第十三讲：强化学习基础

最优控制的智能方法之三

张杰

人工智能学院
中国科学院大学

复杂系统管理与控制国家重点实验室
中国科学院自动化研究所

2017 年 10 月 31 日

Table of Contents

Table of Contents

最优控制问题

问题 (最优控制问题)

1. 被控对象的状态方程为

$$\dot{x}(t) = f(x(t), u(t), t),\ x(t_0) = x_0.$$

2. 容许控制, $u \in U$
3. 目标集, $x(t_f) \in S$
4. 最小化性能指标

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt.$$

离散时间最优控制问题

问题 1 (离散时间最优控制问题)

状态变量为 $x(k):\mathbb{N}\to\mathbb{R}^n$，控制变量为 $u(k):\mathbb{N}\to\mathbb{R}^m$

(1) 被控对象的状态方程

$$x(k+1)=f_D(x(k),u(k),k),\quad k=0,\ldots,N-1. \tag{1}$$

(2) 容许控制：

$$u(k)\in U,\quad x(k)\in X. \tag{2}$$

(3) 目标集：

$$x(N)\in\mathcal{S}. \tag{3}$$

(4) 性能指标：

$$J(u;x(k),k)=h_D(x(N),N)+\sum_{i=k}^{N-1}g_D(x(i),u(i),i). \tag{4}$$

动态规划方法

离散：Bellman 方程，

$$V(x(k),k) = \min_{u(k)\in U}\{g_D(x(k),u(k),k) + V(x(k+1),k+1)\}$$

$$k = k_0,\ldots,N-1 \tag{5}$$

$$V(x(N),N) = h_D(x(N),N). \tag{6}$$

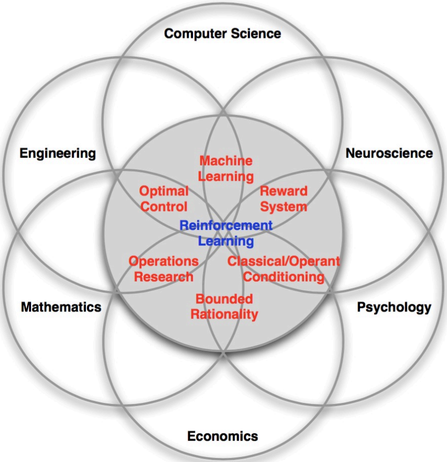
连续：HJB 方程，

$$-\frac{\partial V}{\partial t}(x(t),t) = \min_{u(t)\in\mathbb{R}^m}\mathcal{H}(x(t),u(t),\frac{\partial V}{\partial x}(x(t),t),t), \tag{7}$$

$$V(x(t_f),t_f) = h(x(t_f),t_f). \tag{8}$$

Table of Contents

Many Faces of Reinforcement Learning

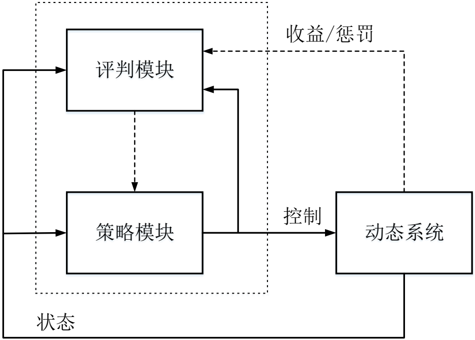


强化学习（Reinforcement Learning）在多个学科中都有着广泛的应用

- 计算机科学
- 工程学
- 数学
- 经济学
- 心理学
- 神经科学

作为控制问题和机器学习问题的强化学习

- 作为控制问题，强化学习研究最优控制问题，考察已知或未知的动态系统，确定或不确定的环境
- 作为机器学习问题，强化学习不同于有监督学习与无监督学习，没有“标注”，仅有“奖励”rewards



rewards

- t 时刻的 reward $R_t \in \mathbb{R}$ 是反馈信号
- “表示” 智能体/被控对象 t 时刻做的多好
- 智能体/被控对象的任务是最大化收益或最小化损失

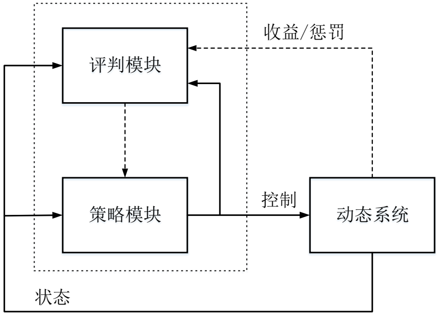
Remark 1

最大化收益 v.s. 最小化损失

Remark 2

强化学习问题可认为是自由终端状态的最优控制问题

智能体与环境



任意时刻 $t \in [t_0, t_f]$，智能体

- 实施行动 (控制) A_t
- 获得观测 (状态) O_t
- 获得奖励 R_t

环境

- 获得行动 A_t
- 时间增加 $t + \Delta t$
- 发出观测值 $O_{t+\Delta t}$
- 发出奖励 $R_{t+\Delta t}$

