CS 395T: Learning Theory Fall 2018

Lower Bounds for Learning

Prof. Adam Klivans *Scribe: Kuan-Yi Ho, Kai-Chi Huang, Jiahui Liu*

1 Introduction

In this short survey we will discuss the hardness of PAC learning, presenting some recent theoretic results in lower bounds for learning and its relation to refutation.

1.1 History Results on Hardness of Learning

Assuming $P \neq NP$, proper PAC learning AND of hyperspace is impossible. [KS09] showed that assuming the hardness of certain lattice problem, improper learning intersection of halfspace is hard. Then in [DS16] it was shown that assuming hardness of refuting random k-SAT, improper learning DNF is hard and improper learning intersection of halfspace is hard. The result of [DS16] was simplified and generalized in [Vad17].

In this survey we mainly discuss the results in [Vad17] – equivalence between learning and refutation and in [KL18], equivalence between agnostic learning over some distribution $\mathcal{D}$ and refutation (with noise) with respect to $\mathcal{D}$.

2 Preliminaries and Definitions

2.1 PAC Learning

Definition 2.1 (PAC Learnable)**.** *$\mathcal{P}$ is PAC learnable if there is an algorithm $\mathcal{A}$ such that for every distribution $\mathcal{D}$, after training in polynomial time and polynomial sample sampled from $\mathcal{D}$, given the next example x, predicts its label y correctly with probability $\geq 2/3$.*

This definition is equivalent to the standard (ϵ, δ) definition within a polynomial factor.

Definition 2.2 (Weakly PAC Learnable)**.** *$\mathcal{P}$ is weakly PAC learnable with advantage α and sample complexity m if there is an algorithm $\mathcal{A}$ such that for every distribution $\mathcal{D}$, after training in polynomial time and polynomial samples from $\mathcal{D}$, given the next example x, predicts its label y correctly with probability $\geq (1+\alpha)/2$.*

It is known that weakly PAC learnability implies full-fledged PAC learnability.

Theorem 2.3 (Schapire (1990) Boosting)**.** *If $\mathcal{P}$ is weakly PAC learnable with sample complexity m and advantage α, then $\mathcal{P}$ is PAC learnable with sample complexity $m \cdot \mathrm{poly}(\frac{1}{\alpha})$.*

For the proofs and theorems given in this survey, we will use the constant probability for convenience.

2.2 Dual Classes

The Dual class is a useful notion we will use in later theorems and proofs.

Definition 2.4 (Dual Classes)**.** *Given an evaluation function* $\mathrm{Eval} : \{0,1\}^s \times \{0,1\}^t \to \{0,1\}$, *we define*

- $\mathcal{P} = \{p_x : \{0,1\}^t \to \{0,1\} | x \in \{0,1\}^s\}$
- $\mathcal{Q} = \{q_y : \{0,1\}^s \to \{0,1\} | y \in \{0,1\}^t\}$
- $p_x(y) = q_y(x) = \mathrm{Eval}(x, y)$

To get some intuition of this definition, suppose $\mathcal{P}$ is the set of possible concepts. Then, p_x is some concept indexed by x and $p_x(y) = 1$ iff x is satisfied by some assignment y. Also, $\mathcal{Q}$ can be seen as the set of possible assignments. Then, q_y is some equation indexed by some assignment y and $q_y(x) = 1$ iff y satisfies x. On the other hand, if we think of $\mathcal{P}$ as the set of possible assignments and $\mathcal{Q}$ as the concepts: $p_x(y) = 1$ iff x satisfies y. This shows how they form duality. Moreover, the definition of dual class gives the explicit connection between the problem of identifying satisfiable constraints and the problem of identifying realizable samples. We will denote $\mathcal{P}^* = \mathcal{Q}$. That is, $\mathcal{Q}$ is the dual class of $\mathcal{P}$.

