Assignment 2

CS289: Algorithmic Machine Learning, Spring 2022

Due: April 27, 10PM

Guidelines for submitting the solutions:

- The assignments need to be submitted on Gradescope. Make sure you follow all the instructions - they are simple enough that exceptions will not be accepted.
- Start each problem or sub-problem on a separate page even if it means having a lot of whitespace and write/type in large font.
- The solutions need to be submitted by 10 PM on the due date. No late submissions will be accepted.
- Please adhere to the code of conduct outlined on the class page.

1. Consider the hypothesis class $\mathcal{H}$ consisting of functions $f : \{0,1\}^d \to \{0,1\}$ of the form $AND_S(x) = \wedge_{i \in S} x_i$ for subsets $S \subseteq [d]$. Now, suppose you are given as input $(example, label)$ pairs $(x^1, y_1), \ldots, (x^m, y_m), \ldots,$ where $x^\ell \in \{0,1\}^d$ and $y^\ell = AND_{S_*}(x^\ell)$ for some unknown non-empty S_*. Give an efficient (the algorithm should run time at most polynomial in d) online learning algorithm in the *mistake bounded* model. For full-credit your algorithm should run in polynomial time for each example, and must make at most a polynomial number of mistakes in its entire run-time (no matter how long). You don't have to prove that your algorithm works. [4 points]

   Hint: Start with the full set of features and think of weeding out coordinates that you know *cannot* be in S_*.

2. Show that no *deterministic algorithm* can do better than a factor two approximation to the loss of the best expert for the *learning with experts* problem. That is, show that for any deterministic algorithm, there exists a sequence of losses where the algorithm is off by at least a factor of two from the performance of the best expert. [4 points]

   Hint: You can come with an example even with just two experts.

3. The goal here is to solve a variant of the learning with experts setup where the experts incur *losses* that take values in $[0,1]$ (instead of $\{0,1\}$ as we did in class). Let us suppose that on each day, we are trying to guess some number between $[0,1]$ (instead of just UP vs DOWN as in class) based on the guesses of the experts.

   Concretely, consider a learning with experts setup where you have n experts $E_1, \ldots, E_n$ and the interaction proceeds as follows:

- At the beginning of each day, the experts make a prediction which is a number between $[0, 1]$.
- Our algorithm then makes a prediction which is also a number between $[0, 1]$.
- At the end of the day, the true number (for that day) is revealed to us and each expert and our algorithm incur a loss that is the absolute-value of the difference between their prediction and the true value.

Given $0 < \epsilon < 1$, design a randomized algorithm whose expected total-loss after T days is at most $(1 + \epsilon)\mathcal{A}_*(T) + (\ln n)/\epsilon$, where $\mathcal{A}_*(T)$ is the loss of the best-expert after T days. Give a complete proof that the algorithm achieves the above bound. [5 points]

Hint: You basically have to slightly change the multiplicative weights algorithm we did in class (in how you update the weights). The analysis also does not change much; you may use the following fact: For $0 \leq \epsilon \leq 1/2$, $(-\ln(1 - \epsilon))/\epsilon \leq (1 + \epsilon)$.

4. Let us now apply MWM to an online learning problem like the perceptron. Suppose we are given some examples $(x_1, y_1), \ldots, (x_t, y_t), \ldots$ in an online manner where $x_i \in [-1, 1]^d$ and $y_i \in \{1, -1\}$. We know that $y_i = sign(\langle w_*, x_i \rangle)$ where w_*, the *unknown true coeffecient* vector satisfies:

   (a) w_* has non-negative coordinates with $\|w_*\|_1 = 1$ (i.e., has sum of entries exactly 1).

   (b) For all i, $y_i \langle w_*, x_i \rangle > \gamma$.

   Use MWM to come up with an online learning algorithm where the number of mistakes (for any arbitrary sequence of points (x_i, y_i)) after T days is $O(\sqrt{T \log d}/\gamma)$.

   Hint: Try to keep things simple (the solution can be reasonably short). Note that there is no *learning with experts* problem here to start with! But we can define a thought experiment where the features of x correspond to experts. The entire problem and solution then comes down to defining a suitable (and perhaps somewhat magical) choice of losses for each day so that applying the regret bound for MWM (the more general version with losses in $[0, 1]$ as in the previous problem) gives you the claim above. You don't have to redo the analysis of MWM. [5 points]

   (Remark: The second condition above on w_* is similar in spirit to the margin condition we had in analysis of perceptron **but** is different. The normalizations are different - the x vectors are not set to be on the unit sphere any more. If you apply the perceptron algorithm, you end up with a guarantee like $O(d/\gamma^2)$ which could be much worse than the above if the dimension is big.)