in T$ we note that $\mathbb{P}(E_h) < (1-\varepsilon)^m$ and since $|\mathsf{PARITIES}_n| = 2^n$ using sub-additivity we see $\mathbb{P}(\cup_{h \in T} E_h) < 2^n(1-\varepsilon)^m$. Simple calculations reveal:

$$m > \frac{1}{\varepsilon}\left(\log(2^n) + \log(\frac{1}{\delta})\right) \implies m \geq \frac{1}{-\log(1-\varepsilon)}\left(\log(2^n) + \log(\frac{1}{\delta})\right) \implies 2^n(1-\varepsilon)^m < \delta$$

Hence if we take at least $m = \lceil \frac{1}{\varepsilon}\left(\log(2^n) + \log(\frac{1}{\delta})\right) \rceil$ -samples from the oracle we have that with probability at least $1-\delta$ a candidate function outputted from the Gaussian elimination has error at most ε. Thus since m is bounded by a polynomial in $\{\text{size}(c), n, \frac{1}{\varepsilon}, \frac{1}{\delta}\}$ and Gaussian elimination is polynomial we see that PARITIES is efficiently PAC-Learnable over PARITIES. (Argument derived from the general ideas present in [BEHW87]). □

Note that this algorithm would require major adaptation to cope with an oracle that supplied examples corrupted by noise, since a consistent solution to the Gaussian elimination would not be guaranteed to exist.

2.2 Queries with Noise

Let us formally introduce the learning model that requires an algorithm to be tolerant to noise. We shall use the parameter $\eta \in [0, \frac{1}{2})$ to denote the probability that each sample is corrupted with noise.

Definition 2.8. (Rademacher Random Variable) We say $Z \sim R(\eta)$ is a Rademacher random variable with parameter η if Z takes values in $\{-1, 1\}$ and $\mathbb{P}(Z = -1) = \eta$.

Definition 2.9. (Random Classification Noise Example Oracle) Suppose c is a concept over X with distribution $\mathcal{D}$. The example oracle with random classification noise rate η, $\mathsf{EX}^{\eta}(c, \mathcal{D})$ is a resource that at unit cost draws an example $(x, \frac{1-Z}{2} + Zc(x))$ where the example is drawn according to the distribution on X and $Z \sim R(\eta)$ is an independent Rademacher random variable. This means for each sample we are given the correct label with probability $1 - \eta$.

Intuitively as the noise our samples are subject to approaches $\frac{1}{2}$ we receive less useful information from the oracle and thus the harder it will be to learn the concept class. We therefore allow for our learning algorithms to depend on the associated parameter $\frac{1}{1-2\eta}$. Certainly one can consider alternative forms of noise in which the noise is correlated across samples and this typically increases the difficulty of learning. Consider for example an oracle returning consistent samples $(x, \frac{1-Z_x}{2} + Z_x c(x))$ where we use the same Rademacher random variable Z_x each time we draw x rather than introducing an independent random variable with each sample. In this setting we would not be able to recover the true value of a corrupted sample through repeated sampling.

Definition 2.10. (PAC-Learnable with Random Classification Noise)

Let $\mathcal{C}, \mathcal{H}$ be concept classes over X. We say that $\mathcal{C}$ is PAC learnable with random classification noise using hypothesis class $\mathcal{H}$ if there is an algorithm $\mathcal{L}$ such that: for any $c \in \mathcal{C}_n$, for any distribution $\mathcal{D}$ on X, given tolerance constants $\varepsilon, \delta \in (0, \frac{1}{2})$, the constants $\text{size}(c), \eta_0 \in [\eta, \frac{1}{2})$, and access to the noisy example oracle $\mathsf{EX}^{\eta}(c, \mathcal{D})$ then $\mathcal{L}$ outputs a concept $h \in \mathcal{H}$ such that with probability at least $1 - \delta$ we have $\text{error}(h) \leq \varepsilon$.

