On Learning versus Refutation

Presenters: Kuan-Yi Ho, Kai-Chi Huang, Jiahui Liu

2018.10.30

Motivation

- We are interested in the hardness of PAC learning
- How do we analyze its hardness?
- We compare it with other hard problems
- Reduction: problem A is reducible to problem B if an algorithm for solving problem B could also be used as a subroutine to solve problem A.

Outline

Introduction: The results of Vadhan17

- Reduction from PAC learning to refutation based on pseudorandomness(reduction of the other direction was proved in [DLS14])
- Equivalence for PAC learning and RRHS-refuting for dual classes
- PAC learning of DNF is equivalent to RRHS-refuting (random right hand side refuting) of DNF
- with assumptions of hardness of DNF refutation, we could obtain stronger lower bounds for PAC learning

Introduction: Some History on Hardness of Learning

- Proper learning intersection of halfspace implies $P = NP$.
- [KS06] Assuming the hardness of certain lattice problem, improper learning intersection of halfspace is hard.
- [DSS14] Assuming hardness of refuting random k-SAT,
  - improper learning DNF is hard.
  - improper learning intersection of halfspace is hard.
- [Vadhan17] Equivalence between learning and refutation.
- [KL17] Equivalence between agnostic learning over some distribution $\mathcal{D}$ and refutation (with noise) with respect to $\mathcal{D}$.

PAC Learning

Definition (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, predicts the next example y correctly with prob $\geq 2/3$.

- This definition is equivalent to the standard definition

PAC Learning

Theorem (Schapire (1990) Boosting)

If $\mathcal{P}$ is weakly PAC learnable with sample complexity m and advantage α, then $\mathcal{P}$ is (ϵ, δ)-PAC learnable with sample complexity $m \cdot \mathrm{poly}(\frac{1}{\alpha}, \frac{1}{\epsilon}, \frac{1}{\delta})$.

- The definition of PAC learning here is the standard (ϵ, δ) definition within a polynomial factor.

Definition: Dual Class

- Given an evaluation function $\mathrm{Eval} : \{0,1\}^s \times \{0,1\}^t \to \{0,1\}$, we define
  - $\mathcal{P} = \{p_x : \{0,1\}^t \to \{0,1\} | x \in \{0,1\}^s\}$
  - $\mathcal{Q} = \{q_y : \{0,1\}^s \to \{0,1\} | y \in \{0,1\}^t\}$
  - $p_x(y) = q_y(x) = \mathrm{Eval}(x,y)$
- Think of $\mathcal{P}$ as the set of possible concepts, then x denotes the index of a certain concept.
- Think of $\mathcal{Q}$ as the set of possible assignments, then y is the assignment and $q_y(x) = 1$ iff y satisfies x.
- 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. (That's why they are dual!)
- Denote $\mathcal{P}^* = \mathcal{Q}$. That is, $\mathcal{Q}$ is the dual class of $\mathcal{P}$.

RRHS-Refuting(Random Right Hand Side refuting)

Definition (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 $y_1, \ldots, y_n \in \{0,1\}^t$ and every right hand side $b_1, \ldots, b_n \in \{0,1\}$, 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 prob $\leq 1/3$.
- **Random**: For every $y_1, \ldots, y_n \in \{0,1\}^t$, 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 prob $\geq 2/3$.

Don't need to consider unsatisfiable but not random labels.

RRHS-Refuting: Equivalent definition using dual class

Equivalently :

- $\mathcal{Q} = \mathcal{P}^*$ is RRHS-refutable using n equations if there is a polynomial-time algorithm $\mathcal{B}$ such that, for every $y_1, \ldots, y_n \in \{0,1\}^t$ and some labels $b_1, \ldots, b_n \in \{0,1\}$,
  - **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 prob $\leq 1/3$.
  - **Random**: If the labels b_i are randomly sampled i.i.d. from $\mathrm{Unif}(\{0,1\})$, then $\mathcal{B}$ must output 1 with prob $\geq 2/3$.
- That is, $\mathcal{B}$ can distinguish between **satisfiable** and **random** labels with advantage $1/3$. (There will be no partially true labels)

Equivalence between PAC-Learnable and RRHS-Refutable

Theorem

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 $\mathrm{poly}(n)$.*

PAC-Learnable Implies RRHS-Refutable

- How to use a PAC learner to check if a system of RRHS equations is **Satisfiable** or **Random**?
- Equivalently: How to use a PAC learner to check if the labels are **Satisfiable** or **Random**?
- Idea: 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).
- Problem: PAC learner requires the examples to be sampled i.i.d. from some distribution $\mathcal{D}$, we can't split the training set arbitrarily.

