Is Efficient PAC Learning Possible with an Oracle That Responds "Yes" or "No"?

Final Presentation: S&DS 669

Anish Lakkapragada[1] Zimeng[2]

[1]Department of Statistics
Yale University

Table of Contents

Table of Contents

Empirical Risk Minimization (ERM)

ERM is great and has led in practice to very good results. But:

- Can require a too computationally expensive *oracle* to perform ERM

Empirical Risk Minimization (ERM)

ERM is great and has led in practice to very good results. But:

- Can require a too computationally expensive *oracle* to perform ERM

Empirical Risk Minimization (ERM)

ERM is great and has led in practice to very good results. But:

- Can require a too computationally expensive *oracle* to perform ERM
- Can we find a *weaker* oracle to still efficiently PAC-learn?

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \text{vc}(\mathcal{H}) < \infty$.

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \text{vc}(\mathcal{H}) < \infty$.
- Assume S realizable by $\mathcal{H}$.

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \mathrm{vc}(\mathcal{H}) < \infty$.
- Assume S realizable by $\mathcal{H}$.
- We can create an oracle $\mathcal{O}^{\text{con, w}}$ that returns YES/NO if S is $\mathcal{H}$-realizable

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \mathrm{vc}(\mathcal{H}) < \infty$.
- Assume S realizable by $\mathcal{H}$.
- We can create an oracle $\mathcal{O}^{\text{con, w}}$ that returns YES/NO if S is $\mathcal{H}$-realizable
- We call $\mathcal{O}^{\text{con, w}}$ a *weak consistency oracle*

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \text{vc}(\mathcal{H}) < \infty$.
- Assume S realizable by $\mathcal{H}$.
- We can create an oracle $\mathcal{O}^{\text{con, w}}$ that returns YES/NO if S is $\mathcal{H}$-realizable
- We call $\mathcal{O}^{\text{con, w}}$ a *weak consistency oracle*
- Then $\mathcal{H}$ can be efficiently-PAC-learned with an algorithm that employs $O(\text{poly}(n))$ calls to $\mathcal{O}^{\text{con, w}}$

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \text{vc}(\mathcal{H}) < \infty$.
- Assume S realizable by $\mathcal{H}$.
- We can create an oracle $\mathcal{O}^{\text{con, w}}$ that returns YES/NO if S is $\mathcal{H}$-realizable
- We call $\mathcal{O}^{\text{con, w}}$ a *weak consistency oracle*
- Then $\mathcal{H}$ can be efficiently-PAC-learned with an algorithm that employs $O(\text{poly}(n))$ calls to $\mathcal{O}^{\text{con, w}}$
- Moreover, sample complexity scales as $\tilde{O}(d^3 \cdot \frac{\log(1/\delta)}{\epsilon})$

A *single bit* oracle for efficient learning

The answer is yes!

- Define $S = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}$ with $d := \text{vc}(\mathcal{H}) < \infty$.
- Assume S realizable by $\mathcal{H}$.
- We can create an oracle $\mathcal{O}^{\text{con, w}}$ that returns YES/NO if S is $\mathcal{H}$-realizable
- We call $\mathcal{O}^{\text{con, w}}$ a *weak consistency oracle*
- Then $\mathcal{H}$ can be efficiently-PAC-learned with an algorithm that employs $O(\text{poly}(n))$ calls to $\mathcal{O}^{\text{con, w}}$
- Moreover, sample complexity scales as $\tilde{O}(d^3 \cdot \frac{\log(1/\delta)}{\epsilon})$
- Similar oracle extension and learning guarantees exist for agnostic setting & regression.

Safety implications beyond efficient learning

Given the rise of cheap query APIs to models, a lot of people are researching:

- How to reconstruct a model from little information

Safety implications beyond efficient learning

Given the rise of cheap query APIs to models, a lot of people are researching:

- How to reconstruct a model from little information
- How to reconstruct the training dataset from little information

Safety implications beyond efficient learning

Given the rise of cheap query APIs to models, a lot of people are researching:

- How to reconstruct a model from little information
- How to reconstruct the training dataset from little information
- Bottom Line: Weak Oracles enable learning (& attacks)!

