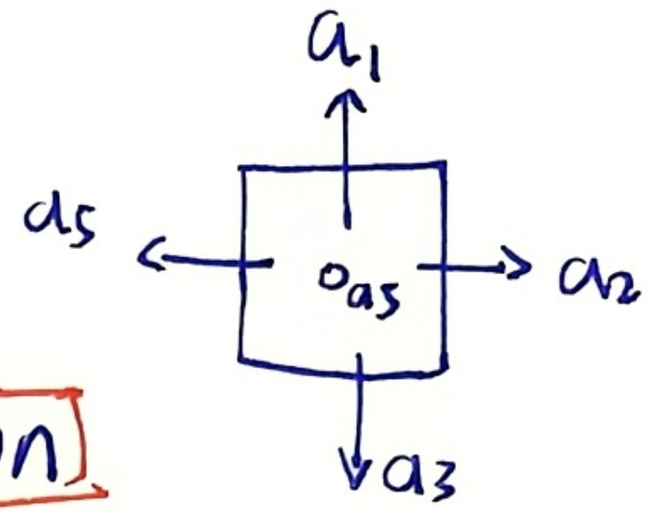
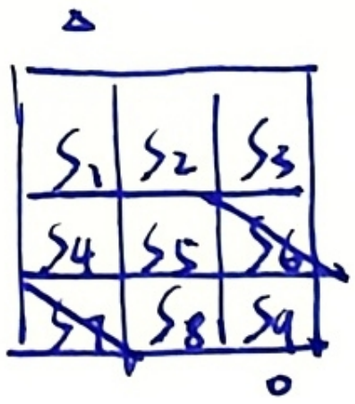
state space $S = \{s_i\}_{i=1}^{9}$

action space $A(s_i) = \{a_i\}_{i=1}^{5}$





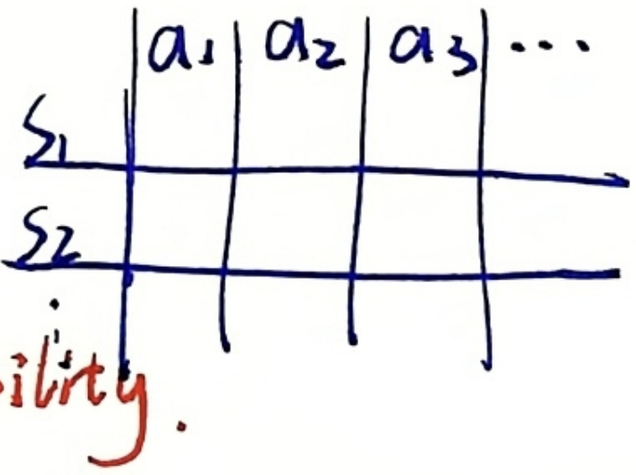
$s_1 \xrightarrow{a_2} s_2$: **state transition**

$s_1 \xrightarrow{a_1} s_1$: 撞墙了.

↘ defines the interaction with env.

two case { $s_5 \xrightarrow{a_2} s_6$ accessible but with penalty
$s_5 \xrightarrow{a_2} s_5$ inaccessible

Tabular representation:



but only deterministic.

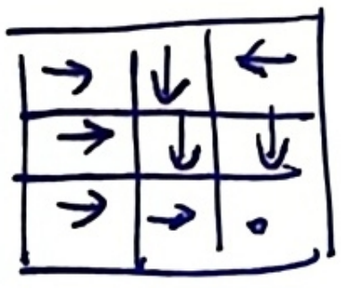
+ probability.

state transition probability.

$s_1 \xrightarrow{a_2} s_2$: $p(s_2|s_1, a_2) = 1$ 条件概率.

$p(s_i|s_1, a_2) = 0 \quad \forall i \neq 2$

☆ policy. what actions to take at a state



different paths with different starting point

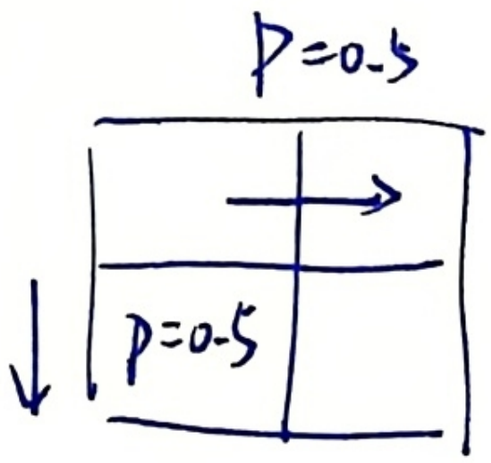
Mathematical representation.

