MIT 6.S890 Topics in Multiagent Learning Fall 2025

Homework 2

Released on: Oct. 2, 2025 **Due on: Oct. 16, 2025, 11:59pm ET**

[*** Your name and MIT ID here ***]

Instructions This homework contains four problems (each split into smaller tasks), for a total of 100 points. Please attach a readable printout of your code for Problem 5 at the end of your submission.

Collaboration policy You are welcome to discuss the problems with other students. However you should write up the solutions by yourself.

1 Optimal Regret Bounds (10 points)

Problem 1.1 (10 points). We have seen several regret minimization methods whose regret over T rounds grows as $O(\sqrt{T})$. Consider a regret minimization setting wherein, over rounds $t = 1, \ldots, T$, a learning algorithm chooses a distribution x_t over two actions $\{a, b\}$ whose utilities $u_a^t, u_b^t \in [0, 1]$ are decided by an adversary after observing x_t. Show that there exists no regret minimization algorithm for this setting whose regret is $o(\sqrt{T})$.

★ Hint: You can use without proof the fact that an unbiased random walk that goes 1 step left with probability 1/2 and 1 step right with probability 1/2 will be at distance at least $\sqrt{T}/2$ from the origin with probability at least 1/2, for large enough T.

[*** your solution here ***]

2 Are all Coarse Correlated Equilibria Reachable? (10pts)

Consider a n-player game and assume for simplicity that, for all players i, $u_i(s) \neq u_i(s')$, for all action profiles s, s'. Let D be a coarse correlated equilibrium of this game.

Problem 2.1 (10pts). Show that there exist no-regret algorithms $A_1, \ldots, A_n$ for the n players of the game such that if, simultaneously over rounds $t = 1, 2, \ldots$, each player i uses algorithm A_i to iteratively update her strategies given the strategies played by her opponents, then the empirical distribution of action profiles played by the players converges to D, as $t \to \infty$. Note that you need two things: (i) each A_i must be a no-regret learning algorithm; and (ii) if all n players use their corresponding algorithm $A_1, \ldots, A_n$, then the empirical distribution of play converges to D. For *slightly* smaller credit you may assume that the probabilities used by D are rational.

[*** your solution here ***]

3 Dominated Strategies and Correlated Equilibria (10pts)

Recall that action a_i of player i in some game is *strictly dominated* by some mixed strategy x_i iff no matter what the other players play, player i receives higher utility by playing x_i rather than a_i. An action is called *strictly dominated* if it is strictly dominated by some mixed strategy. Show the following (4pts each):

Problem 3.1 (3pts). Show that a strictly dominated action cannot be in the support of a correlated equilibrium.

[*** your solution here ***]

Problem 3.2 (3pts). Give an example of a game where a strictly dominated strategy is in the support of a coarse correlated equilibrium.

[*** your solution here ***]

Problem 3.3 (4pts). We saw in class that no-regret learning dynamics converge to coarse correlated equilibria in general games. Now consider specifically the multiplicative-weights-updates (MWU) algorithm, where in each iteration t each player i is playing each action a_i with probability $x_i^t(a_i)$ that is proportional to:

$$x_i^t(a_i) \propto \exp\left\{\frac{1}{\sqrt{T}}\sum_{\tau=1}^{t-1} u_i(a_i, x_{-i}^\tau)\right\} \tag{1}$$

Show that as $T \to \infty$, the probability $x_i^T(a_i)$ of any striclty dominated action a_i converges to 0. Hence, in the limit MWU plays strictly dominated strategies with 0 probability.

[*** your solution here ***]

4 Equivalences and Separations between different game representations (20 pts)

Problem 4.1 (4 points). Prove that any finite normal-form game with N players, each having K available actions, can be represented as an extensive-form game. Your proof should be constructive—that is, you should describe a general procedure that takes the game's payoff structure as input and produces a valid game tree. How many leaves (terminal nodes) does this game tree have?

[*** your solution here ***]

Problem 4.2 (3 points). Apply your procedure to the game of Rock-Paper-Scissors. Draw the complete extensive-form game tree. Assume the standard payoffs: the winner receives +1, the loser receives -1, and a tie results in 0 for both players. Be sure to correctly draw and label all nodes, actions, payoffs, and the information set(s) for Player 2.

