Multi-Agent Learning in a Pricing Game

Bernhard von Stengel

Galit Ashkenazi-Golan, Sahar Jahani, Katerina Papadaki,
Edward Plumb

Department of Mathematics
London School of Economics

June 2023

Overview (everything is work in progress)

Aim: exploring larger games with machine learning

Example: duopoly with demand inertia.

- description of the duopoly game
- **new framework:**
  - learning a strategy in the base game
  - new strategy added to a population game
  - compute a new equilibrium of the population game as the next learning environment
- main advantage: **modularity**, study aspects separately.

Duopoly with demand inertia

Model:

- a multi-stage **pricing game** = our base game
- analysed theoretically (**subgame perfect** equilibrium)

[R. Selten (1965), Game-theoretic analysis of an oligopolic model with buyers' interia. [German] *Zeitsch. gesammte Staatswiss.* 21, 301–304]

- experimentally with subjects and submitted programmed strategies

[C. Keser (1993), Some results of experimental duopoly markets with demand inertia. *Journal of Industrial Economics* 41, 133–151]

[1992 PhD thesis: Springer Lecture Notes Econ. Math. Systems 391]

Demand potential, prices, profits, inertia

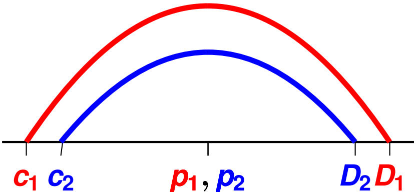
Total demand potential **400** split as $D_1 + D_2$ between two producing firms with costs $c_1 = 57$ and $c_2 = 71$.

Firm i chooses **price** p_i and **sells** $D_i - p_i$ units, gets **profit** $(D_i - p_i)(p_i - c_i)$

Demand potential, prices, profits, inertia

Total demand potential **400** split as $D_1 + D_2$ between two producing firms with costs $c_1 = 57$ and $c_2 = 71$.

Firm i chooses **price** p_i and **sells** $D_i - p_i$ units, gets **profit**
$(D_i - p_i)(p_i - c_i) = -p_i^2 + (D_i + c_i)p_i - D_i c_i$



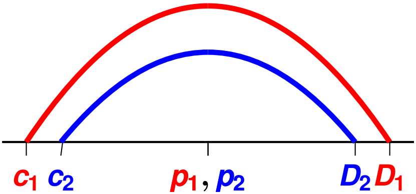
Demand potential, prices, profits, inertia

Total demand potential **400** split as $D_1 + D_2$ between two producing firms with costs $c_1 = 57$ and $c_2 = 71$.

Firm i chooses **price** p_i and **sells** $D_i - p_i$ units, gets **profit** $(D_i - p_i)(p_i - c_i) = -p_i^2 + (D_i + c_i)p_i - D_i c_i$

Optimal **myopic** price $p_i = (c_i + D_i)/2$. **Example:**

$D_1 = 207$, $D_2 = 193$, $p_1 = p_2 = 132$, profits 75^2, 61^2.



Demand potential, prices, profits, inertia

Total demand potential **400** split as $D_1 + D_2$ between two producing firms with costs $c_1 = 57$ and $c_2 = 71$.

Firm i chooses **price** p_i and **sells** $D_i - p_i$ units, gets **profit** $(D_i - p_i)(p_i - c_i) = -p_i^2 + (D_i + c_i)p_i - D_i c_i$

Optimal **myopic** price $p_i = (c_i + D_i)/2$. **Example:**

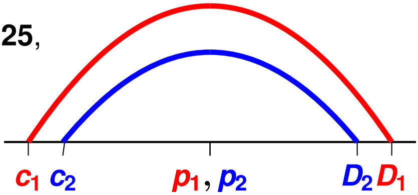
$D_1 = 207$, $D_2 = 193$, $p_1 = p_2 = 132$, profits 75^2, 61^2.

