COMS 4773 Spring 2024 HW 1 (due Feb. 9 at noon)

Please read the course syllabus carefully for the policy on collaboration and homework submission.

Problem 1. Answer the following questions about COMS 4773.

(a) What should you do if you miss a lecture?

(b) What is the maximum size of a group you can form to discuss homework problems?

(c) TRUE or FALSE: If a student discusses the homework with another student, then the two of them may just submit a single write-up with both of their names on it.

(d) What are the LaTeX commands to write "∀" and "∃"?

All of the answers are in the course syllabus or can be found by following some links provided in the course syllabus. You are hereby explicitly permitted to use the course syllabus and the links provided therein to solve this particular problem.

Problem 2. Consider the "Using Expert Advice" problem with N experts, but now for predicting a ternary outcome, so $y_t \in \{1, 2, 3\}$ and $b_{t,i} \in \{1, 2, 3\}$ for all rounds t and all $i \in [N]$. Assume that Nature will ensure that there is an expert who makes no mistakes. Give an algorithm that ensures the learner makes no more than $C \log_2 N$ mistakes, where $C > 0$ is a universal positive constant. Try to make the constant C as small as possible; what C can you achieve? Give a proof that this mistake bound holds for your algorithm. *Bonus*: Show that this value of C cannot be improved.

Problem 3. Suppose that, in the "Using Expert Advice" problem, Nature guarantees that one of the N experts will make no more than $K^\star$ mistakes (ever), but *this number $K^\star$ is not known to the learner*. Explain how to modify WEIGHTED MAJORITY with the "doubling trick" (sketched in lecture) to guarantee a mistake bound of

$$2K^\star + C\left(\sqrt{K^\star \log N} + \log N\right)$$

where $C > 0$ is some universal positive constant. Give a proof that this mistake bound holds for your modified algorithm. (If you get something like $2K^\star + C\left(\sqrt{K^\star \log N} + (\log K^\star)(\log N)\right)$, that's also fine.)

Aside: An approach that seems more natural than the doubling trick is to simply use weights $w_{t,i} = \beta_t^{M_{t-1,i}}$, where $M_{t-1,i}$ is the number of mistakes made by expert i up through round $t-1$, and $\beta_t \in (0,1)$ is chosen in some careful way based on $\min_{i \in [N]} M_{t-1,i}$. I think something like this may work, but also that the proof may be more complicated. (If you manage to make this work with a very simple proof, I would love to see it!)

Problem 4. Consider the following alternative algorithm for Online Allocation (assuming $\ell_t \in [0,1]^N$ for all rounds t):[1]

- Initialize $w_{1,i} = 1$ for all $i \in [N]$.
- For $t = 1, 2, \ldots$:
  - Choose allocation vector $p_t = (p_{t,1}, \ldots, p_{t,N}) \in \Delta^{N-1}$, where

    $$p_{t,i} = \frac{w_{t,i}}{Z_t} \quad \text{for all } i \in [N]$$

    and $Z_t = \sum_{i=1}^N w_{t,i}$.
  - Get loss vector $\ell_t = (\ell_{t,1}, \ldots, \ell_{t,N}) \in [0,1]^N$.
  - If $R_{t,i} \le 0$ for all $i \in [N]$, then set $w_{t+1,i} = 1$ for all $i \in [N]$; otherwise set $w_{t+1,i} = \max\{R_{t,i}, 0\}$ for all $i \in [N]$.
    (Recall that $R_{t,i} = \sum_{s=1}^t \langle p_s, \ell_s \rangle - \sum_{s=1}^t \ell_{s,i}$.)

In this problem, you will prove that this algorithm guarantees $R_T \le \sqrt{TN}$. (Yes, this has quite a bit worse dependence on N than what HEDGE achieves, but at least the dependence on T is sublinear. And the algorithm is simple and has no hyperparameters!)

(a) Define $r_{t,i} = \langle p_t, \ell_t \rangle - \ell_{t,i}$ for all $i \in [N]$. Prove that, for every round t, and every $i \in [N]$,

$$(\max\{R_{t,i}, 0\})^2 \le (\max\{R_{t-1,i}, 0\})^2 + 2\max\{R_{t-1,i}, 0\} r_{t,i} + r_{t,i}^2,$$

where we regard $R_{0,i} = 0$ for all $i \in [N]$.

