Artificial Intelligence, Algorithmic Pricing, and Collusion

Presenter: Chen-Yu Wei

Outline

- Pricing game under duopoly
- Reinforcement learning and Q-learning
- Empirical results
- Theoretical results

References:

Calvano, Calzolari, Denicolo, Pastorello. **Artificial Intellifence, Algorithmic Pricing, and Collusion**. 2020.

Bertrand, Duque, Calvano, Gidel. **Q-Learnings Can Provably Collude in the Iterated Prisoner's Dilemma**. 2023.

Pricing Game

Duopoly

- Two companies dominate the market
- They sell the same or highly substitutable products

Bertrand Duopoly

- Each company determines the price the product: $p_1,\ p_2$
- The marginal cost is the same: $c_1 = c_2 = c$
- The lower priced company wins all demand; each gets half demand if prices are the same

Price relation	$\boldsymbol{q_1}$	$\boldsymbol{q_2}$
$\boldsymbol{p_1 < p_2}$	$D(p_1, p_2)$	0
$\boldsymbol{p_1 > p_2}$	0	$D(p_1, p_2)$
$\boldsymbol{p_1 = p_2}$	$\frac{1}{2}D(p_1, p_2)$	$\frac{1}{2}D(p_1, p_2)$

- Companies i's profit: $R_i = (p_i - c)q_i$

Bertrand Duopoly

- $p_1 = p_2 = c$ is the unique **Nash equilibrium** (NE), called “Bertrand NE”
  - $p_1 = p_2 > c \Rightarrow$ both of them want to slightly decrease the price
  - $p_1, p_2 \geq c$ and $p_1 \neq p_2 \Rightarrow$ the lower priced one wants to slightly increase the price
  - $p_1 < c$ or $p_2 < c \Rightarrow$ the lower priced one wants set price $= c$

- There is no profit at NE!

- **Collusion:** $p_1 = p_2 > c$
  - Both companies get positive profit
  - But this is not stable…? (everyone wants to deviate)

Prisoner’s Dilemma

	Cooperate	Defect
Cooperate	-1,-1	-10,0
Defect	0,-10	-6,-6

Prisoner’s Dilemma

	High	Low
High	5,5	0,6
Low	6,0	3,3

Pricing game

General form:

	C	D
C	x,x	z,w
D	w,z	y,y

w > x > y > z

Unique NE: (D,D)

One-Shot Game vs. Repeated Game

- One-shot game: play the game once
- Repeated game: play the game repeatedly

In repeated games, the players can make decisions based on the history.
This enlarges the policy space, and new NEs become possible.

For example, if both players use the policy:

> "Start from **always cooperate**. Switch to **always defect** if the other player ever plays defect." (the "grim-trigger" policy)

Then they are in a NE.

Of course, if both players "**always defect**," they are also in a NE (but a different one).

Repeated Game (slightly more formalization)

- Assume **perfect information:** every player knows the actions taken by the other players in all previous rounds
- **State:** the information the player can base their decision on.
  - Full state: At round t, the state is

$$(\boldsymbol{a}_{1:N,1},\ \boldsymbol{a}_{1:N,2},\ \ \dots,\ \ \boldsymbol{a}_{1:N,t-1})$$

  - Memory-limited state (memory size = k): At round t, the state is

$$(\boldsymbol{a}_{1:N,\,t-k},\ \boldsymbol{a}_{1:N,2},\ \ \dots,\ \ \boldsymbol{a}_{1:N,t-1})$$

- **Policy:** a mapping from state to action (or distribution over actions)

Repeated Game (slightly more formalization)

Repeated prisoner's dilemma with memory size = 1

State space = {CC, DD, CD, DC}

Always-defect policy:

$$\pi(\mathrm{CC}) = \mathrm{D}$$
$$\pi(\mathrm{DD}) = \mathrm{D}$$
$$\pi(\mathrm{CD}) = \mathrm{D}$$
$$\pi(\mathrm{DC}) = \mathrm{D}$$

Grim-trigger policy:

$$\pi(\mathrm{CC}) = \mathrm{C}$$
$$\pi(\mathrm{DD}) = \mathrm{D}$$
$$\pi(\mathrm{CD}) = \mathrm{D}$$
$$\pi(\mathrm{DC}) = \mathrm{D}$$

Summary for the paper today

- If both players run **Q-learning** (introduced later) individually in a **pricing game**, then they converge to prices higher than the Bertrand Nash price.

- Since Q-learning is an optimization algorithm, the higher prices could be due to
  - Collusion, or
  - Failure of optimization

- The paper confirms that it is due to collusion
  - The learned policies are robust under forced deviation
  - The learned policies have the punishment mechanism

Implications (negative side)

- The antitrust laws (https://www.justice.gov/atr/antitrust-laws-and-you)

The Clayton Act

This law aims to promote fair competition and prevent unfair business practices that could harm consumers. It prohibits certain actions that might restrict competition, like tying agreements, predatory pricing, and mergers that could lessen competition.

