Platforms and Algorithmic Pricing: Collusion and Self-Preferencing

Matteo Courthoud

October 27, 2021

Setting

- Algorithmic pricing is widespread
  - High frequency markets
  - Or multiple interactions
- Recent evidence on **algorithmic collusion**
  - Both theoretical (Calvano et al., 2020)
  - And empirical (Assad et al., 2020)
- These markets are usually private **platforms** that act as aggregators
  - E.g. Amazon Marketplace (Chen et al., 2016)
- Have the ability (and objective) of steering **consumers' attention**
  - E.g. Ranking of search results

Can platforms have a role in managing algorithmic pricing collusion?

Ranking Platforms

- Different platform incentives depending on the business model
  - Per-sale fee: incentives aligned with consumer surplus
  - Revenue fee: ambiguous
  - Profit fee: incentives aligned with firm profits
- **Q**: Is the platform able to foster/disincentivize collusion just by ranking firms?
- Self-preferencing: sometimes these platforms are also active as sellers
- **Q**: How does the platform dual role impact outcomes?
  - Can fairness be achieved? How?

Contribution

1. Show that **algorithmic ranking can impact algorithmic collusion**
    - Platform charging profit fee: facilitates collusion
    - Platform charging quantity fee: hinders collusion

2. Show what happens with the **platform dual role**
    - Baseline: the platform gives preferential treatment to own product
    - Distortion persists even with **separation of ranking and sales divisions**

Literature

- Q-learning algorithms learn grim-trigger strategies
  Calvano et al. (2020); Klein (2019)
- Empirical evidence of the use of pricing algorithms
  Chen et al. (2016)
- Empirical evidence of colluding pricing algorithms
  Assad et al. (2020)
- Economic modeling can help preventing collusion
  Asker et al. (2021)
- Algorithms on platforms: excluding the highest price is not sufficient
  Johnson et al. (2020)

Reinforcement Learning Overview

Q-Learning: Overview

Q-learning algorithms are **model-free** algorithms designed to find optimal policies in dynamic environments.

- Objective: maximize flow of discounted payoffs $\mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t \pi_t\right]$
- In each period, an agent observes a state $s \in S$ (finite)
- The agent has to take an action $a \in A$ (finite)

Note

- Not designed to be deployed in strategic environments
- Most optimality results on Q-learning, derived for stationary environments

Q-Learning: Overview

What does **model-free** means?

The algorithm does not rely on assumptions nor tries to model

- The relationship between states, actions, and payoffs $\pi(s, a)$
- The relationship between states, actions, and future states $pr(s'|s, a)$

Feedback:

- The algorithm knows its own state and actions
- At the end of each periods it observes the realized payoffs and the next state

Q-Learning Algorithm

Timing

- Action-specific value function $Q(s, a)$ is initialized
- In each period
  1. The algorithm observes the state s
  2. Two different ways of selecting an action a
     - **Exploration**: the algorithm takes a random action $a \in A$
     - **Exploitation**: the algorithm takes the action $a = \arg\max_a Q(s, a)$
  3. Algorithm observes the realized payoff $\pi(s, a)$ and next state $s'(s, a)$
  4. Algorithm updates $Q(s, a)$ using $\pi(s, a)$ and $s'(s, a)$
     - Weighted average of observed payoffs in state s, given action a
- Convergence when actions are not updated for enough periods

Setting and Simulation Results

Setting

- Two firms, infinitely repeated game
- Unit mass of consumers with utility

$$u_i(\boldsymbol{p}) = v - \mu p_i - \varepsilon_i \qquad with \qquad \varepsilon_i \sim Gumbel(0, 1)$$

- Logit **demand**

$$q_i(\boldsymbol{p}, r) = \frac{e^{-\mu p_i}}{e^{-\mu p_i} + e^{-\mu p_j} + e^{-\mu_0}}$$

- Platform decides which items to show to consumers
  - After prices are set

Payoffs

Static payoffs of the firms and of the platform

- **Without fees**

$$\pi_i(\boldsymbol{p}, r) = q_i(\boldsymbol{p}, r) \cdot (p_i - c) \qquad \pi_p(\boldsymbol{p}, r) = 0$$

- **Profit fee**

$$\pi_i(\boldsymbol{p}, r) = q_i(\boldsymbol{p}, r) \cdot (p_i - c) \cdot (1 - f) \qquad \pi_p(\boldsymbol{p}, r) = f \cdot (\pi_i + \pi_j)$$

- **Quantity fee**

$$\pi_i(\boldsymbol{p}, r) = q_i(\boldsymbol{p}, r) \cdot (p_i - c - f) \qquad \pi_p(\boldsymbol{p}, r) = f \cdot (q_i + q_j)$$

Firms compete in prices, $\boldsymbol{p}$.

