Homework #2
CS 6501: Learning and Game Theory (Fall'19)

Due Tuesday 10/15 3:30 pm

General Instructions The assignment is meant to be challenging. Feel free to discuss with fellow students, however please write up your solutions independently (e.g., start writing solutions after a few hours of any discussion) and acknowledge everyone you discussed the homework with on your writeup. The course materials are all on the course website: http://www.haifeng-xu.com/cs6501fa19. You may refer to any materials covered in our class. However, any attempt to consult outside sources, on the Internet or otherwise, for solutions to any of these homework problems is *not* allowed.

Whenever a question asks you to "show" or "prove" a claim, please provide a formal mathematical proof. These problems have been labeled based on their difficulties. `Short` problems are intended to take you 5-15 minutes each and `medium` problems are intended to take 15-30 minutes each. `Long` problems may take anywhere between 30 minutes to several hours depending on whether inspiration strikes.

Finally, please write your solutions in latex — hand written solutions will not be accepted. Hope you enjoy the homework!

Problem 1 (`Short, 3 points`)

When we argue that pseudo-regret is at most the (external) regret, we used the following fact: for any random vector $C \in \mathbb{R}^n$, we have $\min_{j\in[n]} \mathbb{E}[C(j)] \geq \mathbb{E}\big[\min_{j\in[n]} C(j)\big]$. Prove this claim.

(For this problem, you can assume C has finite support, i.e., value of C is from a finite set of vectors, though this conclusion does hold in general.)

Problem 2: Regret Analysis for Exponential-Weight Update

During lecture we omitted some details in the analysis of the regret bound for Exponential-Weight (EW) update for the *full information* setting (i.e., the learner can see the whole cost vector). In this problem, you are asked to give a complete proof of the regret upper bound, with the following steps. Recall the notations: (1) there are n actions in set $[n] = \{1, \cdots, n\}$; (2) $c_t \geq 0$ is the cost *vector* the learner observes at round t; (3) $W_t = \sum_{i=1}^n w_t(i)$ is the total weight at round t; (4) the update rule in EW is as follows: at time t, for any action i we set $w_{t+1}(i) = w_t(i)e^{-\epsilon c_t(i)}$.

1. (`Short, 3 points`) Prove that $W_{t+1}/W_t = \sum_{i=1}^n p_t(i)e^{-\epsilon c_t(i)}$ where $p_t(i) = w_t(i)/W_t$.

2. (**Regret Bound of Exponential-Weight Update**, `Medium, 5 points`) Using the above conclusion, together with the fact we proved in class

$$\sum_{t=1}^{T} \log\big(\sum_{i=1}^{n} p_t(i) e^{-\epsilon c_t(i)}\big) \le \sum_{t=1}^{T}\sum_{i=1}^{n} p_t(i)\big(-\epsilon c_t(i) + \frac{\epsilon^2}{2}[c_t(i)]^2\big),$$

prove the following regret bound for EW

$$R_T \le \frac{\ln n}{\epsilon} + \frac{\epsilon}{2}\sum_{t=1}^{T}\sum_{i=1}^{n} p_t(i)[c_t(i)]^2.$$

Problem 3: The Experts' Advice Problem

The *experts' advice* problem is a slight variant of the online learning problem. Here there are n experts, each making a prediction about a *binary* event (e.g., the stock market will go up or down tomorrow). For round $t = 1, \cdots, T$, the following occurs in order: (1) each expert i makes a binary prediction $a_t(i) \in \{0, 1\}$; (2) after observing these predictions, the learner comes up with his own prediction $\overline{a_t}$; (3) the binary event is realized; (4) The learner observes whether she made a correct prediction as well as whether each expert made a correct prediction at this round. The learner's goal is to design an online learning algorithm that makes as few mistakes as possible for any set of experts and any event realization.

1. (`Short,3 points`) Formalize the definition of regret in this setting.

2. (`Short,3 points`) Assume that one of the expert is perfect, i.e., all his predictions are correct. Show that in this case, there exists a learning algorithm that has regret at most $O(\ln n)$.

3. (`Medium, 5 points`) If none of the experts are perfect, one natural algorithm to make predictions is to use a *weighted majority voting* rule. In particular, consider the following algorithm, parameterized by ϵ.

   (a) Initialize $w_1(i) = 1$ for all expert i.