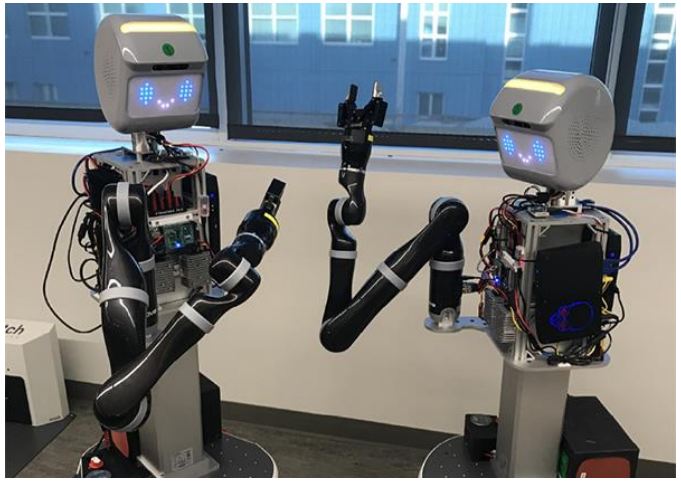
An illegal **merger** occurs when two companies join together in a way that may substantially lessen competition or tend to create a monopoly in a relevant market. This reduction in competition can harm consumers by potentially leading to higher prices or fewer choices for products or services. It can also harm workers by potentially leading to lower wages or fewer choices for employment.

An illegal **tying agreement** happens when a company forces customers to buy one product (the tying product) in order to purchase another product (the tied product). The two products are bundled or "tied" together, which gives the tying agreement its name. This practice restricts a customer's choice and can limit competition. In a fair marketplace, business compete on price and on how good their products are. If an illegal tying arrangement is in place, a seller can use its strong market power on a popular

The collusion by Q-learners are legal, because the collusion is not based on explicit agreement. But it still harms the consumer.

Implications (positive side)

- For example, for multi-agent reinforcement learning, this indicates a **simple** and **decentralized** algorithm that achieves optimal social welfare or globally optimal strategy.



Learning in Games

Learning in Games

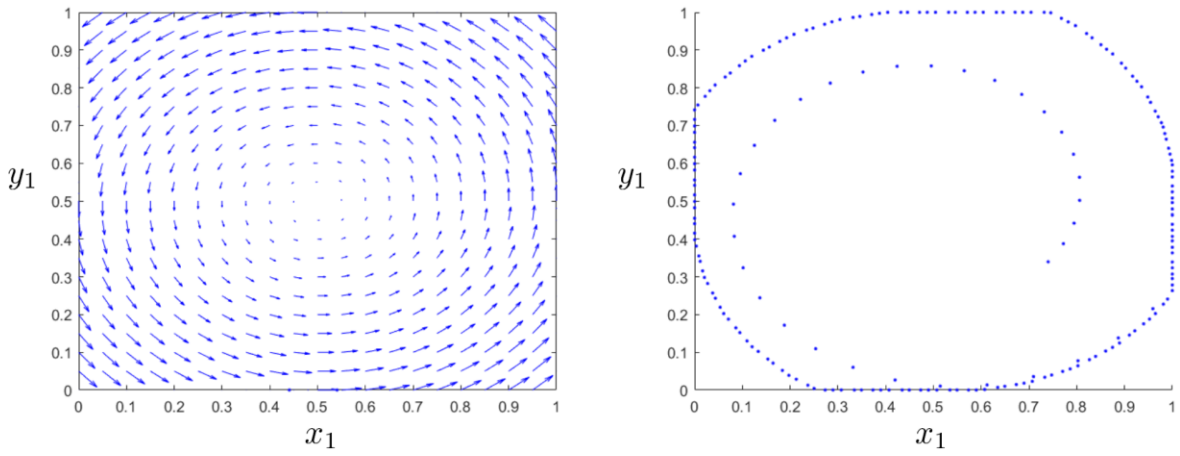
- Game theory focuses a lot on characterizing the (Nash) equilibrium
- “If everyone uses the equilibrium policy, then everyone doesn’t want to deviate”
- However, it is sometimes unclear how the players **reach** such steady state.

Learning in Games

Example. Gradient descent /ascent in two-player zero-sum games leads to *divergence* from the NE.

$$x_{t+1} = x_t + \eta \nabla_x f(x_t, y_t)$$

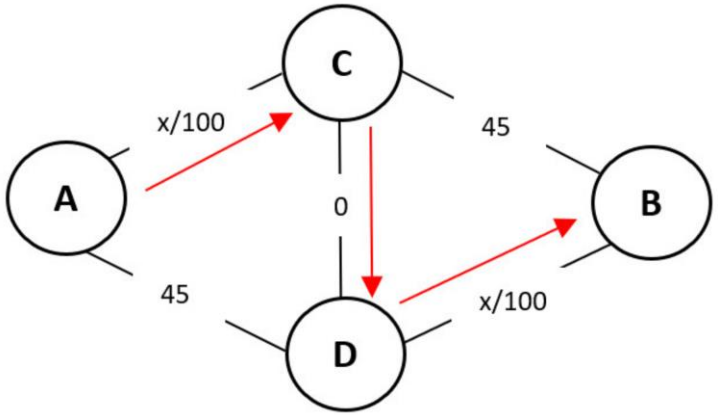
$$y_{t+1} = y_t - \eta \nabla_y f(x_t, y_t)$$



