Algorithmic Collusion

Reinforcement Learning in Markets and Auctions [1]

Pranjal

EGSO Seminar, 16-03-2023

[1] Thanks to Prof Rust, Prof Miller, Simon, Weipeng, Pete.

Algorithmic Collusion

- Humans are increasingly handing off decision-making to algorithms.
- **Reinforcement Learning** is a branch of AI that deals with autonomous decision-making under uncertainty.
  - Superhuman performance in Chess, Go, Atari Games, Starcraft
  - Idea: Explore at lot, then exploit.
  - Economic applications: pricing, bidding, marketing, trading. More
- **Algorithmic collusion** is when algorithms learn to collude without any human interference and communication.

Evidence

- Chen et al (2016) study 1,641 best-seller products on Amazon and detect that about 543 had adopted algorithmic pricing.
- Assad at al (2020) study Germany's Gasoline market and show that adoption of pricing algorithms increases average margins by 9% in competitive markets and 28% in duopolies.
- Brown and Mackay (2021) find that firms with better algorithms update their prices faster and keep them higher.
- Cavallo et al (2019) show that products on Wallmart that are also on display at Amazon remain on the shelf 20% lesser time.
- A 2017 EU survey found that "Two thirds of them [ecommerce firms] use automatic software programmes that adjust their own prices based on the observed prices of competitors."

Gasoline Price Algorithm

Dynamic Pricing am Point of Sale

Wann ist der richtige Zeitpunkt für eine Preisjustierung, und was ist dann der optimale Preis?

Das Pricing-System „PriceCast Fuel" gibt für jeden Kraftstoff an jedem Standort und zu jedem Zeitpunkt den Verkaufspreis vor, bei dem Sie am besten Ihre Mengen- und Margenziele erreichen. Sie legen die Strategien für Ihre Tankstellen fest, und „PriceCast Fuel" bestimmt zu jedem Zeitpunkt 24 Stunden im Voraus die optimalen Preise, entsprechend Ihren Budgets und Zielen.

„PriceCast Fuel" ermittelt die Preise durch Anwendung von künstlicher Intelligenz sowohl beruhend auf Vergangenheitsdaten als auch auf aktuelle Transaktionsdaten an Ihren einzelnen Tankstellen. Auch lokale Wettbewerbspreise werden

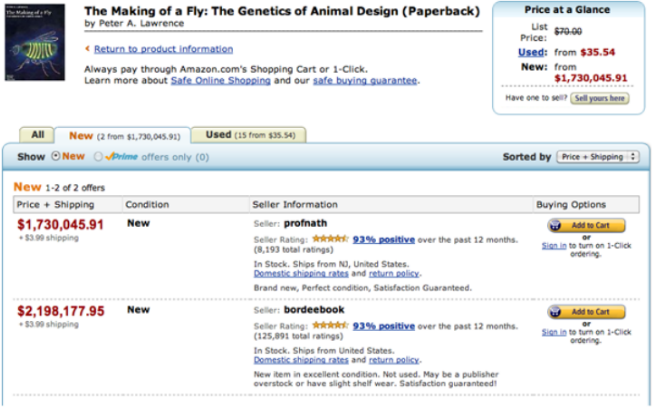
Testen Sie PriceCast Fuel

Je nach Ihrer Markenpositionierung und Ihren Geschäftsziele können Sie mit „PriceCast Fuel" unterschiedliche Strategien einstellen. Am übersichtlichen Kontrollzentrum (Dashboard) des Systems haben Sie jederzeit alle Ihre Tankstellen im Blick: am Ampelsystem – grün, gelb, rot – erkennen Sie sofort, ob Ihre Tankstellen ihre Kennzahlenvorgaben erreichen, und welche Ihre Aufmerksamkeit benötigen.

Buchen Sie unser 14-wöchiges Proof-of-Concept Programm, bei dem Sie an einer Auswahl Ihrer Tankstellen den Effekt von „PriceCast Fuel" überprüfen und nachvollziehen können.

Der Zugewinn an Marge während des Programms übertrifft oft die Kosten für das Proof-of-Concept. Sie werden erleben, wie das tägliche Pricing mit „PriceCast Fuel" transparenter und

24 Million Dollar Book



Environment and Agent

Markov Decision Processes

A MDP is (S, P, R, γ, A):

- (S, P) is a Markov Process
- A is a set of actions
- R is reward matrix that measures $R(s, a, s')$.
- P is transition matrix that measures $P(s'|s, a)$.
- γ is the discount factor

Environment and Agent

Markov Decision Processes

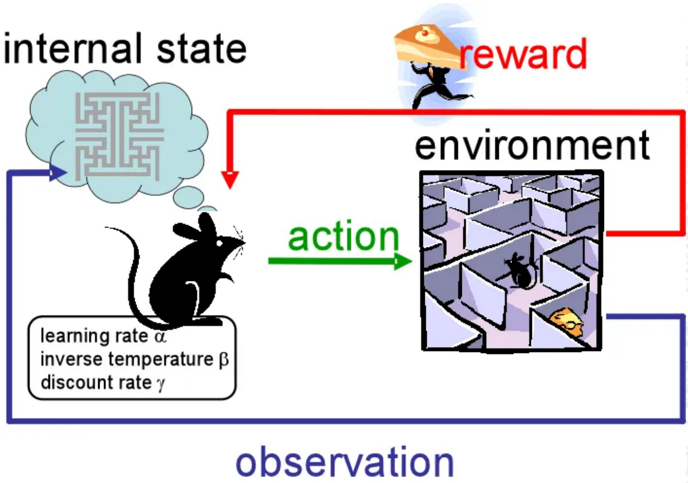
A MDP is (S, P, R, γ, A):

- (S, P) is a Markov Process
- A is a set of actions
- R is reward matrix that measures $R(s, a, s')$.
- P is transition matrix that measures $P(s'|s, a)$.
- γ is the discount factor

Policies

A policy π is a strategy of choosing actions given states.

