Separating Models of Learning with Faulty Teachers

Vitaly Feldman[a,*,1] Shrenik Shah[b]

[a]*IBM Almaden Research Center, San Jose, CA 95120, USA*

[b]*Harvard University, Cambridge, MA 02138, USA*

Abstract

We study the power of two models of faulty teachers in Valiant's PAC learning model and Angluin's exact learning model. The first model we consider is learning from an incomplete membership oracle introduced by Angluin and Slonim (1994). In this model, the answers to a random subset of the learner's membership queries may be missing. The second model we consider is random persistent classification noise in membership queries introduced by Goldman, Kearns, and Schapire (1993). In this model, the answers to a random subset of the learner's membership queries are flipped.

We show that in both the PAC and the exact learning models the incomplete membership oracle is strictly stronger than the noisy membership oracle under the assumption that the problem of PAC learning parities with random classification noise is intractable.

We also show that under the standard cryptographic assumptions the incomplete membership oracle is strictly weaker than the perfect membership oracle. This generalizes the result of Simon (2004) and resolves an open question of Bshouty and Eiron (2002).

Key words: Models of learning, PAC, exact learning, membership query.

[*] Corresponding author.

Email addresses: `vitaly@post.harvard.edu` (Vitaly Feldman), `sshah@fas.harvard.edu` (Shrenik Shah).

[1] Part of the work done while the author was at Harvard University supported by grants from the National Science Foundation NSF 0432037 and NSF 0427129.

1 Introduction

Modeling and handling of faulty information is one of the most important and well-studied topics in learning theory. In this paper we study two natural models of a faulty teacher, where a teacher is represented by access to a *membership oracle*. A membership oracle allows the learning algorithm to obtain the value of the unknown target function f on any point in the domain. In the first model we consider, the faulty teacher answers "I don't know" with some probability p to every membership query (MQ) of the learner. Furthermore, if the learner asks the same membership query again, the answer will be the same (in other words, it *persists*). This model was introduced by Angluin and Slonim [3] in the context of Angluin's exact learning model [1]. Such a faulty membership oracle is referred to as *incomplete*. Angluin and Slonim showed that monotone DNF formulas are exactly learnable with incomplete membership queries for constant p. This result was improved by Bshouty and Eiron who gave an algorithm that can learn monotone DNF even when only an inverse polynomial fraction of membership queries is answered [8]. Bshouty and Owshanko showed learnability of regular sets in this model [9], Goldman and Mathias showed learnability of k-term DNF [15], and Chen showed learnability of some restricted classes of DNF in this model [10]. Given a number of strong positive results for this model a natural question to ask is whether this model is equivalent to learning with perfect membership queries [8]. This question was addressed by Simon who answered it in the negative for exact learning with *proper* equivalence queries (that is, the hypothesis in the equivalence query has to belong to the concept class that is learned) [24]. In this work (Theorem 4.1) we give a more general version of this result that also applies to unrestricted equivalence queries and the PAC model. Our result shows that if there exists a concept class not learnable in the exact model (or the PAC model), then the exact learning model with MQs (the PAC model with MQs) is stronger then the exact learning model with incomplete MQs [2] (the PAC model with incomplete MQs, respectively). In particular, if one-way functions exist, then incomplete MQs are strictly weaker than perfect ones.

The other model of a faulty teacher we study is random persistent noise in membership queries defined by Goldman, Kearns, and Schapire [14] in the context of exact identification using membership queries alone. In this model, the teacher flips the label of the answer to every membership query with some probability η. As in the incomplete MQ model, the answers persist. It is easy to see that learning is this model is at least as hard as learning in the incomplete MQ model. Among the few techniques that manage to exploit noisy MQs is the result of Goldman et al. who prove that certain classes of read-once formulas

[2] The main idea of this simple result is similar to that of Simon and we include it primarily for completeness.

are exactly learnable in this model [14]. It is also not hard to see that concept classes that are exactly learnable using the Kushilevitz-Mansour algorithm [20] can be learned from noisy MQs by using noise tolerant versions of the Kushilevitz-Mansour algorithm given by Jackson, Shamir, and Shwartzman [17] and Feldman [12]. These classes include juntas and $\log n$-depth decision trees [20]. In addition, DNF expressions are known to be PAC learnable with respect to the uniform distribution using noisy membership queries [17]. Exact learnability of monotone DNF with noisy membership queries is an open problem [3].

In the main result of this work, we demonstrate that under the assumption that parities are not learnable with random classification noise, the incomplete membership oracle is strictly stronger than the noisy one. Formally, we prove the following results.

Theorem 1.1 *If the problem of PAC learning parities over the uniform distribution with random classification noise of rate η is intractable, then there exists a concept class $\mathcal{C}$ that is learnable with equivalence and incomplete membership queries, but not learnable from equivalence and noisy membership queries of error rate η.*

We also give a version of this result for the PAC model.

Theorem 1.2 *If the problem of PAC learning parities over the uniform distribution with random classification noise of rate η is intractable, then there exists a concept class $\mathcal{C}$ that is PAC learnable with incomplete membership queries, but not PAC learnable (even weakly) from noisy membership queries of error rate η.*

Our separations are optimal in the sense that they separate learning with any rate of "I don't know"s from learning with a constant rate of noise.

Learning of parities from noisy random and uniform examples (which we refer to as the *noisy parity problem*) is a notoriously hard open problem. It is known to be equivalent to decoding of binary linear codes generated randomly — a long-standing open problem in coding theory (*cf.* [12]). For example, the McEliece cryptosystem is based, among other assumptions, on the hardness of this problem [22]. While the average-case hardness of decoding binary linear codes is unknown, a number of related worst-case problems are known to be NP-hard (*cf.* [5,4,26]). Blum, Furst, Kearns, and Lipton use the assumption that this problem is hard to build simple pseudorandom generators [6]. Furthermore, Feldman, Gopalan, Khot, and Ponuswami show that this problem is central to PAC learning with respect to the uniform distribution by reducing a number of other well-known open problems to it [13]. Other evidence of its hardness include non-learnability in the statistical query model of Kearns [18] and hardness of a generalized version of this problem that was shown by

Regev [23]. The only known non-trivial algorithm for learning parities with noise is a $2^{O(n/\log n)}$-time algorithm by Blum, Kalai, and Wasserman [7].

The general idea behind these separations is simple. Given a concept class $\mathcal{C}$ that is hard to learn in a particular model, one can construct a new class $\mathcal{F}$ in which every concept is almost identical to a concept $c \in \mathcal{C}$, but includes the description of c hidden in an exponentially small subset of the domain. This description is encoded in a way that is hard to read in the weaker model of membership oracle, but easy in the stronger model. Then, in the stronger model we can learn the concept class just by reading this additional information, while in the weaker model, with high probability, it is impossible to discover the added description. This reduces learning of $\mathcal{F}$ to learning of $\mathcal{C}$, which is assumed to be hard.

