COS 511: Theoretical Machine Learning

Homework #4 — Due:
Rademacher & boosting — March 25, 2019

Problem 1

[15] Let $\mathcal{F}_1, \ldots, \mathcal{F}_n$ be families of real-valued functions on some space $\mathcal{Z}$, and let $a_1, \ldots, a_n$ be arbitrary (fixed) real numbers. Let $\mathcal{G}$ be the class of all functions g of the form

$$g(z) = \sum_{i=1}^{n} a_i f_i(z)$$

where $f_i \in \mathcal{F}_i$ for $i = 1, \ldots, n$. For any sample S, find $\hat{\mathcal{R}}_S(\mathcal{G})$ exactly in terms of $a_1, \ldots, a_n$, and $\hat{\mathcal{R}}_S(\mathcal{F}_1), \ldots, \hat{\mathcal{R}}_S(\mathcal{F}_n)$. Be sure to justify your answer.

Problem 2

[15] Suppose, in the usual boosting set-up, that the weak learning condition is guaranteed to hold so that $\epsilon_t \leq \frac{1}{2} - \gamma$ for some $\gamma > 0$ which is *known* before boosting begins. Describe a modified version of AdaBoost whose final classifier is a simple (unweighted) majority vote, and show that its training error is at most $(1 - 4\gamma^2)^{T/2}$.

Problem 3

Let $\mathcal{X}_n = \{0, 1\}^n$, and let $\mathcal{G}_n$ be any class of boolean functions $g : \mathcal{X}_n \to \{-1, +1\}$. In this problem, we will see, roughly speaking, that if a function f can be written as a majority vote of polynomially many functions in $\mathcal{G}_n$, then under any distribution, f can be weakly approximated by some function in $\mathcal{G}_n$. But if f cannot be so written as a majority vote, then there exists some "hard" distribution under which f cannot be approximated by *any* function in $\mathcal{G}_n$.

Let $\mathcal{M}_{n,k}$ be the class of all boolean functions that can be written as a simple majority vote of k (not necessarily distinct) functions in $\mathcal{G}_n$; that is, $\mathcal{M}_{n,k}$ consists of all functions f of the form

$$f(x) = \text{sign}\left(\sum_{j=1}^{k} g_j(x)\right)$$

for some $g_1, \ldots, g_k \in \mathcal{G}_n$. Assume k is odd.

a. [15] Show that if $f \in \mathcal{M}_{n,k}$ then for all distributions D on $\mathcal{X}_n$, there exists a function $g \in \mathcal{G}_n$ for which

$$\text{Pr}_{x \sim D}\left[f(x) \neq g(x)\right] \leq \frac{1}{2} - \frac{1}{2k}.$$

b. [15] Show that if $f \notin \mathcal{M}_{n,k}$ then there exists a distribution D on $\mathcal{X}_n$ such that for every $g \in \mathcal{G}_n$,

$$\text{Pr}_{x \sim D}\left[f(x) \neq g(x)\right] > \frac{1}{2} - \sqrt{\frac{n \ln 2}{2k}}.$$

Problem 4 – Optional (Extra Credit)

[15] Consider the following "mini" boosting algorithm which runs for exactly three rounds:

- Given training data as in AdaBoost, let D_1, h_1, ϵ_1, and D_2, h_2, ϵ_2 be computed exactly as in AdaBoost on the first two rounds.
- Compute, for $i = 1, \ldots, m$:

$$D_3(i) = \begin{cases} D_1(i)/\mathcal{Z} & \text{if } h_1(x_i) \neq h_2(x_i) \\ 0 & \text{else} \end{cases}$$

  where $\mathcal{Z}$ is a normalization factor (chosen so that D_3 will be a distribution).

- Get weak hypothesis h_3.
- Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(h_1(x) + h_2(x) + h_3(x)\right).$$

We will see that this three-round procedure can effect a small but significant boost in accuracy. As a side note (not shown in this problem), this technique can then be applied recursively to boost the accuracy to an arbitrary degree. This exact three-round approach was the main idea underlying the very first known provable boosting algorithm.

As usual, $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$ is the error of h_t on D_t. We assume $0 < \epsilon_t < \frac{1}{2}$ for $t = 1, 2, 3$. Let

$$b = \Pr_{i \sim D_2}[h_1(x_i) \neq y_i \wedge h_2(x_i) \neq y_i],$$

that is, b is the probability with respect to D_2 that both h_1 and h_2 are incorrect.

a. In terms of ϵ_1, ϵ_2, ϵ_3 and b, write *exact* expressions for each of the following:

   (i) $\Pr_{i \sim D_1}[h_1(x_i) \neq y_i \wedge h_2(x_i) \neq y_i]$.

   (ii) $\Pr_{i \sim D_1}[h_1(x_i) \neq y_i \wedge h_2(x_i) = y_i]$.

   (iii) $\Pr_{i \sim D_1}[h_1(x_i) = y_i \wedge h_2(x_i) \neq y_i]$.

   (iv) $\Pr_{i \sim D_1}[h_1(x_i) \neq h_2(x_i) \wedge h_3(x_i) \neq y_i]$.

   (v) $\Pr_{i \sim D_1}[H(x_i) \neq y_i]$.

b. Suppose $\epsilon = \max\{\epsilon_1, \epsilon_2, \epsilon_3\}$. Show that the training error of the final classifier H is at most

$$3\epsilon^2 - 2\epsilon^3,$$

   and show that this quantity is strictly less than ϵ, the (worst) error of the weak hypotheses. Thus, the accuracy receives a boost which is small, but which turns out to be enough, when applied recursively, to achieve arbitrarily high accuracy.