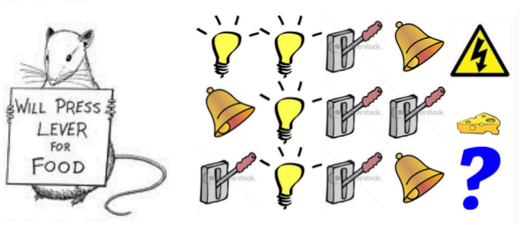
历史与状态

- 历史是观测、行动和奖励的序列

$$H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$$

- 假定:
  - 智能体依据历史决定行动
  - 环境根据历史决定观测和奖励
- 状态 是能决定未来将发生什么的信息
  - 环境状态
  - 智能体状态

一个例子：老鼠的决策依据？



- 最近三个物品作为状态？
- 灯、铃、控制杆在历史上发生的次数作为状态？
- 全部历史作为状态？

Markov 状态

定义 1 (Markov state)

S_t 是 Markov的当且仅当

$$P(S_{t+1}|S_t) = P(S_{t+1}|S_1, \ldots, S_t) \tag{9}$$

- 对 Markov 的状态，给定当前状态时，未来不依赖于过去
- 若智能体能直接观测全部环境状态，称为完全可观测，否则称为部分可观测

智能体的基本要素

在强化学习中，智能体可能具有如下要素

- 策略 Policy：建模智能体的行为
- 值函数 Value function：建模智能体对状态和/或控制的估值
- 模型 Model：智能体对环境的表示 representation

Policy (控制策略，控制律)

控制策略policy从状态到控制的映射。本课考察稳态策略

- 确定策略

$$a = \pi(s)$$

- 随机策略

$$\pi(a|s) = P(A_t = a|S_t = s)$$

Value Function （值函数）

一个控制策略的值函数value function定义为期望累积收益

$$v_\pi(s) = E_\pi(R_{t+1} + R_{t+2} + \ldots | S_t = s)$$

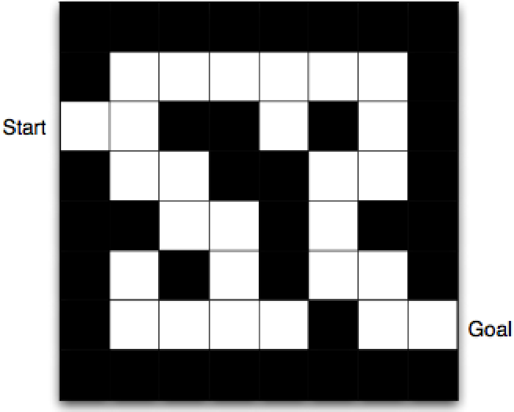
- 一个控制策略的值函数用于估计这个策略下特定状态的优劣
- 同时也是对这个策略的评价
- 最优值函数，或简称值函数，是任意策略的值函数的极大值

Model （模型）

智能体用模型model估计下一时刻的环境状态和奖励，对于未知的随机系统常用状态转移矩阵和期望收益表示

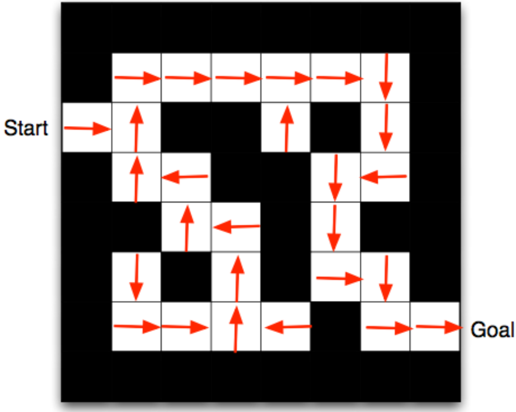
$$\mathcal{P}(s,a,s') = P(S_{t+1}=s'|S_t=s, A_t=a)$$
$$\mathcal{R}(s,a) = E(R_{t+1}|S_t=s, A_t=a)$$

