Mehryar Mohri
Foundations of Machine Learning 2023
Courant Institute of Mathematical Sciences
Homework assignment 1
September 12, 2023
Due: September 26, 2023

A Consistent hypotheses

We showed in class that for a finite hypothesis set $\mathcal{H}$, a consistent learning algorithm $\mathcal{A}$ is a PAC-learning algorithm. Here, we consider a converse question. Let $\mathcal{Z}_k$ be a finite set of m_k labeled points, for $1 \le k \le p$. Suppose that you are given a PAC-learning 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 all the p finite sets $\mathcal{Z}_1, \ldots, \mathcal{Z}_p$, with high probability. [hint: you can select an appropriate distribution $\mathcal{D}$ and give a condition on $R(h)$ for h to be consistent.]

Solution: Since PAC-learning with $\mathcal{A}$ is possible for any distribution, let $\mathcal{D}$ be the uniform distribution over the union of $\mathcal{Z}_k$'s, $\cup_{k=1}^{p} \mathcal{Z}_k$. Let m be the size of the union set $\cup_{k=1}^{p} \mathcal{Z}_k$. Note that, in that case, the cost of an error of a hypothesis h on any point $z \in \cup_{k=1}^{p} \mathcal{Z}_k$ is $\mathbb{P}_{\mathcal{D}}[z] = 1/m$. Thus, if $R_{\mathcal{D}}(h) < 1/m$, we must have $R_{\mathcal{D}}(h) = 0$ and h is consistent. Thus, choose $\epsilon = 1/(m+1)$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over samples S with $|S| \ge P((m+1), 1/\delta)$ points (where P is some fixed polynomial) the hypothesis h_S returned by $\mathcal{A}$ is consistent with $\cup_{k=1}^{p} \mathcal{Z}_k$ since $R_{\mathcal{D}}(h_S) \le 1/(m+1) < 1/m$.

B Learning unions of intervals

Give a PAC-learning algorithm for the concept class $\mathcal{C}_2$ formed by unions of two closed intervals, that is $[a, b] \cup [c, d]$, with $a, b, c, d \in \mathbb{R}$. You should carefully describe and justify your algorithm. Extend your result to derive a PAC-learning algorithm for the concept class $\mathcal{C}_p$ formed by unions of $p \ge 1$ closed intervals, thus $[a_1, b_1] \cup \cdots \cup [a_p, b_p]$, with $a_k, b_k \in \mathbb{R}$ for $k \in [p]$. What are the time and sample complexities of your algorithm as a function of p?

Solution: Given a sample S, our algorithm consists of the following steps:

(a) Sort S in ascending order.

(b) Loop through sorted S, marking where intervals of consecutive positively labeled points begin and end.

(c) Return the union of intervals found on the previous step. This union is represented by a list of tuples that indicate start and end points of the intervals.

This algorithms works both for $p = 2$ and for a general p. We will now consider the problem for $\mathcal{C}_2$. To show that this is a PAC-learning algorithm we need to distinguish between two cases.

The first case is when our target concept is a disjoint union of two closed intervals: $I = [a, b] \cup [c, d]$. Note, there are two sources of error: false negatives in $[a, b]$ and $[c, d]$ and also false positives in (b, c). False positives may occur if no sample is drawn from (b, c). By linearity of expectation and since these two error regions are disjoint, we have that $R(h_S) = R_{\text{FP}}(h_S) + R_{\text{FN},1}(h_S) + R_{\text{FN},2}(h_S)$, where

$$\begin{aligned} R_{\text{FP}}(h_S) &= \mathbb{P}_{x \sim \mathcal{D}}[x \in h_S, x \notin I], \\ R_{\text{FN},1}(h_S) &= \mathbb{P}_{x \sim \mathcal{D}}[x \notin h_S, x \in [a, b]], \\ R_{\text{FN},2}(h_S) &= \mathbb{P}_{x \sim \mathcal{D}}[x \notin h_S, x \in [c, d]]. \end{aligned}$$

Since we need to have that at least one of $R_{\text{FP}}(h_S)$, $R_{\text{FN},1}(h_S)$, $R_{\text{FN},2}(h_S)$ exceeds $\epsilon/3$ in order for $R(h_S) > \epsilon$, by union bound

$$\begin{aligned}\mathbb{P}(R(h_S) > \epsilon) &\le \mathbb{P}(R_{\text{FP}}(h_S) > \epsilon/3 \text{ or } R_{\text{FN},1}(h_S) > \epsilon/3 \text{ or } R_{\text{FN},2}(h_S) > \epsilon/3) \\ &\le \mathbb{P}(R_{\text{FP}}(h_S) > \epsilon/3) + \sum_{i=1}^{2} \mathbb{P}(R_{\text{FN},i}(h_S) > \epsilon/3). \qquad (1)\end{aligned}$$

