Computational Learning Theory
Michaelmas Term 2022

Problem Sheet 6

Instructions: The problem sheets are designed to increase your understanding of the material taught in the lectures, as well as to prepare you for the final exam. You should attempt to solve the problems on your own after reading the lecture notes and other posted material, where applicable. Problems marked with an asterisk are optional. Once you have given sufficient thought to a problem, if you are stuck, you are encouraged to discuss with others in the course and with the lecturer during office hours. You are *not permitted* to search for solutions online.

1 Learning Leaky ReLU

This question concerns learning *leaky* ReLUs. For a positive real, $0 < a < 1$, a leaky rectifier, $\ell\mathsf{r}_a$, is defined as follows:

$$\ell\mathsf{r}_a(z) = \begin{cases} z & \text{if } z \geq 0 \\ az & \text{if } z < 0. \end{cases}$$

We consider the instance space to be the unit ball in $\mathbb{R}^n$, i.e. $X_n = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_2 \leq 1\}$. For any W, we define the concept class, $\ell\mathsf{ReLU}_{n,W}$, as follows:

$$\ell\mathsf{ReLU}_{n,W} = \{\mathbf{x} \mapsto \ell\mathsf{r}_a(\mathbf{w} \cdot \mathbf{x}) \mid 0 < a < 1, \mathbf{w} \in \mathbb{R}^n, \|\mathbf{w}\|_2 \leq W\}.$$

Observe that the parameter a is not fixed and is not known in advance, but also needs to be learned. Design an algorithm that learns the class $\ell\mathsf{ReLU}_{n,W}$ in time polynomial in n, $1/\epsilon$, $1/\delta$, W, provided that it gets data $(\mathbf{x}, y)$ drawn from a distribution D supported on $X_n \times [-W, W]$, and that there exists an $a^\star$ and $\mathbf{w}^\star$, satisfying $\mathbb{E}[y \mid \mathbf{x}] = \ell\mathsf{r}_{a^\star}(\mathbf{w}^\star \cdot \mathbf{x})$. Your algorithm should output a hypothesis $h : \mathbb{R}^n \to \mathbb{R}$, such that,

$$\mathop{\mathbb{E}}_{\mathbf{x} \sim D_X} \left[\left(h(\mathbf{x}) - \ell\mathsf{r}_{a^\star}(\mathbf{w}^\star \cdot \mathbf{x}) \right)^2 \right] \leq \epsilon.$$

Above D_X is the marginal of the distribution D over X_n. You should argue about the correctness of your algorithm, as well as justify bounds on sample complexity and running time.

2 Rademacher Complexity and VC Dimension

Let C be a class of boolean functions defined over an instance space X and for this class, let the Vapnik Chervonenkis dimension, $\mathsf{VCD}(C) = d$. In this question, we will treat boolean functions as taking values in the range $\{-1, 1\}$. You may use the following result.

Lemma (Massart): Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^m$ be n vectors. Then the following holds:

$$\mathbb{E}_\sigma \left[\frac{1}{m} \max_{j \in [n]} \sum_{i=1}^{m} x_{j,i} \sigma_i \right] \leq \max_{j \in [n]} \|\mathbf{x}_j\|_2 \cdot \frac{\sqrt{2 \log n}}{m},$$

where σ_is are independent random variables taking values in $\{-1, 1\}$ with equal probability.

1. Let $S \subseteq X$ be a finite set of size m. Give the tightest possible bound you can on the empirical Rademacher complexity of C over the set S, $\widehat{\mathsf{RAD}}_S(C)$, in terms of m and d.
2. For any function $f : X \to \mathbb{R}$, let us define the function, $\text{sign}(f) : X \to \{-1, 1\}$, by $\text{sign}(f)(x) = \text{sign}(f(x))$. For a class of functions C over X, let $\text{sign}(C) = \{\text{sign}(f) \mid f \in C\}$; we treat $\text{sign}(0) = 1$. Show that one can construct a sequence of a class of functions (not necessarily boolean) $C_{(i)}$ defined over some set X (which can be of your choice), for which for any $m \in \mathbb{N}$, there exists a subset $S \subseteq X$ of size m, such that $\widehat{\mathsf{RAD}}_S(C_{(i)}) \to 0$, as $i \to \infty$, but $\widehat{\mathsf{RAD}}_S(\text{sign}(C_{(i)})) = 1$ for all i.
3. *(Optional)* Prove Massart's Lemma.

3 Mistake Bound of Perceptron

Consider the perceptron algorithm studied in the lectures, the main outline of which is produced below:

- Set $\mathbf{w}_1 = \mathbf{0}$, $(\mathbf{w}_1 \in \mathbb{R}^n)$.
- For $t = 1, 2, \ldots$,
  - When given with $\mathbf{x}_t$, output the prediction $\widehat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$.
  - Observe $y_t \in \{-1, 1\}$.
  - If $y_t \neq \widehat{y}_t$, update $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\mathbf{x}_t$, else $\mathbf{w}_{t+1} = \mathbf{w}_t$.

Suppose that it holds for each $\mathbf{x}_t$ that $\|\mathbf{x}_t\|_2 \leq D$ for some $D > 0$, and that there exists a $\mathbf{w}^* \in \mathbb{R}^n$, such that $\|\mathbf{w}^*\|_2 = 1$ and for every t, $y_t(\mathbf{w}^\star \cdot \mathbf{x}_t) \geq \gamma$ for some $\gamma > 0$. In the lectures, we proved that the number of mistakes made by the perceptron algorithm is bounded by D^2/γ^2. Show that this is tight (at least up to constant factors), that is there exists a sequence of $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots$ satisfying the aforementioned conditions and the number of mistakes made by the algorithm on this sequence is $\Omega(D^2/\gamma^2)$.