多智能体系统与强化学习

主讲人：高阳、杨林、杨天培

https://reinforcement-learning-2025.github.io/

第八讲：多智能体系统与博弈论

主讲人：高阳

大 纲

多智能体系统

博弈论和均衡

经典多智能体学习算法

大 纲

多智能体系统

博弈论和均衡

经典多智能体学习算法

从田忌赛马谈起

田忌赛马的策略

- 齐王以“上马、中马、下马”出赛；
- 齐王的马优于田忌的马；
- 每一场胜者赢黄金100两，负者输黄金100两；
- 田忌以何种次序出马呢？

	f_1	f_2	f_3	f_4	f_5	f_6
	<上，中，下>	<上，下，中>	<中，上，下>	<中，下，上>	<下，中，上>	<下，上，中>
齐王 f_1 <上，中，下>	**-300**	**-100**	**-100**	**-100**	**-100**	**+100**

如果齐王的策略也是变化的？

多智能体系统视角下的投票

- **投票机制：** 每个agent给予投票机输入，投票机产生结果作为对所有agent的输出；
- 假设：有三个候选人，三个投票人

如何设计投票机制使结果公平？

	1	2	3
V_1	A	B	C
V_2	B	C	A
V_3	C	A	B

- 观察系统：比较候选人A和B，A优于B；比较候选人B和C，B优于C；比较候选人C和A，C优于A
- $A \succ B \succ C \succ A$？？？

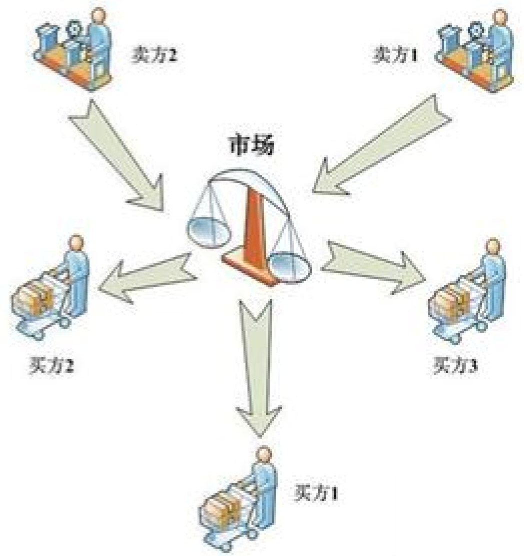
多智能体系统视角下的拍卖

➢拍卖机制

- ➢存在拍卖者和竞拍者两种agent角色。
- ➢拍卖者尽可能高价卖出自己的物品，竞拍者尽可能使自己以低价获得物品。
- ➢拍卖机制的设计，尽可能使拍卖者增加自己的收益。

➢拍卖协议

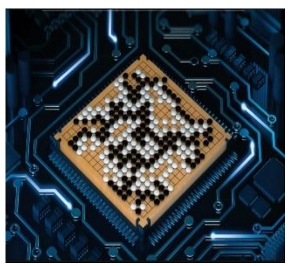
- ➢英格兰拍卖 first-price open-cry
- ➢密封拍卖 first-price sealed-bid
- ➢荷兰式拍卖
- ➢Vickery拍卖second-price sealed-bid



多智能体系统分类和特点

➢ **多智能体系统类型：**按智能体决策目标分为合作型、竞争型及混合型。


- ✓ 任务类型：合作型多智能体系统
- ✓ 问题空间：智能体学习目标一致
- ✓ 决策标准：提高学习速度，获得全局最优解

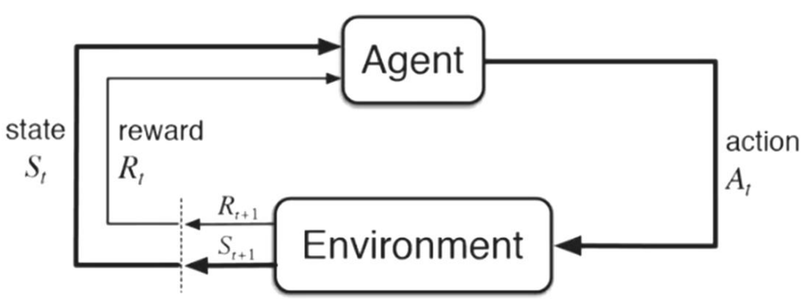



- ✓ 任务类型：竞争型多智能体系统
- ✓ 问题空间：智能体学习目标对立
- ✓ 决策标准：不遗憾 (No-regret)