State and Actions

States

- Firms: $\boldsymbol{s}_{t,i} = (p_{i,t-1}, p_{j,t-1}, r_{t-1})$
  - prices of both firms in the previous period
  - platform ranking in the previous periods
- Platform: $\boldsymbol{s}_{t,p} = (p_{i,t}, p_{j,t})$
  - prices of both firms in the current period

Actions

- Firms: prices from a fixed grid
- Platform: ranking, i.e. $(0, 1)$, $(1, 0)$ or $(1, 1)$

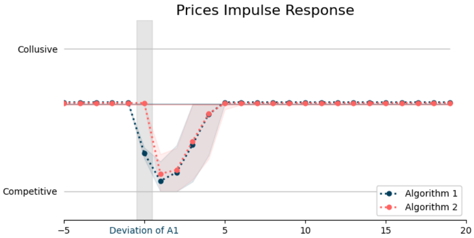
Parametrization

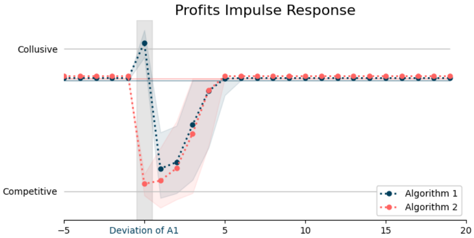
- Same baseline as Calvano et al. (2020)
- Fees are set to give the platform the same revenue, given the equilibrium prices without fees
  - $f^{\text{profit}} = 0.5$
  - $f^{\text{quantity}} \simeq 0.2$ (marginal cost = 1)

  equivalent to $f^{\text{revenue}} \simeq 0.15$, comparable to Apple's, Amazon's, ...

Without fees

Baseline: how do firms behave without fees? Do they collude?

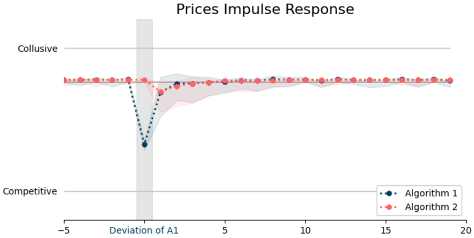


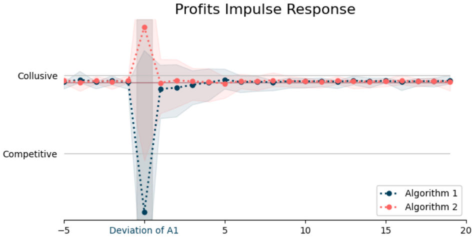


Equilibrium prices are supra-competitive.

- And supported by a reward-punishment scheme

Profit Fee

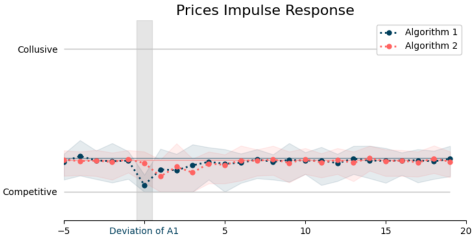


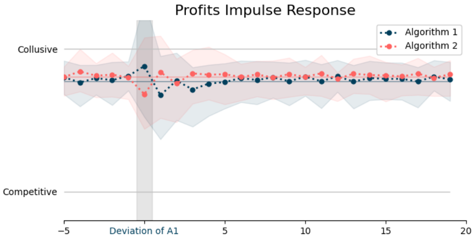


Higher profits and no reward for deviating.

