Haiyang Wang
Oct 5th, 2020
hw1927@nyu.edu

Foundations of Machine Learning
Homework 1

A. Consistent hypothesis

Let $\mathcal{Z}$ be a finite set of m labeled pts. Given PAC algorithm $\mathcal{A}$, show that you can use $\mathcal{A}$ and a finite training sample S to find in polynomial time a hypothesis $h \in \mathcal{H}$ that is consistent with $\mathcal{Z}$, with high probability.

(Hint: select an appropriate distribution $\mathcal{D}$ over $\mathcal{Z}$ and give a condition on $R(h)$ for h to be consistent.)

Proof. I will proceed the proof following the hint.

Let the $\mathcal{D}$ be the uniform distribution over $\mathcal{Z}$. Let $X = (x_i)_{i=1}^m$ be the input space.

Suppose we have training sample $S \sim D^n$, and the algorithm selects hypothesis h_S based on the training sample, the probability that S is inconsistent with $\mathcal{Z}$ is:

$$\begin{aligned}
\mathbb{P}_{S\sim D^n}(S \text{ inconsistent with } \mathcal{Z}) &= \mathbb{P}_{S\sim D^n}(h_S(x_i) \neq \mathcal{Z}(x_i) \text{ for some } i = 1, ..., m) &(1)\\
&= \mathbb{P}_{S\sim D^n}(\cup_{i=1}^m \{S | h_S(x_i) \neq \mathcal{Z}(x_i)\}) &(2)\\
&\leq \sum_{i=1}^m \mathbb{P}_{S\sim D^n}(h_S(x_i) \neq \mathcal{Z}(x_i)) &(3)
\end{aligned}$$

$\mathbb{P}_{S\sim D^n}(h_S(x_i) \neq \mathcal{Z}(x_i))$ can be bound by the probability of $\mathcal{A}$ did not learn x_i at all. This is because, if it is learned, then by consistence, it have to agree with $\mathcal{Z}$. What's the probability of not learning x_i? It is $x_i \notin S$.

$$\begin{aligned}
\sum_{i=1}^m \mathbb{P}_{S\sim D^n}(h_S(x_i) \neq \mathcal{Z}(x_i)) &\leq \sum_{i=1}^m \mathbb{P}_{S\sim D^n}(x_i \notin S) &(4)\\
&= \sum_{i=1}^m (1 - \frac{1}{m})^n \leq me^{-n/m} &(5)
\end{aligned}$$

To bound $\mathbb{P}_{S\sim D^n}(S \text{ inconsistent with } \mathcal{Z})$ by δ, we only need to have: $me^{-n/m} \leq \delta$, which is equivalent of $n \geq m \log \frac{m}{\delta}$. In other words, if size of training sample S is greater than $m \log \frac{m}{\delta}$, we have confidence of $1 - \delta$ that $h_S = \mathcal{Z}$

Time complexity of this algorithm is simply

■

B. Oracle PAC Learning

1. Learning unions of intervals

. **Proof.** I will simply give algorithm for general p and prove its PACness, instead of the specific case $p = 3$.

The algotithm $\mathcal{A}$ is:

- Suppose the training sample $S \sim D^m = \{(x_1, l_1), ..., (x_m, l_m)\}$, where x_i is from the input space, and l_m is the label of the point x_i. Without lost of generality, I take a sort so that $x_1 < x_2 < \cdots < x_m$.
- Let $S_- = \{x_i \in S | l_i = -1\}$ be the set of points not contained in our targer concept.For convenience I denote these points by $y_1, ..., y_{m^-}$, where $m^- = |S_-|$. Similary, I can have $S_+ = \{z_1, \cdots, z_{m^+}\}$.
- The real line excluding S_- becomes union of open intervals:$\cup U_j$, where $U_j = (y_j, y_{j+1}), y_0 = -\infty, y_{m^-+1} = +\infty$.
- For each U_j, if $U_j \cap S_+ = \emptyset$, we can ignore this interval. Otherwise, we took the smallest closed interval inside U_j to be V_j.
- Suppose we get q closed intervals $V_1, ..., V_q$ from last operation. This is of course after reindexing V_j.
- It is obvious that $q \le p$. If $q = p$, we simply return (V_j) as our output.
- Otherwise, We consider the family of open intervals O_r within $\cup V_j$, whose intersection with S_+ is empty. Taking a particular O_{max} with maximum regular borel measure. Suppose $O_{max} = (a, b) \subset V_j = [c, d]$, then we replace V_j in our resulting sequence with $V_j = [c, a], V_{q+1} = [b, d]$. This process will add the number of closed intervals we are outputing. By finite amount of iteration, we can have that $q = p$.
- Output our results $V_1, \cdots, V_p$.

