Multiagent Reinforcement Learning

Chi Jin

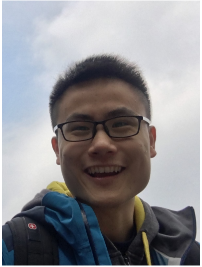
Princeton University.

Slides: on my homepage

Blog post: yubai.org/blog/marl_theory.html

Contributors

Yu Bai
Salesforce

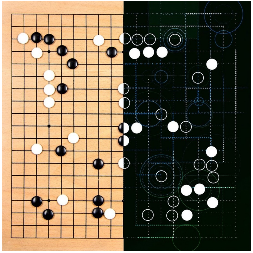


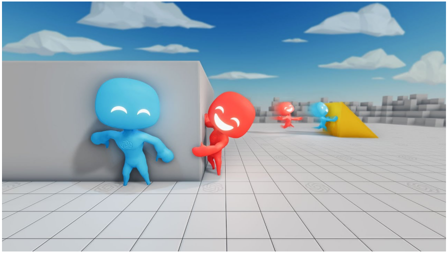
Qinghua Liu
Princeton

Yuanhao Wang
Princeton

Tiancheng Yu
MIT

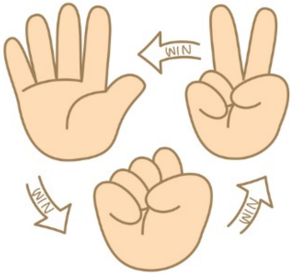
Interesting Problems

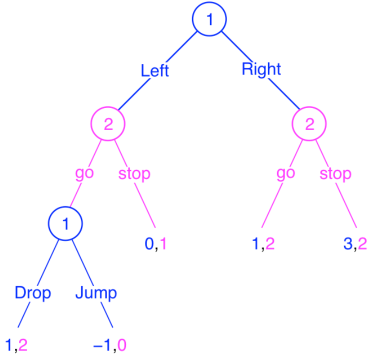




Multiagent **Games** + **Sequential** decision making

Classical Game Theory

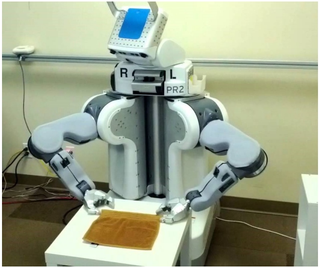


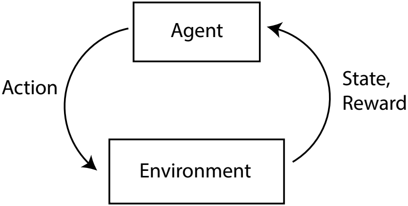


- Normal-form games, Extensive-form games, ...

They don't handle **sequential games** with **long horizon** efficiently.

Single-agent Reinforcement Learning



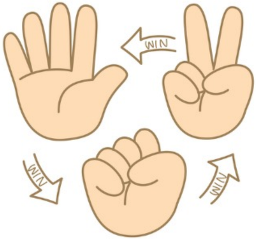


- Goal: find the best policy within a fixed environment.

Opponents in MARL are not fixed, and can be **adaptive**!

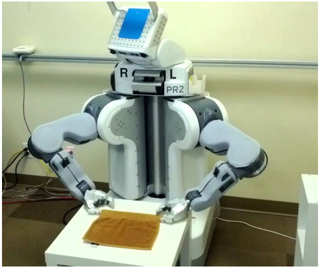
Multiagent Reinforcement Learning

Game theory



+

Reinforcement learning

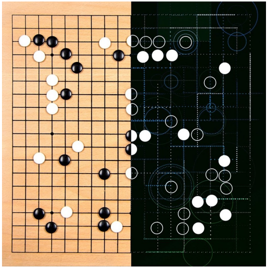


A newer and less developed field, with its own unique challenges and opportunities.

Main Question

Can we establish a solid theoretical foundation for MARL?

Efficiency

Sample efficiency and computational efficiency

AlphaGo Zero: trained on $\geq 10^7$ games, and took ≥ 1 month.

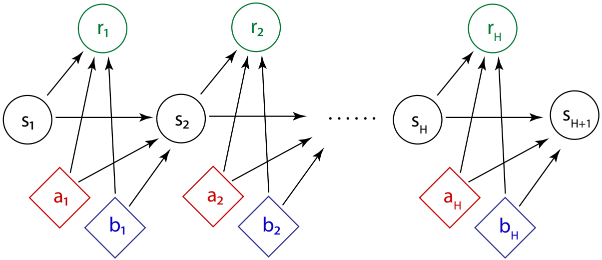
Statistics + Computer Science

Outline

- Formulation and Objectives
- Direct Combinations of Game Theory & Single-agent RL
- Two-player Zero-sum Games
- Multiplayer General-sum Games
- Advanced Topics

Formulation and Objectives

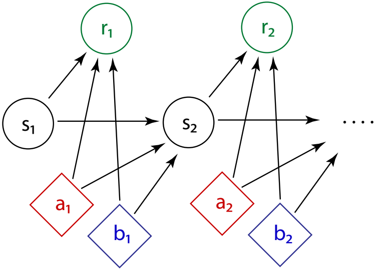
Markov Games (Stochastic Games)



Two-player zero-sum Markov Game $(\mathcal{S}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r, H)$ [Shapley 1953].