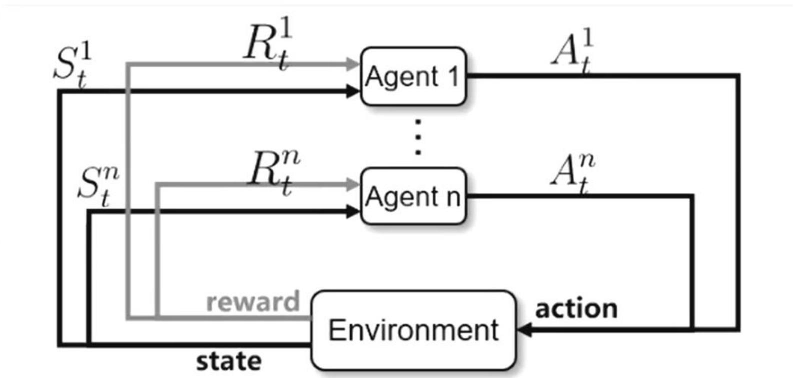

- ✓ 任务类型：混合型多智能体系统
- ✓ 问题空间：智能体学习目标相异（非对立）
- ✓ 决策标准：收敛、均衡解

单智能体 vs 多智能体系统

- 单智能体(Single-agent)
  - 只有一个智能体在与环境交互
  - 智能体在观察、测量环境的动态性，并进行最优决策



- 多智能体系统(Multi-agent system)
  - 有多个智能体在与环境交互
  - 智能体在观察、测量环境的动态性的同时，协同其他智能体的行为，进行最优决策



从奖赏的视角看多智能体系统

多智能体系统分类

合作型

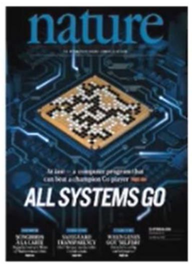
（智能物流）（无人集群）

$$\mathcal{R} = \mathcal{R}^i = \mathcal{R}^{-i}, \forall i \in \mathcal{N}$$

所有玩家共享同一奖励

协作式地最大化整体目标

对抗型



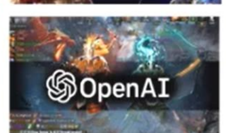
（棋牌类游戏）

$$\mathcal{R}^i = -\mathcal{R}^{-i}, \forall i \in \mathcal{N}$$

玩家之间奖励相反

竞争式地最大化自身目标

合作-对抗混合型



（多单位、多人对抗游戏）

$$\mathcal{R}_{\text{red}} = \mathcal{R}^{i_{\text{red}}}, \forall i_{\text{red}} \in \mathcal{N}_{\text{red}}$$

$$\mathcal{R}_{\text{blue}} = \mathcal{R}^{i_{\text{blue}}}, \forall i_{\text{blue}} \in \mathcal{N}_{\text{blue}}$$

$$\mathcal{R}_{\text{red}} = -\mathcal{R}_{\text{blue}}$$

组内玩家共享奖励

组间奖励相反

大 纲

多智能体系统

博弈论和均衡

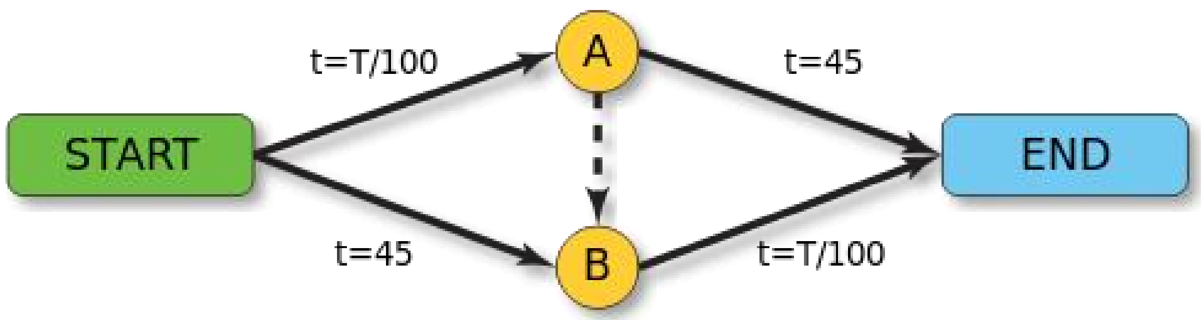
经典多智能体学习算法

囚徒困境

		囚犯B	
		坦白	抗拒
囚犯A	坦白	(-5,-5)	(0,-10)
	抗拒	(-10,0)	(-1,-1)

- 每个囚徒如果只考虑自身的利益，则会选择‘坦白’行为；
- 而囚徒困境的最优策略是双方都选择‘抗拒’行为。

布雷斯悖论 Braess's paradox



- 考虑上图中的交通网，有4000辆车打算在其中路上通行。其中边上的数值表示通行时间，T表示边上的车辆数目。
- A到B的近路不存在
  - 2000辆从起点到A到终点，2000辆从起点到B到终点，65分钟
- A到B存在一条通行时间接近于0的近路
  - 所有司机都会选择从起点到A到B到终点，80分钟
  - 假如约定好不走近路？