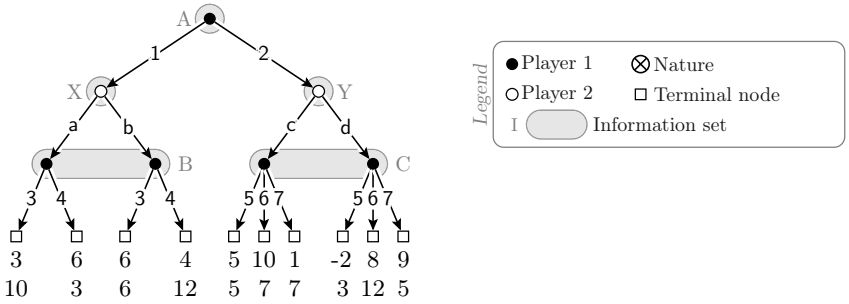
[*** your solution here ***]

The process can also be reversed, though it may lose information about the sequential structure of the game.

Problem 4.3 (4 points). Prove that any finite extensive-form game can be converted into a normal-form representation.

[*** your solution here ***]

Problem 4.4 (3 points). Consider the simple two-player extensive-form game shown below. The top number at each leaf node is the payoff of player 1 and the bottom number the payoff of player 2.



List all distinct pure strategies for Player 1 and Player 2. Then construct the normal-form representation of this game. The rows should correspond to Player 1's strategies and the columns to Player 2's strategies. The entries in the matrix should be the payoff tuples (u_1, u_2) that result from each strategy profile.

[*** your solution here ***]

Let us now analyze a special class of games. A *perfect-information sequential game* is one that can be represented by a game tree where all information sets are singletons (each player knows exactly which node they are at) and there are no chance or "nature" nodes. Games like Tic-Tac-Toe and Chess are classic examples. While normal-form games do not always have a pure Nash equilibrium (for example, think of Rock-Paper-Scissors), it turns out that perfect-information extensive-form games (for example, go or chess) always do.

Problem 4.5 (6 points). Prove that every finite perfect-information sequential game has a *pure* Nash equilibrium, that is, one in which every player uses a *deterministic* strategy (all probability mass placed on one action).

[*** your solution here ***]

5 Implementation of Learning in Extensive-Form Games (40 points)

In this problem, you will implement CFR with regret matching to solve three extensive-form games, which are Rock-Paper-Super-Scissors (RPSS) as well as two small poker variants: Kuhn poker [Kuhn, 1950] and Leduc poker [Southey et al., 2005], the latter being the larger of the two. Please note that RPSS is just rock-paper-scissors with modified payoffs, so you can solve it by hand and check your results with it first! We provide the extensive-form representation of the games in the attached zip file, as listed below:

5.1 Game file format

Each input file contains a description of the game tree across multiple lines, encoding all the information needed to reconstruct the game. The information lists all game states line by line, followed by all the relevant details. Each line describes either a game state or an information set.

5.1.1 Game state

We identify different game states using the sequence of actions taken by all players and the nature. The game history is formatted as:

/<P1>:<A1>/<P2>:<A2>/.../<Pn>:<An>/

where <Pi>:<Ai> represents an action taken, with:

- <Pi> identifying the acting entity: C for the chance (dealer), P1 for Player 1, and P2 for Player 2.
- <Ai> representing the action taken or the card dealt. Specifically:
  - When the action is taken by either player, <Ai> denotes an action such as check or call (c), raise (r), or fold (f).
  - When the action is taken by the environment, <Ai> represents a card.

There are three types of nodes: decision nodes, chance nodes, and terminal nodes.

Decision nodes Decision nodes represent points where a player must choose an action:

node <HISTORY> player <X> actions <A1> ... <An>

where:

- <HISTORY> represents the sequence of actions leading to this node from the root.
- <X> denotes the index of the player (1 or 2).
- <A1>, ..., <An> are the available actions for Player <X> at this node.

