Bai et al. *Near-optimal reinforcement learning with self-play*

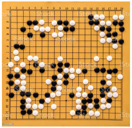
Bohan Li

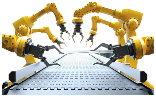
Department of Computer Science
University of Arizona

October 28, 2021

Multi-agent RL is the setting where multiple agents make sequential decisions in an interactive environment. Applications exist in:

- Strategy games
- Robotics systems, AVs
- Social scenarios

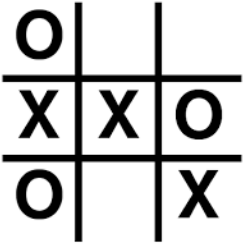


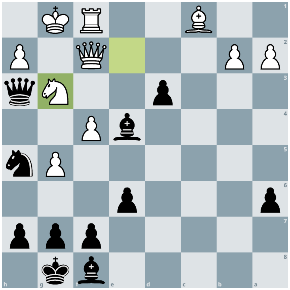



Zero-sum Markov Games (MGs) generalize standard MDP to two player setting, where a max-player μ attempts to maximize the total return and a min-player ν seeks to minimize it. Each game is denoted $MG(H, \mathcal{S}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$.

- H the number of steps in an episode
- $\mathcal{S}$ the set of states, with $|\mathcal{S}| = S$
- $(\mathcal{A}, \mathcal{B})$ the set of actions taken by the max-player and min-player, respectively
- $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$, $\mathbb{P}_h(\cdot|s, a, b)$ is the set of transition matrices
- $r = \{r_h\}_{h\in[H]}$, $r_h : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to [0, 1]$ is the reward function

Examples





Mathematical Definition

Algorithm 1 Markov Game

Given: starting state s_1, max-player policy μ, min-player policy ν
for step $h = 1$ to H **do**
    Max-player takes action $a_h \sim \mu_h(\cdot|s_h)$, min-player takes action $b_h \sim \nu_h(\cdot|s_h)$.
    Both players obtain reward $r_h(s_h, a_h, b_h)$.
    Observe next state $s_{h+1} \sim \mathbb{P}(\cdot|s_h, a_h, b_h)$.
end for

Value Functions

Given the policy of the max-player μ selecting from actions $a \in \mathcal{A}$ and min-player ν selecting actions $b \in \mathcal{B}$, we define the value functions $V_h^{\mu,\nu} : \mathcal{S} \to \mathbb{R}$ and $Q_h^{\mu,\nu} : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$:

$$V_h^{\mu,\nu}(s) \equiv \mathbb{E}_{\mu,\nu}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}, b_{h'}) \middle| s_h = s\right]$$

$$Q_h^{\mu,\nu}(s, a, b) \equiv \mathbb{E}_{\mu,\nu}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}, b_{h'}) \middle| s_h = s, a_h = a, b_h = b\right]$$

Best Response

For any Markov policy of the max-player μ, there exists a **best response** min-player with policy $\nu^\dagger(\mu)$ satisfying

$$\forall (s,h) V_h^{\mu,\nu^\dagger(\mu)}(s) = \inf_\nu V_h^{\mu,\nu}(s)$$

We can define the notion of the best-response max-player with policy $\mu^\dagger(\nu)$ and value function $V_h^{\mu^\dagger(\nu),\nu}(s)$.

- We abbreviate $V_h^{\mu^\dagger(\nu),\nu} \equiv V_h^{\dagger,\nu}$ and $V_h^{\mu,\nu^\dagger(\mu)} \equiv V_h^{\mu,\dagger}$

Nash Equilibrium

In *Competitive Markov Decision Processes*, Filar et al. show that for each player there exists optimal policies against the best responses of their opponents [FV96]. In other words, there exist optimal policies μ^*, ν^* such that:

$$\forall (s,h),\ V_h^{\mu^*,\dagger}(s) = \sup_{\mu} V_h^{\mu,\dagger}(s),\ V_h^{\dagger,\nu^*}(s) = \inf_{\nu} V_h^{\dagger,\nu}(s)$$

Here, the pair μ^*, ν^* is called the Nash equilibrium of the Markov game. It is easy to see that the Nash equilibrium satisfies

$$\sup_{\mu} \inf_{\nu} V_h^{\mu,\nu}(s) = V_h^{\mu^*,\nu^*}(s) = \inf_{\nu} \sup_{\mu} V_h^{\mu,\nu}(s) \tag{1}$$