We say that $\mathcal{C}$ is *efficiently* PAC learnable with random classification noise using hypothesis class $\mathcal{H}$, if $\mathcal{H}$ is polynomially evaluable and there is an algorithm that runs in time polynomial in $\{\frac{1}{1-2\eta_0}, \text{size}(c), n, \frac{1}{\varepsilon}, \frac{1}{\delta}\}$

2.3 Statistical Query Learning

Our next learning framework is rather different in that we do not sample examples from an oracle but instead make *statistical queries*. We will show that these statistical queries can in fact be efficiently simulated by a noisy example oracle and so that it suffices to provide an efficient algorithm that requires statistical queries with a noisy example oracle instead.

Definition 2.11. [Kea98] Let X be an instance space with distribution $\mathcal{D}$, c a concept over X. The *statistical query oracle* $\mathsf{STAT}(c, \mathcal{D})$ takes as input an arbitrary function $\chi : X \times \{0, 1\} \to \{0, 1\}$ and a tolerance parameter τ, and returns a τ-approximate value v of the expectation of the function χ evaluated on the target concept c thought of as a set $c \subset X \times \{0, 1\}$. Alternatively χ may be thought of as a predicate for labeling X.

$$\left|\mathbb{E}_{x \sim \mathcal{D}}[\chi(x, c(x))] - v\right| \leq \tau$$

Definition 2.12. (Statistical Query Learnable)

Let $\mathcal{C}, \mathcal{H}$ be concept classes over X. We say that $\mathcal{C}$ is learnable from statistical queries using hypothesis class $\mathcal{H}$ if there is an algorithm $\mathcal{L}$ such that: for any $c \in \mathcal{C}_n$, for any distribution $\mathcal{D}$ on X, given the constants $\varepsilon \in (0, \frac{1}{2})$, size$(c)$, and access to the statistical query oracle $\mathsf{STAT}(c, \mathcal{D})$ then $\mathcal{L}$ outputs a concept $h \in \mathcal{H}$ such that error$(h) \leq \varepsilon$.

We say that $\mathcal{C}$ is *efficiently* learnable from statistical queries using hypothesis class $\mathcal{H}$, if $\mathcal{H}$ is polynomially evaluable and there are polynomials p, q, r in $\{\text{size}(c), n, \frac{1}{\varepsilon}\}$ such that for any query (χ, τ) made by the algorithm $\mathcal{L}$ the function χ is evaluable in time p, $\frac{1}{\tau}$ is bounded by q, and $\mathcal{L}$ runs in time polynomial in r.

Theorem 2.1. If $\mathcal{C}$ is efficiently SQ-learnable using hypothesis class $\mathcal{H}$ then $\mathcal{C}$ is efficiently PAC-learnable with random classification noise using hypothesis class $\mathcal{H}$.

We shall prove a series of lemmas on the way to a proof for the above theorem. It will be mathematically convenient to redact our concepts c and functions χ to be $\{-1, 1\}$-valued rather than $\{0, 1\}$-valued, and χ to have domain $X \times \{-1, 1\}$.

Definition 2.13. Let $\psi : X \to \{-1, 1\}$ be any function and τ, α a tolerance parameters, c a target concept and $\mathcal{D}$ a distribution on the instance space. We say a resource can simulate a *target independent* query if given (ψ, τ) the resource returns a v such that $|\mathbb{E}_{x \sim \mathcal{D}}[\psi(x)] - v| \leq \tau$ with probability at least $1 - \alpha$. We say a resource can simulate a *correlational* query if given (ψ, τ) the resource returns a v such that $|\mathbb{E}_{x \sim \mathcal{D}}[\psi(x)c(x)] - v| \leq \tau$ with probability at least $1 - \alpha$.

Lemma 2.1. The noisy oracle $\text{EX}^{\eta}(c, \mathcal{D})$ can simulate target independent queries.