To separate learning with an incomplete membership oracle from learning with a perfect membership oracle, we use a description that requires learning of a large number of bits to discover a single bit of the description. With high probability, an incomplete oracle will fail to uncover any of the bits of the description. The same idea was used by Simon [24].

To separate learning with a noisy membership oracle from learning with an incomplete one, we use a Hadamard code to encode the secret. In addition, via a suitable cryptographic primitive, we "convert" learning with membership queries to learning from random and uniform examples. This encoding makes discovering the secret equivalent to learning of parity functions from random and uniform examples. In particular, it is easy given incomplete labels but hard given noisy labels. We are not aware of any similar techniques having been used before and hope that our technique will find other applications.

1.1 Organization

We define the relevant models in Section 2. Separation of learning with an incomplete membership oracle from learning with a noisy one is presented in Section 3. Separation of learning with an incomplete membership oracle from learning with a perfect membership oracle is presented in Section 4.

2 Preliminaries

We study learning with membership queries in two well-known models of learning: Valiant's PAC learning model [25] and Angluin's exact learning model [1]. We start by giving brief definitions of these models.

In both models a learning algorithm is trying to learn a target concept $c : X \to \{0,1\}$ from a concept class $\mathcal{C}$. The set X is called the domain, and in this work we will assume $X = \{0,1\}^n$. It is assumed that every $c \in \mathcal{C}$ can be described using a fixed representation scheme associated with $\mathcal{C}$ (e.g. Boolean formulas or circuits) such that evaluation of a member r of this scheme takes time polynomial in the representation length. The minimum description length of a concept $c \in \mathcal{C}$ in this representation is denoted by $\mathrm{size}(c)$.

In Angluin's exact learning model, the learning algorithm needs to exactly identify the target concept $c \in \mathcal{C}$ and has access to an equivalence query oracle EQ for c. On a query to the EQ oracle, the algorithm submits any hypothesis h. If $h \equiv c$, then the response YES is returned. Otherwise, a point $x \in X$ such that $h(x) \neq c(x)$ is returned. Note that such an x may be chosen in an adversarial way.

Definition 2.1 *We say that a concept class $\mathcal{C}$ is (efficiently) exactly learnable from equivalence queries if there exists a polynomial $p(\cdot,\cdot)$ and an algorithm $\mathcal{A}$, such that for any target concept $c \in \mathcal{C}$, $\mathcal{A}$, given access to an EQ oracle for c, outputs a hypothesis h evaluatable in time $p(n, \mathrm{size}(c))$ such that $h(x) = c(x)$ for all $x \in X$. Furthermore, $\mathcal{A}$ runs in time $p(n, \mathrm{size}(c))$ and only uses query functions that can be evaluated in time $p(n, \mathrm{size}(c))$.*

In the PAC model, for a concept c and distribution $\mathcal{D}$ over X, an *example oracle* $\mathrm{EX}(c, \mathcal{D})$ is an oracle that, upon request, returns an example $\langle x, c(x) \rangle$ where x is chosen randomly with respect to $\mathcal{D}$. For $\epsilon \geq 0$ we say that a function g ϵ-approximates a function f with respect to distribution $\mathcal{D}$ if $\mathbf{Pr}_{\mathcal{D}}[f(x) = g(x)] \geq 1 - \epsilon$.

Definition 2.2 *For a concept class $\mathcal{C}$, we say that an algorithm $\mathcal{A}$* PAC learns *$\mathcal{C}$, if for every $\epsilon > 0$, $\delta > 0$, $c \in \mathcal{C}$, and distribution $\mathcal{D}$ over X, $\mathcal{A}$ given access to $EX(c, \mathcal{D})$, outputs, with probability at least $1 - \delta$, a hypothesis h that ϵ-approximates c. The learning algorithm is* efficient *if its running time and the time to compute h are polynomial in $n, 1/\epsilon, 1/\delta$, and size(c).*

An algorithm is said to *weakly* learn $\mathcal{C}$ if it produces a hypothesis h that $(\frac{1}{2} - \frac{1}{p(n,\mathrm{size}(c))})$-approximates c for some polynomial p. We say that an algorithm learns $\mathcal{C}$ over a distribution $\mathcal{D}$ if it is only guaranteed to be successful when the examples are drawn with respect to $\mathcal{D}$.

The *random classification noise* model introduced by Angluin and Laird formalizes the simplest type of white label noise [2] in the random examples. In this model for any $\eta \leq 1/2$, called the *noise rate*, the regular example oracle $\mathrm{EX}(c, \mathcal{D})$ is replaced with the noisy oracle $\mathrm{EX}^{\eta}(c, \mathcal{D})$. On each call, $\mathrm{EX}^{\eta}(c, \mathcal{D})$, draws x according to $\mathcal{D}$, and returns $\langle x, c(x)\rangle$ with probability $1-\eta$ and $\langle x, \neg c(x)\rangle$ with probability η. When η approaches 1/2 the label of the corrupted example approaches the result of a random coin flip, and therefore the running times of algorithms in this model are allowed to depend polynomially on $\frac{1}{1-2\eta}$.

2.2 Faulty Membership Oracles

In both models we consider three types of membership oracles. A membership oracle for a function c is the oracle that for every point $x \in X$, returns the value $c(x)$. This basic oracle is commonly thought of as modeling access to a teacher or ability to perform experiments. It was introduced to learning by Valiant [25] and Angluin [1] (for the PAC and the exact models, respectively). To emphasize the fact that this oracle always returns the correct answer, we sometimes refer to it as *perfect*.

Angluin and Slonim introduced a faulty variant of this oracle that addresses the fact that the teacher might not be able to answer some of the questions [3] (this is supported by some experiments with MQ learning algorithms [21]). Specifically, they define an *incomplete membership oracle* with failure probability p, denoted by IMQ^p. For a concept c, whenever $\mathrm{IMQ}^p(c)$ is queried on a point x, with probability p, it responds with $\perp$ and, with probability $1-p$, it responds with $c(x)$. The response $\perp$ corresponds to "I don't know". If the oracle is asked on the same point again, it gives the same response (in other words, the answers of the oracle persist). Note that it is possible (if unlikely) that the oracle will answer $\perp$ to any question asked of it. Therefore we only require a learning algorithm to succeed with probability $1-\delta$ over the coin flips of IMQ^p for some negligible δ, where we define a function $\nu : \mathbb{N} \to \mathbb{R}$ to be *negligible* if for every polynomial $p(n)$ there exists a constant N_p such that $\nu(n) \le \frac{1}{p(n)}$ for $n > N_p$. The running time of an efficient learning algorithm with access to IMQ^p is allowed to depend polynomially on $\frac{1}{1-p}$.

