

7.1 Agnostic Model

Last lecture we introduced the PAC Model:

Definition 1 (*The PAC Model*) Let \mathcal{C} be a concept class over X . We say that \mathcal{C} is PAC learnable if there exists an algorithm L with the following property: for every concept $c \in \mathcal{C}$, for every distribution \mathcal{D} on X , and for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, if L is given access to $EX(c, \mathcal{D})$ and inputs ϵ and δ , then with probability at least $1 - \delta$, L outputs a hypothesis concept $h \in \mathcal{C}$ satisfying $\text{error}(h) \leq \epsilon$, where $\text{error}(h) = \Pr_{x \in \mathcal{D}}[h(x) \neq c(x)]$.

One criticism of the PAC Model is that it is rare in the real world for a concept to perfectly fit a concept class. There may be anomalies or noise in the training examples that we may wish to tolerate. In this case, we may like to use a more general model of learning than the PAC Model. The Agnostic Model is a model of learning where we allow the target concept to have error on the training examples.

Definition 2 (*The Agnostic Model*) Let \mathcal{C} be a concept class over X . \mathcal{C} is learnable in the Agnostic Model if for every $f : X \rightarrow \{0, 1\}$, for every distribution \mathcal{D} on X , and for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, there exists a learner L such that if L is given access to $EX(f, \mathcal{D})$ and inputs ϵ and δ , then with probability at least $1 - \delta$, L outputs a hypothesis concept $h \in \mathcal{C}$ satisfying $\text{error}(h) \leq \epsilon + \text{OPT}$, where $\text{error}(h) = \Pr_{x \in \mathcal{D}}[h(x) \neq f(x)]$ and $\text{OPT} = \min_{c \in \mathcal{C}} \{\Pr_{x \in \mathcal{D}}[c(x) \neq f(x)]\}$.

Notice that the PAC Model is a special case of the Agnostic Model when $\text{OPT} = 0$ (Agnostic Model $>$ PAC Model). If a concept is learnable in the Agnostic Model, then it is also learnable in the PAC Model.

7.2 Occam's Razor

Last lecture, we showed how to prove that concept classes are efficiently PAC-learnable. Now we will show a general condition that implies PAC-learnability. Suppose we are given a concept class \mathcal{C} , where $|\mathcal{C}| < \infty$ (we will show how to handle the case where $|\mathcal{C}| = \infty$ when we discuss VC-dimension), instance space $X = \{0, 1\}$ over distribution \mathcal{D} , and m training examples $\{(x_1, c(x_1)), (x_2, c(x_2)), \dots, (x_m, c(x_m))\}$ for target concept c .

Definition 3 A hypothesis h is **consistent** with a set of training examples $S = \{(x_1, c(x_1)), (x_2, c(x_2)), \dots, (x_m, c(x_m))\}$ if it correctly labels all training examples i.e. $c(x) = h(x)$ for $x \in \{x_1, x_2, \dots, x_m\}$. That is, the empirical error of h on S is 0.

Theorem 1 (*Occam's Razor, Cardinality Version*) *If a learner L outputs a hypothesis $h \in \mathcal{C}$ such that h is consistent with m random samples, then for sufficiently large m , L is a PAC-learning algorithm. Such an algorithm is called an **Occam algorithm**.*

Proof: Our goal is to find m such that the probability that our learner L has found a hypothesis h that is consistent with the set of training examples with $error(h) \leq \epsilon$ is at least $1 - \delta$. Call a hypothesis h "bad" if $error(h) = \Pr_{x \in \mathcal{D}}[h(x) \neq c(x)] > \epsilon$. For some bad hypothesis h' , the probability that it is consistent with a random set of m examples is at most $(1 - \epsilon)^m$. The number of bad hypotheses is bounded by the cardinality of the concept class \mathcal{C} , so an upper bound on the probability of some bad hypothesis being consistent with the training examples is $|\mathcal{C}|(1 - \epsilon)^m$ by the union bound. Using the fact that $(1 - \epsilon) < e^{-\epsilon}$, we relax the upper bound to $|\mathcal{C}|e^{-\epsilon m}$. We need for $|\mathcal{C}|e^{-\epsilon m} \leq \delta$ to guarantee that we have found a hypothesis that is consistent with the training examples with $error(h) \leq \epsilon$. Taking logarithms and solving for m , we find that we need $m = \frac{1}{\epsilon}(\log |\mathcal{C}| + \log \frac{1}{\delta})$ training examples to guarantee that the hypothesis h found by L has $error(h) \leq \epsilon$ with probability $1 - \delta$. ■

