Multi-Agent Reinforcement Learning: Concepts and Challenges

Settings

1. Fully cooperative.

   Agents collaborate to optimize a common return.

   有共同的目标，利益一致。例如，生产线机器人都是为了装配好汽车。大家在每个时刻获得的奖励相同。对于一个agent，他会想：我在t时刻的 未来累积奖励 Ut 将会由当前汽车装配的程度(状态)、我和其他agents在t时刻及以后的所有动作来决定。

2. Fully competitive.

   0和博弈。双方获得的奖励和为0。一方的奖励为另一方的损失。

3. Mixed Cooperative & competitive.

4. Self-interested.

第四种设定是 self-interested，也就是“利己主义”。系统里有多个agents，一个agent的动作会改变环境的状态，可能让别人获益或者 受损。利己主义的意思是 每个agent只想要最大化自身利益，至于让别人受益或是受损，它不在乎。这与合作和竞争 有所不同。合作的目标是有钱大家一起赚，竞争的目标是让对手受损。利己主义者不会 刻意帮别人或者害别人，尽管他的行为在客观上会影响别人的利益。例如：股票和期货的自动交易系统可以把它看做 一个agent，自动交易系统的交易量很大，每一笔交易都会影响股价。也就是说，他的动作会影响股市的状态。股市里有非常多的自动交易系统，他们对市场的影响非常大，这些系统是self-interested，目标是最大化自己的收益。他们在最大化自己利益的同时，客观上会导致其他交易系统 受益或者受损。

1. Terminologies

1.1 State, Action, State Transition

- There are n agents.
- Let S be the state. agents处于同一个环境中，所以不同agents在一个时刻具有相同的状态
- Let A^i be the i-th agent's action. 本节中，上标表示 agent 的编号; 下标表示时间点。
大写字母表示随机变量；小写字母表示随机变量的一个观测值，例如: a^2_t表示在t时刻，agent 2的一个通过随机采样得到的动作。
- State transition: 每一个agent动作都会影响下一个状态，从而影响其余所有agents。这说明agents之间以环境为媒介相互影响，而非彼此独立。这里要建立一个agent相互影响的概念

$$p(s'|s, a^1, \cdots, a^n) = \mathbb{P}(S' = s'|S = s, A^1 = a^1, \cdots, A^n = a^n).$$

- The next state, S', depends on all the agents' actions.

1.2 Rewards

- Let R^i be the reward received by the i-th agent.
- Fully cooperative: $R^1 = R^2 = \cdots = R^n$.
- Fully competitive: $R^1 \propto -R^2$.
- R^i depends on A^i as well as all the other agents' actions $\{A^j\}_{j \neq i}$.

在多个agents交互的环境中，一个agent获得的奖励不仅取决于自己的动作，还取决于其他agents的动作

1.3 Returns

- Let R_t^i be the **reward** received by the i-th agent at time t. 这是一个随机变量
- **Return** (of the i-th agent):

$$U_t^i = R_t^i + R_{t+1}^i + R_{t+2}^i + R_{t+3}^i + \cdots$$

- **Discounted return** (of the i-th agent):

$$U_t^i = R_t^i + \gamma \cdot R_{t+1}^i + \gamma^2 \cdot R_{t+2}^i + \gamma^3 \cdot R_{t+3}^i + \cdots$$

Here, $\gamma \in [0, 1]$ is the discount rate.

1.4 Policy Network

- Each agent has its own policy network: $\pi\left(a^i \mid s; \boldsymbol{\theta}^i\right)$.
- Policy networks can be exchangeable: $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^2 = \cdots = \boldsymbol{\theta}^n$.
  - Self-driving cars can have the same policy.
- Policy networks can be nonexchangeable: $\boldsymbol{\theta}^i \neq \boldsymbol{\theta}^j$.
  - Soccer players have different roles, e.g., striker, defender, goalkeeper.

1.5 Uncertainty in the Return

The return, $U_t^i = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k}^i$

- The reward R_t^i depends on S_t and $A_t^1, A_t^2, \cdots, A_t^n$.
- Uncertainty in S_t is from the state transition, p.
- Uncertainty in A_t^i is from the policy network, $\pi(\cdot \mid s_t; \boldsymbol{\theta}^i)$.

The return depends on:

- all the future states: $\{S_t, S_{t+1}, S_{t+2}, \cdots\}$;
- all the future actions: $\{A_t^i, A_{t+1}^i, A_{t+2}^i, \cdots\}$, for all $i = 1, \cdots, n$.

1.6 State-Value Function

- State-value of the i-th agent:

$$V^i(s_t; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n) = \mathbb{E}\left[U_t^i \mid S_t = s_t\right].$$

The expectation is taken w.r.t. all the future actions and states except S_t.