Another variant of faulty membership oracles we address is the *noisy membership oracle*. This oracle was introduced by Goldman et al. in the context of exact identification [14]. A noisy membership oracle with noise rate η, denoted by NMQ^η, is the membership oracle that flips its answer with probability η. That is, for a concept c, when $\mathrm{NMQ}^\eta(c)$ is queried on a point x, it returns $\neg c(x)$ with probability η and $c(x)$ with probability $1-\eta$. As in the case of the incomplete membership oracle, the answers persist and therefore we only require a learning algorithm to succeed with probability $1-\delta$ over the coin flips of NMQ^η for some negligible δ. As in the case of random classification noise, the running time of an efficient learning algorithm with access to NMQ^η is allowed to depend polynomially on $\frac{1}{1-2\eta}$.

3 Separation of Incomplete from Noisy MQ Models

We will now show that learning with noisy membership queries is strictly weaker than learning with incomplete membership queries. First, note that if a concept class is learnable with noisy membership queries, then it can be learned with incomplete membership queries. This follows from the fact that $\mathrm{NMQ}^{\eta}(c)$ can be simulated using $\mathrm{IMQ}^{2\eta}(c)$ by returning the outcome of a fair coin whenever $\mathrm{IMQ}^{2\eta}(c)$ returns "I don't know" and $c(x)$ otherwise (and giving the same label if the same query is made). Note that in this simulation polynomial dependence of the running time of the learning algorithm with access to NMQ^{η} on $1/(1-2\eta)$ ensures that the transformation preserves efficiency.

Our separation results are based on an additional cryptographic assumption. Specifically, we will assume that parities are not PAC learnable with respect to the uniform distribution in the presence of random classification noise. We start by providing several relevant definitions and key facts about this problem.

A parity function $\chi_a(x)$ for a vector $a \in \{0,1\}^n$ is defined as $\chi_a(x) = a \cdot x = \sum_i a_i x_i \pmod 2$. We refer to the vector associated with a parity function as its *index*. We denote the concept class of parity functions $\{\chi_a \mid a \in \{0,1\}^n\}$ by PAR.

Definition 3.1 *The noisy parity problem for noise rate η is the problem of finding the vector $a \in \{0,1\}^n$ given access to $EX^{\eta}(\chi_a, \mathcal{U})$, where $\mathcal{U}$ is the uniform distribution over $\{0,1\}^n$.*

It is well-known that learning a parity with respect to $\mathcal{U}$ in the PAC sense (that is up to accuracy ϵ) is equivalent to finding its index (*cf.* [12]). Another simple observation made by Blum et al. [6] is that the noisy parity problem is randomly self-reducible. That is,

Lemma 3.2 ([6]) *Assume that there exists an efficient algorithm that can solve the noisy parity problem for noise rate η when the target parity function belongs to a subset S of the parity functions on $\{0,1\}^n$, where $|S|/2^n \geq \frac{1}{p(n)}$ for some polynomial p. Then there exists an efficient (randomized) algorithm that can solve the noisy parity problem for noise rate η.*

Blum et al. also prove that if parities are not learnable efficiently then there exist pseudorandom generators [6]. A *pseudorandom generator* $\mathcal{G}$ is a family of functions $G_n : \{0,1\}^n \to \{0,1\}^{g(n)}$ such that for all probabilistic polynomial-time Turing machines T,

$$\left| \Pr_{x \in \{0,1\}^{g(n)}}[T(x) = 1] - \Pr_{x \in \{0,1\}^n}[T(G_n(x)) = 1] \right| \leq \nu(n),$$

where $g(n) : \mathbb{N} \to \mathbb{N}$ is a function such that $g(n) > n$ for all n and ν is a negligible function. Blum et al. prove the following result.

Lemma 3.3 ([6]) *Assume that there exists η such that the noisy parity problem is intractable for noise rate η and $\frac{1}{1-H(\eta)} \leq p(n)$ for some polynomial p and binary entropy function H. Then there exist pseudorandom generators (with $g(n) = n + 1$).*

In particular, by the result of Goldreich, Goldwasser, and Micali, intractability of the noisy parity problem implies existence of *pseudorandom function (PRF) families* [16] that will be a key part of our construction.

Definition 3.4 *A function family $G_{k,n} = \{\sigma_z\}_{z\in\{0,1\}^k}$ (where the key length k is taken to be the security parameter and each σ_z is an efficiently evaluatable function from $\{0,1\}^n$ to $\{0,1\}^n$) is a* pseudorandom function family *if any adversary M (whose resources are bounded by a polynomial in n and k) can distinguish between a function σ_z (where $z \in \{0,1\}^k$ is chosen randomly and kept secret) and a totally random function only with negligible probability. That is, for every probabilistic polynomial time M with oracle access to a function from $\{0,1\}^n$ to $\{0,1\}^n$ and a negligible function $\nu(k)$,*

$$|\mathbf{Pr}[M^{\mathcal{G}_{k,n}}(1^n) = 1] - \mathbf{Pr}[M^{\mathcal{H}_n}(1^n) = 1]| \leq \nu(k),$$

where $\mathcal{G}_{k,n}$ is the random variable produced by choosing $\sigma_z \in G_{k,n}$ for a random and uniform $z \in \{0,1\}^k$ and $\mathcal{H}_n$ is the random variable produced by choosing randomly and uniformly a function from $\{0,1\}^n$ to $\{0,1\}^n$. The probability is taken over the random choice from $\mathcal{G}_{k,n}$ (or $\mathcal{H}_{k,n}$) and the coin flips of M.

We first note that the condition $\frac{1}{1-H(\eta)} \leq p(n)$ for some polynomial p can be replaced by a more standard condition $\frac{1}{1-2\eta} \leq p'(n)$ for some polynomial $p'(n)$.