- $\pi_{a,s} = Prob(A_t = a|S_t = s)$


internal state
reward
environment
action
learning rate α
inverse temperature β
discount rate γ
observation

Functional Equations

Value of a Policy

For each policy π we can "value" states :

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \mid s_0 = s; \pi\right]$$

$$Q^{\pi}(s,a) = r(s,a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, \pi(s_t)) \mid \pi\right]$$

where $s_{t+1} \sim p(\cdot \mid s_{t-1}, a_t = \pi(s_t))$

Functional Equations

Value of a Policy

For each policy π we can "value" states :

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \mid s_0 = s; \pi\right]$$

$$Q^{\pi}(s, a) = r(s, a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, \pi(s_t)) \mid \pi\right]$$

where $s_{t+1} \sim p(\cdot \mid s_{t-1}, a_t = \pi(s_t))$

Bellman Equations

- $V(s) = \max_a r(s, a) + \gamma E[V(s')]$
- $Q(s, a) = r(s, a) + \gamma E[max_{a'} Q(s', a')]$

Q-Learning Algorithm

- Guess $Q_0(s, a)$
- at t, do:
  - observe s_t
  - take action a_t from **exploratory strategy**
  - collect reward r_t
  - observe transition s_{t+1}
  - update Q at point (s_t, a_t):

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha(r_t + \gamma max_{a'} Q_t(s_{t+1}, a'))$$

- Salient Points:
  - Model-free: we do not need to know P and R
  - Off-policy: Evaluate policies from off-policy routes.
  - Incremental: allows for large initial Bellman errors.
  - Watkins and Dayan (1992): with sufficient exploration, we will get to the optimal Q with probability 1.

Exploration and Hyperparameters

- Exploratory Strategies, given s_t:
  - Random: $P(a_t|s_t) = 1/|A|$
  - Greedy: $a_t = \text{argmax}_a Q_t(s_t, a)$
  - ϵ-Greedy: Random with $1 - \epsilon$ and greedy with ϵ
  - Boltzmann Exploration:

$$P(a_t|s_t) = \frac{e^{Q_t(s_t,a_t)/\beta}}{\sum_{a'} e^{Q_t(s_t,a')/\beta}}$$

- Hyperparameters:
  - Learning rate α: weight given to current experiences over past
  - Discount rate γ: weight given to future rewards over current
  - Randomness ϵ: probability of random exploration
  - Temperature β: intensity of explorative-exploitation
  - Initial Valuation Q_0

Demos

- Gridworld
- Hide and Seek

Axelrod (1980)

- Setup: Randomly Repeated Prisoner's Dilemma
- Two Round-Robin tournaments for Game Theorists
- Winner: "Tit-for-tat" - cooperate, then copy opponent's last move.
    - Leads to cooperation against itself and "always-cooperate".
    - Protects itself against "always-defect" and other defectors.
    - Not as harsh as Grim trigger.
    - Suboptimal against random-play.
    - Mistakes can lead to defect-defect spirals.
    - SPNE only if discount factor is high.
    - Tit-for-Tat is not "Evolutionarily Stable Strategy" (ESS).
- Tit-for-tat is a benchmark strategy in generating cooperation in cooperative-conflict games.

Ideas for algorithmic collusion

- "Be nice" - don't defect first, try to break the cycle
- "Retaliate" - Repay kindness and punish defection
- "Be quick" - Delaying leads to ambiguous signals
- "Be clear" - Don't behave randomly
- "Forgive" - When the defector returns to cooperation
- "Don't be envious" - Focus on own reward
- "Exploit the fool" - Defect against random play or always-cooperate
- "Robust to adoption" - Perform well even after everyone copies you

Waltman and Kaymak (2008)

- Setup: Cournot Duopoly with Q-learning Firms
- Simultaneous actions: $q \in [0, 40]$
- Period Reward $\pi = (p(\sum q) - c) * q$
- Without memory or discounting: $Q_{t+1}(q) = (1 - \alpha)Q_t(q) + \alpha\pi$
- Collusion occurs without memory/discounting and with many firms.
    - Collusive-state: $Q(q_C) - Q(q_N) \approx \pi_{CC} - \pi_{NN}$
    - Nash-state: $Q(q_C) - Q(q_N) \approx \pi_{NN} - \pi_{CN}$
    - For low β, $\pi_{CC} - \pi_{NN} > 2(\pi_{NN} - \pi_{CN})$ implies prob of one firm experimenting in collusive-state is lower than prob of both firms experimenting in Nash.
    - If α is high enough, the transition from Nash to Collusion needs only 1 period where both firms experiment together.
    - Firms spend more and more time in collusive-state as β falls.