博弈的定义

- 玩家集合

$$N = \{1,2, \dots n\}$$

- 策略集合

$$A_1,\ A_2\ \dots, A_n$$

- 收益函数

$$r_1 : A_1 \times A_2 \times \cdots \times A_n \rightarrow \mathcal{R}$$

$$r_2 : A_1 \times A_2 \times \cdots \times A_n \rightarrow \mathcal{R}$$

$$. \quad . \quad .$$

$$r_n : A_1 \times A_2 \times \cdots \times A_n \rightarrow \mathcal{R}$$

最优策略

- 在多智能体环境中，四种不同的最优策略定义
  - 社会福利（Social Welfare）
    - 最大化所有参与者的收益和
  - 帕里托优（Pareto Efficiency）
  - 纳什均衡（Nash Equilibrium）
  - 优超（Dominant）
    - 不依赖其他参与者

帕里托优

- 一个方案x是帕利脱最优，当且仅当不存在另一个方案 x’满足
  - $\exists$agent ag: $ut_{ag}(x') > ut_{ag}(x)$
  - $\forall$agent ag': $ut_{ag'}(x') \geq ut_{ag'}(x)$
- 帕利脱最优：不考虑跨Agent效益比较的情况下满足一个全局最优
- 帕利脱改善：在不减少一方利益的同时，通过改变现有的资源配置而提高另一方的利益
- 社会福利是帕利脱最优的一个子集
  - 一个agent要想提高自己的利益，必然存在其他agent的利益受损

纳什均衡

- Agent 的策略依赖于其他agent
  - 如果 $S^*_A=< S^*_1, S^*_2,\ldots\ldots, S^*_{|A|}>$ 为纳什均衡策略, 当且仅当对agent i：S^*_i 对于 agent i 是最优策略当其他agent选择以下策略时 $< S^*_1, S^*_2,\ldots, S^*_{i-1}, S^*_{i+1} \ldots, S^*_{|A|}>$
- 没有参与者可以单边改变策略而增加自己的收益！(请仔细理解)
- 问题
  - 无纯Nash均衡解
  - 多个Nash均衡解

无纯策略Nash均衡解

		女孩		
		剪刀	石头	布
男孩	剪刀	(0,0)	(-1,1)	(1,-1)
	石头	(1,-1)	(0,0)	(-1,1)
	布	(-1,1)	(1,-1)	(0,0)

纳什均衡解是什么？

多个Nash均衡解

		女孩	
		球赛	电影
男孩	球赛	(2,1)	(0,0)
	电影	(0,0)	(1,2)

- 恋爱博弈问题
  - 男孩喜欢看比赛
  - 女孩喜欢看电影
  - 两人都希望一起度过周末

纳什均衡解是什么？

结婚后…

		女孩	
		球赛	电影
男孩	球赛	(2,0)	(3,3)
	电影	(-3,-3)	(0,2)

- 恋爱博弈问题
  - 男孩喜欢看比赛
  - 女孩喜欢看电影
  - 两人倾向于周末分道扬镳…

不同准则下的最优策略

		囚犯B	
		坦白	抗拒
囚犯A	坦白	(-5,-5)	(0,-10)
	抗拒	(-10,0)	(-1,-1)

- 社会福利：<抗拒，抗拒>
- 帕里托优：除了<坦白，坦白>之外的其他情况
- 纳什均衡：<坦白，坦白>
- 优超： <坦白，坦白>

非共享支付的博弈

(A,B)	b_1	b_2
a_1	(0, 0)	(1, 1)
a_2	(2, 2)	(3, 3)

Original Game Matrix

值表在多智能体间共享，为公共知识

- ✓ 透露太多信息，不安全
- ✓ 适用范围有限，分布决策环境中难适用
- ✓ 空间复杂度高

信息分布环境下的博弈形式

Distributed Game Matrix

(A,B)	b_1	b_2
a_1	(0, ?)	(1, ?)
a_2	(2, ?)	(3, ?)

(A,B)	b_1	b_2
a_1	(?, 0)	(?, 1)
a_2	(?, 2)	(?, 3)

非共享支付的博弈

分布式决策下的博弈，绝对理性
的纳什均衡未必最适用

(D, D) 比 (C, C) 更
有利

智能体A

A	Confess	Deny
Confess	-9	0
Deny	-10	-1

(D, D) 比 (C, C) 更
有利

智能体B

B	Confess	Deny
Confess	-9	-10
Deny	0	-1

策略组(D,D) 帕里托优超(Pareto dominates)于纳什均衡策略(C,C)。

一个有趣的博弈

(A,B)	b_1	b_2	b_3
a_1	**(20,40)**	(4,22)	**(29,30)**
a_2	(18,9)	**(36,19)**	(7,4)
a_3	(17,26)	(15,38)	**(27,38)**

$(\boldsymbol{a_1}, \boldsymbol{b_3})$: $(\mathbf{29}, \mathbf{30})$ $(\boldsymbol{a_3}, \boldsymbol{b_3})$: $(\mathbf{27}, \mathbf{38})$

