Journal of Machine Learning Research 10 (2009) 163-182 Submitted 9/08; Published 2/09

On The Power of Membership Queries in Agnostic Learning[*]

Vitaly Feldman[†] VITALY@POST.HARVARD.EDU
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120

Editor: Rocco Servedio

Abstract

We study the properties of the agnostic learning framework of Haussler (1992) and Kearns, Schapire, and Sellie (1994). In particular, we address the question: is there any situation in which membership queries are useful in agnostic learning?

Our results show that the answer is negative for distribution-independent agnostic learning and positive for agnostic learning with respect to a specific marginal distribution. Namely, we give a simple proof that any concept class learnable agnostically by a distribution-independent algorithm with access to membership queries is also learnable agnostically without membership queries. This resolves an open problem posed by Kearns et al. (1994). For agnostic learning with respect to the uniform distribution over $\{0,1\}^n$ we show a concept class that is learnable with membership queries but computationally hard to learn from random examples alone (assuming that one-way functions exist).

Keywords: agnostic learning, membership query, separation, PAC learning

1. Introduction

The agnostic framework (Haussler, 1992; Kearns et al., 1994) is a natural generalization of Valiant's PAC learning model (Valiant, 1984). In this model no assumptions are made on the labels of the examples given to the learning algorithm, in other words, the learning algorithm has no prior beliefs about the target concept (and hence the name of the model). The goal of the agnostic learning algorithm for a concept class $\mathcal{C}$ is to produce a hypothesis h whose error on the target concept is close to the best possible by a concept from $\mathcal{C}$. This model reflects a common empirical approach to learning, where few or no assumptions are made on the process that generates the examples and a limited space of candidate hypothesis functions is searched in an attempt to find the best approximation to the given data.

Designing algorithms that learn efficiently in this model is notoriously hard and very few positive results are known (Kearns et al., 1994; Lee et al., 1995; Goldman et al., 2001; Gopalan et al., 2008; Kalai et al., 2008a,b). Furthermore, strong computational hardness results are known for agnostic learning of even the simplest classes of functions such as parities, monomials and halfspaces (Håstad, 2001; Feldman, 2006; Feldman et al., 2006; Guruswami and Raghavendra, 2006) (albeit only for *proper* learning). Reductions from long-standing open problems for PAC learning to ag-

[*] Parts of this work are published in the Proceedings of 18th Annual Conference on Learning Theory, 2008.

[†] Part of the work done while the author was at Harvard University supported by grants from the National Science Foundation NSF-CCF-04-32037 and NSF-CCF-04-27129.

nostic learning of simple classes of functions provide another indication of the hardness of agnostic learning (Kearns et al., 1994; Kalai et al., 2008a; Feldman et al., 2006).

A membership oracle allows a learning algorithm to obtain the value of the unknown target function f on any point in the domain. It can be thought of as modeling the access to an expert or ability to conduct experiments. Learning with membership queries in both PAC and Angluin's exact models (Angluin, 1988) was studied in numerous works. For example monotone DNF formulas, finite automata and decision trees are only known to be learnable with membership queries (Valiant, 1984; Angluin, 1988; Bshouty, 1995). It is well-known and easy to prove that the PAC model with membership queries is strictly stronger than the PAC model without membership queries (if one-way functions exist).

Membership queries are also used in several agnostic learning algorithms. The first one is the famous algorithm of Goldreich and Levin (1989) introduced in a cryptographic context (even before the definition of the agnostic learning model). Their algorithm learns parities agnostically with respect to the uniform distribution using membership queries. Kushilevitz and Mansour (1993) used this algorithm to PAC learn decision trees and it has since found numerous other significant applications. More efficient versions of this algorithm were also given by Levin (1993), Bshouty, Jackson, and Tamon (2004) and Feldman (2007). Recently, Gopalan, Kalai, and Klivans (2008) gave an elegant algorithm that learns decision trees agnostically over the uniform distribution and uses membership queries.

1.1 Our Contribution

In this work we study the power of membership queries in the agnostic learning model. The question of whether or not membership queries can aid in agnostic learning was first asked by Kearns et al. (1994) who conjectured that the answer is no. To the best of our knowledge, the question has not been addressed prior to our work. We present two results on this question. In the first result we prove that every concept class learnable agnostically with membership queries is also learnable agnostically without membership queries (see Th. 6 for a formal statement). This proves the conjecture of Kearns et al. (1994). The reduction we give modifies the distribution of examples and therefore is only valid for distribution-independent learning, that is, when a single learning algorithm is used for every distribution over the examples. The simple proof of this result explains why the known distribution-independent agnostic learning algorithm do not use membership queries (Kearns et al., 1994; Kalai et al., 2008a,b).

The proof of this result also shows equivalence of two standard agnostic models: the one in which examples are labeled by an unrestricted function and the one in which examples come from a joint distribution over the domain and the labels.

Our second result is a proof that there exists a concept class that is agnostically learnable with membership queries over the uniform distribution on $\{0,1\}^n$ but hard to learn in the same setting without membership queries (see Th. 8 for a formal statement). This result is based on the most basic cryptographic assumption, namely the existence of one-way functions. Note that an unconditional separation of these two models would imply $\mathsf{NP} \neq \mathsf{P}$. Cryptographic assumptions are essential for numerous other hardness results in learning theory (cf., Kearns and Valiant, 1994; Kharitonov, 1995). Our construction is based on the use of pseudorandom function families, list-decodable codes and a variant of an idea from the work of Elbaz, Lee, Servedio, and Wan (2007). Sections 4.1 and 4.2 describe the technique and its relation to prior work in more detail.

This results is, perhaps, unsurprising since agnostic learning of parities with respect to the uniform distribution from random examples only is commonly considered hard and is known to be equivalent to learning of parities with random noise (Feldman et al., 2006), a long standing open problem which itself is equivalent to decoding of random linear codes, a long-standing open problem in coding theory. The best known algorithm for this problem runs in time $O(2^{n/\log n})$ (Blum et al., 2003; Feldman et al., 2006). If one assumes that learning of parities with noise is intractable then it immediately follows that membership queries are provably helpful in agnostic learning over the uniform distribution on $\{0,1\}^n$. The goal of our result is to replace this assumption by a possibly weaker and more general cryptographic assumption. It is known that if learning of parities with noise is hard then one-way functions exist (Blum et al., 1993) but, for all we know, it is possible that the converse is not true. The proof of our result however is substantially less straightforward than one might expect (and than the analogous separation for PAC learning). Here the main obstacle is the same as in proving positive results for agnostic learning: the requirements of the model impose severe limits on concept classes for which the agnostic guarantees can be provably satisfied.

1.2 Organization

Following the preliminaries, our first result is described in Section 3. The second result appears in Section 4.

2. Preliminaries

Let X denote the domain or the *input space* of a learning problem. The domain of the problems that we study is $\{0,1\}^n$, or the n-dimensional *Boolean hypercube*. A *concept* over X is a $\{-1,1\}$ function over the domain and a *concept class* $\mathcal{C}$ is a set of concepts over X. The unknown function $f \in \mathcal{C}$ that a learning algorithm is trying to learn is referred to as the *target concept*.

A parity function is a function equal to the *XOR* of some subset of variables. For a Boolean vector $a \in \{0,1\}^n$ we define the parity function $\chi_a(x) = (-1)^{a \cdot x} = (-1)^{\oplus_{i \leq n} a_i x_i}$. We denote the concept class of parity functions $\{\chi_a \mid a \in \{0,1\}^n\}$ by PAR. A *k-junta* is a function that depends only on k variables.

A *representation class* is a concept class defined by providing a specific way to represent each function in the concept class. In fact all the classes of functions that we discuss are representation classes. We often refer to a representation class simply as concept class when the representation is implicit in the description of the class. For a representation class $\mathcal{F}$, we say that an algorithm outputs $f \in \mathcal{F}$ if the algorithm outputs f in the representation associated with $\mathcal{F}$.

2.1 PAC Learning Model

The learning models discussed in this work are based on Valiant's well-known PAC model (Valiant, 1984). In this model, for a concept f and distribution D over X, an *example oracle* $\mathrm{EX}(D, f)$ is the oracle that, upon request, returns an example $\langle x, f(x) \rangle$ where x is chosen randomly with respect to D. For $\varepsilon \geq 0$ we say that a function g ε-approximates a function f with respect to distribution D if $\mathbf{Pr}_D[f(x) = g(x)] \geq 1 - \varepsilon$. In the PAC learning model the learner is given access to $\mathrm{EX}(D, f)$ where f is assumed to belong to a fixed concept class $\mathcal{C}$.