Abbreviation: $V_h^{\mu^*,\nu^*} \equiv V_h^*$, and similarly $Q_h^{\mu^*,\nu^*} \equiv Q_h^*$

- Objective 1: find an ϵ-approximate best response

Given a fixed opponent policy ν, we would like to find a policy $\hat{\mu}$ such that

$$V_1^{\dagger,\nu}(s_1) - V_1^{\hat{\mu},\nu}(s_1) \leq \epsilon$$

- Objective 2: find a Nash equilibrium of the Markov games where the suboptimality of a pair of policies $\hat{\mu}, \hat{\nu}$ is measured as

$$V_1^{\dagger,\hat{\nu}}(s_1) - V_1^{\hat{\mu},\dagger}(s_1) = \left[V_1^{\dagger,\hat{\nu}}(s_1) - V_1^{*}(s_1)\right] + \left[V_1^{*}(s_1) - V_1^{\hat{\mu},\dagger}(s_1)\right]$$

Furthermore, we define $\hat{\mu}, \hat{\nu}$ to be an ϵ-approximate Nash equilibrium if

$$V_1^{\dagger,\hat{\nu}}(s_1) - V_1^{\hat{\mu},\dagger}(s_1) \leq \epsilon$$

Bellman Equations for Markov Games

- Fixed policies μ, ν:

$$Q_h^{\mu,\nu}(s,a,b) = (r_h + \mathbb{P}_h V_{h+1}^{\mu,\nu})(s,a,b),$$

$$V_h^{\mu,\nu}(s) = (\mathbb{D}_{\mu_h \times \nu_h} Q_h^{\mu,\nu})(s)$$

- Best response for policy of the max-player μ:

$$Q_h^{\mu,\dagger}(s,a,b) = (r_h + \mathbb{P}_h V_{h+1}^{\mu,\dagger})(s,a,b),$$

$$V_h^{\mu,\dagger}(s) = \inf_{\nu \in \Delta_{\mathcal{B}}} (\mathbb{D}_{\mu_h \times \nu} Q_h^{\mu,\nu})(s)$$

- Nash equilibria:

$$Q_h^{*}(s,a,b) = (r_h + \mathbb{P}_h V_{h+1}^{*})(s,a,b),$$

$$V_h^{*}(s) = \sup_{\mu} \inf_{\nu} (\mathbb{D}_{\mu \times \nu} Q_h^{*})(s) = \inf_{\mu} \sup_{\nu} (\mathbb{D}_{\mu \times \nu} Q_h^{*})(s)$$

Sample Complexity of RL Algorithms

RL algorithms typically require a large amount of samples to be effective.

- AlphaGo Zero trained on $O(10^7)$ games and took over a month to train [SSS$^+$17].
- In two player Markov games, VI-ULCB finds an ϵ-approximate Nash equilibrium in $\Theta(\text{poly}(H)SAB/\epsilon^2)$ samples[BJ20].

The theoretical lower bound of samples needed to compute Nash equilibria in two player Markov games is $\Omega(\text{poly}(H)S(A+B)/\epsilon^2)$.

Sample Complexity of RL Algorithms

RL algorithms typically require a large amount of samples to be effective.

- AlphaGo Zero trained on $O(10^7)$ games and took over a month to train [SSS$^+$17].
- In two player Markov games, VI-ULCB finds an ϵ-approximate Nash equilibrium in $\Theta(\text{poly}(H)SAB/\epsilon^2)$ samples[BJ20].

The theoretical lower bound of samples needed to compute Nash equilibria in two player Markov games is $\Omega(\text{poly}(H)S(A+B)/\epsilon^2)$.

- Goal: Design an algorithm that learns a Markov game with near optimal sample complexity

Contributions

The paper:

- proposes an optimistic variant of Nash Q-learning with sample complexity $O(H^5 SAB/\epsilon^2)$ that finds an ϵ-approximate Nash equilibrium.
- describes a new algorithm Nash V-learning that achieves sample complexity $O(H^6 S(A+B)/\epsilon^2)$.
  - This improves on Nash Q-learning in the event that $\min(A, B) > H$.
- demonstrates that learning best responses of fixed opponents is as hard as learning parity with noise, which is thought to be computationally intensive.