Table of Contents

Preliminaries: Partial Binary Concept Classes

Consider a domain $\mathcal{X}$ and a concept class $H \subseteq \{0, 1, *\}^{\mathcal{X}}$

Key components:

- Each hypothesis $h \in H$ maps inputs to $\{0, 1, *\}$
- The symbol $*$ means the hypothesis is undefined at that point
- Special case: Total binary class when no hypothesis outputs $*$

Binary loss function:

$$\ell_{\text{bin}}(y, y') = \mathbf{1}\{y \neq y' \vee y = * \vee y' = *\}$$

A sample $S = \{(x_i, y_i)\}_{i \in [n]}$ is $\mathcal{H}$-realizable if:

$$\exists h \in H \text{ such that } h(x_i) = y_i \neq * \text{ for all } i$$

Weak Consistency Oracle

Definition: A weak consistency oracle $O_{\text{con,w}}$ for class H

Input: A sample $S = \{(x_i, y_i)\}_{i \in [n]} \subseteq (\mathcal{X} \times \{0, 1\})^n$

Output:

- True if S is $\mathcal{H}$-realizable
- False otherwise

Key property: Returns only 1 bit of information

This is a decision problem, not a search problem

Much weaker than a standard oracle that returns an actual hypothesis

Weak ERM Oracle

Definition: A weak ERM oracle $O_{\text{erm,w}}$ for class H

Input: A sample $S = \{(x_i, y_i)\}_{i \in [n]} \subseteq (\mathcal{X} \times \{0,1\})^n$

Output: The value

$$\min_{h \in H} \widehat{\text{er}}_S(h) \in \{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\}$$

where

$$\widehat{\text{er}}_S(h) = \frac{1}{n} \sum_{(x,y) \in S} \mathbf{1}\{h(x) \neq y\}$$

Returns only the minimum empirical risk value, not which hypothesis achieving it. Slightly stronger than weak consistency oracle.

Used for agnostic learning when data may not be realizable

Table of Contents

Oracle-Efficient PAC Learning Main Result (Theorem 3.1)

- Consider $\mathcal{H}$ with $d_{\mathsf{VC}}(\mathcal{H})$ and and weak consistency oracle $\mathcal{O}^{\mathrm{con},w}$

Oracle-Efficient PAC Learning Main Result (Theorem 3.1)

- Consider $\mathcal{H}$ with $d_{\mathsf{VC}}(\mathcal{H})$ and and weak consistency oracle $\mathcal{O}^{\mathrm{con},w}$
- The class $\mathcal{H}$ is $(\mathcal{O}^{\mathrm{con},w}; \epsilon, \delta)$-PAC learnable by with sample complexity $n = \tilde{O}(d_{\mathrm{VC}}^3 \log(1/\delta)/\epsilon)$ and oracle complexity $\mathrm{poly}(n)$.

Oracle-Efficient PAC Learning Main Result (Theorem 3.1)

- Consider $\mathcal{H}$ with $d_{\mathsf{VC}}(\mathcal{H})$ and and weak consistency oracle $\mathcal{O}^{\mathrm{con},w}$
- The class $\mathcal{H}$ is $(\mathcal{O}^{\mathrm{con},w}; \epsilon, \delta)$-PAC learnable by with sample complexity $n = \tilde{O}(d_{\mathrm{VC}}^3 \log(1/\delta)/\epsilon)$ and oracle complexity $\mathrm{poly}(n)$.
- We will actually show this by boosting a weak-learner.

