CSC 482A: Problem set 1: Due by 7:00pm Friday, October 4

Swapnil Daxini (V00861672)

October 5, 2019

1. We want to show that the class $\mathcal{C}$ of concentric circles is PAC-learnable. Assuming realizability, we define the radius of target concept c_{r^*} to be r^*. For a given training set $\mathcal{T}$, our algorithm learns a concept with radius r such that $r < r^*$. This is the case as our training set $\mathcal{T}$ is a subset of point within the radius r^*. Now, lets define the learned concept $\hat{f}$ with the largest radius within the class that is consistent with $\mathcal{T}$ to be $\hat{r}$. Furthermore, we define the probability of a point being between r^* and $\hat{r}$ to be ϵ:
$$\hat{r} = \inf_{c_r \in \mathcal{C}} r : P(r \leq \|x\| \leq r^*) < \epsilon \tag{1}$$

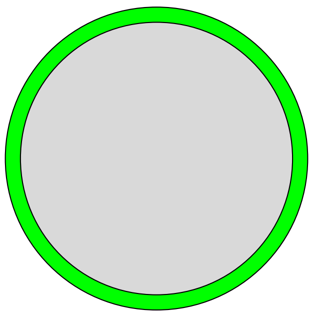

Figure 1: The overarching green circle represents the target concept radius while the grey radius is the concept with the lowest risk based on the training set. We define the annulus between the two circle as A

Now, we define the annulus between a radius of learned concept c_r and r^* to be A. Given our previous definition, we know that:
$$P(A) \geq \epsilon \tag{2}$$

For our algorithm, lets consider the worst case where our risk is greater than ϵ. In this case, we have that $\mathcal{T}$ has no points within A, thus $T \cap A = \emptyset$. The probability that is the case for one sample is given by $(1-\epsilon)$. Thus the probability for n samples is given by:
$$P(R(\hat{f}) > \epsilon) = P(T \cap A = \emptyset) \leq (1-\epsilon)^n \tag{3}$$

We can simplify that to
$$P(R(\hat{f}) \geq \epsilon) \leq (1-\epsilon)^n \leq e^{-n\epsilon} = \delta \tag{4}$$

Thus we get that:
$$n \geq \frac{1}{\epsilon} \log(\frac{1}{\delta}) \tag{5}$$

Therefore, the concept class $\mathcal{C}$ is PAC-learnable.

2. We need to devise an efficient mistake bound learner for k-term DNF. From class, we know that although the sample complexity of k-term DNF is polynomial, the class can not learned efficiently unless RP = NP. Thus, for this question, we can convert k-term DNF into k-term CNF as k-term CNF is more general. For any k-term DNF, its k-term CNF is given by:

$$\bigvee_{i=1}^{k} a_i(x_1) \wedge ... \wedge a_i(x_d) = \bigwedge_{i_1,...,i_k=1}^{d} a_i(x_{i_1}) \vee ... \vee a_i(x_{i_k}) \tag{6}$$

Now we have to provide a learner to efficient k-term CNF. We know that we can learn class of conjunctions with the mistake bound model, thus we need to convert our k-term CNF to strings of conjunctions. In order to do that, lets consider each disjunctive clause T_i of at most d boolean literals:

$$\bigwedge_{i_1,...,i_k=1}^{d} a_i(x_{i_1}) \vee ... \vee a_i(x_{i_k}) = \bigwedge_{i=1}^{k} T_i \tag{7}$$

We can now convert the k-term CNF to conjunctions:

$$\bigwedge_{i=1}^{k} T_i = \bigwedge_{x_1 \in T_1,...,x_k \in T_k} (x_1 \wedge x_2 \wedge ... \wedge x_k) \tag{8}$$

Now that we have just a string of conjunctions, we can apply our mistake bound learner we had used in class. We begin the algorithm with the hypothesis that includes all possible combinations (I will get to the size of this later), and whenever we make an incorrect prediction for a positive label, we remove the string of conjunctions which evaluate to 0.

