Artificial Intelligence, Algorithmic Pricing, and Collusion

Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello

Presenter: **Zhong Hongwei**

China Economics and Management Academy

September 28, 2022

Authors

Emilio Calvano,
University of Rome

Giacomo Calzolari,
European
University Institute

Vincenzo Denicolo,
University of
Bologna

Sergio Pastorello,
University of
Bologna

Presentation Outline

Presentation Outline

What is AI?

What is AI?

- **A practical one:** (Microsoft)[Taddy, Ch. 2] System of intelligence, full end-to-end solution, able to ingest human-level knowledge (e.g., via machine reading and computer vision) and use this information to automate and accelerate tasks that were previously only performed by humans [e.g. learning and problem solving].
- **Computer Science:** Artificial intelligence, is any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals [Poole, Mackworth, Goebel 1998]
- **A super practical one:** machine doing cognitive work (not really the kind of AI we see these days)

Three views of AI in the academic literature

1. Statistical tools to make predictions and decisions (aka 'narrow' AI)
2. Automation
3. Super human/Cognitive machines (philosophical - does not exist yet)

We mainly consider 1 mostly and touch on 2.

For Emilio:

AI solutions are software tools useful to cheaply crack dynamic decision problems under uncertainty, and AI is basically about making choices and predicting their consequences.

AI in marketplaces

1. Matching content and ads to audiences
   - Online advertising (Contextual Bandits)
   - Editorial choices (collaborative filtering)
2. Matching consumers to producers - Recommender systems
3. Pricing goods and services
   - Personalized prices
   - Dynamic/surge pricing
4. Personalized banking
5. Financial advice/Portfolio management
6. Algorithmic trading and bidding

We need to understand what this is doing to our markets and (more generally) our societies.

AI based algos are typically...

1. **Model free**
2. Autonomously **learns from experience**
3. Increasingly available **off the shelf**

Programmers just specify

- the objective function (e.g. profits)
- what contexts to monitor/which data to use (e.g. competitors' choices)

Clear research and policy challenges

- New research please: old theories **do not** 'work'!
- New policies please: current legal doctrines **do not** 'work'!

Pricing algorithms are populating markets

- Firms are increasingly adopting pricing algorithms
- Chen et al. (2016) document that over 500 settlers active on 1,641 top Amazon listings use algorithmic pricing
- Amazon (API interface) *"automation of setting and pricing activities is a source of value and efficiency"*
- Algorithms are not a temporary phenomenon

Presentation Outline

Background

- In recent years, a large number of algorithms have been applied to the pricing of their products or services in the market.
- For example, when using taxi apps, the prices that pop up are basically determined by algorithms.
- Now more and more algos are transferring from *rule-based program* to*reinforcement learning program*.
  - *Rule-based program:* expert system (e.g. Siri).
  - *Reinforcement learning program:* free on rules but fixed on objectives (e.g. AlphaGo).

Question

- *Can pricing algos learn to collude in the markets without any guidance or communication?*
- If so, antitrust policies are faced with challenges that non-human collusion is beyond the regulations.
- In this paper, authors use simulations to study this question.
- They study experimentally the behavior of algorithms powered by Q-Learning in a workhorse oligopoly model of repeated price competition.

Findings

- **They found that algorithms consistently learn to charge supra-competitive prices without communicating with one another.**
- The high prices are sustained by collusive strategies with a finite phase of punishment followed by a gradual return to cooperation.
- This finding is robust to asymmetries in cost or demand, changes in the number of players and various forms of uncertainty.

Presentation Outline

Questions: Back to the origin

- What is the consequence of adoption on price levels?
- What **strategies** will AI pricing agents autonomously learn?
- Do they learn to compete or do they learn to collude?
- If learning to collude: what is the appropriate policy/legal doctrine?

Debates

There is a lively cross-disciplinary debate.

- Some are worried (e.g. Ezrachi and Stucke (2016))
- Some are skeptical (e.g. Tadelis and Kuhn (2018))
- Some are agnostic (e.g. Harrington (2018))
- Policy makers take note: Ohlhayusen (Acting Chairman FTC); FTC Public Hearings 2018; OECD 2017 Round-table on Algorithms and Collusion; Vestager (2017 "Algorithms and Competition"); Currie (CMA - 2018); French and German authorities 2019 joint paper, FTC April 2020 Guidelines for AI and pricing...

Conjecturing about consequences, we decided to get our hands dirty.

Approaches

Idea: Study the outcome of repeated oligopoly interaction when **pricing is delegated** to AI-powered algos

1. Theory (Highly intractable)
2. Empirics (Hard)
3. Experiment (✓)

Paper, Motivation and Agenda

- We 'code' AI pricing agents.
- We train them to price in a synthetic oligopoly setting
- How? Playing against clones of themselves
- Contribution: documenting **strategies** learned

Challenges:

- Algos must be similar to those used in markets
- Environments must be sufficiently realistic

Main Results

- Simple AI algos systematically learn to play sophisticated collusive strategies in rich oligopoly environments without communicating.