Oracle-Efficient PAC Learning Main Result (Theorem 3.1)

- Consider $\mathcal{H}$ with $d_{\mathsf{VC}}(\mathcal{H})$ and and weak consistency oracle $\mathcal{O}^{\mathrm{con},w}$
- The class $\mathcal{H}$ is $(\mathcal{O}^{\mathrm{con},w}; \epsilon, \delta)$-PAC learnable by with sample complexity $n = \tilde{O}(d_{\mathrm{VC}}^3 \log(1/\delta)/\epsilon)$ and oracle complexity $\mathrm{poly}(n)$.
- We will actually show this by boosting a weak-learner.
- Similarly in the agnostic setting, there is an algorithm Alg^A such that For any class $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ satisfying and weak ERM oracle $\mathcal{O}^{\mathrm{erm},w}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathrm{erm},w}; \epsilon, \delta)$-PAC learnable by Alg^A with sample complexity $n = \tilde{O}(d_{\mathrm{VC}}^3 \log(1/\delta)/\epsilon^2)$ and oracle complexity $\mathrm{poly}(n)$.

How will we prove the realizable setting?

- We are going to create an algorithm $\mathrm{WeakRealizable}$ to weak-learn $\mathcal{H}$

How will we prove the realizable setting?

- We are going to create an algorithm $\mathrm{WeakRealizable}$ to weak-learn $\mathcal{H}$
- $\mathrm{WeakRealizable}$ will use polynomially many calls to $\mathcal{O}^{\text{con, w}}$

How will we prove the realizable setting?

- We are going to create an algorithm $\mathrm{WeakRealizable}$ to weak-learn $\mathcal{H}$
- $\mathrm{WeakRealizable}$ will use polynomially many calls to $\mathcal{O}^{\text{con, w}}$
- We will then boost this learner to to achieve (ϵ, δ) error-confidence

Expected LOO Mistake Bound Guarantee of WeakRealizable Algorithm (Theorem 3.2)

Recall the LOO Mistake Bound's control of the population error.

Expected LOO Mistake Bound Guarantee of WeakRealizable Algorithm (Theorem 3.2)

Recall the LOO Mistake Bound's control of the population error.

- Let $\mathcal{H}$ have VC dimension d, let $\delta \in (0, 1)$, and suppose $m \geq C_1 d \log d$.

Expected LOO Mistake Bound Guarantee of $\mathrm{WeakRealizable}$ Algorithm (Theorem 3.2)

Recall the LOO Mistake Bound's control of the population error.

- Let $\mathcal{H}$ have VC dimension d, let $\delta \in (0, 1)$, and suppose $m \geq C_1 d \log d$.
- For an $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0, 1\})^{m-1}$ and $x \in \mathcal{X}$, let $\mathcal{A}(S, x) \in \{0, 1\}$ be the output of

  $$\mathrm{WeakRealizable}(S, x, \ldots, \mathcal{O}^{\mathrm{con},w}),$$

  which is a random variable.

Expected LOO Mistake Bound Guarantee of $\mathrm{WeakRealizable}$ Algorithm (Theorem 3.2)

Recall the LOO Mistake Bound's control of the population error.

- Let $\mathcal{H}$ have VC dimension d, let $\delta \in (0,1)$, and suppose $m \geq C_1 d \log d$.
- For an $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0,1\})^{m-1}$ and $x \in \mathcal{X}$, let $\mathcal{A}(S,x) \in \{0,1\}$ be the output of

$$\mathrm{WeakRealizable}(S, x, \ldots, \mathcal{O}^{\mathrm{con},w}),$$

  which is a random variable.
- Then for any $\mathcal{H}$-realizable sample $S = \{(x_i, y_i)\}_{i\in[m]} \in (\mathcal{X} \times \{0,1\})^m$,

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}[\ell^{\mathrm{bin}}(\mathcal{A}(S_{-i}, x_i), y_i)] \leq \frac{1}{2} - \frac{1}{C_2 m \log m},$$

  where the expectation is taken over randomness in $\mathcal{A}$.

Expected LOO Mistake Bound Guarantee of $\mathrm{WeakRealizable}$ Algorithm (Theorem 3.2)

Recall the LOO Mistake Bound's control of the population error.