第i个agent在t时刻处于局势s_t的好坏，不仅取决于自己的policy theta^i，还取决于其他所有agents的policy。例如，在足球运动员agents中，一个agent当前局势的好坏，与队友的policy(梅西的policy，猪队友的policy)密切相关。

从另一个角度来理解：第i个agent在t时刻处于局势s_t的好坏与其他agents的动作有关。而动作由theta参数来决定，即

Randomness in actions: $A_t^j \sim \pi\left(\cdot \mid s_t; \boldsymbol{\theta}^j\right)$, for all $j = 1, \cdots, n$.

所以： the state-value V^i depends on $\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n$.)

- One agent's state-value, $V^i(s; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)$, depends on all the agents' policies.
- If any agent changes its policy, then all of $V^1, \cdots, V^n$ can change.
- Example: soccer game.
  - A striker improves his policy, while everyone else's policies are fixed.
  - His teammates' state-values all increase.
  - The opposing players' state-values all decrease.

这就是mulit-agent的复杂之处：你的状态价值函数不仅取决于你的策略，还受到其他人策略的影响。改进自己的策略未必使得 V^i变大，因为其他人的策略都在发生变化

2. Convergence

收敛的意思是无法通过改进策略policy来获得更大的期望回报。如果所有的agent都找不到更好的策略就说明已经收敛，可以终止训练

2.1 Single-Agent Policy Learning

- Policy network: $\pi(a \mid s; \boldsymbol{\theta})$.
- State-value function: $V(s; \boldsymbol{\theta})$.
- $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S; \boldsymbol{\theta})]$ evaluates how good the policy is.
- Learn the policy network's parameter, $\boldsymbol{\theta}$, by

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}).$$

- **Convergence:** $J(\boldsymbol{\theta})$ stops increasing.

Multi-Agent Policy Learning

Nash Equilibrium

- While all the other agents' policy remain the same, the i-th agent cannot get better expected return by changing its own policy.

  当其余所有**agents**都不改变策略的情况下，一个agent单独改变策略，不会让自己获得更高的回报

- Every agent is playing a best-response to the other agents' policies.

  有多个**agents**在参与博弈，一个agent制定策略的时候要考虑到其他各方的策略

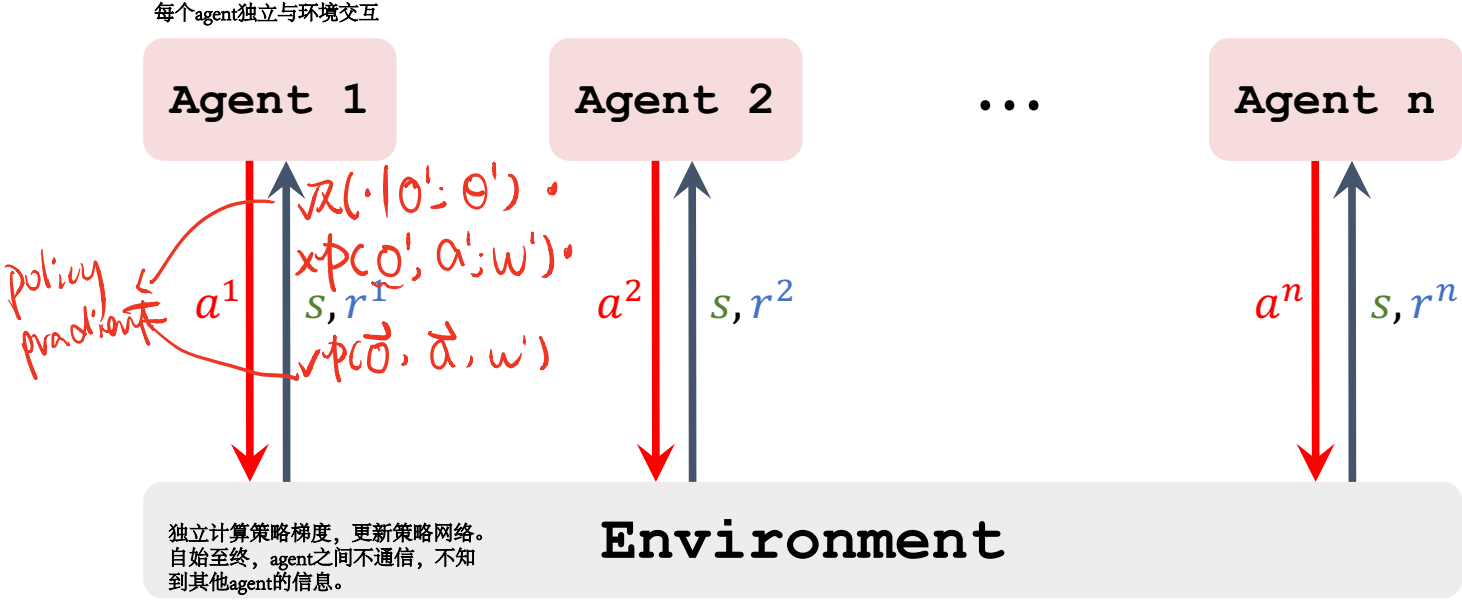
- Nash equilibrium indicates convergence because no one has any incentive to deviate.

