Stochastic Prisoner's Dilemma

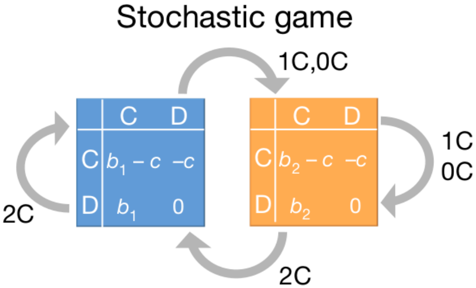
January 1, 2020

Game Rules

	C	D
C	$b_1 - c$	$-c$
D	b_1	0

Game Rules

	C	D
C	$b_1 - c$	$-c$
D	b_1	0



Strategy

1. Choose action based on history.

Strategy

1. Choose action based on history.
2. 1-Memory Strategy: just remember the last actions.

Strategy

1. Choose action based on history.
2. 1-Memory Strategy: just remember the last actions.
3. Example:
   $AllD = [CC- > 0, CD- > 0, DC- > 0, DD- > 0]$
   $WSLS = [CC- > 1, CD- > 0, DC- > 0, DD- > 1]$

Strategy

1. Choose action based on history.
2. 1-Memory Strategy: just remember the last actions.
3. Example:
   $AllD = [CC- > 0, CD- > 0, DC- > 0, DD- > 0]$
   $WSLS = [CC- > 1, CD- > 0, DC- > 0, DD- > 1]$
4. Mixed Strategy:
   $S = [CC- > p_1, CD- > p_2, DC- > p_3, DD- > p_4]$

Strategy

1. Choose action based on history.
2. 1-Memory Strategy: just remember the last actions.
3. Example:
   $AllD = [CC- > 0, CD- > 0, DC- > 0, DD- > 0]$
   $WSLS = [CC- > 1, CD- > 0, DC- > 0, DD- > 1]$
4. Mixed Strategy:
   $S = [CC- > p_1, CD- > p_2, DC- > p_3, DD- > p_4]$
5. ϵ chance to screw up.

Evolution

1. Repeated Game: Agents play with their neighbors; accumulate Π

Evolution

1. Repeated Game: Agents play with their neighbors; accumulate Π (Can be solved analytically)

Evolution

1. Repeated Game: Agents play with their neighbors; accumulate Π (Can be solved analytically)
2. L learns from R, with $\rho = \frac{1}{1+e^{-\beta(\Pi_R - \Pi_L)}}$.

Evolution

1. Repeated Game: Agents play with their neighbors; accumulate Π (Can be solved analytically)
2. L learns from R, with $\rho = \frac{1}{1+e^{-\beta(\Pi_R - \Pi_L)}}$.
3. Or Mutates with γ

Projects

1. Network Effect

Projects

1. Network Effect
2. Neighbor Effect in 2D grid.

Projects

1. Network Effect
2. Neighbor Effect in 2D grid.
3. Prisoner in the Prison. (Segregation)

Projects

1. Network Effect
2. Neighbor Effect in 2D grid.
3. Prisoner in the Prison. (Segregation)
4. "What happens when an unstoppable force meets an immovable object?"

Guideline

1. Start with simple tasks.

Guideline

1. Start with simple tasks.
2. Use Sphinx and Git.

Guideline

1. Start with simple tasks.
2. Use Sphinx and Git.
3. Team Work!

Analytical Calculation for repeated games

States

1. We define a state $v_t = \{w_t, {a_t}^{(1)}, {a_t}^{(2)}\}$ (t : time-step) based on: The world we are currently in, the action of the first player and the action of the second player.
2. For example 1CD is when where are in the good world, player one cooperates and player two, defects.
3. Overall we have 8 states: [1CC, 1CD, ..., 2CC, 2CD, ..., 2DD]

Analytical Calculation for repeated games

Transition Matrix

1. The probability of going from $v_1 = \{w_1, {a_1}^{(1)}, {a_1}^{(2)}\}$ to $v_2 = \{w_2, {a_2}^{(1)}, {a_2}^{(2)}\}$ is dependent on the strategies ($S^{(1)}$ & $S^{(2)}$) of the two individuals, and also world transition Q (how the world reacts).
2. $P_{\{w_1,{a_1}^{(1)},{a_1}^{(2)}\}\rightarrow\{w_2,{a_2}^{(1)},{a_2}^{(2)}\}} = S^{(1)}({a_2}^{(1)}|{a_1}^{(1)}, {a_1}^{(2)}) \times S^{(2)}({a_2}^{(2)}|{a_1}^{(1)}, {a_1}^{(2)}) \times Q(w_2|{a_1}^{(1)}, {a_1}^{(2)})$
3. P can be described as a matrix.
4. State v can be described as a vector.
5. Multiplying v by P will take the state to the next time-step.

$$
P = \begin{array}{c} \\ 1CC \\ 1CD \\ ... \\ 2DD \end{array}
\begin{array}{c} \begin{array}{cccc} 1CC & 1CD & ... & 2DD \end{array} \\
\begin{bmatrix}
P_{\{1CC \to 1CC\}} & P_{\{1CD \to 1CC\}} & \cdot & P_{\{2DD \to 1CC\}} \\
P_{\{1CC \to 1CD\}} & P_{\{1CD \to 1CD\}} & \cdot & P_{\{2DD \to 1CD\}} \\
. & . & \cdot & . \\
P_{\{1CC \to 2DD\}} & P_{\{1CD \to 2DD\}} & \cdot & P_{\{2DD \to 2DD\}}
\end{bmatrix} \end{array} \tag{1}
$$

$$
v_{t+1} = P v_t \tag{2}
$$

1. Where is the final State?
2. When we get to state $v_* = P v_*$ (so we are at a stationary state.)
3. Because of the ϵ it can be proven that such state exists.
4. v_* is the eigenvector of P for eigenvalue 1.
5. After we get to v_* we will stay there forever, so all of the reward should be calculated based on v_*

Analytical Calculation for repeated games

Overview

1. We construct transition matrix P based on two strategies.
2. We calculate its eigenvector for eigenvalue 1 and call it v_*
3. v_* is the stationary state, so we calculate each player's reward based on this state.

References

1. C Hilbe, imsa, K Chatterjee, MA Nowak, Evolution of cooperation in stochastic games, Nature 559 (7713), 246
2. Laura Hindersin, Bin Wu, Arne Traulsen, and Julian Garca, Computation and Simulation of Evolutionary Game Dynamics in Finite Populations
3. Fudenberg, D. & Imhof, L. A. Imitation processes with small mutations. J. Econ. Theory 131, 251262 (2006).
4. Hilbe, C., Martinez-Vaquero, L. A., Chatterjee, K. & Nowak, M. A. Memory-n strategies of direct reciprocity. Proc. Natl Acad. Sci. USA 114, 47154720 (2017). 38. Stewart,