Supergametools: some basic background information

Author: Benjamin Tengelsen[1]
Date: May 11, 2013

1 What is the set of supergame equilibria?

In many economic models, a decision maker (agent) seeks to make an optimal choice from a set of possible actions. A *game* is a scenario in which several agents make decisions, and the outcome of a given agent's decision depends on the decisions made by the other agents. A classic example is the "prisoners dilemma," represented in the following table:

		Person A	
		Collude	Defect
Person B	Collude	(1,1)	(0,4)
	Defect	(4,0)	(2,2)

Here two partners in crime are being interrogated by police in separate rooms. Each must decide whether to testify against the other (defect) or remain silent (collude). If they both collude, they each spend 1 year in prison. If one defects, and his partner colludes, the defector goes free while his partner is imprisoned for 4 years. If they both defect, they each spend 2 years in prison. In this game, the equilibrium outcome is for both individuals to defect. In the corresponding *supergame* (when this game is played repeatedly), the set of equilibria is much larger and diverse.[2] Due to its size and complexity, solving for this set analytically is not feasible and requires a numerical approximation.

Supergametools is a python library that implements an algorithm given in Judd, Yeltekin, and Conklin (2003) to approximate the set of supergame equilibria for a two-person game. It uses a monotone operator on convex polytopes to find set of equilibria as a subset of an initial guess.

2 Some Mathematical Definitions

Let S_i be the set of all possible strategies for agent i, and u_i be a function that maps the strategies of all agents to the payoffs given to individual i. A Nash equilibrium is defined to be the set of strategies $(s_1^*, s_2^*, \dots, s_n^*)$ where $s_i \in S_i$, such that for each player i,

$$\forall s_i \in S_i,\ u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

where s_{-i} is the set of strategies for all players except i. With this definition, we can express any nash equilibrium as the maximizing argument to the following optimization problem:

$$s_i^* = \arg\max_{s_i} u(s_i, s_{-i}) \text{ such that } s_i \in S_i$$

or, for a repeated game

$$\max_{\{s_i\}_{t=1}^{\infty}} \sum_{t=1}^{\infty} \delta^{t-1} u(s_i, s_{-i}) \text{ such that } s_i \in S_i$$

where $\delta \in [0, 1)$ weights future payoffs less than current payoffs. When $\delta = 0$, the set of supergame equilibria will be the same as the 1-period game. When δ increases, the number of equilibrium outcomes also increases.

[1]Tepper School of Business, Carnegie Mellon University. Email: btengels@cmu.edu
[2]See Fudenberg and Maskin (1986), Ely and Välimäki (2002), or Mailath and Samuelson (2011) for proofs.

3 Numerical Algorithm

We make use of the inner/outer hyperplane approximation algorithms given in Judd, Yeltekin, and Conklin (2003). Denote the set of possible choices as A, which is finite. Let W be the set of payoffs resulting from the choices in A, and let $B(\cdot)$ be our monotone operator. Finally, for $Z \subset W \subset R^n$, let $co(Z)$ be the inner approximation to W generated by Z.

Inner/Outer Hyperplane Approximation

Input: Vertices $Z = \{z_1, \ldots, z_M\}$ such that $W = co(Z)$.

(Inner/Outer) Step 1: Find extremal points of $B(W)$. For each search subgradient $h_\ell \in H, \ell = 1, \ldots, L$.

(a) For each $a \in A$, solve

$$c_\ell(a) = \max_w h_\ell \cdot [(1-\delta)\Pi(a) + \delta w], \qquad \text{such that}$$
$$(i) \;\; w \in W$$
$$(ii) \;\; (1-\delta)\Pi^i(a) + \delta w_i \geq (1-\delta)\Pi_i^*(a_{-i}) - \delta \underline{\mathrm{w}}_i$$

(b) Find the best action profile $a \in A$ and corresponding continuation value:

$$a_\ell^* = \arg\max\{c_\ell(a) | a \in A\},$$
$$z_\ell^+ = (1-\delta)\Pi(a_\ell^*) + \delta w_\ell(a_\ell^*)$$

(Inner) Step 2 : Collect the set of vertices $Z^+ = \{z_\ell^+ | \ell = 1, \ldots, L\}$, and set $W^+ = co(Z^+)$.

(Outer) Step 2 : Collect the set of vertices $Z^+ = \{z_\ell^+ | \ell = 1, \ldots, L\}$, and construct a hyperplane through each point with normal vector h_ℓ. The output polytope is the space enclosed by these hyperplanes.

The algorithm is repeated until the vertices Z converge, such that error between iterations is less that a given value of ε.