Definition 1 *For a representation class $\mathcal{C}$, we say that an algorithm* `Alg` PAC learns *$\mathcal{C}$, if for every $\varepsilon > 0$, $\delta > 0$, $f \in \mathcal{C}$, and distribution D over X,* `Alg`*, given access to $EX(D, f)$, outputs, with probability at least $1-\delta$, a hypothesis h that ε-approximates f.*

The learning algorithm is *efficient* if its running time and the time to evaluate h are polynomial in $1/\varepsilon, 1/\delta$ and the *size* σ of the learning problem. Here by the size we refer to the maximum description length of an element in X (e.g., n when $X = \{0,1\}^n$) plus the maximum description length of an element in $\mathcal{C}$ in the representation associated with $\mathcal{C}$.

An algorithm is said to *weakly* learn $\mathcal{C}$ if it produces a hypothesis h that $(\frac{1}{2} - \frac{1}{p(\sigma)})$-approximates f for some polynomial $p(\cdot)$.

2.2 Agnostic Learning Model

The *agnostic* learning model was introduced by Haussler (1992) and Kearns et al. (1994) in order to model situations in which the assumption that examples are labeled by some $f \in \mathcal{C}$ does not hold. In its least restricted version the examples are generated from some unknown distribution A over $X \times \{-1,1\}$. The goal of an agnostic learning algorithm for a concept class $\mathcal{C}$ is to produce a hypothesis whose error on examples generated from A is close to the best possible by a concept from $\mathcal{C}$. Class $\mathcal{C}$ is referred to as the *touchstone* class in this setting. More generally, the model allows specification of the assumptions made by a learning algorithm by describing a set $\mathcal{A}$ of distributions over $X \times \{-1,1\}$ that restricts the distributions over $X \times \{-1,1\}$ seen by a learning algorithm. Such $\mathcal{A}$ is referred to as the *assumption class*. Any distribution A over $X \times \{-1,1\}$ can be described uniquely by its marginal distribution D over X and the expectation of b given x. That is, we refer to a distribution A over $X \times \{-1,1\}$ by a pair (D_A, ϕ_A) where $D_A(z) = \mathbf{Pr}_{\langle x,b\rangle \sim A}[x = z]$ and

$$\phi_A(z) = \mathbf{E}_{\langle x,b\rangle \sim A}[b \mid z = x].$$

Formally, for a Boolean function h and a distribution $A = (D, \phi)$ over $X \times \{-1,1\}$, we define

$$\Delta(A,h) = \mathbf{Pr}_{\langle x,b\rangle \sim A}[h(x) \neq b] = \mathbf{E}_D[|\phi(x) - h(x)|/2] \ .$$

Similarly, for a concept class $\mathcal{C}$, define

$$\Delta(A,\mathcal{C}) = \inf_{h \in \mathcal{C}} \{\Delta(A,h)\} \ .$$

Kearns et al. (1994) define agnostic learning as follows.

Definition 2 *An algorithm* `Alg` agnostically *learns a representation class $\mathcal{C}$ by a representation class $\mathcal{H}$ assuming $\mathcal{A}$ if for every $\varepsilon > 0, \delta > 0, A \in \mathcal{A}$,* `Alg` *given access to examples drawn randomly from A, outputs, with probability at least $1-\delta$, a hypothesis $h \in \mathcal{H}$ such that $\Delta(A,h) \leq \Delta(A,\mathcal{C}) + \varepsilon$.*

The learning algorithm is *efficient* if it runs in time polynomial $1/\varepsilon, \log(1/\delta)$ and σ (the size of the learning problem). If $\mathcal{H} = \mathcal{C}$ then, by analogy with the PAC model, the learning is referred to as *proper*. We drop the reference to $\mathcal{H}$ to indicate that $\mathcal{C}$ is learnable by some representation class.

A number of special cases of the above definition are commonly considered (and often referred to as *the* agnostic learning model). In fully agnostic learning $\mathcal{A}$ is the set of all distributions over $X \times \{-1,1\}$. Another version assumes that examples are labeled by an unrestricted function. That is, the set $\mathcal{A}$ contains distribution $A = (D, f)$ for every Boolean function f and distribution D. Note that access to random examples from $A = (D, f)$ is equivalent to access to $\mathrm{EX}(D, f)$. Following Kearns et al. (1994), we refer to this version as *agnostic PAC learning*. Theorem 6 implies that these versions are essentially equivalent. In *distribution-specific* versions of this model for every $(D, \phi) \in \mathcal{A}$, D equals to some fixed distribution known in advance.

We also note that the agnostic PAC learning model can also be thought of as a model of adversarial classification noise. By definition, a Boolean function g differs from some function $f \in \mathcal{C}$ on $\Delta(g, \mathcal{C})$ fraction of the domain. Therefore g can be thought of as f corrupted by noise of rate $\Delta_D(f, \mathcal{C})$. Unlike in the random classification noise model, the points on which a concept can be corrupted are unrestricted and therefore the noise is referred to as adversarial.

2.2.1 UNIFORM CONVERGENCE

A natural approach to agnostic learning is to first draw a sample of fixed size and then choose a hypothesis that best fits the observed labels. The conditions in which this approach is successful were studied in works of Dudley (1978), Pollard (1984), Haussler (1992), Vapnik (1998) and others. They give a number of conditions on the hypothesis class $\mathcal{H}$ that guarantee *uniform convergence* of empirical error to the true error. That is, existence of a function $m_{\mathcal{H}}(\varepsilon, \delta)$ such that for every distribution A over examples, every $h \in \mathcal{H}$, $\varepsilon > 0$, $\delta > 0$, the empirical error of h on sample of $m_{\mathcal{H}}(\varepsilon, \delta)$ examples randomly chosen from A is, with probability at least $1 - \delta$, within ε of $\Delta(A, h)$. We denote the empirical error of h on sample S by $\Delta(S, h)$. In the Boolean case, the following result of Vapnik and Chervonenkis (1971) will be sufficient for our purposes.

Theorem 3 *Let $\mathcal{H}$ be a concept class over X of VC dimension d. Then for every distribution A over $X \times \{-1, 1\}$, every $h \in \mathcal{H}$, $\varepsilon > 0$, $\delta > 0$, and sample S of size $m = O((d \log(d/\varepsilon) + \log(1/\delta))/\varepsilon^2)$ randomly drawn with respect to A,*

$$\mathbf{Pr}[|\Delta(A, h) - \Delta(S, h)| \geq \varepsilon] \leq \delta.$$

In fact a simple uniform convergence result based on the cardinality of the function class follows easily from Chernoff bounds (Haussler, 1992). That is Theorem 3 holds for $m = O(\log |\mathcal{H}|/\varepsilon^2 \cdot \log(1/\delta))$. This result would also be sufficient for our purposes but might give somewhat weaker bounds.

2.3 Membership Queries

A membership oracle for a function f is the oracle that, given any point $z \in \{0, 1\}^n$, returns the value $f(z)$ (Valiant, 1984). We denote it by $\mathrm{MEM}(f)$. We refer to agnostic PAC learning with access to $\mathrm{MEM}(f)$, where f is the unknown function that labels the examples, as *agnostic PAC+MQ* learning. Similarly, one can extend the definition of a membership oracle to fully agnostic learning. For a distribution A over $X \times \{-1, 1\}$, let $\mathrm{MEM}(A)$ be the oracle that, upon query z, returns $b \in \{-1, 1\}$ with probability $\mathbf{Pr}_A[(x, b) \mid x = z]$. We say that $\mathrm{MEM}(A)$ is *persistent* if given the same query the oracle responds with the same label. When learning with persistent membership queries the learning algorithm is allowed to fail with some negligible probability over the answers of $\mathrm{MEM}(A)$. This is necessary to account for probability that the answers of $\mathrm{MEM}(A)$ might be not “representative” of A (a more formal argument can be found for example in the work of Goldman et al. 2001).

