

On The Power of Membership Queries in Agnostic Learning

Vitaly Feldman*
IBM Almaden Research Center

April 28, 2008

Abstract

We study the properties of the agnostic learning framework of Haussler [Hau92] and Kearns, Schapire and Sellie [KSS94]. 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.* [KSS94]. 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).

1 Introduction

The agnostic framework [Hau92, KSS94] is a natural generalization of Valiant's PAC learning model [Val84]. 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 [KSS94, LBW95, GKS01, KKMS05, GKK08, KMV08]. Furthermore, strong computational hardness results are known for agnostic learning of even the simplest classes of functions such as parities, monomials and halfspaces [Hås01, Fel06, FGKP06, GR06] (albeit only for *proper* learning). Reductions from long-standing open problems for PAC learning to agnostic learning of simple classes of functions provide another indication of the hardness of agnostic learning [KSS94, KKMS05, FGKP06].

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

*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.

or ability to conduct experiments. Learning with membership queries in both PAC and Angluin’s exact models [Ang88] was studied in numerous works. For example monotone DNF formulas, finite automata and decision trees are only known to be learnable with membership queries [Val84, Ang88, Bsh95]. 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 introduced in a cryptographic context (even before the definition of the agnostic learning model) [GL89]. Their algorithm learns parities agnostically with respect to the uniform distribution using membership queries. Kushilevitz and Mansour used this algorithm to PAC learn decision trees [KM93] and it has since found numerous other significant applications. More efficient versions of this algorithm were also given by Levin [Lev93], Bshouty, Jackson and Tamon [BJT99] and Feldman [Fel07]. Recently, Gopalan, Kalai and Klivans gave an elegant algorithm that learns decision trees agnostically over the uniform distribution and uses membership queries [GKK08].

1.1 Our Contribution

In this work we study the power of membership queries in the agnostic learning model. This question was posed by Kearns *et al.* [KSS94] and, to the best of our knowledge, has not been addressed prior to our work. In this 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 Theorem 3.1 for a formal statement). This proves the conjecture of Kearns *et al.* [KSS94]. 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 [KSS94, KKMS05, KMV08].

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. 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 $\text{NP} \neq \text{P}$. Cryptographic assumptions are essential for numerous other hardness results in learning theory (*cf.* [KV94, Kha95]). 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 [ELSW07]. Sections 4.1 and 4.2 describe the technique and its relation to prior work in more detail.

This result 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 decoding of random linear codes, a long-standing open problem in coding theory. The best known algorithm for this problem runs in time $O(2^{n/\log n})$ [FGKP06]. It is therefore natural to expect that membership queries are provably helpful for uniform distribution agnostic learning. The proof of this 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 $\chi_a(x)$ as $\chi_a(x) = (-1)^{a \cdot x} = (-1)^{\oplus_{i \leq n} a_i x_i}$. 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. All of the above concept classes are in fact representation classes. 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 [Val84]. In this model, for a concept f and distribution D over X , an *example oracle* $\text{EX}(D, f)$ is the oracle that, upon request, returns an example $(x, f(x))$ where x is chosen randomly with respect to D . For $\epsilon \geq 0$ we say that function g ϵ -approximates a function f with respect to distribution D if $\Pr_D[f(x) = g(x)] \geq 1 - \epsilon$. In the PAC learning model the learner is given access to $\text{EX}(D, f)$ where f is assumed to belong to a fixed concept class \mathcal{C} .

Definition 2.1 *For a concept class \mathcal{C} , we say that an algorithm Alg PAC learns \mathcal{C} , if for every $\epsilon > 0$, $\delta > 0$, $f \in \mathcal{C}$, and distribution D over X , Alg , given access to $\text{EX}(D, f)$, 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 compute h are polynomial in $1/\epsilon, 1/\delta$ 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 a bound on the length of the description of a concept 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 .

2.2 Agnostic Learning Model

The *agnostic* learning model was introduced by Haussler [Hau92] and Kearns *et al.* [KSS94] 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 $D_A(z) = \Pr_{(x,b) \sim A}[x = z]$ and

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

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

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

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

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

Kearns *et al.* define agnostic learning as follows [KSS94].