Before proceeding to proof, let me first explain why this algorithm will work: It is basiclly selecting p closed intervals with no negative points inside such that the union of these interval has minimal regular borel measure. This minimal-ness certainly prevent us from having significantly large false-positive. Why is the false-negative also small? Because it does not contain any negative points in the training sample, and in fact, tried to stay away from them as far as possibile.

Next, I will try to prove $\mathcal{A}$ is PAC-learning, for a given $p \in \mathbb{N}^+$.

Suppose our target concept c is $[a_1, b_1] \cup \cdots \cup [a_p, b_p]$ with $a_i < b_i < a_{i+1} < b_{i+1}$ for all i. The complement of c can be denoted by $\bar{c} = \cup O_i = \cup_{i=0}^{p} (b_i, a_{i+1})$ with $b_0 = -\infty, a_{p+1} = +\infty$. And for $S \sim D^m$ the algorithm gives us the hypothesis $h_S = [a'_1, b'_1] \cup \cdots \cup [a'_p, b'_p]$. What we are interested in, is the distribution of the measure of $c \Delta h_S$. This Δ is in set-theoratical sense: $A \Delta B = (A \cup \bar{B}) \cup (\bar{A} \cup B)$.

However, it would be too ideal to measure this particular set, which seems natural to our intuition. What we have to do is using the method in the example of axis-aligned rectangle.

$\forall \epsilon > 0$, We can take closed intervals alone the boundaries of target concept c, to make the target concept a bit bigger. Each of these intervals has probability mass equals to $\frac{\epsilon}{2p}$. We now calls those intervals T_i, with $i = 1, \cdots, 2p$. If our h_S meets all those T_i, then its false negative error will be bounded by ϵ:

$$\mathbb{P}(R(h_S)'s \text{ false negative part} > \epsilon) \le \mathbb{P}(h_S \cap (\cup T_i)) \le 2p(1 - \frac{\epsilon}{2k})^m \le 2pe^{-\epsilon m/2p}$$

The false positive error, can be bounded in a different way. Without lost of generality, the O_i has non-zero probability mass. This is because, if it is with zero probability mass, the false positive errors in these places cannot have any effect on you error, the distribution will simply not come here. Also, notice that if our sample contains a point in O_i, then the resulting hypothesis set will have empty intersection with O_i. Because our algorithm tries to stay away from those negative point as far as possible. Thus, we can have the probalistic bound for a non zero false positive.

$$\mathbb{P}_{S \sim D^m}(R_S(h_S)'s \text{ false positive} > 0) \le \sum_{i=0}^{i=p} \mathbb{P}(S \cap O_i =) \le \sum (1 - \mathbb{P}(O_i))^m \le (p+1) \max_i (1 - \mathbb{P}(O_i))^m \le (p+1) e^{-m \min_i \mathbb{P}(O_i)}$$

In total the 2 following senarios can cover the possibility of general error $> \epsilon$:

- false positive is 0 and false negative $> \epsilon$
- false positve is non-zero

Thus we can have the following bound:

$$`\mathbb{P}_{S\sim D^m}(R_S(h_S) > \epsilon) \leq 2pe^{-\epsilon m/2p} + (p+1)e^{-m\min_i \mathbb{P}(O_i)}$$

by taking ϵ smaller enough, we can see that $e^{-m\min_i \mathbb{P}(O_i)} \leq e^{-\epsilon m/2p}$. Thus, the probabilistic bound for error can be simplified into:

$$\mathbb{P}(R_S(h_S) > \epsilon) \leq (3p+1)e^{-\epsilon m/2p}$$

So, if we let $m > \frac{2p}{\epsilon}\log(\frac{3p+1}{\delta})$, we have have $\mathbb{P}(R_S(h_S) \geq \epsilon) \leq \delta$.
By now, we render a PAC algorithm for any concept class of p closed intervals.
The sample complexity is $O(p\log p)$. The time complexity is $O(m^2) = O(p^2\log^2(p))$ ■

2. Hypothesis testing

a Is PAC-learning possible when p is not provided?