2.4 List-Decodable Codes

As we have mentioned earlier, agnostic learning can be seen as recovery of an unknown concept from possibly malicious errors. Therefore, encoding of information that allows recovery from errors, or error-correcting codes, can be useful in the design of agnostic learning algorithms. In our construction we will use binary list-decodable error-correcting codes. A list-decodable code is a code that allows recovery from errors when the number of errors is larger than the distance of the code, and hence there is more than one valid way to decode the corrupted encoding, each giving a different message (see for example the book of van Lint 1998). List-decoding of the code gives the list of all the messages corresponding to the valid ways to decode the corrupt encoding. Formally, let $C : \{0,1\}^u \to \{0,1\}^v$ be a binary code of message length u and block length v. Our construction requires efficient encoding and efficient list-decoding from $1/2-\gamma$ fraction of errors for a $\gamma > 0$ that we will define later. Specifically,

- Efficient encoding algorithm. For any $z \in \{0,1\}^u$ and $j \le v$, $C(z)_j$ (the j^{th} bit of $C(z)$) is computable in time polynomial in u and $\log v$.
- Efficient list-decoding from $(1/2-\gamma')v$ errors in time polynomial in u and $1/\gamma'$ for any $\gamma' \ge \gamma$. That is, an algorithm that given oracle access to the bits of string $y \in \{0,1\}^v$, produces the list of all messages z such that $\mathbf{Pr}_{j\in[v]}[C(z)_j \neq y_j] \le 1/2-\gamma'$ (in time polynomial in u and $1/\gamma'$).

Our main result is achieved using the Reed-Solomon code concatenated with the Hadamard code for which the list-decoding algorithm was given by Guruswami and Sudan (2000). Their code has the desired properties for $v = O(u^2/\gamma^4)$. In the description of our construction, for simplicity, we use the more familiar but exponentially longer Hadamard code.

2.4.1 HADAMARD CODE

The Hadamard code encodes a vector $a \in \{0,1\}^n$ as the values of the parity function χ_a on all the points in $\{0,1\}^n$ (that is the length of the encoding is 2^n). It is convenient to describe list-decoding of the Hadamard code using Fourier analysis over $\{0,1\}^n$ that is commonly used in the context of learning with respect to the uniform distribution (Linial, Mansour, and Nisan, 1993). We now briefly review a number of simple facts on the Fourier representation of functions over $\{0,1\}^n$ and refer the reader to a survey by Mansour (1994) for more details. In the discussion below all probabilities and expectations are taken with respect to the uniform distribution U unless specifically stated otherwise.

Define an inner product of two real-valued functions over $\{0,1\}^n$ to be $\langle f,g\rangle = \mathbf{E}_x[f(x)g(x)]$. The technique is based on the fact that the set of all parity functions $\{\chi_a(x)\}_{a\in\{0,1\}^n}$ forms an orthonormal basis of the linear space of real-valued functions over $\{0,1\}^n$ with the above inner product. This fact implies that any real-valued function f over $\{0,1\}^n$ can be uniquely represented as a linear combination of parities, that is $f(x) = \sum_{a\in\{0,1\}^n} \hat{f}(a)\chi_a(x)$. The coefficient $\hat{f}(a)$ is called Fourier coefficient of f on a and equals $\mathbf{E}_x[f(x)\chi_a(x)]$; a is called the *index* of $\hat{f}(a)$. We say that a Fourier coefficient $\hat{f}(a)$ is θ-heavy if $|\hat{f}(a)| \ge \theta$. Let $L_2(f) = E_x[(f(x))^2]^{1/2}$. Parseval's identity states that

$$(L_2(f))^2 = \mathbf{E}_x[(f(x))^2] = \sum_a \hat{f}^2(a)\ .$$

Let $A = (U,\phi)$ be a distribution over $\{0,1\}^n \times \{-1,1\}$ with uniform marginal distribution over $\{0,1\}^n$. Fourier coefficient $\hat{\phi}(a)$ can be easily related to the error of $\chi_a(x)$ on A. That is,

$$\mathbf{Pr}_{\langle x,b\rangle\sim A}[b \neq \chi_a(x)] = (1-\hat{\phi}(a))/2. \tag{1}$$

Therefore, both list-decoding of the Hadamard code and agnostic learning of parities amount to finding the largest (within 2ϵ) Fourier coefficient of $\phi(x)$. The first algorithm for this task was given by Goldreich and Levin (1989). Given access to a membership oracle, for every $\epsilon > 0$ their algorithm can efficiently find all ϵ-heavy Fourier coefficients.

Theorem 4 (Goldreich and Levin, 1989) *There exists an algorithm* $\mathtt{GL}$ *that for every distribution* $A = (U, \phi)$ *and every* $\epsilon, \delta > 0$*, given access to* $MEM(A)$, $\mathtt{GL}(\epsilon, \delta)$ *returns, with probability at least* $1 - \delta$*, a set of indices* $T \subseteq \{0,1\}^n$ *that contains all a such that* $|\hat{\phi}(a)| \geq \epsilon$ *and for all* $a \in T$, $|\hat{\phi}(a)| \geq \epsilon/2$*. Furthermore, the algorithm runs in time polynomial in n,*$1/\epsilon$ *and* $\log(1/\delta)$.

Note that by Parseval's identity, the condition $|\hat{\phi}(a)| \geq \epsilon/2$ implies that there are at most $4/\epsilon^2$ elements in T.

2.5 Pseudo-random Function Families

A key part of our construction in Section 4 will be based on the use of pseudorandom functions families defined by Goldreich, Goldwasser, and Micali (1986).

Definition 5 *A function family* $\mathcal{F} = \{F_n\}_{n=1}^{\infty}$ *where* $F_n = \{\pi_z\}_{z \in \{0,1\}^n}$ *is a* pseudorandom function family *of Boolean functions over* $\{0,1\}^n$ *if*

- *There exists a polynomial time algorithm that for every n, given* $z \in \{0,1\}^n$ *and* $x \in \{0,1\}^n$ *computes* $\pi_z(x)$.

- *Any adversary M whose resources are bounded by a polynomial in n can distinguish between a function* π_z *(where* $z \in \{0,1\}^n$ *is chosen randomly and kept secret) and a totally random function from* $\{0,1\}^n$ *to* $\{-1,1\}$ *only with negligible probability. That is, for every probabilistic polynomial time M with an oracle access to a function from* $\{0,1\}^n$ *to* $\{-1,1\}$ *there exists a negligible function* $\nu(n)$,

$$|\mathbf{Pr}[M^{\pi_z}(1^n) = 1] - \mathbf{Pr}[M^{\rho}(1^n) = 1]| \leq \nu(n),$$

where π_z *is a function randomly and uniformly chosen from* F_n *and* ρ *is a randomly chosen function from* $\{0,1\}^n$ *to* $\{-1,1\}$*. The probability is taken over the random choice of* π_z *or* ρ *and the coin flips of M.*

Results of Håstad et al. (1999) and Goldreich et al. (1986) give a construction of pseudorandom function families based on the existence of one-way functions.

3. Distribution-Independent Agnostic Learning

In this section we show that in distribution-independent agnostic learning membership queries do not help. In addition, we prove that fully agnostic learning is equivalent to agnostic PAC learning. Our proof is based on two simple observations about agnostic learning via empirical error minimization. Values of the unknown function on points outside of the sample can be set to any value without changing the best fit by a function from the touchstone class. Therefore membership queries do not make empirical error minimization easier. In addition, points with contradicting labels do not influence the complexity of empirical error minimization since any function has the same error on pairs of contradicting labels. We will now provide the formal statement of this result.

