Multi-Agent Learning and Equilibrium

Bernhard von Stengel

joint with: Galit Ashkenazi-Golan, Katerina Papadaki ...

Department of Mathematics
London School of Economics

27 October 2022

[Click on references for URLs]

Overview (everything is work in progress)

Aim: exploring larger games with machine learning

Example: duopoly with demand inertia.

- description of the duopoly game
- existing human-designed strategies for strategic tournament
- **new framework:**
  - learning a strategy in the base game
  - new strategy extends 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

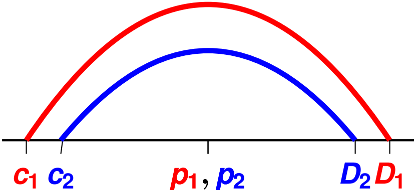
Total demand potential **400** split as $D_1 + D_2$ between two producing firms with costs $c_1 = 57$ and $c_2 = 71$.

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** $(D_i - p_i)(p_i - c_i)$.



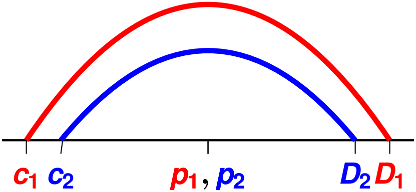
Demand potential, prices, profits, inertia

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** $(D_i - p_i)(p_i - c_i)$.

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** $(D_i - p_i)(p_i - c_i)$.

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

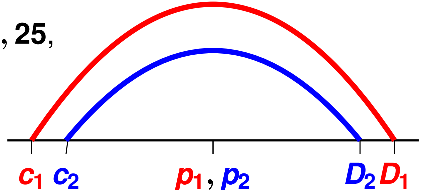
$D_1 = 207$, $D_2 = 193$, $p_1 = p_2 = 132$, profits 75^2, 61^2.

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

$$D_1^1 = D_1^1 = 200$$

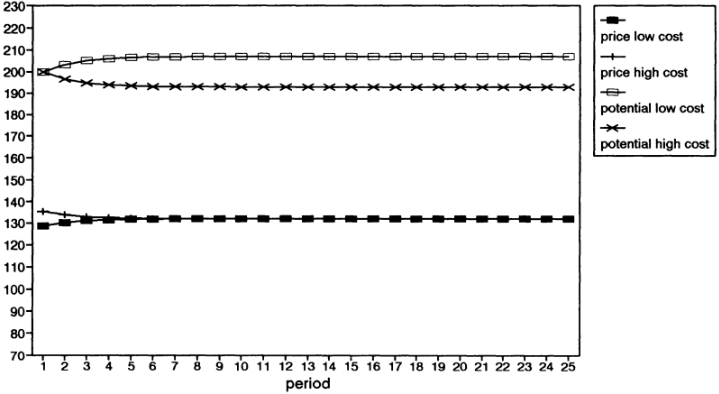
$$D_1^{t+1} = D_1^t + (p_2^t - p_1^t)/2$$

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



Cooperative solution

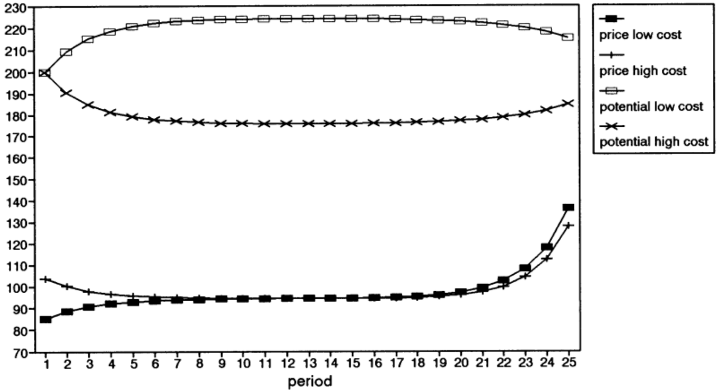
If both producers always choose myopic duopoly price:



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

Subgame perfect equilibrium

Via parameterized backward induction:



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

Strategy experiments

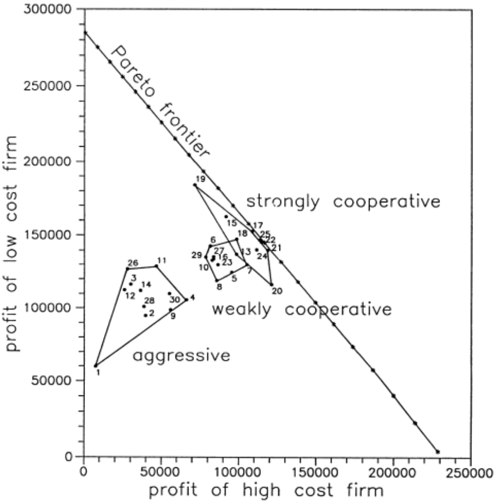
Submitted strategy = flowchart pair, for low-cost and high-cost firm.

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

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
- "Optimization" of parameters typically against self-play.

The learning framework

- Base game = pricing game over 25 rounds, in two roles
  - perhaps better: introduce termination probability of 4% after each round to avoid end effect, leads to random number of rounds; reward is then average profit per round

The learning framework

- Base game = pricing game over 25 rounds, in two roles
  - perhaps better: introduce termination probability of 4% after each round to avoid end effect, leads to random number of rounds; reward is then average profit per round
- strategy (agent) represented by a neural network that chooses the **next price** as a function of data for the last, say, **3** periods

The learning framework

- Base game = pricing game over 25 rounds, in two roles
  - perhaps better: introduce termination probability of 4% after each round to avoid end effect, leads to random number of rounds; reward is then average profit per round
- strategy (agent) represented by a neural network that chooses the **next price** as a function of data for the last, say, **3** periods
- 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

The learning framework

- Base game = pricing game over 25 rounds, in two roles
  - perhaps better: introduce termination probability of 4% after each round to avoid end effect, leads to random number of rounds; reward is then average profit per round
- strategy (agent) represented by a neural network that chooses the **next price** as a function of data for the last, say, **3** periods
- 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
- 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

Learning a new strategy: issues

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** periods with
  - information per period: own price, own profit, opponent price
  - implicit (or explicit?) **state**: demand potential

Learning a new strategy: issues

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** periods with
  - information per period: own price, own profit, opponent price
  - implicit (or explicit?) **state**: demand potential
- **reward** function (critical: when?) : average per-period profit

Learning a new strategy: issues

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** periods with
  - information per period: own price, own profit, opponent price
  - implicit (or explicit?) **state**: demand potential
- **reward** function (critical: when?) : average per-period profit
- for population game: profit recorded and updated per opponent
  - weigh with length of interaction? (. . . if less than 3 periods)?
- how to initialize? vary an existing agent?
- when has an agent learned enough?

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 exponentially α-weighted from past
  - in fact, opponent sales better predictor
- set own price to achieve target demand potential
  - use somewhat lower price to steal customers

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).

- (add only one row/column, or both?)

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).

- (add only one row/column, or both?)

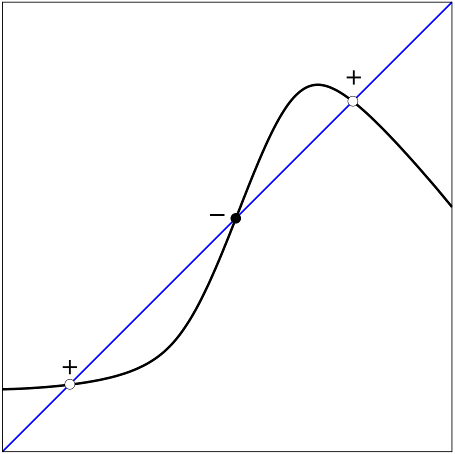
A new **equilibrium** is computed, typically **mixed** and **not unique**.

That mixed equilibrium defines the next learning environment.

Which equilibrium?