Definition 2.2 *An algorithm Alg agnostically learns a concept class \mathcal{C} by a representation class \mathcal{H} assuming \mathcal{A} if for every $\epsilon > 0, \delta > 0, A \in \mathcal{A}$, Alg given access to examples drawn randomly from A , outputs, with probability at least $1 - \delta$, a hypothesis $h \in \mathcal{H}$ such that $\Delta(A, h) \leq \Delta(A, \mathcal{C}) + \epsilon$.*

The learning algorithm is *efficient* if it runs in time polynomial $1/\epsilon, \log(1/\delta)$ 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 for some \mathcal{H} .

A number of versions of the agnostic model 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 $\text{EX}(D, f)$. Following Kearns *et al.*, we refer to this version as *agnostic PAC learning* [KSS94] (they also require that $\mathcal{H} = \mathcal{C}$ but this constraint is unrelated and is now generally referred to as *properness*). Theorem 3.1 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 we refer to it as adversarial noise.

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 [Dud78], Pollard [Pol84], Haussler [Hau92], Vapnik [Vap98] 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 $m_{\mathcal{H}}(\epsilon, \delta)$ such that for every distribution A over examples, every $h \in \mathcal{H}$, $\epsilon > 0$, $\delta > 0$, the empirical error of h on sample of $m_{\mathcal{H}}(\epsilon, \delta)$ 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 will be sufficient for our purposes [VC71].

Theorem 2.3 *Let \mathcal{H} be a concept class over X of VC dimension d . Then for every distribution A over $X \times \{-1, 1\}$, every $h \in \mathcal{H}$, $\epsilon > 0$, $\delta > 0$, and sample S of size $m = O(d/\epsilon^2 \cdot \log(1/\delta))$ randomly drawn with respect to A ,*

$$\Pr[|\Delta(A, h) - \Delta(S, h)| \geq \epsilon] \leq \delta.$$

In fact a simple uniform convergence result based on the cardinality of the function class follows easily from Chernoff bounds (*cf.* [Hau92]). That is Theorem 2.3 holds for $m = O(\log |\mathcal{H}|/\epsilon^2 \cdot \log(1/\delta))$. 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 $f(z)$ [Val84]. We denote it by $\text{MEM}(f)$. We refer to agnostic PAC learning with access to $\text{MEM}(f)$ 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 $\text{MEM}(A)$ be the oracle that, upon query z , returns $b \in \{-1, 1\}$ with probability $\Pr_A[(x, b) \mid x = z]$. We say that $\text{MEM}(A)$ is *persistent* if given the same query the oracle responds with the same label.

2.4 Fourier Transform

Our separation result uses Fourier-analytic techniques introduced to learning theory by Linial, Mansour and Nisan [LMN93]. It is used primarily in the context of learning with respect to the uniform distribution and therefore 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 $|\hat{f}(a)| \geq \theta$. Let $L_2(f) = \mathbf{E}_x[(f(x))^2]^{1/2}$. 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,

$$\Pr_A[b \neq \chi_a(x)] = (1 - \hat{\phi}(a))/2. \tag{1}$$

Therefore, agnostic learning of parities amounts to finding the largest (within ϵ) Fourier coefficient of $\phi(x)$. The first algorithm for this task was given by Goldreich and Levin [GL89]. Given access to membership oracle, for every $\epsilon > 0$ their algorithm can efficiently find all ϵ -heavy Fourier coefficients.

Theorem 2.4 ([GL89]) *There exists an algorithm \mathbf{GL} that for every distribution $A = (U, \phi)$ and every $\epsilon, \delta > 0$, given access to $\text{MEM}(A)$, $\mathbf{GL}(\epsilon, \delta)$ 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 function families defined by Goldreich, Goldwasser and Micali [GGM86].

Definition 2.5 *A function family $\mathcal{F} = \{F\}_{n=1}^\infty$ where $F_n = \{\pi_z\}_{z \in \{0,1\}^n}$ is a pseudorandom function family if*

- *For every n and $z \in \{0, 1\}^n$, π_z is an efficiently evaluable Boolean function on $\{0, 1\}^n$.*
- *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 $\{-1, 1\}$ and a negligible function $\nu(n)$,*

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

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 .

