

Reinforcement Learning for Cooperative Multi-Agent Systems

Liangyawei Kuang

Department of Computer Science and Engineering, Hong Kong University of Science and Technology
lkuang@connect.ust.hk

Abstract

With the rising of robotics, autonomous vehicles, and other notable reinforcement learning (RL) applications in recent years, multi-agent RL (MARL) has reemerged with the advances in single-agent RL. As an interdisciplinary research area that includes learning, game theory, communication, and optimization, there are many technics come with challenges and restrictions in different MARL settings, MARL-related subareas, and theoretical foundation lackings. For example, the centralized training and decentralized execution (CTDE) paradigm is successfully used to tackle communication problems as it enables agents to share information during training and somehow reduce other concerns on nonstationarity and partial observability. However, as CTDE is under homogeneous settings, this paradigm may be not beneficial in a more general multi-agent setting with heterogeneous agents, which means agents are not the same on capabilities and goals. In this short paper which mainly focused on cooperative MARL, I will provide problems background, introduce selective current challenges, and propose potential directions for future solutions.

Background

In this section, I will review the theoretical methods from single-agent RL to MARL, especially methods related to cooperative MARL settings.

Markov Decision Process.

For a reinforcement learning question, we can formulate it as a infinite-horizon discounted Markov Decision Process (MDP). An MDP is defined by a quintuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where \mathcal{S} is the set of states; \mathcal{A} is the set of actions; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ denotes the set of possibility from a state $s \in \mathcal{S}$ to a state $s' \in \mathcal{S}$, given an action $a \in \mathcal{A}$; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the immediate reward function for agents transfer from (s, a) to s' ; $\gamma \in [0, 1)$ is the discount factor.

At time t , the agent in state s_t executes an action a_t by following the policy $\pi : \pi(a|s)$, which is a mapping from states \mathcal{S} to actions \mathcal{A} . The system transit from state s_t to the next state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. For MDPs, the goal is to find the optimal policy π to maximize $a_t \sim \pi(\cdot|s_t)$ and the accumulated rewards

$$\mathbb{E} \left[\sum_{t \geq 0} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), s_0 \right].$$

Accordingly, given policy π , for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we could define the *action-value function* (the Q-function), which is starting from $(s_0, a_0) = (s, a)$, as

$$Q_\pi(s, a) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), a_0 = a, s_0 = s \right],$$

and the *state-value function* (the V-function), starting from $s_0 = s$, as

$$V_\pi(s, a) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), s_0 = s \right]$$

π^* are referred to the optimal policy as the optimal Q-function and V-function respectively. By virtue of the Markov property, the optimal function could be obtained by iteration based on dynamic programming (DP), which is usually required of the complete knowledge of the model.

Value-Based Methods.

The value-based methods are mainly to find the estimate of the optimal Q-function Q_π^* . One famous value-based algorithm is Monte-Carlo tree search (MCTS), which is used under the incomplete environment knowledge. In this method, a monte carlo simulation is executed based on a search tree to estimate the optimal value function.

Temporal-Difference (TD) learning method is a combination of MCTS and DP. It learns the estimates partially based on estimates, which is known as *bootstrapping*. As model free methods, TD methods are implemented more naturally than MCTS and DP with an online and fully incremental way.

Q-learning is one of the most important value-based method, which is actually an off-policy TD control. The optimal policy can be approximated by taking the greedy action of estimation of the Q-value function $\hat{Q}(s, a)$. The Q-function is updated according to

$$\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left[r + \gamma \max_{a'} \hat{Q}(s', a') - \hat{Q}(s, a) \right]$$

with the loss function

$$\mathcal{L}(s, a, r, s') = (r + \gamma \max_{a'} Q(s', a') - Q(s, a))^2.$$

Policy-Based Methods.

The main idea in policy-based method is to update the parameter by following the gradient direction, which is known as policy gradient (PG). The closed-form of PG is given as

$$\nabla J(\theta) = \mathbb{E}_{a \sim \pi_\theta(\cdot|s), s \sim \eta_{\pi_\theta}(\cdot)} \left[\mathcal{Q}_{\pi_\theta}(s, a) \nabla \log \pi_\theta(a|s) \right],$$

where $J(\theta)$ and \mathcal{Q}_π are the expected reward and Q-function with following policy π_θ , respectively, $\pi_\theta(\cdot|s)$ is the approximation of $\pi(\cdot|s)$, η_{π_θ} is the measurement of state occupancy, and $\nabla \log \pi_\theta(a|s)$ is the score of the policy.

Compared with value based methods, policy based one are more powerful with better convergence guarantees with neural networks for function approximation, which is a fashion today with the rise of Deep Learning (DL). And policy-based method are believed to have the ability to handle bigger discrete or even continuous state-action spaces.

Markov Games.

Markov games (MGs), which is also known as stochastic games, is originally a framework for MDP in multi-agent settings. A markov game is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i \in \mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, \dots, N\}$ denotes the set of N agents, \mathcal{S} denotes the globally observed state space by the whole system, \mathcal{A}^i denotes the action space of agent i , $\mathcal{P} : \mathcal{S} \times \{\mathcal{A} := \mathcal{A}^1 \times \dots \times \mathcal{A}^N\} \rightarrow \Delta(\mathcal{S})$ denotes the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ denotes the possibility distribution for mapping from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ via any joint action $a \in \mathcal{A}$, $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ denotes the immediate reward received by agent i via a transition from (s, a) to s' , $\gamma \in [0, 1)$ denotes the discount factor.