Lemma 3.5 *If $\frac{1}{1-2\eta} \leq p(n)$ then $\frac{1}{1-H(\eta)} \leq p(n)^2 + c$ for some constant c.*

Proof: Clearly the condition holds for any constant $\eta < 1/4$. Now let $\alpha = 1/2 - \eta$. By definition, $H(\eta) = -\eta \log \eta - (1-\eta)\log(1-\eta)$. Therefore,

$$\begin{aligned} 1 - H(\eta) &= 1 + \left(\frac{1}{2} - \alpha\right)\log\left(\frac{1}{2} - \alpha\right) + \left(\frac{1}{2} + \alpha\right)\log\left(\frac{1}{2} + \alpha\right) \\ &= \left(\frac{1}{2} - \alpha\right)\log(1 - 2\alpha) + \left(\frac{1}{2} + \alpha\right)\log(1 + 2\alpha) \\ &= \log e\left[\left(\frac{1}{2} - \alpha\right)\ln(1 - 2\alpha) + \left(\frac{1}{2} + \alpha\right)\ln(1 + 2\alpha)\right] \end{aligned}$$

The Taylor series expansion for $\ln(1+x)$ is $\sum_{i=1}^{\infty}(-1)^{i+1}\frac{x^i}{i}$. This implies that for positive $\alpha \leq 1/4$, $\ln(1+2\alpha) \geq 2\alpha - 2\alpha^2$ and $\ln(1-2\alpha) \geq -2\alpha - 4\alpha^2$. By substituting this into the above equation, we obtain:

$$1 - H(\eta) \geq \log e(\alpha^2 + 2\alpha^3) \ .$$

But if $\frac{1}{1-2\eta} = \frac{1}{2\alpha} \leq p(n)$ then

$$\frac{1}{1 - H(\eta)} \leq \frac{1}{\log e(\alpha^2 + 2\alpha^3)} \leq p(n)^2 \ .$$

□

The idea behind our separation is the following. It is easy to see that parities are learnable from "incomplete random examples", that is random examples where the learner does not get the label with some probability p. This is true since the learner can just ignore incomplete examples and only use the random examples with labels (which will still be random and uniform). Our goal is, in a sense, to transform membership queries to the target into random examples of the parity function with index a. This is done by creating a function that maps x to a pair $(\sigma_z(x), \chi_a(\sigma_z(x)))$ where σ_z is a function in a pseudorandom function family. Note that this function is not Boolean but can be converted to a Boolean one via a simple trick. The problem with this construction is that in order to learn the given function, the learner would also need to learn σ_z (which is not possible since σ_z is a pseudorandom function). A way to avoid this problem is to have a encode an address in another part of the domain at which one can find the parameter z (one cannot just have $a = z$ since then the adversary could potentially use information about χ_z to "break" the pseudorandom function). One can use redundant encoding (or any other encoding that tolerates erasures) to make sure that the incomplete MQ will suffice to read $\sigma_z(x)$ and z (at location a). Finally we embed this information into an exponentially small subset of the domain and make the remainder pseudorandom (using the same σ_z) so as to make even weak PAC learning impossible without knowing z. In the following theorem we describe the construction that formalizes the above argument and implies Theorems 1.1 and 1.2.

Theorem 3.6 *If the noisy parity problem for noise rate η is intractable and $\frac{1}{1-2\eta}$ is upper-bounded by some polynomial in n, then there exists a concept class $\mathcal{C}$ that is exactly learnable with incomplete membership queries alone, but not weakly PAC learnable with noisy membership queries of error rate η.*

Proof: We define the concept class $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$, where $\mathcal{C}_n$ is defined over $\{0,1\}^{2n}$ as follows. Let $G_{n,n}$ be a pseudorandom family of functions whose existence is implied by Lemma 3.3. Let $a \in \{0,1\}^n$ and χ_a be the corresponding parity on n variables. For each a and $z \in \{0,1\}^n$, define a function $c_{z,a} : \{0,1\}^{2n} \rightarrow \{0,1\}$ as follows. We split the input x into 5 parts: b, w, y, j, and k, where $b \in \{0,1\}$, $w \in \{0,1\}^{\frac{n}{2}}$, $y \in \{0,1\}^n$, $k \in \{0,1\}^\ell$ for $\ell = \lceil \log{(n+1)} \rceil$

and $j \in \{0,1\}^{\frac{n}{2}-1-\ell}$. For convenience, we view j and k as integers given in binary representation.

If $w \neq 0^{\frac{n}{2}}$, $c_{z,a}$ equals the first bit of $\sigma_z(y)$ that we denote by $\sigma_z(y)_1$. Otherwise, for $b = 0$ $c_{z,a}$ encodes a parity on pseudorandom points and for $b = 1$, $c_{z,a}$ encodes z, the secret key to a pseudorandom function family in a "hidden" location that requires knowing a to uncover. Parameter k indexes the bit of $\sigma_z(y)$ or z that is being encoded and parameter j indexes a copy of each of these bits. Exponentially many copies are used to make sure that incomplete membership queries can still be used to read these bits with negligible failure rate. Formally,

$$c_{z,a}(b,w,y,j,k) = \begin{cases} \sigma_z(y)_1 & \text{if } w \neq 0^{\frac{n}{2}} \\ \chi_a(\sigma_z(y)) & \text{if } w = 0^{\frac{n}{2}} \text{ and } b = j = k = 0 \\ k\text{-th bit of } \sigma_z(y) & \text{if } w = 0^{\frac{n}{2}}, b = 0 \text{ and } 1 \leq k \leq n \\ k\text{-th bit of } z & \text{if } w = 0^{\frac{n}{2}}, b = 1, y = a, \text{ and } 1 \leq k \leq n \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We define $\mathcal{C}_n = \{c_{z,a} \mid z, a \in \{0,1\}^n\}$.

Lemma 3.7 *The concept class $\mathcal{C}$ is exactly learnable from incomplete membership queries.*

Proof: Let $\frac{1}{\tau} = 1-p$ be the success rate of the given IMQ^p oracle. The learning algorithm $\mathcal{A}$ chooses $y_1 \in \{0,1\}^n$ randomly and uniformly and attempts to get $\chi_a(\sigma_z(y_1))$ by querying IMQ$^p(c_{z,a})$ on point $(0, 0^{\frac{n}{2}}, y_1, 0, 0)$. Then for every $1 \leq k \leq n$, it attempts to find the k^{th} bit of $\sigma_z(y_1)$ by querying IMQ$^p(c_{z,a})$ on point $(0, 0^{\frac{n}{2}}, y_1, j, k)$ for $j = 0, 1, \ldots, s(n)$, where $s(n) = 4n\tau$. Thus the probability that the k^{th} bit is not obtained is $(1 - \frac{1}{\tau})^{s(n)} \leq \exp\left(-s(n)/\tau\right) = \exp(-4n)$. This process is repeated $s(n)$ times to obtain $\sigma_z(y_1), \ldots, \sigma_z(y_{s(n)})$ and, the corresponding labels. By the union bound, the probability that any bit of any $\sigma_z(y_i)$ is not obtained is at most $s(n)\exp(-4n)$.

Let $X = \sum_{i=1}^{s(n)} X_i$, where X_i is the event that $\chi_a(\sigma_z(y_i))$ is successfully obtained. For distinct y_i's these events are independent and $\mathbf{E}[X] = s(n)/\tau = 4n$. Therefore, by the multiplicative Chernoff bound [11],

$$\Pr[X < 2n] = \Pr[X < (1 - \frac{1}{2})\mathbf{E}[X]] \leq \exp(-s(n)/(8\tau)) \leq \exp(-n/2),$$

a negligible function. Note that the probability that not all y_i's are distinct is upper-bounded by $s^2(n)/2^n = O(\tau^2 \cdot n^2 \cdot 2^{-n})$. Therefore if $\tau = O(2^{n/3})$ then this probability is negligible. In this case $s(n) < 2^{n/2-\ell-1}$ and therefore there are enough distinct copies of each bit of $\sigma_z(y_i)$ for the above recovery scheme to work and the failure probability $s(n)\exp(-4n)$ is a negligible function. If $\tau = \Omega(2^{n/3})$ then the brute-force algorithm that learns the unknown concept point-by-point (which is possible in both PAC and exact models) would have running time polynomial in $\tau = \frac{1}{1-p}$.