2.3 Random-Right-Hand-Side(RRHS) Refutation

Definition 2.5 (RRHS-refutable)**.** *$\mathcal{Q}$ is RRHS-refutable using n equations if there is a polynomial-time algorithm $\mathcal{B}$ that can distinguish the following two cases,*

- **Satisfiable***: For every $y_1, \dots, y_n \in \{0,1\}^t$ and every right hand side $b_1, \dots, b_n \in \{0,1\}$, if the system of equations $q_{y_i}(x) = b_i$ has a solution $x \in \{0,1\}^s$, then $\mathcal{B}$ must output* 1 *with probability* $\leq 1/3$.
- **Random***: For every $y_1, \dots, y_n \in \{0,1\}^t$, if the right hand sides b_i are randomly sampled i.i.d. from* $\mathrm{Unif}(\{0,1\})$, *then $\mathcal{B}$ must output* 1 *with probability* $\geq 2/3$.

We assume that we won't be given any unsatisfiable but not random labels (There will be no partially true labels). $\mathcal{B}$ can distinguish between **satisfiable** and **random** labels with advantage 1/3.

And using the Dual classes definition, we can reformulate the RRHS-refutation problem as follows:

Definition 2.6 (RRHS-refutable)**.** *$\mathcal{Q} = \mathcal{P}^*$ is RRHS-refutable using n equations if there is a polynomial-time algorithm $\mathcal{B}$ such that, for every $y_1, \dots, y_n \in \{0,1\}^t$ and some labels $b_1, \dots, b_n \in \{0,1\}$,*

- **Satisfiable***: If the labels are generated from some concept p_x, i.e. $b_i = p_x(y_i)$, then $\mathcal{B}$ must output* 1 *with probability* $\leq 1/3$.
- **Random***: If the labels b_i are randomly sampled i.i.d. from* $\mathrm{Unif}(\{0,1\})$, *then $\mathcal{B}$ must output* 1 *with probability* $\geq 2/3$.

2.4 Random k-SAT Refutation and Assumption

Definition 2.7 (Refuting Random k-SAT)**.** *The refuter algorithm $\mathcal{B}$ wants to distinguish the following two cases.*

- **Satisfiable***: For every set of n k-disjunctions on s variables and, if all of them are satisfiable, then $\mathcal{B}$ outputs* 1 *with probability* $\leq 1/3$.
- **Random***: If all n k-disjunctions on s variables are chosen independently and uniformly, then $\mathcal{B}$ outputs* 1 *with probability* $\geq 2/3$.

We make the difference between standard refutation and RRHS-refutation explicit here. In the standard refutation problem, the right hand bits (b_i) of each equation is always 1 and when considering the random case, the randomness comes from the uniform sample of constraints, e.g. disjunctions in the case of k-SAT. In contrast, in the RRHS-refutation problem, b_i can be 0 and in the random case, the randomness comes from the uniformly sampled b_i.

Assumption 2.8 (Refuting Random k-SAT is Hard)**.** *For any sufficiently large s and any polynomial $n = n(s)$, there is a constant k s.t. there is no polynomial time algorithm $\mathcal{B}$ distinguishing the above two case in definition 2.7.*

3 PAC-learning DNF Implies Refuting Random k-SAT

The main conceptual contribution in [DS16] is to interpret k-SAT problem (or more generally, constraint satisfiable problem) as a learning problem, which is reformulated in [Vad17] using definition 2.4. Since all the techniques and concepts in [DS16] are contained in [Vad17], we will use the terminology in [Vad17].

Theorem 3.1. *Suppose assumption 2.8 holds. Then, there is no polynomial time algorithm that learns the class of DNF formulas.*

The proof proceeds in two steps. First, reduce the problem of refuting random k-SAT to the problem of RRHS-refuting k-CNF and then show that learning DNF implies RRHS-refuting k-CNF. Lemma 3.2 is first presented in [DS16] but is rather implicit. It is further simplified and made explicit in [Vad17]. We will present the simplified proof here.

Lemma 3.2. *If k-CNF formulas on s variables with $m = \lceil 2^k \cdot \ln(4n) \rceil$ are RRHS-refutable with n equaltions, then random k-SAT on s variables is refutable using $n' = O(n \cdot m)$ equations*

Proof Sketch. We show this by reducing the instance of random k-SAT to an instance of k-CNF formulas.
Given $n \cdot m$ number of k-way disjunctions $\phi_1, ..., \phi_{nm}$, for $i = 1, ..., n$, construct:

- with probability $\frac{1}{2}$, let ψ_i be the conjunction of first m disjunctions from $\phi_1, ..., \phi_{nm}$ which have not been used in $\psi_1, ..., \psi_{i-1}$; set the right hand bit $b_i = 1$
- with probability $\frac{1}{2}$, let ψ_i be the conjunction of m uniformly random and independent k-way disjuctions; set $b_i = 0$