Played over 25 stages $t = 1, \ldots, 25$,

$D_1^1 = D_1^1 = 200$

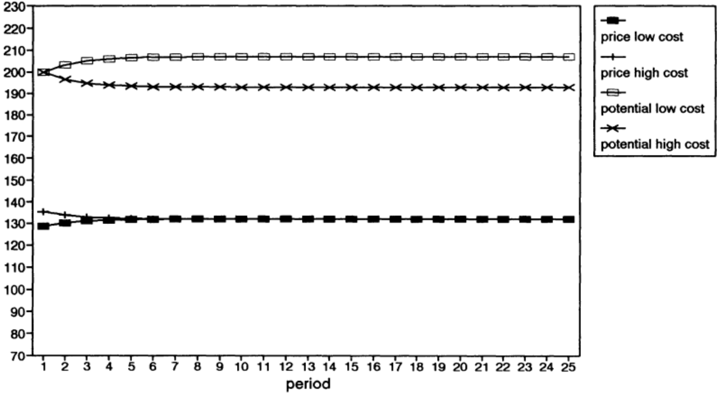
$D_1^{t+1} = D_1^t + (p_2^t - p_1^t)/2$

$D_2^{t+1} = D_2^t + (p_1^t - p_2^t)/2$



Cooperative solution

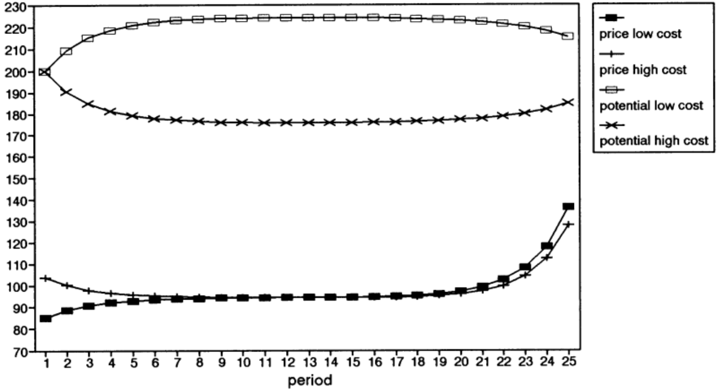
If both firms always choose myopic price:



Total profits over 25 stages about **156**k, **109**k

Subgame perfect equilibrium

Via parameterized backward induction:



Total profits about **137**k, **61**k

Strategy experiments

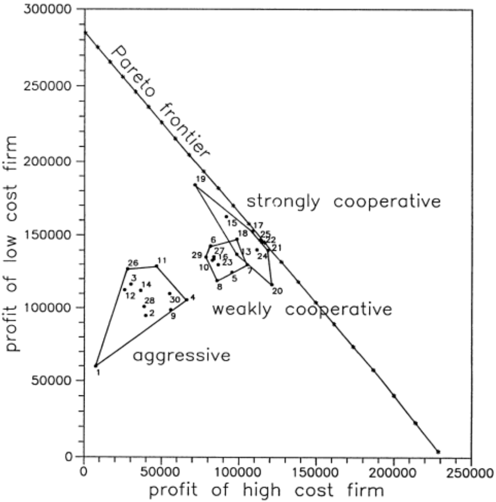
Submitted strategy = flowchart pair, for low-cost and high-cost firm.

Two competition rounds:

first round: 45 entries

(after feedback:)
second round: 34 entries

second-round profits:



Lessons from a participant's perspective

Profits were totalled against all other teams (including own type)

- **Very important for doing well:** understanding the game
  - focus on demand potential, not price
  - smaller price strongly increases future profits
  - avoid wild swings
  - exploit "suckers"

Lessons from a participant's perspective

Profits were totalled against all other teams (including own type)

- **Very important for doing well:** understanding the game
  - focus on demand potential, not price
  - smaller price strongly increases future profits
  - avoid wild swings
  - exploit "suckers"
- No clear "cooperative behavior"; myopic play is a focal point

Lessons from a participant's perspective

Profits were totalled against all other teams (including own type)

- **Very important for doing well:** understanding the game
  - focus on demand potential, not price
  - smaller price strongly increases future profits
  - avoid wild swings
  - exploit "suckers"
- No clear "cooperative behavior"; myopic play is a focal point
- Strategies **react** (typically to last price) but have **no model of the opponent**
  - one team reacted to **predicted** rather than past behavior