Theorem 6 *Let* `Alg` *be an algorithm that agnostically PAC+MQ learns a concept class $\mathcal{C}$ in time $T(\sigma,\epsilon,\delta)$ and outputs a hypothesis in a representation class $\mathcal{H}(\sigma,\epsilon)$. Then $\mathcal{C}$ is (fully) agnostically learnable by $\mathcal{H}(\sigma,\epsilon/2)$ in time $T(\sigma,\epsilon/2,\delta/2)+O(d\cdot\log(d/\epsilon)+\log(1/\delta))/\epsilon^2)$, where d is the VC dimension of $\mathcal{H}(\sigma,\epsilon/2)\cup\mathcal{C}$.*

Proof Let $A=(D,\phi)$ be a distribution over $X\times\{-1,1\}$. Our reduction works as follows. Start by drawing m examples from A for m to be defined later. Denote this sample by S. Let S' be S with all contradicting pairs of examples removed, that is for each example $\langle x,1\rangle$ we remove it together with one example $\langle x,-1\rangle$. Every function has the same error rate of $1/2$ on examples in $S\setminus S'$. Therefore for every function h,

$$\Delta(S,h)=\frac{\Delta(S',h)|S'|+|S\setminus S'|/2}{|S|}=\Delta(S',h)\frac{|S'|}{m}+\frac{m-|S'|}{2m} \qquad (2)$$

and hence

$$\Delta(S,\mathcal{C})=\Delta(S',\mathcal{C})\frac{|S'|}{m}+\frac{m-|S'|}{2m}. \qquad (3)$$

Let $f(x)$ denote the function equal to b if $\langle x,b\rangle\in S'$ and equal to 1 otherwise. Let $U_{S'}$ denote the uniform distribution over S'. Given the sample S' we can easily simulate the example oracle $\mathrm{EX}(U_{S'},f)$ and $\mathrm{MEM}(f)$. We run `Alg`$(\epsilon/2,\delta/2)$ with theses oracles and denote its output by h. Note, that this simulates $\mathcal{A}$ in the agnostic PAC+MQ setting over distribution $(U_{S'},f)$.

By the definition of $U_{S'}$, for any Boolean function $g(x)$,

$$\mathbf{Pr}_{U_{S'}}[f(x)\neq g(x)]=\frac{1}{|S'|}\left|\{x\in S'\mid f(x)\neq g(x)\}\right|=\Delta(S',g)\ .$$

That is, the error of any function g on $U_{S'}$ is exactly the empirical error of g on sample S'. Thus $\Delta((U_{S'},f),h)=\Delta(S',h)$ and $\Delta((U_{S'},f),\mathcal{C})=\Delta(S',\mathcal{C})$. By the correctness of `Alg`, with probability at least $1-\delta/2$, $\Delta(S',h)\leq\Delta(S',\mathcal{C})+\epsilon/2$. By Equations (2) and (3) we thus obtain that

$$\Delta(S,h)=\Delta(S',h)\frac{|S'|}{m}+\frac{m-|S'|}{2m}\leq(\Delta(S',\mathcal{C})+\frac{\epsilon}{2})\frac{|S'|}{m}+\frac{m-|S'|}{2m}=\Delta(S,\mathcal{C})+\frac{\epsilon}{2}\frac{|S'|}{m}.$$

Therefore $\Delta(S,h)\leq\Delta(S,\mathcal{C})+\epsilon/2$. We can apply the VC dimension-based uniform convergence results for $\mathcal{H}(\sigma,\epsilon/2)\cup\mathcal{C}$ (Theorem 3) to conclude that for

$$m(\epsilon/4,\delta/4)=O\left(\frac{d\log(d/\epsilon)+\log(1/\delta)}{\epsilon^2}\right),$$

with probability at least $1-\delta/2$, $\Delta(A,h)\leq\Delta(S,h)+\frac{\epsilon}{4}$ and $\Delta(S,\mathcal{C})+\frac{\epsilon}{4}\leq\Delta(A,\mathcal{C})$. Finally, we obtain that with probability at least $1-\delta$,

$$\Delta(A,h)\leq\Delta(S,h)+\frac{\epsilon}{4}\leq\Delta(S,\mathcal{C})+\frac{3\epsilon}{4}\leq\Delta(A,\mathcal{C})+\epsilon.$$

It easy to verify that the running time and hypothesis space of this algorithm are as claimed. ■

Note that if `Alg` is efficient then $d(\sigma,\epsilon/2)$ is polynomial in σ and $1/\epsilon$ and, in particular, the obtained algorithm is efficient. In addition, in place of VC-dim one can use the uniform convergence result based on the cardinality of the hypothesis space. The description length of a hypothesis output by `Alg` is polynomial in σ and $1/\epsilon$ and hence in this case a polynomial number of samples will be required to simulate `Alg`.

Remark 7 *We note that while this proof is given for the strongest version of agnostic learning in which the error of an agnostic algorithm is bounded by* $\Delta(A,\mathcal{C})+\varepsilon$*, it can be easily extended to weaker forms of agnostic learning, such as algorithms that only guarantee error bounded by* $\alpha\cdot\Delta(A,\mathcal{C})+\beta+\varepsilon$ *for some* $\alpha\geq 1$ *and* $\beta\geq 0$*. This is true since the reduction adds at most* $\varepsilon/2$ *to the error of the original algorithm.*

4. Learning with Respect to the Uniform Distribution

In this section we show that when learning with respect to the uniform distribution over $\{0,1\}^n$, membership queries are helpful. Specifically, we show that if one-way functions exist, then there exists a concept class $\mathcal{C}$ that is not agnostically PAC learnable (even weakly) with respect to the uniform distribution but is agnostically learnable over the uniform distribution given membership queries. Our agnostic learning algorithm is successful only when $\varepsilon\geq 1/p(n)$ for a polynomial p fixed in advance (the definition of $\mathcal{C}$ depends on p). While this is slightly weaker than required by the definition of the model it still exhibits the gap between agnostic learning with and without membership queries. We remark that a number of known PAC and agnostic learning algorithms are efficient only for restricted values of ε (O'Donnell and Servedio, 2006; Gopalan et al., 2008; Kalai et al., 2008a).

Theorem 8 *For every polynomial* $p(\cdot)$*, there exists a concept class* $\mathcal{C}^p$ *over* $\{0,1\}^n$ *such that,*

1. *there exists no efficient algorithm that weakly PAC learns* $\mathcal{C}^p$ *with respect to the uniform distribution over* $\{0,1\}^n$*;*

2. *there exists a randomized algorithm* `AgnLearn` *that for every distribution* $A=(U,\phi)$ *over* $\{0,1\}^n\times\{-1,1\}$ *and every* $\varepsilon\geq 1/p(n),\delta>0$*, given access to MEM*$(A)$*, with probability at least* $1-\delta$*, finds h such that* $\Delta(A,h)\leq\Delta(A,\mathcal{C}^p_n)+\varepsilon$*. The probability is taken over the coin flips of MEM*(A) *and* `AgnLearn`*.* `AgnLearn` *runs in time polynomial in n and* $\log(1/\delta)$*.*

4.1 Background

We first show why some of the known separation results will not work in the agnostic setting. It is well-known that the PAC model with membership queries is strictly stronger than the PAC model without membership queries (under the same cryptographic assumption). The separation result is obtained by using a concept class $\mathcal{C}$ that is not PAC learnable and augmenting each concept $f\in\mathcal{C}$ with the encoding of f in a fixed part of the domain. This encoding is readable using membership queries and therefore an MQ algorithm can "learn" the augmented $\mathcal{C}$ by querying the points that contain the encoding. On the other hand, with overwhelming probability this encoding will not be observed in random examples and therefore does not help learning from random examples. This simple approach would fail in the agnostic setting. The unknown function might be random on the part of the domain that contains the encoding and equal to a concept from $\mathcal{C}$ elsewhere. The agreement of the unknown function with a concept from $\mathcal{C}$ is almost 1 but membership queries on the points of encoding will not yield any useful information.

A similar problem arises with encoding schemes used in the separation results of Elbaz et al. (2007) and Feldman and Shah (2009). There too the secret encoding can be rendered unusable by a function that agrees with a concept in $\mathcal{C}$ on a significant fraction of the domain.

4.2 Outline

We start by presenting some of the intuition behind our construction. As in most other separation results our goal is to create a concept class that is not learnable from uniform examples but includes an encoding of the unknown function that is readable using membership queries. We first note that in order for this approach to work in the agnostic setting the secret encoding has to be "spread" over at least $1-2\varepsilon$ fraction of $\{0,1\}^n$. To see this let f be a concept and let $S \subseteq \{0,1\}^n$ be the subset of the domain where the encoding of f is contained. Assume, for simplicity, that without the encoding the learning algorithm cannot predict f on $\overline{S} = \{0,1\}^n \setminus S$ with any significant advantage over random guessing. Let f' be a function equal to f on $\overline{S}$ and truly random on S. Then

$$\mathbf{Pr}[f = f'] \approx (|\overline{S}| + |S|/2)/2^n = 1/2 + \frac{|\overline{S}|}{2^{n+1}} \ .$$

On the other hand, f' does not contain any information about the encoding of f and therefore, by our assumption, no efficient algorithm can produce a hypothesis with advantage significantly higher than $1/2$ on both S and $\overline{S}$. This means that the error of any efficient algorithm will be higher by at least $|\overline{S}|/2^{n+1}$ than the best possible. To ensure that $|\overline{S}|/2^{n+1} \leq \varepsilon$, we need $|S| \geq (1-2\varepsilon)2^n$.

