CSD 15-784 - Cooperative AI

Homework 2
(due Mar. 10 5:00pm US Eastern time)

Instructions

Submit your work on Gradescope. If you have not been added to the Gradescope course (with ID 976999), contact us. Show all the work you have done in the submission.

You may work alone or discuss with **one** other person, but you must follow the following rules or it will be considered cheating. If you discuss with another person, you **must** explicitly acknowledge that specific person on your writeup. Also, the only way in which you may work with another person is to work on a whiteboard together, and then when you are done discussing, to erase the whiteboard, without taking any notes or other record with you, other than what you remember. (Using the zoom whiteboard is allowed if you want to meet remotely.) **You should write up your code and your writeup alone.**

External tools, including but not limited to the use of generative AI, should generally be treated similarly to a person outside the course. If you happen to find it effective, you may use them, for example, to get more familiar with Python libraries or topics in the course in general. But in the end, you need to do your assignments on your own, without any help from these tools. You may not pass specific information from the assignments to these tools. (This is of course also good practice for exam questions, as you will not have access to such tools on exams at all.) To the extent you use these tools, you are also responsible for ensuring that information from these external tools makes sense; "I got this question on the exam wrong because ChatGPT told me something false while studying" is not a valid excuse. If you use external tools, you **must** explicitly acknowledge the extent to which you have used them.

1 Program Equilibrium (30 points.)

Consider the following n-player version of the Prisoner's Dilemma. For each player i, player i's set of pure strategies is $A_i = \{C, D\}$. The payoffs are given

by

$$u_i(a_1, ..., a_n) = \mathbb{1}[a_i = D] + \sum_{j \neq i} 2\mathbb{1}[a_j = C]/(n-1).$$

($\mathbb{1}[P]$ evaluates to 1 if P is a true proposition, and to 0 if P is a false proposition.) Intuitively, each player chooses between generating one unit of utility for herself by defecting, and generating two units of utility to be distributed equally across the other players by cooperating. The unique Nash equilibrium of this game is $(D, ..., D)$ for a utility of 1 for each player. Meanwhile, in $(C, ..., C)$, everyone's utility is 2.

Consider the following two programs from class that achieve (C, C) in program equilibrium in the case of $n = 2$: Cooperate with Copies and ϵ-grounded Fair Bot. For each of these programs, for $n > 2$, give a version of the program such that everyone using that program is an equilibrium, and the result of everyone using that program is the outcome $(C, ..., C)$. (Something counts as a version of CwC if it only checks for program equality without doing more sophisticated analysis of the program or simulating it. Something counts as a version of ϵ-grounded Fair Bot if all it does is simulate other programs with some probability, and it terminates in finite time with probability 1. You may assume there is a commonly agreed upon indexing of the players (say, 1, 2, 3) that is given as part of the input to the programs.)

3. Extensive-form games. (20 points.) Consider the game in Figure 1.

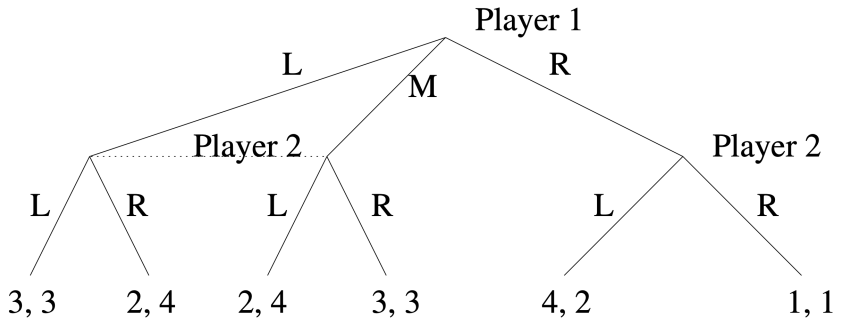

Figure 1: An extensive-form game with imperfect information.

a. (6 points) Give the normal-form representation of this game.

b. (6 points) Give a Nash equilibrium where player 1 sometimes plays Left. (Remember that you must specify each player's strategy at *every* information set.)

c. (8 points) What are the subgame perfect equilibria of the game? (Remember that you must specify each player's strategy at *every* information set.)