  在纳什均衡的情况下，每一个agent 都在以最优的方式 来应对其他各方的策略。如果所有agents都是理性的，那么在纳什均衡的情况下，谁也没有动机去改变自己的策略，因为改变策略不会增加自己的收益。这样就达到了一种平衡状态。所有agents都找不到更好的策略，这种平衡状态就算是收敛。多个agents会让强化学习变得更困难，直接套用single-agent算法效果不好，可能会不收敛。

3. Difficulty of MARL

Single-Agent Policy Gradient for MARL

这就是对上一节 fully decentralized architecture效果不好的解释



但是，这样做的问题是：利用actor-critic 方法来得到的state-value function并不是真实的state-value function。真实的state-value function要考虑其他各个policy theta对其的影响。但是在各个agent独立的情况下，我们没有考虑，如下页所示。

Single-Agent Policy Gradient for MARL

- The i-th agent's policy network: $\pi(a^i \mid s; \boldsymbol{\theta}^i)$.
- The i-th agent's state-value function: $V^i(s; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)$.
- Objective function: $J^i(\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n) = \mathbb{E}_S\left[V^i(S; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)\right]$.
- Learn the policy network's parameter, $\boldsymbol{\theta}^i$, by

$$\max_{\boldsymbol{\theta}^i} J^i(\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n).$$

这是理想情况下的state-value function。如果只用自己的观测，得到的不会是这个吧？

不过，这里想要强调的是，即使得到了，在这种去中心化的设计中，训练效果也不好。

Single-Agent Policy Gradient for MARL

目标函数各不相同

- The 1^{st} agent solves: $\max_{\boldsymbol{\theta}^1} J^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^n)$.
- The 2^{nd} agent solves: $\max_{\boldsymbol{\theta}^2} J^2(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^n)$.

⋮

- The n^{th} agent solves: $\max_{\boldsymbol{\theta}^n} J^n(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^n)$.

It may not converge…

Single-Agent Policy Gradient for MARL

What is wrong?

- The i-th agent found $\boldsymbol{\theta}_{\star}^{i} = \operatorname{argmax} J^{i}(\boldsymbol{\theta}^{1}, \cdots, \boldsymbol{\theta}^{n})$.

  假设agent i已经无法通过改变theta i来增大收益了 $\boldsymbol{\theta}^{i}$
- Now, another agent changes its policy.
- So $\boldsymbol{\theta}_{\star}^{i}$ is no longer the best policy of the i-th agent. The i-th agent has to find a new $\boldsymbol{\theta}^{i}$.
- The other agents' objective functions will change, and therefore they will change their policies…

  所有的agents都在不停的改变自身的策略，永远都不收敛。

4. Summary

Multi-Agent Reinforcement Learning (MARL)

- There are $n > 1$ agents in the system.
- The agents are usually not independent.
  - Every agent's action can affect the next state.
  - Thus, every agent can affect all the other agents.
- Unless the agents are independent of each other, single-agent RL methods do not work well for MARL.

Settings of MARL

1. **Fully cooperative**, e.g., industrial robots.
2. **Fully competitive**, e.g., predator and prey.
3. **Mixed cooperative & competitive**, e.g., robotic soccer.
4. **Self-interested**, e.g., automated trading systems.

Convergence

agent只能改变自己的策略，不能改变别人的策略。他想获得更高的收益，
唯一的办法就是改变自己的策略

- **Convergence:** No agent can get better expected return by improving its own policy.
- If there is only one agent, convergence means the objective function does not increase any more.
- If there are multiple agents, Nash equilibrium means convergence.

在这种平衡的状态下，大家都没有动机去改变自己的策略。

Stationary VS Non-stationary

- Consider single-agent setting.
- Stationary environment requires state transition be fixed throughout.
  - State transition: $p(s'|s, a)$.
  - Given s and a, the probability distribution of the next state s' is always the same.
- All the single-agent RL methods we have learned so far require stationary environment.

Stationary VS Non-stationary

- Consider multi-agent setting.
- Stationary environment requires state transition be fixed throughout.
  - State transition: $p(s'|s, a^1, \cdots, a^n)$.
  - Given s and $a^1, \cdots, a^n$, the probability distribution of the next state s' is always the same.

Stationary VS Non-stationary

- Consider multi-agent setting.
- Stationary environment requires state transition be fixed throughout.
- The environment is typically stationary.
- However, from any single agent's perspective, the environment is non-stationary.
  - p depends not only on s and a^i, but also on the other agents' actions.
  - If the i-th agent knows only s and a^i, then from its perspective, the state transition is not fixed.

Stationary VS Non-stationary

- Consider multi-agent setting.
- Stationary environment requires state transition be fixed throughout.
- The environment is typically stationary.
- However, from any single agent's perspective, the environment is non-stationary.
- Thus, the single-agent RL method we have learned are not applicable.