- Let $\mathcal{H}$ have VC dimension d, let $\delta \in (0,1)$, and suppose $m \geq C_1 d \log d$.
- For an $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0,1\})^{m-1}$ and $x \in \mathcal{X}$, let $\mathcal{A}(S,x) \in \{0,1\}$ be the output of

$$\mathrm{WeakRealizable}(S, x, \ldots, \mathcal{O}^{\mathrm{con},w}),$$

  which is a random variable.
- Then for any $\mathcal{H}$-realizable sample $S = \{(x_i, y_i)\}_{i\in[m]} \in (\mathcal{X} \times \{0,1\})^m$,

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}[\ell^{\mathrm{bin}}(\mathcal{A}(S_{-i}, x_i), y_i)] \leq \frac{1}{2} - \frac{1}{C_2 m \log m},$$

  where the expectation is taken over randomness in $\mathcal{A}$.
- $\mathrm{WeakRealizable}$ makes $\tilde{O}(m^3)$ calls to $\mathcal{O}^{\mathrm{con},w}$ of $m-1$ size datasets.

WeakRealizable Algorithm: Overview

Algorithm 1: WeakRealizable$(S, x, \gamma, \lambda, U, O_{\text{con,w}})$

Inputs:

- $\mathcal{H}$-realizable sample $S = \{(x_i, y_i)\}_{i \in [m-1]}$
- Query point $x \in \mathcal{X}$
- Parameters: $\gamma, \lambda \in (0, 1)$, $U \in \mathbb{N}$
- Consistency oracle $O_{\text{con,w}}$

Goal: Predict the label for x using only the weak oracle

Key idea:

- Consider both possible labels for x: 0 and 1
- Estimate an opposite potential function for each possibility
- Make a randomized prediction based on these potentials

WeakRealizable Algorithm: Main Steps

Step 1: Construct sequence $X \leftarrow (x_1, \ldots, x_{m-1}, x)$
Step 2: Create candidate labelings

$$y^0 \leftarrow (y_1, \ldots, y_{m-1}, 0), \quad y^1 \leftarrow (y_1, \ldots, y_{m-1}, 1)$$

Step 3: Check realizability with only 2 oracle calls. Most important shortcut when it's binary $\rightarrow$ immediately know correct label.

- If $O_{\text{con,w}}(\{(X_j, y_j^b)\}_{j\in[m]}) = \text{False}$ for some $b \in \{0, 1\}$
- Return $1 - b$ (the other label must be correct)

Step 4: Estimate potentials, higher value $\rightarrow$ less likely label (farther)

$$\widehat{F}(y^0) \leftarrow \text{EstimatePotential}(X, y^0, U, \gamma, O_{\text{con,w}})$$
$$\widehat{F}(y^1) \leftarrow \text{EstimatePotential}(X, y^1, U, \gamma, O_{\text{con,w}})$$

Step 5: Return random prediction from $\text{Ber}(\widehat{\sigma})$ where

$$\widehat{\sigma} = \frac{1 + \lambda \cdot (\widehat{F}(y^0) - \widehat{F}(y^1))}{2}$$

If $\widehat{F}(y^0) > \widehat{F}(y^1)$, then $\hat{\sigma} > \frac{1}{2}$, gives a greater chance to predict 1.

EstimatePotential: Random Walk Subroutine

Function: EstimatePotential$(X, y, U, \gamma, O_{\text{con,w}})$, it estimates a "potential" for a vertex in the one-inclusion graph by simulating random walks

For each trial $u = 1$ to U:

Initialize: $Y^{(0)} \leftarrow y$

For step $t = 0$ to the fast $T_{\max} = \lceil \log(32e/(1-\gamma))/\log(1/\gamma) \rceil$:

Check: Is $O_{\text{con,w}}(\{(X_j, Y_j^{(t)})\}_{j \in [m]}) = \text{False}$?