Artificial Intelligence? Reinforcement Learning

Consider an agent (algorithm)

- facing a **sequence** of choice problems over time
- **observing & shaping** environment in which choices are made
- getting a stream of **rewards**

What a Reinforcement Learning AI algo does at any period:

- choose if to **explore or "exploit"**
- **learns** from experience
- with **reinforcement**: observed rewards affect assessments of actions, thus future actions as well

Markov Decision Process framework

Consider an agent in the following environment

- Time steps $t = 0, 1, 2, ..., T$ (possibly $T = \infty$)
- State space $s_t \in S$
- Action space $a_t \in A(s_t)$
- Scalar Reward π_t
- One step dynamics: (time-invariant) distribution $F(s_{t+1}, \pi_t \mid s_t, a_t)$
- Discount Factor δ

- Agent's problem is to choose a policy $\sigma : S \to A$ That solves:

$$\max_{\sigma} E\left[\sum_{t=0}^{\infty} \delta^t \pi_t\right] \tag{1}$$

Q-value function

- Let σ^* be the optimal policy
- Present Value of action a in state s is:

$$Q(a,s) \equiv E_\pi[\pi \mid a,s] + \delta E_{s'}\left[Q\left(\sigma^\star\left(s'\right),s'\right)\right]$$

or in paper:

$$Q(s,a) = E(\pi \mid s,a) + \delta E\left[\max_{a' \in A} Q\left(s',a'\right) \mid s,a\right]$$

- **Q-learning algorithm** is a Reinforcement Learning (CS) algo designed to iteratively estimate Q and thus σ^*
- How? It experiments/explores the environment...

Q-Learning algorithm in a nutshell

- Initialize setting an arbitrary Q-matrix $Q_{t=0}$ (rows: states; columns: actions)

In each $t = 0, 1, 2, ...$, given Q_t and state s_t,

- with prob. $1 - \varepsilon$ agent chooses 'greedy action' for that state

$$a_t = \arg\max_a Q_t(a, s_t),$$

- with prob. ε agent randomizes over A (EXPLORATION)
- observe the "news": π_t and s_{t+1}
- updates $Q_t \rightarrow Q_{t+1}$ (LEARNING)

How do Q-learning agents learn?

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left(\pi_t + \delta \max_{a\in A} Q_t(s',a)\right) \tag{2}$$

- $\alpha \in [0,1]$ is the (exogenous) learning rate.
- learn cell by cell $\rightarrow$ Slow by design! (Q-matrix grows as time goes by.)

How do Q-learning agents explore?

$$\varepsilon = e^{-\beta t}$$

- Exploration decreases over time
- Actions with 'high' q-values more likely to be played

Why independent Q-Learning?

- **Natural & Model Free** Designed to 'crack' MDPs (Markov Decision Process)
- **Not Fancy:** Tabular method with 3 controls
  Initialization Q_0
  Rate of learning $\alpha \in [0, 1]$
  Rate of experimentation $\beta \geq 0$
- **It works:** In nice (stationary) environments Q_t converges optimal policy Q (Watkins and Dayan, 1992)
- **Popular:** Building block of top-notch Deep Learning algos, e.g. Mniv et al. (2016, *Nature*), e.g. Dutch Shell uses it for pricing at gas stations
- **Successful also in games:** Wins against best human players and other algos in complex games, e.g. Silver et al. (2018, Science), with AlphaGo and chess (DeepMind, Google)

Presentation Outline

Baseline Environment: the economic environment

Baseline economic model: repeated game

- 2 firms/algorithms
- Differentiated goods with Logit demand
- Constant marginal cost c_i
- Profits

$$\pi_{i,t} = (p_{i,t} - c_i) \frac{e^{\frac{a_i - p_{i,t}}{\mu}}}{\sum_j e^{\frac{a_j - p_{j,t}}{\mu}} + e^{\frac{a_0}{\mu}}}$$

  - a_i vertical differentiation
  - μ horizontal differentiation (perfect substitution: $\mu \to 0$)
- Symmetric for algos
- Deterministic in demand and cost

Baseline Environment: implementation of algorithms 1

Baseline algorithms:

- Actions: 15 price points $\left(\underline{p} = \frac{9}{10} p_{\text{Nash}}, \bar{p} = \frac{11}{10} p_{\text{mon}}\right)$
- State: $s_t = \left(p_{t-1}^1, p_{t-1}^2\right)$ (1 period memory)
- Reward: π_t profits (deterministic)

- Simple deterministic transition from prices to profits and states: current prices become next state $\left(p_t^1, p_t^2\right) = s_{t+1}$
- An algo just observes last period prices (state), i.e. perfect monitoring, and own profits (reward), no other info on the environment

Baseline Environment: implementation of algorithms 1

Baseline algorithms:

- Actions: 15 price points $\left(\underline{p} = \frac{9}{10} p_{\text{Nash}}, \bar{p} = \frac{11}{10} p_{\text{mon}}\right)$
- State: $s_t = \left(p_{t-1}^1, p_{t-1}^2\right)$ (1 period memory)
- Reward: π_t profits (deterministic)