Learning in Games

- There are games with multiple NEs (e.g., repeated pricing game).
  - Different **initialization** and **policy-update algorithms** converge to different NEs

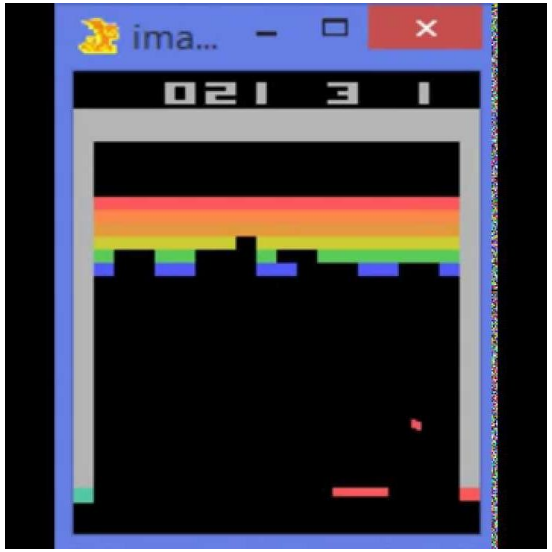
- Usually, games with dominant strategies lead to simpler dynamics (e.g., repeated second-price auction, selfish routing).



Reinforcement Learning and Q-Learning

Reinforcement Learning (RL)

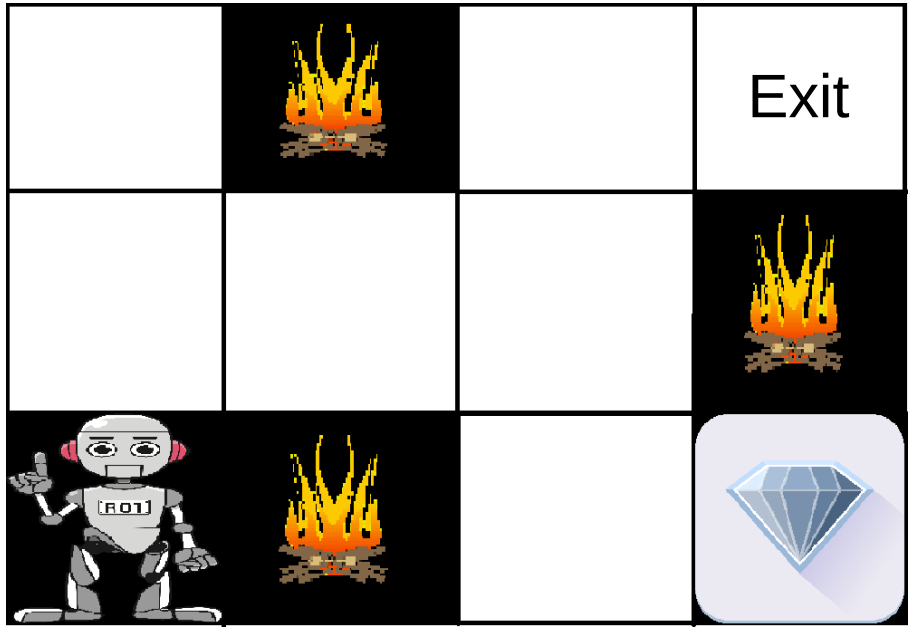
- A framework originally developed for a single player to find optimal policy in a **static** environment.



- **State:** stacking 4 most recent frames (sufficient for us to make decisions, probably)
- **Action:** left / right
- **Reward:** the reward we get in the game

RL: by playing the game repeatedly, find a good mapping from state to action (i.e., a good policy) that maximizes total reward.

Reinforcement Learning (RL)



- **State:**
  - Position of the robot
  - Signals from the sensor (e.g., can detect the objects in 4 neighboring squares)
- **Action:** E / W / N / S
- **Reward:**
  - Fire: -30
  - Diamond: +20
  - Every step before reaching EXIT: -2

Q-Learning

- Idea: dynamic programming
- Try to find a value function Q satisfying

$$Q(s,a) = R(s,a) + \delta\ \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\max_{a'} Q(s',a') \right]$$

then $\pi(s) = \operatorname{argmax}_a\ Q(s,a)$ is an optimal policy (i.e., getting highest cumulative reward in the long-run)

- But we don't know $R(s,a)$ and $P(s'|s,a)$ in advance

Q-Learning

For $t = 1,\ 2, \ldots$

Observe the current state s_t

Choose action $a_t = \begin{cases} \underset{a}{\text{argmax}}\, Q(s_t, a) & \text{with probability } 1 - \epsilon \\ \text{Random} & \text{with probability } \epsilon \end{cases}$

Update

$$Q(s_t, a_t) \leftarrow (1 - \alpha) Q(s_t, a_t) + \alpha \left(R_t + \delta \max_a Q(s_{t+1}, a) \right)$$

Theory for Q-Learning

If

1) State transition probability P(s’|s,a) is fixed over time,
2) Reward function R(s,a) is fixed over time, and
3) every s,a is visited infinitely often,

then Q-learning converges to the fixed point of the Bellman equation (and gives the optimal policy).