The algorithm $\mathcal{A}$ next checks whether the vectors $\sigma_z(y_i)$ for which $X_i = 1$ span $\{0,1\}^n$ (viewed as a vector space over GF(2)), and if so, computes the index a of the parity via Gaussian elimination. If this succeeds, the algorithm queries $\mathrm{IMQ}^p(c_{z,a})$ on points of the form $(1, 0^{\frac{n}{2}}, a, j, k)$ for all $j \leq n$ and $k \leq s(n)$ to obtain the value of z with negligible failure probability in the same way as each $\sigma_z(y_i)$ is obtained. If any of the steps of the algorithm fail, the algorithm outputs $\perp$.

We claim that $\mathcal{A}$ learns $\mathcal{C}$ with negligible failure rate. First we claim that $2n$ uniformly random elements $v_1, \ldots, v_{2n}$ of $\{0,1\}^n$ fail to span the space with negligible probability. If these vectors fail to span the space, they lie in some subspace of dimension $n-1$. There are $2^n - 1$ subspaces of dimension $n-1$. The probability that all $2n$ vectors lie in any particular subspace is 2^{-2n}. By the union bound, the probability that $v_1, \ldots, v_{2n}$ fail to span $\{0,1\}^n$ is upper-bounded by $2^{-2n} \cdot (2^n - 1) \leq 2^{-n}$, a negligible function. This implies that the vectors $\sigma_z(y_i)$ for which $X_i = 1$ fail to span $\{0,1\}^n$ with probability at most $2^{-n} + \nu(n)$ for some negligible $\nu(n)$. This is true since otherwise, values of σ_z on randomly chosen points $y_1, \ldots, y_{s(n)}$ could be efficiently distinguished from truly random and uniform points with non-negligible probability. Therefore $\mathcal{A}$ fails to compute a with negligible probability. The probability that $\mathcal{A}$ fails to compute z after computing a was shown above to be negligible as well and hence the total failure probability of $\mathcal{A}$ on every $c_{z,a}$ is negligible. It is also easy to verify that $\mathcal{A}$ runs in time polynomial in τ and n. □ (Lemma 3.7)

Lemma 3.8 *Under the assumption of Theorem 3.6, the concept class $\mathcal{C}$ is not weakly PAC learnable with noisy membership queries of error rate η.*

Proof: We claim that if $\mathcal{C}$ can be efficiently learned from random examples and NMQ^η by an algorithm $\mathcal{A}$, then we can either:

- Learn parities with noise η.
- Distinguish a function randomly selected from our PRF family from a truly random function.

The latter would also imply learning of parities with noise by Lemma 3.3 of Blum et al. [6].

Assume that there exists an algorithm $\mathcal{A}$ that for every $c \in \mathcal{C}$, given access to random and uniform examples of c and queries to $\mathrm{NMQ}^\eta(c)$ produces a hypothesis h that $(\frac{1}{2} - \frac{1}{q(n)})$-approximates c for some polynomial $q(n)$. Using $\mathcal{A}$ we will build a distinguishing test T with oracle access to a function σ :

$\{0,1\}^n \to \{0,1\}^n$.

The algorithm T first chooses a random $a \in \{0,1\}^n$ and then simulates the algorithm $\mathcal{A}$ with $\delta = \frac{1}{4}$. The algorithm T handles queries from $\mathcal{A}$ as follows:

- **Random examples**: T chooses a random point $x = (b, w, y, j, k)$ in the instance space. If $w \neq 0^{\frac{n}{2}}$ then T returns example $\langle x, \sigma(y)_1 \rangle$. Otherwise, T stops and outputs 0.
- **Membership queries**: If $y = a$ then T returns 1 and stops. Otherwise, T computes $c_{z,a}(b, w, y, j, k)$ according to equation (1) while using σ in place of σ_z, and uses randomness to simulate random persistent noise.

Let h be the hypothesis that $\mathcal{A}$ outputs. T estimates the error of $h(b, w, y, j, k)$ on $\sigma(y)_1$ within $\frac{1}{4q(n)}$ by using $O(nq^2(n))$ random and uniform points. Chernoff bounds imply that the estimate will be correct with probability at least $1 - \nu_1(n)$ for some negligible $\nu_1(n)$ [11]. T returns 1 if the estimate of the error of h on $\sigma(y)_1$ is at most $\frac{1}{2} - \frac{1}{2q(n)}$ and 0 otherwise.

We now claim that T returns 1 with probability $\geq \frac{3}{4} - \nu(n)$ when it has oracle access to a function σ_z randomly chosen from $G_{n,n}$, where $\nu(n)$ is negligible. If $\sigma = \sigma_z$ then the oracles provided by the simulation are valid oracles for $c_{z,a}(x)$ until either a membership query with $y = a$ is made or a random example with $w = 0^{\frac{n}{2}}$ is generated. In the first case T outputs 1. The probability that for any polynomial number of uniform random examples there exists an example with $w = 0^{\frac{n}{2}}$ is negligible. If neither of these events happens then, with probability at least $\frac{3}{4}$, $\mathcal{A}$ has to output a hypothesis h that $(\frac{1}{2} - \frac{1}{q(n)})$-approximates $c_{z,a}$. The function $c_{z,a}(b, w, y, j, k)$ differs from $\sigma_z(y)_1$ only when $w = 0^{\frac{n}{2}}$ and therefore h $(\frac{1}{2} - \frac{1}{q(n)} - 2^{-\frac{n}{2}})$-approximates $\sigma_z(y)_1$. This implies that if $\mathcal{A}$ is successful and the estimate of the error of h is correct then T will return 1 and, in particular, $\mathbf{Pr}[T^{\mathcal{G}_{n,n}}(1^n) = 1] \geq \frac{3}{4} - \nu(n)$ for some negligible $\nu(n)$.