- How? Platform excludes the deviating algorithm
- Very **efficient**
  - No need for further punishment
  - Prices go back to collusive level *faster*

Quantity Fee





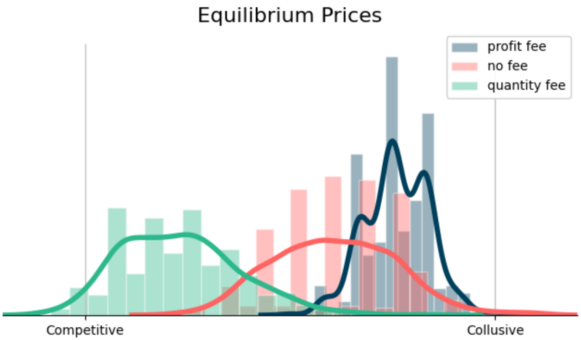
Similar pattern but with **lower profits**.

- Platform punishes deviation keeping firms on low prices

Recap

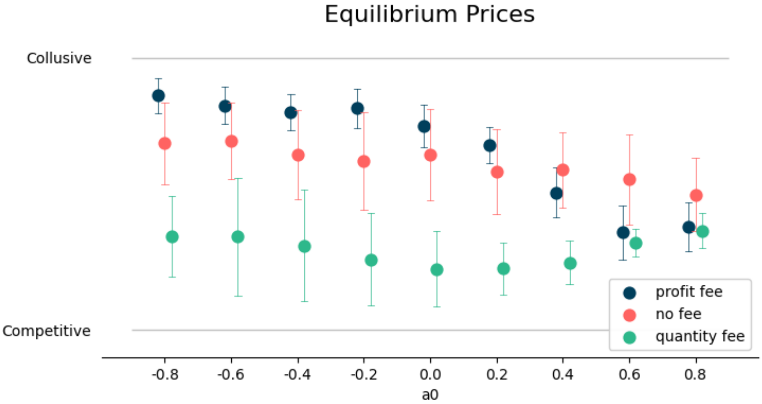
Platform intervention can influence collusion outcomes

- **Profit fee**: increase profitability and effectiveness of collusion
- **Quantity fee**: lowers prices close to competitive outcome



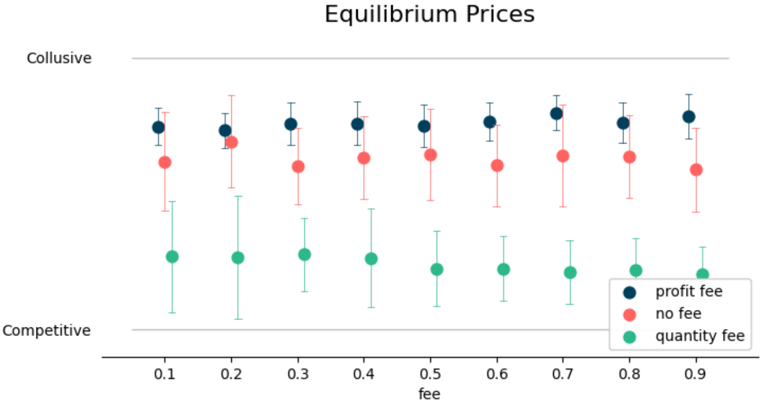
Robustness

- a_0: **price of the outside option**
- f: platform fee
- μ: product differentiation



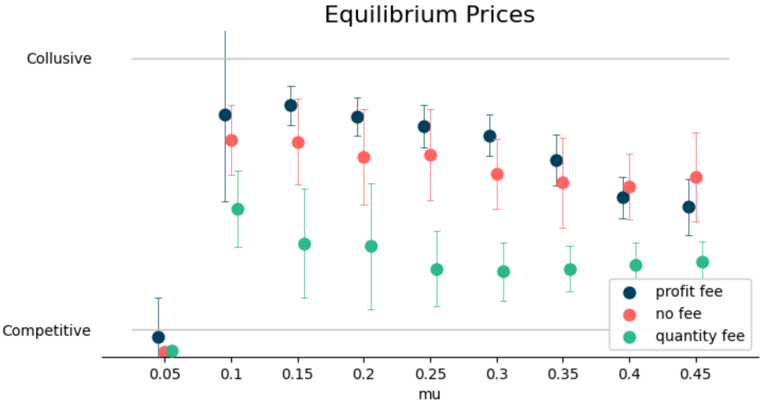
Robustness

- a_0: price of the outside option
- f: **platform fee**
- μ: product differentiation



Robustness

- a_0: price of the outside option
- f: platform fee
- μ: **product differentiation**