分布式决策下的博弈矩阵

A	b_1	b_2	b_3
a_1	20	4	29
a_2	18	36	7
a_3	17	15	27

智能体A

B	b_1	b_2	b_3
a_1	40	22	30
a_2	9	19	4
a_3	26	38	38

智能体B

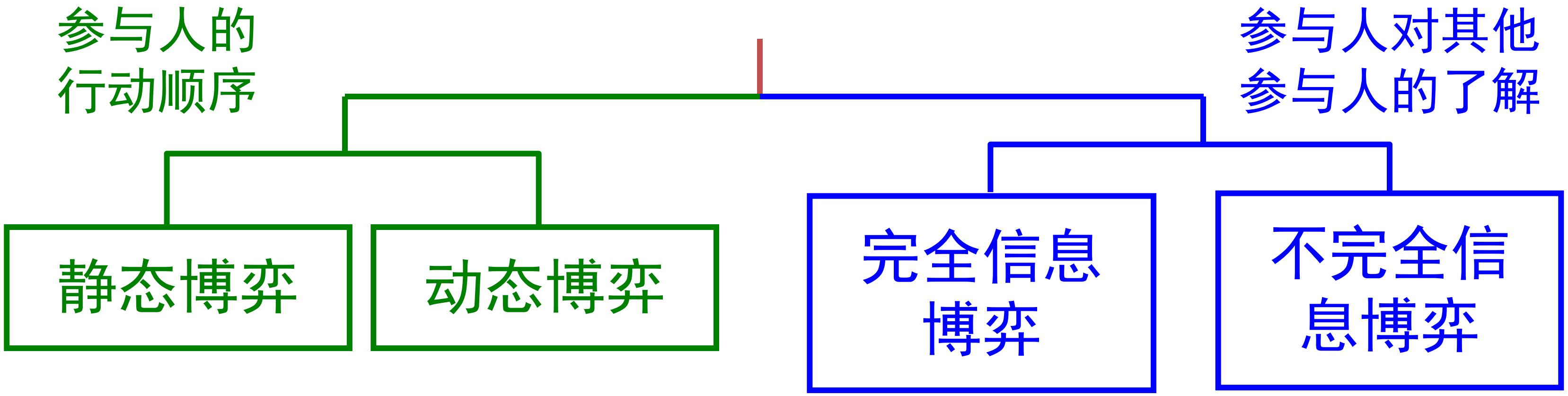
(a_1, b_3) (a_3, b_3) (a_1, b_1)

(a_1, b_3) (a_3, b_3) (a_2, b_2)

博弈的分类

- 根据参与人是否合作
  - 合作博弈
  - 非合作博弈
- 根据参与人的多少
  - 两人博弈
  - 多人博弈
- 根据博弈结果
  - 零和博弈
  - 常和博弈
  - 变和博弈
- 根据行动的先后次序
  - 静态博弈
  - 动态博弈
- 根据参与人对他人的信息的获得差异
  - 完全信息博弈
  - 不完全信息博弈

博弈



信息 \ 行动顺序	静　　态	动　　态
完全信息	纳什均衡 （纳什）	子博弈精练纳什均衡 （泽尔腾）
不完全信息	贝叶斯纳什均衡 （海萨尼）	精练贝叶斯纳什均衡 （泽尔腾等）

特殊的博弈：合作与竞争

- 合作博弈(Cooperative Game)
  - ✓ $r_1(a) = r_2(a) = \cdots = r_n(a), \forall a \in A_1 \times A_2 \times \cdots \times A_n$

	左	右
左	**1, 1**	**0, 0**
右	**0, 0**	**1, 1**

- 竞争博弈(Competitive Game)
  - ✓ $\sum_{i=1}^{n} r_i(a)_1 = C, \forall a \in A_1 \times A_2 \times \cdots \times A_n$

	石头	剪子	布
石头	**0, 0**	**1, -1**	**-1, 1**
剪子	**-1, 1**	**0, 0**	**1, -1**
布	**1, -1**	**-1, 1**	**0, 0**

回顾：纳什均衡(Nash Equilibrium, NE)

描述

- 任何玩家都不能通过独自改变策略而获益的策略组合，即所有玩家均处于最佳应对的策略组合。

数学定义

- 给定一个策略组合 $a = (a_1, a_2, \ldots, a_n) \in A_1 \times A_2 \times \cdots \times A_n$ ，
- 若 $r_1(a_1, a_2, \ldots, a_n) \geq r_1(a_1', a_2, \ldots, a_n), \forall a_1' \in A_1$
  且 $r_2(a_1, a_2, \ldots, a_n) \geq r_1(a_1, a_2{}', \ldots, a_n), \forall a_2' \in A_2$
  且 $\cdots$ 且 $r_n(a_1, a_2, \ldots, a_n) \geq r_n(a_1, a_2, \ldots, a_n'), \forall a_n' \in A_n$
- 那么策略组合 a 是一个纳什均衡。

混合纳什均衡(Mixed Strategy NE)