Now let $\mathcal{H}_n$ be the uniform distribution over functions from $\{0,1\}^n$ to $\{0,1\}^n$, that is, σ is a truly randomly chosen function. If $\mathbf{Pr}[T^{\mathcal{H}_n}(1^n) = 1] \leq \frac{1}{2}$ then T is an efficient distinguisher violating the pseudorandomness property of the family $G_{n,n}$. Therefore we can assume that $\mathbf{Pr}[T^{\mathcal{H}_n}(1^n) = 1] \geq \frac{1}{2}$. It is well-known (and can be easily derived using Chernoff bound [11]) that for a randomly chosen σ, the probability that there exists a hypothesis of polynomial size that $(\frac{1}{2} - \frac{1}{4q(n)})$-approximates $\sigma(y)_1$ is negligible. Therefore the probability that T outputs 1 is upper-bounded by the probability that during the simulation $\mathcal{A}$ asks a membership query with $y = a$ plus some negligible $\nu(n)$ (that also accounts for the probability that the estimate of the error of h is not within $\frac{1}{4q(n)}$). We now claim that $\mathcal{A}$ needs to solve the noisy parity problem to ask a membership query with $y = a$ and therefore this cannot happen with significant probability.

We design a learner M for parities with noise that works as follows. Let $\mathrm{EX}^{\eta}(\chi_{a'},\mathcal{U})$ be the oracle given to M. M simulates $\mathcal{A}$ in the same way as $T^{\mathcal{H}_n}(1^n)$ does but uses random examples from $\mathrm{EX}^{\eta}(\chi_{a'},\mathcal{U})$ in place of noisy examples of a randomly chosen parity function χ_a. Formally, to produce the output of a membership query on a point (b, w, y, j, k) for $\mathcal{A}$, M gets a random example $\langle y', v'\rangle$ from $\mathrm{EX}^{\eta}(\chi_{a'},\mathcal{U})$. If $w = 0^{\frac{n}{2}}$ and $b = j = k = 0$ then M replies with v' otherwise M replies with value v corrupted with random persistent noise of rate η, where

$$v = \begin{cases} \text{first bit of } y' & \text{if } w \neq 0^{\frac{n}{2}} \\ k\text{-th bit of } y' & \text{if } b = 0, w = 0^{\frac{n}{2}} \text{ and } 1 \leq k \leq n \\ 0 & \text{otherwise} \end{cases}$$

Random examples are handled in the same way (but without the noise). As before, if a random example with $w = 0^{\frac{n}{2}}$ is generated the simulation is terminated. If M gets a membership query or produces an example for y which has already been queried or generated via a random example then the same example $\langle y', v'\rangle$ is used as in the first occurrence. Finally, to test if $y = a'$, M tests the hypothesis χ_y on random examples from $\mathrm{EX}^{\eta}(\chi_{a'},\mathcal{U})$. A standard application of Chernoff bound implies that a polynomial in n and $\frac{1}{1-2\eta}$ number of examples is sufficient to ensure that the probability of an outcome of such a test being incorrect is negligible [11]. If the outcome of the test is positive then M returns y. It is straightforward to verify that, conditioned on the results of all these tests being correct, M with access to $\mathrm{EX}^{\eta}(\chi_{a'},\mathcal{U})$ produces exactly the same simulation of $\mathcal{A}$ as $T^{\mathcal{H}_n}$ conditioned on $a = a'$. In particular, the probability that M finds a' is equal (up to a negligible function) to the probability that $T^{\mathcal{H}_n}$ conditioned on $a = a'$ outputs 1.

Now let $T^{\mathcal{H}_n}[a']$ denote the execution of $T^{\mathcal{H}_n}$ conditioned on $a = a'$ and let S be the set of all vectors a' for which the success probability of $T^{\mathcal{H}_n}[a']$ is at least 1/4, that is

$$S = \left\{ a' \;\middle|\; \mathbf{Pr}[T^{\mathcal{H}_n}[a'](1^n) = 1] \geq \frac{1}{4} \right\} .$$

By our assumption,

$$\begin{aligned} \frac{1}{2} \leq \mathbf{Pr}[T^{\mathcal{H}_n}(1^n) = 1] &= \mathbf{E}_{a' \in \{0,1\}^n} \left[\mathbf{Pr}[T^{\mathcal{H}_n}[a'](1^n) = 1] \right] \\ &\leq \mathbf{Pr}[a' \in S] + \frac{1}{4}(1 - \mathbf{Pr}[a' \in S]) = \frac{1}{4} + \frac{3}{4}\mathbf{Pr}[a' \in S] . \end{aligned}$$

This means that $\mathbf{Pr}[a' \in S] \geq \frac{1}{3}$. By combining the arguments above we obtain that for every $a' \in S$, M successfully finds a' given access to $\mathrm{EX}^{\eta}(\chi_{a'},\mathcal{U})$ with probability at least $1/4 - \nu_2(n)$ for some negligible $\nu_2(n)$. This probability can be boosted to $1-\delta$ using the standard confidence boosting procedure (*cf.* [19]). Further, we can use Lemma 3.2 to obtain algorithm M' that efficiently learns all parities in the presence of noise, violating the assumption of Theorem 3.6.

□ (Lemma 3.8)

By combining Lemmas 3.7 and 3.8 we obtain the desired result.

□ (Theorem 3.6)

Theorem 3.6 is stronger than Theorem 1.2. In addition, the standard reduction of exact learning to PAC learning implies that the concept class $\mathcal{C}$ is not exactly learnable from equivalence queries and noisy membership queries of rate η. Therefore Theorem 3.6 also implies Theorem 1.1.

4 Separation of Incomplete from Perfect MQ Models

In this section, we show that the incomplete membership oracle is strictly weaker than the perfect one, in the settings of both PAC and exact learning. An analogue of this result for the special case of proper exact learning was given by Simon [24].

We begin by describing the main idea. Given a concept class $\mathcal{C}$ that is hard to PAC learn, one can construct a new class $\mathcal{F}$ in which every concept is almost identical to a concept $c \in \mathcal{C}$, but includes the description of c hidden in an exponentially small subset of the domain. We encode each bit of the description of c as a XOR of a large number of bits, each of which is allowed to range arbitrarily given this single linear constraint. If just one bit is missed, the entirety of this information becomes useless. With high probability, an incomplete oracle will miss at least one bit of the XOR and, therefore, learning with incomplete MQs is, with high probability, equivalent to learning $\mathcal{C}$, which was assumed hard.

Theorem 4.1 *Suppose a concept class $\mathcal{C}$ cannot be efficiently PAC learned over the uniform distribution with membership queries. For any polynomial $r(n)$, there exists a concept class $\mathcal{F}$ that is efficiently PAC learnable over the uniform distribution with membership queries, but is not efficiently PAC learnable over the uniform distribution with access to* IMQ^p*, for $p = 1/r(n)$.*

Proof: We define the concept class $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$, where $\mathcal{F}_n$ is defined over $\{0,1\}^{n+1}$ as follows. First, we assume (without loss of generality) that all concepts in $\mathcal{C}_n$ have description length exactly $s(n) = 2^{o(n)}$. Concepts of size $2^{\Omega(n)}$ can be learned efficiently in time polynomial in 2^n in the trivial way.

