Notes 11: Proper vs Improper Learning

Proper learning: Algorithm required to output $h \in \mathcal{C}$, i.e. $\mathcal{H} = \mathcal{C}$
Improper learning: Algorithm allowed to output $h \notin \mathcal{C}$, i.e. $\mathcal{H} \supsetneq \mathcal{C}$
(Below) When $\mathcal{C} = \{\text{3-term DNF}\}$ 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: $\frac{1}{\varepsilon}\left(O(n) + \ln\frac{1}{\delta}\right)$ samples

---

1. 3-term DNF vs 3-CNF

Every 3-term DNF is 3-CNF
3-term DNF $f(x) = T_1 \vee T_2 \vee T_3$ where T_i are conjunctions
Since $\vee$ distributes over $\wedge$, i.e. $(u \wedge v) \vee (x \wedge y) = (u \vee x) \wedge (u \vee y) \wedge (v \vee x) \wedge (v \vee y)$

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

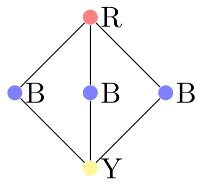
There is efficient improper PAC learning algorithm when $\mathcal{C} \subsetneq \mathcal{H} = \{\text{3-CNF}\}$
e.g. when $\mathcal{C} = \{\text{3-term DNF}\}$
Consistent Hypothesis Algorithm based on Elimination
$|\mathcal{H}| = 2^{\binom{n}{3}2^3} = 2^{O(n^3)} \implies \frac{1}{\varepsilon}\left(O(n^3) + \ln\frac{1}{\delta}\right)$ samples

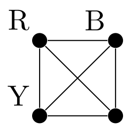
---

2. Graph 3-Coloring

Theorem 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





not 3-colorable

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 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 $\geqslant 1/2$ (can be boosted to $\geqslant 1 - 2^{-n}$)

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 poly(n) time (in particular $|S| \leqslant \text{poly}(n)$)
Labelled samples (S^+, S^-) from R corresponds to PAC-learning task with parameters
$\varepsilon = 1/(2|S|)$ $\delta = 1/2$ $\mathcal{D}$ = 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

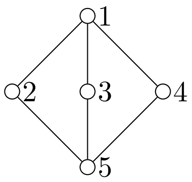
Run reduction R on G to get labelled samples S^+ and S^-
Feed m random samples to A to get its hypothesis h
Return Yes if h is consistent with all labelled samples (S^+, S^-) (Return No otherwise)

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 $\geqslant \delta = 1/2$, 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. 

$$S^+ = \begin{Bmatrix} 01111, \\ 10111, \\ 11011, \\ 11101, \\ 11110 \end{Bmatrix} \qquad S^- = \begin{Bmatrix} 00111, \\ 01011, \\ 01101, \\ 10110, \\ 11010, \\ 11100 \end{Bmatrix}$$

In general $\quad S^+ = \{\mathbb{1}_{\neq v} \mid v \in G\}$ and $S^- = \{\mathbb{1}_{\notin (u,v)} \mid (u, v) \in G\}$

Claim 2. *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
$T_R =$ conjunction of all x_v such that v 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. $x = \mathbb{1}_S$
$T_R(\mathbb{1}_S)$ is true $\iff$ S contains all non red vertices $\iff$ $\overline{S}$ are all red
$c = T_R \vee T_B \vee T_R$ correctly labels (S^+, S^-) because
$c(\mathbb{1}_{\neq v}) = 1$ since $\{v\}$ is all red (or all blue, or all yellow)
$c(\mathbb{1}_{\notin (u,v)}) = 0$ since endpoints u, v of an edge are not both red (nor both blue, nor both yellow) $\square$

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

Proof. Fix 3-term DNF $c = T_R \vee T_B \vee T_Y$ that correctly labels (S^+, S^-)
Color v red if $T_R(\mathbb{1}_{\neq v})$ is true; Similarly for blue and yellow
If a vertex can get multiple colors, pick any one of them
$c(\mathbb{1}_{\neq v}) = 1 \implies$ every vertex v can get at least one color
$c(\mathbb{1}_{\notin (u,v)}) = 0 \iff T_R(\mathbb{1}_{\notin (u,v)}) = T_B(\mathbb{1}_{\notin (u,v)}) = T_Y(\mathbb{1}_{\notin (u,v)}) = 0$
When is $T_R(\mathbb{1}_{\notin (u,v)})$ false?
Let P be the set of vertices whose positive literal appears in T_R; Likewise N for negative
$T_R(\mathbb{1}_{\notin (u,v)})$ 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 $T_R(\mathbb{1}_{\neq u}) = 0$ and u cannot be red (Likewise $v \in P$ implies v cannot be red)
if some $w \in N \setminus \{u, v\}$, then $T_R(\mathbb{1}_{\neq u}) = 0$ and u cannot be red (and neither can v)
Thus $T_R(\mathbb{1}_{\notin (u,v)}) = 0$ implies at least one of u or v can't be red
$T_R(\mathbb{1}_{\notin (u,v)}) = T_B(\mathbb{1}_{\notin (u,v)}) = T_Y(\mathbb{1}_{\notin (u,v)}) = 0$ means u and v can't get the same color $\square$