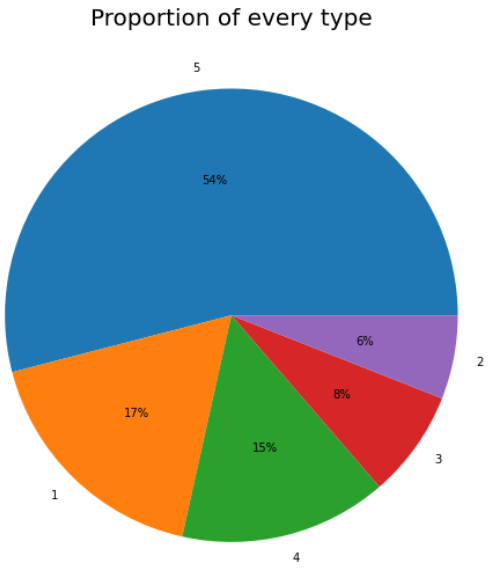

A. Consistent hypotheses

In the second lecture, we showed 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 $\mathcal{Z}$ be a finite set of m labeled points. 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 $\mathcal{Z}$, with high probability. (*Hint*: you can select an appropriate distribution $\mathcal{D}$ over $\mathcal{Z}$ and give a condition on $R(h)$ for h to be consistent.)

B. Oracle PAC learning

1. Learning unions of intervals. Give a PAC-learning algorithm for the concept class $\mathcal{C}_3$ formed by unions of three closed intervals, that is $[a, b] \cup [c, d] \cup [e, f]$, with $a, b, c, d, e, f \in \mathbb{R}$. 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 $p \geq 1$ 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?

2. Hypothesis testing. In the previous problem, the learning algorithm was given k as input.

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

   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 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 $n = \frac{32}{\epsilon}[i \log 2 + \log \frac{2}{\delta}]$. 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 $\widehat{R}(h) \leq 3/4\epsilon$.

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

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

(d) Algorithm $\mathcal{B}$ is defined as follows: we start with $i = 1$ and, at each round $i \geq 1$, we guess the parameter size s to be $\widetilde{s} = \lfloor 2^{(i-1)/\log \frac{2}{\delta}} \rfloor$. 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 $S_{\mathcal{A}}(\epsilon/2, 1/2, \widetilde{s})$, 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. Show that if at iteration i, the estimate $\widetilde{s}$ is larger than or equal to s, then $\mathbb{P}[h_i \text{ is accepted}] \geq 3/8$.

(e) 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 $\delta/2$.

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

(g) 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.