Optimistic Nash Q-learning Algorithm

Algorithm 2 Optimistic Nash Q-Learning

Initialize: for any (s, a, b, h), $\bar{Q}_h(s, a, b) \leftarrow H$, $\underline{Q}_h(s, a, b) \leftarrow 0$, $N_h(s, a, b) \leftarrow 0$, $\pi_h(a, b|s) \leftarrow 1/(AB)$
for episode $k = 1$ to K **do**
  receive s_1.
  for step $h = 1$ to H **do**
    take action $(a_h, b_h) \sim \pi_h(\cdot, \cdot|s_h)$
    observe reward $r_h(s_h, a_h, b_h)$ and next state s_{h+1}
    $t = N_h(s_h, a_h, b_h) \leftarrow N_h(s_h, a_h, b_h) + 1$
    $\bar{Q}_h(s_h, a_h, b_h) \leftarrow (1 - \alpha_t)\bar{Q}_h(s_h, a_h, b_h) + \alpha_t(r_h(s_h, a_h, b_h) + \bar{V}_{h+1}(s_{h+1}) + \beta_t)$
    $\underline{Q}_h(s_h, a_h, b_h) \leftarrow (1 - \alpha_t)\underline{Q}_h(s_h, a_h, b_h) + \alpha_t(r_h(s_h, a_h, b_h) + \underline{V}_{h+1}(s_{h+1}) - \beta_t)$
    $\pi_h(\cdot, \cdot|s_h) \leftarrow \mathrm{CCE}(\bar{Q}_h(s_h, \cdot, \cdot), \underline{Q}_h(s_h, \cdot, \cdot))$
    $\bar{V}_h(s_h) \leftarrow (\mathbb{D}_{\pi_h}\bar{Q}_h)(s_h)$; $\underline{V}_h(s_h) \leftarrow (\mathbb{D}_{\pi_h}\underline{Q}_h)(s_h)$
  end for
end for

Where $\alpha_t = \frac{H+1}{H+t}, \beta_t = c\sqrt{\frac{H^3\iota}{t}}$ are hyperparameters.

Coarse Correlated Equilibrium (CCE)

Introduced by Xie et al.[XCWY20], CCE($\bar{Q}$, $\underline{Q}$) for any matrices $\bar{Q}, \underline{Q} \in [0, H]^{A \times B}$ returns a distribution in polynomial time $\pi \in \Delta_{A \times B}$ such that

$$\mathbb{E}_{(a,b)\sim\pi} \bar{Q}(a, b) \geq \max_{a^*} \mathbb{E}_{(a,b)\sim\pi} \bar{Q}(a^*, b)$$

$$\mathbb{E}_{(a,b)\sim\pi} \underline{Q}(a, b) \leq \min_{b^*} \mathbb{E}_{(a,b)\sim\pi} \underline{Q}(a, b^*)$$

Here, we define the following notation:

- $\alpha_t^0 := \prod_{j=1}^t (1-\alpha_j)$, $\alpha_t^i := \alpha_i \prod_{j=i+1}^t (1-\alpha_j)$, and $\sum_{i=1}^t \alpha_t^i = 1$
- $k_h^m(s,a,b)$ is the index of the episode where (s,a,b) was observed in step h for the m-th time.

Algorithm 3 Certified Policy $\hat{\mu}$ of Nash Q-Learning

1: sample $k \leftarrow \mathrm{Uniform}([K])$
2: **for** step $h = 1$ to H **do**
3:     observe s_h, and take action $a_h \sim \mu_h^k(\cdot|s_h)$
4:     observe b_h, and set $t \leftarrow N_h^k(s_h, a_h, b_h)$
5:     sample $m \in [t]$ with $\mathbb{P}(m = i) = \alpha_t^i$
6:     $k \leftarrow k_h^m(s_h, a_h, b_h)$
7: **end for**

Theorems for Nash Q-learning

We assume that the algorithm has played the game for K episodes, using V^k, Q^k, N^k, π^k to denote quantities at the beginning of the k-th episode.

Lemma 3

For any $p \in (0, 1]$ with $\iota = \log(SABT/p)$, algorithm 2 guarantees

- $\bar{V}_h^k(s) \geq V_h^*(s) \geq \underline{V}_h^k(s)$ for all s, h, k.
- $\frac{1}{K}\sum_{k=1}^{K}(\bar{V}_1^k - \underline{V}_1^k)(s) \leq \mathcal{O}(\sqrt{H^5 SAB\iota/K})$

with probability $1 - p$.