For fixed $\epsilon, \delta > 0, i > 0$, the sample size $n = \frac{32}{\epsilon}\left[i\log 2 + \log\frac{2}{\delta}\right]$. And we draw sample $S \sim D^n$ for unknow distribution D. A hypothesis h is accepted if it makes at most 0.75ϵ errors on S and it is rejected otherwise.

b Chernoff bound for accepted

Suppose $R(h) > \epsilon$. Use mutiplicative Chernoff bound to show that $\mathbb{P}_{S\sim D^n}[h \text{ is accepted}] \leq \frac{\delta}{2^{i+1}}$.

$$\begin{aligned}
\mathbb{P}_{S\sim D^n}[h \text{ is accepted}] &= \mathbb{P}_{S\sim D^n}[\hat{R}_S(h) \leq (1-0.25)\epsilon] && (6)\\
&\leq \mathbb{P}_{S\sim D^n}[\hat{R}_S(h) \leq (1-0.25)R(h)] && (7)\\
&\leq \exp(-nR(h)/32) && (8)\\
&= \exp(-\frac{R(h)}{\epsilon}\left[i\log 2 + \log\frac{2}{\delta}\right]) && (9)\\
&\leq \exp(-\left[\log\frac{2^{i+1}}{\delta}\right]) && (10)\\
&\leq \frac{\delta}{2^{i+1}} && (11)
\end{aligned}$$

c Chernoff bound for rejected

Suppose $R(h) \leq \frac{\epsilon}{2}$, from Chernoff bound we have:

$$\begin{aligned}
\mathbb{P}_{S\sim D^n}[h \text{ is rejected}] &= \mathbb{P}_{S\sim D^n}[\hat{R}_S(h) \geq 0.75\epsilon] && (12)\\
&\leq \mathbb{P}_{S\sim D^n}[\hat{R}_S(h) \geq (1+0.5)R(h)] && (13)\\
&\leq \exp(-nR(h)/8) && (14)\\
&\leq \frac{\delta}{2^{i+1}} && (15)
\end{aligned}$$

By similar computations following from (b).

d Algorithm $\mathcal{B}$'s halting probability

At iteration i of the algorithm $\mathcal{B}$, $\tilde{s} = \left\lfloor 2^{(i-1)/\log\frac{2}{\delta}} \right\rfloor \geq s$. The test sample's size will be $n = \frac{32}{\epsilon}\left[i \log 2 + \log \frac{2}{\delta}\right]$. And h_i by its definition satisfies $\mathbb{P}[R(h_i) \leq \frac{\epsilon}{2}] \geq \frac{1}{2}$, if $\tilde{s} = s$. In case $\tilde{s} > s$, it is true, since we will have more false negative error with the larger parameter $\tilde{s}$ by definition of algorithm $\mathcal{A}$. In other words, with probability $1/2$, we can assume that $R(h_i) \leq \frac{\epsilon}{2}$.

Then, we can have:

$$\mathbb{P}\left[h_i \text{ is accepted }\right] \geq \mathbb{P}\left[h_i \text{ is accepted } \bigwedge R(h_i) \leq \frac{\epsilon}{2}\right] = \mathbb{P}\left[R(h_i) \leq \frac{\epsilon}{2}\right] - \mathbb{P}\left[h_i \text{ is rejected } \bigwedge R(h_i) \leq \frac{\epsilon}{2}\right]$$

$$\geq 1/2 - \frac{\delta}{2^{i+1}} \geq 1/2 - 1/4 = 3/8$$

e not halting probability

$$\mathbb{P}(\text{not halting after j iteration}) \leq (\frac{5}{8})^j \leq (\frac{5}{8})^{\frac{\log\frac{2}{\delta}}{\log\frac{8}{5}}} \leq \frac{\delta}{2}$$

f $\tilde{s} \geq s$

$$\widetilde{s} = \left\lfloor 2^{\,(i-1)/\log\frac{2}{\delta}} \right\rfloor \geq 2^{\log_2 s} = s$$

g desired result

At most after iteration j', the confidence of our learning result with error bound ϵ can be established by **b** and **c**. The probability of halting can be established by **e** and **f**. Thus we finished this problem.