COS 511: Theoretical Machine Learning

Homework #6 Due:
Kernels and online learning April 10, 2019

Problem 1

[Just a reminder that the presentation given in class assumes that the hyperplane found by SVM's passes through the origin, which is different from the presentation in the readings. This problem assumes the approach followed in class.]

Suppose we use support-vector machines with the kernel:

$$K(x,z) = \begin{cases} 1 & \text{if } x = z \\ 0 & \text{otherwise.} \end{cases}$$

As we discussed in class, this corresponds to mapping each x to a vector $\boldsymbol{\psi}(x)$ in some high dimensional space (that need not be specified) so that $K(x,z) = \boldsymbol{\psi}(x) \cdot \boldsymbol{\psi}(z)$.

As usual, we are given m examples $(x_1, y_1), \ldots, (x_m, y_m)$ where $y_i \in \{-1, +1\}$. Assume for simplicity that all the x_i's are distinct (i.e., $x_i \neq x_j$ for $i \neq j$).

a. [10] Recall that the weight vector $\mathbf{w}$ used in SVM's has the form

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{\psi}(x_i).$$

Compute the α_i's explicitly that would be found using SVM's with this kernel.

b. [6] Recall that the SVM algorithm outputs a classifier that, on input x, computes the sign of $\mathbf{w} \cdot \boldsymbol{\psi}(x)$. What is the value of this inner product on training example x_i? What is the value of this inner product on any example x not seen during training? Based on these answers, what kind of generalization error do you expect will be achieved by SVM's using this kernel?

c. [6] Recall that the generalization error of SVM's can be bounded using the margin δ (which is equal to $1/\|\mathbf{w}\|$), or using the number of support vectors. What is δ in this case? How many support vectors are there in this case? How are these answers consistent with your answer in part (b)?

Problem 2

Consider the problem of learning with expert advice when one of the experts is known to give perfect predictions. In this case, it is natural to make predictions as a function only of the surviving experts, that is, the ones that have not made any mistakes so far. In class, we talked about the halving algorithm which predicts with the majority vote of the surviving expert predictions, and we also talked about the randomized weighted majority algorithm, which could be used here with β set to zero, and which predicts with one randomly selected surviving expert.

More generally, on some round t, let q be the fraction of surviving experts that predict 1, and let us suppose that the learner predicts 1 with probability $G(q)$ and 0 with probability $1 - G(q)$, for some function G. For instance, for the halving algorithm, $G(q)$ is 1 if $q > 1/2$ and 0 if $q < 1/2$ (and arbitrary if $q = 1/2$). For the randomized weighted majority algorithm (again, with $\beta = 0$), $G(q) = q$.

Consider now a function $G : [0,1] \to [0,1]$ satisfying the following property:

$$1 + \frac{\lg q}{2} \le G(q) \le -\frac{\lg(1-q)}{2}. \tag{1}$$

a. [15] Suppose we run an on-line learning algorithm that uses a function G satisfying (1) as described above. Show that the expected number of mistakes made by the learning algorithm is at most $(\lg N)/2$, where N is the number of experts. (This is half the mistake bound that was proved for the halving algorithm.)

b. [10] Show that the function

$$G(q) = \frac{\lg(1-q)}{\lg q + \lg(1-q)}$$

has range $[0,1]$ and satisfies (1). (At the endpoints, we define $G(0) = 0$ and $G(1) = 1$ to make G continuous, but you *don't* need to worry about these.)

c. [10] **(Optional – for extra credit)** Suppose now that there are $k \ge 2$ possible outcomes rather than just 2. In other words, the outcome y_t is now in the set $\{1, \ldots, k\}$ (rather than $\{0,1\}$ as we have considered up until now), and likewise, both experts and the learning algorithm make predictions in this set. Assume one of the experts makes perfect predictions. On some round t, let q_j be the fraction of surviving experts predicting outcome $j \in \{1, \ldots, k\}$. Suppose that the learning algorithm predicts each outcome j with probability

$$\frac{\lg(1-q_j)}{\sum_{i=1}^{k} \lg(1-q_i)}.$$

Show that the expected number of mistakes of this learning algorithm is at most $(\lg N)/2$.

Problem 3

[15] For this problem, let us suppose that labels, outcomes, expert/hypothesis predictions, etc. are all defined over the set $\{-1,+1\}$ rather than $\{0,1\}$. Since this does not change what it means for the learner or an expert to make a mistake, this has no effect on any of the results we have discussed regarding online mistake bounds.

Let $\mathcal{H}$ be a finite space of hypotheses $h : \mathcal{X} \to \{-1,+1\}$, and let $S = \langle x_1, \ldots, x_m \rangle$ be any sequence of m distinct points in $\mathcal{X}$. Prove that the empirical Rademacher complexity of $\mathcal{H}$ satisfies

$$\hat{\mathcal{R}}_S(\mathcal{H}) \le O\left(\sqrt{\frac{\ln|\mathcal{H}|}{m}}\right)$$

by applying our analysis of online algorithms for learning with expert advice to an appropriately constructed sequence of expert predictions ξ_i and outcomes y. Give a bound with explicit constants. (For this, it is okay, though not strictly necessary, to assume $m > \ln|\mathcal{H}|$.)

Note that this bound was earlier stated without proof in class (see Eq. (9) in the scribe notes for lecture #10), and is also a special case of Theorem 3.7 in the Mohri et al. book, although it is perfectly fine if the bound you get has weaker constants.

Extra credit [10] will be given for obtaining a bound of exactly $\sqrt{(2\ln|\mathcal{H}|)/m}$, that is, with the constant that was actually stated in class, and for all values of $m \ge 1$ (and of course, using the technique suggested above based on the algorithms we have studied for learning with expert advice). Be forewarned that getting the "right" constant in this way is a difficult challenge — but it is possible.