A. PAC-learning of intervals

Consider the concept class $C$ formed by closed intervals on the real line. Give a PAC-learning algorithm for $C$. The analysis is similar to that of the axis-aligned rectangles given in class, but you should carefully present and justify your proof.

Solution: Given a sample $S$, one algorithm consists of returning the tightest closed interval $I_S$ containing positive points. Let $I = [a, b]$ be the target concept. If $\Pr[I] < \epsilon$, then clearly $R(I_S) < \epsilon$. Assume that $\Pr[I] \geq \epsilon$. Consider two intervals $I_L$ and $I_R$ defined as follows:

\[
I_L = [a, x] \quad \text{with} \quad x = \inf\{x: \Pr[a, x] \geq \epsilon/2\} \\
I_R = [x', b] \quad \text{with} \quad x' = \sup\{x': \Pr[x', b] \geq \epsilon/2\}.
\]

By the definition of $x$, the probability of $[a, x]$ is less than or equal to $\epsilon/2$, similarly the probability of $[x', b]$ is less than or equal to $\epsilon/2$. Thus, if $I_S$ overlaps both with $I_L$ and $I_R$, then its error region has probability at most $\epsilon$. Thus, $R(I_S) > \epsilon$ implies that $I_S$ does not overlap with either $I_L$ or $I_R$, that is either none of the training points falls in $I_L$ or none falls in $I_R$. Thus, by the union bound,

\[
\Pr[R(I_S) > \epsilon] \leq \Pr[S \cap I_L = \emptyset] + \Pr[S \cap I_R = \emptyset] \\
\leq 2(1 - \epsilon/2)^m \leq 2e^{-me/2}.
\]

Setting $\delta$ to match the right-hand side gives the sample complexity $m = \frac{2\epsilon}{\epsilon} \log \frac{2}{\delta}$ and proves the PAC-learning of closed intervals.

B. Bayesian bound

Let $H$ be a hypothesis countable set of functions mapping $X$ to $\{0, 1\}$ and let $p$ be a probability measure over $H$ (prior probability). Use Hoeffding’s inequality to show that for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:

\[
\forall h \in H, R(h) \leq \hat{R}(h) + \sqrt{\log \frac{1}{p(h)} + \frac{1}{2m}}.
\]
Compare this result with the bound given in class in the inconsistent case for finite hypothesis sets (hint: you could use $\delta' = p(h)\delta$ as confidence parameter in Hoeffding’s inequality).

Solution: For any fixed $h \in H$, by Hoeffding’s inequality, for any $\delta > 0$,

$$\Pr \left[ R(h) - \widehat{R}(h) \geq \sqrt{\frac{\log \frac{1}{p(h)\delta}}{2m}} \right] \leq p(h)\delta. \tag{2}$$

By the union bound,

$$\Pr \left[ \exists h: R(h) - \widehat{R}(h) \geq \sqrt{\frac{\log \frac{1}{p(h)\delta}}{2m}} \right] \leq \sum_{h \in H} \Pr \left[ R(h) - \widehat{R}(h) \geq \sqrt{\frac{\log \frac{1}{p(h)\delta}}{2m}} \right] \leq \sum_{h \in H} p(h)\delta = \delta.$$

In the case of a finite hypothesis set and a uniform prior $p(h) = 1/|H|$, the bound coincides with the one presented in class. \qed

C. Learning with an unknown parameter

In class, we showed that the concept class of $k$-CNF is PAC-learnable. Note, however, that the learning algorithm is given $k$ as input. Is PAC-learning possible even when $k$ is not provided? More generally, consider a family of concept classes $(C_s)_s$ where $C_s$ is the set of concepts in $C$ with size at most $s$. Suppose we have a PAC-learning algorithm $\mathcal{A}$ that can be used for learning any concept class $C_s$ 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 this problem.

To do this, we first introduce a method for testing a hypothesis $h$, with high probability. Fix $\epsilon > 0$, $\delta > 0$, and $i \geq 1$ and define the sample size $n$ by $n = \frac{32}{\epsilon} \left[ i \log 2 + \log \frac{2}{\delta} \right]$. Suppose we draw an i.i.d. sample $S$ of size $n$ according to some unknown distribution $D$. 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 $\widehat{R}(h) \leq 3/4\epsilon$.

