第8讲 决策理论规划

尹慧琳，yinhuilin@tongji.edu.cn
同济大学 电子与信息工程学院

第 8 讲 决策理论规划

8.1 决策理论规划概述

8.2 Markov模型

8.3 马尔科夫决策过程MDP的优化控制

8.4 动态规划 Dynamic Programming

决策理论（Decision theory）

- 决策理论
  - 是一种决策的理论框架，用于衡量行动方案的优劣。
- 决策理论的基础
  - 概率论（Game theory）

    用于在给定的状态下求得某个行动可能结果的概率分布、以及合理性偏好函数。
  - 效用论（Utility theory）

    采用效用函数，使得智能主体偏好的规划具有更高的预期效用

    最大期望效用（maximum expected utility, MEU）

但是，决策理论并未涉猎如何构建具有高期望效用的规划。

决策理论规划（Decision-Theoretic Planning）

- 决策理论规划 = 决策理论 + 人工智能规划
  - 形式框架：马尔科夫决策过程（Markov decision process）
  - 优化控制：动态规划（Dynamic programming）、线性规划（Linear programming）
- 决策理论规划 ≒ 不确定性环境规划（planning under uncertainty）
  - 从环境接收的信息是不完全或不完备的
  - 动作并非总是得到同样的结果
  - 需要在规划的不同结果之间做出权衡
- 马尔科夫决策过程 ∈ 马尔科夫模型（Markov models）

第 8 讲 决策理论规划

马尔科夫模型（Markov models）

- 概述
  - 一种统计模型，用于对随机变化的系统进行建模。
- 性质
  - 马尔科夫模型的下一个状态只依赖于当前的状态，而与之前发生的事件无关。

四种马尔科夫模型

	完全可观测（fully observable）	部分可观测（partially observable）
自主（autonomous）	马尔科夫过程（Markov process）	隐马尔科夫模型（Hidden Markov model）
控制（controlled）	马尔科夫决策过程（Markov decision process）	部分可观测马尔科夫决策过程（Partially observable Markov decision process）

随机过程（Stochastic process, SP）

一个随机过程被定义为一组随机变量的集合，即：$\{S(0), S(1), \dots, S(t)\}$，其中：$\{0, 1, \dots, t\}$是一个索引集（index set）；$t$表示时间，其数值为时间点；$S(t)$是时间$t$的一个随机变量，亦被称为随机过程在$t$时刻的状态。

若索引集是一个可数集，则称SP为一个离散时间随机过程，采用概率的方法加以研究；而当索引集为一个连续值时，则SP是一个连续时间随机过程，通常采用分析的方法。

随机过程的实例

细菌种群的增长、由于热噪声或气体分子的移动而导致电流波动等。

随机过程的应用

生物学、化学、生态学、神经科学、物理学、以及工程和技术领域，如：图像处理、信号处理、信息论、计算机科学、密码学、电信等；此外，还被广泛用于金融领域。

马尔科夫性质（Markov property）

在给定当前状态$S(t)$及之前的状态集 $\{S(0), S(1), \ldots, S(t-1)\}$ 的条件下，一个随机过程的下一个状态$S(t+1)$的条件概率仅依赖于当前状态$S(t)$。表示为：

$$P(S(t+1) \mid S(0), S(1), \ldots, S(t)) = P(S(t+1) \mid S(t))$$

这个公式是判断一个随机过程是否具有马尔科夫性质的要素。

- 无记忆性质（memory-less property）

马尔科夫性质是随机过程的无记忆性质：在给定当前状态$S(t)$时，下一个状态$S(t+1)$与之前的状态集 $\{S(0), S(1), \ldots, S(t-1)\}$ 是条件独立的。

所有的马尔科夫模型都具有马尔科夫性质。

马尔科夫过程（Markov process, MP）

一个马尔科夫过程MP表示为一个二元组，即：$\mathrm{MP} = \langle S, T \rangle$。其中：$S$是一个状态集；$T$是当前状态转换为下一个状态的转换函数（transformation function），$T: S \rightarrow S$。

$$T(s_t, s_{t+1}) = \mathbb{P}[S(t+1) = s_{t+1} \mid S(t) = s_t]$$

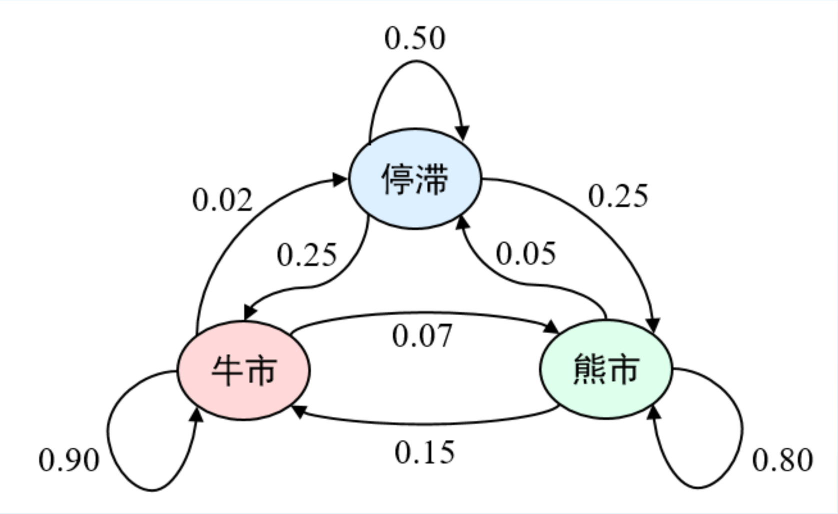
其中，$\mathbb{P}[.]$为一个转换矩阵（transition matrix）：

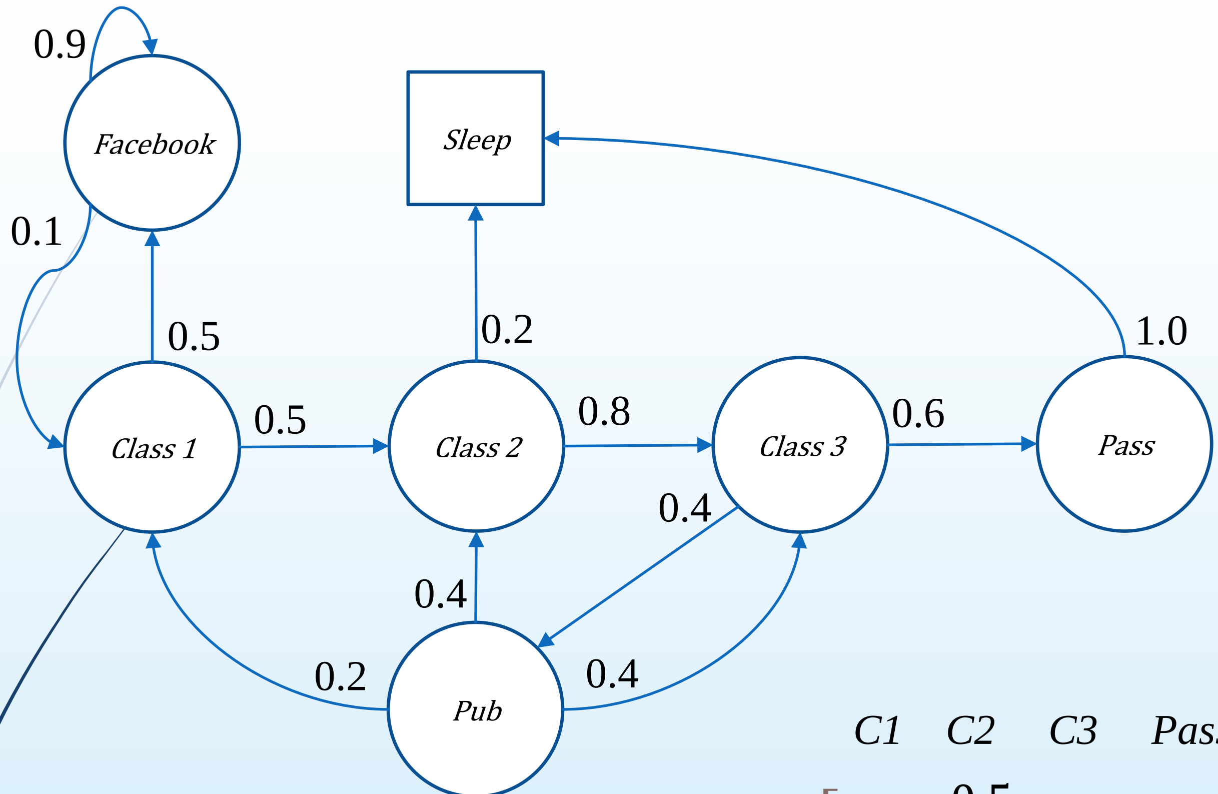
$$\mathbb{P} = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0t} \\ P_{10} & P_{11} & \cdots & P_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ P_{t0} & P_{t1} & \cdots & P_{tt} \end{bmatrix}$$