- $\mathcal{S}$: set of states; $\mathcal{A}, \mathcal{B}$: set of actions for the max-player/the min-player.
- $\mathbb{P}_h(s_{h+1}|s_h, a_h, b_h)$: transition probability.
- $r_h(s_h, a_h, b_h) \in [0, 1]$: reward for the max-player (loss for the min-player).
- H: horizon/the length of the game.

Interaction Protocol



Interaction protocol

Environment samples initial state s_1.

for step $h = 1, \dots, H$,

two agents take their own actions (a_h, b_h) simultaneously.

both agents receive their own immediate reward $\pm r_h(s_h, a_h, b_h)$.

environment transitions to the next state $s_{h+1} \sim \mathbb{P}_h(\cdot | s_h, a_h, b_h)$.

Our Setup

In this talk, we mostly focus on fully observable tabular Markov games.

- Fully observable: joint actions and states are revealed to both agents.
- Tabular: the size of $\mathcal{S}, \mathcal{A}, \mathcal{B}$ is finite and small.

serve as a **foundation** for more advanced setups in the future

Policy and Value

- General policy for the max-player (depends on the entire history):

$$\pi_{1,h} : (\mathcal{S} \times \mathcal{A} \times \mathcal{B})^{h-1} \times \mathcal{S} \to \Delta_{\mathcal{A}}$$

- Markov policy for the max-player (depends on the current state):

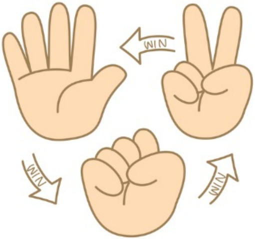
$$\pi_{1,h} : \mathcal{S} \to \Delta_{\mathcal{A}}$$

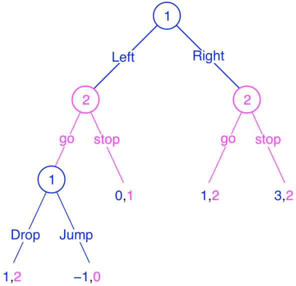
Policy of the min-player can be defined by symmetry.

- Value V^{π} for joint policy $\pi = (\pi_1, \pi_2)$: the expected cumulative reward received by the max-player if both agents follow the joint policy π:

$$V^{\pi} = \mathbb{E}_{\pi}\left[\sum_{h=1}^{H} r_h(s_h, a_h, b_h)\right]$$

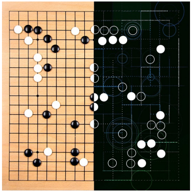
Special Cases





- Normal-form games: no state, no transition.
- Extensive-form games: tree-structured transition.

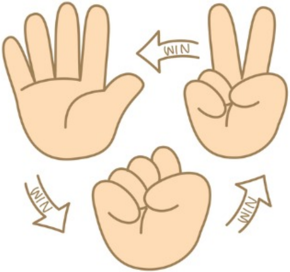
Solution Concepts



What policy is good?

- Beat the world champion by a large margin?
- Beat all players by a large margin?

Best Responses



The policy that best exploits the opponent's policy.

$$\mathrm{BR}(\pi_2) := \operatorname*{argmax}_{\pi_1} V^{\pi_1,\pi_2}$$

Good against a fixed opponent, but can be bad against others.

Nash Equilibria

Nash Equilibria

The policies $(\pi_1^\star, \pi_2^\star)$ is a Nash equilibrium if no player has incentive to deviate from her current policy. That is, for any π_1, π_2

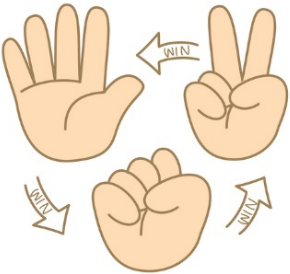
$$V^{\pi_1, \pi_2^\star} \leq V^{\pi_1^\star, \pi_2^\star} \leq V^{\pi_1^\star, \pi_2}$$

In two-player zero-sum Markov games, minimax theorem holds:

$$\max_{\pi_1} \min_{\pi_2} V^{\pi_1, \pi_2} = \min_{\pi_2} \max_{\pi_1} V^{\pi_1, \pi_2}$$

- not due to von Neueman's theorem as V^{π_1, π_2} is not convex-concave.
- can be proved via dynamical programming.

Nash Equilibria II



The optimal strategy if always facing best responses.

"We may not win by a large margin, but no one beats us."

Objective: find ϵ-approximate Nash equilibria $(\hat{\pi}_1, \hat{\pi}_2)$ using a small number of samples with mild dependency on S, A_1, A_2, ϵ, H.

$$\max_{\pi_1} V^{\pi_1, \hat{\pi}_2} - \min_{\pi_2} V^{\hat{\pi}_1, \pi_2} \leq \epsilon.$$

Challenges

To name a few:

- Large size of policy space:

$$\Omega((1/\epsilon)^{HSA}) \text{ Markov policies in the tabular setting}$$

- Nash equilibrium policy is Markov, but the best response may not be.