Q-Learning in Pricing Games / Prisoner's Dilemma

- The existing theory for Q-learning does not apply to this setting
  - The reward function $R(s, a)$ and the transition $P(\cdot \mid s, a)$ depend on the policy of the other player, which changes over time

Empirical Results

Experiment Setting

Demand as a function of price:

$$q_{i,t} = \frac{\exp\left(\frac{a_i - p_{i,t}}{\mu}\right)}{\sum_{j=1}^{2} \exp\left(\frac{a_j - p_{j,t}}{\mu}\right) + \exp\left(\frac{a_0}{\mu}\right)}$$

Profit:

$$R_{i,t} = \left(p_{i,t} - c_i\right) q_{i,t}$$

$$c_1 = c_2 = 1, \qquad a_1 = a_2 = 2, \qquad a_0 = 0, \qquad \mu = 1/4$$

Experiment Setting

- Price discretization: 15 possible prices
- State: memory size = 1

  state = (my price at round t-1, my opponent’s price at round t-1)

Experiment Setting

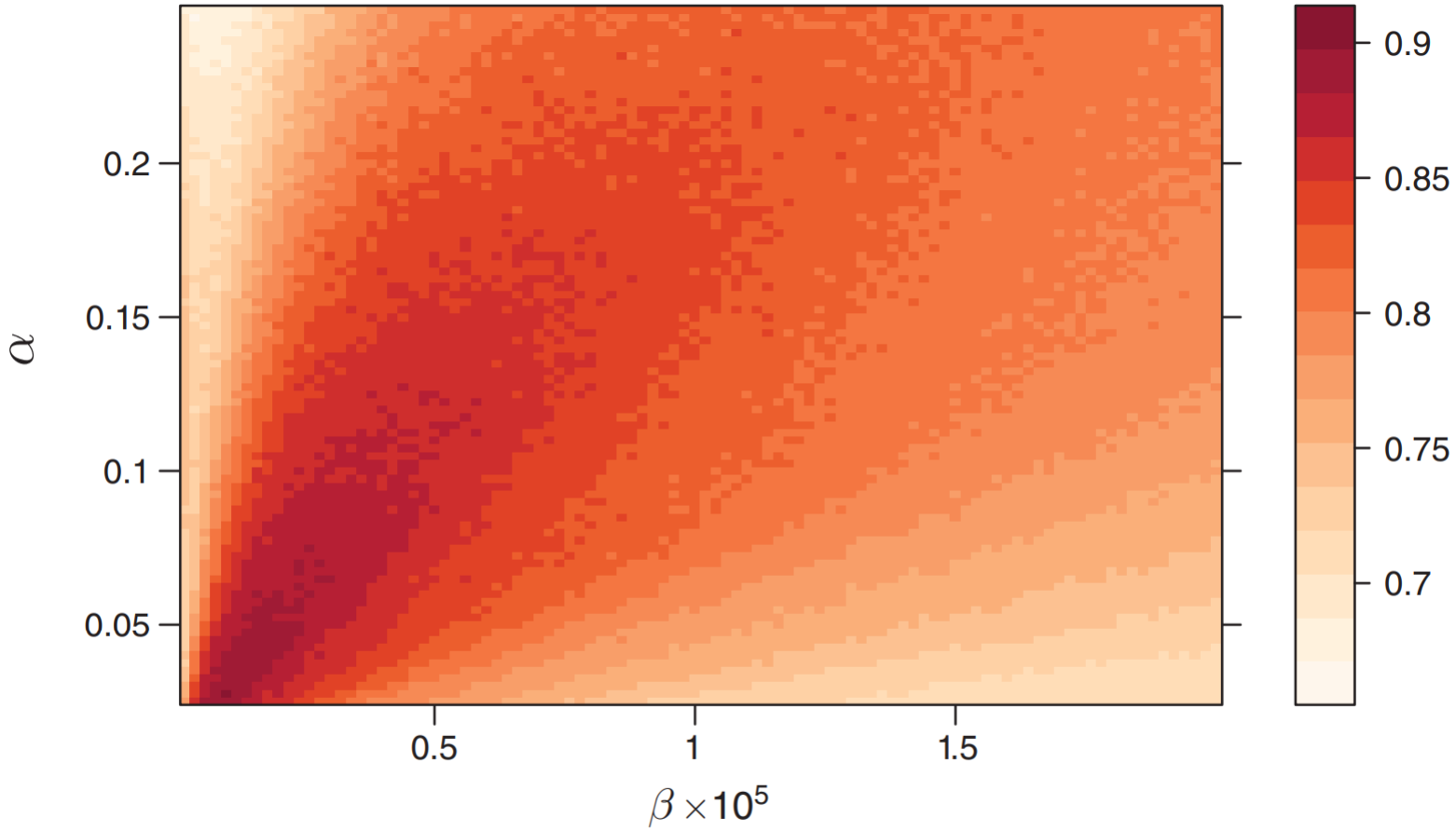
Q-learning parameters (α, β):