马尔科夫过程是具有马尔科夫性质的随机过程

马尔科夫链（Markov chain）

一个过程的离散随机变量序列；通常用有向图来表示，其中图的边被标记为从时间t的状态到时间$t+1$的状态的概率，可以表示为状态转移概率矩阵。比如：用马尔科夫链表示某股票市场一周内的牛市、熊市或停滞的市场趋势。





$$
p = \begin{array}{r} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array}
\begin{array}{c}
\begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\
\begin{bmatrix}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & & 1.0 \\
0.2 & 0.4 & 0.4 & & & & \\
0.1 & & & & & 0.9 & \\
 & & & & & & 1
\end{bmatrix}
\end{array}
$$

马尔科夫奖励过程（Markov Reward Process, MRP）

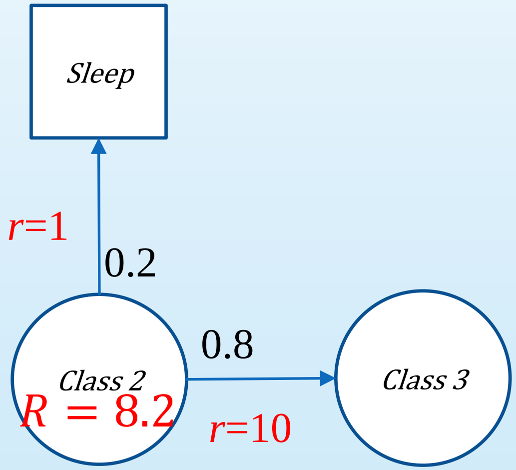
MRP = MP + reward

一个马尔科夫奖励过程MRP是满足马尔科夫性质的四元组，即：$\mathrm{MRP} = \langle S, T, R, \gamma \rangle$。其中：$S$是有限的状态集；$T$是转换函数，$R$是一个奖惩函数（reward function），该奖惩来自于外部环境；γ为折扣（discounting）。

$$T(s_t, s_{t+1}) = \mathbb{P}[S(t+1) = s_{t+1} \mid S(t) = s_t]$$

$$R(s_t) = \mathbb{E}[R(t+1) = r_{t+1} \mid S(t) = s_t]$$

$$(= \textstyle\sum_{s'} P_{ss'} r_{s \to s'})$$



马尔科夫决策过程（Markov Decision Process, MDP）

MDP = MRP + Action

一个马尔科夫决策过程MDP是满足马尔科夫性质的五元组，即：$\text{MDP} = \langle S, A, T, R, \gamma \rangle$。其中：$S$是有限的状态集；$A$是有限的动作集；$T$是转换函数，$T: A \times S \to S$，即在动作$A(t) = a_t$下的条件下，从当前状态$S(t) = s_t$转换为下一个状态$S(t+1) = s_{t+1}$，表示为：

$$T(s_t, a_t, s_{t+1}) = \mathbb{P}[S(t+1) = s_{t+1} \mid S(t) = s_t, A(t) = a_t]$$

R是一个奖惩函数（reward function），$R: S \times A \to \mathbb{R}$，即在动作$A(t) = a_t$和当前状态$S(t) = s_t$条件下得到的奖惩$R(t+1) = r_{t+1}$，

$$R(s_t, a_t) = \mathbb{E}[R(t+1) = r_{t+1} \mid S(t) = s_t, A(t) = a_t]$$

该奖惩来自于外部环境；γ为折扣（discounting）。

第 8 讲 决策理论规划

8.1 决策理论规划概述

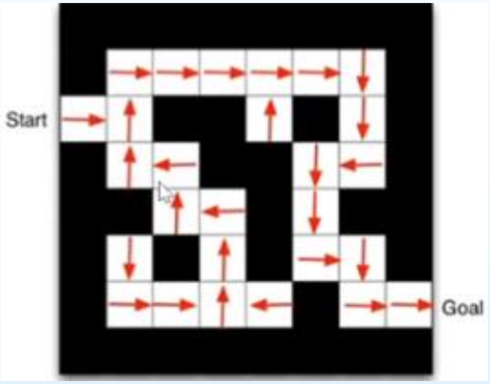
8.2 Markov模型

8.3 马尔科夫决策过程MDP的优化控制

8.4 动态规划

策略（Policy）

给定一个马尔科夫决策过程MDP $= \langle S, A, T, R, \gamma \rangle$，一个策略$\pi$是一个可计算函数（computable function），对每个状态$S(t) = s_t$，输出一个动作$A(t) = a_t$。



