COS511 HW1

Pranjit Kalita
Consulted with - Nikunj Saunshi

February 27, 2017

Ex. 1
Here, we will use the Fundamental Theorem of Learning and VC Dimension of the hypothesis class H.

Step 1: Calculate the VC Dimension of H. Let us calculate the possibility of shattering the set containing only one point $c1 \in X$.
Here, since for any $h \in H$,
$h_r(c1) = 0$ and $h_r(c1) = 1$ is possible (since it is only dependent on what r we choose), hence the VC dimension(H) >= 1 (atleast 1).

Step 2: Prove that VC Dimension of H < 2. Let us calculate the possibility of shattering the set containing two points $c1, c2 \in X$.
Here, $c1 <= c2$. For $h_r(c1) = 0$ and $h_r(c2) = 0$, we see that it is possible since for any h_r, $r >= c2$ (implicitly $r > c1$). Now, let us see if $h_r(c1) = 1$ and $h_r(c2) = 0$ is possible. We find that since $c1 <= c2 <= r$, the above is just not consistent and therefore possible according to the concentric nature of hypothesis class defined. Hence, VC dimension(H) < 2 (it is very easy to generaly this to points > 2 by similar reasoning).

Step 3: From Steps 1 2, VC Dimension(H) = 1. Using Fundamental Theorem of Learning acc. to Notes' 3.2.1,
$m(\epsilon, \delta) = O(d * log(1/\delta)/\epsilon)$

Substituting for d = VC dimension(H) = 1, we have the given proof, and from Fundamental Theorem of Learning, we know that this class of hypothesis functions H is PAC Learnable.

Ex. 2
A/Q, **C** is an agnostic PAC learnable class. Therefore, in order to effectively solve this problem, we first have to bring in the definition of agnostic PAC.
We know, for agnostic PAC learnability, error <= ϵ. If an algorithm A returns a hypothesis h, $err(h_A) <= \epsilon$.
Now, we know that there is a noise error in each sampling of $x \in X$.
Therefore, for each h_A returned by A, there is another innate error e beyond ϵ.
Thus, our new equation for error in this setting looks something like this:
$err(h_A) <= e + \epsilon$

Now, let us calculate this new error e. The noise will create an error only when -

$\hat{h}(x) \neq h(x)$.
This will happen only when $\hat{h}(x) = 0$ and $h(x) = 1$, or $\hat{h}(x) = 1$ and $h(x) = 0$.

Thus, for any given h(x), we essentially have to find that only that binary classifier is returned by $\hat{h}(x)$ which is of opposite value. Thus, that probability is $\epsilon_0/2$, since for the other $\epsilon_0/2$, it will have returned the same binary as h(x) thereby not factoring into the error equation.

Thus, $e = \epsilon_0/2$.
Therefore, $err(h_A) <= \epsilon_0/2 + \epsilon$.

The rest of it, i.e, probability of choosing a representative sample being $1 - \delta$, with sample complexity of polynomial in $1/\epsilon$, $log(1/\delta)$, comes from the definition of C being an agnostic PAC learnable class.

Hence proved.

Ex. 3
$x_1, x_2, ..., x_k$ are independent r.v's each receiving values {-1, 1} w.p 1/2. $X = \sum_{i=1}^{k} x_i$.
Thus, X is a r.v.
Also, A/Q, $\lambda >= 0$.
(1)
Now,X>=t
$=> \lambda * X >= \lambda * t$
$=> e^{(\lambda * X)} >= e^{(\lambda * t)}$

Now, we have the following,
$Pr(\mathrm{e}^{(\lambda * X)} >= e^{(\lambda * t)})$.

Acc. to Markov's inequality,
Pr(X>=a)<=E(X)/a
Now, our RHS of $\Pr(e^{(\lambda * X)} >= e^{(\lambda * t)})$ becomes -
<= $\mathrm{E}(e^{\lambda * X})/e^{\lambda * t}$

Now, we will expand $\mathrm{E}(e^{\lambda * X})$
$\Pr(e^{(\lambda * X)} >= e^{(\lambda * t)}) <= \mathrm{E}(e^{\lambda * X})/e^{\lambda * t}$
$=> \Pr(e^{(\lambda * X)} >= e^{(\lambda * t)}) <= e^{-\lambda * t} * \mathrm{E}(\sum_{i=1}^{k} e^{\lambda * x_i})$
$=> \Pr(e^{(\lambda * X)} >= e^{(\lambda * t)}) <= e^{-\lambda * t} * \mathrm{E}(e^{\lambda * x_1 + \lambda * x_2 + \lambda * x_3 + ... + \lambda * x_k})$
$=> \Pr(e^{(\lambda * X)} >= e^{(\lambda * t)}) <= e^{-\lambda * t} * \mathrm{E}(e^{\lambda * x_1} * e^{\lambda * x_2} * e^{\lambda * x_3} * ..... * e^{\lambda * x_k})$
$=> \Pr(e^{(\lambda * X)} >= e^{(\lambda * t)}) <= e^{-\lambda * t} * \prod_{i=1}^{k} E(\mathrm{e}^{\lambda * x_k})$

Now, since, $\mathrm{E}(e^{\lambda * x_k}) = (e^{\lambda}/2 + e^{-\lambda}/2)$ (since -1 and +1 have probabilities 1/2 for each independent x_i), substituting this in the above inequality yields -
$\Pr(e^{(\lambda * X)} >= e^{(\lambda * t)}) <= e^{-\lambda * t} * (\mathrm{e}^{\lambda}/2 + e^{-\lambda}/2)^k$

Hence proved.