- 描述
  - 混合策略：一个概率分布 $(p_1, p_2, \ldots, p_n), p_i$ 表示选择动作 i 的概率
  - 混合策略纳什均衡：一个混合策略组合，任何玩家都不能通过独自改变混合策略而使自身期望收益提高
- 例如：石头剪子布博弈
  - 玩家1：(1/3,1/3,1/3)
  - 玩家2：(1/3,1/3,1/3)

	石头	剪子	布
石头	0, 0	1, -1	-1, 1
剪子	-1, 1	0, 0	1, -1
布	1, -1	-1, 1	0, 0

- 定理：任一博弈，必存在一个混合策略纳什博弈

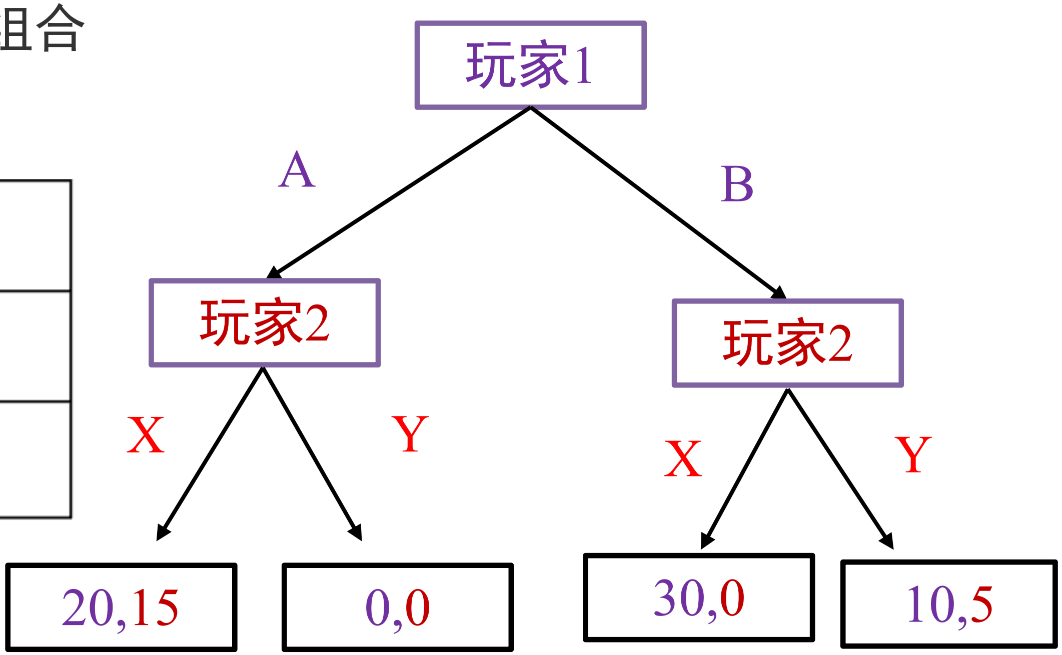
Stackelberg均衡

➢ 描述

- ➢ 改变了一个基本假设：玩家的行动有先后次序
- ➢ 定义：在规定了行动先后次序的情况下，任何玩家都不能通过独自改变策略而获益的策略组合

	X	Y
A	20, 15	0, 0
B	30, 0	10, 5

纳什均衡



Stackelberg均衡

大 纲

多智能体系统

博弈论和均衡

经典多智能体学习算法

两种简化方法

将对手作为环境的一部分

- 独立学习
  - 在合作场景适用，效率较低
- 对手建模
  - 适用于对手策略稳定的场景
- 多智能体强化学习算法设计目标
  - 合理性(rationality): 是指在对手使用一个恒定策略的情况下，当前智能体能够学习并收敛到一个相对于对手策略的最优策略。
  - 收敛性(convergence): 是指在其他智能体也使用学习算法时，当前智能体能够学习并收敛到一个稳定的策略。通常情况下，收敛性针对系统中的所有的智能体使用相同的学习算法。

马尔科夫博弈(Markov Game)

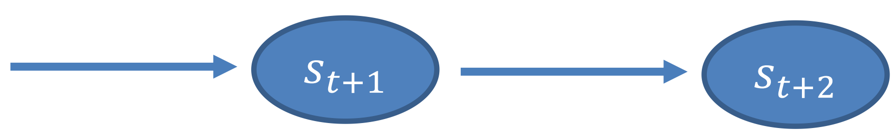
- 数学模型$\langle N,\ \mathcal{S},\ \mathcal{A},\ T,\ R,\ \gamma\rangle$
  - 稳定联合策略下值函数满足bellman方程

$$V_{\pi}^{i}(s) = \mathbb{E}_{\boldsymbol{a}\sim\pi(s)}\left[r(s,\boldsymbol{a}) + \gamma \sum_{s'\sim S} P(s'|s,\boldsymbol{a})\, V_{\pi}^{i}(s')\right]$$