Supergametools features parallelized functions for the outer approximation algorithm. It allocates search gradients in Step 1 to the parallel processes, and improves the speed of computation considerably (see examples below). **The optimal scenario is when the number of processes is equal to (or slightly greater than) the number of search gradients.**

4 Examples

4.1 Cournot Duopoly

Suppose in a given market there are 2 firms. The firms choose how much of a given good to produce each period, which influences the market price according to the simple function:

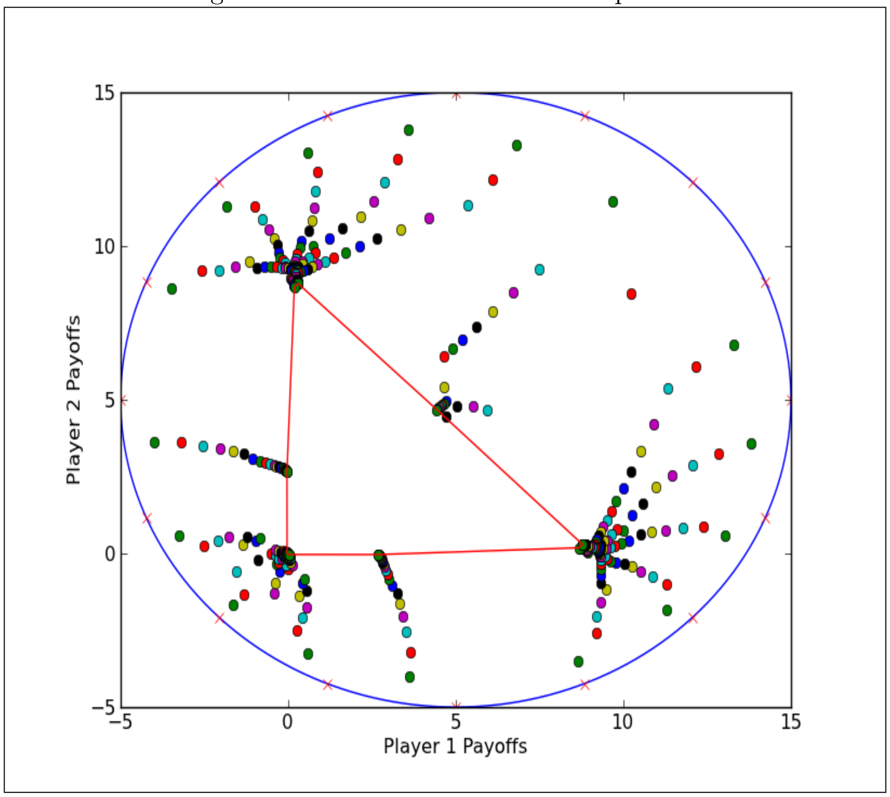
$$P = \max\{6 - q_1 - q_2, 0\}.$$

Let $\Pi_i = q_i(P - c_i)$ denote the single-period profits of firm i, where c_i is the cost of producing q_i. Firms aim to maximize their stream of profits over time:

$$\max_{q_{i,t}} \sum_{t=1}^{\infty} \beta^{t-1} \left[q_i(p - c_i) \right]$$

Consider the discrete case where q_i is one of n equally spaced points over the interval [0,6]. This gives a $n \times n$ payoff matrix. Applying the Outer Hyperplane Approximation of this payoff matrix for $n = 15$ gives the following figure:

Figure 1: Cournot Game - Set of Equilibria



Here the red triangle is the set of all possible equilibria. The colored dots give the vertices of this set with each iteration of the algorithm. The red x's' along the circle denote our initial search points. As shown in the table below, the parallelized functions provide a considerable speed-up from their serial counterparts.

Table 1: Run time for Cournot game, ($\varepsilon = .01$)

# of subgradients	Points in A	Serial	2 proc.	4 proc.	8 proc.
8	10	27 sec	19 sec	14 sec	14 sec
16	10	1 min 2 sec	45 sec	30 sec	31 sec
32	10	2 min 31 sec	1 min 33 sec	1 min 5 sec	1 min 13 sec
8	20	2 min 8 sec	1 min 13 sec	1 min 5 sec	58 sec
16	20	4 min 44 sec	2 min 39 sec	2 min 10 sec	2 min 15 sec
32	20	9 min 55 sec	5 min 20 sec	4 min 39 sec	4 min 41 sec

*Based on root process run times on a basic 2-core MacBook Pro (2.8 GHz).