例子：一个简单的迷宫



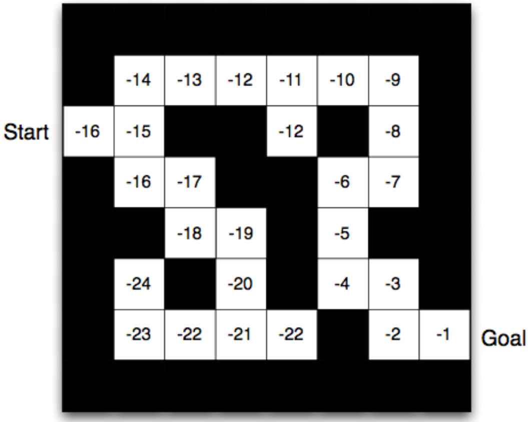
- 奖励：每步 -1
- 备选行动: 上下左右
- 状态: 智能体的位置

例子：确定策略



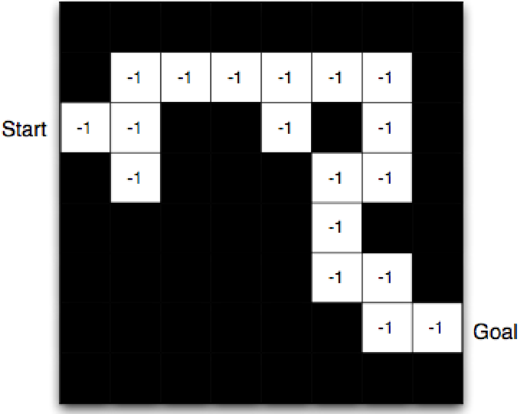
- 确定性策略 $\pi(s)$ 对任意状态 s 都以 100% 概率选择某控制

例子：策略的值函数



- 上述策略的值函数 $v_\pi(s)$

例子：模型



- 智能体对环境的建模未必完全

Exploration (探索) and Exploitation (开发)

Remark 3

强化学习的基本想法是在“试错” trial-and-error 中学习。智能体从环境获得的经验中发现好的策略，尽量避免因此带来过多损失

- 探索获取更多环境信息
- 开发最大化期望累积收益

例 1

每次都去最喜欢的那个窗口吃饭，或尝试一个新窗口

预测和控制

强化学习中两个基本任务:

- 预测（prediction）：评价一个策略（的值函数）
- 控制（control）：寻求最优策略

Markov Decision Processes (MDPs)

Markov 过程：给定当前状态时，未来不依赖于过去

$$P(S_{t+1}|S_t) = P(S_{t+1}|S_1, \ldots, S_t) \tag{10}$$

Markov 过程 + rewards + actions = Markov 决策过程

状态转移矩阵 State Transition Matrix

当前时刻状态 s 下时刻状态 s'，状态转移概率定义为

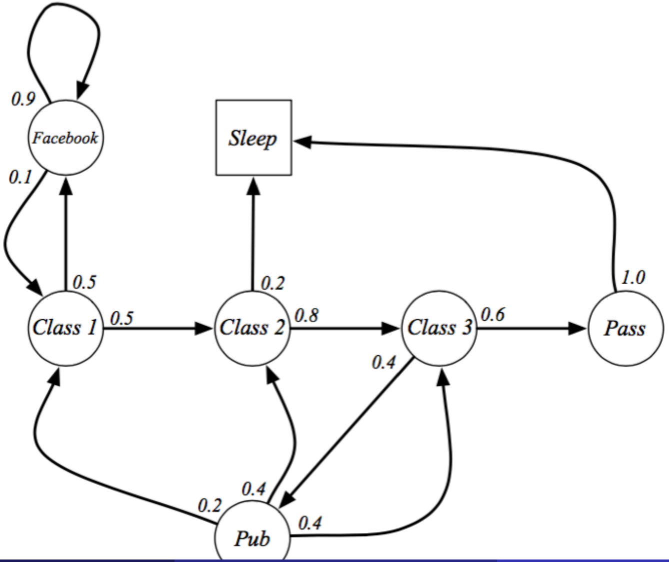
$$\mathcal{P}(s,s^{'}) = P(S_{t+1} = s^{'}|S_t = s) \tag{11}$$

状态转移矩阵 $\mathcal{P}$ 定义当前时刻状态-i s 到下时刻状态-j s' 的概率

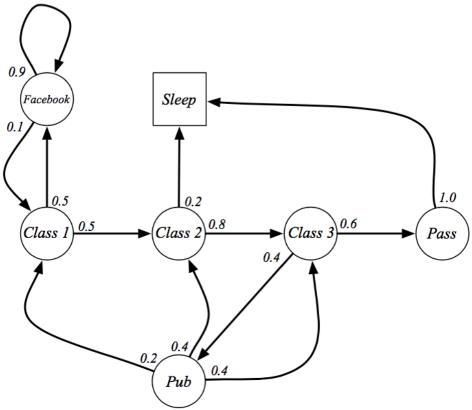
$$\mathcal{P} = \begin{bmatrix} \mathcal{P}(1,1) & \dots & \mathcal{P}(1,n) \\ \vdots & \mathcal{P}(i,j) & \vdots \\ \mathcal{P}(n,1) & \dots & \mathcal{P}(n,n) \end{bmatrix} \tag{12}$$