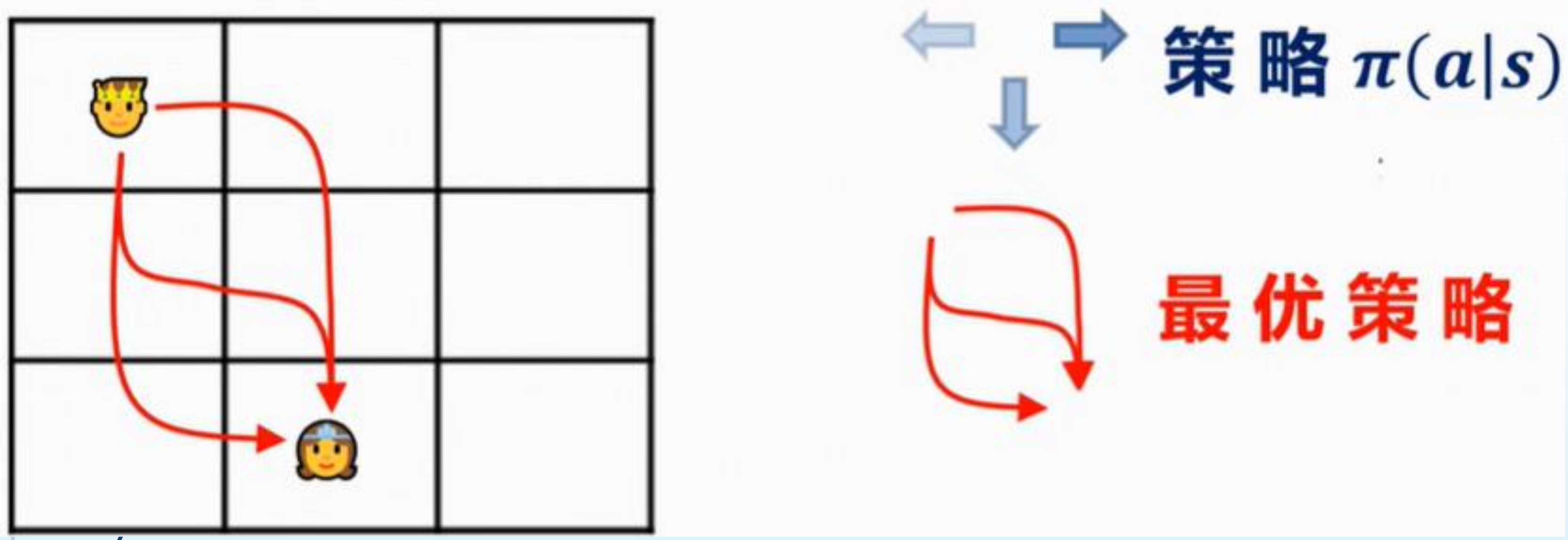
- 确定性策略（deterministic policy）

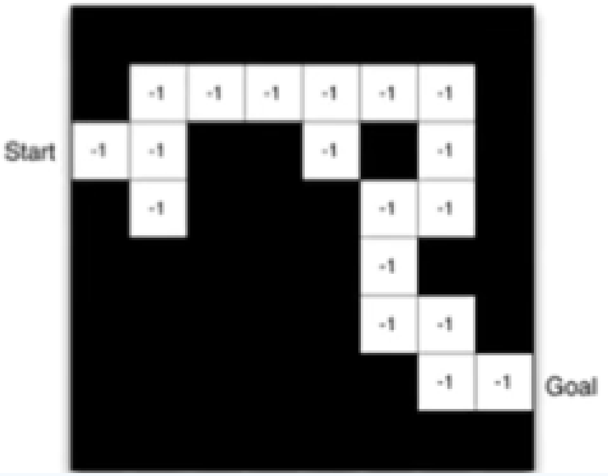
一个确定性策略表示为 $\pi: S \longrightarrow A$，记作 $\pi(s_t) = a_t$

- 随机策略（stochastic policy）

一个随机策略表示为 $\pi: S \times A \longrightarrow [0,1]$，使得对每个$s_t$，满足$\pi(s_t, a_t) \geq 0$，且$\sum_{a_t} \pi(s_t, a_t) = 1$。

马尔科夫决策过程优化控制的核心问题是找到一个最优策略。


策略 $\pi(a|s)$
最优策略



奖惩（Reward）

设时间t之后得到的即时奖惩

序列为 $r_{t+1}, r_{t+2}, r_{t+3}, \ldots,$

回报(Return) G_t为这些奖惩序列的特定函数。

回报G_t的最简单情况是即时奖惩和部分未来奖惩之和，即：

$$G_t = r_t + r_{t+1} + r_{t+2} + \cdots + r_{t+h} = \sum_{k=0}^{h} r_{t+k}$$

其中，h是一个有限时域（finite horizon），即其时间步长（time step）是有限的。

注意：回报是累积的奖惩之和，而每个奖惩是回报的一部分。

折扣（Discounting）

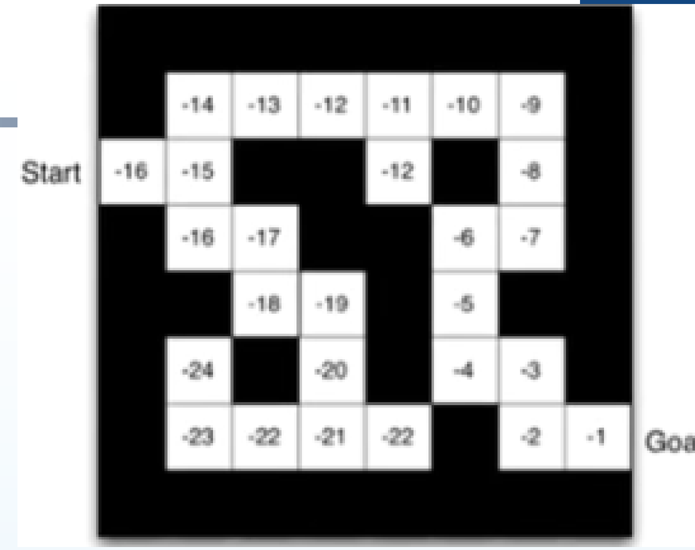
考虑折扣，回报变为：

$$G(t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

其中，γ是折扣因子（discount factor），表示下一个奖惩和当前奖惩之间重要性的差异。

理论上，$0 \le \gamma \le 1$，称为折扣率（discount rate），表示回报G(t)中即时奖惩和未来奖惩的关联程度。

价值函数（value function）



- 策略π的状态价值函数（state value function）

一个状态s在策略π下的价值函数定义为始于s并遵循π的预期回报，表示为：

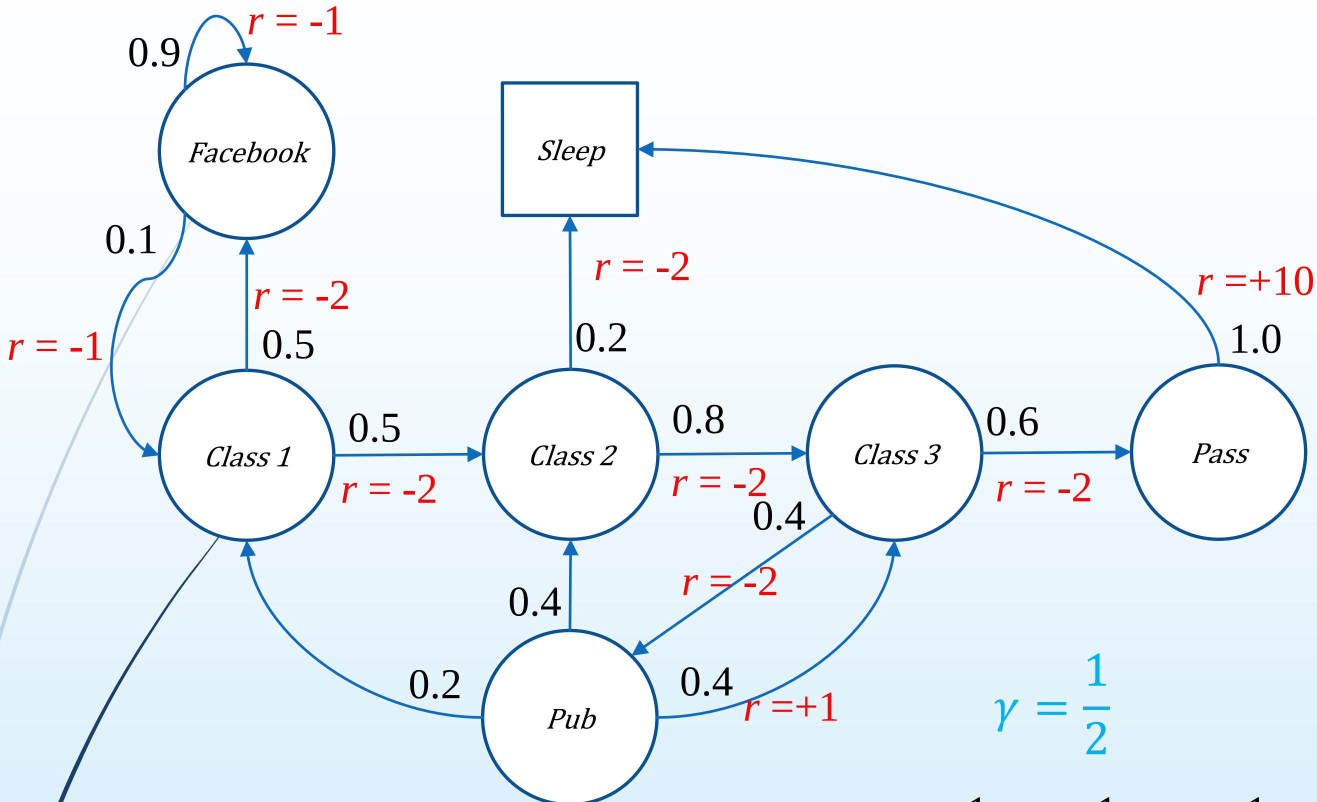
$$V^{\pi}(s) = \mathbb{E}^{\pi}(\mathrm{G}(t)|\mathrm{S}(t) = s) = \mathbb{E}^{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} | \mathrm{S}(t) = s\right]$$