- Simple deterministic transition from prices to profits and states: current prices become next state $\left(p_t^1, p_t^2\right) = s_{t+1}$
- An algo just observes last period prices (state), i.e. perfect monitoring, and own profits (reward), no other info on the environment

Baseline Environment: implementation of algorithms 2

In a word, the parameters are:

- $n = 2$ (duopoly) with $c_i = 1, a_i - c_i = 1, a_0 = 0$
- $\mu = 1/4, \delta = 0.95, \xi = 0.1$
- $k = 1$ one-period memory

Important on algos learning in games

1. Algos interact with clones, i.e. self-learning (CS)
2. We use **Independent Learning** (vs. Joint Learning) (CS): no attempt to predict others' actions that are seen as part of the environment (state variable) by our algos

Challenging environment for algos

- Multi-agents/algos interacting (a game)
- Second agent is a source of 'noise' (its exploration creates noises to the other.)
- Guarantees of convergence to optimal policy do not apply
- No supporting theory
- Learning collusion is not an easy task

Presentation Outline

Experimental Approach

- Look at grid of parameters α, β
- 100×100 parametrizations (baseline).
  - $\alpha \in [0.025, 0.25]$ (0.1 suggested in CS);
  - β s.t. number of visits of a cell (a, s) from 4 to 450;
- Agents play (up to) 1 billion iterations per session
- 1000 sessions for each parametrization
- We report averages across sessions and parameterizations

Q-learning means that strategy $\sigma_i^t\left(p_1^{t-1}, p_2^{t-1}\right)$ is:

- random at beginning
- evolves over time
- How? Actions that 'perform well' are reinforced

We observe both prices and strategies and report on both!

Approach for analysis

- For algorithm i, a given Q_t^i implies a best price in any state p_{t-1}: i.e. strategy $\sigma_t^i(p_{t-1})$;
- Q_t^i evolves over time, so $\sigma_t^i(p_{t-1})$ does
- We are interested at:
  1. the resting point of this dynamic process (i.e. after a convergence test is passed)
  2. there we record prices $\bar{p}_i$ and strategies $\bar{\sigma}_i$
  3. we repeatedly do this for 1000 sessions and for different econ parameters and algos

Convergence

- convergence = strategy does not change for 100k iterations.
- 99.9% sessions converge
- takes 850k iterations on average over the grid

Convergence is **slow**

- **Cautionary note**: tabular methods are slow by design
- More efficient algorithm exists and ought to be considered in future work
- Algos need to be **trained** offline! + educated guess online the market (typical approach in CS)

Equilibrium

- There are $225^{15} \times 225^{15}$ equilibrium candidates!
- Impossible to compute the eq'm set
- However, we can test for eq'm!

For example $\alpha = 0.15, \beta = 3 \times 10^{-5}$ (all cells visited $\approx$ 25 times)

- 70% of sessions agents are individually **best-responding** 'on path'
- In 61.5% **mutual best response**, i.e. Nash Equilibrium.
- When do not play Nash, they are **pretty close** (0.2% profits gain left on the table)

