Sanaz Mostaghim, Alexander Dockhorn, Christian Wustrau

1. Exercise Sheet

Exercise 1 Expected Payoffs

Consider a game with the following payoff matrix:

$$A = \begin{array}{c} \\ C \\ D \end{array} \begin{array}{c} \begin{matrix} C & \quad D \end{matrix} \\ \begin{bmatrix} 0,0 & 1,1 \\ 0,0 & 1,1 \end{bmatrix} \end{array}$$

a) Compute the expected payoff for player 1 playing cooperate (C), when it is already known that player 2 plays a mixed strategy of $x = (1/2, 1/2)$.

b) Compute the expected payoff of player 1, when he plays a mixed strategy of $x_1 = (1/2, 1/2)$ and player 2 plays a mixed strategy of $x_2 = (1/2, 1/2)$.

c) Assume that both players currently play a mixed strategy of $x = (1/2, 1/2)$. What kind of strategy can player 1 play to increase his payoff? What kind of strategy can player 2 play to increase his payoff?

Exercise 2 Nash Equilibria

Consider a 2-Player Prisoner's Dilemma game with the following payoff matrix.

$$A = \begin{array}{c} \\ C \\ D \end{array} \begin{array}{c} \begin{matrix} C & \quad D \end{matrix} \\ \begin{bmatrix} 3,3 & 0,5 \\ 5,0 & 1,1 \end{bmatrix} \end{array}$$

What strategy profile is a Nash-Equilibrium?

Exercise 3 Nash Equilibria

Roger and Colleen play a game. Each one has a coin. They will both show a side of their coin simultaneously. If both show heads, no money will be exchanged. If Roger shows heads and Colleen shows tails then Colleen will give Roger 1 Dollar. If Roger shows tails and Colleen shows heads, then Roger will pay Colleen 1 Dollar. If both show tails, then they both get 2 Dollar.

a) Write the payoff matrix (for both players). Note: You can write in one matrix or in two matrices.

b) What is the best response of Colleen to Roger, when he plays/shows tails?

c) What is the Nash-equilibrium for this payoff matrix?

Exercise 4 Replicator Equations and Fixed Points

Consider the Stag-Hunt game with the following payoff matrix:

$$\begin{array}{c} \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cc} C & D \end{array} \\ \begin{bmatrix} 2,2 & 0,1 \\ 1,0 & 1,1 \end{bmatrix} \end{array}$$

a) Use replicator equations to calculate the fixed point(s), for the number of cooperators.

b) Show to which fixed point the population converges if the initial frequency of cooperators is 75%.

c) Show which fixed point the population converges to for any start-frequency of cooperators.

Sanaz Mostaghim, Alexander Dockhorn, Christian Wustrau

The following task can be solved in pairs of two. Please make sure that your solution includes the name of both group members as a comment at the top of the file

Exercise 5 Programming Exercise - Implementation of Mixed Strategies

We will try to make sure that these additional programming exercises can be solved without too much additional stress in setting up necessary systems. Each of the exercise sheet programming tasks will be based on Python (3.x) and may require the download of additional packages. To reduce the effort in setting up your own Python environment, we will also host all files as a Google Colab project, which lets you program and test the code in your browser.

Setting up your own Python Environment

- install Python 3.7 `https://www.python.org/downloads/`
- (recommended:) install PyCharm `https://www.jetbrains.com/pycharm/download/`
- install the packages *numpy*, *pandas*

Joining the Google Colab project

- Introduction to Google Colab:
  `https://colab.research.google.com/notebooks/welcome.ipynb`
- Join the Google Colab project:
  `https://drive.google.com/open?id=1rEyhJfshFD60Phw9Vpj08OB-4ufKBfxn`

Included files:

- *actions.py* – defines the actions Cooperate and Defect
- *agents.py* – defines the example agents:
  - AllCooperateAgent, AllDefectAgent, RandomAgent, GrudgerAgent, CopycatAgent, DetectiveAgent
- *game.py* – defines the game class, game objects handle easy access to payoff matrixes
- *game_simulation.py* – defines the *simulate_match_up* method
- *test.py* – small test script which lets you define your own agent and test it against others.

Task:

- Define your own agent by implementing the MyAgent methods.
- Try to score more points than the other agents in the provided Prisoner's Dilemma task.
- Test other scenarios (e.g. Stag-Hunt, Snowdrift, Harmony) and describe your results.