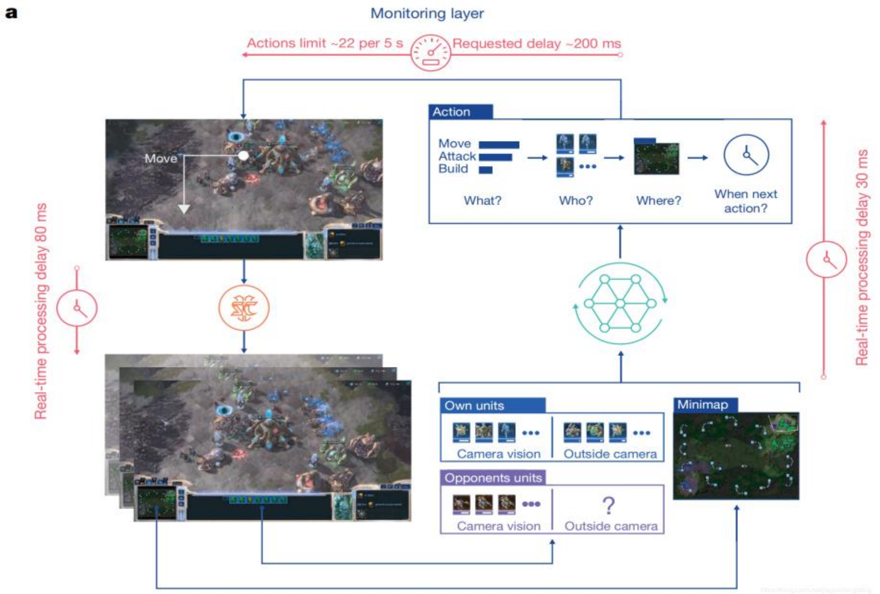
Local Optimization Achieves Global Optimality in Multi-Agent Reinforcement Learning

Yulai Zhao Zhuoran Yang Zhaoran Wang Jason Lee

Backgrounds

- Multi-agent reinforcement learning (MARL) has demonstrated many empirical successes, e.g. strategic games (Go, StarCraft II…)
- Policy optimization methods are widely used in MARL (AlphaGo, LOLA…)



Main challenges in MARL (Zhang 2021)

1. non-stationarity: each action taken by one agent affects the total reward and the transition of state.
2. scalability: taking other agents into consideration, each individual agent would face the joint action space, whose dimension increases exponentially with the number of agents
3. function approximation: closely related to the scalability issue, the state space and joint action space are often immense in MARL

Motivation

Despite the empirical successes, theoretical studies of policy optimization in MARL are very limited. Even for the cooperative setting where the agents share a common goal: maximizing the total reward function

In this paper, we aim to answer the following fundamental question:

Can we design a provably convergent multi-agent policy optimization algorithm in the cooperative setting with function approximation?

Contributions

1. We answer the above question affirmatively.
2. We propose a multi-agent PPO algorithm in which the local policy of each agent is updated sequentially in a similar fashion as vanilla PPO algorithm (Schulman et al., 2017).
3. We adopt the log-linear function approximation for the policies. We prove that multi-agent PPO converges at a sublinear $O\left(\frac{N}{1-\gamma}\sqrt{\frac{\log(|A|)}{K}}\right)$ rate up to some statistical errors incurred in evaluating/improving policies.
4. Moreover, we propose an off-policy variant of the multi-agent PPO algorithm and introduce pessimism into policy evaluation.

Problem Setup

- Fully-cooperative Markov Games
  - ➢ a tuple $M = (N, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathrm{r}, \gamma)$: A party of participants N, a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition probability $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \Delta(\mathcal{S})$, a reward function $\mathrm{r}: \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to [0, 1]$, a discounted factor $\gamma \in [0, 1)$.
  - ➢ define policies as probability distributions over action space: $\pi \in \mathcal{S} \to \Delta(\mathcal{A})$.
- Value function

$$V^{\pi}(s) = E_{a\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\, r(s_t, a_t, b_t)|\, s_0 = s\right]$$

Multi-agent Notations

- We write index k on superscript when we refer to the specific k-th agent. When bold symbols are used without any superscript (e.g., $\boldsymbol{a}$), they consider all agents. For simplicity, let $(m{:}\,m')$ be shorthand for set: $\{i|m \leq i \leq m', i \in N\}$.
- Definition 3.1. Let P be a subset in N . The multi-agent action value function associated with agents in P is

$$Q_\pi^P\left(s, \boldsymbol{a}^P\right) = E_{\widetilde{\boldsymbol{a}} \sim \widetilde{\boldsymbol{\pi}}}[Q_\pi\left(s, \boldsymbol{a}^P, \widetilde{\boldsymbol{a}}\right)]$$

here we use a tilde over symbols to refer to the complement agents, namely $\tilde{a} = \left\{a^i \middle| i \notin P, i \in N\right\}$.