- MGs do not allow efficient no-regret learning [Bai, **Jin**, Yu, 2020].

$$\max_{\pi_1} \sum_{t=1}^{T} V_1^{\pi_1 \times \pi_2^t} - \sum_{t=1}^{T} V_1^{\pi_1^t \times \pi_2^t} \leq \text{poly}(H, S, A, B) T^{1-\alpha}.$$

Direct Combinations

General Recipe

Key observation: given a fixed opponent, computing best response (BR) is a single-agent RL problem.

Nash finding algorithms with BR oracle

self-play
fictitious play
double oracle
. . .

→

Single-agent RL algorithms

value-iteration
Q-learning
DQN
PPO
. . .

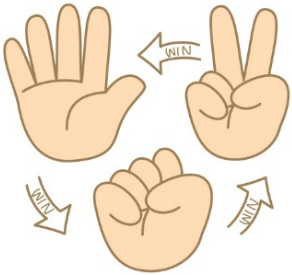
commonly used in practice.

Self-play

Self-play

for $k = 1, \ldots, K$,
$\pi_1^{k+1} = BR(\pi_2^k)$.
$\pi_2^{k+1} = BR(\pi_1^{k+1})$.

π_i^k: the policy of the i^{th} player at the k^{th} iteration



Does not converge to Nash equilibria even in rock-paper-scissor!

Averaging won't help.

Fictitious play

Fictitious play [Brown, 1949]

for $k = 1, \ldots, K$,

$\pi_1^{k+1} = BR[(1/k) \cdot (\pi_2^1 + \ldots + \pi_2^k)]$.

$\pi_2^{k+1} = BR[(1/(k+1)) \cdot (\pi_1^1 + \ldots + \pi_1^{k+1})]$.

π_i^k: the policy of the i^{th} player at the k^{th} iteration

Computing the best response to the average policy of the opponent.

makes more sense in rock-paper-scissor.

Theory of fictitious play

Asymptotic convergence of fictitious play [Robinson 1951]

Ficitious play indeed converges to Nash equilibrium!

However, how **fast**?

- inspecting the proof of [Robinson 1951], it requires $(1/\epsilon)^{\Omega(A)}$ iterations to converge to ϵ-Nash equilibrium for a normal-form game with A actions.
- Karlin conjectured in 1959 that this rate can be improved to $\mathcal{O}(1/\epsilon^2)$.
- Daskalakis and Pan [2014] refute the conjecture, and prove that $(1/\epsilon)^{\Omega(A)}$ is real in the worst case.

Double Oracle

Let $M_k \in \mathbb{R}^{k\times k}$ be the reward matrix of subgame whose row actions are $\{\pi_1^i\}_{i=1}^k$ and column actions are $\{\pi_2^j\}_{j=1}^k$.

$$M_k = \begin{array}{c} \\ \vdots \\ \pi_1^i \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} \cdots & \pi_2^j & \cdots \end{array} \\ \left(\begin{array}{ccc} & \vdots & \\ \ldots & V^{\pi_1^i, \pi_2^j} & \ldots \\ & \vdots & \end{array} \right) \end{array}$$

Double Oracle

for $k = 1, \ldots, K$,

$\boldsymbol{p}, \boldsymbol{q} \leftarrow$ a Nash equilibrium of M_k.

$\pi_1^{k+1} = BR[\sum_{i=1}^k p_i \pi_1^i]$.

$\pi_2^{k+1} = BR[\sum_{j=1}^k q_j \pi_2^j]$.

Theory of Double Oracle

Double oracle represents a class of general approach which uses more informed weights than fictitious play.

Convergence of double oracle [McMahan 2003]

Double oracle algorithm finds Nash equilibrium of a normal-form game with A actions in $\mathcal{O}(A)$ iterations.

- This is because M_A is the full game matrix.
- Directly converting a MG into a norm-form game gives $A = (1/\epsilon)^{HSA'}$ —the size of policy space.

Drawbacks of Direct Combinations

- Algorithms are designed based on black-box usage of single-agent RL, which does not exploit the detailed structure of MGs.

- Converting a MG into a norm-form game gives a number of action $A = (1/\epsilon)^{HSA'}$.

- Finding BR is NOT a easy single-agent RL problem:
  - When the min-player deploys a fixed **non-Markovian** policy, the game is NOT an MDP from the perspective of the max-player.
  - Existing single-agent RL results do not apply.

Two-player Zero-sum Markov Games

Planning

We start with the setting of known transition $\mathbb{P}$ and reward r.

A Nash equilibrium of a MG is a Markov policy.