- 策略π的状态-动作价值函数（state-action value function）

一个策略π的状态-动作价值函数定义为始于状态s、执行动作a并遵循策略π的预期回报，表示为：

$$Q^{\pi}(s,a) = \mathbb{E}^{\pi}(\mathrm{G}(t)|\mathrm{S}(t) = s, \mathrm{A}(t) = a)$$
$$= \mathbb{E}^{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} | \mathrm{S}(t) = s, \mathrm{A}(t) = a\right]$$

举例: MDP奖惩、回报、折扣系数、价值函数



$\gamma = \frac{1}{2}$

C1 C2 C3 Pass Sleep

C1 FB FB C1 C2 Sleep

C1 C2 C3 Pub C2 C3 Pass Sleep

C1 FB FB C1 C2 C3 Pub C1 ...

FB FB FB C1 C2 C3 Pub C2 Sleep

$$G_1 = -2 - 2 \times \frac{1}{2} - 2 \times \frac{1}{4} + 10 \times \frac{1}{8} \qquad = -2.25$$

$$G_2 = -2 - 1 \times \frac{1}{2} - 1 \times \frac{1}{4} - 2 \times \frac{1}{8} - 2 \times \frac{1}{16} \qquad = -3.125$$

$$G_3 = -2 - 2 \times \frac{1}{2} - 2 \times \frac{1}{4} + 1 \times \frac{1}{8} - 2 \times \frac{1}{16} \dots \qquad = -3.41$$

$$G_4 = -2 \dots\dots\dots\dots\dots\dots\dots\dots \qquad = \cdots\dots..$$

……

$$V_{(1)} = \mathbb{E}(G)$$

贝尔曼公式（Bellman equation）

价值函数V^π的贝尔曼公式如下：

$$V^\pi(s)=\mathbb{E}^\pi[r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots|S(t)=s]=\mathbb{E}^\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k}|S(t)=s\right]$$

$$=\mathbb{E}^\pi\left[r_t+\gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|S(t)=s\right]$$

$$=\mathbb{E}^\pi[r_t+\gamma V^\pi(s_{t+1})|S(t)=s]$$

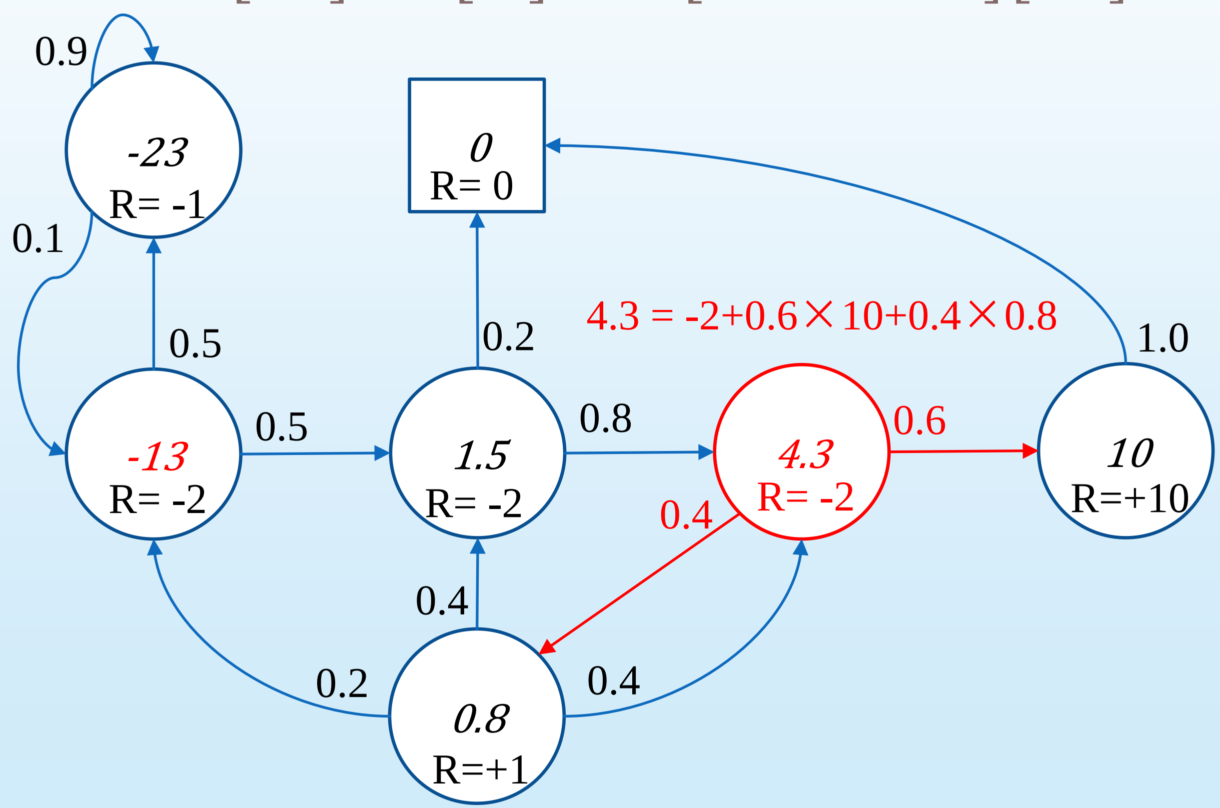
当前价值 = 当前即时奖励 + 后继状态的价值折扣

$$V=R+\gamma PV \qquad \begin{bmatrix} V_{(1)} \\ \vdots \\ V_{(n)} \end{bmatrix}=\begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix}+\gamma\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \cdots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}\begin{bmatrix} V_{(1)} \\ \vdots \\ V_{(n)} \end{bmatrix}$$

