第十六章：强化学习

纲要

- 强化学习问题基本设置
- K-摇臂赌博机
- 有模型学习
- 免模型学习
- 值函数近似
- 模仿学习

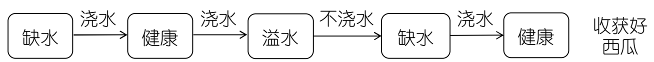
纲要

- 强化学习问题基本设置
- K-摇臂赌博机
- 有模型学习
- 免模型学习
- 值函数近似
- 模仿学习

强化学习问题基本设置

- 例子：瓜农种西瓜

种下瓜苗后：（为简便，仅考虑浇水和不浇水两个动作，不考虑施肥、除草等 ）



强化学习问题基本设置

- 例子：瓜农种西瓜
  - 多步决策过程
  - 过程中包含状态、动作、反馈 (奖赏) 等
  - 需多次种瓜，在过程中不断摸索，才能总结出较好的种瓜策略

种下瓜苗后：(为简便，仅考虑浇水和不浇水两个动作，不考虑施肥、除草等)

缺水 $\xrightarrow{\text{浇水}}$ 健康 $\xrightarrow{\text{浇水}}$ 溢水 $\xrightarrow{\text{不浇水}}$ 缺水 $\xrightarrow{\text{浇水}}$ 健康 收获好西瓜

强化学习问题基本设置

- 例子：瓜农种西瓜
  - 多步决策过程
  - 过程中包含状态、动作、反馈 (奖赏) 等
  - 需多次种瓜，在过程中不断摸索，才能总结出较好的种瓜策略

种下瓜苗后：(为简便，仅考虑浇水和不浇水两个动作，不考虑施肥、除草等)

缺水 $\xrightarrow{\text{浇水}}$ 健康 $\xrightarrow{\text{浇水}}$ 溢水 $\xrightarrow{\text{不浇水}}$ 缺水 $\xrightarrow{\text{浇水}}$ 健康 收获好西瓜

抽象该过程：强化学习 (reinforcement learning)

什么是强化学习

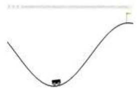
- Reinforcement（来自韦氏词典）
  - something that strengthens or encourages something, such as a response to someone’ s behaviour that is intended to make that person more likely to behave that way again
- Reinforcement Learning（RL）：强化学习、增强学习
  - 让计算机实现从一开始什么都不懂, 通过不断地尝试, 从错误中学习, 最后找到规律, 学会达到目的的方法.
- 里程碑
  - 1998年Richard S.Sutton，Reforcement Learning：An introduction
  - 2013年Deepmind提出DQN（Deep Q network）

什么是强化学习

- 强化学习可以做什么？
  - 非线性控制、下棋、机器人



  - 游戏、人机对话、无人驾驶、机器翻译

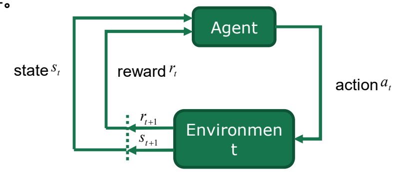


  - 智能决策问题，更确切地说是序列决策问题。

什么是强化学习

在强化学习问题中，决策的主体称为智能体（agent）。

- 强化学习解决序列决策问题：当前采取什么动作（action），可使整个任务序列最优。
- 如何使整个任务序列达到最优呢?
  - 智能体不断地和环境交互，不断尝试（因为智能体一开始不知道在当前状态下哪个动作有利于实现目标）。
  - 根据当前的回报评估（return，多步奖励即多个reward的累积）选择最大回报的那个动作。



强化学习问题基本设置

- 马尔可夫性质
  - 系统的下一个状态q_{t+1}仅与当前状态q_t有关，而与以前的状态无关

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots) = P(q_t = S_j \mid q_{t-1} = S_i) \quad \cdots$$

  - 独立于时间t的随机过程，即所谓的不动性假设，状态与时间无关，那么:

$$P(q_t = S_j \mid q_{t-1} = S_i) = a_{ij}, \quad 1 \le i, j \le N \quad \cdots$$