Håstad *et al.* give a construction of pseudorandom function families based on the existence of one-way functions [HILL99].

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

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 $(x, 1)$ we remove it together with one example $(x, -1)$. Every function has the same error rate of $1/2$ with examples in $S \setminus S'$. Therefore for every function h ,

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

and hence

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

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

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

$$\Pr_{D_{S'}}[f(x) \neq g(x)] = \frac{1}{|S'|} |\{x \in S' \mid f(x) \neq g(x)\}| = \Delta(S', g).$$

That is, the error of any function g on $D_{S'}$ is exactly the empirical error of g on sample S' . Thus $\Delta((D_{S'}, f), h) = \Delta(S', h)$ and $\Delta((D_{S'}, f), \mathcal{C}) = \Delta(S', \mathcal{C})$. By the correctness of \mathbf{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} [VC71] (Theorem 2.3) to conclude that for

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

with probability at least $1 - \delta/2$, $\Delta(A, h) \leq \Delta(S, h) + \frac{\epsilon}{4}$ and $\Delta(S, \mathcal{C}) + \frac{\epsilon}{4} \leq \Delta(A, \mathcal{C})$ (we can always assume that $\mathcal{C} \subseteq \mathcal{H}$). 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. \square

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 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 3.2 *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 $\Delta(A, \mathcal{C}) + \epsilon$, it can be easily extended to weaker forms of agnostic learning, such as algorithms that only guarantee error bounded by $\alpha \cdot \Delta(A, \mathcal{C}) + \beta + \epsilon$ for some $\alpha \geq 1$ and $\beta \geq 0$. This is true since the reduction adds at most $\epsilon/2$ to the error of the original algorithm (and the additional time required is polynomial in $1/\epsilon$).*

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 $\epsilon \geq 1/p(n)$ 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 (*cf.* [KKMS05, OS06, GKK08]).

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.* [ELSW07] and Feldman, Shah and Wadhwa [FSW07]. 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 $1 - 2\epsilon$ 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 $\bar{S} = \{0, 1\}^n \setminus S$ with any significant advantage over random guessing. Let f' be a function equal to f on \bar{S} and truly random on S . Then

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

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 \bar{S} . This means that the error of any efficient algorithm will be higher by at least $|\bar{S}|/2^{n+1}$ than the best possible. To ensure that this difference is at most ϵ , we need $|S| \geq (1 - 2\epsilon)2^n$.

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 that is the concatenation of Reed-Solomon code with the binary Hadamard code [GS00]. 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 $1/2$ [GL89]. Therefore the Hadamard code can be decoded by reading the code in only a polynomial number of (randomly-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. While it is not known how to learn parities with noise from random examples alone and this problem is conjectured to be very hard, for all we know, it is possible that one-way functions exist whereas learning of parities with noise is tractable. It is known however that if learning of parities with noise is hard then one-way functions exist [BFKL93]. Our solution to this problem is to use a pseudo-random function to make values on random examples indistinguishable from random coin flips. Specifically, let $a \in \{0, 1\}^n$ be the vector we want to encode and let $b : \{0, 1\}^n \rightarrow \{-1, 1\}$

be a pseudo-random function. We define a function $g : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{-1, 1\}$ as

$$g(z, x) = b(z) \oplus \chi_a(x) .$$

(\oplus is simply the product in $\{-1, 1\}$). The label of a random example $(z, x) \in \{0, 1\}^{2n}$ 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 examples 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 , $b(z) \oplus \chi_a(x)$ 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.* in their separation result [ELSW07].

