开悟

强化学习基础

策略迭代与值迭代

开悟强化学习

人工智能专业课程

目录

1.策略迭代

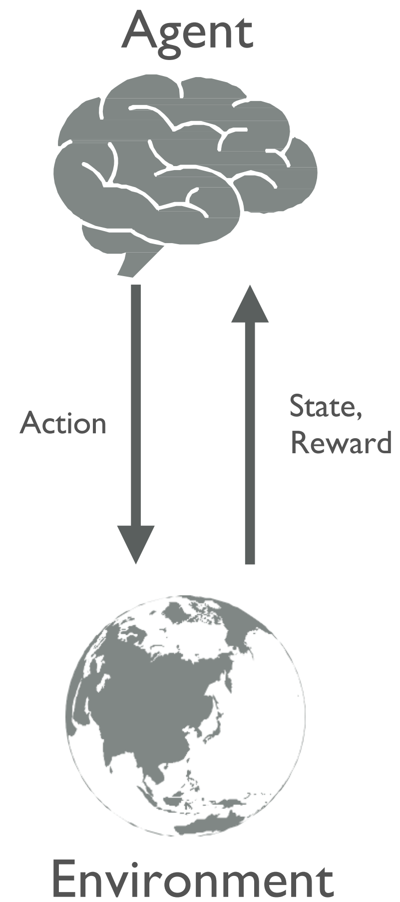
单智能体RL形式化定义：

由六元组构成的马尔可夫决策过程定义

具体定义如下：

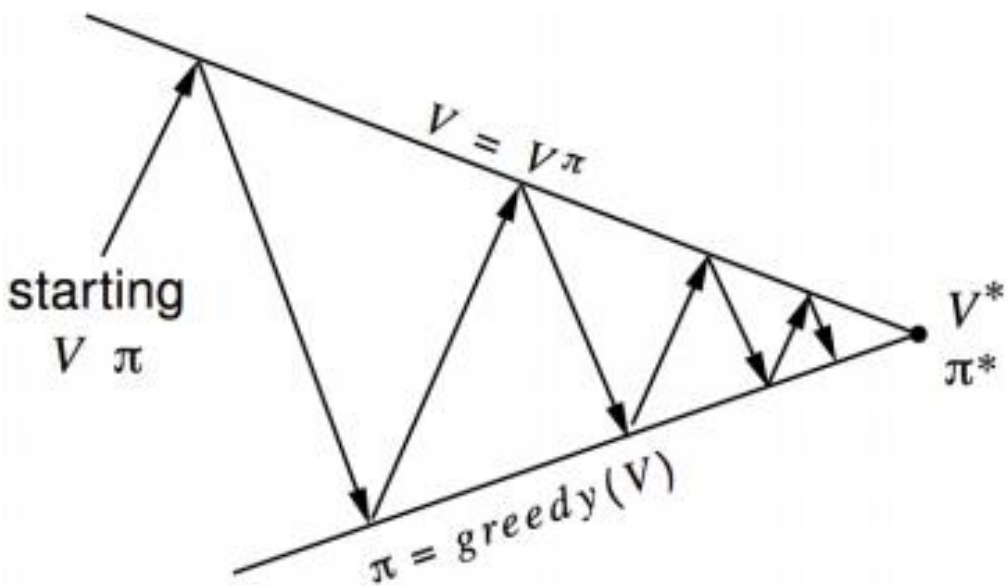
Markov Decision Process(MDP) $(S, A, R, T, P_0, \gamma)$