Proof. It suffices to find a $(1-\alpha)$ confidence interval for $\mathbb{E}[\psi(x)]$ of width less than 2τ. Note that we are only interested in drawing samples $x \sim \mathcal{D}$ and so we are unaffected by the noise applied to $c(x)$ of a given sample. Let us draw m samples from the noisy oracle and define $S(x) = \frac{1}{m} \sum_{i=1}^{m} \psi(x_i)$, and note that $\mu = \mathbb{E}[S(x)] = \mathbb{E}_{y \sim \mathcal{D}}[\psi(y)]$. Then Chebyshev's inequality gives that $\mathbb{P}_{x_i \sim \mathcal{D}}(|\mu - S(x)| > \tau) \leq \frac{\text{Var}(S(x))}{\tau^2}$. Note that $\text{Var}_{y \sim \mathcal{D}}(\psi(y)) \leq 1$ and so $\text{Var}(S(x)) \leq \frac{1}{m}$. Thus for $m > \frac{1}{\alpha \tau^2}$ the sample mean $S(x)$ is a τ-approximate with probability at least $1 - \alpha$. □

Lemma 2.2. Given the noisy oracle $\text{EX}^{\eta}(c, \mathcal{D})$ one can enumerate $\lceil \frac{4\eta_0}{(1-2\eta_0)^2 \tau} \rceil$ simulations and at least one of which simulates correlational queries.

Proof. Let $Z \sim R(\eta)$ be a Rademacher random variable, and note that since we modified the codomain of our concepts, a sample from the noisy oracle takes the form $(x, Zc(x))$. Using independence we observe $\mathbb{E}_{(x,y) \sim \mathsf{EX}^{\eta}(c,\mathcal{D})}[\phi(x)y] = (1 - 2\eta)\mathbb{E}_{x \sim \mathcal{D}}[\phi(x)c(x)]$. Using the same argument in the proof of the previous lemma we may choose m polynomial in τ_1, α so that $\hat{v} = \frac{1}{m} \sum \phi(x_i) y_i$ is a τ_1-approximate of $\mathbb{E}_{(x,y) \sim \mathsf{EX}^{\eta}(c,\mathcal{D})}[\phi(x)y]$ with probability at least $1 - \alpha$.

In the model of learning with random classification noise, the noise parameter is assumed to be unknown, and only an upper bound η_0 provided. Let $\hat{\eta} \leq \eta_0$ and suppose $|\hat{\eta} - \eta| \leq \frac{\Delta}{2}$, then the corresponding difference between our sample correlational query and the true expected correlation is bounded by:

$$\left|\frac{\hat{v}}{1-2\hat{\eta}} - \mathbb{E}_{x\sim\mathcal{D}}[\phi(x)c(x)]\right| \leq \left|\frac{\hat{v}}{1-2\hat{\eta}} - \frac{\hat{v}}{1-2\eta} + \frac{\hat{v}}{1-2\eta} - \frac{\mathbb{E}_{(x,y)\sim\mathsf{EX}^{\eta}(c,\mathcal{D})}[\phi(x)y]}{1-2\eta}\right|$$
$$\leq |\hat{v}|\left|\frac{1}{1-2\hat{\eta}} - \frac{1}{1-2\eta}\right| + \frac{1}{1-2\eta}\left|\hat{v} - \mathbb{E}_{(x,y)\sim\mathsf{EX}^{\eta}(c,\mathcal{D})}[\phi(x)y]\right|$$
$$\leq \frac{2\Delta}{(1-2\eta_0)^2} + \frac{\tau_1}{1-2\eta_0}$$

If $\tau_1 = \frac{(1-2\eta_0)\tau}{2}$ and $\Delta = \frac{(1-2\eta_0)^2\tau}{4}$ then this error is bounded by τ. If we run this correlation query simulation for the $\frac{4\eta_0}{(1-2\eta_0)^2\tau}$ possible values of $\hat{\eta} = n\Delta, n \in [\lfloor\frac{\eta_0}{\Delta}\rfloor]$, then at least one of these simulations simulates correlational queries. □