π → for policy.

$$\left.\begin{array}{l}\pi(a_1|s_1)=0\\ \pi(a_2|s_1)=1\\ \pi(a_3|s_1)=0\\ \cdots\\ \cdots\end{array}\right\}\ \text{plus} = 1.$$

1. deterministic

2. stochastic.



$$\left.\begin{array}{l}\pi(a_1|s_1)=0\\ \pi(a_2|s_1)=0.5\\ \pi(a_3|s_1)=0.5\\ \cdots\cdots\end{array}\right\}\ \text{plus}=1.$$

random.rand(). 创造 0~1 随机数.

若 $x \le 0.5$ take a_2. $x > 0.5$ take a_3.

☆ reward

δ

human-machine

>0 encouragement

<0 punishment.

=0. No punishment.

> 也可以正负互换.

interface

~~out of boundary. / forbidden cell = -1~~

$r_{bound} = -1$ $r_{forbid} = -1$ $r_{target} = +1$

otherwise $r = 0$.

Mathematical description.

$s_1 \xrightarrow{a_1}$ $\longrightarrow$ $r=-1$

$\begin{cases} p(r=-1 \mid s_1, a_1) = 1 \\ p(r \neq -1 \mid s_1, a_1) = 0. \end{cases}$) reward transition · stochastic

正负也许可知. 但具体值未知

Trajectory.

- 扔 轨迹

Tran - Ject - Ory

a state - action - ~~a~~reward chain

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_2} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9.$$

return : the sum of them. $0+0+0+1=1$.

return = 1 .

return and reward: reward (R) 是 action 后立刻的反馈

return (G) 是 reward 累加和

Discounted return

引入 discount rate $\gamma \in [0,1)$

$$= 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \cdots$$

$$= \gamma^3 (1 + \gamma + \gamma^2 + \cdots) = \gamma^3 \frac{1}{1-\gamma}$$ ←

可评估更远更近的 reward.

$\begin{cases} \gamma \text{ close } 0 & \text{near future} \\ \gamma \text{ close } 1. & \text{far future}. \end{cases}$

Episode 有限步 finite 的 tasks.

Continuing tasks vs Episode.

为了数学形式统一. 有以下两种方法统一:

option 1. absorbing state. 仅有留在原地的 action, reward=0

option 2. normal state with policy. reward=1、更一般化.

memoryless property → MDP Markov decision process ← policy

* sets ↑ probility distribution.

1. sets
   - state: the set of states S
   - Action: the set of actions $A(s)$
   - Reward: the set of rewards $R(s,a)$

MDP 无历史无记忆性的…… 关于 policy …… 以及 probility distribution 的状态转化 process

2. probability.
   - state $p(s'|s,a)$
   - Reward $p(r|s,a)$.

3. (policy) s. a. $\pi(a|s)$ $\sum_a \pi(a|s)=1$.

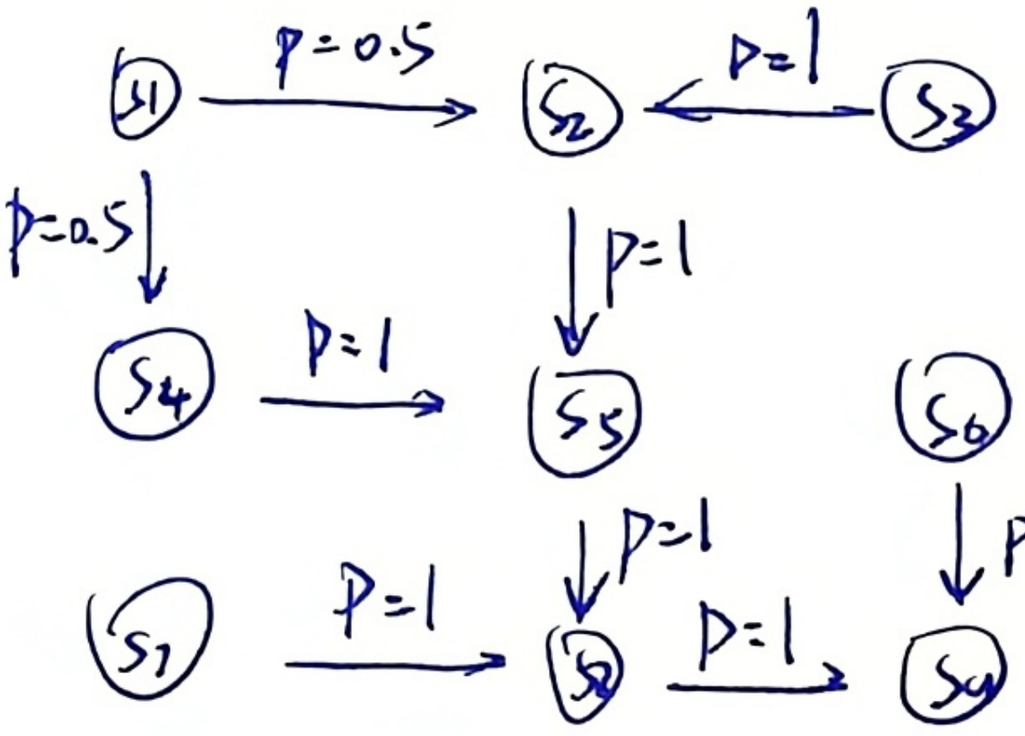
4. Markov property: memoryless property:

~~$p(s_{t+1}|a_{t+1}, s_t)$~~

$$p(s_{t+1}|a_{t+1}, s_t, \dots a_1, s_0) = p(s_{t+1}|a_{t+1}, s_t)$$

$$p(r_{t+1}|a_{t+1}, s_t \dots a_1, s_0) = p(r_{t+1}|a_{t+1}, s_t).$$

Markov process



Markov decision process becomes Markov process

once the policy is given!

↓

$\pi(a|s)$

MDP: $P(s'|s,a)$

MP $P(s'|s)$. a 不见了.

State value ~~G_t~~ $V_\pi(s)$.

$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1}$

S_t, A_t, R_{t+1} are all random variables. 需引入E

$S_t \to A_t$: $\pi(A_t=a \mid S_t=s)$

$S_t, A_t \to R_{t+1}$: $p(R_{t+1}=r \mid S_t=s, A_t=a)$

$S_t, A_t \to S_{t+1}$: $p(S_{t+1}=s' \mid S_t=s, A_t=a)$

↗ ~~状态转移后~~.

trajectory: $S_t \xrightarrow{A_t} R_{t+1}, S_{t+1} \xrightarrow{A_{t+1}} R_{t+2}, S_{t+2} \xrightarrow{A_{t+2}} R_{t+3} \cdots$

discounted return: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

↕

random variables.

state-value ~~function~~ $V_\pi(s) = E[G_t \mid S_t = s]$.

$V(s, \pi)$ $V_\pi(s)$.

$V_\pi(s)$ vs G R:

- $V_\pi(s)$ 是从某个状态出发反复得 所有可能 回报的均值.

$\pi(a|s)$, $p(r|s,a)$, $p(s'|s,a)$ 若都确定. 则 $V_\pi(s)$ = 常数

Bellman Equation { state value
the Bellman equation

return 可评估策略好坏.

$return_1 = 0 + \gamma 1 + \gamma^2 1 + \cdots$

$= \frac{\gamma}{1-\gamma}$

$return_2 = -1 + \gamma 1 + \gamma^2 1 + \cdots$

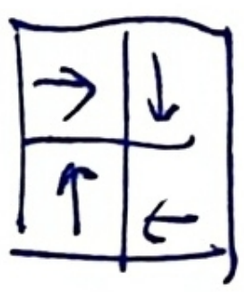
$= -1 + \frac{\gamma}{1-\gamma}$

$return_3 = 0.5(-1 + \frac{\gamma}{1-\gamma}) + 0.5(\frac{\gamma}{1-\gamma})$

$= -0.5 + \frac{\gamma}{1-\gamma}$.

$return_1 > return_3 > return_2$.

Mothod 1:



$v_1 = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots$

$v_2 = r_2 + \gamma r_3 + \gamma^2 r_4 + \cdots$

$v_3 = r_3 + \gamma r_4 + \gamma^2 r_1 + \cdots$

$v_4 = r_4 + \gamma r_1 + \gamma^2 r_2 + \cdots$

The returns rely on each other. Bootstrapping 自举.