Feed the constructed $(\psi_1, b_1), ..., (\psi_n, b_n)$ to the k-CNF RRHS-refuter.
We show that if $\phi_1, ..., \phi_{nm}$ are random, then $(\psi_1, b_1), ..., (\psi_n, b_n)$ are also random. This part is simple . If $\phi_1, ..., \phi_{nm}$ are random then the distribution of ψ_i is the same in the case of $b_i = 1$ as in the case of $b_i = 0$. Also, b_i's are uniformly random and independent of ψ_i. By the property of k-CNF RRHS-refuter, it will output 1 with probability $\geq 2/3$.
On the other hand, we show that if $\phi_1, ..., \phi_{nm}$ are satisfiable by some assignment α, $(\psi_1, b_1), ..., (\psi_n, b_n)$ is also satisfied by α with high probability. For those b_i's such that we set $b_i = 1$, it is clear that they will always be satisfied. For those b_i's that we set $b_i = 0$, note that α will satisfied a random k-CNF with $m = O(2^k \ln(4n))$ with prob. $(1 - 2^{-k})^m \leq 1/4n$. Thus, $\psi_i(\alpha) = 0$ will hold with prob. $> 1 - 1/4n$. By union bound, all the equations are satisfied by α with prob. $> 1 - 1/4 = 3/4$. Thus, the probability that k-CNF refuter outputs 1 is $\leq 1/4 + 1/3$. We conclude the lemma by noting that the gap can be amplified by repetition. □

If we consider that k-CNF is not RRHS-refutable as a hardness assumption (note that, by De-Morgan's law, it is equivalent to stating that k-DNF is hard to RRHS-refute), then lemma 3.2 essentially shows the assumption that k-CNF is hard to RRHS-refute is weaker than that refuting random k-SAT is hard. In fact, we will see in Section 4 that we can obtain an equivalence between RRHS-refuting k-CNF (equivalently, k-DNF) and learning k-DNF. Lemma 3.2 together with corollary 5.7 in Section 4 give the proof of Theorem 3.1.

4 Equivalence between PAC-Learnable and RRHS-Refutable for Dual Classes

Theorem 4.1 (Equivalence between PAC-Learnable and RRHS-Refutable)**.** *([Vad17]) Let $\mathcal{P} = \mathcal{Q}^*$, then*

1. *If $\mathcal{P}$ is PAC learnable with sample complexity m, then $\mathcal{Q}$ is RRHS-refutable using $O(m)$ equations.*
2. *If $\mathcal{Q}$ is RRHS-refutable using n equations, then $\mathcal{P}$ is PAC learnable with sample complexity* $\text{poly}(n)$.

Proof Sketch. 1. **PAC-Learnable Implies RRHS-Refutable:** We want to use a PAC learner to check if a system of RRHS equations is **Satisfiable** or **Random**, i.e. check if the labels are **Satisfiable** or **Random**? The high-level idea is that we split the equations to training and testing sets.

- If **Satisfiable**: with high probability, the prediction on the testing set will be correct.
- If **Random**: the prediction on the testing set will be garbage outputs (i.e. random guesses).

Since PAC learner requires the examples to be sampled i.i.d. from some distribution $\mathcal{D}$, we can't split the training set arbitrarily. So we set $\mathcal{D}$ be uniform on all M examples. Then we follow the standard procedure of PAC learner: firstly,sample m examples (y_i, b_i) i.i.d. for training, and then sample an extra (y', b') as the testing example.

If M is large enough, there is at least a probability of $1 - \epsilon$ that (y', b') doesn't appear in the m training examples. This only requires $M = O(m)$.

- If **Satisfiable**: The prediction will be correct according to the property of PAC learner, which is correct probability $\geq 2/3$.
- If **Random**: Unless (y', b') appears in the training samples, b' will be a new random label that $\mathcal{B}$ has never seen. Since the training set is random, the prediction will be completely random, i.e. correct probability $= 1/2$. Overall correct rate is less than $1/2 + \epsilon$.

The refuter algorithm outputs 0 if the prediction on the testing example is correct, and 1 otherwise.

