Computing Supergame Equilibria with Supergametools

Benjamin Tengelsen

Carnegie Mellon University

Computation in CA 2013

Special thanks to Şevin Yeltekin

Talk Outline

- Brief review of Ken's Monday morning talk
- Closer look at the algorithms
- Software demo (lots of examples)

About Me

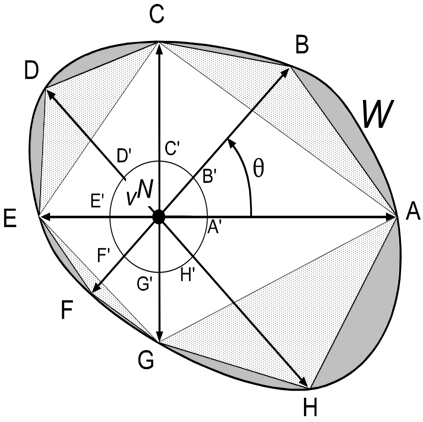
- First/second year Econ PhD student at Carnegie Mellon
- Interested in macro, labor markets, online markets
  - Search, Union/firm strategy, Unemployment
  - "Fiscal multipliers in a search model for unemployment."
  - "Welfare cost of business cycles with unemployment."
- Fan of Python, parallel computing

Nash Carter Tengelsen

Review of Ken's talk

- Single vs. repeated games
- Monotone function over compact space $\rightarrow$ largest fixed point
- Two methods: inner/outer approximation

A Look at the Algorithms: Inner approximation



Inner approximations

Monotone Inner Hyperplane Approximation

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

Step 1 Find extremal points of $B(W)$:

For each search subgradient $h_\ell \in H,\ \ell = 1, .., L$.

(1) For each $a \in A$, solve the linear program

$$
\begin{array}{rcl}
c_\ell(a) & = & \max_w h_\ell \cdot [(1-\delta)\Pi(a) + \delta w] \\
 & \text{(i)} & w \in W \\
 & \text{(ii)} & (1-\delta)\Pi^i(a) + \delta w_i \geq \\
 & & \qquad (1-\delta)\Pi_i^*(a_{-i}) + \delta \underline{w}_i,\ i = 1, .., N
\end{array}
\tag{1}
$$

Let $w_\ell(a)$ be a w value which solves (1).

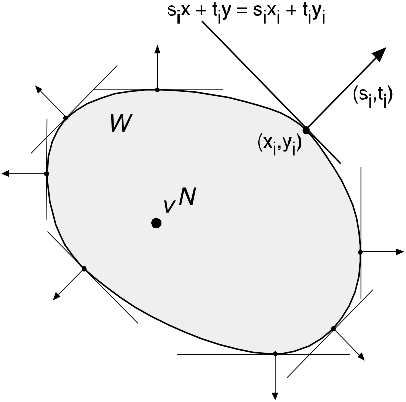
Monotone Inner Hyperplane Approximation cont'd

(2) Find best action profile $a \in A$ and continuation value:

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

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

Outer approximation



A convex set and supporting hyperplanes

Outer vs. Inner Approximations

- Easily paralellizable
- Maximization operation is a linear program

Outer vs. Inner Approximations

- Computations actually constitute a proof that something is in or out of equilibrium payoff set - not just an approximation.
- Any point within the inner approximation is an equilibrium
- No point outside of outer approximation can be an equilibrium
- Can show certain equilibrium payoffs/actions are not possible
  - E.g., that joint profit maximization is not possible

Error Bounds

- Difference between inner and outer approximations is approximation error
- Difference is small in many examples, often decreases as number of search gradients increase

Supergametools

- Open source python library (bitbucket)
- Serial/parallel commands for inner/outer approximation
- Easy to use:
  - payoff matrices as inputs
  - inner/outer approximations as outputs
- Limitations
  - 2-player games only
  - repeated static games only
  - very young in development phase

Example 0: Battle of the Sexes

- Husband and wife are meeting in town for a date
- Traveling separately. Neither remembers where to meet.
- Three options: Opera, Football, Sushi

Example 0: Battle of the Sexes

- Wife (husb.) chooses row (column)
- (wife, husb)

	O	F	S
O	$7,\ 2$	$4,\ 6$	$4,\ 3$
F	$1,\ 1$	$5,\ 8$	$2,\ 2$
S	$2,\ 1$	$3,\ 1$	$4,\ 4$

Example 1a: Cournot Game (two firms)

- Inverse demand function:

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

- Profit function:

$$\Pi_j(q_j) = P \cdot q_j - c_j$$

- Objective:

$$\max_{q_j} E_0 \left[\sum_{t=0}^{\infty} \beta^t q_{j_t} (P - c_j) \right]$$

- Action space: $(0, 6)$ interval discretely spaced with 15 points

Example 1a: Cournot Game (two firms)

- Suppose gov. imposes a tax on firms, increases c_j
- How does that change the equilibria set?

Speedup from Parallelization

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

subgradients	Points in A	Serial	4 proc.
8	10	27 sec	14 sec
16	10	1 min 2 sec	30 sec
32	10	2 min 31 sec	1 min 5 sec
8	20	2 min 8 sec	1 min 5 sec
16	20	4 min 44 sec	2 min 10 sec
32	20	9 min 55 sec	4 min 39 sec

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

Example 1b: Bertrand Game (two firms)

- Inverse demand function: P = 20 - Q/5
- Price function: $P = \min\{p_1, p_2\}$
- Lowest price captures the whole market. $\implies p_1 = p_2$.
- Action space: $(0, 10)$ interval discretely spaced with 15 points

Speedup from Parallelization

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

# of subgradients	Points in S_i	Serial time	4 Proc.
8	10	2 min 15 sec	1 min 58 sec
16	10	5 min 52 sec	2 min 41 sec
32	10	15 min 49 sec	8 min 54 sec
8	20	11 min 40 sec	6 mon 32 sec
16	20	21 min 8 sec	15 min 40 sec
32	20	48 min 3 sec	21 min 27 sec

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

Example 2: Union/Firm Game

- Espinosa and Rhee (1989)
- Union sets wage, $W \in [0, 1]$
- Firm chooses number of workers, $L \in [0, 1]$
- Capital is fixed: $K = \frac{1}{2}$
- Production technology:

$$Q = L^{\alpha} K^{1-\alpha}$$

Example 2: Union/Firm Game

- Profit function for firm:

$$\Pi(L, W) = PQ - WL - RK$$

- Inverse demand function:

$$P = 1 - q/10$$

- Union utility function:

$$U(L) = W \cdot L$$

Example 2: Union/Firm Game

- What if production becomes more capital intensive?
- What if we impose a minimum wage?

$$P = \max\{1 - q/10, .3\}$$

Future work

- A few challenges
  - Complete action space is not always known
  - Most repeated games are dynamic
- How to determine quality of results