Hardness Results for PAC learning

These notes are slightly edited from scribe notes in previous years.

1 Details of Computational Hardness Result

Theorem 1: (more precise statement) If 3-term DNF is PAC-learnable by an algorithm using hypotheses in 3-Term DNF, then 3-colorability ∈ RP (i.e., RP=NP).

Proof:
Given graph G as input, Translate(G) is a set of boolean examples: $S = S^+ \cup S^-$ (positive and negative examples, respectively.) such that G is 3-colorable iff $\exists$ a 3-term DNF consistent with all the examples.

Part 1: Assume Translate(G) is given. We construct an algorithm for 3-Coloring.
1. Calculate $S^+, S^- = T(G)$
2. Run learner with $\epsilon = \frac{1}{2|S^+ \cup S^-|}$, $\delta = 1/4$.
   Use uniform distribution over examples in $S = S^+ \cup S^-$.
3. When algorithm outputs $h$ check if $h$ is consistent with $S$.
   If Yes → output Yes
   Else → output No

Part 2: Translate(G)
$G$ has $n$ nodes, $S$ has $n$-bit strings.
$0_i = a$ string of 1s with a single 0 at bit $i$.
$0_{i,j} = 0$ at $i,j$
$S^+ = \{0_i\}$ for $i=1, \ldots, n$
$S^- = \{0_{i,j} \text{ for } i,j \in E\}$
For example, with 5 nodes, and edges $\{(1,2), (1,3), (1,4), (2,5), (3,5), (4,5)\}$:
$S^+ = \{01111, 10111, 11011, 11101, 11110\}$
$S^- = \{00111, 01011, 01101, 10110, 11010, 11100\}$

Claim: $G$ is 3-colorable iff $\exists$ 3-term DNF consistent with $S$.

Proof:
Assume that $G$ is 3-colorable. We construct a 3-term DNF consistent with $S$ as follows: for color $c$ let $t_c = \land x_i$ and let our DNF expression be $T_R \lor T_G \lor T_B$. In our example, with coloring: RBBBBG we get $x_2x_3x_4x_5 \lor x_1x_5 \lor x_1x_2x_3x_4$.

We next show that the DNF is consistent with $S$. For $S^+$ notice that $0_i$ is satisfied by term $t_{\text{color}(i)}$. For $S^-$, let $(i,j) \in E$ and $0_{i,j}$ be the corresponding assignment in $S$. Then $O_{i,j}$ violates: (1) $T_{\text{color}(i)}$ since it includes $x_j$ (2) $T_{\text{color}(j)}$ since it includes $x_i$ (3) $T_{\text{thirdcolor}}$ since it includes both $x_i, x_j$. 

Assume that $G$ is 3-colorable. We construct a 3-term DNF consistent with $S$ as follows: for color $c$ let $t_c = \land x_i$ and let our DNF expression be $T_R \lor T_G \lor T_B$. In our example, with coloring: RBBBBG we get $x_2x_3x_4x_5 \lor x_1x_5 \lor x_1x_2x_3x_4$.

We next show that the DNF is consistent with $S$. For $S^+$ notice that $0_i$ is satisfied by term $t_{\text{color}(i)}$. For $S^-$, let $(i,j) \in E$ and $0_{i,j}$ be the corresponding assignment in $S$. Then $O_{i,j}$ violates: (1) $T_{\text{color}(i)}$ since it includes $x_j$ (2) $T_{\text{color}(j)}$ since it includes $x_i$ (3) $T_{\text{thirdcolor}}$ since it includes both $x_i, x_j$. 

1
For the other direction assume that $t_1 \land t_2 \land t_3$ is consistent with $S$, then we construct a coloring as follows: node $i$ is colored 1 if $0_i$ satisfied by $t_1$, else colored 2 if $0_i$ is satisfied by $t_2$, else (it must be satisfied by $t_3$) colored 3.

We need to show that $(i,j) \in E \rightarrow \text{color}(i) \neq \text{color}(j)$.

if color($i$) = color($j$) then $0_i, 0_j$ are satisfied by same term. Therefore none of $x_i, \bar{x}_i, x_j, \bar{x}_j$ can appear in this term (since otherwise one of the assignments violates the term). As a result $0_{i,j}$ is also satisfied by this term and the DNF is not consistent, contradicting our assumption.

Proof of Theorem

Case 1: $G$ is not 3-colorable. There is no consistent hypothesis (by the correctness of translate). Thus the hypothesis of the algorithm is also not consistent, and we say ‘No’.

Case 2: $G$ is 3-colorable $\rightarrow$ Learning run is ‘legal’ therefore by the guarantees of PAC learning w.p. $\geq 1-\delta=3/4$, $err(h) < \frac{1}{2|S|}$. For example, if $|S| = 11$, $\epsilon=1/22$; but the probability of every example in $S$ is $p(ex)= 1/11 < 1/22$. Therefore $err<\frac{1}{2|S|}$ implies that $err=0$ for our distribution. This implies that hypothesis is consistent, which in turn implies that algorithm says ‘Yes’.

2 Representation Independent Hardness Results

The previous result shows that learning is hard. But this holds only if we insist on using 3 Term DNF as a representation for hypotheses. Are there classes that are not learnable regardless of how we represent hypotheses? the answer is Yes (under some assumptions that certain cryptographic systems are secure). We only discussed the general idea which goes as follows.

Consider an encryption system with encryption function $E()$ that encrypts message $x$ as $y = E(x)$, and the corresponding decryption function $D(y)$ that returns $x$. Consider any concept class $C$ that is expressive enough to represent (one bit in the output of) the decryption function. If we can learn $C$ then we can break the crypto-system as follows. First we generate a training set by drawing random messages $x$, encrypting them to get $E(x)$ and using $E(x)$ as the example and (one bit of) $x = D(E(x))$ as the label. Then the target concept is the same as $D()$ and learning is is the same as learning to decrypt messages. The details are more complex and they can be found in Chapter 6 of [KV].