(b) Using the result from part (a) and the way the algorithm chooses allocation vectors, prove that

$$\sum_{i=1}^N (\max\{R_{t,i}, 0\})^2 \le \sum_{i=1}^N (\max\{R_{t-1,i}, 0\})^2 + \sum_{i=1}^N r_{t,i}^2.$$

(c) Use the previous two parts to complete the proof that, for any T, the regret of the learner after T rounds satisfies $R_T \le \sqrt{TN}$.

(Remember that the loss vectors are chosen from $[0,1]^N$.)

[1] This algorithm is related to—but not the same as—the "NormalHedge" algorithm from `https://arxiv.org/abs/0903.2851`.

Problem 5. Consider the following modification of WEIGHTED MAJORITY for the "Using Expert Advice" problem (with binary outcomes):

- Initialize $w_{1,i} = 1$ for all $i \in [N]$.
- For $t = 1, 2, \ldots$:
  - Observe experts' predictions $b_{t,1}, \ldots, b_{t,N} \in \{-1, 1\}$.
  - Choose predict $a_t = \operatorname{sign}(\sum_{i=1}^N w_{t,i} b_{t,i})$ (weighted majority vote).
  - Observe outcome $y_t \in \{-1, 1\}$.
  - Update weights: for all $i \in [N]$,

$$w_{t+1,i} = \begin{cases} w_{t,i} & \text{if } y_t = b_{t,i} \text{ or } w_{t,i} < \frac{1}{3} \cdot \frac{1}{N} \sum_{j=1}^N w_{t,j}; \\ \frac{1}{2} w_{t,i} & \text{if } y_t \neq b_{t,i} \text{ and } w_{t,i} \geq \frac{1}{3} \cdot \frac{1}{N} \sum_{j=1}^N w_{t,j}. \end{cases}$$

This is like WEIGHTED MAJORITY (with $\beta = 1/2$), except that a mistaken expert's weight is halved only if its weight was at least $1/3$ of the average weight in that round.

(a) Let $Z_t = \sum_{i=1}^N w_{t,i}$ for all rounds t. Explain why, at the start of any round t, every expert i has weight $w_{t,i}$ satisfying

$$w_{t,i} \geq \frac{Z_t}{6N}.$$

(The modification to how weights are updated is important for this part.)

(b) Prove that this modified version of WEIGHTED MAJORITY has the following guarantee. Pick any segment of consecutive rounds, say, $a, a+1, a+2, \ldots, b$, with $a \leq b$. Let $K_{a,b}$ be the smallest number of mistakes committed by an expert in the rounds $a, a+1, a+2, \ldots, b$. Then the learner makes at most

$$8K_{a,b} + 12 \ln N + 21$$

mistakes in the rounds $a, a+1, a+2, \ldots, b$.

The guarantee in part (b) is useful if the "good expert" changes over time. For example, suppose Expert 1 always predicts 1, and Expert 2 always predicts -1. Further, suppose the outcomes in the first 1000 rounds are all 1's, so $K_{1,1000} = 0$ (achieved by Expert 1), but the outcomes in the next 1000 rounds are all -1's, so $K_{1001,2000} = 0$ (achieved by Expert 2). Each of Expert 1 and Expert 2 makes 1000 mistakes over all 2000 rounds; neither is particularly good for all 2000 rounds. (Think about how well the original WEIGHTED MAJORITY, with $\beta = 1/2$, would do in this example...) The learner, using this modified algorithm, will make at most a small number of mistakes overall:

- at most $8K_{1,1000} + 12 \ln 2 + 21 \approx 30$ mistakes in rounds $1, \ldots, 1000$;
- at most $8K_{1001,2000} + 12 \ln 2 + 21 \approx 30$ mistakes in rounds $1001, \ldots, 2000$.

The modification in how the weights are updated ensures that the learner never stops anticipating that some currently poorly-performing expert may become good later on.