2. **RRHS-Refutable Implies PAC-Learnable**

   Now we construct a PAC learner to learn a concept with only access to a RRHS refuter, i.e. only able to distinguish between all-true and all-random labels. PAC learning can be considered as the problem of distinguishing between **all true** and **all true except one false** samples:

   (a) Last label is correct: $(y_1, b_1), \ldots, (y_m, b_m), (y_{m+1}, b_{m+1}) - p_1$

   (b) Last label is random: $(y_1, b_1), \ldots, (y_m, b_m), (y_{m+1}, c) - p_2$

   (c) Last label is wrong: $(y_1, b_1), \ldots, (y_m, b_m), (y_{m+1}, \neg b_{m+1}) - p_3$

$$
\begin{aligned}
p_2 &= \frac{1}{2}(p_1 + p_3) \\
p_3 - p_1 &= 2(p_2 - p_1)
\end{aligned}
$$

   If we can distinguish Case (a) and Case (b) with advantage γ, then we have a weak PAC learner with advantage γ. To achieve this, we introduce the **next bit predictor**.

   Lemma 4.2 (Yao 1982)**.** *If $\mathcal{B}$ is a RRHS refuter for m equations with advantage γ, then there is a random next-bit predictor $\mathcal{B}'$ that predicts a random next bit with advantage $\frac{\gamma}{m}$.*

   Proof Sketch. Given the examples $(y_1, \ldots, y_m)$, consider the probability of $\mathcal{B}$ outputing 1 on the following labels:
   (All $\mathbf{b}_i$ are true labels, all c_i are i.i.d. random bits)

   - $(c_1, c_2, \ldots, c_{m-1}, c_m) : p_0$
   - $(\mathbf{b}_1, c_2, \ldots, c_{m-1}, c_m) : p_1$
   - $(\mathbf{b}_1, \mathbf{b}_2, \ldots, c_{m-1}, c_m) : p_2$
   - $\vdots$
   - $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1}, c_m) : p_{m-1}$
   - $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1}, \mathbf{b}_m) : p_m$

We have $p_m - p_0 \geq \gamma$ and $\exists i$ such that $p_i - p_{i-1} \geq \frac{\gamma}{m}$. We can make $\mathcal{B}$ a weak i-th bit predictor by checking the output of $\mathcal{B}$, given the previous $i-1$ bits and a uniformly random x_i as input. □

Now we can construct a weak learner.

(a) **Randomly** choose $i \sim \text{Unif}(\{1, \ldots, m\})$

(b) The expected advantage is $\frac{1}{m}\sum_{i=1}^{m}(p_i - p_{i-1}) = \frac{1}{m}(p_m - p_0) \geq \frac{\gamma}{m}$

(c) Examples don't have specific order

(d) $\mathcal{B}$ works for **all** combinations of examples.

(e) First choose i, then reorder the examples such that the unknown example is at position i

We now have a weak PAC learner with advantage $\frac{\gamma}{m}$. Using Boosting, we obtain a PAC learner with sample complexity poly(m). □

We have now shown the equivalence for PAC learning and RRHS refutation for dual classes; in the next section we want to "remove" the duality and show their equivalence for the same class.

5 DNF PAC Learnable Equals RRHS-Refutable

First, we define a reduction scheme called PAC-reduction that preserves PAC-learnability and RRHS-refutability.

Definition 5.1. *Let $\mathcal{P} = \mathcal{Q}^*$ and $\mathcal{P}' = (\mathcal{Q}')^*$ be two classes of boolean function class given by evaluation functions* $\text{Eval}(\cdot,\cdot)$ *and* $\text{Eval}'(\cdot,\cdot)$ *respectively. Then we say $\mathcal{P}$ PAC-reduces to $\mathcal{P}'$, written $\mathcal{P} \leq_{pac} \mathcal{P}'$ (and $\mathcal{Q} \leq_{pac} \mathcal{Q}'$), if there exists polynomial-time computable functions $f : \{0,1\}^s \to \{0,1\}^{s'}$ and $g : \{0,1\}^t \to \{0,1\}^{t'}$ with $s', t' = \text{poly}(s,t)$, such that for all s, t, $x \in \{0,1\}^s, y \in \{0,1\}^t$,*

$$\text{Eval}'(f(x), g(y)) = \text{Eval}(x, y)$$

.