Another requirement that the construction has to satisfy is that the encoding of the secret has to be resilient to almost any amount of noise. In particular, since the encoding is a part of the function, we also need to be able to reconstruct an encoding that is close to the best possible. An encoding with this property is in essence a list-decodable binary code. In order to achieve the strongest separation result we will use the code of Guruswami and Sudan (2000) that is the concatenation of Reed-Solomon code with the binary Hadamard code. However, to simplify the presentation, we will use the more familiar binary Hadamard code in our construction. In Section 4.6 we provide the details on the use of the Guruswami-Sudan code in place of the Hadamard code.

The Hadamard code is equivalent to encoding a vector $a \in \{0,1\}^n$ as the values of the parity function χ_a on all points in $\{0,1\}^n$. That is, n bit vector a is encoded into 2^n bits given by $\chi_a(x)$ for every $x \in \{0,1\}^n$. This might appear quite inefficient since a learning algorithm will not be able to read all the bits of the encoding. However the Goldreich-Levin algorithm provides an efficient way to recover the indices of all the parities that agree with a given function with probability significantly higher than $1/2$ (Goldreich and Levin, 1989). Therefore the Hadamard code can be decoded by reading the code in only a polynomial number of (appropriately-chosen) locations.

The next problem that arises is that the encoding should not be readable from random examples. As we have observed earlier, we cannot simply "hide" it on a negligible fraction of the domain. Specifically, we need to make sure that our Hadamard encoding is not recoverable from random examples. Our solution to this problem is to use a pseudo-random function to make values on random examples indistinguishable from random coin flips in the following manner. Let $a \in \{0,1\}^n$ be the vector we want to encode and let $\pi_d : \{0,1\}^n \to \{-1,1\}$ be a pseudo-random function from some pseudorandom function family $\mathcal{F} = \{\pi_b\}_{b \in \{0,1\}^n}$. We define a function $g : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ as

$$g(z,x) = \pi_d(z) \oplus \chi_a(x)$$

($\oplus$ is simply the product in $\{-1,1\}$). The label of a random point $(z,x) \in \{0,1\}^{2n}$ is a XOR of a pseudorandom bit with an independent bit and therefore is pseudorandom. Values of a pseudorandom function b on any polynomial set of distinct points are pseudorandom and therefore random points will have pseudorandom labels as long as their z parts are distinct. In a sample of polynomial in n size of random and uniform points from $\{0,1\}^{2n}$ this happens with overwhelming probability and therefore $g(z,x)$ is not learnable from random examples. On the other hand, for a fixed z, $\pi_d(z) \oplus \chi_a(x)$ gives a Hadamard encoding of a or its negation. Hence it is possible to find a using membership queries with the same prefix. A construction based on a similar idea was used by Elbaz et al. (2007) in their separation result.

Finally, the problem with the construction we have so far is that while a membership query learning algorithm can find the secret a, it cannot predict $g(z,x)$ without knowing d. This means that we also need to provide d to the learning algorithm. It is tempting to use the Hadamard code to encode d together with a. However, a bit of the encoding of d is no longer independent of π_d, and therefore the previous argument does not hold. We are unaware of any constructions of pseudorandom functions that would remain pseudorandom when the value of the function is "mixed" with the description of the function (see the work of Halevi and Krawczyk (2007) for a discussion of this problem). An identical problem also arises in the construction of Elbaz et al. (2007). They used another pseudorandom function π_{d^1} to "hide" the encoding of d, then used another pseudorandom function π_{d^2} to "hide" the encoding of d^1 and so on. The fraction of the domain used up for the encoding of d^i is becoming progressively smaller as i grows. In their construction a PAC learning algorithm can recover as many of the encodings as is required to reach accuracy ε. This method would not be effective in our case. First, in the agnostic setting all the encodings but the one using the largest fraction of the domain can be "corrupted". This makes the largest encoding unrecoverable and implies that the best ε achievable is at most half of the fraction of the domain used by the largest encoding. In addition, in the agnostic setting the encoding of d^i for every odd i can be completely "corrupted" making all the other encodings unrecoverable. To solve these problems in our construction we split the domain into p equal parts and on part i we use a pseudorandom function π_{d^i} to "hide" the encoding of d^j for all $j < i$. In Figure 1 we provide a schematic view of a concept that we construct (for $p = 4$).

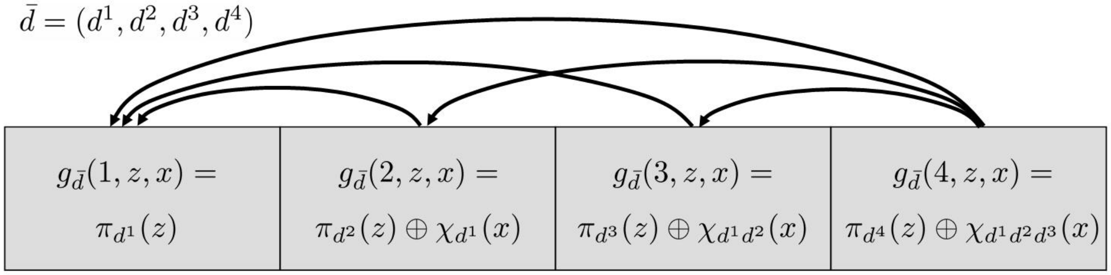
$\bar{d} = (d^1, d^2, d^3, d^4)$
$g_{\bar{d}}(1,z,x) =$
$\pi_{d^1}(z)$
$g_{\bar{d}}(2,z,x) =$
$\pi_{d^2}(z) \oplus \chi_{d^1}(x)$
$g_{\bar{d}}(3,z,x) =$
$\pi_{d^3}(z) \oplus \chi_{d^1 d^2}(x)$
$g_{\bar{d}}(4,z,x) =$
$\pi_{d^4}(z) \oplus \chi_{d^1 d^2 d^3}(x)$

Figure 1: Structure of a concept in $\mathcal{C}_n^p$ for $p = 4$. Arrow from part i to part j indicates that the secret key to part j is encoded using the Hadamard code in part i.

The crucial property of this construction is that the unknown concept can be "recovered" on all but one part of the domain. Specifically, the only part where the unknown concept cannot be "recovered" agnostic is the part i such for all $j > i$ agreement of the target function with every $g_{\bar{d}} \in \mathcal{C}_n^p$ on part j is close to $1/2$ and hence d^j cannot be recovered. Therefore, by making the number of parts p larger than $1/\varepsilon$, we can make sure that there exists an efficient algorithm that finds a hypothesis with the error within ε of the optimum.

4.3 The Construction

We will now describe the construction formally and give a proof of its correctness. Let $p = p(n)$ be a polynomial, let $\ell = \log p(n)$ (we assume for simplicity that $p(n)$ is a power of 2) and let $m = \ell + n \cdot p$. We refer to an element of $\{0,1\}^m$ by triple $(k, z, \bar{x})$ where $k \in [p]$, $z \in \{0,1\}^n$, and

$$\bar{x} = (x^1, x^2, \dots, x^{p-1}) \in \{0,1\}^{n \times (p-1)}.$$

Here k indexes the encodings, z is the input to the k-th pseudorandom function and $\bar{x}$ is the input to a parity function on $n(p-1)$ variables that encodes the secret keys for all pseudorandom functions used for encodings 1 through $k-1$. Formally, let

$$\overline{d} = (d^1, d^2, \dots, d^{p-1})$$

be a vector in $\{0,1\}^{n \times (p-1)}$ (where each $d^i \in \{0,1\}^n$) and for $k \in [p]$ let

$$\overline{d}(k) = (d^1, d^2, \dots, d^{k-1}, 0^n, \dots, 0^n).$$

Let $\mathcal{F} = \{\pi_y\}_{y \in \{0,1\}^*}$ be a pseudorandom function family (Definition 5). We define $g_{\overline{d}} : \{0,1\}^m \to \{-1,1\}$ as follows:

$$g_{\overline{d}}(k, z, \bar{x}) = \pi_{d^k}(z) \oplus \chi_{\overline{d}(k)}(\bar{x}) \; .$$

Finally, we define

$$\mathcal{C}_n^p = \left\{ g_{\overline{d}} \mid \overline{d} \in \{0,1\}^{n \times (p-1)} \right\} \; .$$

4.4 Hardness of Learning $\mathcal{C}_n^p$ From Random Examples

We start by showing that $\mathcal{C}_n^p$ is not agnostically learnable from random and uniform examples only. In fact, we will show that it is not even weakly PAC learnable. Our proof is similar to the proof by Elbaz et al. (2007) who show that the same holds for the concept class they define.