For $c \in \mathcal{C}_n$ let c_i denote the i^{th} bit of the description of c. Let $t(n) = n/p = nr(n)$. For convenience, we view the domain $\{0,1\}^{n+1}$ as $\{0,1\} \times \{0,1\}^{\log s(n)} \times \{0,1\}^{\log t(n)} \times \{0,1\}^{n-\log(s(n)t(n))}$, and refer to a point in the domain by a quadruple (b, i, j, k). Similarly, we also refer to a point in $\{0,1\}^{n+1}$ as a pair (b, x) where $b \in \{0,1\}$ and $x \in \{0,1\}^n$. For a binary string $u \in \{0,1\}^{s(n)t(n)}$, we refer to its bits by $u_{i,j}$ where $i \in [s(n)]$ and $j \in [t(n)]$. For a concept $c \in \mathcal{C}_n$, $u \in \{0,1\}^{s(n)t(n)}$ and a point $y = (b, x) = (b, i, j, k) \in \{0,1\}^{n+1}$ we define

$$f_{c,u}(y) = \begin{cases} c(x) & b = 0 \\ 0 & b = 1, k \neq 0^{n-\log(s(n)t(n))} \\ u_{i,j} & b = 1, k = 0^{n-\log(s(n)t(n))} \end{cases} \tag{2}$$

Concept class $\mathcal{F}_n$ consists of all $f_{c,u}$ such that $c \in \mathcal{C}_n$ and $u \in \{0,1\}^{s(n)t(n)}$ satisfies the constraint

$$c_i = \bigoplus_{j=1}^{t(n)} u_{i,j}, \tag{3}$$

for every $i \in [s(n)]$. Any function $f_{c,u} \in \mathcal{F}_n$ is evaluatable in polynomial time in its description length (which equals $t(n)s(n)$).

We will first prove that $\mathcal{F}$ can be learned using membership queries alone. Define an algorithm $\mathcal{A}$ as follows: $\mathcal{A}$ asks membership queries on $(1, i, j, 0^{n-\log(s(n)t(n))})$ to find the values $u_{i,j}$ for all $i \in [s(n)]$ and $j \in [t(n)]$. Then, by computing $c_i = \bigoplus_{j=1}^{t(n)} u_{i,j}$, the algorithm computes the concept c and thereby finds $f_{c,u}$. Thus our algorithm learns $\mathcal{F}$ efficiently and exactly from membership queries alone. In particular, it is a PAC learning algorithm for any distribution over the domain.

For the second part of the claim, let $\mathcal{A}'$ be any algorithm that efficiently learns $\mathcal{F}$ in the PAC model over the uniform distribution with access to IMQ^p. We construct an algorithm $\mathcal{A}$ to efficiently learn $\mathcal{C}$ in the PAC model over the uniform distribution with membership queries. Let ϵ and δ be the parameters of $\mathcal{A}$ and let $\text{EX}(c, \mathcal{U})$ and $\text{MQ}(c)$ be the example oracle and membership oracle to which $\mathcal{A}$ has access. We can assume that $\delta = 2^{-o(n)}$ and $\epsilon = 2^{-o(n)}$, since if say $\delta = 2^{-\Omega(n)}$, then the trivial learning algorithm for $\mathcal{C}$ is polynomial in $\frac{1}{\delta}$. The algorithm $\mathcal{A}$ works as follows:

(1) Choose $u' \in \{0,1\}^{s(n)t(n)}$ randomly and uniformly.
(2) Run $\mathcal{A}'$ with parameters $\frac{\epsilon}{2}$ and $\frac{\delta}{2}$ by simulating oracles $\text{EX}(f_{c,u'}, \mathcal{U})$ and $\text{MQ}(f_{c,u'})$ as detailed below.
(3) Return $h(x) \equiv h'(0, x)$, where h' is the output of $\mathcal{A}'$.

Whenever $\mathcal{A}'$ requests an example from $\text{EX}(f_{c,u'}, \mathcal{U})$, $\mathcal{A}$ flips a coin. If the coin comes up heads, it requests an example $\langle x, v \rangle$ from $\text{EX}(c, \mathcal{U})$ and returns $\langle (0, x), v \rangle$ to $\mathcal{A}'$. If the coin comes up tails, $\mathcal{A}$ returns $\langle x, f_{c,u'}(1, x) \rangle$, as defined in equation (2). On a membership query (b, x), $\mathcal{A}$ returns $\perp$ with probability p (persistently) and value $f_{c,u'}(b,x)$ with probability $1-p$. To compute $f_{c,u'}(b,x)$ when $b = 0$, $\mathcal{A}$ uses a membership query to $\mathrm{MQ}(c)$ on x.

Let $\delta^* = \delta_1 + \delta_2$, where δ_1 and δ_2 are the probabilities, respectively, that the following events occur.

(1) There exists i such that for all $j \in [t(n)]$, $\mathcal{A}'$ asked for a membership query on $(1, i, j, 0^{n-\log(s(n)t(n))})$ and obtained $u'_{i,j}$.
(2) $\mathcal{A}'$ received the value at $(1, i, j, k)$ for $k = 0^{n-\log(s(n)t(n))}$ as a random example.

If neither of these events occur, there exists u that agrees with all the answers that $\mathcal{A}$ has provided using u' and satisfies the constraints in equation (3). Thus with probability $\geq 1-\delta^*$, the answers of $\mathcal{A}$ are consistent with the simulation of $\mathcal{A}'$ using oracles $\mathrm{EX}(f_{c,u}, \mathcal{U})$ and $\mathrm{MQ}(f_{c,u})$. Therefore, with probability at least $1 - \delta^* - \frac{\delta}{2}$, $\mathcal{A}'$ returns h' that $\frac{\epsilon}{2}$-approximates $f_{c,u}$ with respect to the uniform distribution over $\{0,1\}^n$. The function $f_{c,u}$ equals c when $b = 0$ and therefore the hypothesis h returned by $\mathcal{A}$ ϵ-approximates c with respect to the uniform distribution over $\{0,1\}^n$.

We will now show that for sufficiently large n, $\delta^* \leq \frac{\delta}{2}$ and hence the success probability of $\mathcal{A}$ is at least $1 - \delta$. Each $u'_{i,j}$ is returned with probability $1 - p$ and therefore the probability that event 1 occurs for a fixed $i \in [s(n)]$ is $(1-p)^{t(n)} \leq \exp(-p \cdot t(n)) = \exp(-n)$. By the union bound, the probability that event 1 occurs is at most $s(n)\exp(-n) \leq \delta/4$.