We first bound $\mathbb{P}(R_{\text{FP}}(h_S) > \epsilon/3)$. Note that if $R_{\text{FP}}(h_S) > \epsilon/3$, then $\mathbb{P}((b,c)) > \epsilon/3$ and hence

$$\mathbb{P}(R_{\text{FP}}(h_S) > \epsilon/3) \le (1 - \epsilon/3)^m \le e^{-m\epsilon/3}.$$

Now we can bound $\mathbb{P}(R_{\text{FN},i}(h_S) > \epsilon/3)$ by $2e^{-m\epsilon/6}$ using the same argument as in the previous question. Therefore,

$$\mathbb{P}(R(h_S) > \epsilon) \le e^{-m\epsilon/3} + 4e^{-m\epsilon/6} \le 5e^{-m\epsilon/6}.$$

Setting the right-hand side to δ and solving for m yields that $m \ge \frac{6}{\epsilon} \log \frac{5}{\delta}$.

The second case that we need to consider is when $I = [a,d]$, that is, $[a,b] \cap [c,d] \ne \varnothing$. In that case, our algorithm reduces to that of learning a single $[a,b]$ interval and it is easy to show that only $m \ge \frac{2}{\epsilon} \log \frac{2}{\delta}$ samples is required to learn this concept, as with the axis-aligned rectangles case discussed in class. Therefore, we conclude that our algorithm is indeed a PAC-learning algorithm.

Extension of this result to the case of $\mathcal{C}_p$ is straightforward. The only difference is that in (1), one has two summations for $p-1$ regions of false positives and $2p$ regions of false negatives. In that case sample complexity is $m \ge \frac{2(2p-1)}{\epsilon} \log \frac{3p-1}{\delta}$.

Sorting step of our algorithm takes $O(m \log m)$ time and steps (b) and (c) are linear in m, which leads to overall time complexity $O(m \log m)$.

C Rejection

We first introduce a method for testing a hypothesis h, with high probability. Fix $\epsilon > 0$, $\delta > 0$, and define the sample size n by $n = \frac{32}{\epsilon}[\log 2 + \log \frac{2}{\delta}]$. Suppose we draw an i.i.d. sample S of size n according to some unknown distribution $\mathcal{D}$. We will say that a hypothesis h is *accepted* if it makes at most $(3/4)\epsilon$ errors on S and that it is *rejected* otherwise. Thus, h is accepted iff $\widehat{R}(h) \le (3/4)\epsilon$.

1. Assume that $R(h) \ge \epsilon$. Use the (multiplicative) Chernoff bound to show that in that case:

$$\mathbb{P}_{S \sim \mathcal{D}^n}[h \text{ is accepted}] \le \frac{\delta}{4}.$$

Solution: By definition of acceptance,

$$\begin{aligned}\mathbb{P}[h \text{ is accepted}] &= \mathbb{P}[\widehat{R}_S(h) \le (3/4)\epsilon] \\ &\le \mathbb{P}[\widehat{R}_S(h) \le (3/4)\, R(h)] && (R(h) \ge \epsilon) \\ &\le \exp\Big(-\frac{n}{2} R(h)(1/4)^2\Big) && \text{(Chernoff bound)} \\ &= \exp\Big(-\frac{R(h)}{\epsilon} \log \frac{4}{\delta}\Big) && \text{(def. of } n) \\ &\le \exp\Big(-\log \frac{4}{\delta}\Big) = \frac{\delta}{4}. && (R(h) \ge \epsilon).\end{aligned}$$

2. Assume that $R(h) \le \epsilon/2$. Use the (multiplicative) Chernoff bounds to show that in that case:

$$\mathbb{P}_{S \sim \mathcal{D}^n}[h \text{ is rejected}] \le \frac{\delta}{4}.$$

Solution: By definition, $\mathbb{P}[h \text{ is rejected}] = \mathbb{P}[\widehat{R}_S(h) \geq \frac{3}{4}\epsilon]$. Since $R(h) \leq \epsilon/2$, $\mathbb{P}[h \text{ is rejected}] \leq \mathbb{P}[\widehat{R}_S(h) \geq \frac{3}{4}\epsilon \mid R(h) = \epsilon/2]$. By the Chernoff bounds, we can thus write

$$\begin{aligned}
\mathbb{P}[h \text{ is rejected}] &\leq \exp\Big(-\frac{n}{3}\frac{\epsilon}{2}(1/2)^2\Big) && \text{(Chernoff bound)}\\
&= \exp\Big(-\frac{4}{3}\log\frac{4}{\delta}\Big) && \text{(def. of } n\text{)}\\
&\leq \exp\Big(-\log\frac{4}{\delta}\Big) = \frac{\delta}{4}.
\end{aligned}$$

D Oracle PAC learning

1. In Problem B, the learning algorithm was given p as input.

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

   Solution: Argue that it is not possible in general since the union of all $\mathcal{C}_p$s is not PAC-learnable (infinite VC-dimension).