We define $V_h^\star(s)$, $Q_h^\star(s,a,b)$ which satisfies the **Bellman optimality equation**:

$$
\begin{aligned}
Q_h^\star(s,a,b) =& r_h(s,a,b) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a,b)} V_{h+1}^\star(s') \\
V_h^\star(s) =& \max_{\mu \in \Delta_{\mathcal{A}}} \min_{\nu \in \Delta_{\mathcal{B}}} \sum_{a,b} \mu(a)\nu(b) Q_h^\star(s,a,b) \\
:=& \text{Nash\_Value}(Q_h^\star(s,\cdot,\cdot))
\end{aligned}
$$

Nash Value Iteration

A dynamical programming approach to find a Nash equilibrium.

Nash Value Iteration (Nash VI)

Initialize $V^\star_{H+1}(s) = 0$ for all s.

for $h = H, \ldots, 1$,

  for all (s, a, b),

    $Q^\star_h(s, a, b) \leftarrow r_h(s, a, b) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a,b)} V^\star_{h+1}(s')$

  for all s

    $(\pi^\star_{1,h}(\cdot|s), \pi^\star_{2,h}(\cdot|s)) \leftarrow \mathrm{Nash}(Q^\star_h(s, \cdot, \cdot))$

    $V^\star_h(s) \leftarrow \langle \pi^\star_{1,h}(\cdot|s) \times \pi^\star_{2,h}(\cdot|s), Q^\star_h(s, \cdot, \cdot) \rangle$

Nash VI computes the Nash equilibrium of MGs in poly(H, S, A, B) steps!

More about Planning and Simulator Setting

Known $\mathbb{P}, r$:

- Nash Q-learning also finds Nash equilibrium. [Hu & Wellman 2003]
- ...

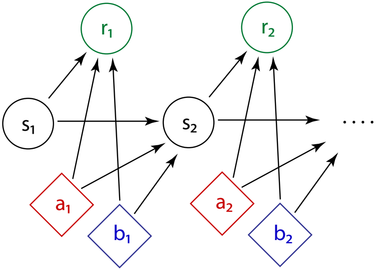
Simulator setting (query any s, a, b, receive reward r and next state s'):

- query all (s, a, b) uniformly and use sample average to estimate $\mathbb{P}$ and r.
- variants of Nash-VI [Zhang et al. 2020]
- variants of Nash Q-learning [Sidford et al. 2019]
- ...

Practical setting (agent can't choose state s):

- need to tradeoff exploration vs. exploitation.
- **will be our focus next**.

Interaction Protocol



Interaction protocol

Environment samples initial state s_1.

for step $h = 1, \dots, H$,

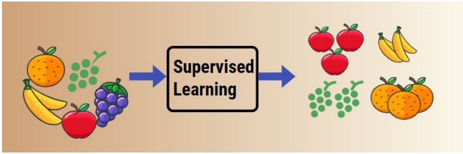
two agents take their own actions (a_h, b_h) simultaneously.

both agents receive their own immediate reward $\pm r_h(s_h, a_h, b_h)$.

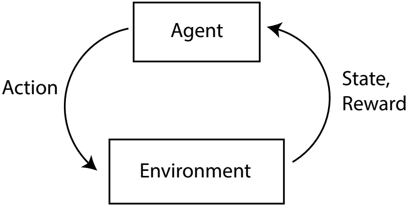
environment transitions to the next state $s_{h+1} \sim \mathbb{P}_h(\cdot|s_h, a_h, b_h)$.

Collecting Samples

Supervised learning: samples are given at the beginning.



RL: agent picks actions/policies to collect samples during training.

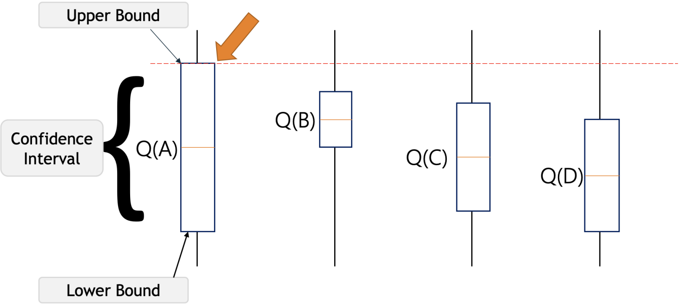


Exploration

$$\epsilon\text{-}\textbf{greedy}\text{: take } \begin{cases} \text{random action,} & \text{with probability } \epsilon \\ \text{greedy action,} & \text{otherwise} \end{cases}$$

needs exponential number of samples in the worst case!

Upper Confidence Bound (UCB)



UCB Algorithm: be optimistic! Pick the action with the largest upper bound on the confidence interval.

