Computation and Economics - Spring 2012
Homework Assignment #01: Prisoner's Dilemma

Professor Sven Seuken
Department of Informatics, University of Zurich
Out Monday, February 27, 2012
Due **23:59** sharp: **Sunday, March 4, 2012**

Total points: 20. This is a single-person assignment. Points will be awarded for clarity, correctness and completeness of the answers. You are free to discuss the problem set with other students. However, you should not share answers.

Introduction

The Prisoner's Dilemma is a canonical example in game theory, which is often used to explore behaviors of people in a situation of strategic interdependence. In its basic version, the Prisoner's Dilemma is a one-shot game. Here, we consider the iterated (i.e., repeated) Prisoner's Dilemma (IPD), where the game is repeated over several rounds, and the players can make their decisions based on knowledge of the outcomes in previous rounds. Cooperation can occur in this version of the game because a player is able to punish a defecting opponent.

In his 1984 book *The Evolution of Cooperation*, Robert Axelrod described the results of a tournament he organized for academics to submit IPD strategies for a fixed number of steps. The winner of Axelrod's competition was the strategy "tit-for-tat" – simply choose the same move as was chosen by the opponent in the previous round. In general, Axelrod found that altruistic strategies triumphed over greedy ones.

For this problem set, you will create a submission to an IPD tournament like the one Axelrod set up in 1984 but with a twist. The information your player receives about its opponent's previous action is noisy, i.e. *with probability 0.05 your player receives incorrect information about the previous action of the other player*. The tit-for-tat strategy is not reliable in noisy environments. To understand why this is so, imagine that Alice is told that her opponent Bob defected, when in fact Bob collaborated. Alice would then defect, falsely punishing Bob, and the mistaken defections will echo until someone gets the wrong information once again. What do you think would be the best strategy to overcome a noisy environment for IPD? In discussing the complications that noise adds to his tournament, Axelrod said that "Noise calls for forgiveness, but too much forgiveness invites exploitation." How will you design a winning agent for such an environment?

Tasks

1. [**12 Points**] Designing your Agent

   Your task is to create an agent for the (finitely) iterated noisy Prisoner's Dilemma described above. In each round, there will be a 5 percent probability of misdetecting your opponent's move. The scores for each round are as follows:

	C	D
C	3,3	0,5
D	5,0	1,1

A strategy in this game can be modeled as a finite state automaton (FSA), as defined in chapter 3. An FSA is comprised of a set of states with associated actions, and transitions between these states. It has one start state, but can end up in any other available state.

For every state, you need to define the probability of cooperating as well as the new state to transition to based on the observed joint action. Remember that with probability 0.05 the action taken by the other player may be different from what you observe. Of course the observation of your own action is correct.

Format

Your agent should have the following format. There may be between 1 and 5 states, where each state has to be specified on a separate line as follows:

$i : P(cooperate)\ S_{CC}\ S_{CD}\ S_{DC}\ S_{DD}$,

where

- i is the id of the state, which must be 0, 1, 2, 3, 4 in that order,
- $P(cooperate)$ is the probability of cooperating when in this state (and thus one minus this value is the probability of defecting),
- S_{CC} is the state to move to on information that both agents cooperated in this round,
- S_{CD} is the state to move to on information that your agent cooperated and the other agent defected in this round,
- S_{DC} is the state to move to on information that your agent defected and the other agent cooperated in this round,
- S_{DD} is the state to move to on information that both agents defected in this round.

Note that state 0 is always the initial state. Your goal is to define probabilities and transitions for each of the states in the above format. You need not use all 5 states.

Example. Consider the following agent:

0: 0.6 1 0 0 1
1: 0.8 1 0 0 1

This FSA has two states (0 is always the initial state). When in state 0 the FSA will cooperate with probablity 0.6 and in two situations (if both players cooperated or both players defected) it will move to state 1. When in state 1, it will cooperate with probablity 0.8 and will return to state 0 if the players acted differently (one defected and one cooperated).

Testing

For your convenience we have provided a mechanism to test your agents. At the website http://pall.ifi.uzh.ch/pd/compete.php, you can enter two agents and run a simulation for a specified number of rounds (not more than 100). The agents must be entered in the above format. The simulation result reports the round number, what action was taken by the two agents, what actions were observed (because the data is noisy this may be different than the actual actions taken), the score after the round, and what states were moved to at the end of the round.

2. [**6 Points**] Analysis

   Explain in a few sentences how your agent works and why you designed it that way. How do you think it will do against the agents of the other students?

3. [**2 Points**] Competition

   We will run each submitted agent against each of the others 10 times, with each game lasting 100 rounds. We will rank agents by their cumulative score, and each agent that is ranked in the top third gets 2 points, the others get 0.

Submission format: Please use the webform at the following address to submit the assignment: http://tinyurl.com/6r3caw8 There is a field for your name, email address, agent and explanation. Again, each person must submit a separate agent and explanation.