COS 511 - Theoretical Machine Learning
Final Exam

Instructions

1. You have at most 48 hours from receipt of this exam it to the professor/TAs, in hand, or by email.
2. The **only** literature allowed for this exam are lecture notes, any notes taken during class, and the main course textbooks.
3. **Collaboration of any kind is not allowed.**
4. You are required to keep the contents of this exam confidential until all students have completed the course.
5. Please write in your handwriting on the first page of the exam the honor code pledge and sign it. Put your initials on every page. Full statement and explanation on the honor code:
https://www.princeton.edu/honor/constitution/
6. Please do not post questions on piazza. Only by email, to all professor/TA simultaneously.

Question 1

A function $h : \{0,1\}^n \to \{0,1\}$ is symmetric if its value is uniquely determined by the number of 1's in the input. Let H be the class of all symmetric functions. Determine the VC-dimension of H. Conclude lower and upper bounds on the sample complexity of any (agnostic) PAC learning algorithm.

Question 2

Consider the setting of expert advice with m experts. Given a learning algorithm, assume that at each iteration, an adversary is guaranteed to choose a loss function $\mathbf{g}_t$ $\|\mathbf{g}_t\| \leq 1$ such that $p_t \cdot \mathbf{g}_t > \frac{1}{2} + \gamma$ where $p_t \in \Delta_m$ is the choice of the algorithm.

Show that for any online learning algorithm with guaranteed regret $\text{Regret}_T = O(\sqrt{T \log m})$, for $T > \frac{1}{\gamma^2} \log m$ we will have that for every i:

$$\sum_{t=1}^{T} \mathbf{g}_t(i) > \frac{T}{2}.$$

Use this to derive a Boosting algorithm, compare the algorithm you received to AdaBoost, learned in class:

hint/spoiler:

Given a distribution p_t over sample $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ a weak learner returns a hypothesis such that $\sum p_t(i)\frac{|h_t(i)-y_i||}{2} < \frac{1}{2} - \gamma$. Set $\mathbf{g}_t(i) = 1 - \frac{|h_t(i)-y_i|}{2}$

Question 3

Consider the binary classification setting. Denote S_m the set of all samples of size m. Namely $S_m = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \subseteq \mathcal{X} \times \mathcal{Y}\}$.

A d–size compression scheme for a *binary* learning problem $(\mathcal{X}, \mathcal{H}, \ell_{0,1})$ is defined as a pair of functions, κ and ρ:

$$\kappa : \cup_{m=1}^{\infty} S_m \to \cup_{k=1}^{d} S_k, \quad \rho : \cup_{k=1}^{d} S_k \to \{f : \mathcal{X} \to \mathcal{Y}\}$$

κ is a compression function that receives finite sample $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ of arbitrary finite length m and returns a subsample of size at most d such that $\kappa(S) \subseteq S$. The reconstruction function ρ is defined as a function from sample of size at most d to classifier f.

The pair (κ, ρ) needs to satisfy the following inorder to be called a d-size compression scheme:

For every finite sample $S_m = \{(\mathbf{x}_i, h(\mathbf{x}_i)\}_{i=1}^{m}$ labeled by some $h \in H$, if $f_{S_m} = \rho \circ \kappa(S_m)$ then

$$f_{S_m}(\mathbf{x}_i) = h(\mathbf{x}_i).$$

Show that if $(\mathcal{X}, \mathcal{H}, \ell_{0,1})$ has a compression scheme of size d, then the VC-dim$(\mathcal{H}) = \tilde{O}(d)$.

Hints and guidelines

1. Let $f_{1,2,\ldots,d} = \rho((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_d, y_d))$. Bound the probability (over a sample S of size m) that

$$P\left((\text{err}(f_{1,2,\ldots,d}) > \epsilon) \wedge (\text{err}_S(f_{1,2,\ldots,d}) = 0)\right)$$

2. For every subsequence: $\{(\mathbf{x}_{i_1}, y_{i_1}), \ldots, (\mathbf{x}_{i_k}, y_{i_k})\}$ such that $k \leq d$, similarly define and show a bound on the error of $f_{i_1,\ldots,i_k}$, and apply union bound to bound:

$$P\left((\text{err}(\rho \circ \kappa(S)) > \epsilon) \wedge (\text{err}_S(\rho \circ \kappa(S) = 0)\right)$$

3. Conclude that the class is learnable with sample size $O(\frac{d}{\epsilon} \log 1/\delta)$ in the realizable setting.

4. Conclude that the class has VC dimension of order d (up to logarithmic factors).

Question 4

In this question we will build an algorithm for classification of infected files. We will represent a file as a sequence of size d written in some finite alpha-bet (say English Letters). There exists a sequence (e.g. “tihiiamavirus”) such that if the sequence of letters appear in the string then the file is infected. Needless to say the sequence is unkown.

- Define the hypothesis class to be learned, and give a bound on the size of the sample complexity needed inorder to learn the problem. (Here classified sample means a sequence of files with the labeling "infected/not infected".
- Construct an embedding of the domain into a linear space, where the hypothesis class to be learned becomes a subset of the class of linear separators.
- Can you propose an efficient learning algorithm for the problem, in the realizable case? In particular, both sample complexity and time complexity need be polynomial in d