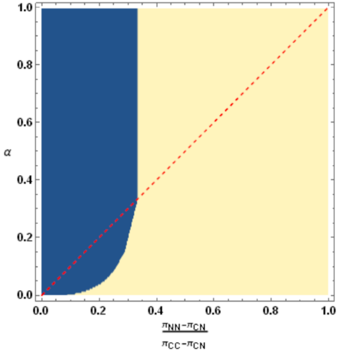
Exploration leads to collusion

Results of computer simulations with firms that did not have a memory

		Nash	$\alpha = 0.05$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 1.00$
$n = 2$	Quantity	24.0	22.8 (1.3)	21.2 (1.4)	20.8 (1.2)	20.8 (1.4)
	Profit	288.0	299.1 (11.6)	312.0 (10.2)	314.7 (6.2)	314.3 (7.0)
$n = 3$	Quantity	27.0	25.1 (1.6)	22.0 (1.8)	21.5 (1.9)	22.1 (1.9)
	Profit	243.0	270.7 (22.9)	304.6 (14.5)	307.8 (14.3)	303.7 (16.6)
$n = 4$	Quantity	28.8	26.3 (1.8)	22.6 (1.9)	22.1 (2.4)	22.9 (2.6)
	Profit	207.4	252.1 (29.8)	299.0 (18.7)	301.4 (19.2)	293.2 (25.8)
$n = 5$	Quantity	30.0	27.6 (1.6)	23.2 (1.8)	22.2 (2.2)	23.3 (2.5)
	Profit	180.0	229.3 (30.0)	294.1 (17.3)	301.1 (19.2)	290.2 (28.7)
$n = 6$	Quantity	30.9	28.3 (1.5)	23.3 (2.2)	22.6 (2.6)	23.1 (3.1)
	Profit	158.7	215.4 (32.2)	290.7 (23.5)	296.3 (27.1)	289.1 (34.8)

Dolgopolov (2022)

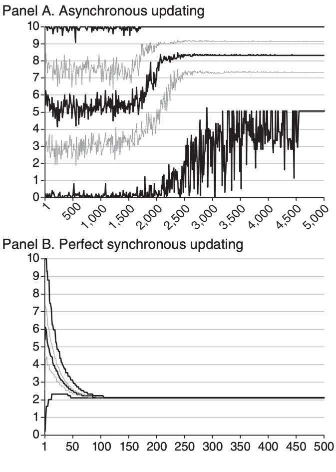
- ϵ-greedy exploration leads to Nash but not Boltzmann.
- Blue parameter region leads to cooperation using Boltzmann.



Asker et al (2022)

- Setup: Static Bertrand with two NE (Discontinuous demand)
- Collusive NE has higher prices/profits. No discounting/memory/exploration.
- Actions $p_i \in [0, 10]$ and rewards $\pi_i = \pi(p_i, p_{-i})$.
- Firm i's initial valuation: $Q^i(p) \sim UNIF(10, 20)$
- at t, firm i greedily selects $p_i = \text{argmax}_p Q^i(p)$ and gets π_i
- Updates $Q^i_{t+1}(p) = (1-\alpha)Q^i_t(p) + \alpha\pi^e(p)$
  - (1) Asynchronous: Update only at $p = p_i$ and $\pi^e(p_i) = \pi_i$
  - (2) Synchronous: Update at all price points $p \in [0, 10]$
- Types of Synchronous Updating:
  - (1) Perfect: $\forall p, \pi^e(p) = \pi(p, p_{-i})$ "demand-knowing"
  - (2) Imperfect: $\forall p, \pi^e(p) = \pi_i$ if (a) $p > p_i$ and $Q_i(p) > \pi_i$ or (b) $p < p_i$ and $Q_i(p) < \pi_i$. This only implies "downward-sloping demand".

Imperfect Updating leads to Collusion

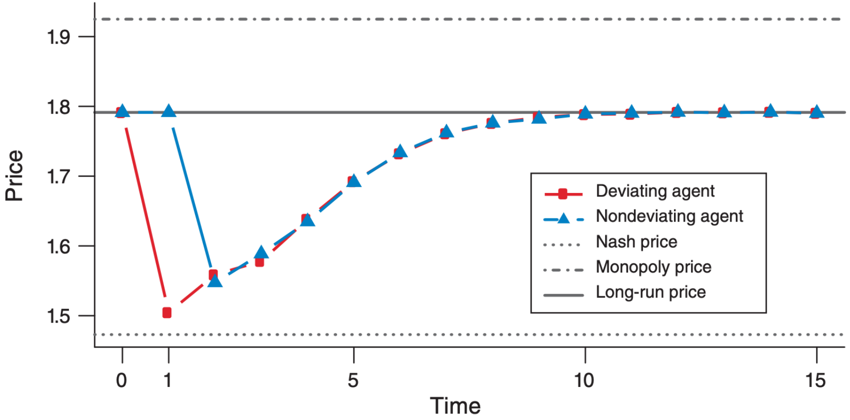


Calvano et al (2020)

- Setup: Bertrand Duopoly with Logit Demand
- Simultaneous actions p lie $[p^c, p^m]$ with Boltzmann exploration.
- Rewards $\pi_{it} = (p_{it} - c_{it})q(p_{it}, p_{-it})$
- State is a set of prices in the last K periods.
- $V(p_-, p_-^{-i}) = \max_p \pi(p, p^{-i}) + \gamma V(p, p^{-i})$
- Average profit gain: $\Delta = \frac{\pi_{cnvg} - \pi_c}{\pi_m - \pi_c}$
- Δ always above 0.5!
- Rises to 0.9 with more experimentation, slower learning, and higher discount rate.
- Robust to increasing the number of firms (2 to 4), adding cost asymmetry and changing uncertainty.