Proposition 5.2. *Suppose $\mathcal{Q}^* = \mathcal{P} \leq_{pac} \mathcal{P}' = (\mathcal{Q}')^*$. Then,*

1. *If $\mathcal{P}'$ is PAC-learnable with sample complexity m, then so is $\mathcal{P}$.*
2. *If $\mathcal{Q}'$ is RRHS-refutable with n equations, then so is $\mathcal{Q}$.*

First, we will show the PAC-equivalence between DNF and its own dual DNF^*.

Definition 5.3. *A DNF formula f is said to be* **monotonic** *if it contains no negated terms. The set of such formulas is denoted as $MONDNF$.*

Lemma 5.4 ((Kearns et al. (1987)) D)**.** $DNF \leq_{pac} MONDNF$

Proof. For every DNF formula $\phi(x_1,\dots,x_t)$ with t variable, define $f(\phi)$ as replacing all negated variable $\neg x_i$ with a new variable x_{t+i}, then $f(\phi)$ is a monotonic DNF formula with $\phi(x_1,\dots,x_t) = f(\phi)(x_1,\dots,x_{2t})$, and the mapping f can be computed in polynomial time. This gives a PAC-reduction from DNF to $MONDNF$. □

Lemma 5.5. *$MONDNF \leq_{pac} DNF^*$*

Proof. Let ϕ be a monotonic DNF formula of s clauses,

$$\phi(x_1,\dots,x_t) = c_1 \vee \dots \vee c_s = \bigvee_{i=1}^{s} \bigwedge_{x_j \in c_i} x_j$$

We encode ϕ as a bitstring a of length $s \cdot t$: $a_{i,j} = 1$ iff x_j is in clause c_i. Then,

$$\begin{aligned}
\phi(x_1,\dots,x_t) &= \bigvee_{i=1}^{s} \bigwedge_{j:a_{i,j}=1} x_j \\
&= \bigvee_{i=1}^{s} \bigwedge_{j=1}^{t} \neg a_{i,j} \vee x_j \\
&= \bigvee_{i=1}^{s} \bigwedge_{j:x_j=0}^{t} \neg a_{i,j} \\
&=: \psi_x(a_{1,1},\dots,a_{s,t})
\end{aligned}$$

The last formula ψ_x is a DNF formula of $s \cdot t$ variables $(a_{1,1},\dots,a_{s,t})$, index by $(x_1,\dots,x_j)$. Hence, as a function of x, we have $\phi_{(\cdot)}(a_{1,1},\dots,a_{s,t}) \in DNF^*$.

The size and number of variables of the resulting DNF formula is $s' = s$ and $t' = s \cdot t$ respectively, which are both polynomial in s, t. Also, these mappings can be computed in polynomial time in s, t. Thus, $MONDNF$ can be PAC-reduced to DNF^*. □

Theorem 5.6. *$DNF \equiv_{pac} DNF^*$.*

Proof. By the above two lemmas, we have $DNF \leq_{pac} MONDNF \leq_{pac} DNF^*$. By the duality, we also have $DNF^* \leq_{pac} MONDNF^* \leq_{pac} DNF$ from the same mappings. Hence, $DNF \equiv_{pac} DNF^*$. □

Corollary 5.7. *DNF is PAC-learnable iff DNF is RRHS-refutable.*

6 Learning versus Refutation in the Noisy Regime

So far, we have seen that, in the realizable case, learning some function class is equivalent to refuting its dual class. The next natural question to ask is if an analogue result holds in the noisy setting, i.e. we are only guaranteed that the correlation between the concept class and the underlying concept is large. In fact, in [KL18], they independently discovered a similar connection between learning and refutation in the noisy case. In this survey, we will give some proof sketch of their theorem. To facilitate our discussion, we first present some definitions in the noisy case.