Optimistic Nash-VI

Optimistic Nash VI [Liu, Yu, Bai, Jin, 2020]

for $k = 1, \ldots, K$,

  for $h = H, \ldots, 1$,

    for all (s, a, b),

      $\overline{Q}_h(s, a, b) \leftarrow r_h(s, a, b) + \mathbb{E}_{s' \sim \hat{\mathbb{P}}_h(\cdot|s,a,b)} \overline{V}_{h+1}(s') + \beta$

      $\underline{Q}_h(s, a, b) \leftarrow r_h(s, a, b) + \mathbb{E}_{s' \sim \hat{\mathbb{P}}_h(\cdot|s,a,b)} \underline{V}_{h+1}(s') - \beta$

    for all s

      $\pi_h(\cdot, \cdot|s) \leftarrow \mathrm{CCE}(\overline{Q}_h(s, \cdot, \cdot), \underline{Q}_h(s, \cdot, \cdot))$

      $\overline{V}_h(s) \leftarrow \langle \pi_h(\cdot, \cdot|s), \overline{Q}_h(s, \cdot, \cdot) \rangle$

      $\underline{V}_h(s) \leftarrow \langle \pi_h(\cdot, \cdot|s), \underline{Q}_h(s, \cdot, \cdot) \rangle$

  execute policy π, collect samples, and update estimation $\hat{\mathbb{P}}$.

$$\hat{\mathbb{P}}_h(s'|s, a, b) = \frac{N(s, a, b, s')}{N(s, a, b)}$$

can be viewed as a multiagent version of UCB-VI algorithm [Azar et al. 2017].

Main techniques

- Use sample average $\hat{\mathbb{P}}$ to estimate transition.
- Maintain upper and lower bound $\overline{Q}$ and $\underline{Q}$ to be optimistic.
  - The choice of bonus β is different from single-agent RL for sharp guarantee.
- Compute coarse correlated equilibrium (CCE) of $(\overline{Q}, \underline{Q})$ instead of Nash. [Xie et al. 2020]
  - computing Nash equilibria of general-sum games is PPAD-hard. [Daskalakis et al. 2008]

Theory of Optimistic Nash VI

Theorem [Liu, Yu, Bai, Jin 2020]

With high probability, optimistic Nash VI finds an ϵ-Nash equilibrium in $\tilde{\mathcal{O}}(H^3SAB/\epsilon^2)$ episodes.

H: horizon; S: number of states; A, B: number of actions for each player.

Optimistic Nash VI finds ϵ-Nash in polynomial time and samples!

Information theoretical lower bound: $\Omega(H^3S\max\{A, B\}/\epsilon^2)$

Unique Challenge I: Centralized vs. Decentralized Algorithms

Optimistic Nash VI is a **centralized** algorithm

- at each step, centralized solver finds CCE of

$$\overline{Q}_h(s, \cdot, \cdot), \underline{Q}_h(s, \cdot, \cdot)$$

Decentralized algorithms: each agent runs the same algorithm using her own observations as if in the single-agent setting.

- easier to implement.
- more versatile, agnostic to the actions of other agents.
- faster, less communication.

Unique Challenge II: Bypassing the estimation of Q-value

- Most single-agent RL algorithm relies on estimating $Q^\star$.
- In MGs, $Q^\star$ has $\Omega(HAB)$ entries, which requires at least $\Omega(HAB)$ samples to estimate.
- We need new mechanism to match the lower bound $\Omega(H^3 S \max\{A, B\}/\epsilon^2)$

Can we design decentralized MARL algorithms that achieves $O(\max\{A, B\})$ sample complexity?

Simple Case: Normal-form Games

Yes! but in a much simplier setting.

Each agent runs no-regret algorithm for adversarial bandit (e.g. EXP3) independently.

$$\sum_{t=1}^{T} \langle \mu_t, \ell_t \rangle - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \langle a, \ell_t \rangle \leq \mathrm{poly}(A) T^{1-\alpha}.$$

- two-player zero-sum games: $(\mathbb{E}_{t \sim \mathrm{Unif}(T)} \mu_t^{(1)}) \times (\mathbb{E}_{t \sim \mathrm{Unif}(T)} \mu_t^{(2)}) \rightarrow$ Nash.
- sample complexity scales with $\tilde{\mathcal{O}}(A + B)$.

Extension to Markov Games?

Why not just run no-regret algorithms for MGs?

$$\max_{\pi_1} \sum_{t=1}^{T} V_1^{\pi_1 \times \pi_2^t} - \sum_{t=1}^{T} V_1^{\pi_1^t \times \pi_2^t} \leq \mathrm{poly}(H, S, A, B) T^{1-\alpha}.$$

WE CANNOT! MGs do not allow efficient no-regret learning.

- Computational hardness [Bai, **Jin**, Yu, 2020]:
  The existence of polynomial time no-regret algorithm for MGs implies the existence of polynomial time algorithm for learning party with noise.

- Statistical hardness [Liu, Wang **Jin**, 2022]:
  No regret learning in MGs is at least as hard as learning the best Markov policy in partial observable MDPs.

V-learning

V-learning [Bai, Jin, Yu, 2020] [Jin, Liu, Wang, Yu, 2021]

for $k = 1, \ldots, K$, receive s_1,

for step $h = 1, \ldots, H$,

take action $a_h \sim \pi_h(\cdot|s_h)$, observe reward r_h and next state s_{h+1}.

$t = N_h(s_h) \leftarrow N_h(s_h) + 1$.

$V_h(s_h) \leftarrow (1 - \alpha_t)V_h(s_h) + \alpha_t(r_h + V_{h+1}(s_{h+1}) + \beta_t)$.

$\pi_h(\cdot|s_h) \leftarrow$ Adv_Bandit_Update$(a_h, r_h + V_{h+1}(s_{h+1}))$

on the $(s_h, h)^{\text{th}}$ adversarial bandit.