  - 状态转移概率受环境和智能体共同影响
  - 以纳什均衡为目标:

$$\forall s \in \mathcal{S}, i \in N,\ \forall \pi^{i} \in \Pi,\ V_{\pi}^{i}(s) \geq V_{\pi^{i},\pi^{-i}}^{i}(s)$$

s_t		Agent b	
		b_1	b_2
Agent a	a_1	$Q_a^t(s_t,a_1,b_1), Q_b^t(s_t,a_1,b_1)$	…
	a_2	…	…



Minimax-Q

- 适用场景
  - 零和博弈：一个智能体的收益完全是另一个智能体损失的场景
- 更新公式

$$Q_i(s,a_i,a_{-i}) \leftarrow Q_i(s,a_i,a_{-i}) + \alpha[r_i + \gamma V_i(s') - Q_i(s,a_i,a_{-i})]$$

$$V_i^*(s) = max_{\pi_i(s,\cdot)} min_{a_{-i}\in A_{-i}} \sum_{a_i\in A_i} Q_i^*(s,a_i,a_{-i})\pi_i(s,a_i)\,,\ i = 1,2$$

- 优缺点
  - 收敛性、对抗性、学习性
  - 计算复杂度高、非最优性、依赖充分探索

Minimax-Q

➢ 算法伪代码

Algorithm 1 Minimax-Q 算法

初始化：
for 所有 s, a_i, a_{-i} **do**
    $Q_i(s, a_i, a_{-i}) \leftarrow 0$
end for
设置 $\alpha, \gamma, \pi_i(s, a_i)$
主循环：
for 每个回合 **do**
    初始化 s
    while s 非终止 **do**
        选择 $a_i \sim \pi_i(s), a_{-i} \sim \pi_{-i}(s)$
        执行 (a_i, a_{-i})，得 r_i, s'
        $V_i(s') = \max_{\pi_i(s')} \min_{a_{-i}} \sum_{a_i} Q_i(s', a_i, a_{-i})\pi_i(s', a_i)$
        $Q_i(s, a_i, a_{-i}) \leftarrow Q_i(s, a_i, a_{-i}) + \alpha[r_i + \gamma V_i(s') - Q_i(s, a_i, a_{-i})]$
        更新 $\pi_i(s)$ 最大化 $\min_{a_{-i}} \sum_{a_i} Q_i(s, a_i, a_{-i})\pi_i(s, a_i)$
        $s \leftarrow s'$
    end while
end for
返回： Q_i, π_i

Nash-Q

- 给定联合策略 $\boldsymbol{\pi}$，其智能体 i 的价值函数为

$$V_i^{\boldsymbol{\pi}}(s) = V_i(s; \boldsymbol{\pi}) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{E}_{\boldsymbol{\pi},p}[r_i^t | s_0 = s]$$

- 因此，针对智能体 i 优化 $V_i^{\boldsymbol{\pi}}(s)$ 依赖于联合策略 $\boldsymbol{\pi}$

- 在随机博弈中的纳什均衡(Nash Equilibrium)可以由一个联合策略 $\boldsymbol{\pi}^*$ 来表示

$$\boldsymbol{\pi}^* = [\pi_1^*, \pi_2^*, \ldots, \pi_N^*]$$

如果没有任何一个智能体有动机去进一步修改它的策略

$$V_i^{\boldsymbol{\pi}^*}(s) = V_i(s; \boldsymbol{\pi}^*) = V_i(s; \pi_i^*, \boldsymbol{\pi}_{-i}^*) \geq V_i(s; \pi_i, \boldsymbol{\pi}_{-i}^*) \text{ for } \forall \pi_i$$

$$\boldsymbol{\pi}_{-i}^* = [\pi_1^*, \pi_2^*, \ldots, \pi_{i-1}^*, \pi_{i+1}^*, \ldots, \pi_N^*]$$

Nash-Q

- 给定纳什策略$\boldsymbol{\pi}^*$，其纳什价值函数为

$$V_{\text{Nash}}(s) = \left[V_1^{\boldsymbol{\pi}^*}(s), V_2^{\boldsymbol{\pi}^*}(s), \dots, V_N^{\boldsymbol{\pi}^*}(s)\right]$$

- 考虑到智能体 i 在状态行动对 $(s, \boldsymbol{a})$ 下的价值为

$$Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r_i(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim p}[V_i^{\boldsymbol{\pi}}(s')]$$

- 纳什Q学习不断进行以下两步操作，直到收敛
  - 基于当前每个状态的Q值表，求解纳什均衡$\boldsymbol{\pi}^*$，以及每个状态的V_{Nash}
  - 基于纳什均衡价值V_{Nash}，使用上式更新Q值表

Nash-Q

- 给定纳什策略$\boldsymbol{\pi}^*$，其纳什价值函数为

$$V_{\text{Nash}}(s) = \left[V_1^{\boldsymbol{\pi}^*}(s), V_2^{\boldsymbol{\pi}^*}(s), \dots, V_N^{\boldsymbol{\pi}^*}(s)\right]$$