每行加和等于 1

例子：一个学生的 Markov 链



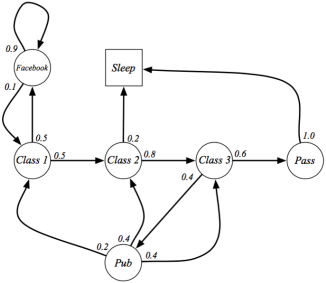
例子：采样



依 Markov 过程可从初始状态 $S_1 = C_1$ 到终端状态 $S_T =$“Sleep” 采样

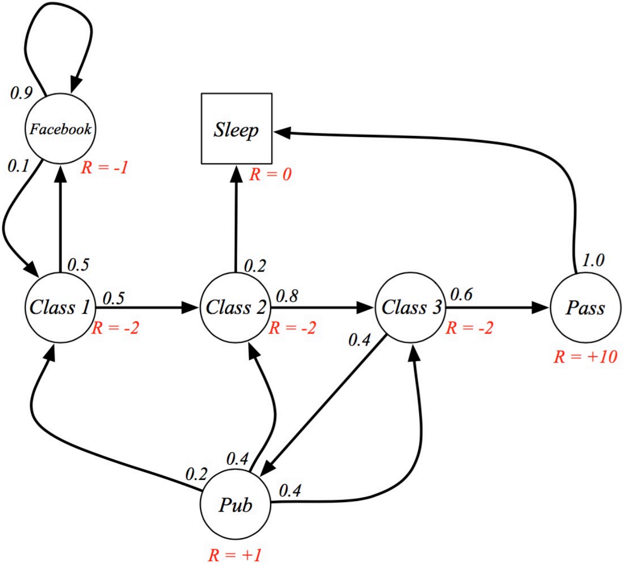
- C_1, C_2, C_3, Pass, Sleep
- C_1, FB, FB, C_1, C_2, Sleep
- C_1, C_2, C_3, Pub, C_2, C_3, Pass, Sleep
- C_1, FB, FB, C_1, C_2, C_3, Pub, C_1, FB, FB, FB C_1, C_2, C_3, Pub, C_2, Sleep

例子：Markov 过程的状态转移矩阵



$$
\begin{array}{c}
\\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\
\left[\begin{array}{ccccccc}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & & 1.0 \\
0.2 & 0.4 & 0.4 & & & & \\
0.1 & & & & & 0.9 & \\
 & & & & & & 1
\end{array}\right]
\end{array}
$$

例子：Markov 过程 + rewards



总收益 Return

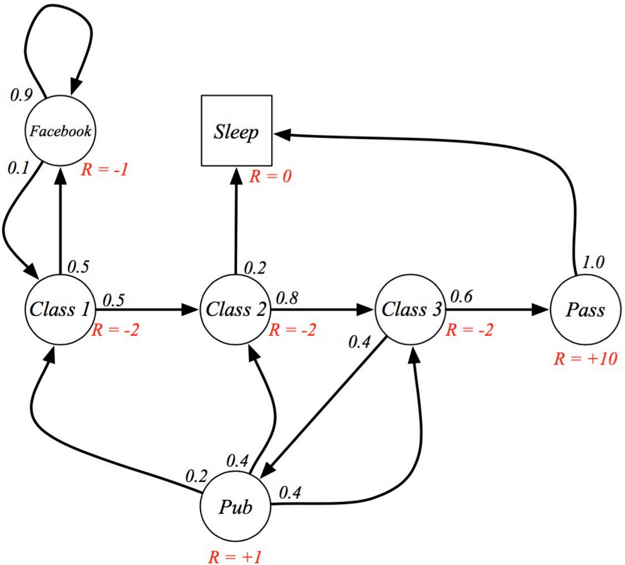
定义 2 (总收益 Return)