(2)
Acc. to Taylor's Theorem,
$e^x = \sum_{n=0}^{\infty} x^n/n!$.

e^{λ}=$(\lambda/1!) + (\lambda)^2/2! + (\lambda)^3/3! + ...(a)$
$e^{-\lambda}$=$-(\lambda/1!) + (\lambda)^2/2! - (\lambda)^3/3! + ...(b)$

Combining (a) and (b), and cancaling out the odd powers of λ, we have -
$((e^{\lambda}/2) + (e^{-\lambda}/2))$=$(\lambda)^2/2! + (\lambda)^4/4! + (\lambda)^6/6! + (\lambda)^8/8! + ...$ $^{(c)}$

Now, in RHS, we have using Taylor's Theorem -
$e^{\lambda^2/2}$=$(\lambda^2/2 * 1!) + (\lambda^4/4 * 2!) + (\lambda^6/8 * 3!) + (\lambda^8/16 * 4!) + ...$ $^{(d)}$

Comparing equations (c) and (d), due to the denominators in (c) being larger than their corresponding denomitors in (d) relative to each power of $\lambda, hence -$

$((e^{\lambda}/2) + (e^{-\lambda}/2)) <= e^{\lambda^2/2}$

Hence proved.

(3)
From parts 1 and 2,
Pr[X >= t] <= $e^{-\lambda * t}$ * $(e^{\lambda}/2 + e^{-\lambda}/2)^k$ (from part 1)
=> Pr[X >= t] <= $e^{-\lambda * t}$ * $e^{\lambda^2 * k/2}$ (from part 2)
=> Pr[X >= t] <= $e^{(\lambda^2 * k - 2 * \lambda * t)/2}$ (Equation (i))
Now, we know that this equation (i) holds true for all λ>= 0.
Now, if it holds true for the lowest λ, then it will hold true for every λ.
Let us differentiate $e^{(\lambda^2 * k - 2 * \lambda * t)/2}$ w.r.t λ and equate to 0.
=> $\frac{d(e^{(\lambda^2 * k - 2 * \lambda * t)/2})}{d\lambda} = 0$
=> $2 * \lambda * k - 2 * t$=0
=> λ=t/k
Substituting this value of λ to the inequality proven by Part (1), we get -
Pr[X >= t] <= $e^{((t^2/k^2) * k - 2 * (t/k) * t)/2}$
=> Pr[X >= t]<= $e^{-t^2/2k}$
Hence proved.

Ex. 4
(a)
Statement: For a finite domain X, if a concept h is PAC learnable by an algorithm with a hypothesis class **H**, then $h \in \mathbf{H}$.
We will use proof by contradiction.
Let us assume that for a finite domain X, if a concept h is PAC learnable by an algorithm with a hypothesis class **H**, then $h \notin \mathbf{H}$.
Then, acc. to the definition of PAC learnability,
err(h)<= $\min_{h* \in \mathbf{H}} err(h*) + \epsilon$.

Since we know that for any algorithm returning hypothesis h, the error above $\min_{h* \in \mathbf{H}} err(h*)$ has to be at most ϵ, therefore we will see if this is consistent under our beginning assumption.

Since $h \notin \mathbf{H}$, if we assume a uniform distribution **D**, h will have an error of

at least 1/n.
Thus,
$err(h) <= \min_{h* \in \mathbf{H}} err(h*) + (1/n)$.

Thus, maximum allowable error is (1/n).
Thus, for PAC learnability,
$(1/n) <= \epsilon$
Thus, h is not PAC learnable for all ϵ, which violates the definition of PAC learnability. Hence, our premise was wrong. Thus, due to proof by contradiction, $h \in \mathbf{H}$.
Hence proved.
(b)
Statement: For an infinite domain X, if a concept h is PAC learnable by an algorithm with a hypothesis class **H**, then $h \in \mathbf{H}$.
We will use proof by example.
Let a concept h_r be such that it returns 1 for x >= r, and returns 0 for x < r.

Let the hypothesis class be class of functions that only returns 1 for every $x \in \mathbf{X}$.

If we we show that h can learn **H** under conditions of PAC learnability, then $h \notin \mathbf{H}$, and therefore we will have disproven the given statement.

If we set for the given h_r a particular point r on the real line, then -
$\text{err}(h_r) = \Pr(h_r(\text{x}) \neq 1) = Pr(X_{¡}=r) = \epsilon$, which is the error allowed. Now we will try to prove that this error is within bounds and consistent with PAC learnability.

Now, this is the good case if we get error within ϵ.

Pr(bad case) = $(1 - \epsilon)$, for each of the sample instances.

Thus, for m instances, Pr(bad case) = $(1 - \epsilon)^m$
=> Pr(bad case) = $e^{-\epsilon * m} <= \delta$
Therefore, Pr(good case) >= $(1 - \epsilon)$
Thus, we see, $\text{err}(h_r) <= \epsilon$, w.p. $(1 - \delta)$.

We see that this is the definition of PAC learnability, and we will see what the value of m (sample complexity) is so that this holds.
$e^{-\epsilon * m} <= \delta$
=> $(\epsilon * m) >= log(1/\delta)$
=> m >= $log(1/\delta)/\epsilon$

Therefore, this is consistent with PAC learnability, with $h \notin \mathbf{H}$, for an infinite sample size **X**.

Hence, the given statement does not hold.