Notes 11: Proper vs Improper Learning

Proper learning: Algorithm required to output $h \in \mathcal{C}$, i.e. $H = \mathcal{C}$

Improper learning: Algorithm allowed to output $h / \in \mathcal{C}$, i.e. $H \supseteq \mathcal{C}$

(Below) When $\mathcal{C} = \{3\text{-term DNF}\}$ over $X = \{0, 1\}^n$

Can efficiently PAC-learn $\mathcal{C}$ with improper algorithm

No efficient algorithm can properly PAC-learn $\mathcal{C}$ (under standard complexity assumption)

By contrast, 1-term DNF (= disjunctions) can be efficiently PAC-learned properly using Consistent Hypothesis Algorithm:

$$\varepsilon \left( O(n) + \ln \frac{1}{\delta} \right)$$

samples

1. 3-TERM DNF VS 3-CNF

Every 3-term DNF is 3-CNF

3-term DNF $f(x) = T_1 \lor T_2 \lor T_3$ where $T_i$ are conjunctions

Since $\lor$ distributes over $\land$, i.e. $(u \land v) \lor (x \land y) = (u \lor x) \land (u \lor y) \land (v \lor x) \land (v \lor y)$

$$f(x) = T_1 \lor T_2 \lor T_3 = \bigwedge \text{literals } x \text{ in } T_1, y \text{ in } T_2, z \text{ in } T_3$$

There is efficient improper PAC learning algorithm when $\mathcal{C} \subseteq H = \{3\text{-CNF}\}$

Consistent Hypothesis Algorithm based on Elimination

$|H| = 2^{|V|^2} = 2^{O(n^2)} \implies \frac{1}{\varepsilon} \left( O(n^3) + \ln \frac{1}{\delta} \right)$

samples

2. GRAPH 3-COLORING

Theorem 2.1. If some efficient algorithm $A$ properly PAC-learns 3-term DNF, then some efficient randomized algorithm $B$ solves Graph-3-Coloring (and violates standard complexity assumption)

Graph-3-Coloring problem

Input: $n$-vertex undirected graph $G$

Goal: Decides if vertices of $G$ can be colored using 3 colors so that no edge has both endpoints with the same color

Graph-3-Coloring is NP-complete

widely believed not solvable in polynomial time: current fastest algorithm takes $2^{\Theta(n)}$ time

In the theorem, efficient randomized algorithm $B$ for Graph-3-Coloring on graph $G$

(1) always runs in $\text{poly}(n)$ time

(2) If $G$ is not 3-colorable, $B$ always says No

(3) If $G$ is 3-colorable, $B$ says Yes with probability $\geq 1/2$ (can be boosted to $\geq 1 - 2^{-n}$)

Standard complexity assumption is NP $\neq$ RP

The theorem is proved via reduction from Graph-3-Coloring to proper PAC-learning of 3-term DNF

An algorithm $R$ that maps $n$-vertex graph $G$ to set $S = S^+ \cup S^-$ of labelled examples over $\{0, 1\}^n$
s.t. $G$ has 3-coloring $\iff (S^+, S^-)$ is consistent with some 3-term DNF

$R$ runs in $\text{poly}(n)$ time (in particular $|S| \leq \text{poly}(n)$)

Labelled samples $(S^+, S^-)$ from $R$ corresponds to PAC-learning task with parameters

$\varepsilon = 1/(2|S|)$  $\delta = 1/2$  $\mathcal{D} = \text{uniform distribution over } S$

Suppose some algorithm $A$ solves proper PAC-learning of 3-term DNF
Randomized algorithm $B$ to solve Graph-3-Coloring on graph $G$

Run reduction $R$ on $G$ to get labelled samples $S^+$ and $S^-$
Feed $m$ random samples to $A$ to get its hypothesis $h$
Return Yes if $h$ is consistent with all labelled samples $(S^+, S^-)$ (Return No otherwise)