Chance nodes Chance nodes represent randomness in the outcome of the environment. In the specific case of poker, these nodes represent the stochastic outcome of dealing cards:

node <HISTORY> chance actions <A1>=<P1> ... <An>=<Pn>

where:

- <HISTORY> represents the sequence of actions leading to this node from the root.
- <A1>, ..., <An> are the possible actions taken by the chance player.
- <P1>, ..., <Pn> are the probabilities associated with each action, satisfying P1 + ... + Pn = 1.

Terminal nodes Terminal nodes represent the end of a game sequence, where payoffs are assigned to the players:

node <HISTORY> terminal payoffs 1=<Q1> 2=<Q2>

where:

- <HISTORY> represents the sequence of actions leading to this node from the root.
- <Q1> and <Q2> are the payoffs for Player 1 and Player 2, respectively.

5.1.2 Information sets

Information sets group nodes that are indistinguishable to a player due to incomplete information. Each information set is defined by:

infoset <NAME> nodes <N1> ... <Nn>

where:

- <NAME> is a unique identifier for the information set.
- <N1>, ..., <Nn> are the histories of the nodes within the information set.

5.2 Find the best response

When implementing learning algorithms, it is always a good start to implement the best-response oracle. This allows you to verify the performance of your algorithm.

Problem 5.1 (10 points). Implement a function that computes the best response for a given strategy. In both games, find the best response for Player 1 against a uniform strategy for Player 2, who plays uniformly over all valid actions at each decision node. Report the expected utility for Player 1. Then, compute the Nash equilibrium gap of the strategy profile in which both players play the uniform strategy. Recall that the Nash equilibrium gap of a strategy profile (x, y) is given by

$$\gamma(x, y) := \max_{x^* \in \mathcal{X}} u_1(x^*, y) - \min_{y^* \in \mathcal{Y}} u_1(x, y^*)$$

where $u_1(x, y)$ is the expected payoff for Player 1 when both players play strategies x and y, respectively.

5.3 Learning to best respond

After implementing CFR, a good way to verify its functionality is to check if it can learn the best response. In particular, you will verify that your implementation, applied to Player 1, learns a best response against Player 2 when Player 2 plays the *uniform* strategy at each information set.

Problem 5.2 (15 points). Implement CFR for Player 1 with regret matching at each decision node against a uniform strategy for Player 2. Compute the average strategies $(\bar{x}^T, \bar{y}^T)$ by averaging the *reach probability* in sequence form:

$$\bar{x}^{(T)} := \frac{1}{T} \sum_{t=1}^{T} x^{(t)}, \qquad \bar{y}^{(T)} := \frac{1}{T} \sum_{t=1}^{T} y^{(t)}. \tag{2}$$

Plot the expected utility (for Player 1) of the average strategies $(\bar{x}^T, \bar{y}^T)$ as a function of the number of iterations $T = 1, \ldots, 1000$. You will find the expected utility finally converge to the value reported in the previous problem.

Your solution should include three plots (one for each game). Don't forget to turn in your implementation.

5.4 Learning the Nash equilibrium

Now, it is time to find the Nash equilibrium in both games! The easiest way to do this is to run CFR for both players.

Problem 5.3 (15 points). Instantiate your implementation of CFR for both players to run against each other. Compute the average strategy for both players. Plot the Nash equilibrium gap $\gamma(\bar{x}^T, \bar{y}^T)$ and the expected utility $u_1(\bar{x}^T, \bar{y}^T)$ for Player 1 of the average strategies as a function of the number of iterations $T = 1, \ldots, 1000$.

Your solution should include six plots (two for each game—one for the Nash equilibrium gap and one for the utility). Don't forget to turn in your implementation.

6 Project Proposal Submission (10 points)

Problem 6.1 (10 points)**.** For this assignment, you are required to submit a concise proposal for your course project. Your proposal should clearly define the problem you aim to address, and must contain the following elements:

- Challenge type or project title
- Names of team members (between one to four members, more work expected for bigger groups)
- Main objectives of the project
- Summary of the potential approach
- How will you spend the project breaks

You are encouraged to choose any project that captures your interest. For those interested in empirical work, we recommend considering the "Team Leduc Poker Challenge". Detailed descriptions of these projects will be made available in the course modules.

References

H. W. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 550–558, 2005.