- 马尔可夫过程
  - 数学中用来描述随机变量序列的过程叫做**随机过程**。若随机过程所有的状态都具有马尔可夫性质，则称此随机过程为**马尔可夫随机过程**，亦称**马尔可夫过程**。

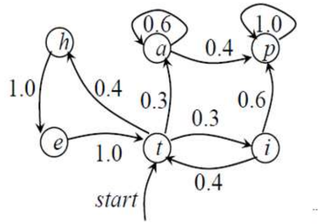
强化学习问题基本设置

马尔可夫过程是一个二元组(S, P)，且满足：S 是有限状态集合， P 是状态转移概率。状态转移概率矩阵为：

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}$$

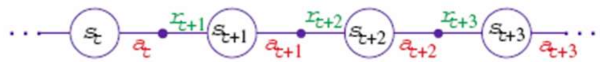
$$p_{ij} \ge 0$$

$$\sum_{j=1}^{n} p_{ij} = 1$$



马尔可夫决策过程

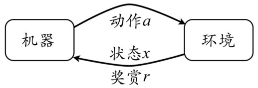
- 马尔可夫决策过程
  - 将动作（策略）和回报考虑在内的马尔科夫过程称为马尔科夫决策过程。



马尔可夫决策过程对应的马尔可夫链

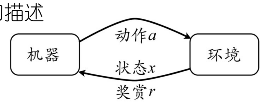
强化学习问题基本设置

- 强化学习常用马尔可夫决策过程 (Markov Decision Process, MDP) 描述
  - 机器所处的环境 E
    - 例如在种西瓜任务中，环境是西瓜生长的自然世界



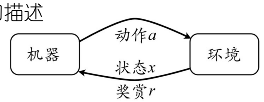
强化学习问题基本设置

- 强化学习常用马尔可夫决策过程 (Markov Decision Process, MDP) 描述
  - 机器所处的环境 E
    - 例如在种西瓜任务中，环境是西瓜生长的自然世界
  - 状态空间 X : $x \in X$ 是机器感知到的环境的描述
    - 瓜苗长势的描述



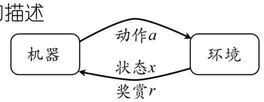
强化学习问题基本设置

- 强化学习常用马尔可夫决策过程 (Markov Decision Process, MDP) 描述
  - 机器所处的环境 E
    - 例如在种西瓜任务中，环境是西瓜生长的自然世界
  - 状态空间 X : $x \in X$ 是机器感知到的环境的描述
    - 瓜苗长势的描述
  - 机器能采取的行为空间 A
    - 浇水，施肥等



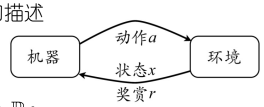
强化学习问题基本设置

- 强化学习常用马尔可夫决策过程 (Markov Decision Process, MDP) 描述
  - 机器所处的环境 E
    - 例如在种西瓜任务中，环境是西瓜生长的自然世界
  - 状态空间 X : $x \in X$ 是机器感知到的环境的描述
    - 瓜苗长势的描述
  - 机器能采取的行为空间 A
    - 浇水，施肥等
  - 策略(policy) $\pi: X \to A$ (或 $\pi: X \times A \to \mathbb{R}$)
    - 根据瓜苗状态是缺水时，返回动作浇水



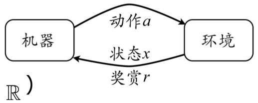
强化学习问题基本设置

- 强化学习常用马尔可夫决策过程 (Markov Decision Process, MDP) 描述
  - 机器所处的环境 E
    - 例如在种西瓜任务中, 环境是西瓜生长的自然世界
  - 状态空间 $X: x \in X$ 是机器感知到的环境的描述
    - 瓜苗长势的描述
  - 机器能采取的行为空间 A
    - 浇水，施肥等
  - 策略(policy) $\pi: X \to A$ (或 $\pi: X \times A \to \mathbb{R}$)
    - 根据瓜苗状态是缺水时，返回动作浇水
  - 潜在的状态转移(概率)函数 $P: X \times A \times X \to \mathbb{R}$
    - 瓜苗当前状态缺水，选择动作浇水，有一定概率恢复健康，也有一定概率无法恢复