6.1 Definition

Definition 6.1 (Agnostic Learning w.r.t. Distribution $\mathcal{D}$)**.** *Let $\mathcal{P}$ be a class of Boolean functions $\mathcal{P} \subseteq \{f : \{-1,1\}^n \to \{-1,1\}\}$ and $\mathcal{D}$ be a distribution on $\{-1,1\}^n$. $\mathcal{P}$ is said to be ϵ-agnostic learnable w.r.t. distribution $\mathcal{D}$ in time $T = T(n,\epsilon)$ and samples $S = S(n,\epsilon)$ if there is an algorithm running in time T such that for every distribution $\mathcal{D}'$ on $\{-1,1\}^n \times \{-1,1\}$ whose marginal distribution on the input domain $\{-1,1\}^n$ is $\mathcal{D}$, given S samples $\{(x_i,y_i)\}_{i\le S}$ i.i.d. sampled from $\mathcal{D}'$, with prob. $\ge 2/3$, outputs a hypothesis $h : \{-1,1\}^n \to \{-1,1\}$ s.t.*

$$\mathbf{E}_{(x,y)\sim\mathcal{D}'}[\mathbf{1}_{h(x)\neq y}] \le \min_{f\in\mathcal{P}} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[\mathbf{1}_{f(x)\neq y}] + \epsilon.$$

Note that this definition of agnostic learnability differs from the standard one in that it only requires the algorithm to work on a given marginal distribution $\mathcal{D}$. In contrast, the standard definition requires the algorithm to work on every marginal distribution.

Definition 6.2 (Refutation Algorithm for Distribution $\mathcal{D}$)**.** *Let $\mathcal{P}$ be a class of Boolean functions $\mathcal{P} \subseteq \{f : \{-1,1\}^n \to \{-1,1\}\}$ and $\mathcal{D}$ be the distribution over $\{-1,1\}^n$. A δ-refutation algorithm for $\mathcal{P}$ on $\mathcal{D}$ with $m = m(n)$ samples is an algorithm $\mathcal{A}$ that takes m input-label pairs $\{(x_i,\sigma_i) \in \{-1,1\}^n \times \{-1,1\}\}_{i\le m}$ with the following guarantees:*

1. ***Satisfiable:*** *If $\{(x_i,y_i)\}_{i\le m}$ are i.i.d. sampled from a distribution $\mathcal{D}'$ on $\{0,1\}^n \times \{0,1\}$ whose marginal distribution on input domain $\{0,1\}^n$ is $\mathcal{D}$ and $\max_{f\in\mathcal{P}} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[f(x)y] \ge \delta$, then*
$$Pr_{\{(x_i,y_i)\}_{i\le m}\sim_{i.i.d.}\mathcal{D}',\mathcal{A}}[\mathcal{A}=1] \le 1/3.$$

2. ***Random:***
$$Pr_{\{x_i\}_{i\le m}\sim_{i.i.d.}\mathcal{D},\{y_i\}_{i\le m}\sim\mathcal{U}^m,\mathcal{A}}[\mathcal{A}=1] \ge 2/3.$$

Roughly speaking, the definition requires the algorithm to distinguish between the case that labels come from a true underlying distribution and the case that the labels are uniformly random.

6.2 Learning versus Refutation

We first show the direction from learning to refutation. The overall proof is similar to the one in realizable case. However, since in this case, the learning algorithm only works on specific distribution, we cannot hope to skew the distribution of the samples fed to the learning algorithm as we do in the realizable case.

Lemma 6.3 (Learning Implies Refutation in Agnostic Case)**.** *Suppose $\mathcal{P}$ is ϵ-agnostically learnable w.r.t. distribution $\mathcal{D}$ in time $T(n,\epsilon)$ and samples $S(n,\epsilon)$. Then, there is a δ-refutation algorithm $\mathcal{A}$ with respect to $\mathcal{D}$ running in time $T(n,\delta/4)$ with $2S(n,\delta/4)+128/\delta^2$ samples.*

Proof Sketch. Let $m = S(n,\delta/4)+64/\delta^2$. The δ-refutation algorithm $\mathcal{A}$ will run the ϵ-agnostically learning algorithm in a black-box way. Specifically, given $2m$ input-label pairs $(x_1,y_1),...,(x_{2m},y_{2m})$, $\mathcal{A}$ runs the ϵ-agnostically learning algorithm with $\epsilon = \delta/4$ on the first m input-label pairs and obtains a hypothesis h. Let $cor_h = \sum_{i=m+1}^{2m} h(x_i)y_i$. If $cor_h \ge \delta/2$, then output 0; otherwise output 1. Intuitively, since if the correlation between h and the remaining input-label pairs are large enough,

