Machine Learning 4771: Midterm

The exam is worth 25% of your grade. You can choose to do any two of the three problems below (16 points for two) + exercises (9 points total) to get 25 points. If you do all three problems + exercises, you will get 8 bonus points.

Problem 1

(8 points) A q-quantile for a distribution D over $[0,1]$ is a value Q such that

$$\mathbf{Pr}_{y\sim D}[y \le Q] \ge q \quad \text{and} \quad \mathbf{Pr}_{y\sim D}[y \ge Q] \ge 1-q.$$

The 1/2-quantile is known as the median.

Recall that the median is the minimizer of the absolute loss. More formally, for every distribution D over $[0,1]$, the median of D is

$$\text{argmin}_{a\in[0,1]}\mathbf{E}_{y\sim D}\left[\,|y-a|\,\right].$$

Problem: Produce a generalization of the absolute loss function which is minimized (over all distributions) by a q-quantile, $0 \le q \le 1$. In other words, find a function $\ell_q(y,a)$ such that a q-quantile is

$$\text{argmin}_{a\in[0,1]}\mathbf{E}_{y\sim D}\left[\,\ell_q(y,a)\,\right].$$

Hint: ℓ_q is an appropriately tilted absolute loss.

Solution: The q-quantile is the minimizer of $\mathbf{E}_{y\sim D}\ell_q(y,a)$, where

$$\ell_q(y,a) = \begin{cases} q(y-a), & y \ge a \\ (1-q)(a-y), & y < a \end{cases}$$

(The q-quantile may not be unique when D has regions with zero mass.)

Problem 2

(8 points) Consider the following version of the Winnow algorithm (for OR functions):

- Initialize the weights $w_1 = w_2 = \ldots = w_n = 1$ on the n variables.
- Given an example $x = (x_1, \ldots, x_n)$, output 1 if $\sum_{i=1}^n w_i x_i \ge n$, else output 0.
- Update step:
  - If the label of x is 1, **double** the value of w_i for each i such that $x_i = 1$ (regardless of whether we made a mistake on x or not).
  - If the label of x is 0, **halve** $w[i]$ for all i such that $x_i = 1$.

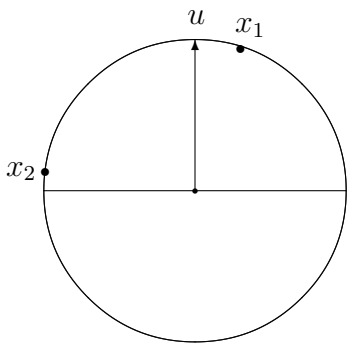
Problem: Produce an infinite sequence of examples (consistent with some OR function on n boolean variables) that forces this algorithm to make an infinite number of mistakes.

Solution: Let $n = 2$. We will construct an infinite sequence of examples consistent with the disjunction $f(x_1, x_2) = x_1$. By repeating the following block of two examples, we continue forcing the algorithm to make one mistake per block.

x_1	x_2	$f(x_1, x_2)$	prediction
1	1	1	1
0	1	0	1
...	...	...	...

Problem 3

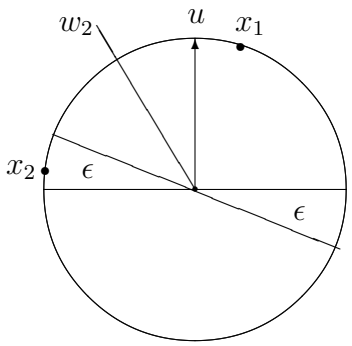
(8 points)



Perceptron: Suppose that data points are distributed uniformly over the unit circle $S = \{x \in \mathbb{R}^2 \mid \|x\| = 1\}$. The target function is $\text{sign}(u \cdot x)$ represented by a unit vector $u \in \mathbb{R}^2$, which classifies all points perfectly. (See the picture above.)

- Starting the perceptron algorithm with $w_0 = \mathbf{0}$, show the hypothesis w_1 after observing x_1, and show the region(s) of S where u and w_1 disagree (i.e., show all $x \in S$ such that $\text{sign}(w_1 \cdot x) \neq \text{sign}(u \cdot x)$).
- If the angle between u and w_1 is $10°$, compute the error rate of w_1 on the underlying data distribution D (which is uniform over S), i.e., compute $\mathbf{Pr}_{x \sim D}[\text{sign}(w_1 \cdot x) \neq \text{sign}(u \cdot x)]$.
- Show the hypothesis w_2 after observing x_2, starting with w_1 as the current hypothesis. Is the error rate of w_2 smaller than that of w_1 (on D)? If not, how would you change the update rule to correct the problem?

Solution: The first hypothesis is $w_1 = x_1$. The arcs of disagreement correspond to the regions marked with ϵ.



The error rate of w_1 on D is given by (where $\theta(u, w_1)$ is the angle between u and w_1 in degrees)

$$\mathbf{Pr}_{x\sim D}[\text{sign}(u \cdot x) \neq \text{sign}(w_1 \cdot x)] = \frac{\theta(u, w_1)}{180^\circ} = \frac{10^\circ}{180^\circ} = 1/18.$$

The hypothesis $w_2 = x_1 + x_2$. The error rate of w_2 is higher, because the perceptron update overshoots and swings too far to the other side of u. To prevent the update from overshooting when the current hypothesis w_t is already close to u, the norm of w_t should be fairly high, on the order of $1/\sin\theta(w_t, u)$ (easiest to see it geometrically). On the other hand, we know that the norm grows quite slowly, as the square root of the number of mistakes. To avoid the oscillations caused by points close to the half space represented by the current hypothesis, we can scale the update rule with $w_t \cdot x$.

Exercises

1. (3 points) A fair coin is flipped until the first head occurs. Let Z denote the number of flips required. Find the entropy $H(Z)$ and the expected value of Z. The following expression may be useful:

$$\sum_{i=1}^{\infty} ir^i = \frac{r}{(1-r)^2}$$

Solution:
Since $\mathbf{Pr}[Z = i] = (1/2)^i$, we have

$$E(Z) = \sum_{i=1}^{\infty} i(1/2)^i = 2.$$

The entropy

$$H(Z) = -\sum_{i=1}^{\infty} (1/2)^i \log(1/2)^i = \sum_{i=1}^{\infty} i(1/2)^i = 2.$$

2. (3 points) Concisely (2-3 sentences) state the idea behind the kernel trick. Be precise.

   Look up any credible source.

3. (3 points) Consider a variation of the deterministic Weighted Majority designed to make it more adaptive:

   (a) Each expert begins with weight 1 (as before)

   (b) We predict the result of a weighted-majority vote of the experts (as before)

(c) If an expert makes a mistake, we penalize it by dividing its weight by 2, but only if its weight is at least 1/4 of the average weight of experts.

Suppose that we have two experts and suppose that on the first 50 examples the first expert is correct and the second is wrong, and then on the next 50 examples the second expert is correct and the first is wrong.

How many mistakes does the original deterministic Weighted Majority algorithm make in this case (assume that it always makes the wrong choice when there is a tie)? How many mistakes does the modified version make?

Solution:
The original version makes 51 mistakes (the first tie + the last 50 examples).

The modified version makes 5 (the first tie + examples 51 through 54 to bring the weight of the first expert down from 1 to 1/8; example 54 is a tie).