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left[R_t + \delta \max_{a'} Q_t(s',a')\right]$$

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

Evaluating the Degree of Collusion

$$\Delta = \frac{\bar{R} - R^{\text{Bertrand}}}{R^{\text{Monopoly}} - R^{\text{Bertrand}}}$$



Frequency of Playing Nash

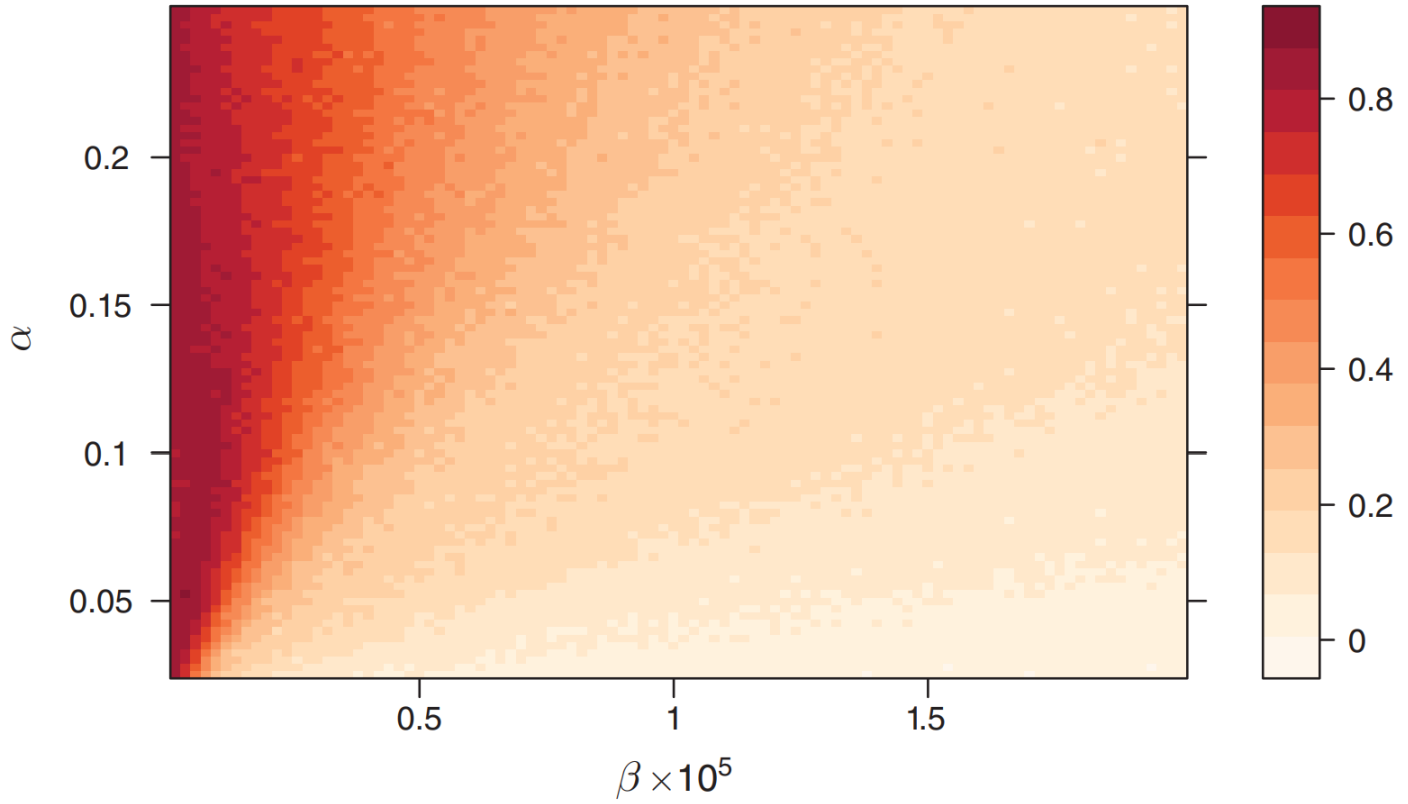