Theorem 9 *There exists no efficient algorithm that weakly PAC learns $\mathcal{C}_n^p$ with respect to the uniform distribution over $\{0,1\}^m$.*

Proof In order to prove the claim we show that a weak PAC learning algorithm for $\mathcal{C}_n^p$ can be used to distinguish a pseudorandom function family from a truly random function. A weak learning algorithm for $\mathcal{C}_n^p$ implies that every function in $\mathcal{C}_n^p$ can be distinguished from a truly random function on $\{0,1\}^m$. If, on the other hand, in the computation of $g_{\overline{d}}(k, z, \bar{x})$ we used a truly random function in place of each $\pi_{d^k}(z)$ then the resulting labels would be truly random and, in particular, unpredictable.

Formally, let `Alg` be a weak learning algorithm for $\mathcal{C}_n^p$ that, with probability at least $1-\delta$, produces a hypothesis with error of at most $1/2 - 1/q(m)$ and runs in time $t(m, 1/\delta)$ for some polynomials $t(\cdot,\cdot)$ and $q(\cdot)$. Our concept class $\mathcal{C}_n^p$ uses numerous pseudorandom functions from F_n and therefore we use a so-called "hybrid" argument to show that one can replace a single $\pi_{d^k}(z)$ with a truly random function to cause `Alg` to fail.

For $0 \leq i \leq p$, let $\mathbb{O}(i)$ denote an oracle randomly chosen according to the following procedure. First choose randomly and uniformly $\pi_{d^1}, \pi_{d^2}, \dots, \pi_{d^i} \in F_n$ and then choose randomly and uniformly $\rho_{i+1}, \rho_{i+2}, \dots, \rho_p$ from the set of all Boolean functions over $\{0,1\}^n$. Upon request such an oracle returns an example $\langle (k, z, \bar{x}), b \rangle$ where $(k, z, \bar{x})$ is chosen randomly and uniformly from $\{0,1\}^m$ and

$$b = \begin{cases} \pi_{d^k}(z) \oplus \chi_{\overline{d}(k)}(\bar{x}) & 0 \leq k \leq i, \\ \rho_k(z) & i < k \leq p. \end{cases}$$

We note that in order to simulate such an oracle it is not needed to explicitly choose $\rho_{i+1}, \rho_{i+2}, \ldots, \rho_p$ (and, indeed that would not be possible in polynomial time). Instead their values can be generated upon request by flipping a fair coin. This means that for every i, $\mathbb{O}(i)$ can be chosen and then simulated in time polynomial in m and the number of examples requested. Let $M(i)$ denote the algorithm that performs the following steps.

- Choose $\mathbb{O}(i)$ randomly according to the above procedure.
- Simulate `Alg` with random examples from $\mathbb{O}(i)$ and $\delta = 1/2$. Let h be the output of `Alg`.
- Produce an estimate $\tilde{e_h}$ of the error of h on distribution defined by $\mathbb{O}(i)$ that, with probability at least $7/8$, is within $1/(3q(m))$ of the true error. Chernoff bounds imply that this can be done using an empirical estimate on $O(q^2(m))$ random samples.
- Output 1 if $\tilde{e_h} \leq 1/2 - 2/(3q(m))$ and 0 otherwise.

We denote by δ_i the probability that $M(i)$ outputs 1. The probability is taken over all the random choices made by $M(i)$: the random choice and simulation of $\mathbb{O}(i)$, the coin flips of `Alg` and the estimation of the error of h.

Claim 10 $\delta_p - \delta_0 \geq 1/4$.

Proof To see this we first observe that $\mathbb{O}(0)$ is defined using p truly random functions and therefore, the probability that there exists a hypothesis of size at most $t(m,2)$ that has error less than $1/2 - 1/3q(m)$ is some negligible function $\nu(n)$. In particular, the error of the hypothesis produced by `Alg` is at least $1/2 - 1/3q(m)$ (with probability at least $1 - \nu(n)$). This means that $\tilde{e_h} \leq 1/2 - 2/(3q(m))$ only if the estimation fails. By the definition of our error estimation procedure, this happens with probability at most $1/8$ and therefore $\delta_0 \leq 1/8 + \nu(n)$. On the other hand, $\mathbb{O}(p)$ is equivalent to $\mathrm{EX}(U, g_{\bar{d}})$ for some randomly chosen $\bar{d}$. This implies that with probability at least $1/2$, `Alg` outputs a hypothesis with error of at most $1/2 - 1/q(m)$. With probability at least $7/8$, $\tilde{e_h} \leq 1/2 - 2/(3q(m))$, and hence $\delta_p \geq 7/16$. This implies our claim. ■

We now describe our distinguisher M^{π}, where π denotes the function given to M as an oracle. Let $\mathbb{O}^{\pi}(i)$ denote the example oracle generated by using π in place of π_{d^i} in the definition of $\mathbb{O}(i)$. That is, first choose randomly and uniformly $\pi_{d^1}, \pi_{d^2}, \ldots, \pi_{d^{i-1}} \in F_n$ and then choose randomly and uniformly $\rho_{i+1}, \rho_{i+2}, \ldots, \rho_p$ from the set of all Boolean functions over $\{0,1\}^n$. Upon request $\mathbb{O}^{\pi}(i)$ returns an example $\langle (k, z, \bar{x}), b \rangle$ where $(k, z, \bar{x})$ is chosen randomly and uniformly from $\{0,1\}^m$ and

$$b = \begin{cases} \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k < i, \\ \pi(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k = i, \\ \rho_k(z) & k > i. \end{cases}$$

Similarly, we denote by $M^{\pi}(i)$ the algorithm that is the same as $M(i)$ but chooses a random $\mathbb{O}^{\pi}(i)$ in place of $\mathbb{O}(i)$. The distinguishing test M^{π} chooses a random $i \in [p]$ and runs $M^{\pi}(i)$.

We first observe that if π is chosen randomly from F_n then choosing and simulating a random $\mathbb{O}^{\pi}(i)$ is equivalent to choosing and simulating a random $\mathbb{O}(i)$. Therefore for every $i \in [p]$, $M^{\pi}(i)$ is equivalent to $M(i)$. This implies that in this case M^{π} will output 1 with probability

$$\frac{1}{p} \sum_{i \in [p]} \delta_i.$$

On the other hand, if π is chosen randomly from the set of all Boolean function over $\{0,1\}^n$ then $\mathbb{O}^{\pi}(i)$ is equivalent to $\mathbb{O}(i-1)$. Therefore in this case M^{π} will output 1 with probability

$$\frac{1}{p}\sum_{i\in[p]}\delta_{i-1}.$$

Therefore, by Claim 10 this implies that the difference in the probability that M outputs 1 in these two cases is

$$\frac{1}{p}\sum_{i\in[p]}\delta_i - \frac{1}{p}\sum_{i\in[p]}\delta_{i-1} = \frac{1}{p}(\delta_p - \delta_0) \geq 1/(4p),$$

that is, non-negligible.

The efficiency of M follows readily from the efficiency of `Alg` and the efficiency of the steps we described. This gives us the contradiction to the pseudorandomness property of function family $\mathcal{F}$. ■

4.5 Agnostic Learning of $\mathcal{C}_n^p$ with Membership Queries

We now describe a (fully) agnostic learning algorithm for $\mathcal{C}_n^p$ that uses membership queries and is successful for any $\varepsilon \geq 1/p(n)$.

Theorem 11 *There exists a randomized algorithm* `AgnLearn` *that for every distribution $A = (U,\phi)$ over $\{0,1\}^m \times \{-1,1\}$ and every $\varepsilon \geq 1/p(n), \delta > 0$, given access to MEM$(A)$, with probability at least $1-\delta$, finds h such that $\Delta(A,h) \leq \Delta(A,\mathcal{C}_n^p)+\varepsilon$. The probability is taken over the coin flips of MEM(A) and* `AgnLearn`. `AgnLearn` *runs in time polynomial in m and* $\log(1/\delta)$.