Finally, the problem with the construction we have so far is that while a membership query learning algorithm can find the secret, it cannot predict the encoding of the secret $g(z, x)$ without knowing $b(z)$. This means that we also need to provide a description of $b(z)$ to the learning algorithm. It is tempting to use the Hadamard code to encode the description of $b(z)$ together with a . However, a bit of the encoding of b is no longer independent of $b(z)$, and therefore the previous argument does not hold. We refer to the vector that describes $b(z)$ by $d(b)$. 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. An identical problem also arises in the construction of Elbaz *et al.* [ELSW07]. They used another pseudorandom function b_1 to encode $d(b)$, then used another pseudorandom function b_2 to encode $d(b_1)$ and so on. The fraction of the domain used up for the encoding of $d(b_i)$ 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 $d(b_i)$ for every odd i can be completely corrupted making all the other encodings unrecoverable. To solve this problem in our construction we use a pseudorandom function b_i to encode $d(b_j)$ for all $j < i$. We also use encodings of the same size. In this construction at most one of the encodings that are not completely corrupted cannot be recovered. It is the encoding with $b_i(z)$ such that the encodings with $b_j(z)$ are completely corrupted for all $j > i$ (since those are the ones that contain the encoding of $d(b_i)$). Therefore by making the number of encodings larger than $1/\epsilon$, 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 brief 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

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

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

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

Let $\mathcal{F} = \{\pi_y\}_{y \in \{0, 1\}^*}$ be a pseudorandom function family (Definition 2.5). We define $g_{\bar{d}} : \{0, 1\}^m \rightarrow \{-1, 1\}$ as follows:

$$g_{\bar{d}}(k, z, \bar{x}) = \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) \quad (4)$$

Denote

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

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 analogous to the proof by Elbaz *et al.* who show that the same holds for the concept class they define [ELSW07].

Theorem 4.1 *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 $g_{\bar{d}}(k, z, \bar{x})$ 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 $1/2$, produces a hypothesis with error of at most $1/2 - 1/q(m)$ and runs in time polynomial in $t(m)$ for some polynomials t and q . 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 $\rho_{i+1}, \rho_{i+2}, \dots, \rho_k$ from the set of all Boolean functions over $\{0, 1\}^n$. Upon request such an oracle returns an example $((k, z, \bar{x}), b)$ where (k, z, \bar{x}) is chosen randomly and uniformly from $\{0, 1\}^m$ and

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

We note that in order to simulate such an oracle it is not needed to explicitly choose $\rho_{i+1}, \rho_{i+2}, \dots, \rho_k$. 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. We denote by δ_i the probability of the following event: **Alg** with oracle $\mathbb{O}(i)$ outputs a hypothesis that has error of at most $1/2 - 2/(3q(m))$ relative to $\mathbb{O}(i)$. We refer to this condition as success. The error is obtained by estimating it on new random examples from $\mathbb{O}(i)$ to within $1/(3q(m))$ and with probability at least $7/8$. The probability is taken over the random choice and simulation of $\mathbb{O}(i)$ and the coin flips of **Alg**. The bounds on the running time of **Alg** and Chernoff bounds imply that this test can be performed in time polynomial in m .

Claim 4.2 $\delta_p - \delta_0 \geq 1/4$.

Proof: To see this we first observe that $\mathbb{O}(0)$ is a truly random oracle and therefore the error of the hypothesis produced by **Alg** is at least $1/2 - \nu(m)$ for some negligible ν . This means that the error estimate can be lower than $1/2 - 2/(3q(m))$ only if the estimation fails. By the definition of our error estimation procedure this implies that $\delta_0 \leq 1/8$. On the other hand, $\mathbb{O}(p)$ is equivalent to $\text{EX}(U, g_{\bar{d}})$ for a 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 $7/8$ the estimate of the error is correct and therefore $\delta_p \geq 3/8$. $\square(\text{Cl.4.2})$

We now describe our distinguisher M . Let $\pi(x)$ denote the function given to M as an oracle. Our distinguisher chooses a random $i \in p$ and a random oracle $\mathbb{O}(i)$ as described above but using the oracle π in place of $\pi_{\bar{d}^i}$. That is it generates examples $((k, z, \bar{x}), b)$ where (k, z, \bar{x}) is chosen randomly and uniformly from $\{0, 1\}^m$ and

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

Denote this oracle by $\mathbb{O}^\pi(i)$. The distinguisher simulates **Alg** with examples from $\mathbb{O}^\pi(i)$ and outputs 1 whenever the test of the output of **Alg** is successful.