To bound δ_2, observe that the probability that for a random point (b, i, j, k), $k = 0^{n-\log(s(n)t(n))}$ is $s(n)t(n)/2^n$. Therefore $\delta_2 \leq q(n)s(n)t(n)/2^n$. By our assumption, $s(n), 1/\epsilon$ and $1/\delta$ are $2^{o(n)}$. The running time of $\mathcal{A}'$ is polynomial in n, $s(n)t(n) = nr(n)s(n)$, $1/\epsilon$ and $1/\delta$ and therefore $q(n)$ is $2^{o(n)}$. This implies that $\delta_2 \leq 2^{-n/2} \leq \delta/4$ for sufficiently large n. Finally note that the running time of $\mathcal{A}$ is polynomial in n, $s(n)t(n) = nr(n)s(n)$, $1/\epsilon$ and $1/\delta$ and hence $\mathcal{A}$ is an efficient PAC learning algorithm for $\mathcal{C}$. This contradicts our assumption and therefore implies that $\mathcal{F}$ is not efficiently PAC learnable over the uniform distribution with access to IMQ^p. □

Corollary 4.2 *Assume that there exists a concept class $\mathcal{C}$ that is not exactly learnable from equivalence and membership queries. Then, for any polynomial $r(n)$, there exists a concept class $\mathcal{F}$ that is exactly learnable from equivalence and membership queries, but is not exactly learnable from equivalence queries and incomplete membership queries with failure rate $p = 1/r(n)$.*

Proof: We construct the class $\mathcal{F}$ exactly as in the proof of Theorem 4.1. The learning algorithm for $\mathcal{F}$ that we described is exact and uses only membership queries. Therefore we get that $\mathcal{F}$ is learnable in the exact model with membership queries.

We also use the learning algorithm $\mathcal{A}'$ for $\mathcal{F}$ to construct $\mathcal{A}$ that learns $\mathcal{C}$ as before, but replace handling of random examples with analogous handing of equivalence queries. If $\mathcal{A}'$ makes an equivalence query g', then $\mathcal{A}$ submits the equivalence query $g(x) \equiv g'(0,x)$ to its equivalence oracle. Given a counterexample $\langle x, v\rangle$, $\mathcal{A}$ returns $\langle (0,x), v\rangle$ as a counterexample to $\mathcal{A}'$. Event 2 does not occur and therefore we only need to ensure that event 1 occurs with some negligible probability. As we have showed, $\delta_1 \leq s(n)\exp(-n)$ and therefore, by analogous analysis, $\mathcal{A}$ succeeds with probability at least $1-\nu(n)$ for some negligible $\nu(n)$. This contradicts our assumption and implies the desired result. □

These separation results are optimal in the following sense. Every concept class that is efficiently learnable in the exact model with perfect MQs is also efficiently exactly learnable with EQ and $\mathrm{IMQ}^{1/r(n)}$ for sufficiently large polynomial $r(n)$. This is true since for a sufficiently low omission rate, with high probability, the learner will not encounter any omissions in the answers to a polynomial number of membership queries.

Corollary 4.3 *If one-way functions exist, then PAC learning with incomplete membership queries is strictly harder than PAC learning with perfect membership queries. Similarly, exact learning with equivalence and incomplete membership queries is strictly harder than exact learning with equivalence and perfect membership queries.*

Proof: Valiant observed that if one-way functions exist, then polynomial size circuits are hard to learn in the PAC model with membership queries [25]. Since any concept class learnable in the exact model may also be learned in the PAC model with membership queries, polynomial size circuits are also hard to learn in the exact model under the assumption that one-way functions exist. This gives us the desired results by Theorem 4.1 and Corollary 4.2. □

5 Concluding Remarks

In this paper, we gave two separation results for exact learning with faulty membership queries. Perhaps the most interesting aspect of the second separation result is a surprising connection to learning of parities in the PAC model with noise. It appears to be the first result that is based on the intractability of the noisy parity problem. An interesting related question is whether this assumption can be replaced by a more general complexity theoretic assumption.

Acknowledgments

We would like to thank Nader Bshouty and Leslie Valiant for useful discussions on this research. We are grateful to Salil Vadhan for contributing an important insight to the proof of Theorem 1.1. We also thank Neal Wadhwa for his help in the write-up of an early version of these results.

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

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

[3] D. Angluin and D. K. Slonim. Randomly fallible teachers: Learning monotone DNF with an incomplete membership oracle. *Machine Learning*, 14(1):7–26, 1994.

[4] A. Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(046), 1997.

[5] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24, 1978.

[6] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Proceedings of International Cryptology Conference on Advances in Cryptology (CRYPTO)*, pages 278–291, 1993.

[7] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *Proceedings of STOC*, pages 435–440, 2000.

[8] N. Bshouty and N. Eiron. Learning monotone DNF from a teacher that almost does not answer membership queries. *Journal of Machine Learning Research*, 3:49–57, March 2003.

[9] N. H. Bshouty and A. Owshanko. Learning regular sets with an incomplete membership oracle. In *Proceedings of COLT '01/EuroCOLT '01*, pages 574–588, 2001.

[10] Z. Chen. A note on learning DNF formulas using equivalence and incomplete membership queries. In *AII '94: Proceedings of the 4th International Workshop on Analogical and Inductive Inference*, pages 272–281. Springer-Verlag, 1994.

[11] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statist.*, 23:493–507, 1952.

[12] V. Feldman. Attribute efficient and non-adaptive learning of parities and DNF expressions. *Journal of Machine Learning Research*, (8):1431–1460, 2007.

[13] V. Feldman, P. Gopalan, S. Khot, and A. Ponuswami. New results for learning noisy parities and halfspaces. In *Proceedings of FOCS*, pages 563–574, 2006.

[14] S. Goldman, M. Kearns, and R. Schapire. Exact identification of read-once formulas using fixed points of amplification functions. *SIAM Journal on Computing*, 22(4):705–726, 1993.

[15] S. Goldman and H. Mathias. Learning k-term DNF formulas with an incomplete membership oracle. In *Proceedings of COLT*, pages 77–84, 1992.

[16] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[17] J. Jackson, E. Shamir, and C. Shwartzman. Learning with queries corrupted by classification noise. In *Proceedings of the Fifth Israel Symposium on the Theory of Computing Systems*, pages 45–53, 1997.

[18] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[19] M. Kearns and U. Vazirani. *An introduction to computational learning theory.* MIT Press, Cambridge, MA, 1994.

[20] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[21] K.J. Lang and E.B. Baum. Query learning can work poorly when a human oracle is used. In *Proceedings of International Joint Conference on Neural Networks*, pages 609–614, 1992.

[22] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42-44, 1978.

[23] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proccedings of STOC*, pages 84–93, 2005.

[24] H. Simon. How many missing answers can be tolerated by query learners? *Theory of Computing Systems*, 37(1):77–94, 2004.

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

[26] A. Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *Proceedings of STOC*, pages 92–109, 1997.