Project #2

Due Date: 04/11/2022, Monday, 10 pm

We have seen a few online learning algorithms in class — like multiplicative weight update, EXP3 — which provably have sublinear regret for adversarial bandit problems. In this project, you will study how these no regret learning algorithms work in games. We shall examine two simplest possible classes of games – zero-sum games and dominance elimination solvable games. As you will see, learning in such basic *multi-agent* setups is already complex and exhibit surprising behaviors.

Some notations and the setup. For convenience, we shall only look at games with two players, denoted as player A and B, in this entire project. Player A has n actions, denoted by set $[n] = \{1, \cdots, n\}$; similarly, player B has m actions, denoted by $[m]$. If A plays $i \in [n]$ and B plays $j \in [m]$, then A gets *utility* $u^A(i,j)$ and B gets $u^B(i,j)$. A mixed strategy for player A is denoted by $x \in \Delta_n = \{x \in [0,1]^n : \sum_{i=1}^n x_i = 1\}$ (Δ_n is called the *n-dimensional simplex* and contains all discrete distributions over set $[n]$). Similarly, a mixed strategy for player B is denoted by $y \in \Delta_m$.
Before describing the concrete project task, let us recall the setup of the *multi-agent learning* problems in a game. The game will be repeatedly played for T rounds.[1] At each round $t = 1, \cdots, T$, the following occurs in order

1. Player A picks a mixed strategy $x^t \in \Delta_n$; simultaneously, Player B picks a mixed strategy $y^t \in \Delta_m$.
2. Player A then samples action i^t from distribution x^t; simultaneously, Player B samples action j^t from distribution y^t;
3. Player A receives interim utility $u^A(i^t, y^t) = \sum_{j=1}^m y_j^t u^A(i^t, j)$; Player B's interim utility is $u^B(x^t, j^t) = \sum_{i=1}^n x_i^t u^b(i, j^t)$.[2]

The above procedure is almost fully specified, up to the following two key ingredients:

(a) What information can each player see at each round? There are two different situations.

- **Full Information.** This case assumes that each player at any round t can observe her opponent's strategy. That is, both players know the x^t, y^t played at round t. Consequently, player A can know her own utility for any action i, i.e., $u^A(i, y^t)$ (similarly for B).
- **Bandit Information.** This case assumes that each player at any round t cannot observe her opponent's strategy, and only sees the particular utility value $u^A(i^t, y^t)$ for the action i^t she played .

[1] For example, you play rock-paper-scissor with a friend for $T = 1000$ rounds.

[2] Here, you may wonder why a player does not receive utility $u^A(i^t, j^t)$, but instead receives $u^A(i^t, y^t)$. For now, you can think of it as just an assumption in this project, but there are real reasons to assume this (typically because each strategy y^t is not played only once but in a period, and thus we take expectation).

(b) What algorithm that each player uses to pick their strategy x^t, y^t at each round? In this project, we shall consider the situation where both players will use the same no regret learning algorithm (this setup is also called "Uncoupled No-Regret Learning Dynamics"). Notably, just like in the standard online learning setups as discussed in the class, a player can use all her previously observed history — though what she can observe depends on whether she is in full information or bandit information feedback — to determine the current round's mixed strategy.

Finally, we discuss two types of convergence that we expect to see:

- **Average convergence**: the average strategy profile $(\frac{\sum_{t=1}^{T} x^t}{T}, \frac{\sum_{t=1}^{T} y^t}{T})$ converges.
- **Last-iterate convergence**: the latest strategy profile (x^T, y^T) converges.

While we will not prove it here, but it is not difficult to see that last-iterate convergence is strictly stronger than average convergence. Also, in modern deep learning era, last-iterate convergence is preferred since in complex games, agent's strategies x^t, y^t will be represented as deep neural networks and it is impossible to average many neural networks.

Okay, so now we are ready to describe the tasks for Project 2!

Task I: Uncoupled No-Regret Learning Dynamics under Full Information.

(a) Consider the following simple 2×2 matching pennies game.