Price Wars

Impulse responses show algorithms have learned to wage price wars.



Klein (2021)

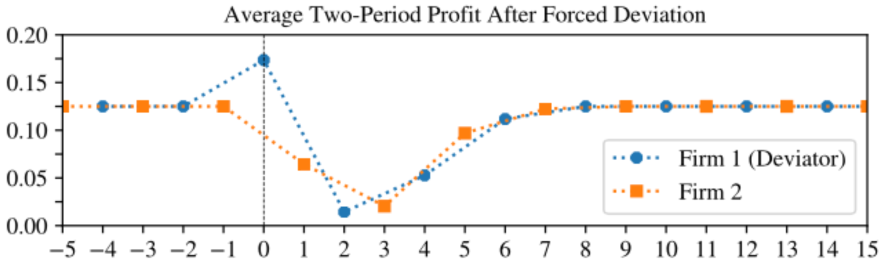
- Setup: Sequential Bertrand Duopoly (Tirole and Maskin 1988)
- Homogenous goods, linear demand, and ϵ-greedy exploration.
- Firm i, state is $p_{-i,t-1}$ and actions $p_{it} \in |P|$ and value function:

$$V(p_{-i}) = \max_{p_i} \pi(p_i, p_{-i}) + \gamma E[\pi(p_i, p'_{-i}) + \gamma V(p'_{-i})]$$

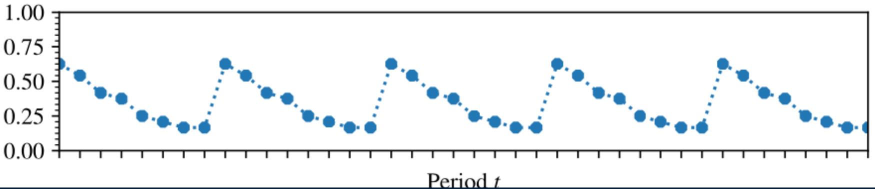
- Theoretical Equilibria:
  - (1) Competitive Constant Prices (at or near cost).
  - (2) Competitive Cyclical Price: Constant undercutting and resetting.
  - (3) Collusive Constant Prices (fear of price war).
- Results: For small $|P|$ we get (3) for high $|P|$ we get (2).

Few Price Points lead to Collusion

Small $|P|$: Avg Profit in Forced Deviation



High $|P|$: Avg. Market Price in Edgeworth Cycle

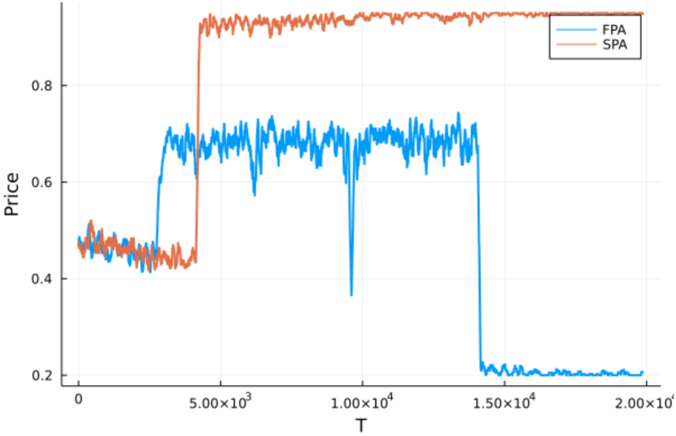


Banchio and Skrzypacz (2022)

- Setup: Q-learning in Repeated First and Second Price Auctions.
- Actions $b_i \in \{b_1, b_2 ... b_m\}$ and valuation $v_i = 1$
- Rewards $v_i - b_{-i}$ (SA) and $v_i - b_i$ (FA).
- Competitive Nash: Both play b_m (FA/SA) or both play b_{m-1} (SA).
- If discount γ high, we can get collusive Nash:
  - Strongly Symmetric - b_1 until deviation
  - Bid Rotation - b_1/b_2 (FA) or b_1/b_m (SA) until deviation
  - After deviation back to Competitive Nash forever.
- Results:
  - SA leads to Competitive Nash while we can see collusion in FA.
  - Revealing bids + synchronous update restores competition in FA.

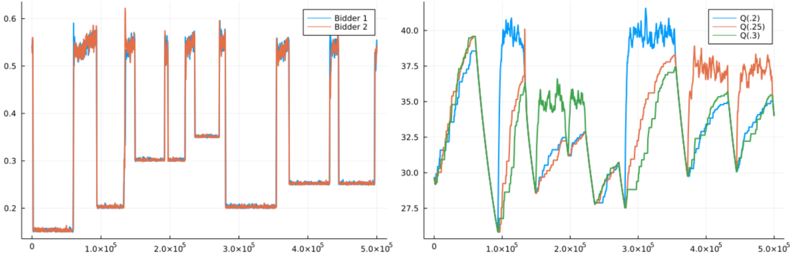
First Price Auctions lead to collusion

Bids vs Games



Exploration breaks stability

Temporary stability only at identical bids; vulnerable to exploration.



