Reinforcement Learning and Collusion by Clemens Possnig (UBC)

Drew Van Kuiken

January 30, 2023

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Competition via Algorithm Has Uncertain Consequences

- Companies increasingly use algorithms to optimize price/quantity decisions
- Algorithms may learn to collude[1] $\Rightarrow$ antitrust concerns
- Important to know how collusion arises

[1] See, e.g., **Assad et al. (2020)**, **Klein (2021)**, and **Calvano et al. (2021)**.

Research Question

What outcomes can we expect when algorithms compete against each other?

Proceed in Three Steps

1. Model of reinforcement learning algorithms playing Cournot quantity competition repeatedly

2. Can algorithms learn static Nash equilibrium?

3. What are the channels under which collusion happens?

Preview of Results

1. Model of reinforcement learning algorithms playing Cournot quantity competition repeatedly
   - Algorithms observe common state variable
   - But don't know payoff function or state transitions
   - Experiment with quantity $\Rightarrow$ estimate value function
   - Long-run behavior characterized by stable rest points of differential equation
2. Can algorithms learn static Nash equilibrium?
3. What are the channels under which collusion happens?

Preview of Results

1. Model of reinforcement learning algorithms playing Cournot quantity competition repeatedly
2. Can algorithms learn static Nash equilibrium?
   - It depends on which state variables are tracked and how states evolve
   - Richer states lead to collusion with higher probability
3. What are the channels under which collusion happens?

Preview of Results

1. Model of reinforcement learning algorithms playing Cournot quantity competition repeatedly
2. Can algorithms learn static Nash equilibrium?
3. What are the channels under which collusion happens?
   - Conditions on payoffs/observables leading to collusive equilibria
   - Simulations to demonstrate theoretical results

Outline of Presentation

1. Brief Overview of Actor-Critic Reinforcement Learning
2. Setting and General Limiting Results
3. Application to Repeated Cournot Game
    - Example of Collusive Equilibrium

1. Brief Overview of Actor-Critic Reinforcement Learning

Single-Agent Characterization of Use Case

- Agent chooses $q \in A$ repeatedly, state variable $s \in S$
- Discount rate $\delta \in (0, 1)$
- Find policy $\rho : S \to A$ maximizing future expected discounted payoffs:

$$W(s_0) = E \sum_t \delta^t u_t$$

- Agent can maximize W by computing the value function:

$$V(s) = \max_{q \in A} \{u(q, s) + \delta E[V(s')|q, s]\}$$

Reinforcement Learning (RL) is useful when information about u and transition probabilities isn't available

Our RL Algorithm: Q-Learning

General rule: RL updating rules move policies towards successful actions and away from bad options

Q-Learning Algorithm:

- Estimates function $Q : S \times A \to \mathcal{R}$, targeting:

$$Q^*(s, q) = u(q, s) + \delta E[\max_{q' \in A} Q^*(s', q')|q, s]$$

- I.e., Q evaluates payoff from playing q in current state s and playing optimally afterwards

Algorithm to Estimate Q^*

- Model-free: works without knowledge of u_t or transition function
- Initialize with Q_0, and algorithm updates as follows:

$$Q_{t+1}(s,q) = \begin{cases} Q_t(s,q) + \beta_t[u_t + \delta \max_{q' \in A} Q_t(s_{t+1}, q') - Q_t(s,q)] & \text{if } s_t = s, q_t = q \\ Q_t(s,q) & \text{otherwise} \end{cases}$$

- Assess value of playing a new action
- β_t (learning rate) is a sequence converging to 0
- Specifies a performance criterion, not a policy

Explore Policy Space via ε-Greedy Sampling

Agents face a trade-off: follow current optimal action or try to find something better?

Enter ε-greedy sampling:

- Fix ε. In each period, take $\arg\max_{q'} Q_t(s_t, q')$ with probability $1 - \varepsilon$
- With probability ε, sample uniformly from A
- Under ε-greedy sampling and for suitable β_t, Q_t converges in probability to Q^* if states form a Markov chain controlled by q_t

Extension to Multiple Agents: Agents Now Play a Game

Assume agents use Actor-Critic Q-learning (ACQ) to update their policy function:

Definition 1

Each algorithm i updates policies ρ_t^i according to:

$$\rho_{t+1}^i(s) \in \rho_t^i(s) + \alpha_t[\arg\max_{q' \in A} Q_t^i(s, q') - \rho_t^i(s) + M_{t+1}^i] \tag{1}$$

where $\alpha_t > 0$ is a sequence converging to 0 and M_{t+1}^i is an i.i.d., zero-mean, bounded variance noise generated to explore policy space.