- Incremental updates of V instead of Q!
- Learning rate $\alpha_t = (H+1)/(H+t)$ same as Q-learning.

Properties of V-learning

- Is a single-agent algorithm.
- Use adversarial bandit algorithms (with weighted regret guarantee) as black-box.

$$\sum_{t=1}^{T} \alpha_T^t \langle \mu_t, \ell_t \rangle - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \alpha_T^t \langle a, \ell_t \rangle \leq \text{poly}(A) T^{1-\alpha}.$$

- Has no regret guarantee for each state with feeded loss.
- is NOT a no-regret algorithm for Markov games.

Guarantees

- Multiagent setting: both agents run V-learning independently.
- Adversarial bandit subroutine: FTRL.

Theorem [Bai, Jin, Yu, 2020]

In two-player zero-sum Markov games, V-learning with FTRL finds ϵ-Nash in $\tilde{O}(H^5 S \max\{A, B\}/\epsilon^2)$ episodes.

V-learning is a decentralized algorithm that achieves optimal $O(\max\{A, B\})$ sample complexity!

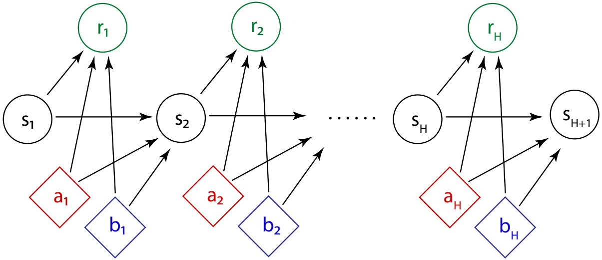
Sharp H dependency waits for future work.

Summary of Algorithms

Algorithm	Training	Main estimand	Sample complexity
Nash-VI	centralized	$\mathbb{P}_h(s'\|s, a, b)$	$\tilde{\mathcal{O}}(H^3 SAB/\epsilon^2)$
Nash Q-Learning	centralized	$Q_h^\star(s, a, b)$	$\tilde{\mathcal{O}}(H^5 SAB/\epsilon^2)$
V-Learning	decentralized	$V_h^\star(s)$	$\tilde{\mathcal{O}}(H^5 S \max\{A, B\}/\epsilon^2)$
Lower bound	-	-	$\Omega(H^3 S \max\{A, B\}/\epsilon^2)$

Multiplayer General-Sum Markov Games

General-Sum Markov Games



Markov Game $(\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, \mathbb{P}, \{r_i\}_{i=1}^m, H)$ [Shapley 1953].

- $\mathcal{S}$: set of states; $\mathcal{A}_i$: set of actions for the i^{th} player.
  let $\boldsymbol{a}_h = (a_h^{(1)}, \ldots, a_h^{(m)})$ be the joint action of all players at step h.
- $\mathbb{P}_h(s_{h+1}|s_h, \boldsymbol{a}_h)$: transition probability.
- $r_{i,h}(s_h, \boldsymbol{a}_h) \in [0, 1]$: reward for the i^{th} player.
- H: horizon/the length of the game.

Policy and Value

- General policy for the i^{th} player (depends on the entire history):

$$\pi_{i,h} : (\mathcal{S} \times (\otimes_{i=1}^{m} \mathcal{A}_i))^{h-1} \times \mathcal{S} \to \Delta_{\mathcal{A}_i}$$

- Markov policy for the i^{th} player (depends on the current state):

$$\pi_{i,h} : \mathcal{S} \to \Delta_{\mathcal{A}_i}$$

- Value V_i^{π} for joint policy $\pi = (\pi_1, \ldots, \pi_m)$: the expected cumulative reward received by the i^{th} player if all agents follow the joint policy π:

$$V_i^{\pi} = \mathbb{E}_{\pi}\left[\sum_{h=1}^{H} r_{i,h}(s_h, \boldsymbol{a}_h)\right]$$

General-sum Nash Equilibria

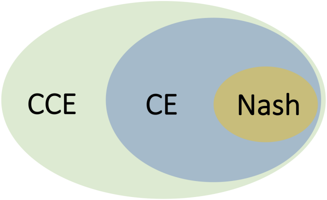
Nash Equilibria

The product policies $\pi^\star = (\pi_1^\star \times \ldots \times \pi_m^\star)$ is a Nash equilibrium if no player has incentive to deviate from her current policy. That is, for any π and any $i \in [m]$ we have

$$V_i^{\pi_i \times \pi_{-i}^\star} \leq V_i^{\pi_i^\star \times \pi_{-i}^\star}$$

Even in the special case of normal-form games, computing Nash equilibria of general-sum games is PPAD-hard. [Daskalakis et al. 2008]

Other Equilibria



- **Correlated equilibrium** (CE): a *correlated* policy π, where no player can gain by deviating from her own policy if she can still observe her sampled actions from the correlated policy.
- **Coarse correlated equilibirum** (CCE): a *correlated* policy π, where no player can gain by deviating ... if she can not observe ...