PAC-Learnable Implies RRHS-Refutable

Solution: Set $\mathcal{D}$ be uniform on all M examples. Then we follow the standard procedure of PAC learner: first i.i.d. sample m examples (y_i, b_i) 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, i.e. correct prob $\geq 2/3$.
- If **Random**: Unless (y', b') appears in the training samples, b' will be a new random label that $\mathcal{B}$ has never seen, so the prediction will be completely random, i.e. correct prob $= 1/2$. Overall correct rate $\leq 1/2 + \epsilon$.

Output: 0 if the prediction on the testing example is correct, 1 otherwise.

RRHS-Refutable Implies PAC-Learnable

- How to PAC learn a concept if we only have 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:
  1. Last label is correct: $(y_1, b_1), \ldots, (y_m, b_m), (y_{m+1}, b_{m+1}) - p_1$
  2. Last label is random: $(y_1, b_1), \ldots, (y_m, b_m), (y_{m+1}, c) - p_2$
  3. Last label is wrong: $(y_1, b_1), \ldots, (y_m, b_m), (y_{m+1}, \neg b_{m+1}) - p_3$
- $p_2 = \frac{1}{2}(p_1 + p_3)$
- $(p_3 - p_1) = 2(p_2 - p_1)$
- If we can distinguish Case 1 and Case 2 with advantage γ, then we can construct a weak PAC learner with advantage γ.
- What we need is essentially a **next bit predictor**.

Pseudorandomness vs Next-bit predictability

Lemma (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 $\frac{\gamma}{m}$.

Pseudorandomness vs Next-bit predictability

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)

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

- $p_m - p_0 \geq \gamma$
- $\exists i$ such that $p_i - p_{i-1} \geq \frac{\gamma}{m}$
- $\mathcal{B}$ is a weak i-th bit predictor given the previous $i-1$ bits!
- Problem: We don't know the value of i

Pseudorandomness vs Next-bit predictability

- **Randomly** choose $i \sim \mathrm{Unif}(\{1, \ldots, m\})$
- The expected advantage is
  $\frac{1}{m} \sum_{i=1}^{m} (p_i - p_{i-1}) = \frac{1}{m}(p_m - p_0) \geq \frac{\gamma}{m}$
- Examples don't have specific order
- $\mathcal{B}$ works for **all** combinations of examples.
- First choose i, then reorder the examples such that the unknown example is at position i

Result: Weak PAC learner with advantage $\frac{\gamma}{m}$
Again with Boosting, we obtain a PAC learner with sample complexity $\mathrm{poly}(m)$.

PAC Reduction

Definition

Let $\mathcal{P} = \mathcal{Q}^*$ and $\mathcal{P}' = (\mathcal{Q}')^*$ be two classes of boolean function class given by evaluation functions $\mathrm{Eval}(\cdot,\cdot)$ and $\mathrm{Eval}'(\cdot,\cdot)$ 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 $s', t' = \mathrm{poly}(s,t)$, such that for all s, t, $x \in \{0,1\}^s, y \in \{0,1\}^t$,

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

PAC Reduction

Proposition

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* $\mathcal{Q}$.

Theorem

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

Corollary

DNF is PAC-learnable iff DNF is RRHS-refutable.

Refuting Random k-SAT

Assumption (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 following two case.

- **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 $\geq 2/3$.*

[DSS16] PAC-learning DNF implies refuting random k-SAT.

RRHS-Refuting DNF Implies Refuting Random k-SAT

Proposition

If k-DNF 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

- Reduce the instance of random k-SAT to the instance of k-DNF formulas.
- In particular, we want to show
  - If k-SAT is satisfiable, then the constructed k-DNF is satisfiable.
  - If k-SAT is random, then the constructed k-DNF is random.