- 考虑到智能体 i 在状态行动对 $(s, \boldsymbol{a})$ 下的价值为

$$Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r_i(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim p}[V_i^{\boldsymbol{\pi}}(s')]$$

- 作为一种完全中心化方法，Nash-Q学习仍然有以下缺点
  - 非常高的计算复杂度
  - 无法处理非合作的博弈场景

Nash-Q

➢ 算法伪代码

算法： Nash-Q学习算法伪代码

Initialize： $Q_i(s, a_1, \cdots, a_n) = 0,\ \forall a_i \in A_i$

for 迭代次数$1 \rightarrow E$ **do**

  第i个智能体根据当前状态s采用探索策略得到a_i并执行

  得到下一个状态s'以及智能体i观测所有智能体的奖励$r_1, \cdots, r_n$，并且观测所有智能体在状态s执行的所有动作$a_1, \cdots, a_n$

  更新$Q_i(s, a_1, \cdots, a_n)$:

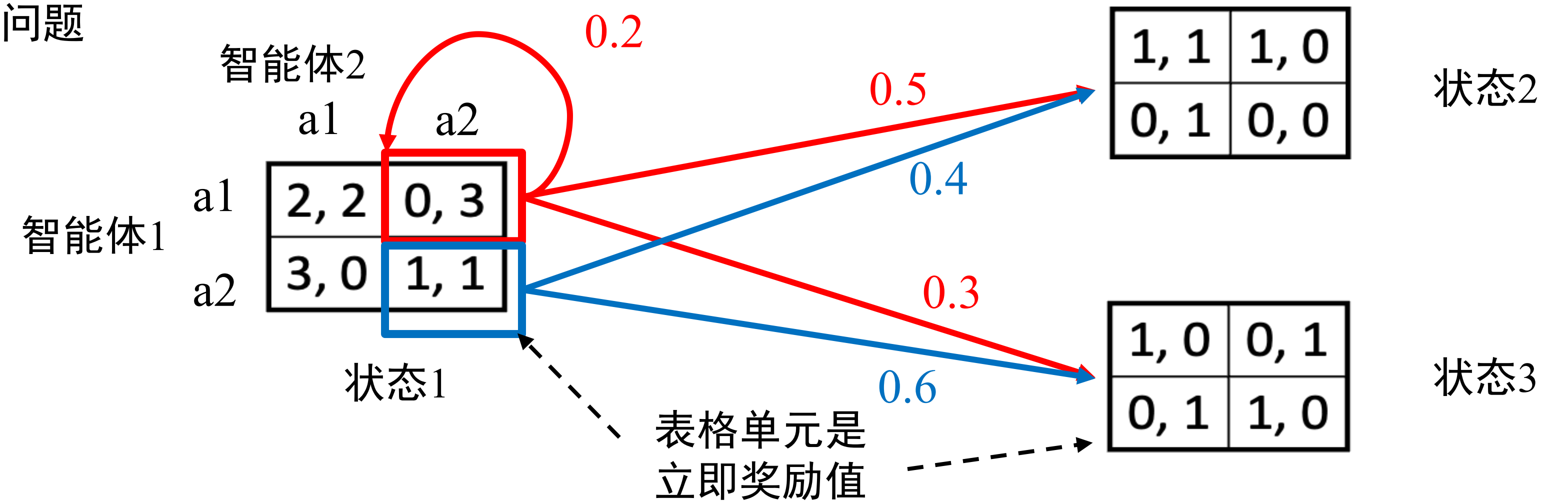
$$Q_i(s, a_1, \cdots, a_n) \leftarrow Q_i(s, a_1, \cdots, a_n) + \alpha[r_i + \gamma \mathrm{Nash}Q_i(s') - Q_i(s, a_1, \cdots, a_n)]$$