$\rightarrow$ Hence, once they learned algos **cannot be exploited**

Cooperation

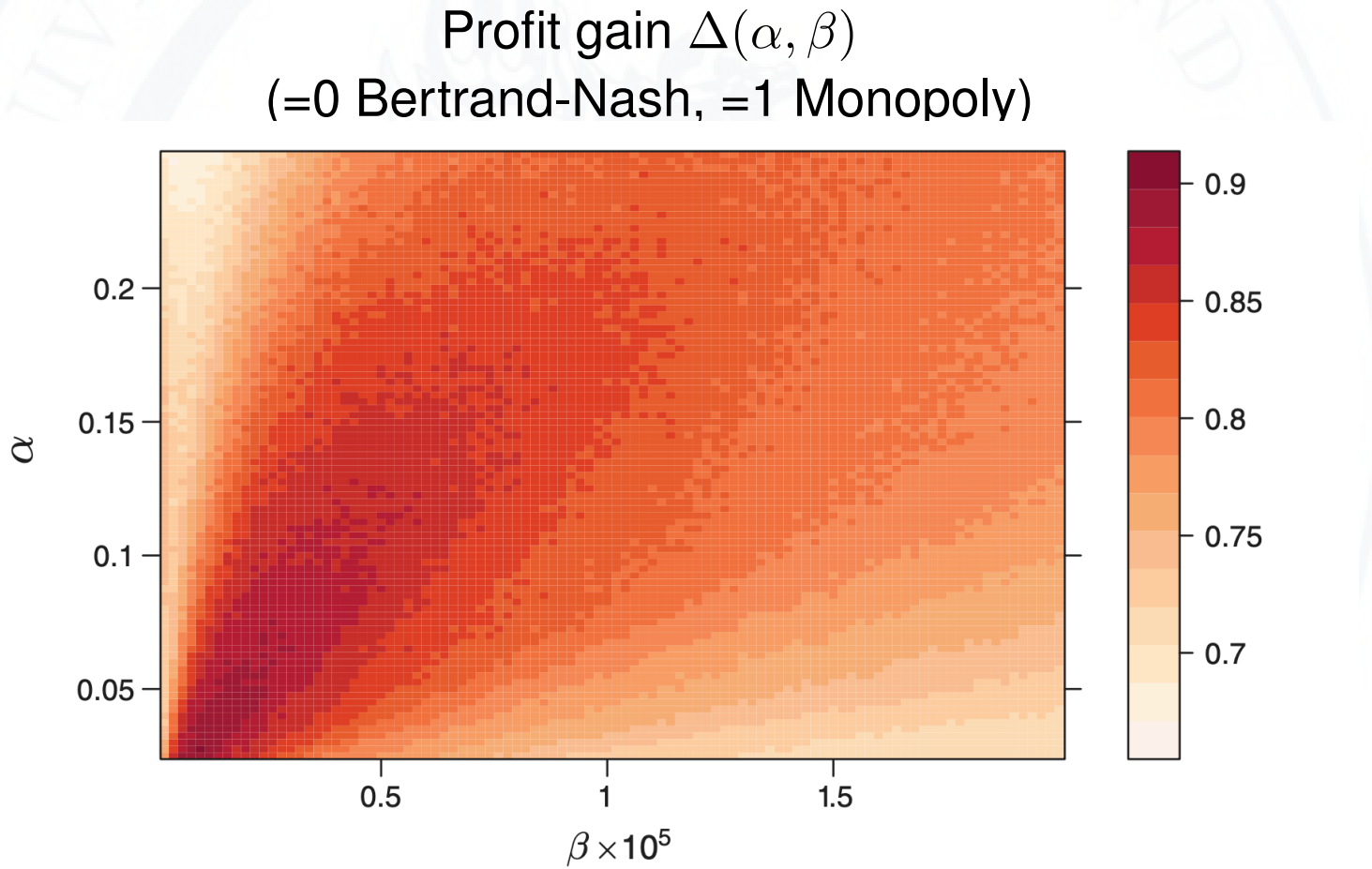
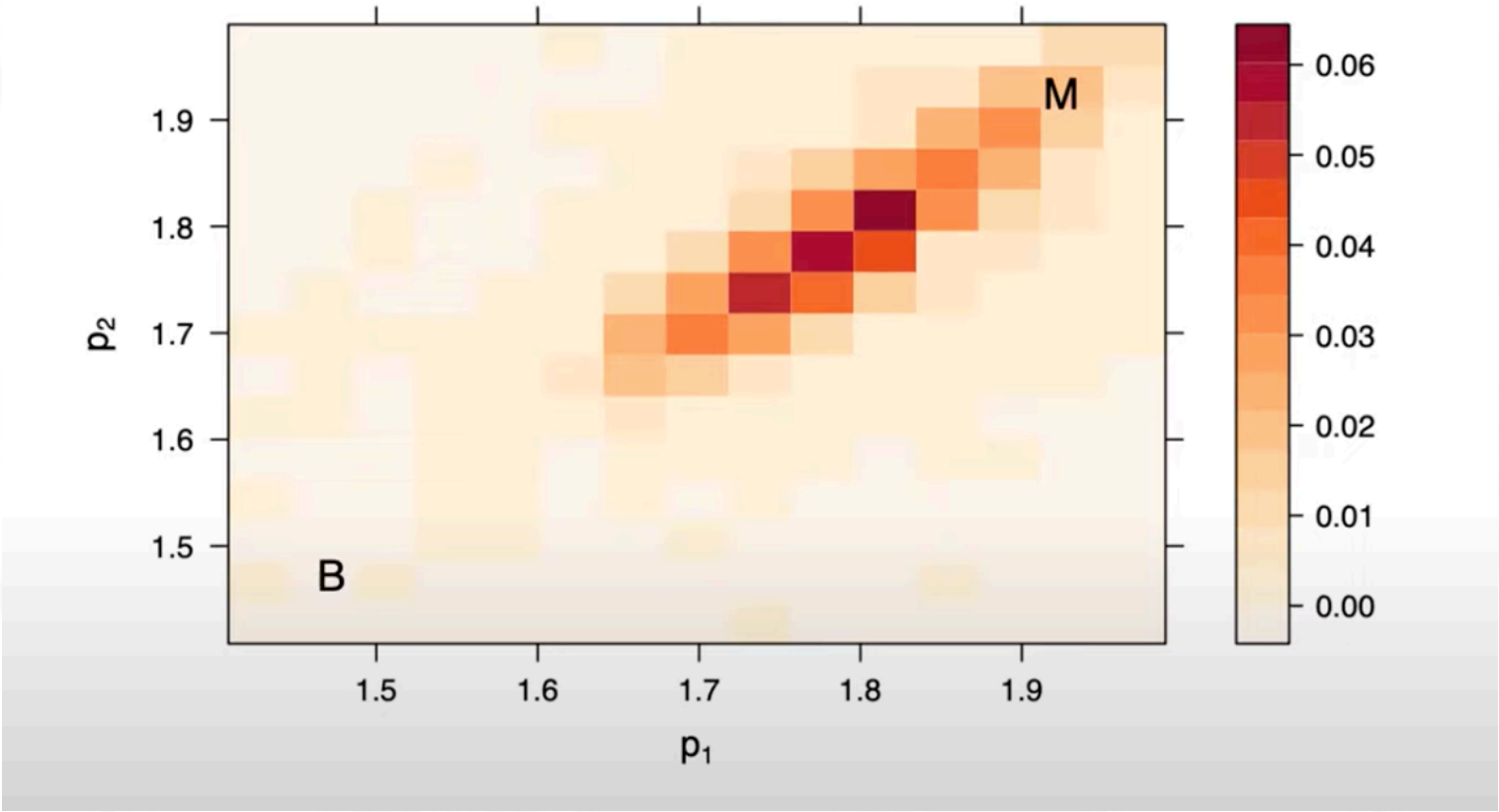