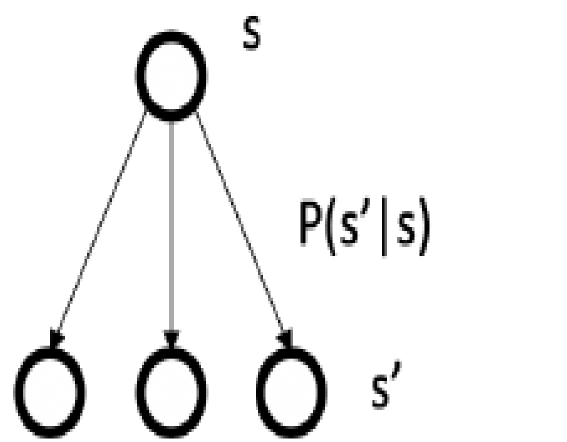
关于Bellman公式的理解（MRP）

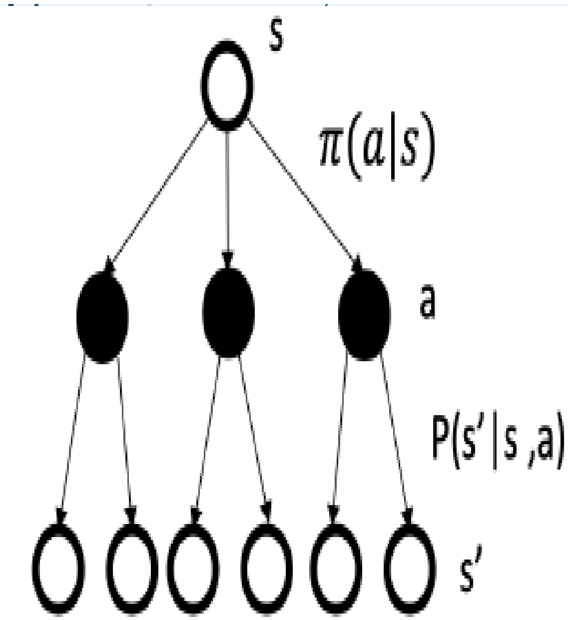
当前价值 = 当前即时奖励 + 后继状态的价值折扣

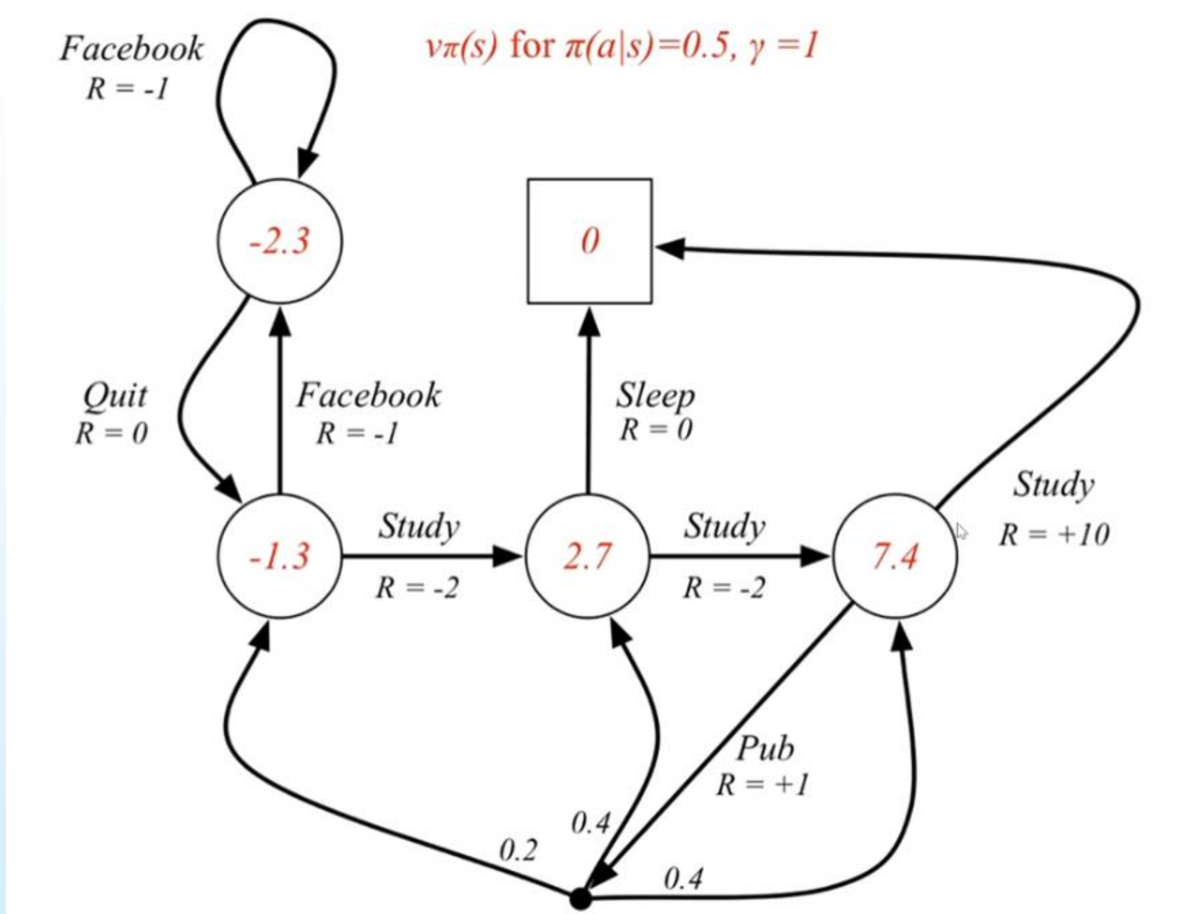
$$V = R + \gamma P V \qquad \begin{bmatrix} V_{(1)} \\ \vdots \\ V_{(n)} \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix} + \gamma \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \dots & \vdots \\ P_{n1} & \dots & P_{nn} \end{bmatrix} \begin{bmatrix} V_{(1)} \\ \vdots \\ V_{(n)} \end{bmatrix}$$



关于Bellman公式的理解（MDP）







$$v^{\pi}(s) = \sum_{a \in A} \pi(a|s)\Big(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v^{\pi}(s')\Big)$$

验证： 7.4 = 0.5(10+0) + 0.5(1+ 0.2(-1.3)+0.4*2.7+0.4*7.4)