总收益定义为

$$G_t := R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \tag{13}$$

- γ 表示折现
- $\gamma = 0$ “myopic” 近视评价
- $\gamma = 1$ “far-sighted” 远见评价

例子：Markov 过程 + rewards



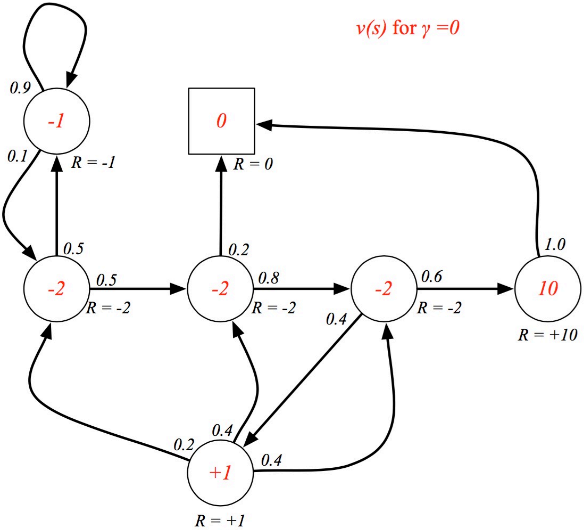
例子：计算一下总收益

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

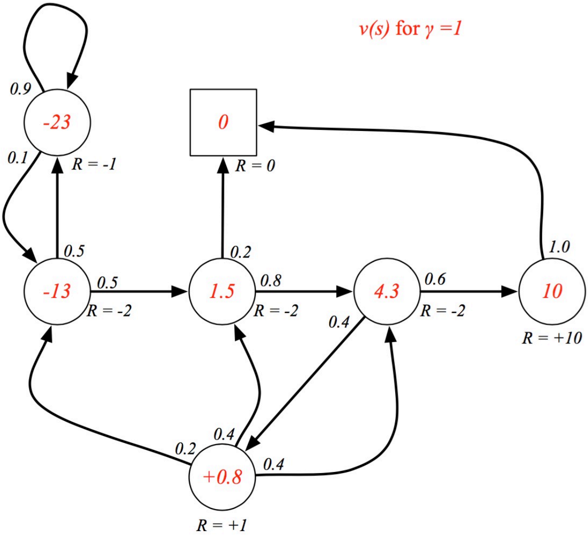
$$G_1 = R_2 + \gamma R_3 + \ldots + \gamma^{T-2} R_T$$

C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	$=$	-2.25
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	$=$	-3.125
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	$=$	-3.41
C1 FB FB C1 C2 C3 Pub C1 ...	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$		
FB FB FB C1 C2 C3 Pub C2 Sleep		$=$	-3.20

Example: State-Value Function - 1



Example: State-Value Function - 2



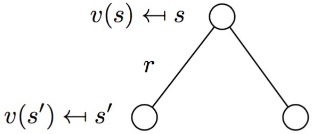
“没有控制”的 Bellman 方程

值函数 $v(s) = E[G_t|S_t = s]$ 可被分解成即刻的奖励和未来的折现奖励两部分

$$\begin{aligned} v(s) &= E[G_t|S_t = s] \\ &= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots |S_t = s] \\ &= E[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots)|S_t = s] \\ &= E[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= E[R_{t+1} + \gamma v(S_{t+1})|S_t = s] \end{aligned}$$

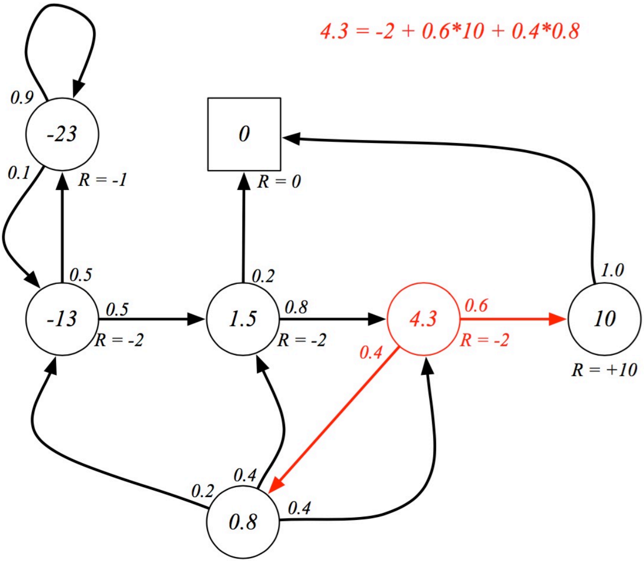
“没有控制”的 Bellman 方程：条件概率

$$
\begin{aligned}
v(s) &= E[R_{t+1} + \gamma v(S_{t+1}) | S_t = s] \\
\mathcal{R}(s) &= E[R_{t+1} | S_t = s] \\
\mathcal{P}(s, s') &= P(S_{t+1} = s' | S_t = s)
\end{aligned}
$$



$$
v(s) = \mathcal{R}(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s, s') v(s') \tag{14}
$$

例子:Bellman 方程, $\mathcal{R}(s) + \gamma \sum \mathcal{P}(s, s')v(s')$



矩阵形式的 Bellman 方程

$$v(s) = \mathcal{R}(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s, s') v(s')$$

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

其中 v 写成列向量形式

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}(1) \\ \vdots \\ \mathcal{R}(n) \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}(1,1) & \dots & \mathcal{P}(1,n) \\ \vdots & \mathcal{P}(i,j) & \vdots \\ \mathcal{P}(n,1) & \dots & \mathcal{P}(n,n) \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

$$\begin{aligned} v &= \mathcal{R} + \gamma \mathcal{P} v \\ (I - \gamma \mathcal{P}) v &= \mathcal{R} \\ v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R} \end{aligned}$$

Policy (控制策略，控制律)