Fig. 1. AVERAGE PROFIT GAIN Δ FOR A GRID OF VALUES OF α AND β

Partial cooperation and price dispersion



B refers to the Bertrand competition (duopoly) price

M refers to the Monopoly price

How are supra-competitive prices supported?

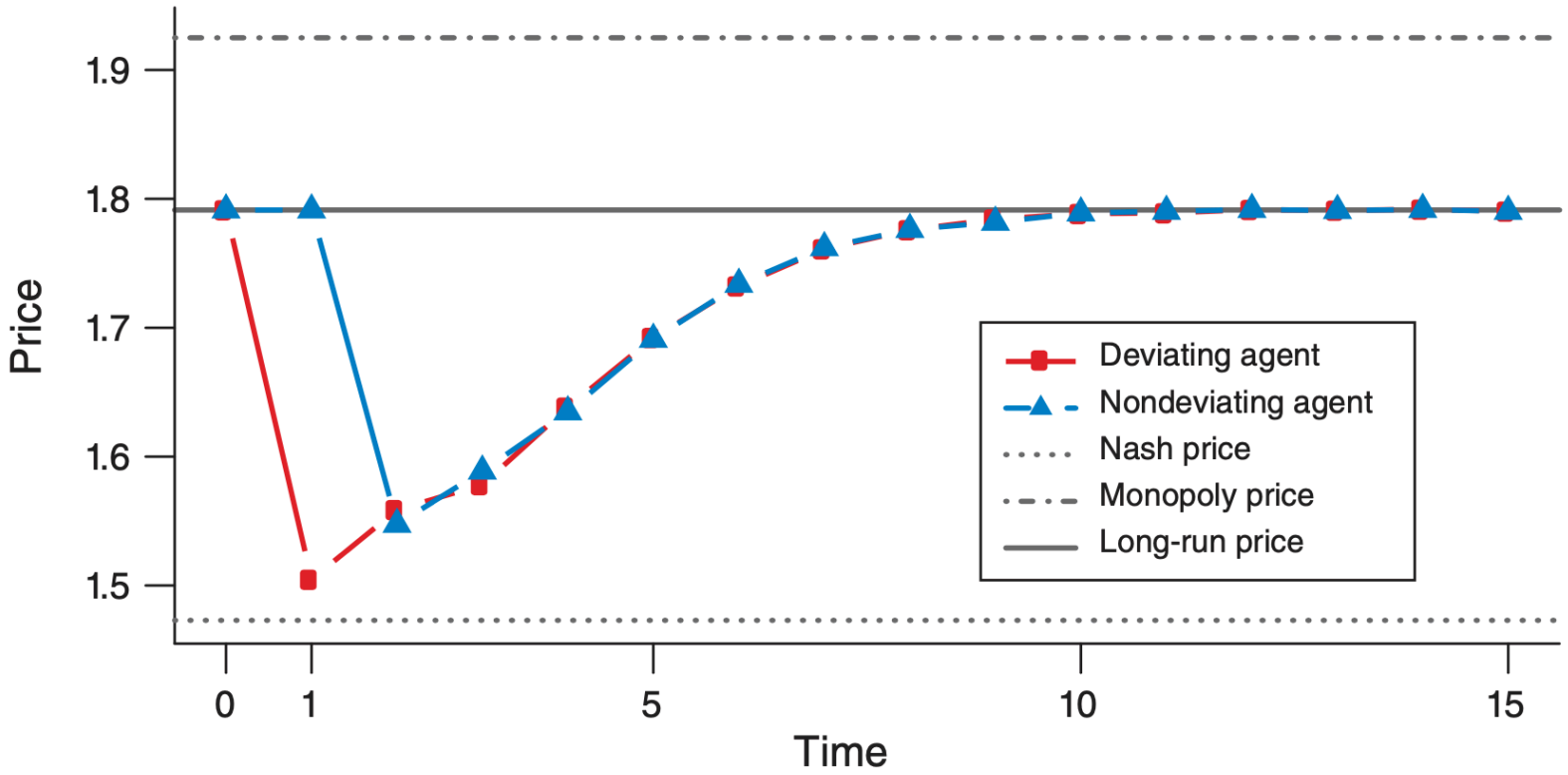
Why algos do not undercut the rivals?