RRHS-Refuting DNF Implies Refuting Random k-SAT

Proof sketch:

- By De Morgan's law: refuting k-DNF is equivalent to refuting k-CNF
- Given $n \cdot m$ number of k-way disjunctions $\phi_1, ..., \phi_{nm}$, for $i = 1, ..., n$, construct:
  - with prob $\frac{1}{2}$, 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 prob $\frac{1}{2}$, 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 refuter

RRHS-Refuting DNF Implies Refuting Random k-SAT

Proof sketch:

- Satisfiable case
  - If $\phi_1 = 1, ..., \phi_{nm} = 1$ are satisfiable by some assignment α, then $\psi_1 = b_1, ..., \psi_n = b_n$ is satisfiable by α w.h.p.
  - The case that $b_i = 1$ is clear.
  - In the other case, we can show that given any assignment α, α cannot satisfy a random ψ_i w.h.p.
- Random case
  - Since the distribution of ψ_i is the same for the case that $b_i = 0$ and $b_i = 1$ and happen with the same probability, b_i's are uniformly random and independent of ψ_i's.

Summary of Vadhan17 (Realizable Case)

- Equivalence between PAC-learning function class $\mathcal{P}$ and RRHS-refuting its dual class.
  - This holds for any distribution.
- DNF is PAC-learnable iff DNF is RRHS-refutable.
- RRHS-refuting DNF implies refuting random k-SAT.
  - The assumption used is weaker than the assumption in [DSS16].

We will compare it to a similar result in the agnostic setting.

Summary of KL17 (Agnostic Case)

- Also shows the equivalence between learnability and refutability but in the agnostic case.
- All the ingredients are essentially the same.
  - Similar argument from learnability to refutability.
  - The other side also uses similar hybrid argument (used in the proof of Yao's lemma) and boosting.
- The learnability and refutability are defined with respect to specific distribution.

We will mainly discuss the difference of definitions used between these work.

Refutation in Realizable Case

Recall the definition of RRHS-refutable.

Definition (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 $y_1, \ldots, y_n \in \{0,1\}^t$ and every right hand side $b_1, \ldots, b_n \in \{0,1\}$, 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 prob $\leq 1/3$.
- **Random**: For every $y_1, \ldots, y_n \in \{0,1\}^t$, 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 prob $\geq 2/3$.

Refutation in Agnostic Case

Definition (Refutable in Agnostic Case)

$\mathcal{Q}$ is δ-refutable using n equations with respect to a distribution $\mathcal{D}$ if there is a polynomial-time algorithm $\mathcal{B}$ that can distinguish the following two cases,

- **Satisfiable**: If $(y_i, b_i) \in \{0,1\}^t \times \{-1,1\}$ is i.i.d. sampled from a distribution $\mathcal{D}'$ whose marginal distribution on y_i is $\mathcal{D}$ and $\max_x E_{y,b}[q_y(x)b] \geq \delta$, then $\mathcal{B}$ must output 1 with prob $\leq 1/3$.
- **Random**: If each y_i is i.i.d. sampled from $\mathcal{D}$ and the right hand sides b_i are randomly sampled i.i.d. from $\mathrm{Unif}(\{0,1\})$, then $\mathcal{B}$ must output 1 with prob $\geq 2/3$.

Boosting in Agnostic Case

Theorem (Agnostic Boosting)

If $\mathcal{P}$ is weakly agnostic learnable over some distribution $\mathcal{D}$, then $\mathcal{P}$ is agnostic PAC learnable over $\mathcal{D}$.

- The existing boosting algorithm in agnostic learning is distribution-dependent.
- This makes the equivalence result in [KL17] distribution-dependent.

References

- Cryptographic Hardness for Learning Intersections of Halfspaces by *Adam Klivans and Alexander Sherstov*
- Complexity theoretic limitations on learning DNF's by *Amit Daniely and Shai Shalev-Shwatz*
- On Learning versus Refutation by *Salil Vadhan*
- Agnostic Learning by Refuting by *Pravesh Kothari and Roi Livni*