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 M will output 1 with probability

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

On the other hand, if π is a truly random function then $\mathbb{O}^\pi(i)$ is equivalent to $\mathbb{O}(i-1)$ and hence the simulator will output 1 with probability

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

Therefore, by Claim 4.2 this implies that M distinguishes F_n from a truly random function with probability at least

$$\frac{1}{p(n)} \left(\sum_{i \in [p]} \delta_i - \delta_{i-1} \right) \geq \frac{1}{p(n)} (\delta_p - \delta_0) \geq 1/4p(n).$$

The efficiency of M follows readily from the efficiency of the test we demonstrated above and gives us the contradiction to the properties of \mathcal{F} . $\square(\text{Th.4.1})$

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 $\epsilon \geq 1/p(n)$.

Theorem 4.3 *There exists a randomized algorithm **AgnLearn** that for every distribution $A = (U, \phi)$ over $\{0, 1\}^m$ and every $\epsilon \geq 1/p(n), \delta > 0$, given access to $\text{MEM}(A)$, with probability at least $1 - \delta$, finds h such that $\Delta(A, h) \leq \Delta(A, \mathcal{C}_n^p) + \epsilon$. The probability is taken over the coin flips of $\text{MEM}(A)$ and **AgnLearn**. **AgnLearn** runs in time polynomial in m and $\log(1/\delta)$.*

Proof: Let $g_{\bar{e}}$ for $\bar{e} = (e^1, e^2, \dots, e^{p-1}) \in \{0, 1\}^{(p-1) \times n}$ 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 encoding A agrees with the encoding of $\bar{e}(j) = (e^1, e^2, \dots, e^{j-1}, 0^n, \dots, 0^n)$ with probability at least $1/2 + \epsilon/4$. Such j can be used to find $\bar{e}(j)$ and therefore allows us to reconstruct $g_{\bar{e}}$ on all points (k, z, \bar{x}) for $k < j$. For points with $k \geq j$ our hypothesis is 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

$$1/2 - \epsilon/4 - 1/(2p) \geq 1/2 - \epsilon$$

on this part of the domain and therefore the hypothesis has error close to that of $g_{\bar{e}}$.

We now describe **AgnLearn** formally. For every $i \in [p]$, **AgnLearn** chooses $y \in \{0, 1\}^n$ randomly and uniformly. Then **AgnLearn** runs Goldreich-Levin algorithm over $\{0, 1\}^{(p-1) \times n}$ using $\text{MEM}(A_{i,y})$. When queried on a point $\bar{x} \in \{0, 1\}^{(p-1) \times n}$ $\text{MEM}(A_{i,y})$ returns the value of $\text{MEM}(A)$ on query (i, y, \bar{x}) . That is $\text{MEM}(A_{i,y})$ is a restriction of A to points in $\{0, 1\}^m$ with prefix i, y . Let T denote the set of indices of heavy Fourier coefficients returned by $\text{GL}(\epsilon/4, 1/2)$. For each vector $\bar{d} \in T$ and $b \in \{-1, 1\}$, let $h_{\bar{d},i,b}$ be defined as

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

(Here π_{d^k} is an element of the pseudorandom function family \mathcal{F} used in the construction.) Next **AgnLearn** approximates $\Delta(A, h_{\bar{d},i,b})$ to within accuracy $\epsilon/8$ with confidence $1 - \delta/t$ using random samples from A (for t to be defined later). We denote the estimate obtained by $\tilde{\Delta}_{\bar{d},i,b}$. **AgnLearn** repeats this r times (generating new y each time) and returns $h_{\bar{d},i,b}$ for which $\tilde{\Delta}_{\bar{d},i,b}$ is the smallest. For $i = 1$ and any \bar{d} , $h_{\bar{d},1,b} \equiv b$. Therefore for $i = 1$ instead of the above procedure **AgnLearn** tests two constant hypotheses $h_1 \equiv 1$ and $h_{-1} \equiv -1$.