- Do agents fail to learn to compete? Or...
- Do agents actually learn to collude?

- Policy implications radically different (the first, we can go home, the second we must say...)

Learn to collude: Impulse response of prices 1

- Let agents play according to learnt strategies (baseline)
- We force agent 1 (Red) to deviate (one period): lower price in $t = 1$



Learn to collude: Impulse response of prices 2

- Let agents play according to learnt strategies (baseline)
- Small cut leads to larger deviation

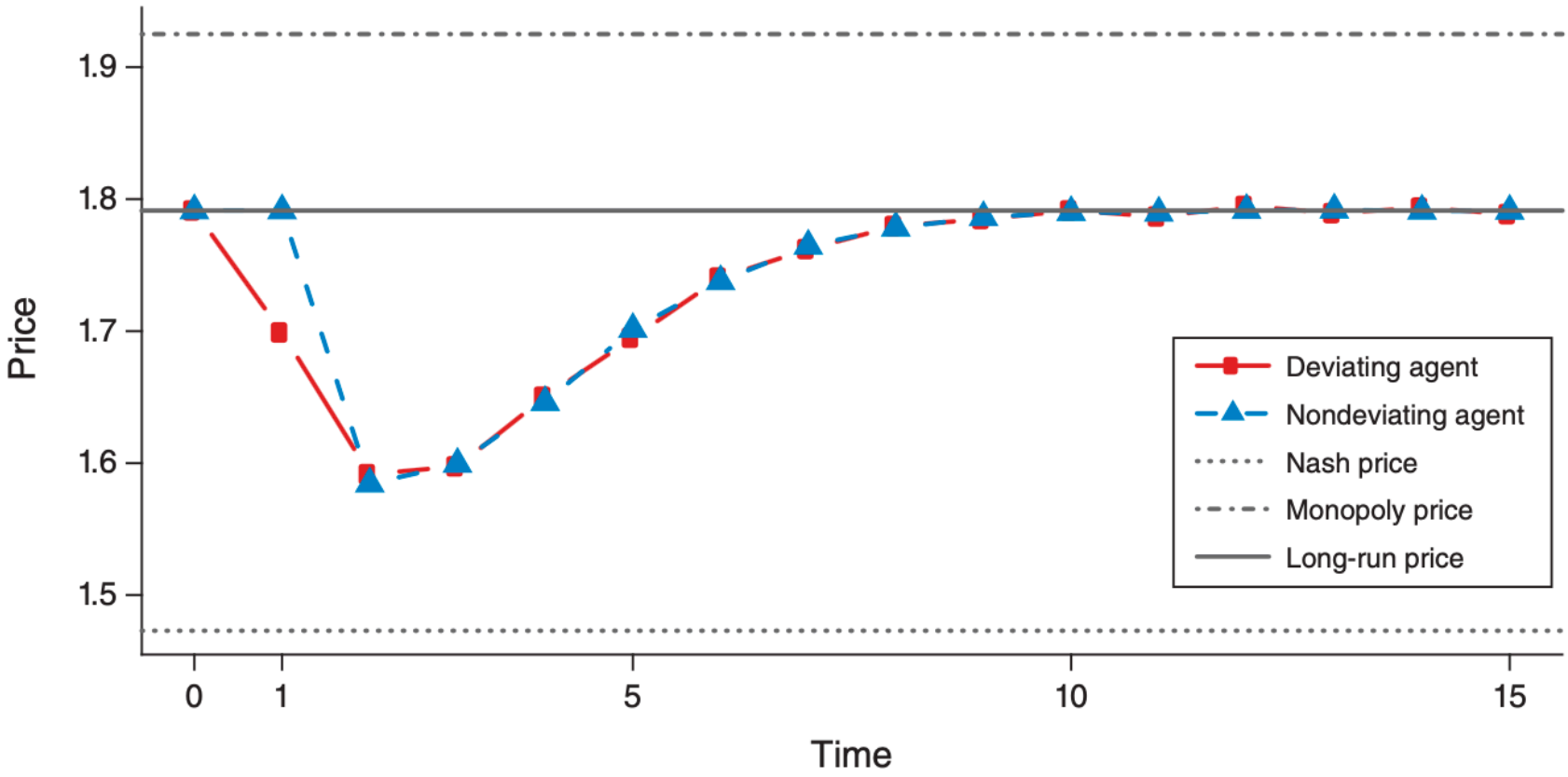
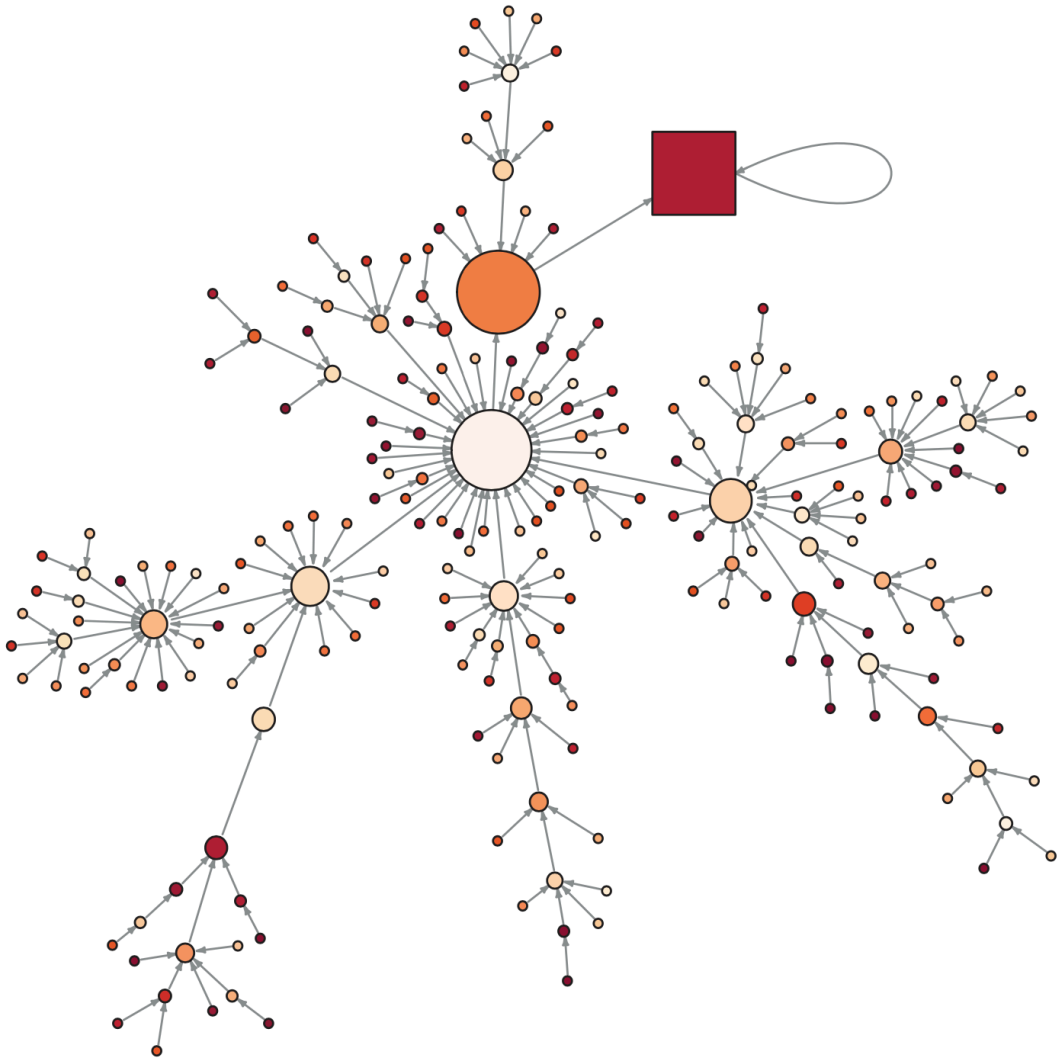