1. Assume that $R(h) \geq \epsilon$. Use the (multiplicative) Chernoff bounds to show that in that case $\Pr_{S \sim D^n}[h \text{ is accepted}] \leq \frac{\delta}{2^{i\epsilon}}$. 

2
Solution: By definition of acceptance,

$$\Pr[h \text{ is accepted}] = \Pr[\hat{R}_S(h) \leq 3/4\epsilon]$$

$$\leq \Pr[\hat{R}_S(h) \leq 3/4R(h)] \quad (R(h) \geq \epsilon)$$

$$\leq \exp \left(-\frac{n}{2}R(h)(1/4)^2\right) \quad \text{(Chernoff bound)}$$

$$= \exp \left(-\frac{R(h)}{\epsilon}\log \frac{2^{i+1}}{\delta}\right) \quad \text{(def. of } n)$$

$$= \exp \left(-\log \frac{2^{i+1}}{\delta}\right) = \frac{\delta}{2^{i+1}}. \quad (R(h) \geq \epsilon)$$

$$\square$$

2. Assume that $R(h) \leq \epsilon/2$. Use the (multiplicative) Chernoff bounds to show that in that case $\Pr_{S \sim D^n}[h \text{ is rejected}] \leq \frac{\delta^2}{i+1}$.

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

$$\Pr[h \text{ is rejected}] \leq \exp \left(-\frac{n\epsilon}{3/4}(1/2)^2\right) \quad \text{(Chernoff bound)}$$

$$= \exp \left(-\frac{4\epsilon}{3} \log \frac{2^{i+1}}{\delta}\right) \quad \text{(def. of } n)$$

$$\leq \exp \left(-\log \frac{2^{i+1}}{\delta}\right) = \frac{\delta}{2^{i+1}}. \quad (R(h) \geq \epsilon)$$

$$\square$$

3. Algorithm $B$ is defined as follows: we start with $i = 1$ and, at each round $i \geq 1$, we guess the parameter size $s$ to be $\tilde{s} = \lfloor 2^{(i-1)/\log \frac{\epsilon}{2}} \rfloor$. We draw a sample $S$ of size $n$ (which depends on $i$) to test the hypothesis $h_i$ returned by $\mathcal{A}$ when it is trained with a sample of size $S_A(\epsilon/2, 1/2, \tilde{s})$, that is the sample complexity of $\mathcal{A}$ for a required precision $\epsilon/2$, confidence $1/2$, and size $\tilde{s}$ (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. Show that if at iteration $i$, the estimate $\tilde{s}$ is larger than or equal to $s$, then $\Pr[h_i \text{ is accepted}] \geq 3/8$.

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

4. Show that the probability that \( B \) does not halt after \( j = \lceil \log \frac{2}{\delta} / \log \frac{s}{\delta} \rceil \) iterations with \( \tilde{s} \geq s \) is at most \( \delta/2 \).

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

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

**Solution:** By definition,
\[
\tilde{s} \geq s \iff \left\lfloor 2^{(i-1)/\log \frac{2}{\delta}} \right\rfloor \geq s \\
\iff 2^{(i-1)/\log \frac{2}{\delta}} \geq s \\
\iff \frac{i - 1}{\log \frac{2}{\delta}} \geq \log_2 s \\
\iff i \geq 1 + (\log_2 s) \log \frac{2}{\delta} \\
\iff i \geq \lceil 1 + (\log_2 s) \log \frac{2}{\delta} \rceil.
\]
\( \square \)

6. Show that with probability at least \( 1 - \delta \), algorithm \( B \) 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 \( \epsilon \).

**Solution:** In view of the two previous questions, with probability at least \( 1 - \delta/2 \), algorithm \( B \) halts after at most \( j' \) iterations. The probability that the hypothesis it returns be accepted while its error is greater than \( \epsilon \) 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 \( \epsilon \). \( \square \)

7. ([bonus question]) Show that the running time complexity of the algorithm is polynomial.