强化学习问题基本设置

- 强化学习常用马尔可夫决策过程 (Markov Decision Process, MDP) 描述
  - 机器所处的环境 E
    - 例如在种西瓜任务中, 环境是西瓜生长的自然世界
  - 状态空间 $X: x \in X$ 是机器感知到的环境的描述
    - 瓜苗长势的描述
  - 机器能采取的行为空间 A
    - 浇水，施肥等
  - 策略(policy) $\pi: X \to A$ (或 $\pi: X \times A \to \mathbb{R}$)
    - 根据瓜苗状态是缺水时，返回动作浇水
  - 潜在的状态转移(概率)函数 $P: X \times A \times X \to \mathbb{R}$
    - 瓜苗当前状态缺水，选择动作浇水，有一定概率恢复健康，也有一定概率无法恢复
  - 潜在的奖赏(reward)函数 $R: X \times A \times X \to \mathbb{R}$ (或 $R: X \times X \to \mathbb{R}$)
    - 瓜苗健康对应奖赏+1，瓜苗凋零对应奖赏-10



强化学习问题基本设置

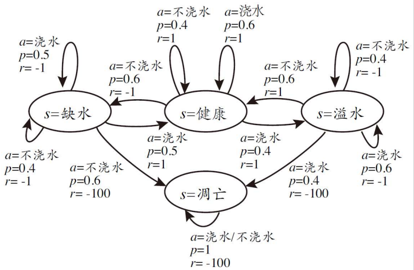
- 强化学习对应了四元组

$$E = < X, A, P, R >$$

•X：状态空间

•A：动作空间

•R(x,a) = E[R_{t+1}|x, a] :表示agent采取某个动作后的即时奖励，它还有R(x, a,x'), R(x)等表现形式，采用不同的形式，意义略有不同



•Policy π(x) - > a:根据当前state来产生action,可表现为a= π(x)或π(a|x) = P[a|x],后者表示某种状态下执行某个动作的概率

$$a = \pi(x)$$

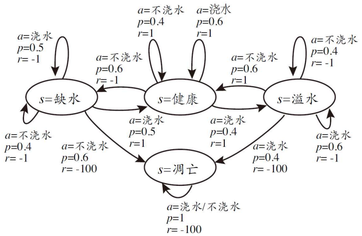
$$< x_0, a_0, r_1, x_1, a_1, r_2, \ldots, x_{T-1}, a_{T-1}, r_T, x_T >$$

强化学习问题基本设置

- 强化学习对应了四元组

$$E = < X, A, P, R >$$

- 强化学习的目标
  - 机器通过在环境中不断尝试从而学到一个策略 π，使得长期执行该策略后得到的累积奖赏最大



T 步累积奖赏: $\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T} r_t]$

γ 折扣累积奖赏: $\mathbb{E}[\sum_{t=0}^{+\infty} \gamma^t r_{t+1}]$

当下的 回报比未来反馈的回报 更重要

$$a = \pi(x)$$

$$< x_0, a_0, r_1, x_1, a_1, r_2, \ldots, x_{T-1}, a_{T-1}, r_T, x_T >$$

最大化 $\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T} r_t]$

记为U

强化学习问题基本设置

- 基于回报(return)， 我们再引入两个函数
  - 状态价值函数: $v(x)=E[U_t|X_t=x]$, 义为基于t时刻的状态x能获得的未来回报(return)的期望,加入动作选择策略后可表示为$v_\pi(s)=E_\pi[U_t|X_t=x](U_t=r_{t+1}+\gamma r_{t+2}+\ldots+\gamma^{T-t-1}r_T)$
  - 动作价值函数: $q_\pi=E_\pi[U_t|X_t=x, A_t=a]$,义为基于t时刻的状态x,选择- 个action后能获得的未来回报(return) 的期望
  - 价值函数用来衡量某-状态或动作状态的优劣，即对智能体来说是否值得选择某-状态或在某一状态下执行某一动作。