The learning framework

- Base game = pricing game over 25 stages, in two roles

The learning framework

- Base game = pricing game over 25 stages, in two roles
- strategy represented by an RL agent that chooses the **next price** as a function of data for the last, say, **3** stages

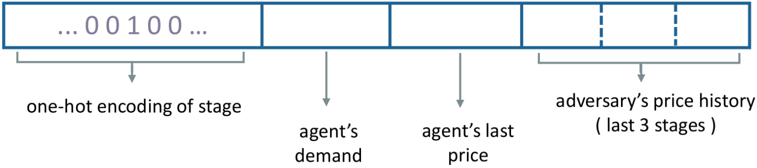
The learning framework

- Base game = pricing game over 25 stages, in two roles
- strategy represented by an RL agent that chooses the **next price** as a function of data for the last, say, **3** stages
- agent is **trained** by repeatedly meeting another random agent, **drawn** from a **mixed equilibrium** of existing strategies, which define the population game of pairwise interactions

Learning a new strategy: not easy

Main assumption: the learning environment is constant (not evolving with the learning agent) but random (mixed equilibrium)

- a whole strategy, for unknown situations, must be learned
- assumption: learn next price as **function** of last **3** stages with
  - information per state : own price, own profit, opponent price
  - explicit **state**: demand potential, stage
  - e.g. using Policy Gradient Method (PGM):



- learning is **slow** – Q-learning needs large Q-table

A custom learning agent for this game

Suppose the aim is a **strong strategy** for this game ("feature engineering", as in AlphaGo).

(Not for a general base game; use as benchmark?)

- Tune a small set of **parameters** for a special own strategy:
  - aim for a "fair split" of demand potential
  - **predict** opponent price with multiplicative weights
  - set own price to achieve **target** demand potential
  - use somewhat lower price to steal customers

A custom learning agent for this game

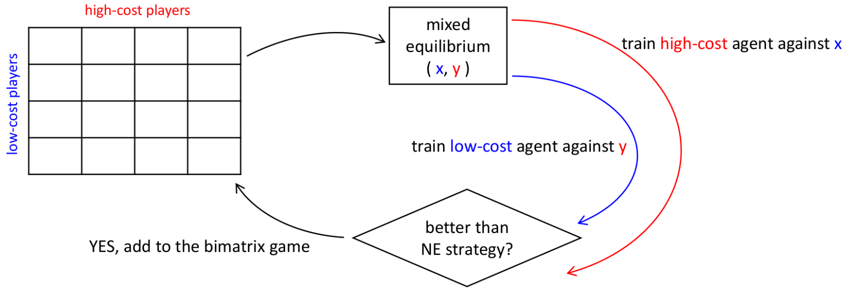
Suppose the aim is a **strong strategy** for this game ("feature engineering", as in AlphaGo).

(Not for a general base game; use as benchmark?)

- Tune a small set of **parameters** for a special own strategy:
  - aim for a "fair split" of demand potential
  - **predict** opponent price with multiplicative weights
  - set own price to achieve **target** demand potential
  - use somewhat lower price to steal customers
- optimization of $\sim$ **4** parameters close to human-designed agent (not by gradient descent, too many "cliffs")

Add newly RL trained agents

- a successfully trained strategy is **added** to the population game
  - new entrant has payoffs against each existing strategy
  - defines a bimatrix game with new equilibrium as next learning environment



The population game

A successfully trained strategy is **added** to the population game, as a row or column depending on its role (low- or high-cost firm).

The population game

A successfully trained strategy is **added** to the population game, as a row or column depending on its role (low- or high-cost firm).

A new **equilibrium** is computed, typically **mixed** and **not unique**.

That mixed equilibrium defines the next learning environment.

The population game

A successfully trained strategy is **added** to the population game, as a row or column depending on its role (low- or high-cost firm).