Advantage function and Bellman operators

- Let $P, P' \subset N$ be two disjoint subsets of agents. The multiagent advantage function is defined below. Essentially, it accounts for the improvements of setting agents $a^{P'}$ upon setting agents a^{P}, while all other agents follow $\boldsymbol{\pi}$

$$A_{\pi}^{P'}\left(s, \boldsymbol{a}^{P}, \boldsymbol{a}^{P'}\right) = Q_{\pi}^{P \cup P'}\left(s, \boldsymbol{a}^{P}, \boldsymbol{a}^{P'}\right) - Q_{\pi}^{P}(s, \boldsymbol{a}^{P})$$

- For $m \in N$ and any function $f: S \times A^{m} \rightarrow R$ we define multi-agent Bellman operator $T_{\pi}^{1:m}: R^{S \times A^{m}} \rightarrow R^{S \times A^{m}}$ as

$$T_{\pi}^{1:m} f\left(s, a^{1:m}\right) = E_{\tilde{a} \sim \tilde{\pi}} r\left(s, a^{1:m}, \tilde{a}\right) + \gamma E_{\tilde{a} \sim \tilde{\pi}, s' \sim P} f(s', \pi^{1:m})$$

where $f(s', \pi^{1:m})$ is shorthand for $E_{a' \sim \pi^{1:m}} f(s', a')$

Population Algorithm for online setting

Parametrization For the m-th agent ($m \in \mathcal{N}$), its conditional policy depends on all prior ordered agents $\mathbf{a}^{1:m-1}$. Given a coefficient vector $\theta^m \in \Theta$, where $\Theta = \{\|\theta\| \leq R | \theta \in \mathbb{R}^d\}$ is a convex, norm-constrained set. The probability of choosing action a^m under state s is

$$\pi_{\theta^m}(a^m | s, \mathbf{a}^{1:m-1}) = \frac{\exp\left(\phi^\top(s, \mathbf{a}^{1:m-1}, a^m)\theta^m\right)}{\sum_{a^m \in \mathcal{A}} \exp\left(\phi^\top(s, \mathbf{a}^{1:m-1}, a^m)\theta^m\right)} \tag{2}$$

Policy Improvement At the k-th iteration, we define $\hat{\pi}^m_{k+1}$ as the ideal update based on $\hat{Q}^{1:m}_{\boldsymbol{\pi}_{\theta_k}}$ (for agent $m \in \mathcal{N}$), which is an estimator of $Q^{1:m}_{\boldsymbol{\pi}_{\theta_k}}$. The ideal update is obtained via the following update

$$\hat{\pi}^m_{k+1} \leftarrow \arg\max_{\pi^m} \hat{F}(\pi^m) \tag{3}$$

$$\hat{F}(\pi^m) = \mathbb{E}_{\sigma_k}\left[\langle \hat{Q}^{1:m}_{\boldsymbol{\pi}_{\theta_k}}(s, \mathbf{a}^{1:m-1}, \cdot), \pi^m(\cdot | s, \mathbf{a}^{1:m-1})\rangle - \beta_k KL\left(\pi^m(\cdot|s, \mathbf{a}^{1:m-1}) \| \pi_{\theta^m_k}(\cdot | s, \mathbf{a}^{1:m-1})\right)\right]$$

where θ^m_k is the parameter of the current conditional policy of the m-th agent. In above equation, the distribution is taken over $(s, \mathbf{a}^{1:m-1}) \sim \nu_k \boldsymbol{\pi}^{1:m-1}_{\theta_k}$, we write σ_k for simplicity.

Population Algorithm for online setting

- In practice, we approximate such policy within a certain parametrization class, which is the log-linear function class.

To approximate the ideal $\hat{\pi}^m_{k+1}$ using a parameterized $\pi_{\theta^m_{k+1}} \propto \exp\{\phi^\top \theta^m_{k+1}\}$, we minimize the following mean-squared error (MSE) as a sub-problem

$$\theta^m_{k+1} \leftarrow \operatorname*{arg\,min}_{\theta^m \in \Theta} L(\theta^m) \tag{4}$$

where $L(\theta^m)$ is defined as

$$L(\theta^m) = \mathop{\mathbb{E}}_{\sigma_k} \left((\theta^m - \theta^m_k)^\top \phi(s, \mathbf{a}^{1:m-1}, a^m) - \frac{\hat{Q}^{1:m}_{\boldsymbol{\pi}_{\theta_k}}(s, \mathbf{a}^{1:m-1}, a^m)}{\beta_k} \right)^2$$

- Note that this MSE can be minimized using the classic SGD updates.

Population Algorithm for online setting

Policy Evaluation In this step, we aim to examine the quality of the attained policy. Thereby, a Q-function estimator is required. We make the following assumption.