then it should imply that the given input-label pairs comes from a true distribution.
Suppose the samples come from some true distribution $\mathcal{D}'$. Let $cor_f(\mathcal{D}') = \mathbf{E}_{(x,y)\sim\mathcal{D}}[f(x)y]$ Then, by the definition of ϵ-agnostic learnable, with probability $\geq 2/3$ over the sample drawn and randomness of learning algorithm, $cor_h \geq cor_h(\mathcal{D}') - \epsilon \geq cor_f(\mathcal{D}') - 2\epsilon$ for every $f \in \mathcal{P}$. Thus, in this case, $\mathcal{A}$ will output 0 with high probability. On the other hand, if labels are drawn i.i.d. from uniform distribution, then since $\mathbf{E}_{x\sim\mathcal{D},y\sim\mathcal{U}}[h(x)y] = 0$, by Chernoff bound and that $m > 64/\delta^2$, with high probability, $cor_h \leq 4/\sqrt{m} < \delta/2$. □

Now we show the other direction. Specifically, we show the following lemma.

Lemma 6.4 (Learning from Refutation in Agnostic Case)**.** *Suppose there is a δ-refutation algorithm for some function class $\mathcal{P}$ w.r.t. distribution $\mathcal{D}$ running in time $T(n)$ and samples m. Then, there is an $(\delta + \epsilon)$-agnostically learning algorithm for $\mathcal{P}$ on $\mathcal{D}$ that runs in time $T(n)m^2/\epsilon^2$ and uses $O(m^3/\epsilon^2)$ samples.*

Similar to the realizable case, we will first construct a weak agnostic learner from the refutation algorithm and then use existing boosting theorem to get a full-fledged agnostic learner. We first give the definition of weak agnostic learner.

Definition 6.5 (Weak Agnostic Learner)**.** *We say an algorithm $\mathcal{A}$ is a (γ, α)-weak agnostic learner for some Boolean function class $\mathcal{P}$ w.r.t. a distribution $\mathcal{D}$ if $\mathcal{A}$ takes as samples input-label pairs $\{(x_i, y_i)\}$ i.i.d. drawn from some distribution $\mathcal{D}'$ whose marginal distribution on x is $\mathcal{D}$ and with probability $\geq 2/3$ outputs a hypothesis $h : \{-1,1\}^n \to \{-1,1\}$ s.t.*

$$\mathbf{E}_{(x,y)\sim\mathcal{D}'}[h(x)y] \geq \gamma(\max_{f\in\mathcal{P}} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[f(x)y]) - \alpha.$$

Now we are able to show how to construct a weak agnostic learner from a refutation algorithm. The proof strategy is also similar to the previous result. Specifically, suppose we have a refutation algorithm that can distinguish between the two cases with probability β.

a. All the samples are drawn from true distribution.

b. All the labels are drawn from uniform distribution (independent of input).

Then, considering all the intermediate cases from case a to case b, by an average argument, we see that we can distinguish between two neighboring intermediate cases with probability β/m where m is the number of samples. We will see in the following lemma that, using a similar argument, we can actually construct a hypothesis with large correlation.

Lemma 6.6 (Refutation to Weak Agnostic Learner)**.** *Suppose there is a δ-refutation algorithm for some Boolean function class $\mathcal{P}$ w.r.t. distribution $\mathcal{D}$ running in time $T(n)$ and samples m. Then, there is a $(2/(3m), (2\delta)/(3m))$-weak agnostic learner for $\mathcal{P}$ w.r.t. distribution $\mathcal{D}$ running in time $T(n)$ and samples m.*

Proof Sketch. Let $\mathcal{D}'$ be some distribution over $\{-1,1\}^n \times \{-1,1\}$ whose marginal distribution over $\{-1,1\}^n$ is $\mathcal{D}$. Since the case that $\max_{f\in\mathcal{P}} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[f(x)y] < \delta$ is trivial, we can assume

$$\max_{f\in\mathcal{P}} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[f(x)y] \geq \delta.$$

Thus, by our parameter setting, we only need to show that the hypothesis h outputted by the weak learner has the following property.