(a) Bids (b) Q-values

Deep Q-Learning Networks (DQN)

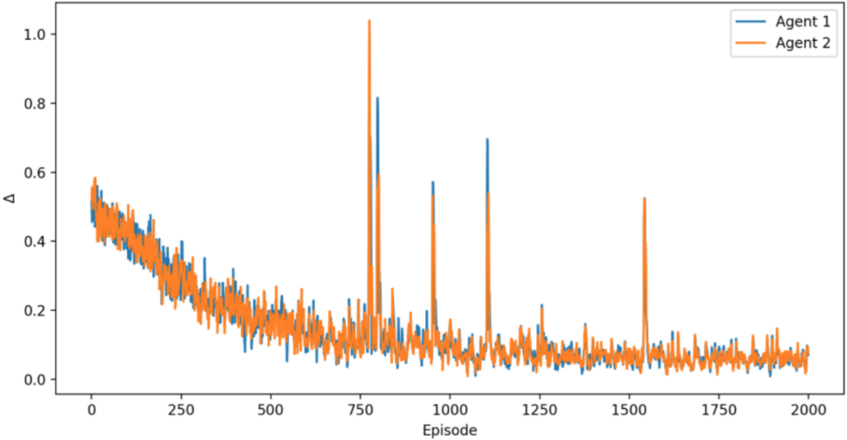
- In practice firms use Deep Networks to represent Q.
- Initialize "target" and "policy" networks: $Q(s, a; \theta_P), Q(s, a; \theta_T)$.
- At t, given s_t use exploratory strategy with $Q(.; \theta_P)$ to get a_t.
- Play game and get $(s_{t+1}, r_t) = f(s_t, a_t)$.
- When you have played sufficient games, draw a sample $(\bar{s}, \bar{s}', \bar{r}, \bar{a})$.
- Bellman Error: $e_t = Q(\bar{s}, \bar{a}; \theta_{P,t}) - (\bar{r} + \gamma E[max_{a'} Q(\bar{s}', a'; \theta_{T,t})])$.
- Loss: $L(\theta_{P,t}) = e_t' e_t + \Omega(\theta_{P,t})$.
- Update Parameters: $\theta_{P,t+1} = \theta_{P,t} - \delta L'(\theta_{P,t})$.
- Update "Target Network": $\theta_{T,t+1} = \tau\theta_{T,t} + (1 - \tau)\theta_{P,t+1}$.
- End when $\theta_P \approx \theta_T$ and they do not change anymore.

Weipeng (2022)

- Setup: Repeated Bertrand with DQN Agents
- Cases:
  - Deep Neural Network (DNN)
  - Recurrent Neural Network (RNN)
  - Long Short Memory Network (LSTM)
- "Experience replay" - large random samples of $(s_t, a_t, s_{t+1}, a_{t+1})$ are drawn **uniformly** from memory to form a loss function.
- This eliminates temporal correlations in "update data"!

Experience Replay eliminates Collusion

Avg of $\Delta = \frac{\pi_{cnvg} - \pi_c}{\pi_m - \pi_c}$ goes to 0.

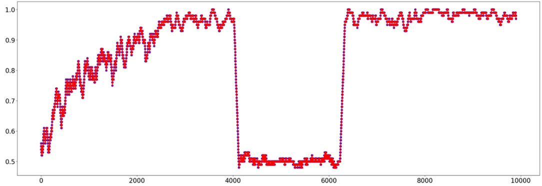


Repeated Prisioners' Dilemma

		Robot 1	
		Cooperate	Defect
Robot 2	Cooperate	$2.5, 2.5$	$0, 3$
	Defect	$3, 0$	$1, 1$

- Robot i: $\max E \sum_t^{\infty} \gamma^t r(a_{i,t}, a_{-i,t})$
- DQN bots with memory of past k games contained in vector s.
- Actions: $a \in \{C, D\}$
- Can we get the Robots to (C, C)?

Case 3: High discounting ($\gamma = 0.99$) and with memory ($k = 1$), and playing against "Tit-for-Tat" leads to sustained cooperation.

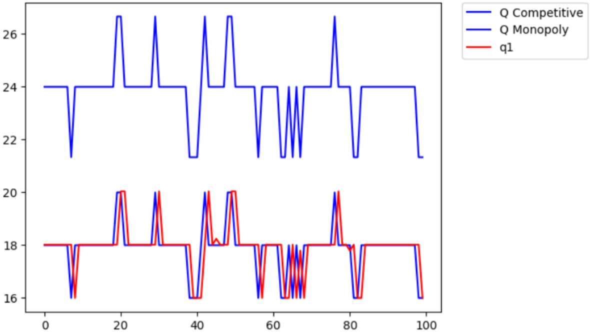


- $Q(D, s = D) = 2.2224$, $Q(C, s = D) = 2.4138$
- $Q(D, s = C) = 2.9495$, $Q(C, s = C) = 3.0665$
- At 4000-6000 iterations, the bot had $Q(D, s = C) > Q(C, s = C)$
- Agent has learned to "always-cooperate" against "Tit-for-tat"

Cournot Duopoly with Stochastic Demand

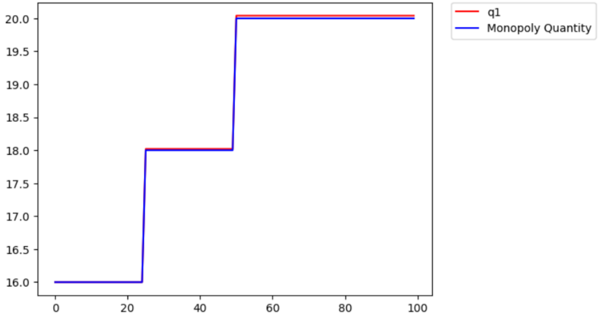
- Setup: Same as Waltman and Kayak (2008)
- Q-learning with Boltzmann Exploration
- Demand: $P = u - v \sum_i q_i$
- $u_t = 40 + e_t e_t \sim \{-4, 0, 4\}$ with known transition matrix P.
- (u_{t-1}, q_{t-1}) is state, q_t is action.