   (b) At round $t = 1 \cdots, T$

      i. After observing each expert's prediction, the learner computes the total weight for prediction 1 and 0, as $W_t^1 = \sum_{i=1}^{n} w_t(i) \cdot \mathbb{I}(a_t(i) = 1)$ and $W_t^0 = \sum_{i=1}^{n} w_t(i) \cdot \mathbb{I}(a_t(i) = 0)$, respectively, and predict 1 if and only if $W_t^1 \ge W_t^0$. Here, $\mathbb{I}(A)$ is the indicator function, which equals 1 if and only if A is true and 0 otherwise.

      ii. After the binary event is realized, update expert i's weight as follows: $w_{t+1}(i) = w_t(i)(1-\epsilon)$ if i made a wrong prediction and $w_{t+1}(i) = w_t(i)$ if i made a correct prediction

   Derive a regret upper bound for this online learning algorithm, as a function of the parameter ϵ. Is your regret bound sublinear for any problem instance?

4. (`Short, 3 points`) Show that there exists an online learning algorithm for the experts' advice problem which has sublinear regret for any problem instance.

	Silent	Betray
Silent	(-1, -1)	(-3, 0)
Betray	(0, -3)	(-2, -2)

Table 1: Payoffs of the Prisoner's Dilemma

Problem 4: Convergence of No-Regret Dynamics

- (`Medium, 5 points`) Recall that the Prisoner's dilemma has the payoff matrix as in Table 1

  Assume that two prisoners play the above game repeatedly for T rounds and each player uses a no-external-regret learning algorithm with $x_t^i \in \Delta_2$ as the mixed strategy of player $i = 1, 2$ at round $t = 1, \cdots, T$ (note: x_t^i is a vector in Δ_2, satisfying $x_t^i(1) + x_t^i(2) = 1$).

  Prove that the *average history* $(\frac{1}{T}\sum_{t=1}^T x_t^1, \frac{1}{T}\sum_{t=1}^T x_t^2)$ converges to the Nash equilibrium of this game as $T \to \infty$.

  Hint: be careful that the strategy profile $(\frac{1}{T}\sum_{t=1}^T x_t^1, \frac{1}{T}\sum_{t=1}^T x_t^2)$ is *not* the strategy profile we constructed in class when proving convergence to coarse correlated equilibrium.

- (`Medium, 5 points`) Show that the above convergence to Nash equilibrium is *not* true in general, i.e., there exists a two-player game for which no-external-regret dynamics do not converge to a Nash equilibrium. Do this by explicitly describing such a game, and showing that when both players use the multiplicative weights algorithm, the average history of joint play is far from a Nash equilibrium regardless of the time horizon T.

Problem 5: Properties of Swap Regret

In this problem, you will prove the following properties about swap regret.

1. (`Long,10 points`) Show that a policy minimizing the expected loss does not necessarily minimize the swap regret. In particular, consider an online learning instance with n actions and $T = n^2$ rounds. The cost for each action at each round t is drawn from $\{0, 1\}$ uniformly at random, except that for any round $t = (i-1) * n + 1, \cdots, i * n$, the cost of action i is 1 with only probability $\frac{1}{2} - \frac{1}{T}$. Which policy minimizes the expected loss in this example? What is the swap regret of this policy? Is there a policy that has smaller swap regret?

2. (`Long,10 points`) The multiplicative weight (MW) algorithm has sublinear regret. In this problem, we show that MW does *not* guarantee sublinear *swap regret*.

   In particular, consider an online learning instance with $n = 3$ actions A, B, C and $T = 3k$ rounds. The $3k$ rounds are divided into 3 equal-length regimes and the cost vector within each regime remains the same. In particular, the cost vector at each round is described as follows:

round	$c_t(A)$	$c_t(B)$	$c_t(C)$
$1 \le t \le k$	0	1	5
$k+1 \le t \le 2k$	1	0	5
$2k+1 \le t \le 3k$	2	1	0

Table 2: Descriptions of the Cost for the Constructed Example

Assume that we run the MW with $\epsilon = \sqrt{\ln n/T} = \sqrt{\ln 3/T}$. We proved in class that this ϵ guarantees external regret at most $O(\sqrt{T})$. In this question, you are asked to prove that MW will have linear swap regret in the above example.

Hint: The high-level idea is as follows. Since all the cost vectors have already been given to us, we can explicitly compute the probability that MW picks each action at any round. It is intuitive to see that in the first $2k$ rounds, MW will mostly pull action A and from round $2k+1$ to $3k$, MW will quickly switch to mostly pull action B. Indeed, it turns out that for any small $\delta \in (0,1)$, within the first $2k$ rounds, the total number of rounds at which action B is pulled with probability at least δ is at most $O(\sqrt{k}\ln\frac{1}{\delta})$, but for round $t = 2k+1, \cdots, 3k$, action B will be pulled with large probability at every round (you will need to prove all these). However, at round $t = 2k+1, \cdots, 3k$ you really regret a lot for not swapping to action C. This implies that the simple swap function $s(B) = C, s(A) = A, s(C) = C$ will result in linear swap regret already.