- Nash $\subset$ CE $\subset$ CCE hold true in both normal-form games and MGs.
- CEs and CCEs can be solved by linear programming.

Optimistic Nash-VI (zero-sum)

Recall:

Optimistic Nash VI [Liu, Yu, Bai, Jin, 2020]

for $k = 1, \ldots, K$,

  for $h = H, \ldots, 1$,

    for all (s, a, b),

      $\overline{Q}_h(s, a, b) \leftarrow r_h(s, a, b) + \mathbb{E}_{s' \sim \hat{\mathbb{P}}_h(\cdot|s,a,b)} \overline{V}_{h+1}(s') + \beta$

      $\underline{Q}_h(s, a, b) \leftarrow r_h(s, a, b) + \mathbb{E}_{s' \sim \hat{\mathbb{P}}_h(\cdot|s,a,b)} \underline{V}_{h+1}(s') - \beta$

    for all s

      $\pi_h(\cdot, \cdot|s) \leftarrow \mathrm{CCE}(\overline{Q}_h(s, \cdot, \cdot), \underline{Q}_h(s, \cdot, \cdot))$

      $\overline{V}_h(s) \leftarrow \langle \pi_h(\cdot, \cdot|s), \overline{Q}_h(s, \cdot, \cdot) \rangle$

      $\underline{V}_h(s) \leftarrow \langle \pi_h(\cdot, \cdot|s), \underline{Q}_h(s, \cdot, \cdot) \rangle$

  execute policy π, collect samples, and update estimation $\hat{\mathbb{P}}$.

Optimistic Nash VI (general-sum)

- Maintain an upper bound $\overline{Q}_{i,h}(s,\cdot)$.
- CCE subroutine changed to (Equilibrium = Nash or CE or CCE)

$$\pi_h(\cdot|s) \leftarrow \text{Equilibrium}(\overline{Q}_{1,h}(s,\cdot), \ldots, \overline{Q}_{m,h}(s,\cdot))$$

Theorem [Liu, Yu, Bai, Jin 2020]

With high probability, optimistic Nash VI finds an ϵ-{Nash, CE, CCE} of a general-sum MG in $\tilde{\mathcal{O}}(H^4 S \prod_{i=1}^{m} A_i/\epsilon^2)$ episodes.

H: horizon; S: number of states; A_i: number of actions for the i^{th} player.

Unique Challenge: Curse of Multiagents

The sample complexity scales with $\Omega(\prod_{i=1}^m A_i) \approx \Omega(A^m)$.

—the size of joint action space.

- grows exponentially w.r.t. number of agents m.
- the size of Q-table $Q(s, \boldsymbol{a})$: $\Omega(S \prod_{i=1}^m A_i)$.

Can we achieve poly(m) sample complexity?

Simple Case: Normal-form Games

Each agent runs no-regret algorithm for adversarial bandit independently.

$$\sum_{t=1}^{T} \langle \mu_t, \ell_t \rangle - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \langle a, \ell_t \rangle \le \mathrm{poly}(A) T^{1-\alpha}.$$

- $\mathbb{E}_{t \sim \mathrm{Unif}(T)} (\mu_t^{(1)} \times \ldots \times \mu_t^{(m)}) \to$ CCE.
- sample complexity scales with $\tilde{\mathcal{O}}(\max_{i \in [m]} A_i)$.

Each agent runs no-swap-regret algorithm for adversarial bandit independently.

$$\sum_{t=1}^{T} \langle \mu_t, \ell_t \rangle - \min_{\psi \in \Psi} \sum_{t=1}^{T} \langle \psi \diamond \mu_t, \ell_t \rangle \le \mathrm{poly}(A) T^{1-\alpha}.$$

$\Psi = \{f : \mathcal{A} \to \mathcal{A}\}$ all possible swap of actions.

- $\mathbb{E}_{t \sim \mathrm{Unif}(T)} (\mu_t^{(1)} \times \ldots \times \mu_t^{(m)}) \to$ CE.
- sample complexity scales with $\tilde{\mathcal{O}}((\max_{i \in [m]} A_i)^2)$.

V-learning

Not a no-regret algorithm for MGs, but enjoys similar properties.

Theorem (CCE & CE) [Song et al. 2021][Jin, Liu, Wang, Yu, 2021]

In general-sum Markov games,
(1) V-learning with FTRL finds ϵ-CCE in $\tilde{\mathcal{O}}(H^5 S(\max_{i\in[m]} A_i)/\epsilon^2)$ episodes;
(2) V-learning with FTRL_swap finds ϵ-CE in $\tilde{\mathcal{O}}(H^5 S(\max_{i\in[m]} A_i)^2/\epsilon^2)$ episodes.

*Mao & Basar [2021] achieves similar results for CCE with slightly worse rate.

V-learning is a decentralized alg that breaks the curse of multiagents!

Summary of the Results

Sample complexity of V-learning for learning MGs.

Objective	Multi-player general-sum	
	Two-player zero-sum	-
Nash	$\tilde{\mathcal{O}}(H^5 SA/\epsilon^2)$	PPAD-complete
CCE	$\tilde{\mathcal{O}}(H^5 SA/\epsilon^2)$	
CE	$\tilde{\mathcal{O}}(H^5 SA^2/\epsilon^2)$	

where $A = \max_{i\in[m]} A_i$.