If yes: Set $T_u \leftarrow t$ and stop this trial

If no: Take random step

Choose $i \sim \text{Unif}([m])$

Flip coordinate i: $Y^{(t+1)} \leftarrow (Y^{(t)})^{\oplus i}$

Return: Average over trials

$$\frac{1}{U} \sum_{u=1}^{U} \gamma^{T_u}$$

The *Alg* performs random walks on the hypercube, checking at each step whether the current vertex is in $H|_X$ (realizable) or its complement.

Table of Contents

WeakRealizable Random Orientations Yield OIG Out-Degree Bound

- Consider some edge (y^0, y^1) in the OIG

WeakRealizable Random Orientations Yield OIG Out-Degree Bound

- Consider some edge (y^0, y^1) in the OIG
- $\hat{\sigma}$ and $1 - \hat{\sigma}$ give probability mass on y^1 and y^0 respectively

WeakRealizable Random Orientations Yield OIG Out-Degree Bound

- Consider some edge (y^0, y^1) in the OIG
- $\hat{\sigma}$ and $1 - \hat{\sigma}$ give probability mass on y^1 and y^0 respectively
- Define y to be the "correct" vertex in $\{y^0, y^1\}$.

$\mathrm{WeakRealizable}$ Random Orientations Yield OIG Out-Degree Bound

- Consider some edge (y^0, y^1) in the OIG
- $\hat{\sigma}$ and $1 - \hat{\sigma}$ give probability mass on y^1 and y^0 respectively
- Define y to be the "correct" vertex in $\{y^0, y^1\}$.
- $\mathrm{WeakRealizable}$ causes out-degree of y to be $\leq m[\frac{1}{2} - \Omega(\frac{1}{m \log m})]$

$\mathrm{WeakRealizable}$ Random Orientations Yield OIG Out-Degree Bound

- Consider some edge (y^0, y^1) in the OIG
- $\hat{\sigma}$ and $1 - \hat{\sigma}$ give probability mass on y^1 and y^0 respectively
- Define y to be the "correct" vertex in $\{y^0, y^1\}$.
- $\mathrm{WeakRealizable}$ causes out-degree of y to be $\leq m[\frac{1}{2} - \Omega(\frac{1}{m \log m})]$
- We will prove the above statement soon

$\mathrm{WeakRealizable}$ Random Orientations Yield OIG Out-Degree Bound

- Consider some edge (y^0, y^1) in the OIG
- $\hat{\sigma}$ and $1 - \hat{\sigma}$ give probability mass on y^1 and y^0 respectively
- Define y to be the "correct" vertex in $\{y^0, y^1\}$.
- $\mathrm{WeakRealizable}$ causes out-degree of y to be $\leq m[\frac{1}{2} - \Omega(\frac{1}{m \log m})]$
- We will prove the above statement soon
- Finally recall from lecture:

$$\max_{S \in \mathrm{Re}_{\mathcal{H}}(m)} \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{Q_O(S_{-i}, x_i) \neq y_i\} = \max_{v \in V_{\mathrm{OIG}}} \frac{\mathrm{outdeg}(v; O)}{m}$$

How do we create such an Out-Degree Bound?

- Long proof. Define $X = (x_1, \ldots, x_m)$

How do we create such an Out-Degree Bound?

- Long proof. Define $X = (x_1, \ldots, x_m)$
- Consider graph $G_m = (V_m = \{0,1\}^m, E_m)$ where OIG $G(\mathcal{H}\,|_X) \subset G_m$

How do we create such an Out-Degree Bound?

- Long proof. Define $X = (x_1, \ldots, x_m)$
- Consider graph $G_m = (V_m = \{0,1\}^m, E_m)$ where OIG $G(\mathcal{H}\,|_X) \subset G_m$
- Choose $v \in V_m$ with m edges and define $Z_v^{(0)} = v$

How do we create such an Out-Degree Bound?

- Long proof. Define $X = (x_1, \ldots, x_m)$
- Consider graph $G_m = (V_m = \{0,1\}^m, E_m)$ where OIG $G(\mathcal{H}\mid_X) \subset G_m$
- Choose $v \in V_m$ with m edges and define $Z_v^{(0)} = v$
- Flip coin.
  - If heads, $Z_v^{(1)} = v$.
  - If tails, choose $Z_v^{(1)}$ to be one of the vertices connected via edge.