Want to characterize agents' long-run policy functions

▸ Existence and Uniqueness of Nash equilibria

2. Setting and General Limiting Results

Definitions and Primitives

- n algorithms, compact action space A_i, profile space $A = \times_i A_i$
- Finite state space, $|S| = L$, transition probability function $T : S^2 \times A \to (0,1)$
- **Assumption 1**: For all $\rho \in \overline{A}$, the Markov chain induced by playing ρ is irreducible
- Expected future discounted payoffs $W^i(\rho^i, \rho^{-i}, s_0)$, defined given stationary policy profiles $[\rho^i, \rho^{-i}]$
- Define $B^i_S(\rho^{-i})$ as the optimal policy given profile ρ^{-i}:
$$B^i_S(\rho^{-i}) = \arg\max_{\rho \in \overline{A}_i} W^i(\rho, \rho^{-i}, s_0) \tag{2}$$

Recovering Q^*

Thus, conditional on opponents playing ρ_t^{-i} forever, $Q_t^i(s,q)$ is an estimator of:

$$Q^{i*}(s,q,\rho_t^{-i}) = u(q,s) + \delta E[\max_{q' \in A} Q^{i*}(s',q',\rho_t^{-i})|q,s] \tag{3}$$

Q^* is related to W as:

$$\max_{q' \in A} Q^{i*}(s,q',\rho^{-i}) = \max_{\rho \in \overline{A}_i} W^i(\rho,\rho^{-i},s) \tag{4}$$

Assumption 2: There exists a bounded function $g^i(s,q,\rho^{-i})$ that represents the limiting difference between Q_t and Q^* with probability 1.

Limiting Behavior: Asymptotic Stability

Definition 2

Given some ODE $\dot{\rho} = f(\rho)$, let ρ^* be a rest point of $f(\rho)$. Let $\Lambda = eigv[Df(\rho^*)]$ be the set of eigenvalues of the linearization of f at ρ^*. For a complex number z, let $\mathbf{Re}[z] \in \mathbb{R}$ be the real part. ρ^* is:

- Hyperbolic if $\mathbf{Re}[\lambda] \neq 0$ holds for all $\lambda \in \Lambda$
- Asymptotically stable if $\mathbf{Re}[\lambda] < 0$ holds for all $\lambda \in \Lambda$
- Linearly unstable if $\mathbf{Re}[\lambda] > 0$ holds for at least one $\lambda \in \Lambda$

We can connect the long-run behavior of ρ_t to limiting sets of the solutions to the above ODE.

ACQ, Asymptotic Stability, and Best Response Dynamics

Define $F_B^S(\rho) = \overline{B}_S(\rho) - \rho$ as the state dependent best response dynamics vector field

Proposition 1

Let ρ^* be asymptotically stable for F_B^S. Then for all γ small enough and all $g(s, q, \rho^{-i})$ with bounded derivatives, there is a profile ρ^g such that:

1. $\sup_g |\rho^g - \rho^*| \to 0$ *as* $\gamma \to 0$.
2. *The probability that the limit set of $\rho = \rho^g$ is bounded above 0.*

Basic proof sketch: For every ρ^*, there is a unique rest point ρ^g. The stability of ρ^* carries over to the stability of ρ^g. Full Proof Sketch Limit Set Definition

Asymptotic Instability and ρ in the Limit

Proposition 2

Let ρ^* be linearly unstable for F_B^S. Then for all γ and all $g(s, q, \rho^{-i})$ with bounded derivatives, there is an open neighborhood $\mathcal{U}_\gamma$ with $\rho^* \in \mathcal{U}_\gamma$ such that the probability that the limit set of the algorithm is contained in $\mathcal{U}_\gamma$ equals 0.

Proof sketch:

- Establish 1:1 relationship between stability of ρ^* and rest points ρ^g
- Instability + variance of $M_{t+1} \Rightarrow \rho_t$ will land on unstable manifolds and move away from ρ^g
- Hyperbolicity of $\rho^*, \rho^g \Rightarrow$ there is a neighborhood U around ρ^g with $\rho^* \in U$ such that ρ^g is the only internally chain transitive set within U.