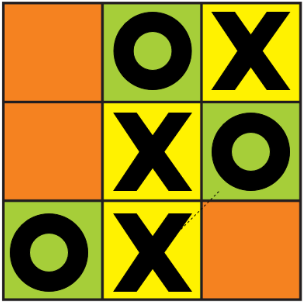
Theorem 4 (Sample complexity of Nash Q-learning)

For any $p \in (0, 1]$ with $\iota = \log(SABT/p)$, if we run algorithm 2 for K episodes where $K \geq \Omega(H^5 SAB\iota/\epsilon^2)$, the certified policies $\hat{\mu}, \hat{\nu}$ computed using algorithm 3 will be ϵ-approximate Nash with probability $1 - p$.

Improving on Nash Q-learning

- For each state, we have a fixed set of actions that yield varying unknown rewards
- Analogous to a bandit learning problem where $\mu(\cdot|s)$ can be represented as a set of weights for selecting each action

We can use bandit techniques to learn Nash equilibria.



Optimistic Nash V-learning algorithm

Algorithm 4 Optimistic Nash V-Learning (max-player version)

Initialize: for any (s, a, b, h), $\bar{V}_h(s) \leftarrow H$, $\bar{L}_h(s,a) \leftarrow 0$, $N_h(s) \leftarrow 0$, $\mu_h(a|s) \leftarrow 1/(A)$
for episode $k = 1$ to K **do**
    receive s_1.
    for step $h = 1$ to H **do**
        take action $(a_h) \sim \mu_h(\cdot|s_h)$, observe action b_h from opponent
        observe reward $r_h(s_h, a_h, b_h)$ and next state s_{h+1}
        $t = N_h(s_h) \leftarrow N_h(s_h) + 1$
        $\bar{V}_h(s_h) \leftarrow \min\{H, (1-\alpha_t)\bar{V}_h(s_h) + \alpha_t(r_h(s_h, a_h, b_h) + \bar{V}_{h+1}(s_{h+1}) + \beta_t)\}$
        for all $a \in A$ **do**
            $\bar{\ell}_h(s_h, a) \leftarrow [H - r_h(s_h, a_h, b_h) - \bar{V}_h(s_h)]\mathbb{I}\{a_h = a\}/[\mu_h(a_h|s_h) + \bar{\eta}_t]$
            $\bar{L}_h(s_h, a) \leftarrow (1-\alpha_t)\bar{L}_h(s_h, a) + \alpha_t \bar{\ell}_h(s_h, a)$
        end for
        set $\mu(\cdot|s_h) \propto \exp[-(\bar{\eta}_t/\alpha_t)\bar{L}_h(s_h, \cdot)]$
    end for
end for

Where we have hyperparameters

$$\alpha_t = \frac{H+1}{H+t}, \bar{\eta}_t = \sqrt{\frac{\log A}{At}}, \underline{\eta}_t = \sqrt{\frac{\log B}{Bt}}, \bar{\beta}_t = c\sqrt{\frac{H^4 A\iota}{t}}, \underline{\beta}_t = c\sqrt{\frac{H^4 B\iota}{t}}$$

Policy and Resulting Theorem

Algorithm 5 Certified Policy $\hat{\mu}$ of Nash V-Learning

sample $k \leftarrow \text{Uniform}([K])$
for step $h = 1$ to H **do**
    observe s_h, and set $t \leftarrow N_h^k(s_h)$
    sample $m \in [t]$ with $\mathbb{P}(m = i) = \alpha_t^i$
    $k \leftarrow k_h^m(s_h)$
    take action $a_h \sim \mu_h^k(\cdot|s_h)$
end for

Theorem 5 (Sample Complexity of Nash V-learning)

For any $p \in (0, 1]$ with $\iota = \log(SABT/p)$, if we run algorithm 4 for K episodes where $K \geq \Omega(H^6 S(A + B)\iota/\epsilon^2)$, the certified policies $\hat{\mu}, \hat{\nu}$ computed using algorithm 3 will be ϵ-approximate Nash with probability $1 - p$.

Theorem 6 (Hardness for learning the best response)

There exists a Markov game with deterministic transitions and rewards defined for any horizon $H \geq 1$ with $S = 2$, $A = 2$, and $B = 2$, such that if there exists a polynomial time algorithm for learning the best response for this Markov game, then there exists a polynomial time algorithm for learning parity with noise.

Two-state Markov Game

We define a game with two actions $\{a_0, a_1\}$ and $\{b_0, b_1\}$ for each player, H episodes, and therefore $2H$ states $\{i_0, i_1\}_{i=2}^{H}$ with 1_0 as the initial state and $\perp$ as the terminal state.