Proof Let $g_{\bar{e}}$ for $\bar{e} = (e^1,e^2,\ldots,e^{p-1}) \in \{0,1\}^{(p-1)\times n}$ be the function for which $\Delta(A,g_{\bar{e}}) = \Delta(A,\mathcal{C}_n^p)$. The goal of our algorithm is to find the largest j such that on random examples from the j-th part of the domain (i.e., for $k = j$) A agrees with the encoding of $\bar{e}(j) = (e^1,e^2,\ldots,e^{j-1},0^n,\ldots,0^n)$ with probability at least $1/2+\varepsilon/4$. Using Goldreich-Levin algorithm such j can be used to recover $\bar{e}(j)$ and therefore will allow us to reconstruct $g_{\bar{e}}$ on all points $(k,z,\bar{x})$ for $k < j$. For points with $k \geq j$, our hypothesis will be either constant 1 or constant -1, whichever has the higher agreement with A. This guarantees that the error on this part is at most $1/2$. By the definition of j, $g_{\bar{e}}$ has error of at least $1/2-\varepsilon/4-1/(2p) \geq 1/2-\varepsilon$ on this part of the domain and therefore our hypothesis has error close to that of $g_{\bar{e}}$.

We now describe `AgnLearn` formally. For every $i \in [p]$ and $y \in \{0,1\}^n$, let $A_{i,y}$ be A restricted to points in $\{0,1\}^m$ with prefix i,y. That is $A = (U_{(p-1)\times n},\phi_{i,y})$ where $\phi_{i,y}(\bar{x}) \equiv \phi(i,y,\bar{x})$ and $U_{(p-1)\times n}$ is the uniform distribution over $\{0,1\}^{(p-1)\times n}$. Note that MEM$(A_{i,y})$ can be simulated using MEM(A): when queried on a point $\bar{x} \in \{0,1\}^{(p-1)\times n}$ MEM$(A_{i,y})$ returns the answer of MEM(A) on point $(i,y,\bar{x})$. Further, for each vector $\bar{d} \in \{0,1\}^{(p-1)\times n}$ and $b \in \{-1,1\}$, let $h_{\bar{d},i,b}$ be defined as

$$h_{\bar{d},i,b}(k,z,\bar{x}) = \begin{cases} \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k < i, \\ b & k \geq i. \end{cases} \tag{4}$$

(Here π_{d^k} is an element of the pseudorandom function family F_n used in the construction.)

`AgnLearn` performs the following steps.

1. Initializes $H = \{h_1, h_{-1}\}$, where $h_1 \equiv 1$ and $h_{-1} \equiv -1$.
2. For each $2 \leq i \leq p$:
   (a) Chooses r independent random and uniform points in $\{0,1\}^n$, for r to be defined later. Denote the obtained set of points by Y_i.
   (b) For each $y \in Y_i$:
      i. Runs $\mathtt{GL}(\varepsilon/4, 1/2)$ over $\{0,1\}^{(p-1)\times n}$ using $\mathrm{MEM}(A_{i,y})$. Let T denote the set of indices of heavy Fourier coefficients returned by GL.
      ii. For each vector $\overline{d} \in T$ and $b \in \{-1,1\}$ adds $h_{\overline{d},i,b}$ to the set of hypotheses H.
3. For each $h \in H$, estimates $\Delta(A,h)$ to within accuracy $\varepsilon/8$ and with overall confidence $1-\delta/2$ using the empirical error on random samples from A. Chernoff bounds imply that this can be done using samples of size $O(\log(|H|/\delta)/\varepsilon^2)$. Denote the estimate obtained for a hypothesis $h_{\overline{d},i,b} \in H$ by $\tilde{\Delta}_{\overline{d},i,b}$.
4. Returns $h \in H$ with the lowest empirical error.

Claim 12 *For $r = O(\log(1/\delta)/\varepsilon)$, with probability at least $1-\delta$,* AgnLearn *returns h such that $\Delta(A,h) \leq \Delta(A,\mathcal{C}_n^p) + \varepsilon$.*

Proof We show that in the set H of hypotheses considered by AgnLearn there will be a hypothesis h' such that $\Delta(A,h') \leq \Delta(A,g_{\overline{e}}) + 3\varepsilon/4$ (with sufficiently high probability). The estimates of the error of each hypothesis are within $\varepsilon/8$ of the true error and therefore the hypothesis h with the smallest empirical error will satisfy

$$\Delta(A,h) \leq \Delta(A,h') + \varepsilon/4 \leq \Delta(A,g_{\overline{e}}) + \varepsilon \ .$$

For $i \in [p]$, denote

$$\Delta_i = \mathbf{Pr}_A[b \neq g_{\overline{e}}(k,z,\overline{x}) \mid k = i]$$

(here and below by probability with respect to A we mean that a labeled example $\langle (k,z,\overline{x}), b\rangle$ is chosen randomly according to A). By the definition,

$$\frac{1}{p}\sum_{i\in[p]} \Delta_i = \Delta(A,g_{\overline{e}}). \tag{5}$$

Let j be the largest $i \in [p]$ that satisfies $\Delta_i \leq 1/2 - \varepsilon/4$. If j is undefined (when no i satisfies the condition) then by Equation (5), $\Delta(A,g_{\overline{e}}) > 1/2 - \varepsilon/4$. Either h_1 or h_{-1} has error of at most $1/2$ on A and therefore there exists $h' \in H$ such that $\Delta(A,h') \leq \Delta(A,g_{\overline{e}}) + 3\varepsilon/4$.

We can now assume that j is well-defined. For $i \in [p]$ and $y \in \{0,1\}^n$ denote

$$\Delta_{i,y} = \mathbf{Pr}_A[b \neq g_{\overline{e}}(k,z,\overline{x}) \mid k = i,\ z = y] = \mathbf{Pr}_{\langle \overline{x},b\rangle \sim A_{i,y}}[b \neq g_{\overline{e}}(i,y,\overline{x})].$$

The function $g_{\overline{e}}(j,y,\overline{x})$ equals $\pi_{d^j}(y)\cdot\chi_{\overline{e}(j)}(\overline{x})$. If $\pi_{d^j}(y) = 1$ then by Equation (1) and the definition of $A_{j,y}$,

$$\Delta_{j,y} = \frac{1 - \widehat{\phi_{j,y}}(\overline{e}(j))}{2} \ ,$$

and therefore $\widehat{\phi_{j,y}}(\bar{e}(j)) = 1 - 2\Delta_{j,y}$. If $\pi_{d^j}(y) = -1$ then

$$\Delta_{j,y} = 1 - \frac{1 - \widehat{\phi_{j,y}}(\bar{e}(j))}{2} = \frac{1 + \widehat{\phi_{j,y}}(\bar{e}(j))}{2}$$

and thus $\widehat{\phi_{j,y}}(\bar{e}(j)) = -(1 - 2\Delta_{j,y})$. In either case

$$|\widehat{\phi_{j,y}}(\bar{e}(j))| \geq 1 - 2\Delta_{j,y}. \tag{6}$$

By the definition,

$$\mathbf{E}_{y \in \{0,1\}^n}[\Delta_{i,y}] = \Delta_i.$$

This implies that for a randomly and uniformly chosen y, with probability at least $\varepsilon/4$, $\Delta_{j,y} \leq 1/2 - \varepsilon/8$. This is true since otherwise

$$\Delta_j \geq (1 - \frac{\varepsilon}{4})(\frac{1}{2} - \frac{\varepsilon}{8}) > \frac{1}{2} - \frac{\varepsilon}{4} \, ,$$

contradicting the choice of j.

Together with Equation (6) we obtain that for a randomly chosen y, with probability at least $\varepsilon/4$, $|\widehat{\phi_{j,y}}(\bar{e}(j))| \geq \varepsilon/4$. In this case, by Theorem 4, $\mathtt{GL}(\varepsilon/4, 1/2)$ with access to $\mathrm{MEM}(A_{j,y})$ will return $\bar{e}(j)$ with probability at least $1/2$ (possibly, among other vectors). This means that $\bar{e}(j)$ will be found with probability at least $\varepsilon/8$. By taking $r = 8\ln(2/\delta)/\varepsilon$ we ensure that `AgnLearn` finds $\bar{e}(j)$ with probability at least $1 - \delta/2$.