How do we create such an Out-Degree Bound?

- Long proof. Define $X = (x_1, \ldots, x_m)$
- Consider graph $G_m = (V_m = \{0,1\}^m, E_m)$ where OIG $G(\mathcal{H}\,|_X) \subset G_m$
- Choose $v \in V_m$ with m edges and define $Z_v^{(0)} = v$
- Flip coin.
  - If heads, $Z_v^{(1)} = v$.
  - If tails, choose $Z_v^{(1)}$ to be one of the vertices connected via edge.
- Proceed $Z_v^{(t)}$ in this way and define hitting time of $\mathcal{S} \subset V_m$ as:

$$\tau_{S,v} = \min\{t \geq 0 : Z_v^{(t)} \in \mathcal{S}\}$$

How do we create such an Out-Degree Bound?

- Long proof. Define $X = (x_1, \ldots, x_m)$
- Consider graph $G_m = (V_m = \{0,1\}^m, E_m)$ where OIG $G(\mathcal{H}\,|_X) \subset G_m$
- Choose $v \in V_m$ with m edges and define $Z_v^{(0)} = v$
- Flip coin.
  - If heads, $Z_v^{(1)} = v$.
  - If tails, choose $Z_v^{(1)}$ to be one of the vertices connected via edge.
- Proceed $Z_v^{(t)}$ in this way and define hitting time of $\mathcal{S} \subset V_m$ as:

$$\tau_{S,v} = \min\{t \geq 0 : Z_v^{(t)} \in \mathcal{S}\}$$

- Define *generating function* $M_{\mathcal{S},v}(\gamma) : (0,1) \to \mathbb{R}$ as:

$$M_{\mathcal{S},v}(\gamma) = \mathbb{E}[\gamma^{\tau_{\mathcal{S},v}}]$$

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H}\,|_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H} \mid_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

- Consider $\sigma_{F,\lambda=1} = \frac{1+(F(y^0)-F(y^1))}{2}$ to be an orientation

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H}\,|_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

- Consider $\sigma_{F,\lambda=1} = \frac{1+(F(y^0)-F(y^1))}{2}$ to be an orientation
- Using a graph-theory result, they show:

$$\max_{v \in V_{\mathrm{OIG}}} \mathrm{outdeg}(v; \sigma_{F,\lambda=1}) \le \frac{m}{2} - (1-\gamma)m \cdot \min_{v \in V_m} F(v)$$

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H}\mid_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

- Consider $\sigma_{F,\lambda=1} = \frac{1+(F(y^0)-F(y^1))}{2}$ to be an orientation
- Using a graph-theory result, they show:

$$\max_{v \in V_{\text{OIG}}} \text{outdeg}(v; \sigma_{F,\lambda=1}) \leq \frac{m}{2} - (1-\gamma)m \cdot \min_{v \in V_m} F(v)$$

- Observe $|\mathcal{S}^c| \leq (em)^d$ by Sauer-Shelah-Perles Lemma

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H}\mid_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

- Consider $\sigma_{F,\lambda=1} = \frac{1+(F(y^0)-F(y^1))}{2}$ to be an orientation
- Using a graph-theory result, they show:

$$\max_{v \in V_{\mathrm{OIG}}} \mathrm{outdeg}(v; \sigma_{F,\lambda=1}) \leq \frac{m}{2} - (1-\gamma)m \cdot \min_{v \in V_m} F(v)$$

- Observe $|\mathcal{S}^c| \leq (em)^d$ by Sauer-Shelah-Perles Lemma
- Using $m \geq \Omega(d \log d)$ makes $\frac{|\mathcal{S}^c|}{|V_m|} \downarrow \implies$ hitting time $\tau_{S,v} \uparrow$

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H}\,|_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