We can use Occam's razor to prove the PAC learnability of a concept class when $|\mathcal{C}| < \infty$. If we can provide an Occam algorithm that always outputs $h \in \mathcal{C}$ such that h is consistent with any set of m training examples, we can use the equation for m to get the number of training examples sufficiently large for our algorithm to find a hypothesis h with $error(h) \leq \epsilon$ with probability at least $1 - \delta$. Here are a few examples of this approach.

Example 1 (*PAC-learning 1-decision lists*) *The cardinality of the concept class \mathcal{C} of 1-decision lists over n variables is given by $|\mathcal{C}| = 4^n$ since there are $2n$ basic blocks and each block can have output either 0 or 1. If we take $m = \frac{1}{\epsilon}(\log 4 \cdot n + \log \frac{1}{\delta})$ random examples using an algorithm that will output a hypothesis $h \in \mathcal{C}$ that is consistent with the set of m examples, then we will have proven the PAC learnability of 1-decision lists. Here is such an algorithm:*

Algorithm 1: Occam Algorithm for 1-decision lists

```

Initialize  $S$  with a set of  $m$  training examples, where  $m$  is calculated as above using the given
 $\delta$  and  $\epsilon$ 
Initialize the decision list to an empty list
while  $S \neq \emptyset$  do
    Find a basic block  $X$  such that when activated gives the correct label for all examples in  $S$ 
    Append building block  $X$  to the end of the decision list
    Remove from  $S$  all examples where building block  $X$  is activated
end

```

Example 2 (*PAC-learning disjunctions*) *We will use the Greedy Set Cover algorithm to PAC-learn disjunctions of length k . Below is a brief review of the Set Cover Problem and the Greedy Set Cover algorithm. More details can be found in Computational Learning Theory by Kearns and Vazirani, pages 38-42, and Algorithm Design by Kleinberg and Tardos, pages 612-617. We can cast learning disjunctions as a Set Cover Problem by letting U be the set of positive training examples and letting the subsets \mathcal{S}_i be set of positive training examples where literal i is positive. If the length of our target disjunction is of length k , then the Greedy Set Algorithm guarantees that our hypothesis h*

has length at most $k \log n$. By Occam's Razor, we can PAC-learn disjunctions of length k by using a random set of $m \approx O\left(\frac{k \log n}{\epsilon}\right)$ examples.

Definition 4 (Set Cover Problem) Given as input a collection \mathcal{S} of subsets of $U = \{1, \dots, m\}$, find a subcollection $\mathcal{T} \subseteq \mathcal{S}$ such that $|\mathcal{T}|$ is minimized and the sets in \mathcal{T} form a cover of U :

$$\bigcup_{t \in \mathcal{T}} t = U$$

Let $\text{opt}(\mathcal{S})$ denote the number of sets in a minimum cardinality cover. The Greedy Set Cover Algorithm below is guaranteed to find a cover of the elements of U using at most $\text{opt}(\mathcal{S}) \log m$ sets.

Algorithm 2: Greedy Set Cover

```

Initialize  $R = U$  and  $\mathcal{T} = \emptyset$ 
while  $R \neq \emptyset$  do
    Select set  $\mathcal{S}_i$  that maximizes  $|\mathcal{S}_i \cap R|$  and add  $\mathcal{S}_i$  to  $\mathcal{T}$ 
    Delete set  $\mathcal{S}_i$  from  $R$ 
end

```

Example 3 (PAC-learning k -decision lists) We might try to use a greedy algorithm that selects at each step the basic block that correctly labels the most training examples to add to the decision list, but this approach does not work. Finding a learner for k -decision lists that is consistent on set of random examples is an open problem.