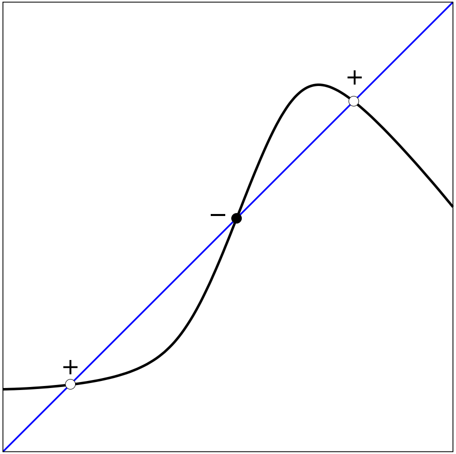
A new **equilibrium** is computed, typically **mixed** and **not unique**.

That mixed equilibrium defines the next learning environment.

Which equilibrium?

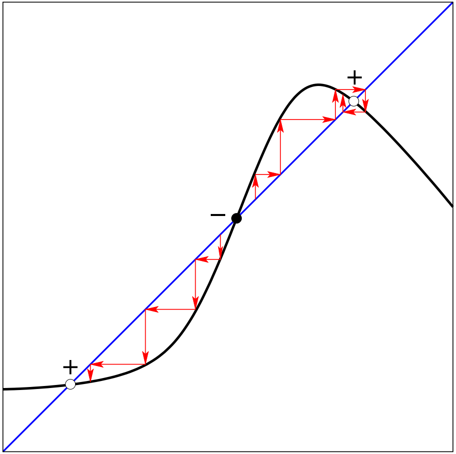
- equilibrium selection via computing an equilibrium from random starting profile as **prior** (tracing procedure)
  - as **proxy for evolutionary dynamics** (TBC!)
  - finds only positive-**index** equilibria (for dynamic stability)
  - the prior could be the previous equilibrium
- has typically small support (no issue with PPAD-hardness)

Index of a fixed point



Fixed point $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$: index$(\boldsymbol{x}) =$ sign det $\boldsymbol{D}(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x}))$

Index of a fixed point



Fixed point $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$:   index$(\boldsymbol{x}) =$ sign det $\boldsymbol{D}(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x}))$

positive index necessary for dynamic stability

Pure NE vs. mixed NE with higher payoff

$(\boldsymbol{B}, \boldsymbol{b})$ = **PGM-learned** pure NE, stops at size $\mathbf{8} \times \mathbf{5}$ (SPNE?)

Pure NE vs. mixed NE with higher payoff

(***B***, ***b***) = **PGM-learned** pure NE, stops at size **8** $\times$ **5** (SPNE?)

Restricted to equilibrium supports:

1 \ 2	myopic	*a*	*b*
A	155, $\boxed{80}$	$\boxed{142}$, 79	124, 70
B	$\boxed{179}$, 46	116, 53	$\boxed{127}$, $\boxed{56}$

Pure NE vs. mixed NE with higher payoff

(***B***, ***b***) = **PGM-learned** pure NE, stops at size **8** × **5** (SPNE?)

Restricted to equilibrium supports:

	1 \ 2	**0.52** myopic	**0.48** *a*	**0** *b*	mixed NE payoffs
0.875	***A***	$\boxed{80}$ 155	79 $\boxed{142}$	70 124	75.75 148.76
0.125	***B***	46 $\boxed{179}$	53 116	$\boxed{56}$ 127	

Advantage of this framework

It is **modular** rather than a huge simulation:

- the base game (pricing game)
  - is complex (too complex?) as an interesting learning scenario
  - allows competition and cooperation
  - potentially has “hand-made” good (benchmark?) strategies
  - can be replaced by another game
- the population game . . . uses game theory
  - provides via equilibria a “stable” learning environment
  - has typically mixed, non-unique equilibria
  - allows different equilibrium concepts (mixed, evolutionary)

$\Rightarrow$ can independently investigate different aspects

Challenges ahead

- RL agents learn **slowly** (investigate further approaches)
  - often too jittery (oscillations)
- Learning for **evolutionary success** different from high-payoff
- comparison with existing approaches, e.g.
  [E. Calvano, G. Calzolari, V. Denicolò, and S. Pastorello (2020), Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review* 110(10), 3267–3397.]

Future extension:

- competition between more than two firms (better model)
- different **base games**

Thank you!