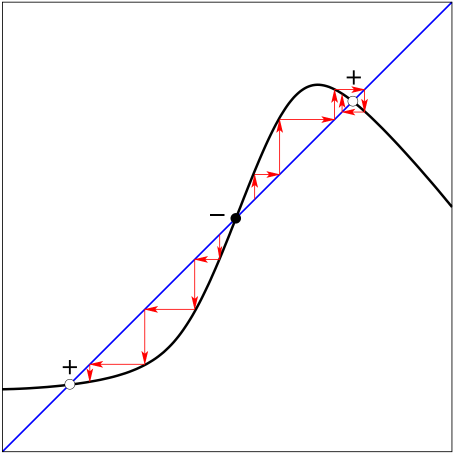
- equilibrium selection via computing an equilibrium from random starting profile as **prior** (tracing procedure)
  - as **proxy for evolutionary dynamics**
  - finds only positive-index equilibria (for dynamic stability)
  - the prior could be the previous equilibrium
- has typically small support (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

Example of a mixed equilibrium

L \ H	*a*	*b*	*c*	*d*
A	$\boxed{100}$ 152	86 $\boxed{180}$	98 157	99 154
B	$\boxed{110}$ 74	66 170	47 $\boxed{178}$	75 130
C	102 $\boxed{155}$	$\boxed{103}$ 160	$\boxed{103}$ 156	$\boxed{103}$ 157
D	$\boxed{105}$ 154	$\boxed{105}$ 158	$\boxed{105}$ 155	104 $\boxed{159}$

Example of a mixed equilibrium

		0.05	0.03	0.58	0.34	
	L \ H	***a***	***b***	***c***	***d***	
0.02	***A***	$\boxed{100}$ 152	86 $\boxed{180}$	98 157	99 154	equilibrium payoffs:
0.01	***B***	$\boxed{110}$ 74	66 170	47 $\boxed{178}$	75 130	**103** **156**
0.67	***C***	102 $\boxed{155}$	$\boxed{103}$ 160	$\boxed{103}$ 156	$\boxed{103}$ 157	
0.30	***D***	$\boxed{105}$ 154	$\boxed{105}$ 158	$\boxed{105}$ 155	104 $\boxed{159}$	

"Market share" as different reward function?

In a mixed equilibrium, all pure best responses have equal payoff.

$\Rightarrow$ mixed-strategy probabilities depend on opponent payoffs

Example: Inspection game

	0.8 comply	0.2 cheat
0.8 Don't inspect	$\boxed{0}$, 0	-10, $\boxed{10}$
0.2 Inspect	-1, $\boxed{0}$	$\boxed{-6}$, -40

“Market share” as different reward function?

In a mixed equilibrium, all pure best responses have equal payoff.

$\Rightarrow$ mixed-strategy probabilities depend on opponent payoffs

Example: Inspection game

	0.8 comply	0.2 cheat
0.9 Don’t inspect	$\boxed{0}$, 0	-10, $\boxed{10}$
0.1 Inspect	-1, $\boxed{0}$	$\boxed{-6}$, -90

“Market share” as different reward function?

In a mixed equilibrium, all pure best responses have equal payoff.

$\Rightarrow$ mixed-strategy probabilities depend on opponent payoffs

Example: Inspection game

	0.8 comply	**0.2** cheat
0.9 Don't inspect	0 (col.), $\boxed{0}$ (row)	$\boxed{10}$ (col.), -10 (row)
0.1 Inspect	$\boxed{0}$ (col.), -1 (row)	-90 (col.), $\boxed{-6}$ (row)

“Market share” as different reward function?

In a mixed equilibrium, all pure best responses have equal payoff.

$\Rightarrow$ mixed-strategy probabilities depend on opponent payoffs

Example: Inspection game

	0.8 comply	0.2 cheat
0.9 Don’t inspect	0 / $\boxed{0}$	$\boxed{10}$ / -10
0.1 Inspect	$\boxed{0}$ / -1	-90 / $\boxed{-6}$

Learn to **treat opponents equally** to get high population share?

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 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

- "Under control": equilibrium computation for the population game, tournament set-up
- **not yet:** implementing the learning agents
- 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!