强化学习问题基本设置

- 我们需要找到最优的策略使未来回报最大化，求解过程大致可分为两步,具体内容会在后面展开
  - **预测**：给定策略，评估相应的状态价值函数和状态-动作价值函数
  - **行动**：根据价值函数得到当前状态对应的最优动作

强化学习问题基本设置

- 对于任意MDP:
  - 总是存在一个最优策略π* ,它比其它任何策略都要好，或者至少-样好
  - 所有最优决策都达到最优值函数，$V_{\pi^*}(s)=V_*(s)$
  - 所有最优决策都达到最优行动值函数， $q_{\pi^*}(s,a)=q_*(s,a)$
- 最优策略可从最优状态价值函数或者最优动作价值函数得出:

$$\pi_*(a|s)=\begin{cases}1, & \text{if } a=\arg\max_{a\in A} q_*(s,a)\\ 0, & otherwise\end{cases}$$

强化学习问题基本设置

- **强化学习 vs. 监督学习**
  - 监督学习：给有标记样本

$$\boxed{\text{示例}} \xrightarrow{\text{分类器/回归器 } f} \boxed{\text{标记}}$$

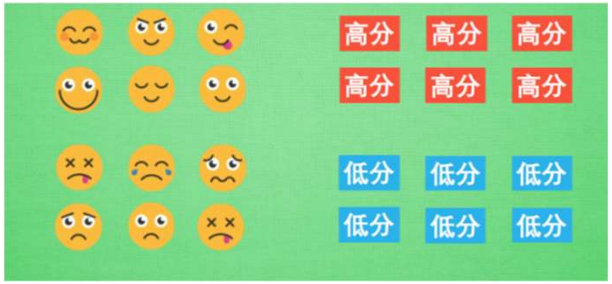
  - 强化学习：没有有标记样本，通过执行动作之后反馈的奖赏来学习

$$\boxed{\text{状态}} \xrightarrow{\text{策略 } \pi} \boxed{\text{动作}}$$

强化学习在某种意义上可以认为是具有“延迟标记信息”的监督学习

强化学习问题基本设置

监督学习 vs 强化学习

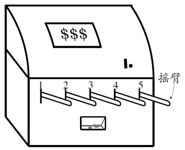


纲要

- 强化学习问题基本设置
- K-摇臂赌博机
- 有模型学习
- 免模型学习
- 值函数近似
- 模仿学习

K-摇臂赌博机

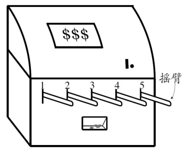
- K-摇臂赌博机 (K-Armed Bandit)
  - 只有一个状态，K个动作
  - 每个摇臂的奖赏服从某个期望未知的分布
  - 执行有限次动作
  - 最大化累积奖赏



- 强化学习面临的主要困难：探索-利用窘境 (Exploration-Exploitation dilemma)
  - 探索：估计不同摇臂的优劣 (奖赏期望的大小)
  - 利用：选择当前最优的摇臂

K-摇臂赌博机

- K-摇臂赌博机 (K-Armed Bandit)
  - 只有一个状态，K个动作
  - 每个摇臂的奖赏服从某个期望未知的分布
  - 执行有限次动作
  - 最大化累积奖赏



- 强化学习面临的主要困难：探索-利用窘境 (Exploration-Exploitation dilemma)
  - 探索：估计不同摇臂的优劣 (奖赏期望的大小)
  - 利用：选择当前最优的摇臂