Advanced Topics

Challenge: Large State Space

Classical RL: Tabular Case

The numbers of states & actions are finite and small.

Strategy: visit all "reachable" states, and learn directly.

Many existing theoretical results.

Challenge: Large State Space II

Modern RL:
Function Approximation

The number of states in practice is typically $\geq 10^{100}$.

Most states are not visited even once.

Strategy: approximate "value" or "policy" by functions in a parameteric class $\mathcal{F}$ (such as deep nets).

Objective: sample complexity depends on complexity of $\mathcal{F}$ instead of S.

Linear MGs

Linear MGs:

$$\mathbb{P}_h(s'|s,a,b) = \langle \phi(s,a,b), \mu_h(s') \rangle,$$
$$r_h(s,a,b) = \langle \phi(s,a,b), \theta_h \rangle,$$

Theorem (linear MGs) [Xie et al. 2020]

For zero-sum linear MGs with ambient dimension d, there exists an algorithm that learns an ϵ-Nash within $\tilde{\mathcal{O}}(d^3 H^4/\epsilon^2)$ episodes.

Algorithm combines Optimistic Nash VI with least-squares.

General Function Approximation

Theorem (general function approximation) [Jin, Liu, Yu, 2021]

For zero-sum MGs equipped with a Q-function class $\mathcal{F}$ whose multiagent Bellman Eluder dimension is $\tilde{d}$, *GOLF_with_Exploiter* learns an ϵ-Nash within $\tilde{\mathcal{O}}(H^2\tilde{d}\log(|\mathcal{F}|)/\epsilon^2)$ episodes.

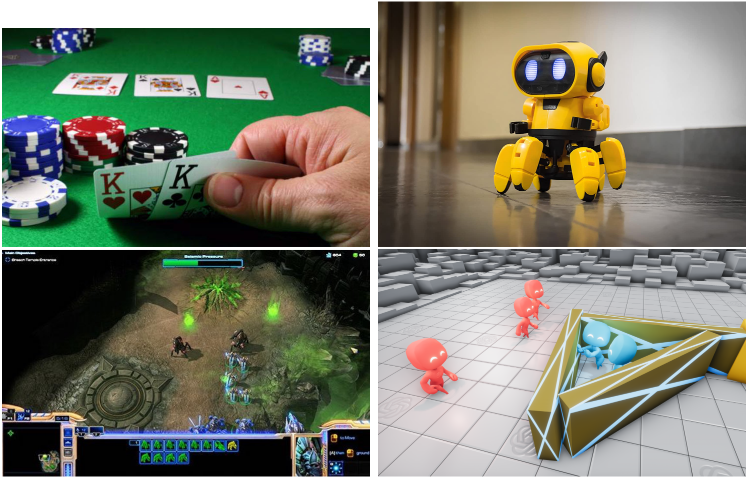
Exploiter style of exploration:

- Main agent: play optimistic Nash policy.
- Exploiter: play optimistic best response to the main agent.

Applies to a rich class of models including tabular MGs, MGs with linear or kernel function approximation, and MGs with rich observations.

Computationally inefficient.

Partial Observability



Common in the real world.

Require agents to maintain memories, and infer based on the entire history.

Imperfect Information Extensive-form game

Algorithm	OMD	CFR	Sample Complexity
Farina and Sandholm [2021]		✓	$\tilde{\mathcal{O}}(\mathrm{poly}(X, Y, A, B)/\varepsilon^4)$
Farina et al. [2021]	✓		$\widetilde{\mathcal{O}}\left(\left(X^4A^3 + Y^4B^3\right)/\varepsilon^2\right)$
Kozuno et al. [2021]	✓		$\widetilde{\mathcal{O}}\left(\left(X^2A + Y^2B\right)/\varepsilon^2\right)$
[Bai, **Jin**, Mei, Yu, 2022]	✓	✓	$\widetilde{\mathcal{O}}\left(\left(XA + YB\right)/\varepsilon^2\right)$
Lower bound	-	-	$\Omega\left(\left(XA + YB\right)/\varepsilon^2\right)$

X, Y are number of info sets for each player.

General POMG

POMDP/POMG is hard if observation contains no information about states.

Theorem [Liu, Szepesvari, Jin, 2022]

For general POMGs where observation contains proper infomation about the states, there exists an algorithm that learns the ϵ-NE of POMG in a polynomial number of samples.

Other Topics

- Further design and analysis of decentralized algorithms.
- Policy optimization algorithms for Markov Games.
- Other notions of equilibria (e.g. Stackelberg equilibria).
- Markov potential games.
- ...

Conclusion

Road Map

- Formulation and Objectives
- Direct Combinations of Game Theory & Single-agent RL
- Tabular Markov Games (Zero-sum & General-sum)
  - Optimistic Nash VI
  - V-learning
- Advanced Topics
  - Function approximation
  - Partial observability
  - Other topics
  - ...

Thank you!