Player B

		Heads	Tails
Player A	Heads	(1,-1)	(-1,1)
	Tails	(-1,1)	(1,-1)

Implement the Multiplicative Weight Update (MWU) algorithm from Lecture 10 with the following different parameter choices of ϵ_t (i.e., the ϵ at round t): (1) $\epsilon_t = 0.5$; (2) $\epsilon_t = \frac{1}{t}$; (3) $\epsilon_t = \frac{1}{t^{2/3}}$; (4) $\epsilon_t = \frac{1}{\sqrt{t}}$; (5) $\epsilon_t = \frac{1}{t^{1/3}}$. *Moreover, and importantly, please initialize your weights randomly (please do not set the weight all as* 1 *like we did in class).*[3] Then let both players use a version of the MWU above to play the matching pennies game in *full information* setup for 3000 rounds.

For each of the five different versions of the MWU algorithm, plot your algorithm's sequence of (x_1^t, y_1^t) for $t = 1, \cdots, T = 3000$ in a figure (x_1^t is player A's probability of playing action "Heads" at round t). Does the sequence converge? Briefly describe what you see or discovered (e.g., for what choice of ϵ do you see convergence?).

For each of the five different versions of the MWU algorithm, plot your algorithm's sequence of $(\frac{\sum_{\tau=1}^{t} x_1^\tau}{t}, \frac{\sum_{\tau=1}^{t} y_1^\tau}{t})$ for $t = 1, \cdots, T = 3000$ in a figure. Does this average sequence converge? Briefly describe what you see or discovered.

[3]This should have not been needed in general, but the matching pennies game happens to have some degeneracy, which needs randomized initialization to overcome; otherwise, your algorithm may get stuck from the second step.

(b) Repeated task (a) above still with $T = 3000$, but for a different zero-sum game, the rock-paper-scissor game. In this case, do you observe convergence (average or last-iterate) to the unique equilibrium of the game? Explain what you observed. Try to find a good way to present what you found (this may require some creativity from you. For instance, think about what kind of plots can clearly illustrate your message).

		Player B		
		Rock	Paper	Scissors
Player A	Rock	(0,0)	(-1,1)	(1,-1)
	Paper	(1,-1)	(0,0)	(-1,1)
	Scissors	(-1,1)	(1,-1)	(0,0)

(c) In this problem, we consider a different class of two-player games, called the Diamond-in-the-Rough (DIR) game, defined as follows.

Definition 1 (The Diamond-In-the-Rough (DIR) Games) *A DIR game is a two-player game parameterized by (K, c). Each agent have K actions and utility function*

$$u^A(i,j) = \begin{cases} i/\rho & i \le j+1 \\ -c/\rho & i > j+1 \end{cases}, \qquad u^B(i,j) = \begin{cases} j/\rho & j \le i \\ -c/\rho & j > i \end{cases}. \tag{1}$$

where $c > 0$ and $\rho = \max\{K, c\}$ is for normalization purpose. Hence, the payoff matrix of the $DIR(K, c)$ game is given by (the $1/\rho$ in front of the matrix is for normalization reason)

$$\frac{1}{\rho} \times \begin{bmatrix} (1,1) & (1,-c) & (1,-c) & \cdots & (1,-c) \\ (2,1) & (2,2) & (2,-c) & \cdots & (2,-c) \\ (-c,1) & (3,2) & (3,3) & \cdots & (3,-c) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & (K-1,K-1) & (K-1,-c) \\ (-c,1) & (-c,2) & \cdots & (K,K-1) & (K,K) \end{bmatrix}. \tag{2}$$

Write down the payoff matrix for $DIR(4, 8)$, and show that the $DIR(4, 8)$ game has a unique Nash equilibrium and also a unique correlated equilibrium (correlated equilibrium will soon be covered in the lectures).[4]

Repeated task (b) above for $DIR(4, 8)$, $DIR(4, 16)$, $DIR(10, 15)$ and $DIR(10, 30)$.

Task II: Uncoupled No-Regret Learning Dynamics under Bandit Information. Repeated Task I above, but for the *bandit information* setup. That is, at each round t, player A can only observe the utility value $u^A(i^t, y^t)$ for the action i^t she played, and similarly player B observes only $u^B(x^t, j^t)$. In this case, you will need to implement EXP3 algorithm to handle such bandit feedback.
Your task: implement EXP3 with the five different choice of ϵ above, and answer the three questions as in Task I.

[4] This uniqueness holds for any $DIR(K, c)$ game actually, but for this project you only need to prove the uniqueness for the particular $DIR(4, 8)$ game.