- 在探索与利用之间进行折中
  - ε-贪心
  - Softmax

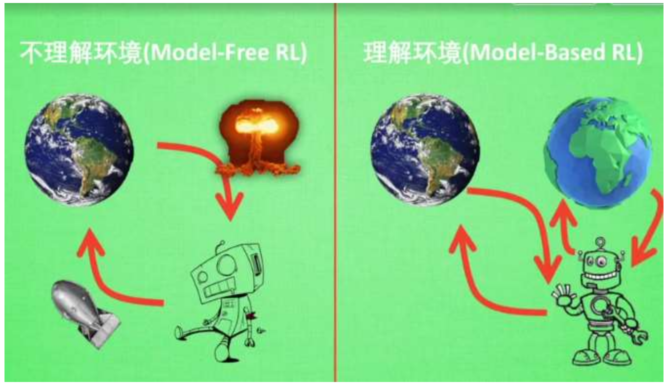
K-摇臂赌博机

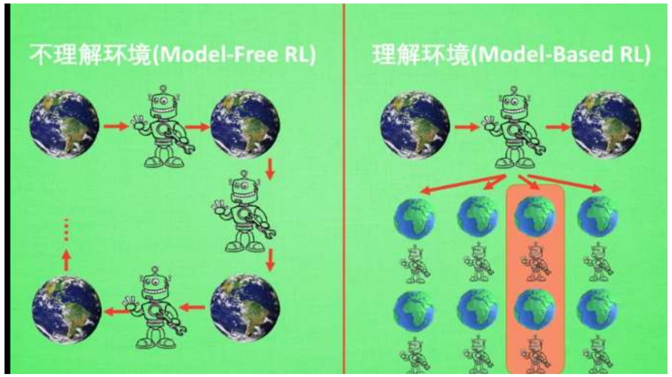
- ε-贪心
  - 以 ε 的概率探索：均匀随机选择一个摇臂
  - 以 1-ε 的概率利用：选择当前平均奖赏最高的摇臂
- Softmax：基于当前已知的摇臂平均奖赏来对探索与利用折中
  - 若某个摇臂当前的平均奖赏越大，则它被选择的概率越高
  - 概率分配使用Boltzmann分布：

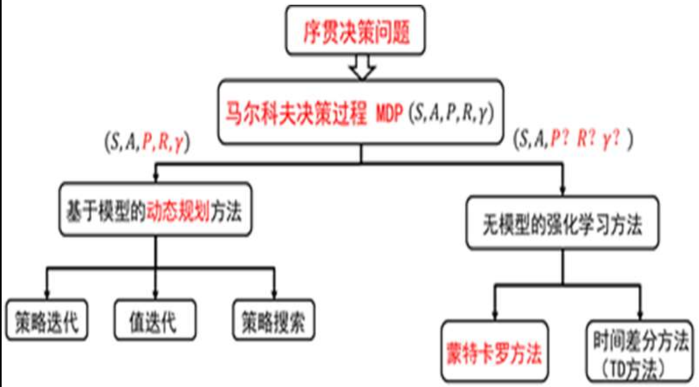
$$P(k)=\frac{e^{\frac{Q(k)}{\tau}}}{\sum_{i=1}^{K}e^{\frac{Q(i)}{\tau}}}$$

其中，$Q(i)$ 记录当前摇臂的平均奖赏

- 两种算法都有一个折中参数(ε，τ)，算法性能孰好孰坏取决于具体应用问题







纲要

- 强化学习问题基本设置
- K-摇臂赌博机
- 有模型学习
- 免模型学习
- 值函数近似
- 模仿学习

有模型学习

- 有模型学习 (model-based learning)：$E=<X,A,P,R>$
  - X,A,P,R 均已知
  - 方便起见，假设状态空间和动作空间均有限

有模型学习

- 有模型学习 (model-based learning)：$E=<X,A,P,R>$
  - X,A,P,R 均已知
  - 方便起见，假设状态空间和动作空间均有限
- 强化学习的目标：找到使累积奖赏最大的策略 π

有模型学习

- 有模型学习 (model-based learning)：$E=<X,A,P,R>$
  - X,A,P,R 均已知
  - 方便起见，假设状态空间和动作空间均有限
- 强化学习的目标：找到使累积奖赏最大的策略 π
- 策略评估：使用某策略所带来的累积奖赏

状态值函数：从状态 x 出发，使用策略 π 所带来的累积奖赏

$$\begin{cases} V_T^\pi(x)=\mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^{T} r_t | x_0=x\right], & T \text{ 步累积奖赏;} \\ V_\gamma^\pi(x)=\mathbb{E}_\pi\left[\sum_{t=0}^{+\infty}\gamma^t r_{t+1} | x_0=x\right], & \gamma \text{ 折扣累积奖赏.} \end{cases}$$