Claim 4.4 *For $t = O(p \cdot \log(1/\delta)/\epsilon^3)$ and $r = O(\log(1/\delta)/\epsilon)$, with probability at least $1 - \delta$, **AgnLearn** returns h such that $\Delta(A, h) \leq \Delta(A, \mathcal{C}_n^p) + \epsilon$.*

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

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

For $i \in [p]$, denote

$$\Delta_i = \mathbf{Pr}_{((k,z,\bar{x}),b) \sim A} [b \neq g_{\bar{e}}(k, z, \bar{x}) \mid k = i].$$

By the definition,

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

Let j be the largest i such that $\Delta_{i'} \leq 1/2 - \epsilon/4$ and for all $i' > i$, $\Delta_{i'} > 1/2 - \epsilon/4$. If such j does not exist then $\Delta(A, g_{\bar{e}}) > 1/2 - \epsilon/4$. Either h_1 or h_{-1} has error of at most $1/2$ on A and therefore for $i = 1$ **AgnLearn** will find a hypothesis h' such that $\Delta(A, h') \leq \Delta(A, g_{\bar{e}}) + 3\epsilon/4$.

We can now assume that j as above exists. Denote

$$\Delta_{i,y} = \mathbf{Pr}_{((k,z,\bar{x}),b) \sim A} [b \neq g_{\bar{e}}(k, z, \bar{x}) \mid k = i, z = y].$$

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 $\epsilon/4$, $\Delta_{j,y} \leq 1/2 - \epsilon/8$. This is true since otherwise

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

contradicting the choice of j . We now note that by the definition of $A_{i,y}$,

$$\Delta_{i,y} = \mathbf{Pr}_{(\bar{x},b) \sim A_{i,y}} [b \neq g_{\bar{e}}(i, y, \bar{x})].$$

The function $g_{\bar{e}}(i, y, \bar{x})$ equals $\pi_{d^j}(y) \oplus \chi_{\bar{e}(j)}(\bar{x})$, and therefore if $\Delta_{i,y} \leq 1/2 - \epsilon/8$ then by equation (1),

$$|\widehat{A_{i,y}}(\bar{e}(j))| \geq \epsilon/4.$$

This implies that $\mathbf{GL}(\epsilon/4, 1/2)$ with $\mathbf{MEM}(A_{i,y})$ will return $\bar{e}(j)$ (possibly, among other vectors). Let

$$b_j = \mathbf{sign}(\mathbf{E}_{((k,z,\bar{x}),b) \sim A} [b \mid k \geq j])$$

be the constant with the lowest error on examples from A for which $k \geq j$. Clearly, this error is at most $1/2$. The hypothesis $h_{\bar{e}(j),j,b_j}$ equals $g_{\bar{e}}$ on points for which $k < j$ and equals b_j on the rest of the points. Therefore

$$\Delta(A, h_{\bar{e}(j),j,b_j}) \leq \frac{1}{p} \left(\sum_{i < j} \Delta_i + \frac{p-j+1}{2} \right).$$

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

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

By combining these equations we obtain that

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

All that is left to show now are the choices of r and t for which the desired h will be found with probability at least $1 - \delta$. As we have observed, for a randomly and uniformly chosen y , with probability at least $\epsilon/4$, $\Delta_{j,y} \leq 1/2 - \epsilon/8$ and in this case $\text{GL}(\epsilon/4, 1/2)$ will find $\bar{e}(j)$ with probability at least $1/2$. By repeating this procedure $O(\log(1/\delta)/\epsilon)$ times we can ensure that $\bar{e}(j)$ is found with probability at least $1 - \delta/2$. By Parseval's identity there are $O(1/\epsilon^2)$ elements in each set of vectors returned by GL . Hence the number of error estimations performed by AgnLearn is $O(p \cdot r/\epsilon^2)$. This means that for $t = O(p \cdot \log(1/\delta)/\epsilon^3)$ all estimations will be within $\epsilon/8$ with probability $1 - \delta/2$. $\square(\text{Cl.4.4})$

Given Claim 4.4, we only need to check that the running time of AgnLearn is polynomial in m and $\log(1/\delta)$. This follows easily from the polynomial bound on the running time of GL and computation of each $\pi \in F_n$, and polynomial number of samples required to estimate the errors of the candidate hypotheses. $\square(\text{Th.4.3})$