- Consider $\sigma_{F,\lambda=1} = \frac{1+(F(y^0)-F(y^1))}{2}$ to be an orientation
- Using a graph-theory result, they show:

$$\max_{v \in V_{\text{OIG}}} \text{outdeg}(v; \sigma_{F,\lambda=1}) \le \frac{m}{2} - (1-\gamma)m \cdot \min_{v \in V_m} F(v)$$

- Observe $|\mathcal{S}^c| \le (em)^d$ by Sauer-Shelah-Perles Lemma
- Using $m \ge \Omega(d \log d)$ makes $\frac{|\mathcal{S}^c|}{|V_m|} \downarrow \implies$ hitting time $\tau_{S,v} \uparrow$
- This causes $F(v)$ to be lower-bounded: $\min_{v \in V_m} F(v) \ge \Omega(1)$

How do we create such an Out-Degree Bound (Cont.)?

- Now define the following:

$$\mathcal{S} = (\mathcal{H}\,|_X)^c, \quad \gamma := 1 - \Theta(\frac{1}{m \log m}), \quad F(v) = M_{\mathcal{S},v}(\gamma)$$

- Consider $\sigma_{F,\lambda=1} = \frac{1+(F(y^0)-F(y^1))}{2}$ to be an orientation
- Using a graph-theory result, they show:

$$\max_{v \in V_{\text{OIG}}} \text{outdeg}(v; \sigma_{F,\lambda=1}) \leq \frac{m}{2} - (1-\gamma)m \cdot \min_{v \in V_m} F(v)$$

- Observe $|\mathcal{S}^c| \leq (em)^d$ by Sauer-Shelah-Perles Lemma
- Using $m \geq \Omega(d \log d)$ makes $\frac{|\mathcal{S}^c|}{|V_m|} \downarrow \implies$ hitting time $\tau_{S,v} \uparrow$
- This causes $F(v)$ to be lower-bounded: $\min_{v \in V_m} F(v) \geq \Omega(1)$
- So altogether this yields:

$$\max_{v \in V_{\text{OIG}}} \text{outdeg}(v; \sigma_{F,\lambda=1}) \leq m \cdot \left(\frac{1}{2} - \Omega(\frac{1}{m \log m})\right)$$

Tying all loose ends

- Observe that $F(v)$ is estimated by $\mathrm{EstimatePotential}$

Tying all loose ends

- Observe that $F(v)$ is estimated by $\mathrm{EstimatePotential}$
- Brushing aside technicalities, this yields the expected LOO Bound on $\mathrm{WeakRealizable}$

Tying all loose ends

- Observe that $F(v)$ is estimated by $\mathrm{EstimatePotential}$
- Brushing aside technicalities, this yields the expected LOO Bound on $\mathrm{WeakRealizable}$
- From the LOO Bound, we can control the population error (i.e. $< \frac{1}{2}$)

Tying all loose ends

- Observe that $F(v)$ is estimated by $\mathrm{EstimatePotential}$
- Brushing aside technicalities, this yields the expected LOO Bound on $\mathrm{WeakRealizable}$
- From the LOO Bound, we can control the population error (i.e. $< \frac{1}{2}$)
- We then use AdaBoost on $\mathrm{WeakRealizable}$ weak learner to achieve arbitrarily low error w.h.p

Table of Contents

Review: What We Learned

Main result: Efficient PAC learning with weak oracles is possible

Key components:

- Partial binary concept classes with VC dimension d_{VC}
- Weak consistency oracle: Returns only 1 bit (yes/no)
- Sample complexity: $\widetilde{O}(d_{\mathrm{VC}}^3)$ samples needed
- Oracle complexity: poly(n) calls to the weak oracle
- Algorithm: Random walks on one-inclusion graph
- Achieves weak learning: Error $\leq 1/2 - \Omega(1/(m \log m))$
- Boosting: Amplify weak learner to achieve arbitrary accuracy

Price: Factor of d_{VC}^2 increase compared to optimal $O(d_{\mathrm{VC}})$