Mechanism

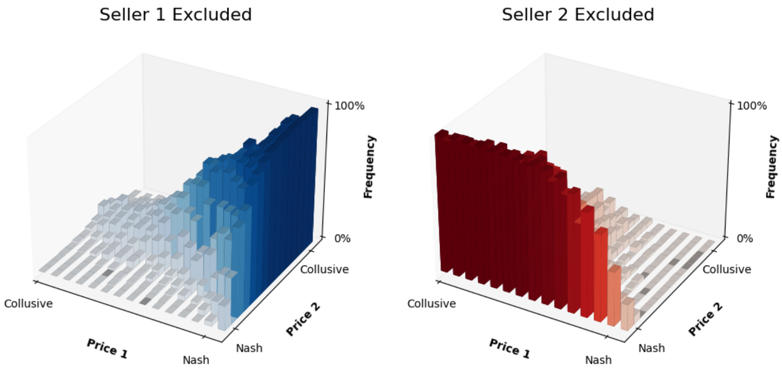
Mechanism

We explore two dimensions in which the platform can influence outcomes:

- In equilibrium
- During learning

Mechanism in equilibrium

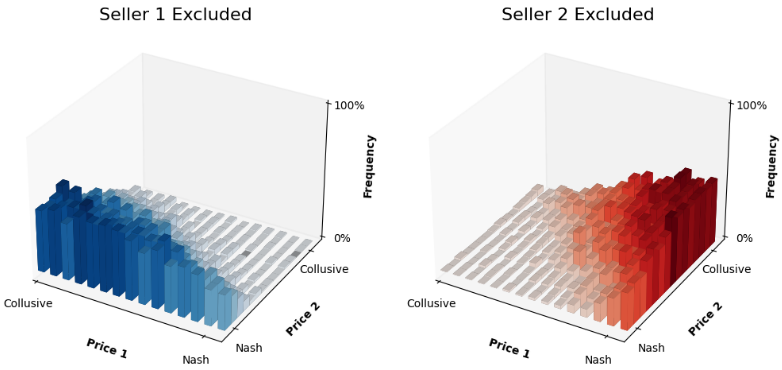
Equilibrium strategy of the platform: frequency of exclusion.



Profit fee: platform punishes firms that undercut collusive prices.

Mechanism in equilibrium

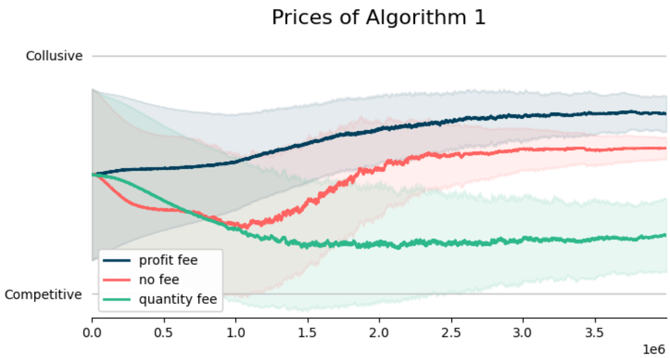
Equilibrium strategy of the platform: frequency of exclusion.



Quantity fee: platform rewards firms that undercut collusive prices.

Mechanism during learning

Median prices during learning (over 4M periods).



Effect of the platform

- Faster learning
- Less noisy for profit fee
- More noisy for quantity fee

Platform Dual Role

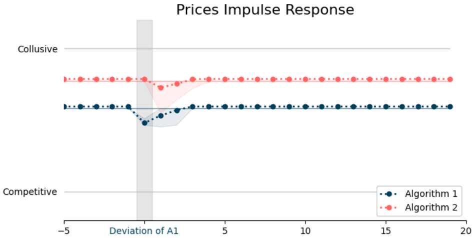
Dual Role - Setting

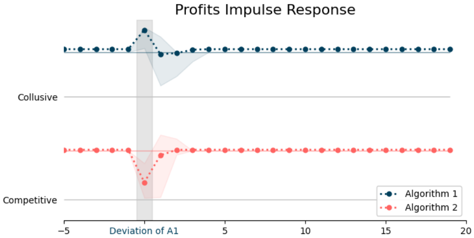
What happens when the platform is also active as a seller?

