Theoretical Machine Learning - COS 511

Homework Assignment 2

Due Date: 15 Mar 2017, till 22:00

(1) **Solve 4 out of the following 5 problems.**

(2) **Consulting other students from this course is allowed. In this case - clearly state whom you consulted with for each problem separately.**

(3) **Searching the internet or literature for solutions, other than the course lecture notes, is NOT allowed.**

(4) **All problems are weighted equally at 10 points each. Indicate on your problem set which four problems you choose to solve. Feel free to write down solutions for the other two as well, but your homework grade will only depend upon the four you mark to be graded.**

Ex. 1:

Recall that the growth function $\tau_{\mathcal{F}}(m)$ of a class of functions $\mathcal{F} \subseteq \{f : X \to Y\}$ is defined as

$$\tau_{\mathcal{F}}(m) = \max_{|S|=m} |\{f : S \to Y : \exists f' \in \mathcal{F}, f'(s) = f(s) \forall s \in S\}|$$

- Prove that If $\mathcal{F} = \mathcal{F}_1 \circ \mathcal{F}_2 \circ \cdots \circ \mathcal{F}_t = \{f_1 \circ f_2 \circ \cdots \circ f_t : f_i \in \mathcal{F}_i\}$ then

$$\tau_{\mathcal{F}}(m) \leq \prod_{i=1}^{t} \tau_{\mathcal{F}_t}(m)$$

- suppose $Y = Y_1 \times, \ldots \times Y_t$ and each $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_t$ where $f \in \mathcal{F}$ can be written as $f = (f_1, \ldots, f_t)$. show that

$$\tau_{\mathcal{F}}(m) \leq \prod_{i=1}^{t} \tau_{\mathcal{F}_t}(m)$$

Ex. 2:
Let $\mathcal{H}_1, \ldots, \mathcal{H}_m$ be hypotheses class of dimension VC-dim$(H_i) = d_i$ show that

$$\text{VC-dim}(\cup_{i=1}^m \mathcal{H}_i) = O(d \log d)$$

where $d = \sum d_i$.

Ex. 3[Structural Risk Minimization]
In the next exercise we are going to develop a strategy to learn classes of infinite dimensional VC–dimension. We consider a hypothesis class that can be written as $\mathcal{H} = \cup_{n\in\mathbb{N}} \mathcal{H}_n$, where VC-dim$(\mathcal{H}_n) \le n$ for all $n \in \mathbb{N}$. We next define the notion of "non–uniform" learnability:

Definition 0.1 (Non-Uniform Learnability)**.** *A class $\mathcal{H}$ is said to be non-uniform learnable if there is an algorithm A and a function $m(0,1) \times \mathcal{H} \to \mathbb{N}$ such that, for every $\epsilon, \delta \in (0,1)$ and $h \in \mathcal{H}$, if $m > m(\epsilon, \delta, h)$ then with probability $(1-\delta)$ over a sample S drawn i.i.d we have that*

$$err(h_S^A) \le err(h) + \epsilon$$

In other words, if we fix an hypothesis $h \in \mathcal{H}$ then after enough examples, our performance will be better then the performance of h.

- For each $n \in \mathbb{N}$ set

  $$\epsilon_n(m) = \Omega\left(\sqrt{\frac{n}{m}\log\frac{2^n}{\delta}}\right)$$

  Show that with probability at least $1-\delta$ (over a sample S of size m drawn i.i.d) we have: For every $n \in \mathbb{N}$ and for every $h \in \mathcal{H}_n$

  $$\text{err}(h) \le \text{err}_S(h) + \epsilon_n(m)$$

- Consider an algorithm A that given a sample S of size m chooses $n \in \mathbb{N}$ and $h_n \in \mathcal{H}$ that minimize the term

  $$\text{err}_S(h) + \epsilon_n(m)$$

  Show that A non–uniformly learns $\mathcal{H}$.

Ex. 4[Boosting the confidence]

Let A be an algorithm that receives a sample S of size $m(\epsilon)$ an returns with probability at least $2/3$ (over the sample S drawn i.i.d) a hypothesis h such that

$$\mathrm{err}(h) \leq \min_{h^* \in \mathcal{H}} \mathrm{err}(h) + \epsilon$$

Use this algorithm to show that $\mathcal{H}$ is learnable. (In other words, if a class is learnable for a fixed confidence $\delta = 1/3$ then it is learnable).

hint:

Consider an algorithm A' that receives a sample S' of size

$$|S'| = \Omega\left(m(\epsilon)\log 1/\delta + \frac{\log\log 1/\delta}{\epsilon^2}\log 1/\delta\right)$$

and does the follow: the algorithm partition the sample $S' = S_0 \cup \cup_{i=1}^{|\log 1/\delta|} S_i$ samples,

$$|S_0| = \frac{\log\log 1/\delta}{\epsilon^2}\log 1/\delta \;\text{ and } |S_i| = m(\epsilon)\; \forall i = 1, \ldots, |\log 1/\delta|$$

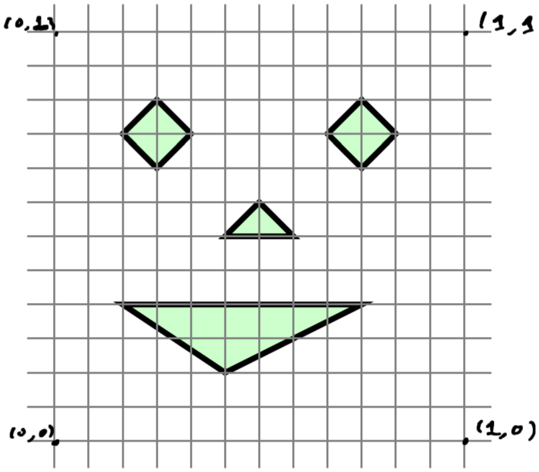
each of size m (for simplicity assume $\log 1/\delta$ is an integer). Then the algorithm runs algorithm A on each sample, to receive hypotheses $h_{S_1}, \ldots, h_{S_{\log 1/\delta|}}$. the algorithm then returns:

$$h^* = \arg\min_i \mathrm{err}_{S_0}(h_{S_i})$$

Ex. 5[Happy Face]

Consider neural networks with with two input nodes that can attain real values between $[0, 1]$: i.e. the input of the networks is the domain $\chi = [0, 1]^2$. Implement some neural network in $\mathcal{N}_{(V,E),\sigma_{\mathrm{sgn}}}$ for some (V, E)[1], that realizes the following classification (i.e. green blobs get positive sign, white areas are negative). Define the architecture as you wish:

[1] where $|V^{(0)}| = 2$


(0,1)
(1,1
(0,0)
(1,0)