$v_1 = r_1 + \gamma(r_2 + \gamma r_3 + \cdots) = r_1 + \gamma v_2$

$v_2 = r_2 + \gamma(r_3 + \gamma r_4 + \cdots) = r_2 + \gamma v_3$

$v_3 = r_3 + \gamma(r_4 + \gamma 1 + \cdots) = r_3 + \gamma v_4$

$v_4 = r_4 + \gamma(r_1 + \gamma r_2 + \cdots) = r_4 + \gamma v_1$

$$V_\pi(s) = E[G_t | S_t = s]$$

$$= E[R_{t+1} + \gamma G_{t+1} | S_t = s]$$

$$= \underbrace{E[R_{t+1} | S_t = s]}_{(1)} + \underbrace{\gamma E[G_{t+1} | S_t = s]}_{(2)}$$

(1): $E[R_{t+1} | S_t = s] = \sum_a \pi(a|s) E[R_{t+1} | S_t = s, A_t = a]$

immediate rewards $= \sum_a \pi(a|s) \sum_r p(r|s,a) r$

(2): $E[G_{t+1} | S_t = s] = \sum_{s'} E[G_{t+1} | S_t = s, S_{t+1} = s'] p(s'|s)$

(马尔可夫性质) $= \sum_{s'} E[G_{t+1} | S_{t+1} = s'] p(s'|s)$

$= \sum_{s'} V_\pi(s') p(s'|s)$

$= \sum_{s'} V_\pi(s') \sum_a p(s'|s,a) \pi(a|s)$

对所有状态成立: Bellman:

$$V_\pi(s) = \sum_a \pi(a|s) \left[\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) V_\pi(s') \right]$$

△ immediate △ future.

s. state v s' state. v

Bootstrapping 计算 $\pi(a|s)$ 是给出的. policy evaluation

$p(r|s,a)$ $p(s'|s,a)$. dynamic model 环境.

贝尔曼方程

不同策略的条件概率累加（加权）.

[每个不同策略下的 reward 的累加.加权].

+ [折现后未来状态的状态函数的累加加权].

累加权与实际奖项.

$$V_\pi(s) = \sum_a \pi(a|s)\Big[\underbrace{\sum_r p(r|s,a)\,r}_{\text{大累加套.}} + \gamma \underbrace{\sum_{s'} p(s'|s,a)\,V_\pi(s')}_{V_\pi(s')}\Big]$$

本质是 $V_\pi(s) = r + \gamma V_\pi(s')$ 的一般化.

由 $V_\pi(s) = r + \gamma r_1 + \gamma r_2 + \gamma r_3 \cdots$ 得到.

$\sum_a \pi(a|s)$.

引入不同 action/策略.

$\sum_r p(r|s,a)$.

(不同策略下)引入不同的 reward 概率.与 r.

$\sum_{s'} p(s'|s,a)$

(不同策略下).去到 s' 新状态的概率.与 $V_\pi(s')$

policy evaluation.

a policy. finding out the corresponding state Values is called policy evaluation.

solve Bellman 迭代☆.

$$V_\pi = r_\pi + \gamma P_\pi V_\pi$$

closed-form : $V_\pi = (I - \gamma P_\pi)^{-1} r_\pi$.

interative solution. $V_{k+1} = r_\pi + \gamma P_\pi V_k$.

$V_0 \to V_1$

$V_1 \to V_2$

$V_2 \to V_3$.

sequence $\{V_0, V_1, V_2 \cdots\}$.

$V_k \to V_\pi = (I - \gamma P_\pi)^{-1} r_\pi, \quad k \to \infty$.

$V_\pi = r_\pi + \gamma P_\pi V_\pi$. 设 $F(V_\pi) = r_\pi + \gamma P_\pi V_\pi$

则 V_π 为 $F(V_\pi)$ 的不动点.

$$V_{k+1} = r_\pi + \gamma P_\pi V_k$$
$$V_\pi = r_\pi + \gamma P_\pi V_\pi.$$

$$|V_{k+1} - V_\pi| = \gamma P_\pi |V_k - V_\pi|.$$

新误差 $\leq \gamma \times$ 旧误差.

☆ Action value.

S_v : the average return the agent can get starting from a state.

A_v : the average return the agent can get starting from a state and taking an action.

$$q_\pi(s,a) = E[G_t | S_t = s,\ A_t = a]$$

↓

function of state-action pair.

$$E[G_t | S_t = s] = \sum_a E[G_t | S_t = s,\ A_t = a]\pi(a|s)$$

$V_\pi(s)$. $q_\pi(s,a)$.

即 $$V_\pi(s) = \sum_a \pi(a|s) \cdot q_\pi(s,a) \quad (2)$$

也有. $$q_\pi(s,a) = \sum_r p(r|s,a)\, r + \gamma \sum_{s'} p(s'|s,a) V_\pi(s') \quad (4)$$

(2): $V_\pi(s)$ 如何由 $q_\pi(s,a)$ 得到

(3): $q_\pi(s,a)$ 如何由 $V_\pi(s')$ 得到.