Proof. (Proof of Theorem)

Let us first observe that we can decompose a statistical query into target independent and correlational queries:

$$\mathbb{E}_{x\sim\mathcal{D}}[\chi(x,c(x))] = \mathbb{E}_{x\sim\mathcal{D}}[\chi(x,1)\mathbf{1}\{c(x)=1\}] + \mathbb{E}_{x\sim\mathcal{D}}[\chi(x,-1)\mathbf{1}\{c(x)=-1\}]$$
$$= \mathbb{E}_{x\sim\mathcal{D}}\left[\chi(x,1)\frac{1+c(x)}{2}\right] + \mathbb{E}_{x\sim\mathcal{D}}\left[\chi(x,-1)\frac{1+c(x)}{2}\right]$$
$$= \frac{1}{2}\Bigg(\mathbb{E}_{x\sim\mathcal{D}}[\chi(x,1)] + \mathbb{E}_{x\sim\mathcal{D}}[\chi(x,-1)]$$
$$+ \mathbb{E}_{x\sim\mathcal{D}}[\chi(x,1)c(x)] - \mathbb{E}_{x\sim\mathcal{D}}[\chi(x,-1)c(x)]\Bigg)$$

Given the previous two lemmas with the noisy oracle we can enumerate $\lceil\frac{4\eta_0}{(1-2\eta_0)^2\tau}\rceil$ simulations of which at least one will simulate any statistical query with probability at least $(1-\alpha)$. Moreover note that in the previous lemmas the sample size dependence and running times are polynomial in the appropriate variables. Thus given we may convert an SQ-learning algorithm into $\lceil\frac{4\eta_0}{(1-2\eta_0)^2\tau}\rceil$ efficient algorithms only requiring access to the noisy oracle - each corresponding to a different choice of $\hat{\eta}$. Thus we produce $\lceil\frac{4\eta_0}{(1-2\eta_0)^2\tau}\rceil$ hypotheses of which at least one has small error with high probability. Taking a new set of samples from the noisy oracle and examining the empirical errors is sufficient to pick out a desired hypothesis [AL88]. □

One can show that PARITIES is not SQ-learnable. One considers the uniform distribution on the instance space and uses information theory arguments to show for sufficiently small ε the number of statistical queries required is exponential in the size of the instance space n. In the paper [BFJ$^+$94] it is shown that in the SQ-learning model one cannot weakly-learn concept classes containing a large number of uncorrelated concepts:

Definition 2.14. (SQ-dimension)[BFJ$^+$94] Let $\mathcal{C}$ be a concept class and $\mathcal{D}$ a distribution on instance space X. The SQ-dimension of $\mathcal{C}$ over $\mathcal{D}$ is the largest integer d such that there is a d-sized subset $\mathcal{I} \subset \mathcal{C}$ with:

$$\sup_{c \neq c' \in \mathcal{I}} \left| \mathbb{P}_{x \sim \mathcal{D}}(c(x) = c'(x)) - \mathbb{P}_{x \sim \mathcal{D}}(c(x) \neq c'(x)) \right| \leq \frac{1}{d^3}$$

Theorem 2.2. [BFJ+94] Let $\mathcal{C}$ be a concept class on X with SQ-dimension d. In order to learn $\mathcal{C}$ in the SQ-model with error less than $\frac{1}{2} - \frac{1}{d^3}$ then the number of queries called by the algorithm or the reciprocal of the smallest tolerance queried $\frac{1}{\tau}$ is at least $\frac{1}{2}d^{\frac{1}{3}}$

This demonstrates that in fact PARITIES cannot even be *weakly* learned in the SQ-model.

Given that any SQ-learnable concepts are efficiently PAC-learnable with RCN, a subject of interest is identifying concepts which are PAC-learnable with RCN but not SQ-learnable. We will discuss a concept class related to PARITIES which lies in between these learning models in the next section.