Case 1: Cournot Monopoly



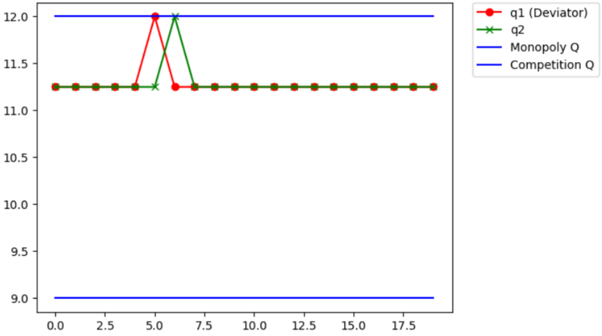
- In the last 100 games, bot responds rapidly to demand.

Case 1: Cournot Monopoly



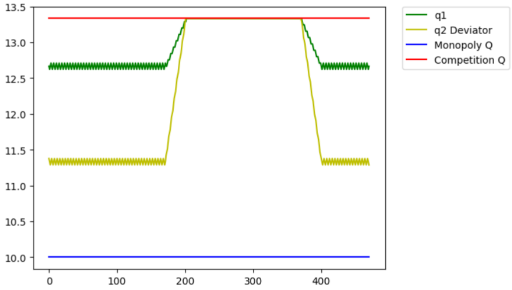
- Bot's impulse response to changing demand.

Case 2: Cournot Duopoly



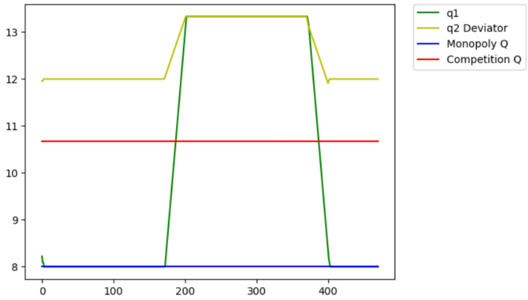
- Impulse Response: Bots learn to wage quantity wars.

Case 3a: Cournot Duopoly in Good times



- Impulse response: Quantity wars when demand rises.
- Here $e_t \in \{-4, +4\}$ with equal probability.

Case 3b: Cournot Duopoly in Bad times



- Impulse response: Quantity wars when demand falls.

What have we learned so far?

- Collusion-inducing strategies will be improved versions of "tit-for-tat".
- Exploration is sufficient for algorithmic collusion.
- Limited information updates can lead to collusion.
- Fewer price points make collusion easier.
- Learning from recent experience necessary for collusion.

More on Algorithmic Collusion

- **Algorithmic collusion** is when algorithms learn to collude without any human interference and communication.
- Tacit collusion i.e price coordination without communication is not covered under US Sherman Act. More
- Economic theory suggests that repeated interaction and high transparency make tacit collusion more likely. More
- Colluding to raise prices is the “meta” strategy for pricing algorithms.
- There is a rise in legal cases related to algorithmic pricing. More

Revenue Management Back

- RM applies in situations where we have:
  - Perishable good in a finite selling season
  - Finite amount of inventory
  - Dynamic pricing and availability
- Applications: transport, hospitality, rentals, retail, entertainment, and advertising.
- Rough Steps:
  - Model demand in monopoly/oligopoly setting.
  - Describe resource constraints.
  - Solve a dynamic program.
- Common use-cases: Overbooking, Offer Management

Regulatory: Anti-Trust Laws Back

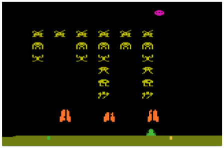
- US law requires evidence of "actionable agreement" over mere interdependent behavior. Firms are free to build algorithms that incorporate current and past information about other firms' prices and price algorithms.
- European Law (Article 101 TFEU) outlaws three types of collusion: agreements, decisions, and concerted practices. This does not include Tacit collusion.

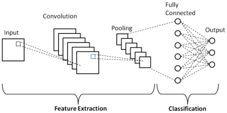
Regulatory: Litigation Back

- In 1994, DOJ prosecuted six airlines that used a common online booking system. Airlines were able to carry out private dialogues via the forum.
- In 2017, the DOJ began an investigation into RealPage which designs rental algorithms.
- In 2016, the DOJ charged two competitors for designing pricing algorithms on Amazon Marketplace that would undercut the rest of the market but not compete any further.

RL: Atari Games Back

- Left image: 84x84 image directly as s instead of hand-crafted features.
- actions are simple “left”,“right”, “up”, “down”.
- Right: One $Q(s)$ (Convolutional Neural Network) for each action.
- Experience Replay: randomly sample from experience when constructing bellman error to prevent use of correlated data. And reuse data.
- ϵ-greedy with ϵ going from 1 to 0.1 over 1 million games.
- Model free: only needs a simulator and does not model MDP.
- Off Policy: Bellman error is constructed assuming a greedy policy in next period when actually the policy taken was exploratory.





