Drunk Game Theory Implementation

Cole Mathis

March 12, 2016

1 The Game of the Pub

In this version of DGT there are two well defined games, the sober game, and the drunk game. These represent two different perspectives on the same interaction. Both games are defined by a payoff matrix, which is of the form:

		Player 2	
		O	S
Player 1	O	(A, W)	(B, X)
	S	(C, Y)	(D, Z)

As per usual, this payoff matrix represents the outcomes for Player 1. The values of $\{A, B, C, D\}$ represent the payoffs for those outcomes, while the values of $\{W, X, Y, Z\}$ represent the beers consumed for those outcomes. The payoffs of individuals DO NOT effect their perception of the game (i.e. payoffs do not change alpha level). The number of beers consumed DOES effect the perception of the game, (i.e. drinking beers changes the alpha level). Separating these two values allows us to consider situations where everyone is receiving the similar object (beer, money, conversation, sex), but the worth of that object can vary. The object, which we refer to from now on as **beer**, should be fixed by some shared reality, while the worth of such an object, which we refer to from now on as **payoff**, can change depending on the state of the individual. The values of $\{A, B, C, D\}$ and $\{W, X, Y, Z\}$ can be changed (for both the drunk and sober perspectives) can be changed *Parameters.py* file. Based on our current theory, the values $\{W, X, Y, Z\}$ should be the same for both the drunk and sober games.

When two players engage in the game of the pub, they might think they are playing either the drunk game or the sober game. When we refer to the sober game, we are referring to the low alpha value game, while the drunk game corresponds to the high alpha value game. Each player has their own perspective. The probability of a player playing the sober game is given by $1 - \alpha$, while the probability of playing the drunk game is given by α. The players receive pay-offs and beers based on the game they thought they were playing (i.e. my opponents perception of the game does not effect my perception of the outcome). As we decided during the Ariel sessions, the updating rule for a players alpha level, after b beers is:

$$\alpha_t(b) = (1 - \alpha_{t-1})\frac{\kappa}{r}b - (1 - d)\alpha_{t-1} \tag{1}$$

κ respresents the strength of the alcohol being consumed while d controls how quickly players sober up, r is set by the maximum number of beers consumed in any given interaction.

2 Population Structure

There are 4 options for structured populations. These different topologies can be toggled in the *Parameters.py* file.

1. Complete Graph/ Well - mixed Population:'*None*'
2. Erdos-Renyi Random: '*Random*', *edge_p* is the probability of any two nodes having an edge
3. Barabasi-Albert Graphs: '*BA*', '*barabasi_albert_m*' is the number of edges to attach to each node during the generating process, see
4. Groups: '*Groups*', '*group_p*' gives the probability of playing the game with someone in your own group.

For (1-3) the topology of the graph is fixed during the evolution of the game (this can be changed in the future). For the case of group dynamics, an individual will talk to someone outside the group more often when she has a higher alpha shown with equation 2. However the person in the other group will reject the interaction with a probability proportional to the difference between the two players alpha levels. given by equation 3.

$$p_{out}(\alpha) = (1-\alpha)p_0 + \frac{\alpha}{2} \tag{2}$$

$$p_{reject}(\alpha_1, \alpha_2) = |\alpha_2 - \alpha_1| \tag{3}$$

3 Strategy Updating Rules

Currently strategies are updated **asynchronously**, using **multiple replicator rule** outlined in Roca et al. [2009]. A randomly selected player will maintain her strategy with probability P given by 4. The product if over all of her neighbors, this is therefore a local updating rule. Φ is set to ensure normalization, again according to Roca et al. [2009].

$$P = \Pi(1 - p_{ij}) \tag{4}$$

$$p_{ij} = \begin{cases} (W_j - W_i)/\Phi & W_j > W_i \\ 0 & W_j \leq W_i \end{cases} \tag{5}$$

$$\Phi_{i,j} = \max(k_i, k_j)(\max(A, C) - \min(B, D)) \tag{6}$$

References

Carlos P Roca, José A Cuesta, and Angel Sánchez. Evolutionary game theory: Temporal and spatial effects beyond replicator dynamics. *Physics of life reviews*, 6(4):208–249, 2009.