State/Action	(a_0, b_0)	(a_0, b_1)	(a_1, b_0)	(a_1, b_1)
i_0	$(i+1)_0$	$(i+1)_0$	$(i+1)_0$	$(i+1)_1$
i_1	$(i+1)_1$	$(i+1)_0$	$(i+1)_1$	$(i+1)_1$

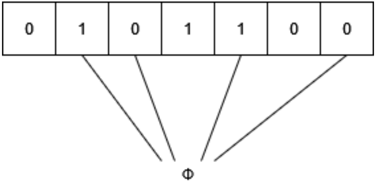
Table 1: Transition Kernel of the Markov Game

State/Action	$(\cdot, b_0)$	$(\cdot, b_1)$
H_0	1	0
H_1	0	1

Table 2: Reward matrix of the Markov Game

Learning Parity with Noise Problem

Given x a vector of 0s and 1s of size n, parity is defined as a function $\phi_T(x)$ that returns 0 if the number of ones in the subvector $(x_i)_{i \in T}$ is even and 1 otherwise.



Suppose we have a noisy query function $f(x)$ such that $f(x) = \phi_T(x)$ with probability α and $f(x) = 1 - \phi_T(x)$ with probability $1 - \alpha$.

Set of Computational Problems

1. The max-player ϵ-approximates the best response for any general policy ν in the Markov game with probability at least $1/2$ in poly$(H, 1/\epsilon)$ time.
2. Suppose we have $x \in \{0,1\}^n$, $T \subseteq [n]$, and the noisy query function $f(x)$. Find a function $h : \{0,1\}^n \to \{0,1\}$ such that:
    1. With probability at least $1/2$, $\mathbb{E}_h P_x[h(x) \neq \phi_T(x)] \leq \epsilon$ in poly$(n, 1/\epsilon)$ time.
    2. With probability at least $1/4$, $P_x[h(x) \neq \phi_T(x)] \leq \epsilon$ in poly$(n, 1/\epsilon)$ time.
    3. With probability at least $1 - p$, $P_x[h(x) \neq \phi_T(s)] \leq \epsilon$ in poly$(n, 1/\epsilon, 1/p)$ time.

Problem 2.3 reduces to Problem 2.2

1. Repeatedly apply algorithm for problem 2.2 ℓ times to obtain $h_1, \cdots, h_\ell$ such that
$$\min_i P_x[h_i(x) \neq \phi_T(x)] \leq \epsilon \text{ w.p at least } 1 - (3/4)^\ell$$
Define $i_* = \text{argmin}_i \text{err}_i$ where $\text{err}_i = P_x[h_i(x) \neq \phi_T(x)]$.
2. Construct estimators using N additional data points $(x^{(j)}, y^{(j)})_{j=1}^N$,
$$\hat{\text{err}}_i := \frac{\frac{1}{N}\sum_{j=1}^N \mathbb{I}\{h_i(x^{(j)} \neq y^{(j)}\} - \alpha}{1 - 2\alpha}$$
Choose $\hat{i} = \text{argmin}_i \hat{\text{err}}_i$. For $N \geq \ln(1/p)/\epsilon^2$, w.p at least $1 - p/2$, we have
$$\max_i |\hat{\text{err}}_i - \text{err}_i| \leq \frac{\epsilon}{1 - 2\alpha}$$
This step takes $\text{poly}(n, N, \ell) = \text{poly}(n, 1/\epsilon, \log(1/p), \ell)$ time. We therefore have:
$$\text{err}_{\hat{i}} \leq \hat{\text{err}}_{\hat{i}} + \frac{\epsilon}{1 - 2\alpha} \leq \hat{\text{err}}_{i_*} + \frac{\epsilon}{1 - 2\alpha} \leq \text{err}_{i_*} + \frac{2\epsilon}{1 - 2\alpha} \leq O(1)\epsilon$$

Problem 2.2 reduces to Problem 2.1

Markov's inequality states that for a non-negative RV X,

$$X \leq \frac{\mathbb{E}[X]}{1-p}$$

with probability $1-p$. Suppose we have an algorithm that gives h such that $\mathbb{E}_h P_x[h(x) \neq \phi_T(x)] \leq \epsilon$ with $1/2$ probability. Assuming this condition is satisfied, we can then sample an $\hat{h}$ such that with probability $1/2$,

$$P_x[h(x) \neq \phi_T(x)] \leq 2\epsilon$$

by Markov's inequality. Thus, with probability $1/4$, we have

$$P_x[h(x) \neq \phi_T(x)] \leq 2\epsilon$$

Problem 2.1 reduces to Problem 1

State/Action	(a_0, b_0)	(a_0, b_1)	(a_1, b_0)	(a_1, b_1)
i_0	$(i+1)_0$	$(i+1)_0$	$(i+1)_0$	$(i+1)_1$
i_1	$(i+1)_1$	$(i+1)_0$	$(i+1)_1$	$(i+1)_1$