  利用二次规划求解状态s处的纳什均衡策略并更新Nash$Q_i(s)$和$\pi_i(s, \cdot)$

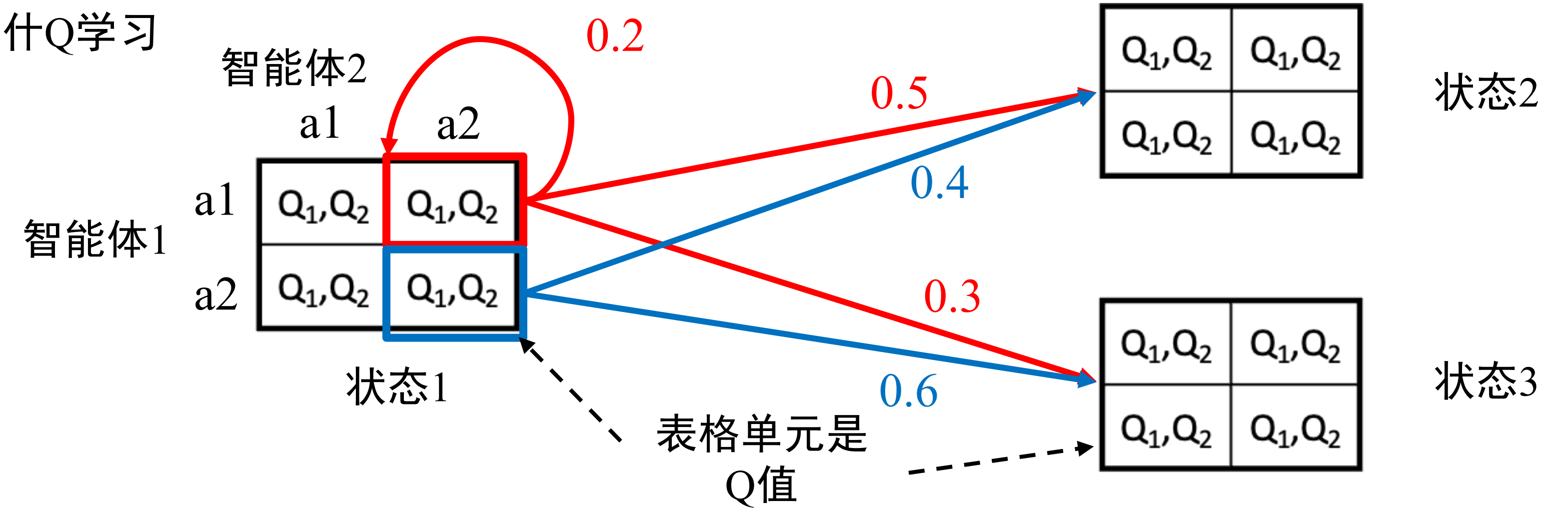
end for

Nash-Q

原问题



纳什Q学习



Friend-or-Foe Q

➢ 适用场景

- 一般和博弈：对一个智能体i，将其他所有智能体分为两组，一组是friend，一组是foe。

➢ 更新公式

$$V_i(s) = \max_{\pi_1(s),\dots,\pi_{n_1}(s)} \min_{o_1,\dots,o_{n_2} \in O_1\times\cdots\times O_{n_2}} \sum_{a_1,\dots,a_{n_1} \in A_1\times\cdots\times A_{n_1}} Q_i(s, a_1, \dots, a_{n_1}, o_1, \dots, o_{n_2}) \pi_1(s, a_1) \dots \pi_{n_1}(s, a_{n_1})$$

$$Q_i(s, a_1, \dots, a_{n_1}, o_1, \dots, o_{n_2}) \leftarrow Q_i(s, a_1, \dots, a_{n_1}, o_1, \dots, o_{n_2}) + \alpha[r_i + \gamma V_i(s') - Q_i(s, a_1, \dots, a_{n_1}, o_1, \dots, o_{n_2})]$$

Friend-or-Foe Q

➢ 算法伪代码

Algorithm 2 Friend-or-Foe Q-Learning (FFQ) 算法

初始化：

初始化 $V_i(s)=0$，$Q_i(s,a_1,\ldots,a_{n_1},o_1,\ldots,o_{n_2})=0$，表示智能体 i 所有的 friend 动作 $(a_1,\ldots,a_{n_1})$ 和 foe 动作 $(o_1,\ldots,o_{n_2})$。

for 每次迭代 **do**

  对于第 i 个智能体，根据当前状态 s 和探索-利用策略，选择动作 a_i 并执行。

  得到下一状态 s'，观察自身的奖励 r_i，并且观察所有 friend 动作 $(a_1,\ldots,a_{n_1})$ 和 foe 动作 $(o_1,\ldots,o_{n_2})$。

  更新 $Q_i(s,a_1,\ldots,a_{n_1},o_1,\ldots,o_{n_2})$：

$$Q_i(s,a_1,\ldots,a_{n_1},o_1,\ldots,o_{n_2}) \leftarrow Q_i(s,a_1,\ldots,a_{n_1},o_1,\ldots,o_{n_2})+\alpha\left[r_i+\gamma V_i(s')-Q_i(s,a_1,\ldots,a_{n_1},o_1,\ldots,o_{n_2})\right]$$

  计算 $V_i(s)$ 和 $\pi_i(s,\cdot)$，更新公式为：

$$V_i(s)=\max_{\pi_1(s),\ldots,\pi_{n_1}(s)}\ \min_{o_1,\ldots,o_{n_2}\in O_1\times\cdots\times O_{n_2}}\ \sum_{a_1,\ldots,a_{n_1}\in A_1\times\cdots\times A_{n_1}} Q_i(s,a_1,\ldots,a_{n_1},o_1,\ldots,o_{n_2})\pi_1(s,a_1)\ldots\pi_{n_1}(s,a_{n_1})$$

end for

思考和讨论

1. 多智能体系统的分类和最优解的设定
2. 纳什均衡解的概念
3. Nash-Q学习算法

谢　谢　！