Markov 过程 + rewards + actions = Markov 决策过程

控制策略用分布函数表示

$$\pi(a|s) = P(A_t = a|S_t = s) \tag{15}$$

依赖于当前状态 $S_t = s$，闭环策略

奖励和转移矩阵

Remark 4

给定一个 Markov 决策过程和一个策略 π,

- 则状态序列 $S_1, S_2, \ldots,$ 是 Markov 过程
- 转移矩阵和期望的奖励为

$$\mathcal{P}_\pi(s, s^{'}) = \sum_{a\in\mathcal{A}} \pi(a|s)\mathcal{P}(s, a, s^{'}) \tag{16}$$

$$\mathcal{R}_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\mathcal{R}(s, a) \tag{17}$$

State-Value Function (状态值函数) and Action-Value Function (行动值函数)

定义 3 (状态值函数和行动值函数)

给定一个 Markov 决策过程，策略 π 的状态值函数 $v_\pi(s)$ 定义为以 s 为初始状态，使用控制策略 π 的总期望收益

$$v_\pi(s) = E_\pi[G_t | S_t = s]. \tag{18}$$

给定一个 Markov 决策过程，策略 π 的行动值函数 $q_\pi(s,a)$ 定义为以 s 为初始状态，以 a 为立即实施的行动，以 π 为接下来使用的控制策略，所获总期望收益

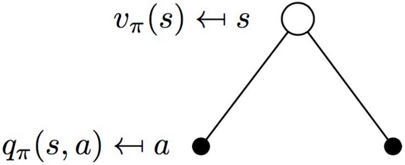
$$q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]. \tag{19}$$

Bellman 期望方程

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s]. \tag{20}$$

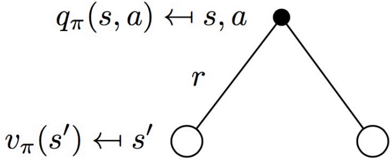
$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]. \tag{21}$$

Bellman 期望方程 $v - q$



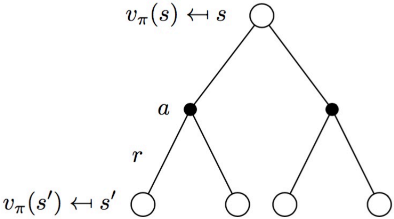
$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a) \tag{22}$$

Bellman 期望方程 $q - v$



$$q_\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') v_\pi(s') \tag{23}$$

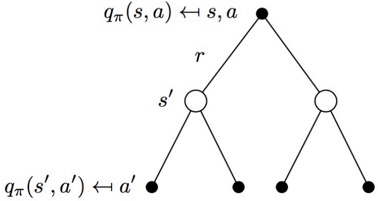
Bellman 期望方程 $v - v$



$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s]. \tag{24}$$

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s)(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s')v_\pi(s')) \tag{25}$$

Bellman 期望方程 $q-q$

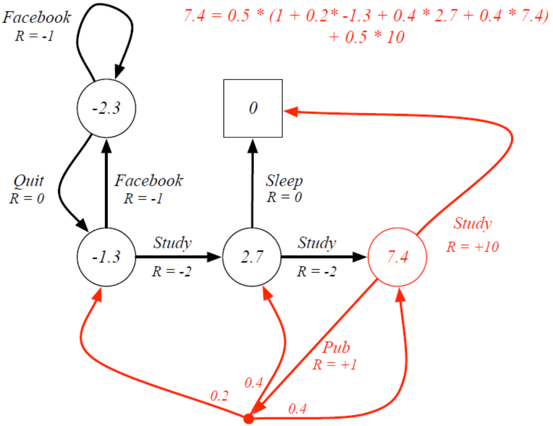


$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]. \tag{26}$$

$$q_\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s',a') \tag{27}$$

例子：Bellman 期望方程 $v - v$

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s)\left(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s')v_\pi(s')\right)$$



Bellman 期望方程矩阵形式

$$v_\pi = \mathcal{R}_\pi + \gamma \mathcal{P}_\pi v_\pi \tag{28}$$

可解得

$$v_\pi = (I - \gamma \mathcal{P}_\pi)^{-1} \mathcal{R}_\pi \tag{29}$$

和 Markov 过程类似，当矩阵较大时，难以计算

最优值函数

定义 4 (值函数/最优值函数)

状态值函数 $v_*(s)$ 定义为所有策略的状态值函数的最大值

$$v_*(s) = \max_{\pi} v_\pi(s) \tag{30}$$

行动值函数 $q_*(s,a)$ 定义为所有策略的行动值函数的最大值

$$q_*(s,a) = \max_{\pi} q_\pi(s,a) \tag{31}$$

Markov 决策过程的最优策略

定理 1

对于任意 Markov 决策过程，

- *存在最优策略 π_*，*

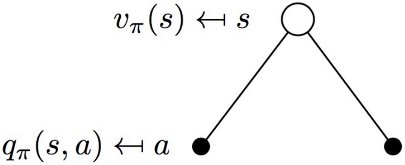
$$v_{\pi_*}(s) = v_*(s)$$
$$q_{\pi_*}(s,a) = q_*(s,a)$$