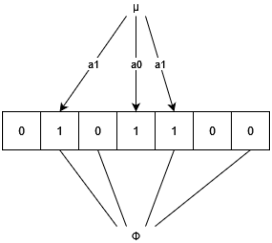
State/Action	$(\cdot, b_0)$	$(\cdot, b_1)$
H_0	1	0
H_1	0	1

Consider the Markov game constructed previously with $H - 1 = n$. We define the policy of the min-player ν as follows:

- Draw a sample (x, y) from the noisy query function.
- For each step $h \leq H - 1$, if x_h=0, take action b_0. Otherwise, take action b_1.
- At step H, take b_0 if $y = 0$ and b_1 otherwise.

Problem 2.1 reduces to Problem 1, cont.

The policy $\hat{\mu}$ can be thought of as a set of indices $\hat{T} \subseteq [H]$ where it takes action a_1 at all indices in $\hat{T}$ and a_0 otherwise.



The max-player only receives a nonzero reward iff $\phi_{\hat{T}}(s) = y$

In the Markov game, we have

$$V_1^{\mu,\nu}(s_1) = \mathbb{E}[\mathbb{I}(\phi_{\hat{T}}(s) = y)] = \mathbb{P}(\phi_{\hat{T}}(s) = y)$$

This implies that the optimal policy μ^* corresponds to the actual parity set T. By the ϵ-approximation guarantee,

$$\begin{aligned}(V_1^{\dagger,\nu} - V_1^{\hat{\mu,\nu}})(s_1) &= \mathbb{P}_{x,y}(\phi_T(x) = y) - \mathbb{P}_{x,y}(\phi_{\hat{T}}(x) = y) \\ &= (1 - \mathbb{P}_{x,y}(\phi_T(x) \neq y)) - (1 - \mathbb{P}_{x,y}(\phi_{\hat{T}}(x) \neq y)) \\ &= \mathbb{P}_{x,y}(\phi_{\hat{T}}(x) \neq y) - \mathbb{P}_{x,y}(\phi_T(x) \neq y) \leq \epsilon\end{aligned}$$

Problem 2.1 reduces to Problem 1, cont.

Next, we condition over the actual parity set T:

$$\begin{aligned}\mathbb{P}_{x,y}(\phi_{\hat{T}}(x) \neq y) - \mathbb{P}_{x,y}(\phi_T(x) \neq y) &= (1-\alpha)\mathbb{P}_x(\phi_{\hat{T}}(x) \neq \phi_T(x)) \\ &\quad + \alpha\mathbb{P}_x(\phi_{\hat{T}}(x) = \phi_T(x)) - \alpha \\ &= (1-2\alpha)\mathbb{P}_x(\phi_{\hat{T}}(x) \neq \phi_T(x))\end{aligned}$$

Thus,

$$\mathbb{P}_x(\phi_{\hat{T}}(x) \neq \phi_T(x)) \leq \frac{\epsilon}{1-2\alpha}$$

Conclusion

This paper:

- proposed an Optimistic Nash Q-Learning, which finds ϵ-approximate Nash equilibrium with sample complexity $O(H^5SAB/\epsilon^2)$.
- introduces a new algorithm Nash V-learning that achieves sample complexity $O(H^6S(A+B)/\epsilon^2)$, which matches the theoretical lower bound for zero-sum MGs.
- shows the difficulty in computing optimal policies in MGs by proving equivalence of solving a fixed Markov game with the problem of learning parity with noise.

[BJ20] Yu Bai and Chi Jin.
Provable self-play algorithms for competitive reinforcement learning, 2020.

[FV96] Jerzy A. Filar and Koos Vrieze.
Competitive markov decision processes.
1996.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George Driessche, Thore Graepel, and Demis Hassabis.
Mastering the game of go without human knowledge.
Nature, 550:354–359, 10 2017.

[XCWY20] Qiaomin Xie, Yudong Chen, Zhaoran Wang, and Zhuoran Yang.
Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium, 2020.

Thanks!