Let’s check that $B$ satisfies the three conditions of an RP algorithm
Since $A$ efficiently PAC-learns 3-term DNF
- Number of samples needed by $A$ is $m = \text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta}) = \text{poly}(n)$
- Overall, $B$ always runs in $\text{poly}(n)$ time
If $G$ has no 3-coloring, no 3-term DNF $c(x)$ is consistent with all labelled samples
- Neither is $A$’s hypothesis $h(x)$ that is 3-term DNF
  - $B$ always says No
If $G$ has 3-coloring, some 3-term DNF $c(x)$ is consistent with all labelled samples
- With probability $\geq \delta = 1/2$, $A$ must output $h = c$ because $\varepsilon = 1/(2|S|)$ (effectively no error)
  - $B$ will say Yes

3. The Reduction

Reduction algorithm $R$ reads $G$ and outputs $S^+$ and $S^-$
Every vertex $v$ in $G$ yields a positive sample in $S^+$ that has 0 at position $v$ and 1 everywhere else
Every edge $(u, v)$ in $G$ yields a negative sample in $S^-$ that has 0 at positions $u$ and $v$ and 1 elsewhere

- e.g. \[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\end{array}
\]
\[
S^+ = \{01111, 10111, 11011, 11101, 11110\} \quad S^- = \{00111, 01011, 01101, 10110, 11010, 11100\}
\]

In general $S^+ = \{1_{\bar{v}} \mid v \in G\}$ and $S^- = \{1_{\bar{g}(u,v)} \mid (u,v) \in E\}$

Claim 3.1. If $G$ has 3-coloring, then $(S^+, S^-)$ is labelled by some 3-term DNF

Proof. Fix 3-coloring $f$ of $G$ using colors R, B, Y
$T_R = \text{conjunction of all } x_v \text{ such that } v \text{ is not red in } f$
$T_B, T_Y$ defined similarly (not blue, not yellow respectively)

When is $T_R(x)$ true? Every $x \in \{0,1\}^n$ is the indicator of some subset $S \subseteq V$, i.e. $x = 1_S$
- $T_R(1_S)$ is true $\iff$ $S$ contains all non red vertices $\iff$ $\overline{S}$ are all red
- $c = T_R \lor T_B \lor T_Y$ correctly labels $(S^+, S^-)$ because
  - $c(1_{\bar{v}}) = 1$ since $\{v\}$ is all red (or all blue, or all yellow)
  - $c(1_{\bar{g}(u,v)}) = 0$ since endpoints $u, v$ of an edge are not both red (nor both blue, nor both yellow) $\square$

Claim 3.2. If $(S^+, S^-)$ is labelled by some 3-term DNF, then $G$ has 3-coloring

Proof. Fix 3-term DNF $c = T_R \lor T_B \lor T_Y$ that correctly labels $(S^+, S^-)$
Color $v$ red if $T_R(1_{\bar{v}})$ is true; Similarly for blue and yellow
- If a vertex can get multiple colors, pick any one of them
  - $c(1_{\bar{v}}) = 1 \implies$ every vertex $v$ can get at least one color
  - $c(1_{\bar{g}(u,v)}) = 0 \iff T_R(1_{\bar{g}(u,v)}) = T_B(1_{\bar{g}(u,v)}) = T_Y(1_{\bar{g}(u,v)}) = 0$

When is $T_R(1_{\bar{g}(u,v)})$ false?
Let $P$ be the set of vertices whose positive literal appears in $T_R$; Likewise $N$ for negative
- $T_R(1_{\bar{g}(u,v)})$ is false $\iff$ $u \in P$ or $v \in P$ or some vertex $w \in N$ is distinct from $u, v$
  - if $u \in P$ then $T_R(1_{\bar{u}}) = 0$ and $u$ cannot be red (Likewise $v \in P$ implies $v$ cannot be red)
  - if some $w \in N \setminus \{u, v\}$, then $T_R(1_{\bar{u}}) = 0$ and $u$ cannot be red (and neither can $v$)
  - Thus $T_R(1_{\bar{g}(u,v)}) = 0$ implies at least one of $u$ or $v$ can’t be red
- $T_R(1_{\bar{g}(u,v)}) = T_B(1_{\bar{g}(u,v)}) = T_Y(1_{\bar{g}(u,v)}) = 0$ means $u$ and $v$ can’t get the same color $\square$