Now let b_j be the constant with the lowest error on examples from A for which $k \geq j$, that is

$$b_j = \mathtt{sign}(\mathbf{E}_A[b \mid k \geq j]).$$

Clearly, the error of b_j on A when $k \geq j$ is at most $1/2$. By the definition of $h_{\bar{e}(j),j,b_j}$ (Equation 4), $h_{\bar{e}(j),j,b_j}$ equals $g_{\bar{e}}$ on points for which $k < j$ and equals b_j on the rest of the domain. Therefore

$$\begin{aligned}\Delta(A, h_{\bar{e}(j),j,b_j}) &= \frac{j-1}{p}\mathbf{Pr}_A[b \neq g_{\bar{e}}(k,z,\bar{x}) \mid k < j] + \frac{p-j+1}{p}\mathbf{Pr}_A[b \neq b_j \mid k \geq j] \\ &\leq \frac{1}{p}\left(\sum_{i<j}\Delta_i + \frac{p-j+1}{2}\right).\end{aligned}$$

On the other hand, by the properties of j, for all $i > j$, $\Delta_i \geq 1/2 - \varepsilon/4$ and thus

$$\Delta(A, g_{\bar{e}}) = \frac{1}{p}\left(\sum_{i \in [p]}\Delta_i\right) \geq \frac{1}{p}\left(\sum_{i<j}\Delta_i + (p-j)\left(\frac{1}{2} - \frac{\varepsilon}{4}\right)\right) \, .$$

By combining these equations we obtain that

$$\Delta(A, h_{\bar{e}(j),j,b_j}) - \Delta(A, g_{\bar{e}}) \leq \frac{1}{2p} + \frac{\varepsilon}{4} \leq \frac{3\varepsilon}{4}.$$

As noted before, this implies the claim. ■

Given Claim 12, we only need to check that the running time of `AgnLearn` is polynomial in m and $\log(1/\delta)$. By Parseval's identity there are $O(1/\varepsilon^2)$ elements in each set of vectors returned by `GL` and $r = 8\ln(2/\delta)/\varepsilon$. Therefore the efficiency of `GL` implies that H is found in polynomial time and the size of H is $O(p \cdot \log(1/\delta)/\varepsilon^3)$. This implies that the error estimation step of `AgnLearn` takes polynomial time. ■

Remark 13 *In Theorem 11 we assumed that MEM(A) is not persistent. If MEM(A) is persistent then executions of* `GL` *for different y's are not completely independent and* `GL` *might fail with some negligible probability. A simple and standard modification of the analysis (as in the work of Bshouty et al. (2004) for example) can be used to show that the probability of failure of* `AgnLearn` *in this case is negligible. This implies that* `AgnLearn` *agnostically learns $\mathcal{C}_n^p$ from persistent membership queries.*

4.6 Bounds on ε

In Theorem 11 $\mathcal{C}_n^p$ is defined over $\{0,1\}^m$ for $m = n \cdot p(n) + \log p(n)$ and is learnable agnostically for any $\varepsilon \geq 1/p(n)$. This means that this construction cannot achieve dependence on ε beyond $1/m$. To improve this dependence we use a more efficient list-decodable code in place of the Hadamard code. Specifically, we need a list-decodable code $C : \{0,1\}^u \rightarrow \{0,1\}^v$ that can be list-decoded from $(1/2 - \gamma')v$ errors in time polynomial in u and $1/\gamma'$ for any $\gamma' \geq \varepsilon/8$. Guruswami and Sudan (2000) gave a list-decoding algorithm for the Reed-Solomon code concatenated with the Hadamard code that has the desired properties for $v = O(u^2/\varepsilon^4)$. Note that this is exponentially more efficient than the Hadamard code for which $v = 2^u$. In fact for this code we can afford to read the whole corrupt message in polynomial time. This means that we can assume that the output of the list-decoding algorithm is exact (and not approximate as in the case of list-decoding using the Hadamard code using the Goldreich-Levin algorithm).

In our construction $u = n(p(n) - 1)$. To apply the above code we index a position in the code using $\log v = O(\log(n/\varepsilon))$ bits. Further we can use pseudorandom functions over $\{0,1\}^{n/2}$ instead of $\{0,1\}^n$ in the definition of $\mathcal{C}_n^p$. We would then obtain that the dimension of $\mathcal{C}_n^p$ is $m = n/2 + \log v + \log p(n) \leq n$ for any polynomial $p(n)$ and $\varepsilon \geq 1/p(n)$. This implies that our learning algorithm is successful for every $\varepsilon \geq 1/p(n) \geq 1/p(m)$. It is easy to verify that Theorems 9 and 11 still hold for this variant of the construction and imply Theorem 8.

5. Discussion

Our results clarify the role of membership queries in agnostic learning. They imply that in order to extract any meaningful information from membership queries the learner needs to have significant prior knowledge about the distribution of examples. Specifically, either the set of possible classification functions has to be restricted (as in the PAC model) or the set of possible marginal distributions (as in distribution-specific agnostic learning).

A interesting result in this direction would be a demonstration that membership queries are useful for distribution-specific agnostic learning of a natural concept class such as halfspaces.

Finally, we would be interested to see a proof that membership queries are useful in distribution-specific agnostic learning that places no restriction on ε.

Acknowledgments

I thank Parikshit Gopalan, Salil Vadhan and David Woodruff for valuable discussions and comments on this research. I also thank anonymous reviewers of COLT 2008 and JMLR for their numerous helpful comments.

References

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

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.

A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

N. Bshouty. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153, 1995.

N. Bshouty, J. Jackson, and C. Tamon. More efficient PAC learning of DNF with membership queries under the uniform distribution. In *Proceedings of COLT*, pages 286–295, 1999.

N. Bshouty, J. Jackson, and C. Tamon. More efficient PAC-learning of DNF with membership queries under the uniform distribution. *Journal of Computer and System Sciences*, 68(1):205–234, 2004.

R. Dudley. Central limit theorems for empirical measures. *Annals of Probability*, 6(6):899–929, 1978.

A. Elbaz, H. Lee, R. Servedio, and A. Wan. Separating models of learning from correlated and uncorrelated data. *Journal of Machine Learning Research*, 8:277–290, 2007.

V. Feldman. Optimal hardness results for maximizing agreements with monomials. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 226–236, 2006.

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

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.

V. Feldman and S. Shah. Separating models of learning with faulty teachers. *Theoretical Computer Science*, doi:10.1016/j.tcs.2009.01.017, 2009.

S. A. Goldman, S. Kwek, and S. D. Scott. Agnostic learning of geometric patterns. *Journal of Computer and System Sciences*, 62(1):123–151, 2001.

O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of STOC*, pages 25–32, 1989.

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

P. Gopalan, A. Kalai, and A. Klivans. Agnostically learning decision trees. In *Proceedings of STOC*, pages 527–536, 2008.

G. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of STOC*, pages 181–190, 2000.

V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Proceedings of FOCS*, pages 543–552, 2006.

S. Halevi and H. Krawczyk. Security under key-dependent inputs. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 466–475, 2007.

J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. ISSN 0004-5411.

J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992. ISSN 0890-5401.

A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008a.

A. Kalai, Y. Mansour, and E. Verbin. Agnostic boosting and parity learning. In *Proceedings of STOC*, pages 629–638, 2008b.

M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17 (2-3):115–141, 1994.

M. Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. *Journal of Computer and System Sciences*, 50:600–610, 1995.

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

W. S. Lee, P. L. Bartlett, and R. C. Williamson. On efficient agnostic learning of linear combinations of basis functions. In *Proceedings of COLT*, pages 369–376, 1995.

L. Levin. Randomness and non-determinism. *Journal of Symbolic Logic*, 58(3):1102–1103, 1993.

N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

Y. Mansour. Learning boolean functions via the fourier transform. In V. P. Roychodhury, K. Y. Siu, and A. Orlitsky, editors, *Theoretical Advances in Neural Computation and Learning*, pages 391–424. Kluwer, 1994.

R. O'Donnell and R. Servedio. Learning monotone decision trees in polynomial time. In *Proceedings of IEEE Conference on Computational Complexity*, pages 213–225, 2006.

D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.

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

J.H. van Lint. *Introduction to Coding Theory*. Springer, Berlin, 1998.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probab. and its Applications*, 16(2):264–280, 1971.