状态-动作值函数：从状态 x 出发，执行动作 a 后再使用策略 π 所带来的累积奖赏

$$\begin{cases} Q_T^\pi(x,a)=\mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^{T} r_t | x_0=x, a_0=a\right]; \\ Q_\gamma^\pi(x,a)=\mathbb{E}_\pi\left[\sum_{t=0}^{+\infty}\gamma^t r_{t+1} | x_0=x, a_0=a\right]. \end{cases}$$

有模型学习

- 给定π，值函数的计算：值函数具有简单的递归形式
  - T 步累积奖赏：

$$V_T^\pi(x)=\mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^{T} r_t|x_0=x\right]$$

$$=\mathbb{E}_\pi\left[\frac{1}{T}r_1+\frac{T-1}{T}\frac{1}{T-1}\sum_{t=2}^{T} r_t|x_0=x\right]$$ （全概率公式）

$$=\sum_{a\in A}\pi(x,a)\sum_{x'\in X}P_{x\to x'}^a\left(\frac{1}{T}R_{x\to x'}^a+\frac{T-1}{T}\mathbb{E}_\pi\left[\frac{1}{T-1}\sum_{t=1}^{T-1} r_t|x_0=x'\right]\right)$$

$$=\sum_{a\in A}\pi(x,a)\sum_{x'\in X}P_{x\to x'}^a\left(\frac{1}{T}R_{x\to x'}^a+\frac{T-1}{T}V_{T-1}^\pi(x')\right).$$ Bellman等式

有模型学习

- 给定π，值函数的计算：值函数具有简单的递归形式
  - T 步累积奖赏：

$$V_T^\pi(x)=\mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^{T} r_t|x_0=x\right]$$

$$=\mathbb{E}_\pi\left[\frac{1}{T}r_1+\frac{T-1}{T}\frac{1}{T-1}\sum_{t=2}^{T} r_t|x_0=x\right]$$ （全概率公式）

$$=\sum_{a\in A}\pi(x,a)\sum_{x'\in X}P_{x\to x'}^a\left(\frac{1}{T}R_{x\to x'}^a+\frac{T-1}{T}\mathbb{E}_\pi\left[\frac{1}{T-1}\sum_{t=1}^{T-1} r_t|x_0=x'\right]\right)$$

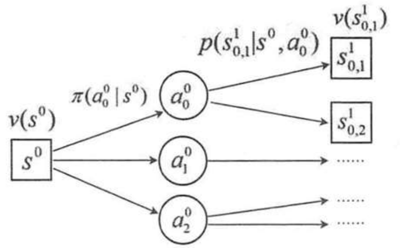
$$=\sum_{a\in A}\pi(x,a)\sum_{x'\in X}P_{x\to x'}^a\left(\frac{1}{T}R_{x\to x'}^a+\frac{T-1}{T}V_{T-1}^\pi(x')\right).$$ Bellman等式

  - γ折扣累积奖赏：

$$V_\gamma^\pi(x)=\sum_{a\in A}\pi(x,a)\sum_{x'\in X}P_{x\to x'}^a\left(R_{x\to x'}^a+\gamma V_\gamma^\pi(x')\right).$$

有模型学习

- 价值函数



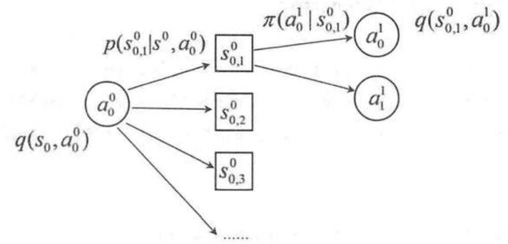
我们计算左边的s^0状态的$v(s^0)$，我们可以通过它后面的$r_{a_i}^{s^1_{0,i}}+s^1_{0,i}$加权的和，其中$r_{a_i}^{s^1_{0,i}}$是采取行动$a_i$后获得的奖励。

$$v_\pi(s_t)=\sum_{a_t}\pi(a_t|s_t)\sum_{s_{t+1}}p(s_{t+1}|s_t,a_t)[r_{a_t}^{s_{t+1}}+\gamma * v_\pi(s_{t+1})]$$ Bellman等式