Lets now consider the runtime and mistake bound for this learner. When we first transform k-term DNF to k-term CNF, the number of new variables we have is $O(d^k)$. In order to learn k-term CNF, we have that all the possible combinations of boolean literals is on the order of $3^{O(d^k)}$. We have 3 as each literal can be positive, its negation, or not be included. Thus, assuming the worst case, we have that the mistake bound is:

$$m \geq \log 3^{O(d^k)} = d^k \log 3 \tag{9}$$

We have that the mistake bound is polynomial in d.

For the runtime, we have that $|H| = 3^{O(d^k)}$. Thus, our runtime is

$$O(d^k \log 3) \tag{10}$$

This is polynomial in d.

3. a) For our problem, we have that:

$$Pr(\tilde{Y} = 0|X) = \mathbb{1}[c(X) = 0](1 - \alpha(X)) + \mathbb{1}[c(X) = 1]\alpha(X) \tag{11}$$

and

$$Pr(\tilde{Y} = 1|X) = \mathbb{1}[c(X) = 1](1 - \alpha(X)) + \mathbb{1}[c(X) = 0]\alpha(X) \tag{12}$$

Thus the optimal Bayes classifier for this problem will be $f_{bayes} = \mathbb{1}[Pr(\tilde{Y} = 1|x) \geq \frac{1}{2}]$:

$$f_{bayes} = \mathbb{1}[Pr(\mathbb{1}[c(X) = 1](1 - \alpha(X)) + \mathbb{1}[c(X) = 0]\alpha(X)) \geq \frac{1}{2}] \tag{13}$$

b) $\alpha = 1/4$
Using our previously found result, we have that the Bayes classifier is:

$$f_{bayes} = \mathbb{1}[Pr(\frac{3}{4}\mathbb{1}[c(X) = 1] + \frac{1}{4}\mathbb{1}[c(X) = 0]) \geq \frac{1}{2}] \tag{14}$$

c) In our case, the Bayes risk is the expected value of the noise. Given our noise is constant of all $x \in \mathcal{X}$, we have the at our Bayes risk is equal to $R^* = \frac{1}{4}$

d) The risk is given by

$$R(f_\epsilon) = \mathbb{E}[\mathbb{1}\{f(X) \neq \tilde{Y}\}] \tag{15}$$

In our case, we have that

$$R(f_\epsilon) = \epsilon\frac{3}{4} + (1 - \epsilon)\frac{1}{4} = \frac{1}{4} + \frac{\epsilon}{2} \tag{16}$$

Thus the total risk is given by the cases when the the hypothesis is right and noise flips the label or when the hypothesis is wrong and noise does not affect the label.

e) Lets consider the hypothesis from last part where $Pr(c_\epsilon(X) \neq c(X)) = \epsilon$. We found that the risk for this case is $R(f_\epsilon) = \frac{1}{4} + \frac{\epsilon}{2}$.

We can now consider a hypothesis such that $R(\hat{f}) < \frac{1}{4} + \frac{\epsilon}{2}$ but $Pr(c_{\hat{f}}(X) \neq c(X)) \leq \epsilon$.
Thus, we have

$$Pr(R(\hat{f}) < (\frac{1}{4} + \frac{\epsilon}{2}) \wedge Pr(c_{\hat{f}}(X) \neq c(X)) \leq \epsilon) \tag{17}$$

(I was stuck and found something online about Hoeffding bounds)
This probability can be upper bound using a Hoeffding bound:

$$Pr(R(\hat{f}) < (\frac{1}{4} + \frac{\epsilon}{2}) \wedge Pr(c_{\hat{f}}(X) \neq c(X)) \leq \epsilon) \leq e^{-2n(\frac{1}{4}+\frac{\epsilon}{2})} \tag{18}$$

Thus we have that

$$n \geq \frac{\log\frac{1}{\delta}}{2(\frac{1}{4} + \frac{\epsilon}{2})^2} \tag{19}$$

Thus this is PAC-learnable. This algorithm works as it takes into account how many more samples are need due to the noise present in the system. It accounts for the bad hypothesis that could be seen as good hypothesis due to the flipping of the label by the noise.