From time t to time $t + 1$, agent $i \in \mathcal{N}$ executes action a_t^i , the system will transite from s_t to s_{t+1} , and all agents get immediate reward by $R^i(s_t, a_t, s_{t+1})$. For every individual agent i , the goal is to maximize its own reward in a long finite horizon or infinite horizon by finding the optimal policy $\pi^i : \mathcal{S} \rightarrow \Delta(\mathcal{A}^i)$ so that $a_t^i \sim \pi^i(\cdot|s_t)$. The joint policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ is $\pi(a|s) := \prod_{i \in \mathcal{N}} \pi^i(a^i|s)$. For any state $s \in \mathcal{S}$ and joint policy π ,

$$V_{\pi^i, \pi^{-i}}^i(s) := \mathbb{E} \left[\sum_{t \geq 0} \gamma^t R^i(s_t, a_t, s_{t+1}) \middle| a_t^i \sim \pi^i(\cdot|s_t), s_0 = s \right],$$

where $-i$ denotes the indices of all other agents in \mathcal{N} except agent i . A nash equilibrium (NE) of a *markov game* $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i \in \mathcal{N}}, \gamma)$ is a joint policy $\pi^* = (\pi^{1,*}, \dots, \pi^{N,*})$ so that for any $s \in \mathcal{S}$, $i \in \mathcal{N}$, and π^*

$$V_{\pi^{i,*}, \pi^{-i,*}}^i(s) \geq V_{\pi^i, \pi^{-i,*}}^i(s)$$

The nash equilibrium point π^* is a fixed point so that all agent won't transite to a better point as there is not any incentive to do so. For any agent $i \in \mathcal{N}$, $\pi^{i,*}$ is the best response to $\pi^{-i,*}$. For MARL settings, finding the NE is a standard learning goal and NE always exists for finite-space infinite-horizon discounted MGs (?).

Cooperative Settings.

In this short paper, we only consider cooperative settings

which mean all the agents collaborate with each other to achieve shared goal. In a fully cooperative setting, all agents share one reward function $\mathcal{R}^1 = \mathcal{R}^2 = \dots = \mathcal{R}^N = \mathcal{R}$. With this model, the Q-function is identical to all agents so that Q-learning updates could be applied with taking the max over the joint action space $a' \in \mathcal{A}$. In a more general cooperative setting, agents have their own reward function and the goal is to optimize the long-term reward for all agents. One common reward model is *team-averager* reward $\bar{R}(s, a, s') := N^{-1} \cdot \sum_{i \in \mathcal{N}} R^i(s, a, s')$ for any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$.

Partial Observability.

In multi-agent settings, we can not ignore the influence from the real environment, the noise and limited sensors may prevent the agent from observing the state of the environment (?). However, MGs can only handle the fully observed environment. In this case, *Partially Observable Markov Decision Process* (POMDP) is more suitable to represent such state uncertainty by incorporating observations and their probability of occurrence conditional on the state of the environment (?). More generally speaking, this partially observed setting can be modeled by a decentralized POMDP (Dec-POMDP), which shares most of the elements, including the reward function and the transition model. Based on MGs, a Dec-POMDP is formally defined by the tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i \in \mathcal{N}}, \mathcal{Z}, \mathcal{O}, \gamma)$, where $\mathcal{Z} := \{\mathcal{Z}^1 \times \dots \times \mathcal{Z}^N\}$ denotes the joint observations, $\mathcal{O} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{Z}$ denotes the observation probabilities, the others denotation remain the same as MGs. Under this setting, At time t , agent i has its *action-observation* history $\mathcal{H}_i := [\mathcal{O}_{i,1}, \mathcal{A}_{i,1}, \dots, \mathcal{O}_{i,t-1}, \mathcal{A}_{i,t-1}]$, $\mathcal{H}_i \in \mathcal{H}$, and a stochastic policy for agent i is $\pi^i(\mathcal{A}^i|\mathcal{H}^i)$. Given the history \mathcal{H} , the goal for each agent is to maximize its expected discounted rewards in a long term.

Challenges

The challenges in *Cooperative Multi-Agent Reinforcement Learning* (CMARL) not only lie in problems with the models based on MDP but also in the different training schemes and the lacking of theoretical foundations. In this session, I will cover several important challenges.

Non-Stationarity.

In CMARL, one major issue is that the environment for each agent is non-stationary during the learning process as other agents are learning at the same time. One agent's action influences other agents' reward functions and the transition functions. In single agent RL, the stationarity Markovian property is assumed so that the convergence could be guaranteed. However, in CMARL settings, the agents need to model other agents' behavior and adapt to the *joint behavior*.

Related Work

Proposed Methods

Conclusion

References

Filar, J.; and Vrieze, K. 2012. *Competitive Markov decision processes*. Springer Science & Business Media.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2): 99–134.

Oliehoek, F. A.; and Amato, C. 2016. *A concise introduction to decentralized POMDPs*. Springer.