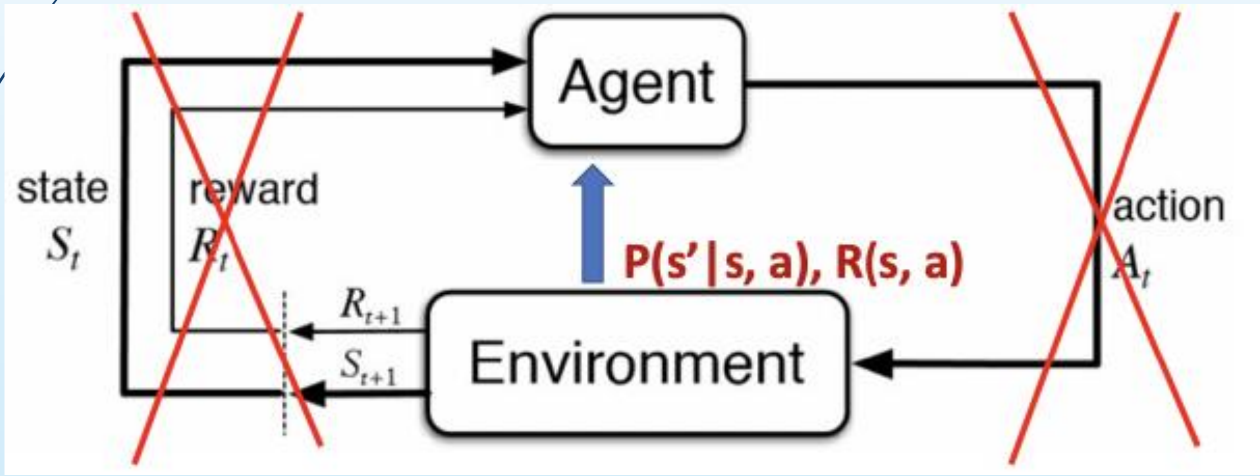
优化控制方法

给定一个马尔科夫决策过程MDP，要考虑的是如何计算一个最优策略 π^* 。

主要方法：基于模型（Model-based）、模型无关（Model-free）。

基于模型 vs 模型无关

动态规划 vs 强化学习



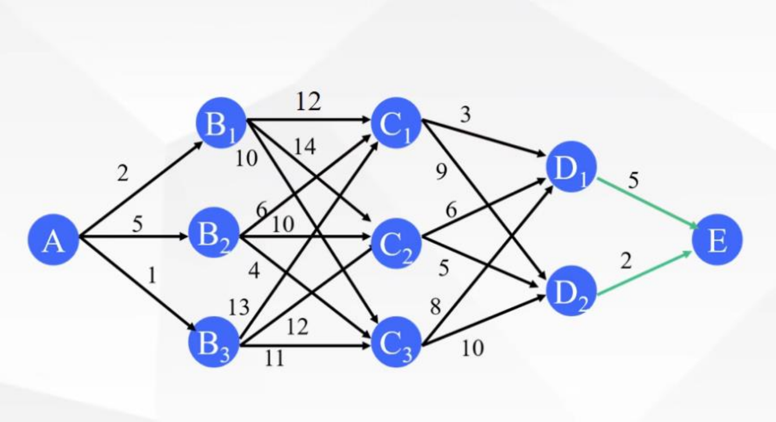
第 8 讲 决策理论规划

动态规划（Dynamic Programming）

1950年代初，美国数学家理查德·贝尔曼（Richard Bellman）在研究多步决策过程（multistep decision process）的优化问题时，将多步过程转化为一系列单步问题，利用各阶段之间的关系逐个加以解决，从而创立了动态规划理论（Theory of Dynamic Programming）。

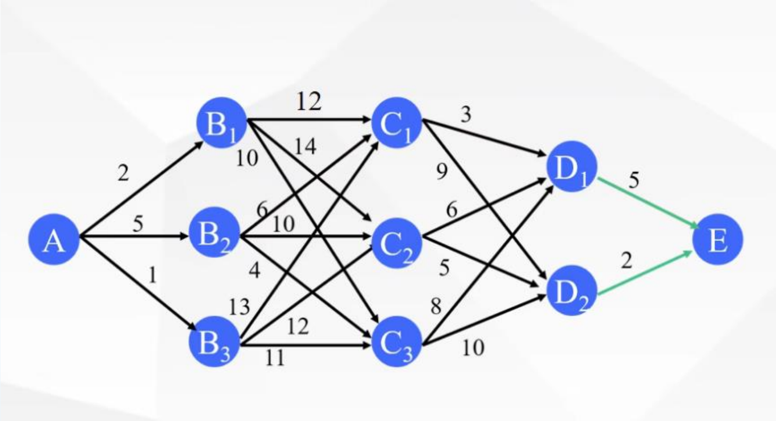
满足条件：

- 最优子结构（最优化原理）
- 子问题的重叠性
- 无后效性



动态规划（Dynamic Programming）

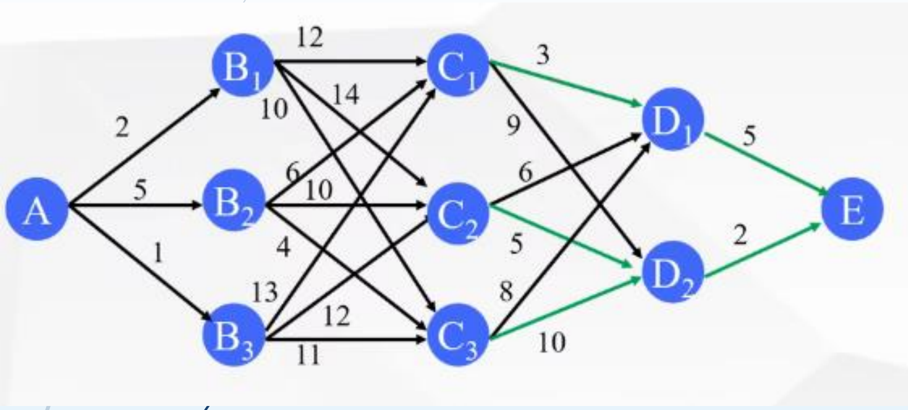
- 算法思想
- 逆向寻优，正向求解
- DP算法本质由三层循环构成：
  1. 第一层遍历每一个阶段；
  2. 第二层遍历第i个阶段的每一个状态；
  3. 第三层循环遍历第i+1个阶段的每一个状态。



第四阶段(D➞E)：D有两条路径到E

$f_4(D_1) = 5 \qquad f_4(D_2) = 2$

动态规划（Dynamic Programming）



第三阶段(C→D)：C到D有6条路线

其中C有3个状态，分别讨论经过

该状态的最优路线

经过C1：

$$f_3(C_1) = \min\begin{Bmatrix} d(C_1, D_1) + f_4(D_1) \\ d(C_1, D_2) + f_4(D_2) \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 3+5 \\ 9+2 \end{Bmatrix} = 8$$

最短路线为$C_1 \rightarrow D_1 \rightarrow E$

经过C2：

$$f_3(C_2) = \min\begin{Bmatrix} d(C_2, D_1) + f_4(D_1) \\ d(C_2, D_2) + f_4(D_2) \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 6+5 \\ 5+2 \end{Bmatrix} = 7$$

最短路线为$C_2 \rightarrow D_2 \rightarrow E$

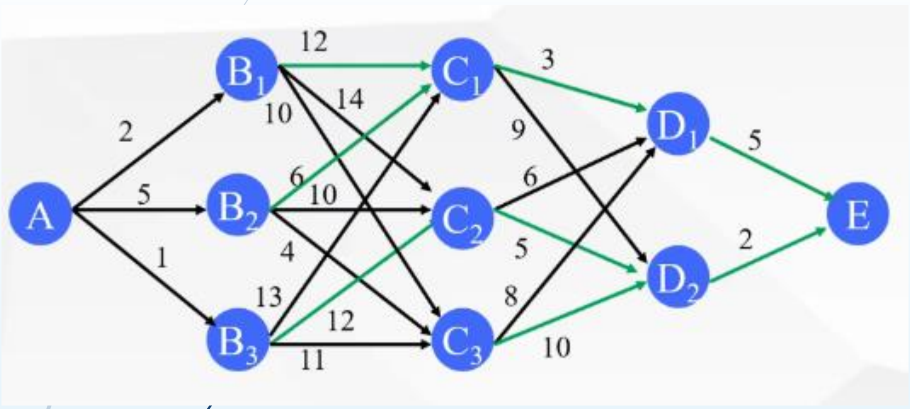
经过C3：

$$f_3(C_3) = \min\begin{Bmatrix} d(C_3, D_1) + f_4(D_1) \\ d(C_3, D_2) + f_4(D_2) \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 8+5 \\ 10+2 \end{Bmatrix} = 12$$