FIGURE 2. FRACTION OF SESSIONS CONVERGING TO A NASH EQUILIBRIUM, FOR A GRID OF VALUES OF α AND β

The effect of Discount Factor δ

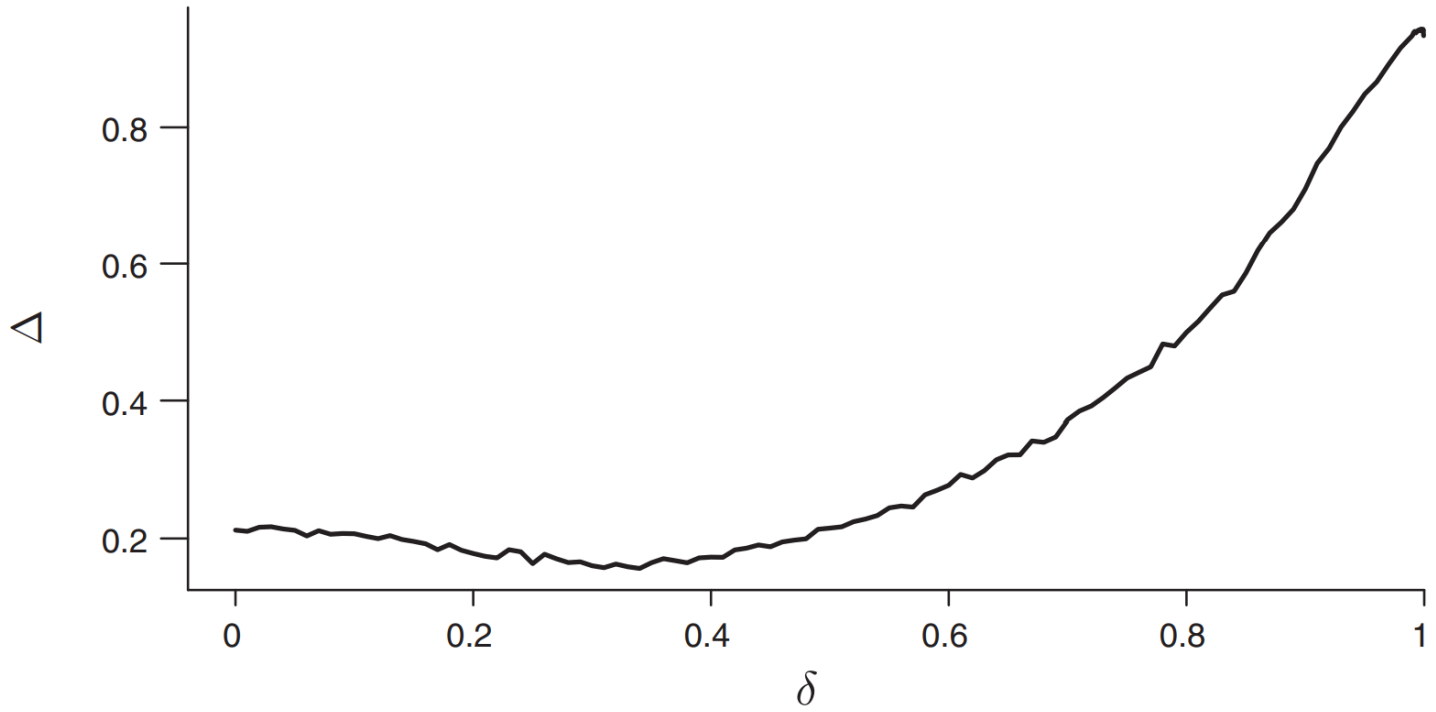


FIGURE 3. THE AVERAGE PROFIT GAIN Δ AS A FUNCTION OF THE DISCOUNT FACTOR δ IN OUR REPRESENTATIVE EXPERIMENT

Deviation and Punishment

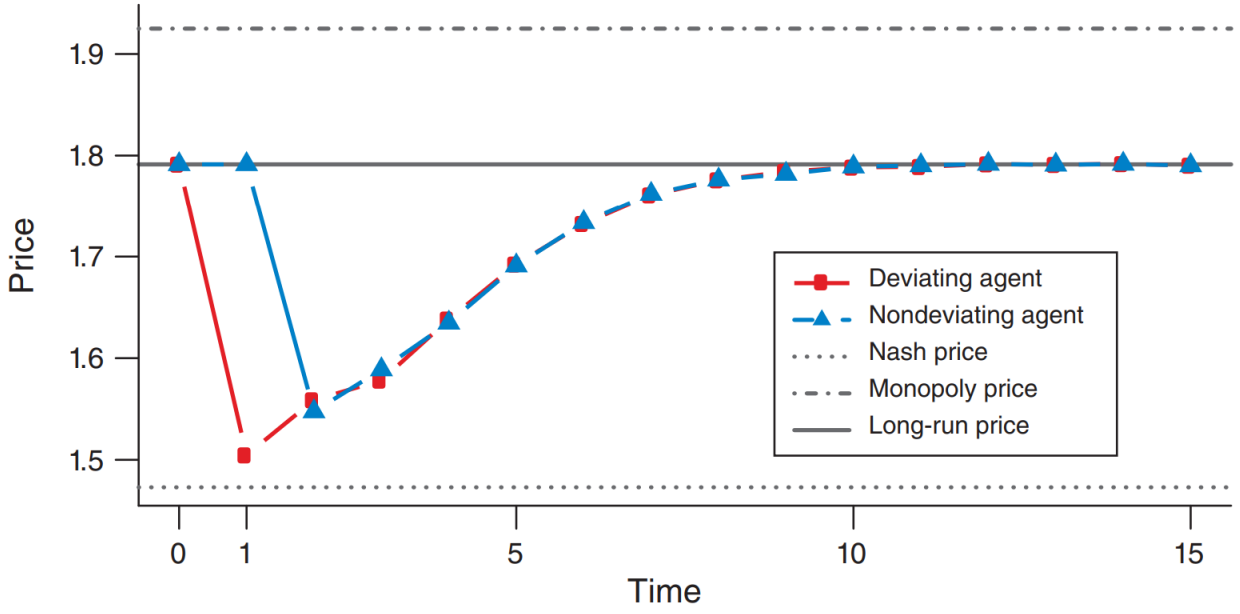


FIGURE 4

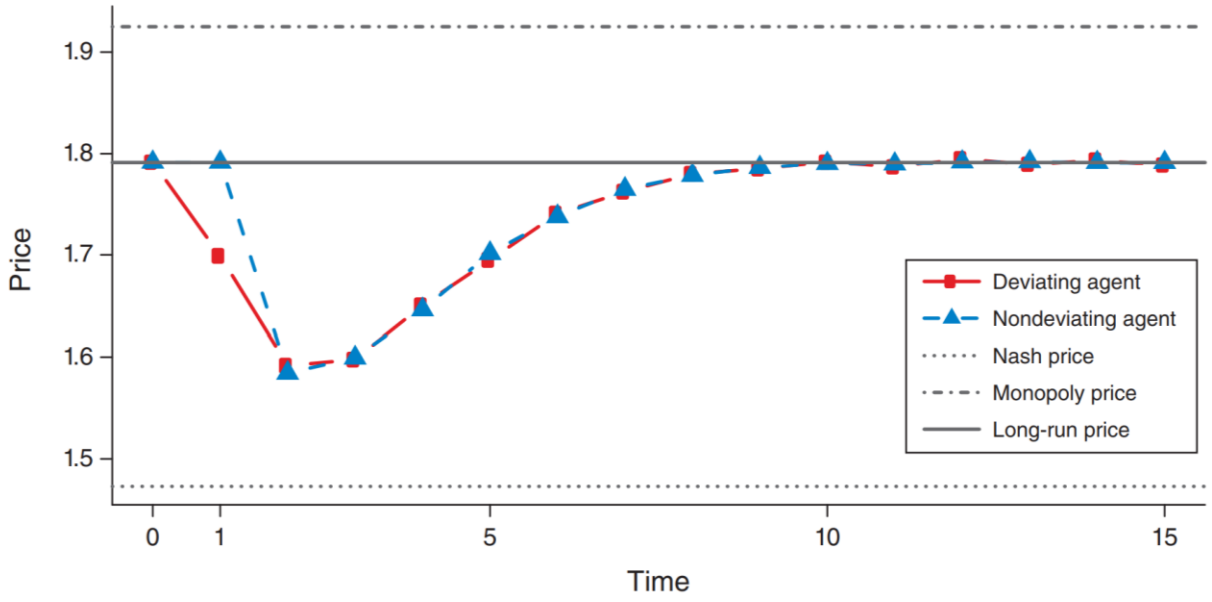


FIGURE 6

