MCS 548 – Mathematical Theory of Artificial Intelligence
Fall 2020
Problem Set 3

Lev Reyzin

Due: 11/16/20 at the beginning of class

Instructions: Atop your problem set, please write your name and list your collaborators. You may consult outside references, but cite all the resources used (e.g. which resources on the internet you consulted). You should not, however, search for answers to these questions. All problems in this assignment require proof.

Problems

1. [**20 pts.**] Assume we are given m examples $(x_1, y_1), \dots (x_m, y_m)$ with $y_i \in \{-1, 1\}$ and that all x_is are distinct. Consider training an SVM on this dataset with the kernel: $K(x, x') = \{1 \text{ if } x = x' \text{ and } 0 \text{ otherwise}\}$.

- **a.** [**5 pts.**] Prove that K (as defined above) is positive definite symmetric. (This implies there is a function ϕ mapping each x to a high dimensional space so that $K(x, x') = \phi(x) \cdot \phi(x')$.)
- **b.** [**5 pts.**] Explicitly compute the α_is (as a function of the other specified quantities) that would be found by an SVM using this kernel to (implicitly) make the weight vector $\mathbf{w} = \sum_i \alpha_i y_i \phi(x_i)$.
- **c.** [**5 pts.**] How many support vectors will this SVM have? What margin $\rho = 1/||\mathbf{w}||$ will this SVM achieve?
- **d.** [**5 pts.**] What will this SVM do on unseen data? Do you expect this SVM to perform well? Give both an intuitive and a mathematical explanation.

2. [**10 pts.**] Suppose you have two coins, one perfectly fair, and one with bias toward H of $1/2 + \epsilon$ for some $\epsilon > 0$. It is known that to tell which coin is biased (with probability $> 3/4$) one needs to perform at least c/ϵ^2 coin flips ($c > 0$ is some constant). Show that this implies that asymptotic regret of $O(T^{1/2})$ achieved by Randomized Weighted Majority cannot be improved to $O(T^{1/2-\delta})$ for any constant $\delta > 0$.

3. [**10 pts.**] Imagine that an online learning algorithm A that runs in T rounds and has an expected regret bound of $\epsilon + T/\epsilon$, where ϵ is set by the algorithm. Clearly the optimal setting is $\epsilon = \sqrt{T}$. The problem is that sometimes T is not known in advance. How do we fix this issue? We can have an algorithm A' that does the following: A' starts with a parameter ϵ_1 and runs A for T_1 rounds, then adjusts the parameter to ϵ_2 and runs A for T_2 rounds, and so on. Construct a schedule of (ϵ_i, T_i) that asymptotically achieves the $\sqrt{T}$ expected regret bound without knowing T in advance.

4. [**10 pts.**] We proved a margin bound on the number of mistakes for the Perceptron algorithm for the update rule $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$. Consider the general update rule $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta y_t \mathbf{x}_t$, where $\eta > 0$. Prove a bound on the maximum number of mistakes for this rule. How does η affect the bound?