策略只take a_2 情况下. 其他a也都是可以计算的. 不等于0.

① 先算 state values 后算 action values.

② ★直接算 action values ／ with models
　　　　　　　　　　＼ without models.

Bellman Optimality Equation　BOE.

$$a^* = \arg\max_a q_\pi(s_1, a).$$

$$\pi_{new}(a|s_1) = \begin{cases} 1, & a = a^* \\ 0, & a \neq a^* \end{cases}$$

π^* 取大是.

greedy 的好处.

定义：$v_{\pi_1}(s) \geq v_{\pi_2}(s)$. for all $s \in S$.

π_1 is better than π_2.

"A policy π^* is optimal if $v_{\pi^*}(s) \geq v_\pi(s)$ for all s and for any other policy π"

BOE

$$v(s) = \max_\pi \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v(s') \right)$$

$$= \max_\pi \sum_a \pi(a|s) q(s,a).$$

$$v = \max_\pi (r_\pi + \gamma P_\pi v).$$

(BOE).

~~$V = \max_{\pi} (r_\pi + \gamma R$~~

$$V = \max_{\pi} (r_\pi + \gamma P_\pi v).$$ tidy and elegant.

How to solve $\max_{\pi} \sum_a \pi(a|s) q(s,a)$

对于 q.(action value) 最大的. 将 策略 pi(a|s) 都向大的q分配. 可以确保 V 的 optimal.

$$V = \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v(s') \right)$$

$$= \max_{\pi} \sum_a \pi(a|s) q(s,a).$$

considering $\sum_a \pi(a|s) = 1$. $\max_{\pi} \sum_a \pi(a|s) q(s,a) = \max_{a \in A(s)} q(s,a)$

取 optimality when $\pi(a|s) = \begin{cases} 1, & a = a^* \\ 0, & a \neq a^* \end{cases}$

$$a^* = \arg\max_a q(s,a)$$

3.3 How to solve BOE.

solve BOE:

$$f(v) = \max_{\pi} (r_\pi + \gamma P_\pi v) \qquad v = f(v)$$

$$[f(v)]_s = \max_{\pi} \sum_a \pi(a|s) q(s,a)$$

收缩

Contraction mapping theorem.

fixed point: $x \in X$. $f(x) = x$. 即 fixed point.

$$\|f(x_1) - f(x_2)\| \le \gamma \|x_1 - x_2\|$$

① 一定有 x^* 使 $f(x^*) = x^*$

② unique.

③ $x_{k+1} = f(x_k)$ $x_k \to x^*$ as $k \to \infty$.

Convergence rate is exponentially fast

对矩阵同理.

BOE $v = f(v) = \max_{\pi} (r_\pi + \gamma P_\pi v)$.

是 contraction mapping.

Policy optimality

v^* is optimality equation to BOE.

$$v^* = \max_{\pi} (r_\pi + \gamma P_\pi v^*).$$

suppose $\pi^* = \arg\max_{\pi} (r_\pi + \gamma P_\pi v^*)$

$$\therefore \quad v^* = r_{\pi^*} + \gamma P_{\pi^*} v^*.$$

$v^* = v_{\pi^*}$. 是特殊的 BE.

$v^* \geq v_\pi$. π 为 optimality policy.

π^*

$$\pi^*(a|s) = \begin{cases} 1 & a = a^*(s) \\ 0 & a \neq a^*(s). \end{cases}$$

π^* deterministic. greedy.

BOE

Reward design: r

System model: $p(s'|s,a)$, $p(r|s,a)$

Discout rate: γ.

$\vec{r} \to ar+b$. 只要 $a>0$, $v' = av^\pi + \frac{b}{1-\gamma}1$.

state value 唯一、但 π 不一定唯一

Value 值迭代算法 (比较蒙. 得多看几遍).

step1. policy update.

step2. value update. V_k 并非 state value.

$V_k(s) \rightarrow q_k(s,a) \rightarrow$ greedy policy $\pi_{k+1}(a|s) \rightarrow$ new value V_{k+1}

$= \max_a q_k(s,a)$

老师的例子讲得很好.

有 Q-table. * 对应的 q 由 a, V 决定.

V 随机一下. a 可以找到 q 大的 V_1, V_2, V_3 ...

~~将 V_1,~~ 以及 V_1, V_2, V_3 所 take 的 action

↓ value update.

↓ policy update.

↓ 对于每一个 state 都有当前最优 value

进入下一个迭代

继续又能算出, Q-table 中每个 state 对应的最优 Q

又能更新 value 与 policy.