4.6 Bounds on ϵ

Theorem 4.3 shows that \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 $\epsilon \geq 1/p(n)$. This means that this construction cannot achieve dependence on ϵ beyond $1/m$. To improve this dependence we can use a more efficient encoding scheme in place of Hadamard code. Let $C : \{0, 1\}^k \rightarrow \{0, 1\}^v$ be a binary code of message length k and block length v . The following properties of the code are required by our construction:

- Efficient encoding algorithm. For any $z \in \{0, 1\}^k$ and $j \leq v$, $C(z)_j$ (the j^{th} bit of $C(z)$) is computable in time polynomial in k and $\log v$.
- Efficient local list decoding from $(1/2 - \gamma)v$ errors in time polynomial in k and $1/\gamma$ for any $\gamma \geq \epsilon/8$. 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 $\Pr_{j \in [v]}[C(z)_j \neq y_j] \leq 1/2 - \gamma$ (in time polynomial in k and $1/\gamma$).

Guruswami and Sudan gave a list decoding algorithm for Reed-Solomon code concatenated with Hadamard code that has the desired properties for $v = O(k^2/\epsilon^4)$ [GS00] (see also [Tre05, Lecture 14] for a simplified presentation). Note that this is exponentially more efficient than Hadamard code for which $v = 2^k$. In fact for this code we can afford to read the whole codeword 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 Goldreich-Levin algorithm).

In our construction $k = n(p(n) - 1)$. To apply the above code we index a position in the code using $\log v = O(\log(n/\epsilon))$ 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 $\epsilon \geq 1/p(n)$. This implies that our learning algorithm is successful for every $\epsilon \geq 1/p(n) \geq 1/p(m)$. It is easy to verify that Theorems 4.1 and 4.3 still hold for this variant of the construction.

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.

Acknowledgments

I thank Parikshit Gopalan, Salil Vadhan and David Woodruff for valuable discussions and comments on this research.

References

- [Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
- [BFKL93] 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.
- [BJT99] 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.
- [Bsh95] N. Bshouty. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153, 1995.
- [Dud78] R. Dudley. Central limit theorems for empirical measures. *Annals of Probability*, 6(6):899–929, 1978.
- [ELSW07] 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.
- [Fel06] V. Feldman. Optimal hardness results for maximizing agreements with monomials. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 226–236, 2006.
- [Fel07] V. Feldman. Attribute efficient and non-adaptive learning of parities and DNF expressions. *Journal of Machine Learning Research*, (8):1431–1460, 2007.
- [FGKP06] 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.
- [FSW07] V. Feldman, S. Shah, and N. Wadhwa. Separating models of learning with faulty teachers. In *Proceedings of ALT*, pages 94–106, 2007.
- [GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
- [GKK08] P. Gopalan, A. Kalai, and A. Klivans. Agnostically learning decision trees. In *Proceedings of STOC*, 2008.

- [GKS01] 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.
- [GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of STOC*, pages 25–32, 1989.
- [GR06] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Proceedings of FOCS*, pages 543–552, 2006.
- [GS00] G. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of STOC*, pages 181–190, 2000.
- [Hås01] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.
- [Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.
- [HILL99] 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.
- [Kha95] M. Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. *Journal of Computer and System Sciences*, 50:600–610, 1995.
- [KKMS05] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of FOCS*, pages 11–20, 2005.
- [KM93] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.
- [KMV08] A. Kalai, Y. Mansour, and E. Verbin. Agnostic boosting and parity learning. In *Proceedings of STOC*, 2008.
- [KSS94] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.
- [KV94] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.
- [LBW95] 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.
- [Lev93] L. Levin. Randomness and non-determinism. *Journal of Symbolic Logic*, 58(3):1102–1103, 1993.
- [LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.
- [OS06] 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.

- [Pol84] D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.
- [Tre05] L. Trevisan. Pseudorandomness and combinatorial constructions (lecture notes). Available at <http://www.cs.berkeley.edu/~luca/pacc/>, 2005.
- [Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- [Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.
- [VC71] 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.