FIGURE 6

Anatomy of collusion: a strategy - betweenness



Presentation Outline

Robustness

- We explored many dimensions:
  - economics
  - complexity of the environment
  - faster-learning
- Main message:
  - (very) high prices still there with changes that make much "economic sense"
  - incentive compatibility and rewards-punishment still there

Factors affecting collusion sustainability

- **Number of Players:** 3 or 4 firms in the markets
- **Asymmetric Firms:** cost and demand asymmetries of different degrees
- **Stochastic Demand:** stochastic demand
- **Variable Market Structure:** stochastic entry and exit of one 'outsider'
- **Product Substitutability:** parameter μ
- **Initialization:** various Q_0 according to other strategies
- **Action Set:** more cells of pricing

Other robustness checks are reported in the online Appendix, including the case of longer memory , linear demand, Boltzmann experimentation, and algorithms with different learning and experimentation parameters.

Presentation Outline

Time scale

Convergence takes periods and could be achieved in years even if each period lasts only minutes.

In this section, we discuss the extent to which this(time scale) limits the practical implications of our results.

- **Transition**
- **Offline Training**
- **Financial Markets**
- **More Advanced Algorithms**
- **More Advanced Algorithms**

Transition

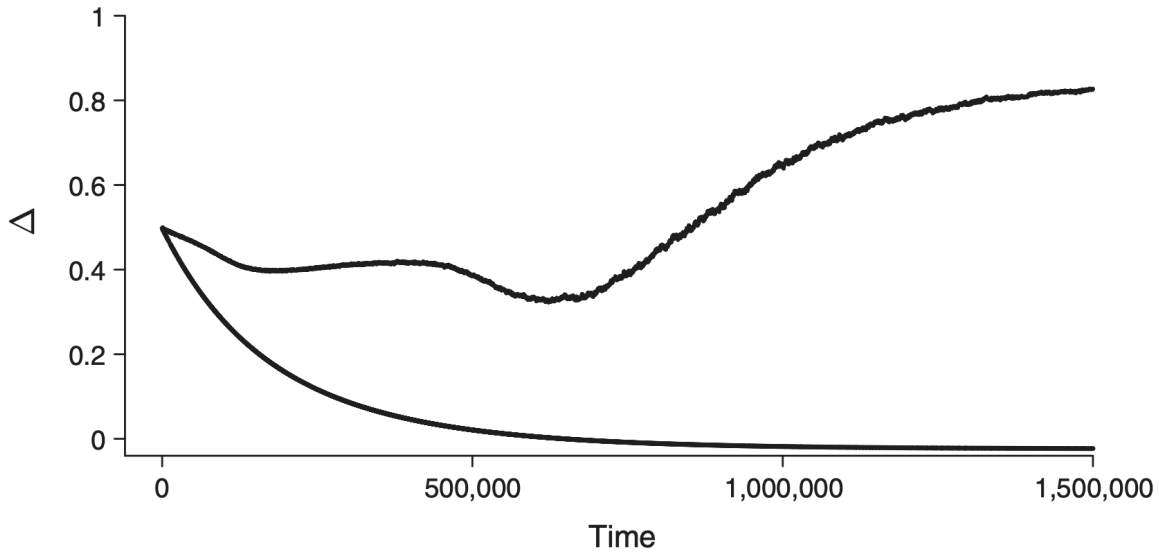


FIGURE 10

Notes: The average profit gain as a function of the number of repetitions (moving average over the last 100 repetitions). The dashed line is the profit gain that results from exogenous exploration, on the assumption that when they do not explore, the algorithms set the Bertrand-Nash price (approximated by defect).

- The difference between two curves are the extra profit gain with collusion, which shows that the collusion has happened in the early periods before algos have completed learning.

Offline training

Offline learning may not be completely useless after all. XD

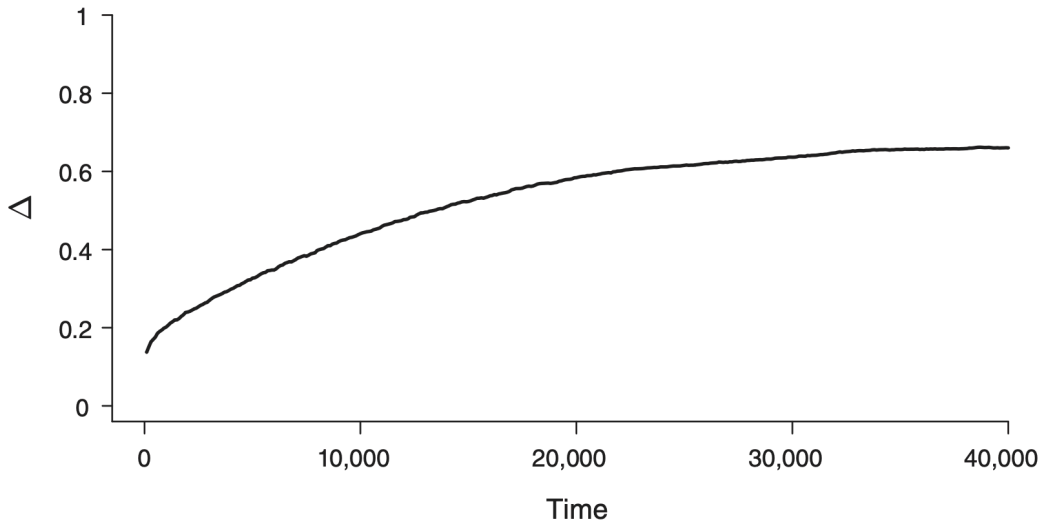


FIGURE 11

Note: The average profit gain as a function of the number of repetitions for pairs of algorithms rematched as described in the text (moving average over the last 100 repetitions).

- Since real markets require coordination in different ways, offline learning is not likely to be representative in the reality.
- The learning is pair-specific but re-matched algos complete learning faster. And the collusion still exists.

The other two concerns

- **Financial markets:** the Q-learning collusion do not apply with higher frequency.
- **More advanced algos:** faster algos require modeling choices that are somewhat arbitrary from an economic viewpoint, resembling black boxes. But it is worth trying.

Presentation Outline

Ongoing analysis and open questions

- **Understanding Learning Process:** Are there key episodes in learning that induce collusion?
- **Increasing complexity:** Highly non stationary economic environments require more more sophisticated algo. Value function approximation with neural-networks, i.e. Deep Learning
- **Meta-game:** Can algos be exploited when learning?

Current legal doctrine

- What if tacit algo collusion detected today?
- Current legal doctrine rooted in conspiracy

- Managers liable only if 'conspire to raise prices'
- Need evidence to convict

$\rightarrow$ In most countries today tacit algo collusion is perfectly legal

There is a reason for this!
However, these standards are clearly inadequate
Thus, new regulations are supposed to be enacted!

See more at *https://www.youtube.com/watch?v=pQFaY6zZzqI*

Discussion