Some Intuition

Asymptotically stable equilibria can be limit points of the RL procedure, but unstable equilibria cannot

- Agents make errors due to estimation and to explore action space $\Rightarrow$ opponent strategy profiles constantly perturbed

- Updating rules track F_B^S, so an agent's policy will only stay close to ρ^* if the dynamics of F_B^S are robust to deviations

3. Application to Repeated Cournot Game

Setup for Cournot Game

- 2 agents $i \in \{1, 2\}$
- Stochastic binary price outcome $Y \in \{P_L, P_H\}$
- Quantity choice $q \in I = [0, M]$, $M > 0$, aggregate quantity Q
- Probability of outcome: $Pr[Y = P_L | Q] = h(Q)$
- Expected Price: $Y(Q) = P_L h(Q) + P_H(1 - h(Q))$
- Cost $c(q)$ is twice differentiable
- Stage game payoffs: $u^i(q_1, q_2) = Y(Q)q_i - c(q_i)$
- Transition probabilities depend on aggregate quantities:
  $P_{sB}(q_1, q_2) = Pr[s' = B | s; q_1 + q_2]$

Payoffs

Given the binary state space, we can parametrize W^i as follows:

$$W^i(\rho, A) = \omega^{-1}[(1 - \delta P_{BB}(\rho))u^i(\rho^i(A), \rho^{-i}(A)) + \delta P_{AB}(\rho)u^i(\rho^i(B), \rho^{-i}(B))]$$
$$W^i(\rho, B) = \omega^{-1}[\delta(1 - P_{BB}(\rho))u^i(\rho^i(A), \rho^{-i}(A)) + (1 - \delta(1 - P_{AB}))(\rho)u^i(\rho^i(B), \rho^{-i}(B))]$$

where

$$\omega = [1 + \delta(P_{AB}(\rho) - P_{BB}(\rho))]$$

Idea: W^i is a convex combination of u^i over two states, weights a function of transition probabilities

Direction Switching Policies

Definition 3

A binary state policy is direction-switching (DS) if the underlying state transitions are irreducible and $P_{AB} = 1 - P_{BB}(Q)$. Denote the state space as S^{DS}.

DS-policy Can Lead to Dynamically Unstable Equilibrium

Proposition 3

Let u satisfy standard assumptions for Cournot competition. Let ζ_N be the DS-policy that plays q_N in every state. Then ζ_N is dynamically unstable (i.e., unstable w.r.t. $F_B^{S^{DS}}$) if

$$-\frac{u_{12}^N}{u_{11}^N} + 2D_N > 1$$

where

$$D_N = \delta \frac{P'_{AB}(Q_N)}{\omega} \frac{\delta u_2^N}{u_{11}^N}$$

Dynamic Vs. Static Stability Using Eigenvalues

Proof sketch:
To prove proposition 3, linearize best responses at ζ_N. This yields:

$$\begin{bmatrix} -\frac{u_{12}^N}{u_{11}^N} + D_N & -D_N \\ -D_N & -\frac{u_{12}^N}{u_{11}^N} + D_N \end{bmatrix}$$

with eigenvalues $\lambda_j \in \{-\frac{u_{12}^N}{u_{11}^N}, -\frac{u_{12}^N}{u_{11}^N} + 2D_N\}$ for $j \in \{1, 2\}$
Thus, if $P'_{AB}(Q_N)$ large enough, static equilibrium is dynamically unstable

Dynamic Vs. Static Stability Using Eigenvalues

Proof sketch:
To prove proposition 3, linearize best responses at ζ_N. This yields:

$$\begin{bmatrix} -\frac{u_{12}^N}{u_{11}^N} + D_N & -D_N \\ -D_N & -\frac{u_{12}^N}{u_{11}^N} + D_N \end{bmatrix}$$

with eigenvalues $\lambda_j \in \{-\frac{u_{12}^N}{u_{11}^N}, -\frac{u_{12}^N}{u_{11}^N} + 2D_N\}$ for $j \in \{1, 2\}$
Thus, if $P'_{AB}(Q_N)$ large enough, static equilibrium is dynamically unstable

$-\frac{u_{12}^N}{u_{11}^N}$: Slope of static best-response
$2D_N$: dynamic incentive when DS-policies are played