最短路线为$C_3 \rightarrow D_2 \rightarrow E$

动态规划（Dynamic Programming）



第二阶段(B→C)：B到C有9条路线

其中B有3个状态，分别讨论经过

该状态的最优路线

经过B1：

$$f_2(B_1)=\min\begin{Bmatrix} d(B_1,C_1)+f_3(C_1)\\ d(B_1,C_2)+f_3(C_2)\\ d(B_1,C_3)+f_3(C_3)\end{Bmatrix}$$

$$=\min\begin{Bmatrix}12+8\\14+7\\10+12\end{Bmatrix}=20$$

最短路线为$B_1\to C_1\to D_1\to E$

经过B2：

$$f_2(B_2)=\min\begin{Bmatrix} d(B_2,C_1)+f_4(C_1)\\ d(B_2,C_2)+f_4(C_2)\\ d(B_2,C_3)+f_4(C_3)\end{Bmatrix}$$

$$=\min\begin{Bmatrix}6+8\\10+7\\4+12\end{Bmatrix}=14$$

最短路线为$B_2\to C_1\to D_1\to E$

经过B3：

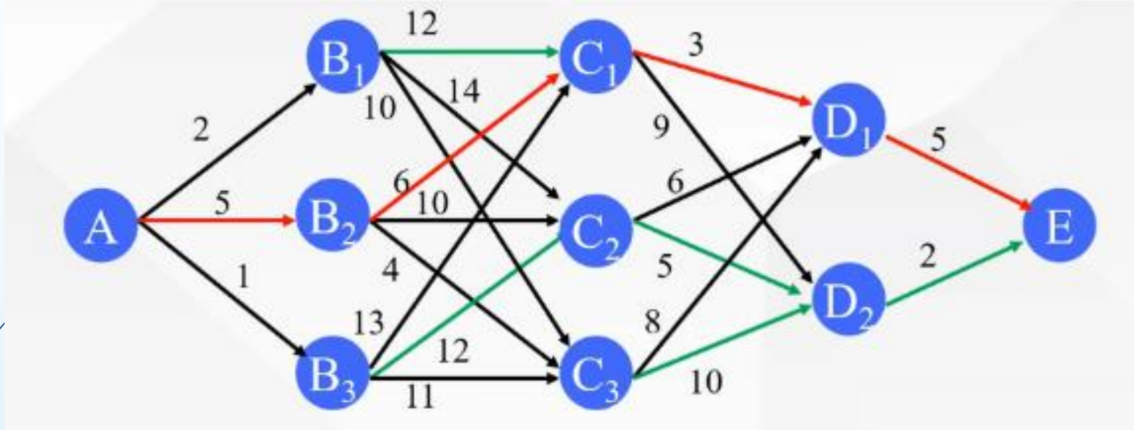
$$f_2(B_3)=\min\begin{Bmatrix} d(B_3,C_1)+f_3(C_1)\\ d(B_3,C_2)+f_3(C_2)\\ d(B_3,C_3)+f_3(C_3)\end{Bmatrix}$$

$$=\min\begin{Bmatrix}13+8\\12+7\\11+12\end{Bmatrix}=19$$

最短路线为$B_3\to C_2\to D_2\to E$

动态规划（Dynamic Programming）

核心思想：最优策略的子策略也必然是最优的



第一阶段(A→B)：B到C有3条路线

经过A：

$$f_1(A)=\min\begin{Bmatrix}d(A,B_1)+f_2(B_1)\\d(A,B_2)+f_2(B_2)\\d(A,B_3)+f_2(B_3)\end{Bmatrix}=\min\begin{Bmatrix}2+20\\5+14\\1+19\end{Bmatrix}=19$$

最短路线为$A\to B_2\to C_1\to D_1\to E$

动态规划（Dynamic Programming）

在决策理论规划中，动态规划被用于对马尔科夫决策过程进行优化控制，计算马尔科夫决策过程的最优策略。

动态规划的两个核心方法：策略迭代（Policy iteration）和价值迭代（Value iteration ）。

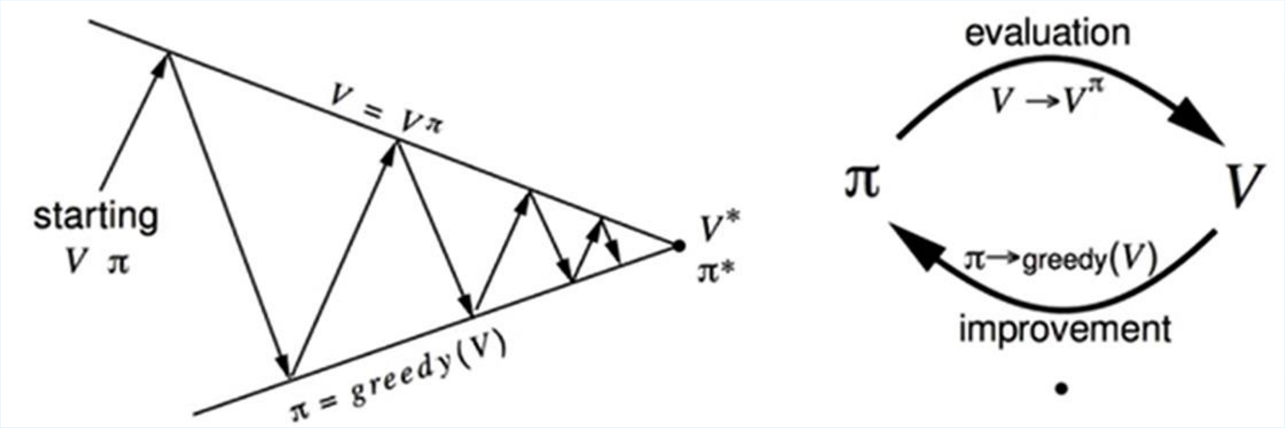
策略迭代（Policy iteration）

策略迭代算法：

1）策略评估（policy evaluation），计算当前策略的价值函数；

2）策略改进（policy improvement），通过价值函数的最大化来计算改善的策略；

3）重复上述操作，直到收敛于一个最优策略。



策略评估（policy evaluation）举例

s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

1. $R = [5, 0, 0, 0, 0, 0, 10]$
2. Practice 1: Deterministic policy $\pi(s) = Left$ with $\gamma = 0.5$ for any state s, then what are the state values under the policy?
3. Practice 2: Stochastic policy $P(\pi(s) = Left) = 0.5$ and $P(\pi(s) = Right) = 0.5$ and $\gamma = 0.5$ for any state s, then what are the state values under the policy?
4. Iteration t:
$v_t^{\pi}(s) = \sum_a P(\pi(s) = a)(r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v_{t-1}^{\pi}(s'))$

策略评估（policy evaluation）举例 Practice1

同步动态规划：同时计算整个state set的value，计算成本高；
异步动态规划：每个状态更新都用最新值；有选择性地更新一些状态的value，不是更新整个状态集。