RL: Economic Applications Back

- Deng et al (2016) use a Deep Recurrent Neural Network (RNN) to parse financial data on stock and futures and use RL to learn optimal trading strategies.
- Lu et al (2018) use an RL algorithm to handle demand response to energy demand-supply mismatches. Q-learning is used to solve for optimal dynamic pricing in a hierarchical electricity market.
- Cai et al (2017) build a Deep RL model for real-time bidding in online ad auctions - to handle both valuations of ads and strategically bid against opponents. Their model is successful in a live A/B test.
- Zou et al (2019) incorporate RL into recommender systems. They model user behavior in an LSTM and use RL to optimize user engagement through its recommendations.
- Mishra et al (2019) imbed microeconomic theory into a multi-armed bandit to minimize the cost of price experimentation. They demonstrate the success of this method in a field experiment.

Economic Theory: Tacit collusion Back

Ivaldi et al (2002) summarizes that the following facilitates tacit collusion:

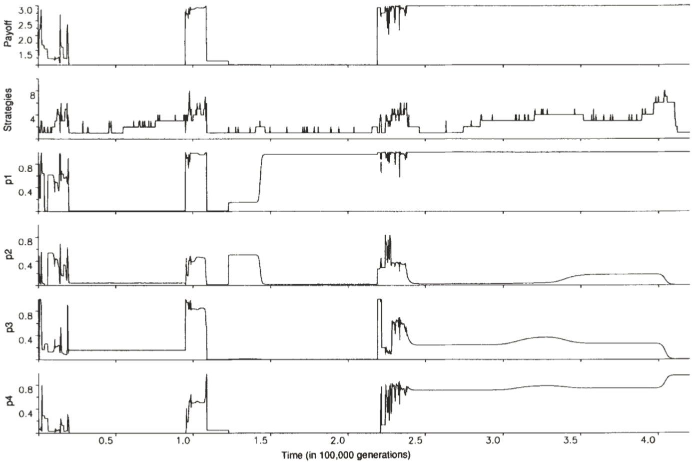
- High discount rate and infinite horizons.
- Less market participants.
- Symmetric conditions in costs, capacity, product, innovation.
- High entry barriers.
- Higher frequency of (inter)action.
- Higher availability of data.
- Growing and less volatile demand conditions.
- Lesser demand elasticity, reduced buyers' power.
- Lesser differentiation and quality improvement.
- Less network effects.
- Mergers reduce competitive pressures but create asymmetries.

Nowak and Sigmund (1993)

- Setup: Evolutionary Prisoner's Dilemma
- State: $s_t \in \{R, S, T, P\}$, Action $a_t \in \{D, C\}$
- Policy/Strategy is $a_t \sim (p_1, p_2, p_3, p_4)$ cooperation probabilities.
  - Always cooperate $(1, 1, 1, 1)$
  - Always defect $(0, 0, 0, 0)$
  - GRIM $(1, 0, 0, 0)$
  - Tit-for-tat $(1, 0, 1, 0)$
  - "Pavlov" i.e. win-stay, lose-shift $(1, 0, 0, 1)$
- Initial population has a large number of strategies.
- Mistakes can happen with ϵ probability.
- High-paying policies have more offspring.
- Even 100 periods we see random mutations random (p_1, p_2, p_3, p_4).

Pavlov is Evolutionary Stable

Emergence: Random, always-defect, TFT, GTFT, GRIM, TFT, Pavlov



Case 1: No discounting ($\gamma = 0$) and no memory ($k = 0$), leads to sustained defection.

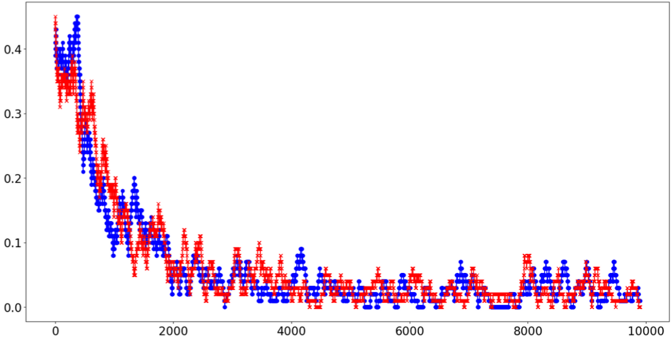
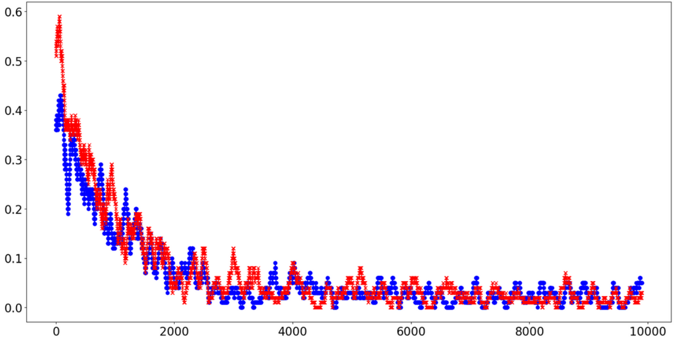


Figure 1: Fraction of Cs in 100 Periods vs Periods

$Q(D) = 1.0645,\ Q(C) = 0.1722$

Case 2: High discounting ($\gamma = 0.99$) and no memory ($k = 0$), leads to sustained defection.



$Q(D) = 1.1062, \ Q(C) = 0.2819$

Next Steps

- Policy-learning and mixed strategies.
- Prioritized Experience Replay.
- Directed exploration and Human-Expert Recommendations.
- Model-based learning.
- One-Shot/Repeated Dynamic Games.
- Reputation and Communication.