- *且*

$$\pi_*(a|s) = \begin{cases} 1 & \mathit{if}\ a = \operatorname{argmax}_{a\in A} q_*(s,a), \\ 0 & \mathit{otherwise}. \end{cases} \tag{32}$$

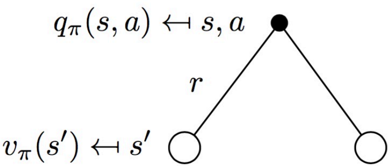
- *Markov 决策过程总有确定性的最优策略*

Bellman 方程 $v - q$



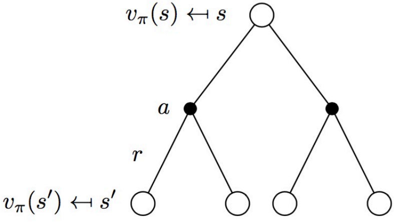
$$
\begin{aligned}
v_\pi(s) &= \sum_{a\in\mathcal{A}} \pi(a|s) q_\pi(s,a) \\
v_*(s) &= \max_{a\in\mathcal{A}} q_*(s,a)
\end{aligned}
\tag{33}
$$

Bellman 方程 $q - v$



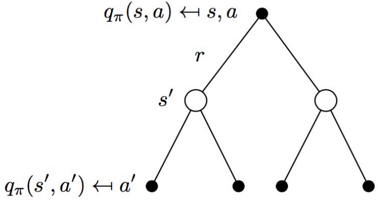
$$
\begin{aligned}
q_\pi(s,a) &= \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') v_\pi(s') \\
q_*(s,a) &= \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') v_*(s')
\end{aligned} \tag{34}
$$

Bellman 方程 $v-v$



$$
\begin{aligned}
v_\pi(s) &= \sum_{a\in\mathcal{A}} \pi(a|s)(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s,a,s')v_\pi(s')) \\
v_*(s) &= \max_{a\in\mathcal{A}} [\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s,a,s')v_*(s')]
\end{aligned}
\tag{35}
$$

Bellman 方程 $q - q$



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

$$q_*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') \max_{a' \in \mathcal{A}} q_*(s',a') \tag{36}$$

求解 Bellman 方程

- Bellman 方程是非线性的，一般情况下没有解析解
- 使用迭代方法
    - 值迭代
    - 策略迭代
    - Q-学习

- Prediction
    - Input: MDP $<\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma>$ 状态空间、控制空间、转移概率、奖励函数、折现。策略 π
    - Output: 策略 π 的状态值函数 v_π
- Control
    - Input: MDP $<\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma>$
    - Output: 最优策略 π^*（和值函数 v^*）

Table of Contents

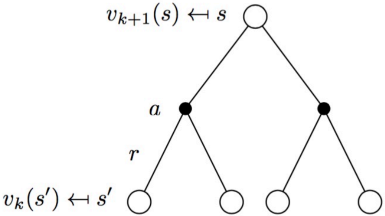
迭代策略评估 1/2

- 任务: 计算策略 π 的总期望收益
- 解: 迭代利用 Bellman 期望方程

$$v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_\pi$$

- 在迭代 $k+1$
- 对任意状态 $s \in S$
- 根据 $v_k(s')$ 更新 $v_{k+1}(s)$

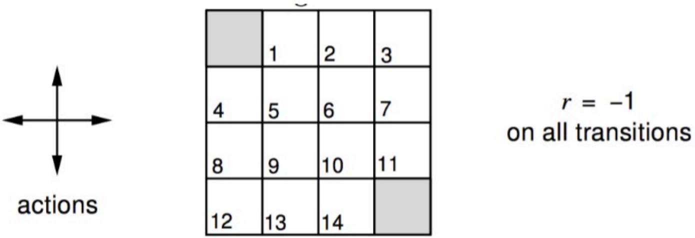
迭代策略评估 2/2



$$v_{k+1}(s) = \sum_{a\in\mathcal{A}} \pi(a|s)(\mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s,a,s')v_k(s'))$$