3 A Noise Tolerant Algorithm

The paper [BKW03] introduced the first example of an efficient noise tolerant algorithm for a concept class that is not SQ-learnable. The notion of efficiently PAC-learnable with RCN used in this paper deviates from our definition since it is assumed that the error rate η is known and moreover the efficient algorithms proposed are demonstrably super-polynomial in $\frac{1}{1-2\eta_0}$.

The concept class that we work with throughout shall be denoted PARITIES' $= \bigcup_n$ PARITIES'$_n$ and is the subclass consisting of parity functions which are dependent only on the first $k(n)$ literals where $k = O(\log n \log\log n)$. Note that the information theoretic result at the end of the previous section demonstrate that this class is not SQ-learnable over the uniform distribution, since PARITIES'$_n$ contains $n^{O(\log\log n)}$ pairwise uncorrelated functions.

Theorem 3.1. [BKW03]

Let $a(n), b(n)$ be positive-integer valued such that $ab \geq n$, $a = O(\log n), b = O(\frac{n}{\log n})$. Then PARITIES is PAC-learnable with RCN, where given that the target parity function lies in PARITIES$_n$ the algorithm requires sample size and computation time polynomial in $\{2^b, (\frac{1}{1-2\eta})^{2^a}, \frac{1}{\varepsilon}, \frac{1}{\delta}\}$.

An immediate consequence of this theorem is that PARITIES' is efficiently PAC-learnable with RCN (η known).

Corollary 3.1. The class PARITIES' is efficiently PAC-learnable with RCN.

Proof. Let $a = \frac{1}{2}\log n$ and $b = \frac{2n}{\log n}$ then we note that for a constant noise-rate of η we can bound the sample size and computation time by order $2^{O(\frac{n}{\log n})}$. Thus we can bound the computation time and sample size for PARITIES'$_n$ by order $2^{O(\frac{\log n \log\log n}{\log\log n})} = O(n)$. □

We will be using an algorithm that has the flavour of a statistically robust Gaussian elimination in which we use linear combinations of the samples to infer labels on other examples. The following lemma follows from a simple induction:

Lemma 3.1. Let $\{(x_i, y_i)\}$ be a collection of m samples drawn from the noisy oracle $\mathsf{EX}^{\eta}(c, \mathcal{D})$ then:

$$\mathbb{P}(c(\sum x_i) = \sum y_i) = \frac{1}{2} + \frac{1}{2}(1 - 2\eta)^m$$

Where our sums are taken in $\mathbb{F}_2$.

We can interpret the lemma intuitively. Suppose we wish to infer the label y of an instance x from our samples. Then the more samples we sum to find a corresponding label for x, the more likely we are to introduce noise that makes us err.

Let us outline our algorithm. For clarification of notation, in the following discussion $M^{(j)} \in M_{a2^b \times ab}(\mathbb{F}_2)$ with rows indexed by $(i, v) \in [a] \times \{0,1\}^b$. We denote the l^{th} block of b elements of a given row as $M^{(j)}_{i,v}[l]$ for $l \in [a]$.

For each j we seek to produce a matrix M with the following properties.

$$M_{i,v}[l] = 0 \in \{0,1\}^b \text{ for } l \in [i-1] \text{ and } M_{i,v}[i] = v$$

Given these simple properties we observe that for any vector $x \in \{0,1\}^{ab}$ we can express x as a sum of a rows of M by recursively defining $x_0 = x, \quad x_i = x_{i-1} + M_{i,x_{i-1}[i]}$, and unravelling the series of equations as in Euclid's Algorithm. Note that the first i blocks of x_i are all zero and the addition of $M_{i,x_{i-1}[i]}$ nullifies the i^{th} block of x_{i-1}.

Let us define a function Decomp : $M_{a2^b \times ab}(\mathbb{F}_2) \times \{0,1\}^{ab} \to \mathsf{TRUE} \cup [a]$. If the sum of the rows in a matrix M corresponding to the decomposition of x described above coincides with x then $\text{Decomp}(M, x) = \mathsf{TRUE}$. Else $\text{Decomp}(M, x) = i$ where i is the minimum index such that $x_i[i] \neq 0$.