Now, consider, more generally, a family of concept classes $\{\mathcal{C}_s\}_s$ where $\mathcal{C}_s$ is the set of concepts in $\mathcal{C}$ with size at most some integer s. Suppose we have a PAC-learning algorithm $\mathcal{A}$ that can be used for learning any concept class $\mathcal{C}_s$ when s is given. Can we convert $\mathcal{A}$ into a PAC-learning algorithm $\mathcal{B}$ that does not require the knowledge of s? This is the main objective of the rest of this problem.
To do so, we will use the definitions and results introduced in the previous problem.
Assume that the target concept belongs to some class $\mathcal{C}_s$, with s unknown to the learner. Algorithm $\mathcal{B}$ is then defined as follows: we start with $i = 1$ and, at each round $i \geq 1$, we guess the parameter size s to be $\widetilde{s} = \lfloor 2^{(i-1)/\log\frac{2}{\delta}} \rfloor$. We draw a sample S of size $n = \frac{32}{\epsilon}[i\log 2 + \log\frac{2}{\delta}]$ (which depends on i) to test the hypothesis h_i returned by $\mathcal{A}$ when it is trained with a sample of size $S_{\mathcal{A}}(\epsilon/2, 1/2, \widetilde{s})$, that is the sample complexity of $\mathcal{A}$ for a required precision $\epsilon/2$, confidence $1/2$, and size $\widetilde{s}$ (we ignore the size of the representation of each example here). If h_i is accepted, the algorithm stops and returns h_i, otherwise it proceeds to the next iteration.

   (b) Show that if at iteration i, the estimate $\widetilde{s}$ is larger than or equal to s, then $\mathbb{P}[h_i \text{ is accepted}] \geq 3/8$.

   Solution: The estimate $\widetilde{s}$ is then an upper bound on s and thus, by definition of algorithm $\mathcal{B}$, $\mathbb{P}[R(h_i) \leq \epsilon/2] \geq 1/2$. If a hypothesis h_i has error at most $\epsilon/2$ it is rejected with probability at most $\delta/2^{i+1} \leq \delta/4 \leq 1/4$, therefore, it is accepted with probability at least $3/4$. Thus, for $\widetilde{s} \geq s$, $\mathbb{P}[h_i \text{ is accepted}] \geq (1/2) \times (3/4) = 3/8$.

   (c) Show that the probability that $\mathcal{B}$ does not halt after $j = \lceil \log\frac{2}{\delta}/\log\frac{8}{5} \rceil$ iterations with $\widetilde{s} \geq s$ is at most $\delta/2$.

   Solution: By the previous question, the probability that algorithm $\mathcal{B}$ fails to halt while $\widetilde{s} \geq s$ is at most $1 - 3/8 = 5/8$. Thus, the probability that it does not halt after j iterations is at most $(5/8)^j \leq (5/8)^{\log\frac{2}{\delta}/\log\frac{8}{5}} = \exp\Big(\log\frac{2}{\delta}/\log\frac{8}{5}\log\frac{5}{8}\Big) = \delta/2$.

   (d) Show that for $i \geq \lceil 1 + (\log_2 s)\log\frac{2}{\delta} \rceil$, the inequality $\widetilde{s} \geq s$ holds.

Solution: By definition,

$$\begin{aligned}
\widetilde{s} \geq s &\iff \lfloor 2^{(i-1)/\log\frac{2}{\delta}} \rfloor \geq s \\
&\iff 2^{(i-1)/\log\frac{2}{\delta}} \geq s \\
&\iff \frac{i-1}{\log\frac{2}{\delta}} \geq \log_2 s \\
&\iff i \geq 1 + (\log_2 s)\log\frac{2}{\delta} \\
&\iff i \geq \lceil 1 + (\log_2 s)\log\frac{2}{\delta} \rceil.
\end{aligned}$$

(e) Show that with probability at least $1-\delta$, algorithm $\mathcal{B}$ halts after at most $j' = \lceil 1 + (\log_2 s)\log\frac{2}{\delta}\rceil + j$ iterations and returns a hypothesis with error at most ϵ.

Solution: In view of the two previous questions, with probability at least $1-\delta/2$, algorithm $\mathcal{B}$ halts after at most j' iterations. The probability that the hypothesis it returns be accepted while its error is greater than ϵ is at most $\delta/2^{j'+1} \leq \delta/2$. Thus, with probability $1-\delta$, the algorithm halts and the hypothesis it returns has error at most ϵ.