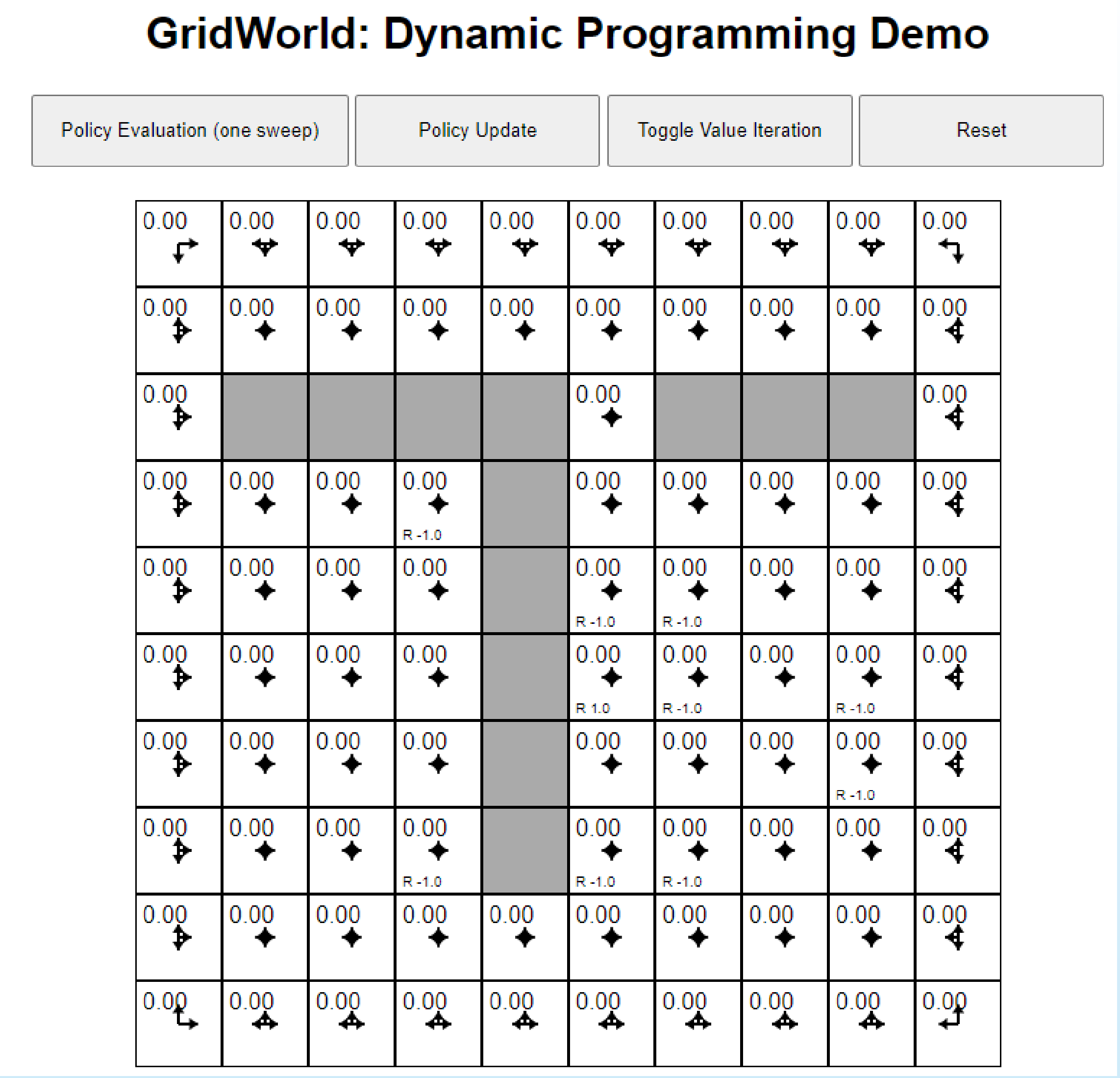
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Step	$V(S_1)$	$V(S_2)$	$V(S_3)$	$V(S_4)$	$V(S_5)$	$V(S_6)$	$V(S_7)$
	5	0	0	0	0	0	10
1	5	$0+5\times0.5$	0	0	0	0	10
2	5	5×0.5	$5\times0.5\times0.5$	0	0	0	10
3	5	5×0.5	$5\times0.5\times0.5$	5×0.5^3	0	0	10
4	5	5×0.5	5×0.5^2	5×0.5^3	5×0.5^4	0	10
5	5	5×0.5	5×0.5^2	5×0.5^3	5×0.5^4	5×0.5^5	10
6	……						$10+5\times0.5^6$
7	……						
8 直到收敛	……						

$$v_t^{\pi}(s)=\sum_a P(\pi(s)=a)\big(r(s,a)+\gamma\sum_{s'\in S}P(s'|s,a)v_{t-1}^{\pi}(s')\big)$$

异步：$[5, 5\times0.5, 5\times0.5^2, 5\times0.5^3, 5\times0.5^4, 5\times0.5^5, 10+5\times0.5^6]$

策略评估 demo

https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_dp.html



策略改进（policy improvement）

首先，使用如下等式来识别所有动作的价值：

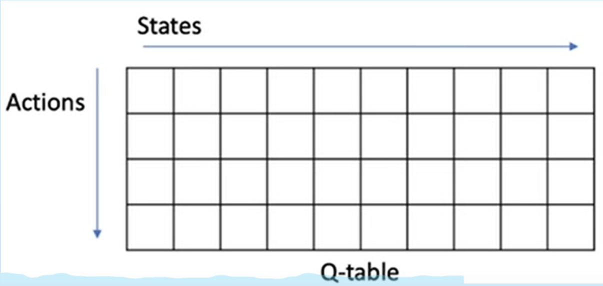
$$Q^{\pi}(s,a) = \mathbb{E}^{\pi}[r_t + \gamma V_k^{\pi}(s_{t+1})|\mathrm{S}(t) = s, \mathrm{A}(t) = a]$$

$$= \sum_{s'} T(s,a,s')\left(R(s,a,s') + \gamma V^{\pi}(s')\right)$$

若对于某个动作$a \in \mathrm{A}$，存在$Q^{\pi}(s,a)$大于$V^{\pi}(s)$时，则直接选择动作a会更好，而不必用当前的$\pi(s)$。事实上，可以评估所有状态下的所有动作，并且选择所有状态下的最佳动作。

$$\pi'(s) = \arg\max_a Q^{\pi}(s,a) = \arg\max_a \mathbb{E}(r_t + \gamma V^{\pi}(s_{t+1})|\mathrm{S}(t) = s, \mathrm{A}(t) = a)$$

$$= \arg\max_a \sum_{s'} T(s,a,s')\left(R(s,a,s') + \gamma V^{\pi}(s')\right)$$



价值迭代（Value iteration）

价值迭代算法实时计算必要的更新，将策略评估与策略改进结合在一起，没必要等待全面收敛，而是提前停止评估，根据当前评估改进其策略。

- 对于每个状态，初始化 $V(S) = 0$
- 重复循环取max
- 价值函数收敛时，输出最优策略

价值迭代（Value iteration）举例

- 对于每个状态，初始化 V(S) =0
- 重复循环取max $V^*(S) = max_a[R_S^a + \gamma \sum_{s'} P_{ss'}^a V^*(S')]$
- 价值函数收敛时，输出最优策略

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

作业： Practice2

s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

1. $R = [5, 0, 0, 0, 0, 0, 10]$
2. Practice 1: Deterministic policy $\pi(s) = Left$ with $\gamma = 0.5$ for any state s, then what are the state values under the policy?
3. Practice 2: Stochastic policy $P(\pi(s) = Left) = 0.5$ and $P(\pi(s) = Right) = 0.5$ and $\gamma = 0.5$ for any state s, then what are the state values under the policy?
4. Iteration t:
$v_t^{\pi}(s) = \sum_a P(\pi(s) = a)(r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) v_{t-1}^{\pi}(s'))$