Supposing $\text{Decomp}(M, x) = \mathsf{TRUE}$ let the function $\text{Predict}(M, x) \in \{0,1\}$ denote the label predicted by the decomposition into previously observed samples.

Starting from the zero matrix we will try and construct a matrix like M using samples from the noisy oracle. A matrix constructed from perfect samples encodes the target parity function, but since our data is imperfect we construct N matrices so that with high probability the majority prediction is correct.

Algorithm 1: Statistically Robust Gaussian Elimination

Data: Matrices $M^{(j)}$, the labels associated to the samples used to previously update the matrices, and a new sample x from the noisy oracle

Result: We update a matrix $M^{(j)}$ with the new sample and report "I don't know", or make a prediction for the label of x

```
begin
    for j ∈ [N] do
        if Decomp(M^(j), x) ≠ TRUE then
            M^(j)_{i,x_{i-1}[i]} ⟵ x_{i-1}
            Prediction = I don't know
            break
        else
            p_j = Predict(M^(j), x)
        end
    end
    if Decomp(M^(j), x) = TRUE for all j then
        Prediction = round(∑_j p_j)
    end
    return Prediction
end
```

We observe that each row $M_{i,v}$ is the sum of at most 2^{i-1} previously observed examples, and so a prediction p_j is derived from the sum of at most 2^a examples. Let us analyse the probability that when we report a prediction it is incorrect using Lemma 3.1. Each matrix predicts correctly with probability at least $\frac{1}{2} + \frac{1}{2}(1-2\eta)^{2^a}$. Using standard arguments with Chernoff-Hoeffding bounds the majority decision is incorrect on an individual sample with probability less than $e^{-\frac{N}{2}(1-2\eta)^{2^{a+1}}}$. Using the union bound we note that the probability of erring on any prediction is bounded by $2^{ab}e^{-\frac{N}{2}(1-2\eta)^{2^{a+1}}}$.

Moreover note that each of our N matrices has $a2^b$ rows and each row is modified once by a new sample, thus we report the prediction "I don't know" $Na2^b$ times.

Given tolerance parameters δ, ϵ we calculate that for $N = 2^{O(\sqrt{n})}\log\frac{1}{\delta}$ then we report "I don't know" $Na2^b = 2^{O(n/\log n)}\log\frac{1}{\delta}$ times and make no errors in our prediction with probability at least $1 - 2^n e^{-\frac{N}{2}(1-2\eta)^{2^{a+1}}} = 1 - 2^n e^{-2^{O(\sqrt{n})}\log\frac{1}{\delta}(1-2\eta)^{O(n)}} = 1 - 2^n\delta^{2^{O(\sqrt{n})}(1-2\eta)^{O(n)}} \geq 1 - \delta$. Thus if we take $2^{O(n/\log n)}\frac{1}{\varepsilon}\log\frac{1}{\delta}$ samples then we are correct in our prediction on at least a $1-\varepsilon$ fraction of the examples.

The above algorithm in fact satisfies a more difficult learning setting (online mistake-bound model [Lit88]) whereby an adversary presents an arbitrary sequence of examples where the label is flipped with probability η and the algorithm reports a prediction or "I don't know". In this setting a successful algorithm is required to be correct with high probability and report "I don't know" on only a small proportion of examples.

Most importantly this example exhibits the set of SQ-learnable concept classes as a proper subset of the set of concept classes PAC-learnable with RCN (η known).

References

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

[BEHW87] Alselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Inf. Process. Lett.*, 24(6):377–380, April 1987.

[BFJ+94] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. 10 1994.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, November 1998.

[KV94] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, USA, 1994.

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, Apr 1988.

[Pie12] Krzysztof Pietrzak. Cryptography from learning parity with noise. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science*, pages 99–114, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.