Some Theoretical Analysis

Setting

- In Prisoner's dilemma, running Q-learning with
  - Sufficiently large discount factor δ
  - Some conditions on the Q-value initialization
  - Symmetric initialization

  the policies of the players can adapt from **Always-defect** to **Lose-shift** to **Pavlov**

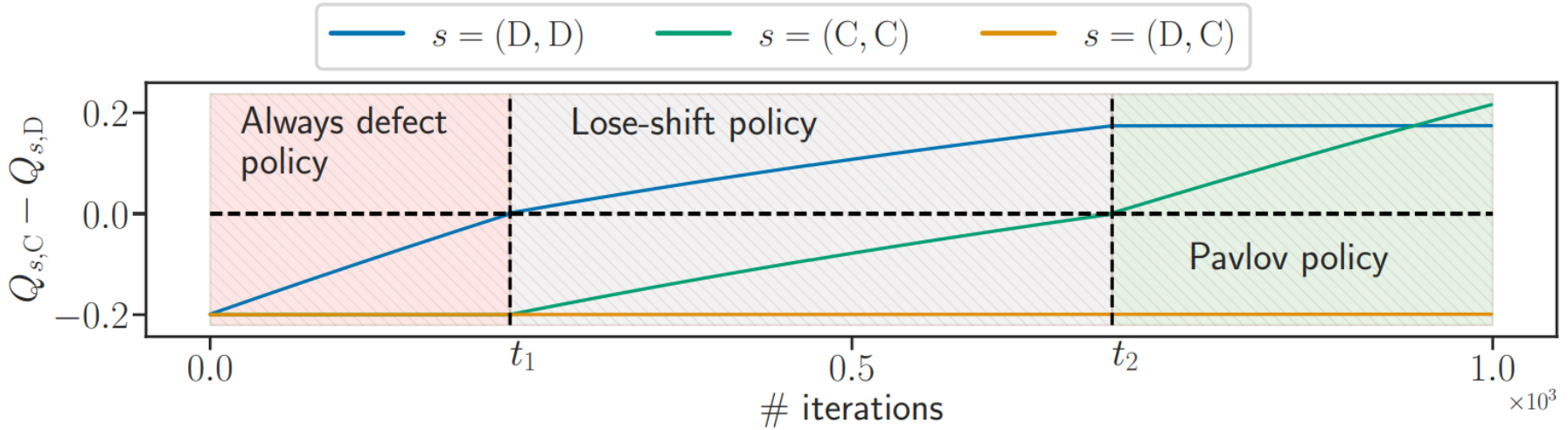
Assumption 7 (Q-values Initialization).