Assumption 4.3. Assume we can access an estimator of Q function that returns $\hat{Q}$. The returned $\hat{Q}$ satisfies the following condition for all $m \in \mathcal{N}$ at the k-th iteration

$$\left[\mathbb{E}_{\sigma_k}\left(\hat{Q}^{1:m}_{\boldsymbol{\pi}_{\theta_k}}(s, \mathbf{a}^{1:m-1}, a^m) - Q^{1:m}_{\boldsymbol{\pi}_{\theta_k}}(s, \mathbf{a}^{1:m-1}, a^m)\right)^2\right]^{1/2} \leq \xi^m_k.$$

Algorithm

Algorithm 1 Multi-Agent PPO

Input: Markov game $(\mathcal{N}, \mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{P}, r, \gamma)$, penalty parameter β, stepsize η for sub-problem, number of SGD iterations T, number of iterations K.
Output: Uniformly sample k from $0, 1, \cdots K-1$, return $\bar{\boldsymbol{\pi}} = \boldsymbol{\pi}_{\theta_k}$.

Initialize $\theta_0^m = 0$ for every $m \in \mathcal{N}$.
for $k = 0, 1, \ldots, K-1$ **do**
  Set parameter $\beta_k \leftarrow \beta\sqrt{K}$
  for $m = 1, \cdots, N$ **do**
    Sample $\{s_t, \mathbf{a}_t^{1:m-1}, a_t^m\}_{t=0}^{T-1}$ from $\sigma_k = \nu_k \boldsymbol{\pi}_{\theta_k}$.
    Obtain $\hat{Q}_{\boldsymbol{\pi}_{\theta_k}}^{1:m}(s, \mathbf{a}^{1:m-1}, a^m)$ for each sample .
    Feed samples into Algorithm 3, obtain θ_{k+1}^m.
  end for
end for

Algorithm 3 Policy Improvement Solver for MA-PPO

Input: MG $(\mathcal{N}, \mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{P}, r, \gamma)$, iterations T, stepsize η, samples $\{s_t, \mathbf{a}_t^{1:m-1}, a_t^m\}_{t=0}^{T-1}$.
Output: Policy update θ.

Initialize $\theta_0 = 0$.
for $t = 0, 1, \ldots, T-1$ **do**
  Let $(s, \mathbf{a}^{1:m-1}, a) \leftarrow (s_t, \mathbf{a}_t^{1:m-1}, a_t^m)$.
  $\theta(t+\frac{1}{2}) \leftarrow \theta(t) - 2\eta\phi(s, \mathbf{a}^{1:m-1}, a)\left(\left(\theta(t) - \theta_k^m\right)^\top \phi(s, \mathbf{a}^{1:m-1}, a^m) - \beta_k^{-1}\hat{Q}_{\boldsymbol{\pi}_k}^{1:m}(s, \mathbf{a}^{1:m-1}, a^m)\right)$.
  $\theta(t+1) \leftarrow \Pi_\Theta \theta(t+\frac{1}{2})$
end for
Calculate average: $\bar{\theta} \leftarrow \frac{1}{T}\sum_{t=1}^{T} \theta_t$.

Theoretical results

- Theorem 1 (informal): For this setting, after K iterations, we have $J(\pi^*) - J(\bar{\pi})$ upper bounded by

$$\mathcal{O}\left(\frac{B\sqrt{N}}{1-\gamma}\sqrt{\frac{N\log|\mathcal{A}| + \sum_{m=1}^{N}\sum_{k=0}^{K-1}(\Delta_k^m + \delta_k^m)}{K}}\right)$$

where $\Delta_k^m = \sqrt{2}(\phi_k^m + \phi_k^{m-1}) \cdot \left(\epsilon_k^m + \frac{\xi_k^m}{\beta_k}\right)$ *and* $\delta_k^m = 2\phi_k^{m-1}\epsilon_k^m$. *Here* ϵ_k^m *is the statistical error of a PPO iteration: for agent* $m \in \mathcal{N}$,

$$\mathbb{E}_{\sigma_k}\left((\theta_{k+1}^m - \theta_k^m)^\top \phi - \beta_k^{-1}\hat{Q}_{\boldsymbol{\pi}_{\theta_k}}^{1:m}\right)^2 \leq (\epsilon_k^m)^2$$

Pessimistic MA-PPO with Linear Function Approximation

- We perform pessimistic policy evaluation via regularization to reduce such overestimation aligning with experimental works.

- Theorem 1 (informal): For this setting, after K iterations, we have $J(\pi^*) - J(\bar{\pi})$ upper bounded by

$$\mathcal{O}\left(\frac{N}{(1-\gamma)^2}\sqrt{\frac{\log|\mathcal{A}|}{K}} + \frac{C_\mu^{d_{\boldsymbol{\pi}*}}}{(1-\gamma)^2}\sqrt[3]{\frac{d\log\frac{nLR}{\delta}}{n}}\right)$$

Thank you and some more information

SCAN ME

paper

Yulai Zhao
PhD student @ Princeton