Week Four - Prisoner Dilemma

This is like the game drift in some circumstances. Two individuals who can cooperate, C or defect; D C \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix}

Clearly if both cooperate they get more than if they both defect; however if one defects, then he will do better than if he doesn’t. If he cooperates, he will obviously defect. Indeed, D is a strict Nash equilibrium. Clearly “natural selection” favors defection since one defector will be more fit and can eventually invade and take over. In experimental game theory, people are more likely to cooperate than would be predicted by the PD. Rational players will always defect, so one of the big questions in theoretical and evolutionary biology is how does cooperation evolve into a population.

Consider the general matrix

Prisoner Dilemma C \begin{pmatrix} R & S \\ T & P \end{pmatrix}

R = reward, T = temptation, P = punishment, S = sucker

In reality, the game is not played once, rather it is played many times.

So that if you defect, maybe you will be known as a jerker so that others will never cooperate. Consider the strategy GRIM = they cooperated on the first move and then cooperate as long as opponent doesn’t defect. Once opponent defects, there is no forgiveness and you always D. Consider GRIM vs ALLD, suppose you play exactly m rounds. Here is the payoff:

\begin{pmatrix} G-R \quad A-L-D \end{pmatrix} \begin{pmatrix} m R \\ S + (m-1) P \end{pmatrix}

\begin{pmatrix} T + (m-1) P \\ m P \end{pmatrix}
If \( M > T \), then GRIM is a S.N.E. Note that ALLD is also S.N.E.

Since \( M_p > 5 + (m-1)p \) as \( p > s \), GRIM is S.N.E. if \( m > \frac{I-P}{n-p} \)

for our model \( m > 2 \). This seems to be a way to stabilize C.

Since ALLD is also S.N.E., need a way to establish C in a world of D

Even worse - if we know there are exactly \( n \) rounds then why not defect in the last round. Let's call this GRIM*.

GRIM: \( (R, R) \)

GRIM*: \( (m-1)R + T \)

\[ \text{suckered last round} \]

\[ \text{small gain last round} \]

Since \( T > R \), GRIM is not a N.E. Since \( p > s \), GRIM* is a S.N.E., so will take over. But then once every one defects on last round, why not defect on penultimate round, etc until once again ALLD!

Humans do not use this strategy, perhaps because you never know when you might run into someone again.

Let's do repeated R.D. with variable # rounds. After each round, play \( w \) at another round. Expected # rounds is \( \bar{m} = \frac{1}{1-w} \) (prove this! \( P(1) = 1-w \)
\( P(2) = w(1-w) \)
\( P(3) = w^2(1-w) \)
\( P(n) = w^{n-1}(1-w) \)

\[ \bar{m} = 1 - w + 2w(1-w) + 3w^2(1-w) + \cdots + (1-w) \left[ \frac{1 + 2w + 3w^2 + \cdots}{1-w} \right] = (1-w) \left[ \frac{\frac{1}{1-w} - 1}{1-w} \right] = \frac{1}{1-w} \]
Thus, we have:

$$\begin{pmatrix}
\overline{m} \; r \\
\overline{m} \; (s + (\overline{m} - 1) \; p)
\end{pmatrix}
\begin{pmatrix}
T + (\overline{m} - 1) \; p \\
\overline{m} \; p
\end{pmatrix}
\begin{pmatrix}
G \; R \; I \; M \\
A \; C \; L \; L \; O
\end{pmatrix}
\begin{pmatrix}
1 - p \\
p
\end{pmatrix}
$$

But if \( \overline{m} \) is large enough, the strategy will be to use ALLO as well. Is NIM the best strategy? What if someone defects only once to test if he can get away with it? No reconciliation → ALLO, so maybe NIM is a better strategy.

Set of all strategies is infinite, naturally. Could use strategy which depends on history, e.g.,

Each round has 4 possible outcomes: CC, CD, DC, DD

So there are \( 4^2 = 16 \) strategies. That looks at last round only: 0000 always defect | 0001 always defect | 0010 always C | 0011 always D

else 1111 always C, 216 strategies using last 2 moves: \( 2^{n-2} \) that use last \( n \) moves. Rank them.

Strategy span \( 4^{n-2} \) strategy space, \([0,1] \times \cdots \times [0,1]\)

Axelrod set up a bunch of tournaments—people submitted strategies and all played each other multiple times (not knowing whom they ended). Winner had highest total payoff. TFT won out of 14 entrants: C on first move, then answer C for C and D for D.
In a real tournament 63 strategies against TFT won. TFT never tried to get more than 0 points, but overall matches TFT won. On average it does better against X than DERS OR against X.

\[
\begin{align*}
\text{TFT} & \quad \text{ALLO} \\
T & \quad mR \quad S + (\bar{m} - 1)P \\
\bar{m} & > \frac{1 - l}{R - P}
\end{align*}
\]

(Like C-RII against ALLO)

In all interactions, there are sometimes mistakes so that may be someone wants to push the "C" button. Pushing the "D" TFT cannot correct mistakes:

\[
\begin{align*}
C & \quad C \quad C \quad D \quad C \quad D \quad O \quad D \quad O \\
& \quad \text{could fall back} \\
C & \quad C \quad C \quad D \quad C \quad O \quad D \quad O \quad D \quad O \\
& \quad \text{of course!}
\end{align*}
\]