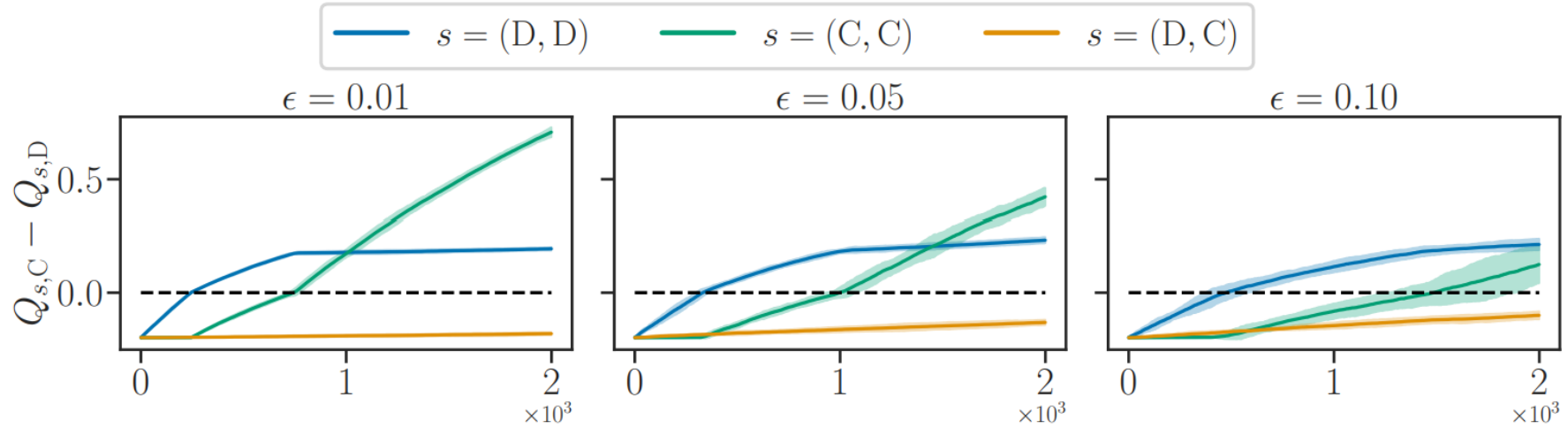
i) $Q^{\star,\text{Defect}}_{(\text{D,D}),\text{D}} \triangleq \frac{r_{\text{DD}}}{1-\gamma} < Q^{t_0}_{(\text{D,D}),\text{C}}$,
ii) $Q^{\star,\text{Lose-shift}}_{(\text{C,C}),\text{D}} \triangleq \frac{r_{\text{DD}}+\gamma r_{\text{CC}}}{1-\gamma^2} < Q^{t_0}_{(\text{C,C}),\text{C}}$,
iii) $Q^{t_0}_{(\text{C,C}),\text{C}} < \frac{r_{\text{CC}}}{1-\gamma} \triangleq Q^{\star,\text{Pavlov}}_{(\text{C,C}),\text{C}}$,
iv) *and for all* $s \in \{\text{C},\text{D}\}^2$, $Q^{t_0}_{s,\text{D}} > Q^{t_0}_{s,\text{C}}$.

Lose-shift: Play { C, if players play (D,D) in the previous round
D, otherwise

Pavlov: Play { C, if players play (C,C) or (D,D) in the previous round
D, otherwise

Experiments





Conclusion

- Repeated games have more complex structure than one-shot games
  - One-shot pricing game / prisoner's dilemma → unique NE
  - Repeated pricing game / prisoner's dilemma → multiple NEs
- The field of learning in games discusses the evolution of the players' policies over time
  - Could converge towards NE (but which NE does it converges to depends on initialization)
  - Could diverge from NE
  - Heavily depends on the learning algorithms they use
- In pricing game / prisoner's dilemma, independent Q-learners can learn to collude / cooperate