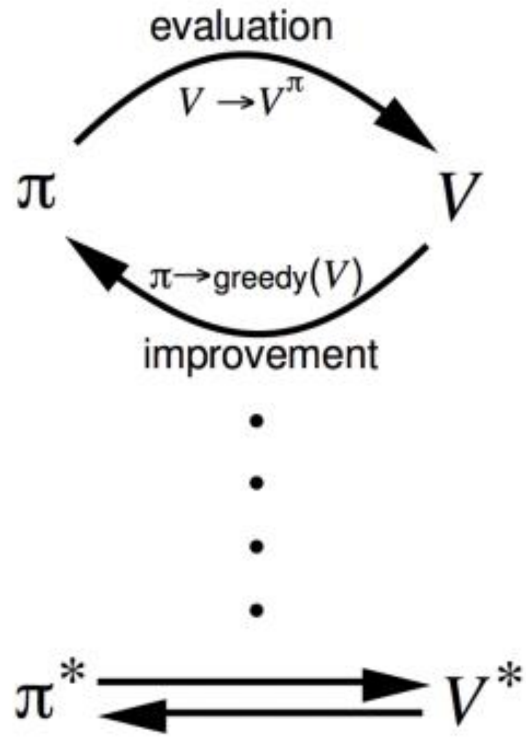
- S denotes the state space
- A is the action space
- $R = R(s, a)$ is the reward function
- $T: S \times A \times S \rightarrow [0,1]$ is the state transition function
- P_0 is the distribution of the initial state
- γ is a discount factor
- Goal: find the optimal policy that maximizes expected reward



1.策略迭代

从一个初始化的策略出发，先进行策略评估（policy evaluation），然后改进策略（policy improvement），评估改进的策略，再进一步改进策略，经过不断迭代更新，直达策略收敛，这种算法被称为“策略迭代”





1.策略迭代

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

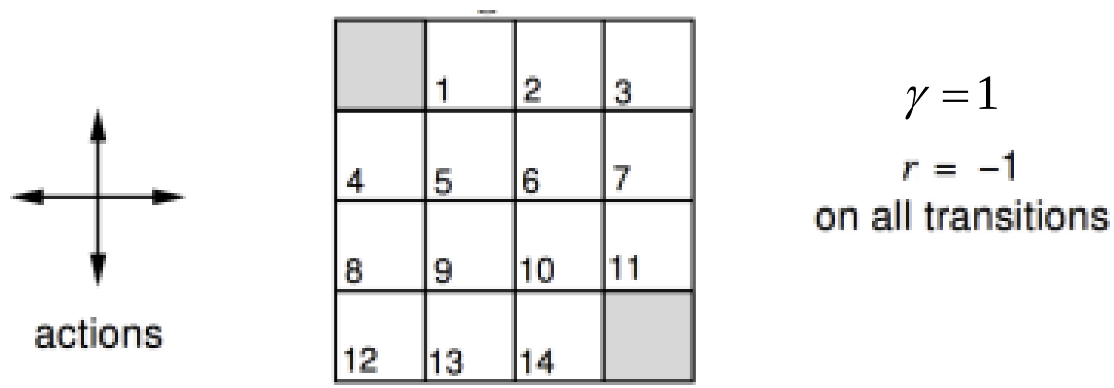
1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s',r|s,\pi(s))\big[r + \gamma V(s')\big]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

1.策略迭代

实例：一个4×4的小网格世界，左上角和右下角是目的地，每个格子行动方向为上下左右，每走一步reward-1，求一个在每个状态都能以最少步数到达目的地的最优行动策略。解决思路：我们从最开始的随机（1/4）策略开始，对其进行policy evaluation, 然后进行policy iteration by acting greedy



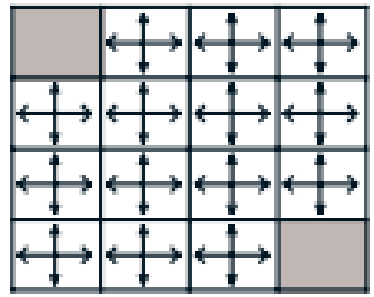
1.策略迭代

v_k for the Random Policy

Greedy Policy w.r.t. v_k

$k = 0$

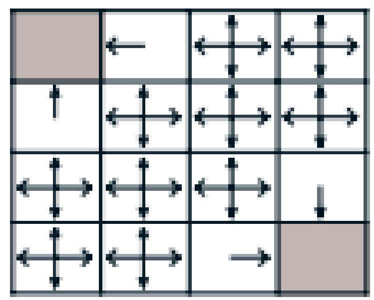
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



random policy

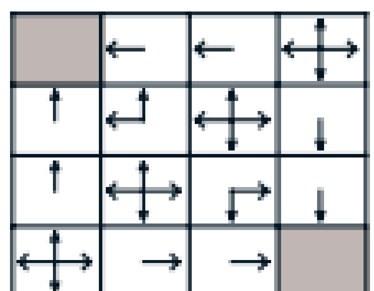
$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