**Times are based on single executions and may differ from actual run times.

4.2 Bertrand Duopoly

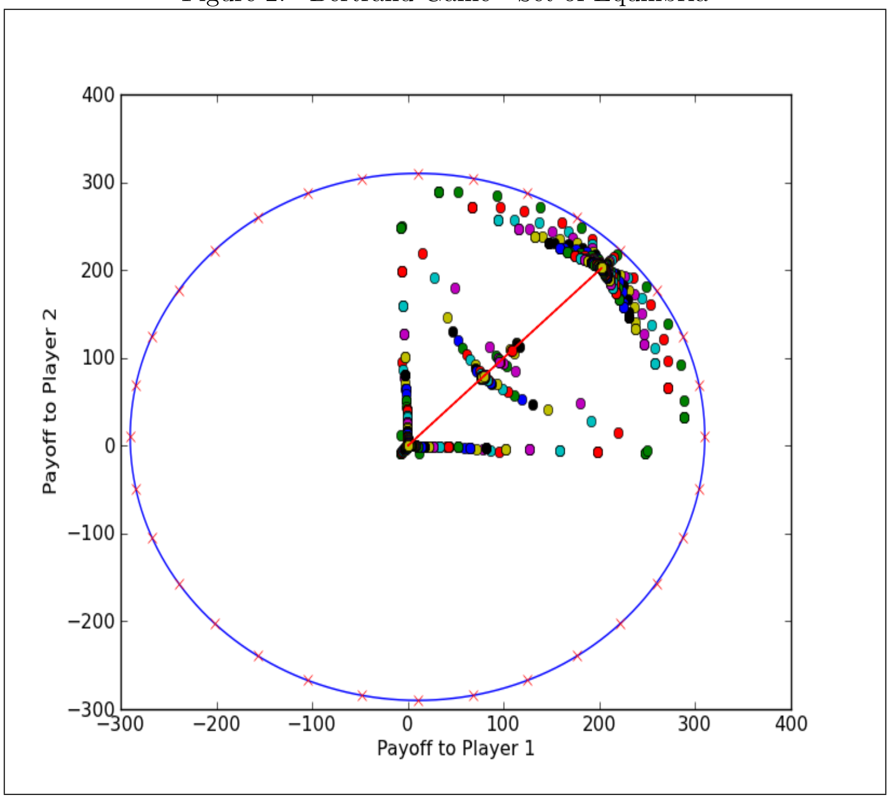
Suppose that there are two firms. Instead of choosing quantity of output (as with Cournot duopoly), they choose the price at which to sell. Demand is then given by the function:

$$Q = 100 - 5P.$$

The firm with the lowest price will capture the entire market. If the firms have the same price, then they split the market evenly. Profits are defined as before, with costs fixed at 2 for each firm in every period.

As before, the set of possible strategies is discretized so that P is one of n equally spaced points in the interval $[0, 20]$. The outer hyperplane approximation generates the following set of Nash equilibria.

Figure 2: Bertrand Game - Set of Equilibria



Here the red line is the set of all possible equilibria. This suggests the two firms will only ever set their prices equally in equilibrium, though for many possible prices.

Table 2: Serial Run time for Bertrand game, ($\varepsilon = .001$)

# of subgradients	Points in S_i	Serial time	2 Proc.	4 Proc.	8 Proc.
8	10	2 min 15 sec	1 min 59 sec	1 min 58 sec	1 min 55 sec
16	10	5 min 52 sec	4 min 27 sec	2 min 41 sec	3 min 9 sec
32	10	15 min 49 sec	9 min 6 sec	8 min 54 sec	7 min 43 sec
8	20	11 min 40 sec	7 min 18 sec.	6 mon 32 sec	7 min 13 sec
16	20	21 min 8 sec	16 min 6 sec	15 min 48 sec	15 min 58 sec
32	20	48 min 3 sec	34 min 32 sec	21 min 27 sec sec	29 min 51 sec

*Based on single executions on a basic 2-core MacBook Pro (2.8 GHz).

5 Publication

Supergametools is available for download at: bitbucket.org/ben_tengelsen/supergametools.
Additional information is on my research website: sites.google.com/site/btengelsenresearch/python.

References

Ely, J. C., and J. Välimäki (2002): "A robust folk theorem for the prisoner's dilemma," *Journal of Economic Theory*, 102(1), 84–105.

Fudenberg, D., and E. Maskin (1986): "The folk theorem in repeated games with discounting or with incomplete information," *Econometrica: Journal of the Econometric Society*, pp. 533–554.

Judd, K. L., S. Yeltekin, and J. Conklin (2003): "Computing supergame equilibria," *Econometrica*, 71(4), 1239–1254.

Mailath, G. J., and L. Samuelson (2011): "Repeated games and reputations: long-run relationships," *OUP Catalogue*.