Artificial Intelligence and Collusion: An Experiment

Calvano, Calzolari, Denicolò, Pastorello
Discussion by E. Tarantino

December 4, 2018

- Research question: study pricing by algorithms in competition games. Does cooperation arise in a model-free environment without communication?

- Methodology: Q-learning implementing a Reinforcement Learning algorithm. Key ingredients of the analysis:
  - DGP (*the model*); multiple machines (*the algorithms*).
  - Let the algorithm *learn* about the model by taking actions.
  - Study properties of the resulting market dynamics.

- With Q-learning, in each t the algorithm can
  - Exploit: take action a that, given a state s, returned higher reward r in the past.
  - Explore: take a random action in the support of possible actions.

- That is, the algorithm learns by experimenting. What does it mean?

- Let
$$Q(s,a) \equiv max_a \mathbb{E}[Reward(a,s)].$$
- Values of $Q(s,a)$ stored in a matrix, containing a reward for each (s,a).
- To find $Q(s,a)$, repeat the following procedure infinitely:
  - Given s and a, observe r and resulting next state s'.
  - Adjust $\hat{Q}(s,a)$ like this
$$\hat{Q}(s,a) \leftarrow (1-\alpha_t)\hat{Q}(s,a) + \alpha_t(r + \delta max_{a'}\hat{Q}(s',a')) \qquad (1)$$
- If algorithm visits each (s,a) infinitely often, $\hat{Q}(s,a)$ likely to converge to $Q(s,a)$ and solve Bellman equation.
- To achieve this, assume that with exogenous (small) prob, algorithm takes a random action. This helps it explore more of the solution space.

"Model-free" algorithm (1)

Model – DGP	**Algorithm information**
N	?
Demand	?
Strategic relationship	?
Costs	?
Profit function	?
Transition probabilities	?

- So, even if DGP fully stationary, problem is highly non-stationary from algorithm point of view!

"Model-free" algorithm (2)

Algorithm i**:**	sets own action (x_1)	observes environment (x_2)	and reward (x_3)
	p_i	p_{-i}	$\pi_i(p_i, p_{-i})$ in Bertrand duopoly
	q_i	$P(Q)$	$\pi(Q)$ in Cournot N-firm oligopoly
	q_i	spread on Italian bonds	$\pi(q_i)$ in monopoly
	...	...	...

"Model-free" algorithm (2)

Algorithm i**:**	sets own action (x_1)	observes environment (x_2)	and reward (x_3)
	p_i	p_{-i}	$\pi_i(p_i, p_{-i})$ in Bertrand duopoly
	q_i	$P(Q)$	$\pi(Q)$ in Cournot N-firm oligopoly
	q_i	spread on Italian bonds	$\pi(q_i)$ in monopoly
	...	...	...

- CCDP show convergence to supra-competitive pricing, and document that algorithms learn to cooperate (i.e., do not fail to learn to compete).
  - Algorithms mutually best-respond 55.5% of the times.
- Very nice, thought provoking. Opens a window onto a methodology largely unexplored by IO economists.
- But how is *any* learning to cooperation possible at all? $(\mathbf{o\_O})$
  Back to basics!

Gas stations example

- Maureen Ohlhausen, Acting Chair of the FTC: consider a situation where the owners of two gas stations on opposite sides of a road signal price increases to each other by changing the prices on the board.
- As long as owners are acting unilaterally, she claims, this practice falls outside the scope of antitrust liability.
- Same if each gas station determines the price to charge using a computer programmed to take into account the price of the other.
- Agencies' conclusion: "[w]ithout proof of collusion or evidence that the knowing parallel adoption of pricing formulas *narrowed the range of prices over time*, parallel pricing conduct may be outside the reach of the antitrust laws."

DGP	Algorithm information
$N = 2$	$\surd$
Linear demand	$\surd$
Bertrand pricing	$\surd$
Strategic complementarity	$\surd$
Symmetric firms	$\surd$
No uncertainty	$\surd$

- We know that, if $\delta \geq \delta^*$, grim-trigger strategies allow firms to sustain any price between mc and p^m.
- However, huge multiplicity of equilibria. Typical resolution: focus on (symmetric) profit-maximizing SPE. Idea: firms grope their way to it!

DGP	Algorithm information
$N = 2$	$\surd$
Linear demand	$\surd$
Bertrand pricing	$\surd$
Strategic complementarity	$\surd$
Symmetric firms	$\surd$
No uncertainty	$\surd$

- We know that, if $\delta \geq \delta^*$, grim-trigger strategies allow firms to sustain any price between mc and p^m.
- However, huge multiplicity of equilibria. Typical resolution: focus on (symmetric) profit-maximizing SPE. Idea: firms grope their way to it!
- Any role for experimentation? Maybe!

  Conjecture: By "exploring," algorithms find their way to profit max p.
- To disprove conjecture, let $\delta < \delta^*$: cooperation should be more difficult.
  - Meaningful case: δ captures freq of interaction, market growth.

Performance measures

- Statements like "convergence is somewhat fast" are difficult to interpret. What is fast? Compared to what? What is the counterfactual world?
- We already know from literatures in computer science and operations research that static pricing is beatable by algorithmic dynamic pricing, including use of neural networks and Q-learning itself (e.g., van den Boer, 2015).
- Maybe more reasonable to consider how *different* machine learning algorithms perform when put one against the other?
  - For example, what if a RL-algorithm is faced with a competing neural network?
  - Otherwise, comparing algorithms that are able to communicate and algorithms that cannot may inform authorities on the statistics that can be used to detect communication.

External validity

- Algorithms (not only in this paper!) are evaluated in simulated and rather standard DGP.
- Considering that one of machine learning's upsides is ability to cope with uncertainty, it seems odd that these methods are not tested in a proper marketplace.
- For example, Fisher, Gallino and Li (2016) test a best-response pricing algorithm in a field experiment.
  - They find evidence consistent with increase in revenue thanks to use of algorithmic pricing.

Convergence and stationarity

- As the majority of papers in this literature, CCDP looks at a Markov Decision Process (MDP). Justified based on simplicity and tractability.
- However, most real-world settings do not fulfil Markov properties (bacause of, e.g., non-stationary and history dependence).
- This motivates use of Partially Observable MDP, in which the agent does not necessarily know which state it is in.

Minor comments

- The section considering robustness to use of $\delta \to 1$ shows that cooperation is difficult to achieve for $\delta > 0.99$.
- My understanding is that in studies like these δ is bounded away from 1 to have a well-behaved program.
- Since time horizon is infinite, the value of the sequences will be infinite.
- Then, the condition that $\delta < 1$ necessary to find a finite value to an infinite sequence.
- Unclear what properties of the algorithm are explored in a setting with δ close to 1.