Dynamic and Static Stability Coincide Under 1R Policies

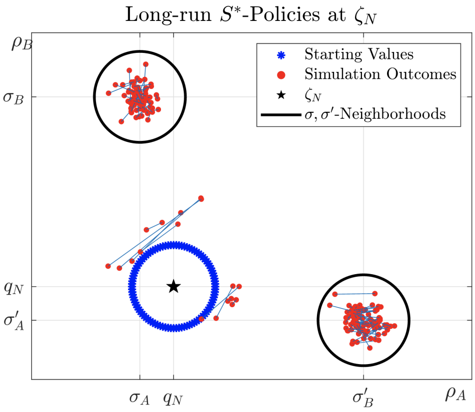
Definition 4

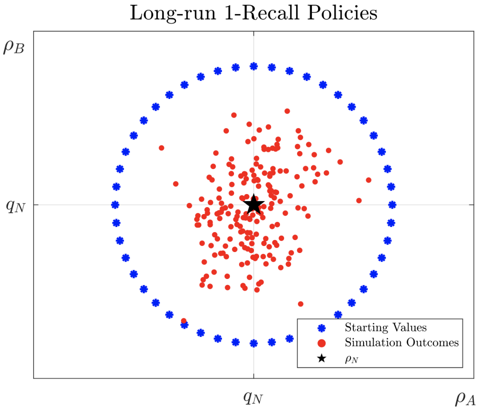
A public 1R-policy can be defined as policy $\rho : \boldsymbol{P} = \{P_L, P_H\} \rightarrow I$ so that states are price realizations representing last period's observed price. This can equivalently be defined as having a state space $\boldsymbol{P}$ with transition function $T(s, P) \in \boldsymbol{P}$ such that $T(s, P) = P$ for all $s \in \boldsymbol{P}$ and all price observations $P \in \boldsymbol{P}$.

Proposition 4

Let ρ_N be the 1R-policy that plays stage game Nash quantity q_N in every state. Then ρ_N is asymptotically stable if and only if q_N is.

Visualizing ζ_N- and 1R-Policy Algorithms





3a. Example of Collusive Equilibrium

Price Signals

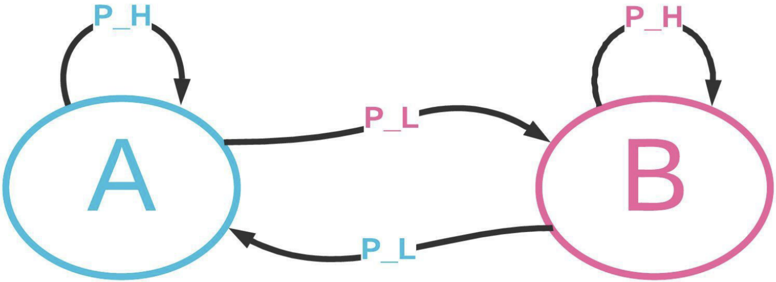


FIGURE 1. State Transition Diagram

P_L signals switch and P_H signals remain

Price Signals Can Support Collusion

Suppose S^{DS} is satisfied:

$$P_{AB}(Q) = Pr[P_L|Q] = h(Q); P_{BB}(Q) = Pr[P_H|Q] = 1 - h(Q)$$

and $h(Q)$ takes an S-shaped form.

Proposition 5

There exists h, $P_H > P_L \geq 0$ and convex $c(q)$ such that resulting u satisfies Cournot assumptions, ζ_N is dynamically unstable, and there exists a symmetric equilibrium σ with $0 < \sigma_A < q_N < \sigma_B$.

Appendix

Existence and Uniqueness: Definition

Define $E_S \subset \overline{A}$ to be set of Nash equilibria in policy profiles:

Definition 5

Nash equilibrium $\rho^* \subset E_S$ is called a 'differential Nash equilibrium' if first order conditions hold for each agent at ρ^* and the Hessian of each agent's optimization problem at ρ^* is negative definite.

Thus, if ρ^* is a differential Nash equilibrium, then there is an open neighborhood around ρ^* such that best responses are single-valued for all ρ that neighborhood.

Existence and Uniqueness: Assumptions

Assumption 1

- *Given state space S, stationary equilibrium profiles $\rho^* \in \overline{A}$ exist. Call the set of such equilibria E_S.*
- *There exist $\rho^* \in E_S$ that are differential Nash equilibria*

For Assumption to hold, we need an interior static Nash equilibrium to exist given $u(r, s)$ for all $s \in S$.

Return

Proposition 1: Proof Sketch

Define:

$$\dot{\rho} \in F_g(\rho(t)) \equiv conv[F_B^S(\rho(t))] + g(\rho(t))$$

- If F_B^S satisfies a linear growth condition, there is a global solution to the differential inclusion F_B^S
- Since α_t converges to 0, the time-interpolated version of ρ_t stays close to the solutions to F_B^S
- can recover limit behavior of ρ_t from limit behavior of F_B^S.
- Thus, the attracting points of the differential system also attract ρ_t over time

Return

Limit Set Definition

Definition 6

Using the ACQ algorithm, the limit set is defined as

$$L_{S,g} = \bigcap_{t \geq 0} \{\overline{\rho_s | s \geq t}\}$$

the set of limits of convergent subsequences ρ_{t_k}.

The limit set depends specifically on the state space S and bias function g. Return