1.策略迭代

$k = 3$

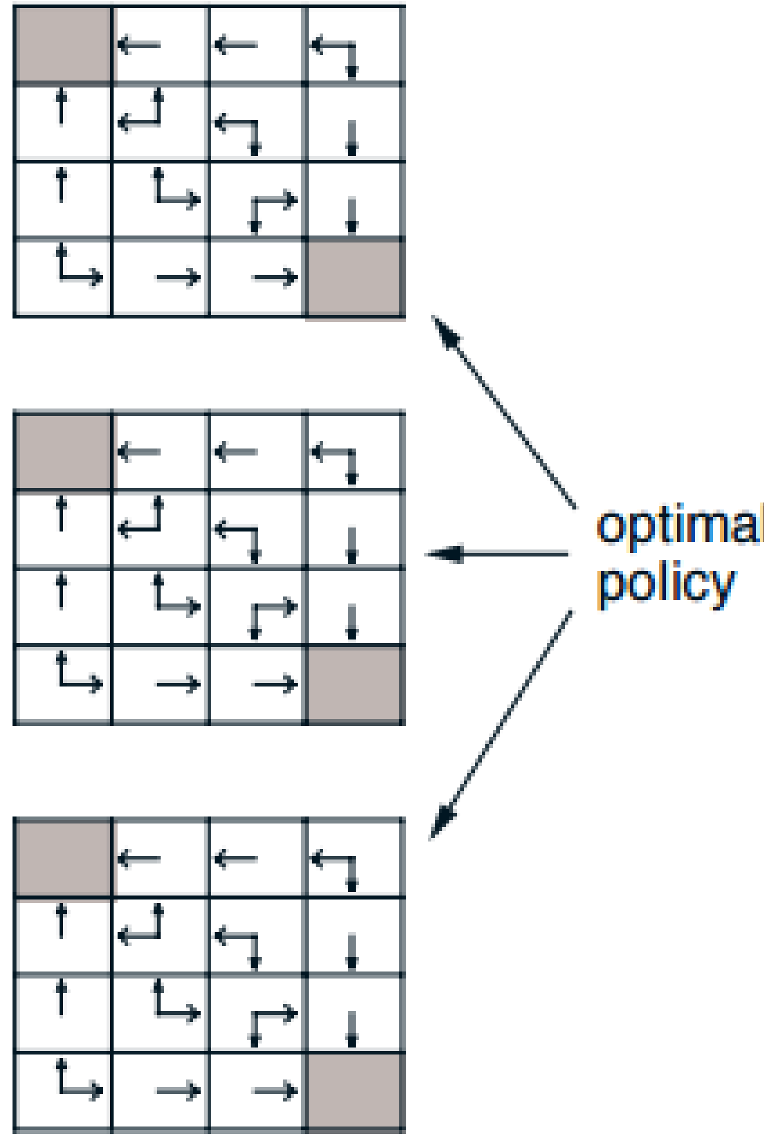
0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



2.值迭代

对每一个当前状态 s ,对每个可能的动作 a 都计算一下采取这个动作后到达的下一个状态的期望价值。看看哪个动作可以到达的状态的期望价值函数最大，就将这个最大的期望价值函数作为当前状态的价值函数 V(s) ，循环执行这个步骤，直到价值函数收敛。

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma V(s')\right]$

作业

开悟

- 在网格环境MiniWorld上实现策略迭代算法或者值迭代算法其中之一(取折扣因子$\gamma = 0.9$).
- 详见【第8次作业.pdf】
- 期限: 2023年5月14日23:59