$$\mathbf{E}_{(x,y)\sim\mathcal{D}'}[h(x)y] \geq 1/(3m).$$

Note that now we can adapt lemma 4.2 here by replacing the RRHS-refuter with δ-refutation algorithm as its proof will hold for arbitrary true distribution. However, in Section 4, the PAC learner only need to predict the next example. Although two definitions are equivalent in some sense, to make the proof rigorous, we make the construction of hypothesis explicit here. For $b \in \{-1, 1\}$ and $0 \leq i \leq m+1$, we define $W_{i,b}(x)$ as follows:

1. Draw $(x_1, \sigma_1), ..., (x_{i-1}, \sigma_{i-1})$ i.i.d. from $\mathcal{D} \times \mathcal{U}_1$.
2. Draw $(x_{i+1}, y_{i+1}), ..., (x_m, y_m)$ i.i.d. from $\mathcal{D}'$.
3. Run δ-refutation algorithm on input
$$(x_1, \sigma_1), ..., (x_{i-1}, \sigma_{i-1}), (x, b), (x_{i+1}, y_{i+1}), ..., (x_m, y_m).$$
4. Let c be the output of δ-refutation algorithm. Output $\neg c$.

For $0 \leq i \leq m+1$, $h_i(x) = W_{i,1}(x) - W_{i,-1}(x)$. We will show that outputting $h_i(x)$ for a uniformly random i gives us a weak agnostic learner. Observe that $\mathbf{E}[W_{0,b}(x)] \geq 2/3$ and $\mathbf{E}[W_{m+1,b}(x)] \leq 1/3$ for any b. Thus,

$$\sum_{i=0}^{m+1} \mathbf{E}_{(x,y)\sim\mathcal{D}'}[W_{i,y}(x) - W_{i+1,y}(x)] \geq 1/3.$$

Now observe that by our construction,

$$W_{i,y}(x) = \frac{y+1}{2} W_{i,1}(x) - \frac{y-1}{2} W_{i,-1}(x) = \frac{1}{2} y h_i(x) + \frac{1}{2}(W_{i,1}(x) + W_{i,-1}(x)),$$

and also, $\mathbf{E}[\frac{1}{2}(W_{i,1}(x) + W_{i,-1}(x))] = \mathbf{E}[W_{i+1,y}(x)]$. Taking expectation on the both side, we get $\sum_{i=0}^{m+1} \mathbf{E}[y h_i(x)] \geq 2/3$. From this, we conclude that for a uniformly random i, h_i has the following correlation bound.

$$\mathbf{E}_{(x,y)\sim\mathcal{D}'}[h_i(x)y] \geq 2/(3m).$$

□

Now we can apply the existing boosting theorem [KK09] to get a full-fledged agnostic learner from the weak agnostic learner.

Theorem 6.7 (Agnostic Boosting)**.** *Let $\mathcal{P}$ be some class of Boolean functions, $\mathcal{D}$ be some distribution over $\{-1, 1\}^n$, and $\epsilon > 0$. If there is a (γ, α)-weak agnostic learner $\mathcal{A}$ for $\mathcal{P}$ w.r.t. $\mathcal{D}$, then there is an $(\alpha/\gamma + \epsilon)$-agnostic learner $\mathcal{A}'$ for $\mathcal{P}$ w.r.t. $\mathcal{D}$ that invokes $\mathcal{A}$ $O(1/(\gamma^2\epsilon^2))$ times. Specifically, $\mathcal{A}'$ takes $S(n)O(1/(\gamma^2\epsilon^2))$ samples and runs in time $T(n)O(1/(\gamma^2\epsilon^2))$ where $S(n)$ and $T(n)$ are the sample complexity and time complexity of $\mathcal{A}$.*

We finish this section by some remark. Note that both lemma 6.3 and lemma 6.6 will work even when the refutation and learning algorithms are distribution-independent. Thus, it is essentially the lack of distribution-independent agnostic boosting algorithm that makes this result distribution-dependent.

References

[DS16] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 815–830, 2016.

[KK09] Adam Kalai and Varun Kanade. Potential-based agnostic boosting. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 880–888, 2009.

[KL18] Pravesh K. Kothari and Roi Livni. Improper learning by refuting. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 55:1–55:10, 2018.

[KS09] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci.*, 75(1):2–12, 2009.

[Vad17] Salil P. Vadhan. On learning vs. refutation. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 1835–1848, 2017.