- Commonly known as **dual role**
- The platform algorithm jointly maximizes marketplace + sales profits
  - Tweak: scale down payoffs from excluding rival so that it is never profitable in the long run
- Outcome: platform "*bullies*" rival into selecting the desired price

Dual Role

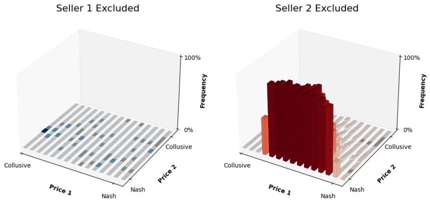
What are the equilibrium prices?



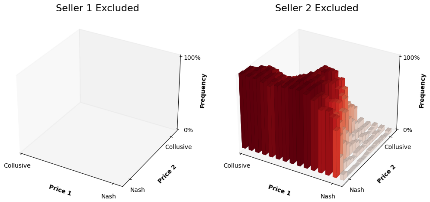


Platform's algorithm (1) undercuts rival (2).

Mechanism



(1) Profit Fee



(2) Quantity Fee

- Exclusion of competitor unless it selects the preferred price
- No exclusion of own product in equilibrium

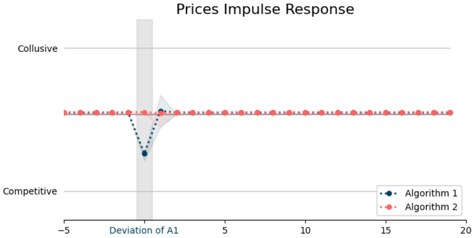
Policy

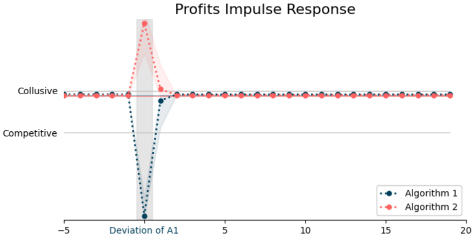
Solution: separate marketplace and sales departments of the platform

- Separate algorithms, i.e. separate objective functions
- The platform's pricing algorithm does not pay the fee
- But the platform acts as if it did

What are the effects? Is it enough to ensure impartial treatment?

Equilibrium - Profit Fee

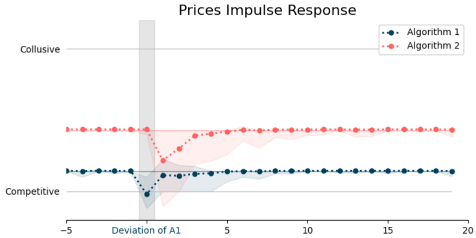


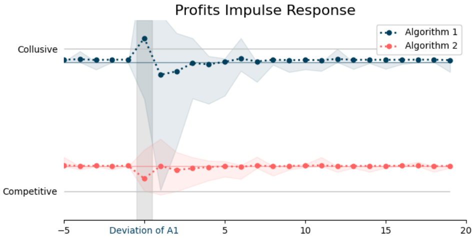


No self-preferencing

- Why? Profit fee is non-distortive

Equilibrium - Quantity Fee





Self-preferencing for algorithm 1

- Why? Quantity fee is distortive
- Platform thinks algorithm 1 is more efficient and gives it preferential treatment

Conclusion

- Having a third algorithm can help preventing algorithmic collusion
  - Must have power to influece payoffs
  - And the *appropriate* objective function
- Self-preferencing can emerge also among algorithms

Conclusion

- Having a third algorithm can help preventing algorithmic collusion
    - Must have power to influece payoffs
    - And the *appropriate* objective function
- Self-preferencing can emerge also among algorithms

Thank you!

Bibliography

Bibliography i

Asker, J., Fershtman, C., & Pakes, A. (2021). *Artificial intelligence and pricing: The impact of algorithm design* (Tech. Rep.). National Bureau of Economic Research.

Assad, S., Clark, R., Ershov, D., & Xu, L. (2020). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market.

Calvano, E., Calzolari, G., Denicolo, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, *110*(10), 3267–97.

Chen, L., Mislove, A., & Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th international conference on world wide web* (pp. 1339–1349).

Johnson, J., Rhodes, A., & Wildenbeest, M. R. (2020). Platform design when sellers use pricing algorithms.

Klein, T. (2019). Autonomous algorithmic collusion: Q-learning under sequential pricing. *Amsterdam Law School Research Paper*(2018-15), 2018–05.