Open Questions

How is algorithmic collusion affected by:

Demand

- Multiple products
- Personalized pricing
- Forward-looking consumers
- Network effects
- Brand loyalty
- Buyer Power

Market structure

- Entry/Exit
- Auctions, Matching, Platforms

Learning

- Model-based vs model-free
- On-policy vs off-policy
- Value vs policy learning
- Reputation systems
- Cooperative/Social Learning

Exploration

- Upper confidence bounds
- Noise-based
- Diversity as a virtue

Experimental Design

- Outcomes
  - Avg Profit Gain, Prices and Quantities
  - Impulse Responses to Forced Deviation
- Controls
  - discount rate γ
  - risk aversion ρ
  - learning rate α
  - decay in exploration β
  - modes of exploration
  - size of memory M
  - threat of entry E
  - number of consumer groups G
  - price sensitivity of groups τ_g
  - modes of demand forecasting
  - uncertainty in demand σ
  - persistence in demand ϕ
  - number of firms F
  - number of brands B

Practical Considerations-I

- Q-learning with single agent exploration only is too slow. We need a way for managers to pass on best practises to Q-learning agents.
- Demand is highly seasonal, cyclical and has inertia (brand loyalty,etc.). Business cycle effects can also be considered. Firms thus use demand forecasting.
- Personalization - with better information about consumers, firms are trying to target them with offers/discounts/availabilities, to extract more consumer surplus.
- In reality, firms don't choose quantity sold. They choose both the price and inventory level. Depending on price, demand arrives and depletes inventory. Firms choose size and timing of inventory replinishment.
- Firms will be risk averse. They will be more scared of losing money than making surplus profits. Risk aversion should be a core component of the model.

Practical Considerations-II

- Consumers may be forward looking/strategic. They may understand that firms are optimizing - and they will change their own behaviour accordingly. Can model this with firms setting prices first and then consumers choosing how much and when to buy.
- Some products are time-sensitive and have no prior history. It becomes hard to know about demand in that case. There many be no inventory replinishment either.
- Firms sell many products which can be substitutes/compliments - this effects pricing between them.
- Data collection creates a network effects - firms that are able to get a large user base can exploit that to better understand demand and take away a large chunk of consumer surplus even while locking in customers through loyalty programs.

References-I

- Assad, Stephanie, Robert Clark, Daniel Ershov, and Lei Xu. 'Algorithmic Pricing and Competition: Empirical Evidence from the German Retail Gasoline Market'. SSRN Electronic Journal, 2020. https://doi.org/10.2139/ssrn.3682021.
- Bandyopadhyay, Subhajyoti, Jackie Rees, and John M. Barron. 'Reverse Auctions with Multiple Reinforcement Learning Agents'. Decision Sciences 39, no. 1 (February 2008): 33–63. https://doi.org/10.1111/j.1540-5915.2008.00181.x.
- Brown, Zach Y, and Alexander MacKay. 'Competition in Pricing Algorithms', n.d.
- Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. 'Artificial Intelligence, Algorithmic Pricing, and Collusion'. American Economic Review 110, no. 10 (1 October 2020): 3267–97. https://doi.org/10.1257/aer.20190623.
- Cavallo, Alberto. 'More Amazon Effects: Online Competition and Pricing Behaviors'. Cambridge, MA: National Bureau of Economic Research, October 2018. https://doi.org/10.3386/w25138.
- Chen, Le, Alan Mislove, and Christo Wilson. 'An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace'. In Proceedings of the 25th International Conference on World Wide Web, 1339–49. Montréal Québec Canada: International World Wide Web Conferences Steering Committee, 2016. https://doi.org/10.1145/2872427.2883089.
- Klein, Timo. 'Autonomous Algorithmic Collusion: Q[U+2010]learning under Sequential Pricing'. The RAND Journal of Economics 52, no. 3 (September 2021): 538–58. https://doi.org/10.1111/1756-2171.12383.

References-II

- Leisten, Matthew. 'Algorithmic Competition, with Humans', n.d. Mehra, Salil K. 'Antitrust and the Robo-Seller: Competition in the Time of Algorithms'. MINNESOTA LAW REVIEW, n.d.
- Noel, Michael D. 'Edgeworth Price Cycles and Focal Prices: Computational Dynamic Markov Equilibria'. Journal of Economics Management Strategy 17, no. 2 (June 2008): 345–77. https://doi.org/10.1111/j.1530-9134.2008.00181.x.
- Tesauro, Gerald, and Jeffrey O. Kephart. 'Pricing in Agent Economies Using Multi-Agent Q-Learning'. In Game Theory and Decision Theory in Agent-Based Systems, edited by Simon Parsons, Piotr Gmytrasiewicz, and Michael Wooldridge, 5:293–313. Multiagent Systems, Artificial Societies, and Simulated Organizations. Boston, MA: Springer US, 2002. https://doi.org/10.1007/978-1-4615-1107-6-14.
- Waltman, Ludo, and Uzay Kaymak. '-Learning Agents in a Cournot Oligopoly Model'. Journal of Economic Dynamics and Control 32, no. 10 (October 2008): 3275–93. https://doi.org/10.1016/j.jedc.2008.01.003.
- Zhao, Jun, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. 'Deep Reinforcement Learning for Sponsored Search Real-Time Bidding'. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 1021–30. London United Kingdom: ACM, 2018. https://doi.org/10.1145/3219819.3219918.