In the end, two TFT players with small mistakes have same payoff as players going randomly:

\[
A(TFT, TFT)_{\text{noise}} = \frac{R + T + P + S}{4}
\]

Since \( R > \frac{T + S + R + P}{2} \), Average < R

So TFT is weak in sense of noise

\[
2.25 = \frac{9}{4} = <P_{\text{E}}>
\]

TFT + Noise can be invaded by TFT strategy

Another weakness is

That ALL C VS TFT is

\[
\begin{align*}
(mR & \quad mR) \quad \text{neutrally stable} \\
\bar{m} & \quad \bar{m} \quad \text{so could drift to}
\end{align*}
\]

ALL C which is takeover by ALLO
Reactive Strategies

\[ p, q, P = \text{prob of cooperating if opponent } C \]
\[ q = \text{prob of } C \text{ if opponent } D \]
Have short memory, but can parametrize TFT \((1,0)\), ALLC \((1,1)\), ALLD \((0,0)\).

\(S(p,q)\) is a point in unit square. One last corner is No world \((0,1)\) where \(C\) when opponent \(D\) \& \(D\) when opponent \(C\).

Repeated PD as a Markov Chain on 4 states:

- CC
- CD
- DC
- DD

Two para:

Let there be 2 strategies \(S_1: (p_1,q_1)\) and \(S_2: (p_2,q_2)\), so for example CC \(\rightarrow\) CC if both cooperate on next move: \(p_1, p_2\)

\[
\begin{pmatrix}
CC & CD & DC & DD \\
CC & P_{11} & P_{12} & (1-Q_1)P_2 & (1-Q_1)(1-Q_2) \\
CD & Q_1P_2 & Q_1(1-Q_2) & (1-Q_1)P_2 & (1-Q_1)(1-Q_2) \\
DC & (1-Q_2)P_1 & (1-Q_2)Q_1 & (1-Q_1)P_2 & (1-Q_1)(1-Q_2) \\
DD & Q_2P_1 & Q_2(1-Q_1) & (1-Q_2)P_1 & (1-Q_2)(1-Q_1)
\end{pmatrix}
\]

\[ M_{ij} = \text{prob state } i \rightarrow \text{state } j \]

A stochastic matrix \(M\) a matrix with nonnegative entries whose row sums are 1. We define a SM, \(M\) as regular or irreducible if \(\exists \lambda > 0 \text{ s.t. all entries of } M^\lambda \text{ are positive.}\)

This means if we start from node \(i\) to node \(j\) if \(M_{ij} > 0\).

The length from \(i\) to \(j\) for all \(\lambda > 0\) (i.e., in finite steps).
Clearly $M \mathbf{1} = \mathbf{1}$ so 1 is an eigenvalue. Let $\mathbf{v}$ be the right eigenvector of $M$ with $M^T \mathbf{v} = \mathbf{v}$. Theorem (Perron-Frobenius) says all entries of $\mathbf{v}$ are positive. Further more $\lambda = 1$ is the principal eigenvalue of $M$. If $\lambda_1$ s.t. $M \mathbf{v} = \lambda_1 \mathbf{v}$ for $\lambda_1 < 1$. Suppose $M$ is irreducible. Check (4.3.1) below.

Let $\mathbf{X}_t = \begin{pmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{kt} \end{pmatrix}$ be the probability state at time $t$. Write $\mathbf{X}_t = \begin{pmatrix} c_1^T \mathbf{v}_1 + \cdots + c_k^T \mathbf{v}_k \end{pmatrix}$

$\mathbf{X}_{t+1} = M^T \mathbf{X}_t$. Write $\mathbf{X}_t = \begin{pmatrix} c_1^T \mathbf{v}_1 + \cdots + c_k^T \mathbf{v}_k \end{pmatrix}$

$\Rightarrow c_j^{t+1} = \lambda_1 c_j^t$ \Rightarrow $\mathbf{c}^T \mathbf{v}_1$ at $t \rightarrow \infty$

Unless $j = 1$, so $\mathbf{v}_t \rightarrow \mathbf{v}_1$ as $t \rightarrow \infty$. So stationary density $\mathbf{v}_1 \rightarrow \mathbf{v}_1$

We can compute $\mathbf{v}_1$. If $r_1 = p_1 - q_1$ and $r_2 = p_2 - q_2$ then $r_i \in (0, 1)$

$\mathbf{v}_1 = \begin{pmatrix} s_1 s_2, s_1 (1-s_2), (1-s_1) s_2, (1-s_1) (1-s_2) \end{pmatrix}$

$s_1 = \frac{q_2 r_1 + q_1}{1 - r_1 r_2}$

Clearly $E(S_j S_k) = r_1 s_1 s_2 + s_1 s_1 (1-s_2) + T (1-s_1) s_2 + \mathbb{P}(1-s_1) (1-s_2)$

If $\mathbb{P}(r_1 r_2) = 1$ or 2, then get deterministic strategies.

**Experiment.** Generate $n = 100$ randomly chosen reactive strategies. E.g. choose $p_{2, 1}$, compute the next payoff matrix using above $100 \times 100$.

Assume $x_j(0) = \frac{1}{10}$ and solve for equations in this. Watch the evolution.