综述 Multi-Agent Reinforcement Learning

MARL 算法可分为 fully cooperative → 合作优化总收益

fully competitive → 通常收益和为0

和 a mix of the two → 两者都有

Markov forms

其他 extentive forms

MARL 的三个特点...

① Multi dimensional

多维度无对齐，又能以评估即定义收益

② 每个 agent 只看重自己的收益 → Non stationary

③ 联合 action space 随 agent 数量指数增加

Who knows what.

Background

2.1 Single-Agent RL — 常由 MDP 建模

定义 ① MDP 定义为 (S, A, P, R, γ)

- S: state
- A: action
- P: $S \times A \to \Delta S$ transition probability
- R: reward
- γ: discount factor $[0, 1]$

$s_t \xrightarrow{a_t} s_{t+1}$ $s_{t+1} \sim P(\cdot | s_t, a_t)$ → agent 得到 reward 为 $R(s_t, a_t, s_{t+1})$

Core Goals: 寻找 policy $\pi: S \to \Delta(A)$ 即 S 到 A 分布的 mapping

$$a_t \sim \pi(\cdot|s_t)$$

方法：最大化 $E\left(\sum_{t\geq 0} \gamma^t R(s_t, a_t, s_{t+1}) \mid a_t \sim \pi(\cdot|s_t), s_0\right)$

State-action function / Q function

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

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

可由 BP 和 DP 求得 optimal policy.

分类 (1) **Value - Based Methods**

为寻找最好的 Q-function

policy 则是由 Q function 经过 greedy 得到

update 公式

$$\hat{Q}(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left[r + \gamma \max_{a'} \hat{Q}(s',a')\right]$$

on-policy 有限决策限 ↑ space　learning rate　reward

SARSA, DQN 只是 target 不同 DQN 用 NN 估计 a'

另一种 algorithm 为 MCTS 蒙特卡洛树搜索, UCB

还有一种为用已知 Policy 估计 value function, TD 算法 时序差分

policy evaluation

(2) **Policy Based Methods** 直接在 Policy Space

用参数化函数如 NN 估计 π　　寻找最

$\pi(\cdot|s) \approx \pi_\theta(\cdot|s)$　用梯度更新的为 Policy Gradient

PG 的 closed-form ↴

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

J_θ 和 Q_{π_θ} 是 reward 和 在 Policy π_θ 下的 Q function 的期望

PG 的常用算法：REINFORCE，G(PO)MDP，actor-critic

非 gradient 方法，有用 PPO，TRPO 和 soft actor-critic

总的来说 gradient based 比 value based 要好收敛

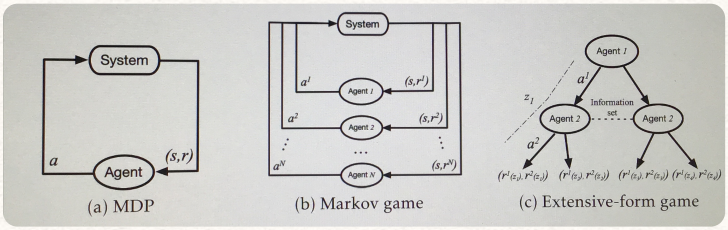
2.2 Multi-Agent RL Framework

系统中每个 agent 状态 state 和 reward 的更新都受其他所有 agent 的联合动作

每个 agent 有自己要优化的长期收益，为其他所有 agents 的函数

可将 MARL 分为两个 frameworks：Markov/stochastic games；extensive-form games

3个对比



① Markov/Stochastic Games　action　完全信息静态博弈

此时所有 agents 同时选择 a^i after 观察状态 s，得到独立的 individual reward r^i

② Extensive-form games 可用于非完美信息博弈

此时 agents 选择 a^i 在 game 结束收到 $r^i(z)$

↳ 完全/非完全静态博弈

① MG 的部分定义

Markov Games 定义为一个 tuple $(N, S, \{A^i\}_{i\in N}, P, \{R^i\}_{i\in N}, \gamma)$

- N: N agents
- S: state space observed by all agents
- A^i: agent i 的 action set
- P: $S\times A \to \Delta(S)$, 全联合 $A := A^1\times\cdots\times A^N$, 由联合 action 从任意 $s\in S$ 到 $s'\in S$ 的转移概率
- R^i: $S\times A\times S\to R$
- γ: discount factor

> At time t, each agent $i \in \mathcal{N}$ executes an action a_t^i, according to the system state s_t. The system then transitions to state s_{t+1}, and rewards each agent i by $R^i(s_t, a_t, s_{t+1})$. The goal of agent i is to optimize its own long-term reward, by finding the policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ such that $a_t^i \sim \pi^i(\cdot|s_t)$. As a consequence, the value-function $V^i : \mathcal{S} \to \mathbb{R}$ of agent i becomes a function of the joint policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ defined as $\pi(a|s) := \prod_{i\in\mathcal{N}} \pi^i(a^i|s)$. In particular, for any joint policy π and state $s \in \mathcal{S}$,
>
> $$V^i_{\pi^i,\pi^{-i}}(s) := \mathbb{E}\left[\sum_{t\geq 0}\gamma^t R^i(s_t, a_t, s_{t+1})\,\middle|\, a_t^i \sim \pi^i(\cdot|s_t), s_0 = s\right], \tag{2.3}$$
>
> where $-i$ represents the indices of all agents in $\mathcal{N}$ except agent i. Hence, the solution concept of MG deviates from that of MDP, since the *optimal* performance of each agent is controlled not only by its own policy, but also the choices of all other players of the game.
>
> 7

常见解的形式 Nash equilibrium (NE) 纳什平衡点定义如下

定义: MG 的纳什平衡点 求得最优联合策略 $\pi^* = (\pi^{1*}, \cdots, \pi^{N,*})$

$$V^i_{\pi^{i*},\pi^{-i*}}(s) \geq V^i_{\pi^i,\pi^{-i,*}}(s) \quad \text{for any } \pi$$

基于 MG 的九种方法、

(1) Cooperative Setting 此时所有 agent 有共同的 reward 即 $R^1 = R^2 = \cdots = R^N = R$

↳ 称为 MMDPs in AI 社区

Multiagent MDPs

Markov teams / teams Markov games in Game Theory 社区

此设置 Q 和 Value function 对每个 agent 都相同，可以用 Single agent 得 Optimal Policy 纳什平衡点

更 General 一点 每个 agent 的 reward 不同，称为 team-average reward 即每个 agent 有自己的 reward function 但最终 Goal 为优化 long term average reward

使 decentralized MARL 算法应用 $\bar{R}(s,a,s') := N^{-1}\sum_{i\in N} R^i(s,a,s')$ for any (s,a,s')

这使得 MARL 的 communication 和 communication efficient 也很必要.

(2) Competitive Setting.

在 MG 中完全 Competitive 的建模为 zero-sum Markov games

即 $\sum_{i\in N} R^i(s,a,s') = 0$ for any (s,a,s')

常指 2 个 agents 竞争，一个 agent 的 reward 就是另一个的 loss

Goal：优化 最差的长期收益

(3) Mixed setting General sum game setting Open AI

此时 goal 和 agents 间的联系没有限制.

每个 agent 只关心自己的收益

2.2 Extensive - Form Games

Markov Games 无法处理 非完美信息 情况，所以用 Extensive-form game