There is a second version of Occam's Razor that can be used to prove PAC-learnability of a concept class. If an algorithm for learning the concept class satisfies the condition that the size of any hypothesis output by the algorithm is at most $(n \cdot \text{size}(c))^{\alpha} m^{\beta}$ and the hypothesis is consistent with any set of m random examples, where $\alpha > 0$, $0 < \beta < 1$, and $\text{size}(c)$ is the size of the smallest representation of c (we can assume $\text{size}(c)$ to be polynomial in n), then the concept class is PAC-learnable. The condition says that compression implies learning, since a set of training examples represented by $O(mn)$ bits is represented with fewer bits by the hypothesis output by the algorithm.

Theorem 2 Let L be an algorithm for learning a target concept $c \in \mathcal{C}$ such that L is consistent over any set S of m random training examples. Let \mathcal{D} be a distribution over instance space X . Fix a δ and ϵ such that $0 < \delta < 1/2$ and $0 < \epsilon < 1/2$. Assume that there exists $\alpha > 0$ and $0 < \beta < 1$ such that for any random m examples, L outputs a hypothesis h such that $\text{size}(h) \leq (n \cdot \text{size}(c))^{\alpha} m^{\beta}$ where $\text{size}(c)$ is the size of the smallest representation of c . If L is given a set S of m random examples where $m \geq 2 \cdot \left(\frac{1}{\epsilon} \log \frac{1}{\delta} + \left(\frac{(n \cdot \text{size}(c))^{\alpha}}{\epsilon} \right)^{\frac{1}{1-\beta}} \right)$, then \mathcal{C} PAC-learnable.

Proof: The proof is similar to the cardinality version of Occam's razor, except that we use $\text{size}(h)$ to bound the size of the hypothesis space \mathcal{H} . Again, for some bad hypothesis h' , the probability that it is consistent with a random set of m examples is at most $(1 - \epsilon)^m$. Since $\text{size}(h') \leq (n \cdot \text{size}(c))^{\alpha} m^{\beta}$, an upper bound on the size of the hypothesis space is $|\mathcal{H}| \leq 2^{(n \cdot \text{size}(c))^{\alpha} m^{\beta}}$. Thus, the probability that some bad hypothesis is consistent is at most $2^{(n \cdot \text{size}(c))^{\alpha} m^{\beta}} (1 - \epsilon)^m$ by the union bound. Again,

using $(1 - \epsilon) < e^{-\epsilon}$ we relax the bound to $2^{(n \cdot \text{size}(c))^\alpha m^\beta} e^{-\epsilon m}$. We need for $2^{(n \cdot \text{size}(c))^\alpha m^\beta} e^{-\epsilon m} \leq \delta$ to guarantee that we have found a hypothesis that is consistent with the training examples with $\text{error}(h) \leq \epsilon$ with probability at least $1 - \delta$. Taking logarithms and solving for m we get $m \geq \frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{1}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta$. If $\frac{1}{\epsilon} \log \frac{1}{\delta} \geq \frac{1}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta$, then $\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{1}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta \leq \frac{2}{\epsilon} \log \frac{1}{\delta}$. Else, $\frac{1}{\epsilon} \log \frac{1}{\delta} < \frac{1}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta$, and solving $m \geq \frac{2}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta$, we get $\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{1}{\epsilon} (n \cdot \text{size}(c))^\alpha m^\beta < 2 \cdot \left(\frac{(n \cdot \text{size}(c))^\alpha m^\beta}{\epsilon} \right)^{\frac{1}{1-\beta}}$. In either case, we can choose $m \geq 2 \cdot \left(\frac{1}{\epsilon} \log \frac{1}{\delta} + \left(\frac{(n \cdot \text{size}(c))^\alpha}{\epsilon} \right)^{\frac{1}{1-\beta}} \right)$ to guarantee PAC-learnability of \mathcal{C} using L . ■