

HW2A: PAC learning

1 Written Exercises

1. Generalize the algorithm for the rectangle learning problem to D -dimensional space. In particular, suppose \mathcal{C} is the class of all D -dimensional axis aligned hyperrectangles (i.e., high-dimensional boxes), and \mathcal{H} is the same. Show that \mathcal{C} is efficiently PAC learnable using \mathcal{H} . In particular, show what the sample complexity of this algorithm is. How does this relate to our discussion of feature selection?
2. Consider our standard decision tree learning algorithm on D binary features.
 - (a) Suppose that the concept class is *any* binary function and we allow ourselves to build decision trees that are as deep as we need. (In the noise-free setting, we know that a decision tree can compute any binary function.) Use the Occam bound to show that this problem is PAC learnable. What is the sample complexity?
 - (b) Suppose we know that the concept class is limited to decision trees of maximum depth κ and we limit our hypothesis space to decision trees of maximum depth κ . (The depth of a decision tree is the maximum number of features used in any decision.) Can you use the Occam bound to show whether this problem is PAC learnable? Why or why not? What is the sample complexity? What is the sample complexity in terms of κ and D ?
3. **6350 Only:** Prove the Occam bound. In particular, consider a fixed but unknown concept class \mathcal{C} , distribution \mathcal{X} and concept $c \in \mathcal{C}$. Suppose we have N labeled examples from \mathcal{D} (labeled with c). Let \mathcal{L} be a polytime learning algorithm that outputs $h \in \mathcal{H}$ that is consistent with the training sample, and \mathcal{H} is *finite*. Show that \mathcal{L} PAC-learns \mathcal{C} using \mathcal{H} so long as $\log |\mathcal{H}| \leq bN\epsilon - \log(1/\delta)$ for some constant b .
4. Consider boosting decision stumps. I claimed in class that this learns a linear classifier. Show formally that this is true. (For simplicity, you may assume boolean features.) What advantages does boosting decision stumps have over, say, SVM learning?