$$v_{k+1} = \mathcal{R}_\pi + \gamma \mathcal{P}_\pi v_k$$

例子：Grid-world 评估一个策略 1/3



- $\gamma = 1$
- 两个灰色是终端状态
- 跳出方框将保持状态不变
- 除了终端无收益，Reward 总是 -1
- 策略：

$$\pi(n|\cdot) = \pi(e|\cdot) = \pi(s|\cdot) = \pi(w|\cdot) = 0.25$$

例子：Grid-world 评估一个策略 2/3

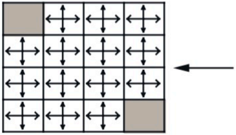
初始化策略的状态值函数

v_k for the Random Policy

Greedy Policy w.r.t. v_k

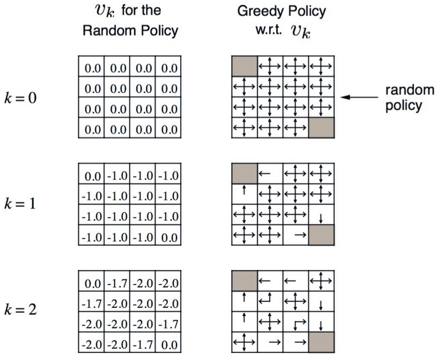
$k = 0$

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



若该值函数为“真”，依此可得右侧的贪婪策略

例子：Grid-world 评估一个策略 3/3



$k=0$

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

$k=1$

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

以 $k=1$，1 行 2 列为例

$$1/4[-1] + 1/4[-1+0]*3 = -1.0$$

$$k = 2$$

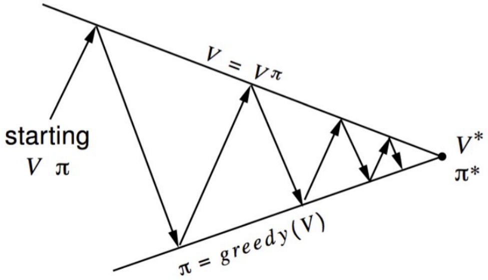
$$1/4*[-1] + 1/4[-1-1]*3 = -1.75$$

$$1/4*[-1-1]*4 = -2$$

$$v_{k+1}(s) = \sum_{a\in\mathcal{A}} \pi(a|s)(\mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s,a,s')v_k(s'))$$

Table of Contents

Policy Iteration，策略迭代

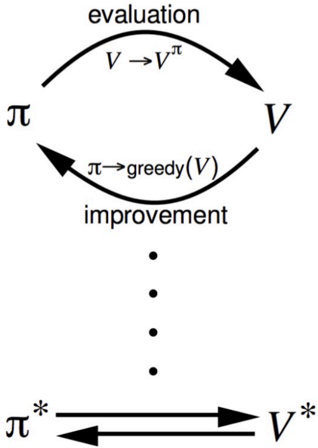


Policy evaluation Estimate v_π

Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$

Greedy policy improvement



策略改进 1/2

- 考察一个确定性策略 $a=\pi(s)$，其值函数为 $q_\pi(s,a), v_\pi(s)$
- 定义贪婪策略

$$\pi'(s) = \operatorname*{argmax}_{a\in A} q_\pi(s,a)$$

$$q_\pi(s,\pi'(s)) = \max_{a\in A} q_\pi(s,a) \geq q_\pi(s,\pi(s)) = v_\pi(s)$$

于是

$$\begin{aligned}
v_\pi(s) \leq q_\pi(s,\pi'(s)) &= E_{\pi'}(R_{k+1} + \gamma v_\pi(S_{k+1})|S_k = s) \\
&\leq E_{\pi'}(R_{k+1} + \gamma q_\pi(S_{k+1},\pi'(S_{k+1}))|S_k = s) \\
&\leq E_{\pi'}(R_{k+1} + \gamma R_{k+2} + \gamma^2 q_\pi(S_{k+2},\pi'(S_{k+2}))|S_k = s) \\
&\leq E_{\pi'}(R_{k+1} + \gamma R_{k+2} + \dots|S_k = s) = v_{\pi'}(s)
\end{aligned}$$

改进停止 2/2

若策略改进停止，即对于任意状态，

$$q_\pi(s,\pi'(s)) = \max_{a\in A} q_\pi(s,a) = q_\pi(s,\pi(s)) = v_\pi(s)$$

则 Bellman 方程已经满足

$$v_\pi(s) = \max_{a\in A} q_\pi(s,a)$$

即，$v_\pi(s) = v^*(s), \forall s \in S$. π 最优

Table of Contents

值迭代 1/2

- 任务: 求解最优策略 π (或值函数 v)
- 解: 迭代利用 Bellman 方程

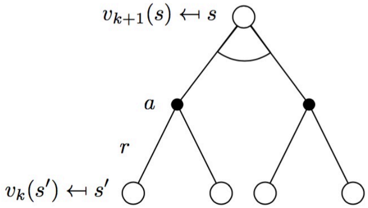
$$v_1 \to v_2 \to \ldots \to v^*$$

- 在迭代 $k+1$
- 对任意状态 $s \in S$
- 根据 $v_k(s')$ 更新 $v_{k+1}(s)$

Remark 5

解得过程中并无显式策略，v_k 也不是某个策略的值

值迭代 2/2



$$v_{k+1}(s) = \max_{a \in \mathcal{A}} [\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') v_k(s')]$$

$$v_{k+1} = \max_{a \in \mathcal{A}} [\mathcal{R}_a + \gamma \mathcal{P}_a v_k]$$

例子：最短路

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

小结

- 强化学习中的预测和控制
- 动态规划拓展至随机系统
- 动态系统已知
- 下节课：动态系统未知