Simulation of Complex Systems

Homework 2: Evolutionary Games CA

Assessment date: November 14 2014

In this exercise we study the effects of space on an evolutionary model of cooperation, based on the finitely iterated prisoners dilemma in a cellular automaton formulation.

The prisoners dilemma[1] is a two-player, non-zero-sum game that is often used as a model of the problem of cooperation. Its main feature is that the best individual actions for each player, defection, causes them to end up in a situation that leaves them worse off than what they would achieve if they could cooperate.

In the standard story[2], we have two partners in crime in jail. They are kept separately and cannot communicate. They both get an offer: if they confess and witness (known as defecting) against their partner they will be released without punishment and the partner gets five years in prison, unless the partner also confesses in which case they both get three years. If none confesses (cooperates), they will both get a one year sentence.

For each individual, it is clearly best to defect: if their partner cooperates (does not witness) they will go free instead of getting a one year sentence, while if he/she defects they will get three years instead of five. However, as the game is symmetric both will follow this logic and end up in the Nash equilibrium of both defecting with both getting three years, as opposed to the one year they would get if they could cooperate.

[1] For an entertaining listening that relates to the prisoner's dilemma, check out this episode of Radiolab.

[2] ...which doesn't really drive home the point as there still are a lot of extra considerations real prisoners would have towards each other, their own reputation and so on. For an example of an effort to get at the (in the author's words) "*true* prisoners dilemma", see Yudkowski's short sci-fi story Three Worlds Collide.

For a finitely repeated game, where the number of rounds is fixed, the Nash equilibrium is still to defect from the first round. This is because you certainly should defect on the last round as this is the same situation as in the one-shot version of the game. But knowing this, you should also defect on the next-to-last round, and thus also on the one before that etc. This way of reasoning is called backward induction.

In this model we will use a reduced strategy space that highlights the backward induction. This space contains strategies on the form "Cooperate until round i or the other player defects, whichever comes first, and then defect for the rest of the game." We denote these strategies $S(i)$, where in a game of N rounds i runs from 0 (always defect) to N (always cooperate).

Use a two dimensional cellular automaton (CA) approach, where each cell has a state $i \in \{0, 1, \dots N\}$, representing a player with strategy $S(i)$. Each time-step has two stages: competition and reproduction. The competition step has all players playing the N-round IPD with their von Neumann-neighbors[3] and thus each cell gets a total score. This score is then used in the reproduction step by replacing any cells' state with that of its neighbors (including itself) that received the highest total score (break ties randomly). The reproduction step is finished by mutating each cell into a random different state with some small probability ϵ.

Note that in a strict CA formulation this would be a range-2 interaction, as a cell's new state depends on its neighbors' score, which in turn depend on how these neighbors played against their neighbors. However, this is a side note as the simplest way to program the model is to use the above formulation with two steps.

We will denote the payoffs for the different scenarios as: T (temptation) for the payoff for defecting when the opponent cooperates; R (reward) when both cooperate; S (sucker) for cooperating with a defector; and P (punishment) when both defects. To make this a prisoner's dilemma, rather than some other two-person game like the hawk-dove game, we need to have $T > R > P > S$[4]. However, we don't need the full four parameters. In

[3]The nearest von Neumann neighbors of a cell is the four cells to its immediate right, left, top, and right.

[4]For the iterated prisoners dilemma we in general also need $2R > T + P$ so that taking alternative turns cooperating and defecting doesn't beat the pure cooperation. In our case however, this is a strategy that isn't available due to the reduced strategy space, so we disregard this requirement.

the evolutionary dynamics above, only ranking matters and thus adding or multiplying each parameter with a constant gives the same dynamics. We can thus use a reduced parameter space where $R = 1$ and $S = 0$. The requirements then becomes $T \in [1, 2]$ and $P \in [0, 1]$.

Examination: Work in your assigned groups. During lab hour, either 14/11 or the extra lab 17/12, you should together demonstrate your results to a tutor in the way indicated at the respective exercise. (We might be a bit flexible here, in the sense that you can have your work assessed also in the other lab sessions if you have a compelling reason for doing so. But sticking to the schedule is the preferred options as that will make things run more smoothly). Also email one set of code per group to kolbjorn@chalmers.se with "SoCS HP2" and your group number in the subject.

Make sure you go through your demonstration by yourself before so that everything works. Everyone involved will appreciate the reduced queue times. Feel free to show just a subset of the exercises if you haven't done them all (whether you plan to do so later or not).

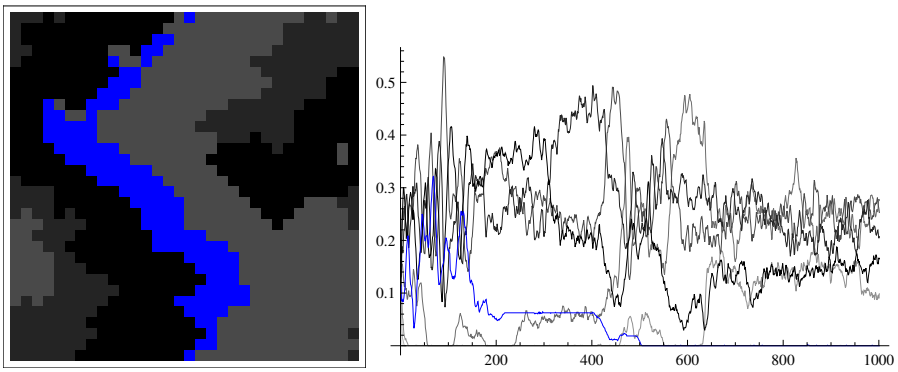

Figure 1: A snapshot of the CA and the time evolution of its population fractions for $T = 1.5$ and $P = 0.5$ with $N = 7$ rounds. Blue is all cooperate ($S(7)$), black to lighter gray is $S(6)$ to $S(0)$ (only $S(7)$ to $S(4)$ present in the snapshot).

Exercises:

1. Implement the basic model and visualize it. Also plot the time evolution on the population level, see Fig. 1. For the general parameters

you can use a lattice of size $L = 32$, a mutation rate $\epsilon = 0.1/L^2$, and a seven round game.
To demonstrate: The visualization and the population plot. **(9p)**

2. The model has several regimes depending on the score parameters of the game. Find the three main regimes and characterize them. Do you get persistent cooperation for any parameter values? (There is at least a fourth regime; can you find it?)
   To demonstrate: Visualizations of the CA in each regime and a description of how they differ. **(7p)**

3. Map out the phase diagram of the model. Grid the $T, P-$square with something like 10 interior points in each dimension and determine what phase (regime) the model assumes at each point. You can do the identification by just looking at the population plots if you wish. If you want to do something automatic, you need to define more than one measure (order parameter) as you try to identify more than two phases.
   To demonstrate: The phase diagram with some suitable coloring for the different phases. **(9p)**