有模型学习

- 给定π，状态-动作值函数的计算：通过值函数来表示

$$\begin{cases} Q_T^\pi(x,a)=\sum_{x'\in X}P_{x\to x'}^a\left(\frac{1}{T}R_{x\to x'}^a+\frac{T-1}{T}V_{T-1}^\pi(x')\right); \\ Q_\gamma^\pi(x,a)=\sum_{x'\in X}P_{x\to x'}^a\left(R_{x\to x'}^a+\gamma V_\gamma^\pi(x')\right). \end{cases}$$ Bellman等式



有模型学习

- 给定π，状态-动作值函数的计算：通过值函数来表示

$$\begin{cases} Q_T^\pi(x,a) = \sum_{x'\in X} P_{x\to x'}^a \left(\frac{1}{T}R_{x\to x'}^a + \frac{T-1}{T}V_{T-1}^\pi(x')\right); \\ Q_\gamma^\pi(x,a) = \sum_{x'\in X} P_{x\to x'}^a \left(R_{x\to x'}^a + \gamma V_\gamma^\pi(x')\right). \end{cases}$$

- 最优策略，最优值函数，最优状态-动作值函数
  - 最优策略：最大化累积奖赏

$$\pi^* = \underset{\pi}{\arg\max} \sum_{x\in X} V^\pi(x).$$

  - 最优值函数：

$$\forall x \in X : V^*(x) = V^{\pi^*}(x).$$

  - 最优状态-动作值函数

MDP 的问题主要分两类
- •Prediction 问题
  - •输入：MDP 和策略（policy）
  - •输出：状态价值函数
- •Control 问题
  - •输入：MDP
  - •输出：最优态价值函数和最优策略

解决也是分两种：
策略迭代
价值迭代

策略迭代

- 可以采用策略迭代法的思路
  - 1.以某种策略T开始，计算当前策略下的值函数v_π (s)。
  - 2..利用这个值函数，更新策略，得到π *。
  - 3.再用这个策略π *继续前行，更新值函数，得到v_π’ (s)，一直到v_π (s)不再发生变化。
- 计算当前的状态值函数的过程，称为--策略评估(policy evaluation) $v_\pi^T(s_t) = \sum_{a_t} \pi^{T-1}(a_t|s_t) \sum_{s_{t+1}} p(s_{t+1}|s_t,a_t)[r_{a_t}^{s_{t+1}} + \gamma * v_\pi^{T-1}(s_{t+1})]$
- 计算最优策略的过程，称为--策略提升(policy improvement)

$$q_\pi^T(s_t,a_t) = \sum_{s} p(s_{t+1}|s_t,a_t)[r_{a_t}^{s_{t+1}} + \gamma * v_\pi^T(s_{t+1})]$$

$$\pi^{T+1}(s) = argmax_a q_\pi^T(s,a)$$

1 策略评估步骤

1. 输入：策略π^{T-1}，状态转移概率$p(s_{t+1}|s_t,a_t)$，奖励r，衰减因子γ，值函数$v^{T-1}(s)$
2. $v_0(s) = v^{T-1}(s)$
3. $Repeat \quad k = 0, 1 \ldots$
4. $\quad for \;\; every \;\; s \;\; do$
5. $\quad\quad v_{k+1}(s_t) = \sum_{a_t} \pi^{T-1}(a_t|s_t) \sum_{s_{t+1}} p(s_{t+1}|s_t,a_t)[r_{a_t}^{s_{t+1}} + \gamma * v_k(s_{t+1})]$
6. $Until \;\; v_{k+1}(s) = v_k(s)$
7. 输出$v^T(s) = v_{k+1}(s)$

2 策略迭代步骤

1. 输入：策略π_0，状态转移概率$p(s_{t+1}|s_t,a_t)$，奖励r，衰减因子γ，值函数$v_0(s)$
2. $Repeat \quad k=0,1...$
3. $\quad policy \quad evaluation$
4. $\quad q_k(s_t,a_t)=\sum_{s_{t+1}} p(s_{t+1}|s_t,a_t)[r_{a_t}^{s_{t+1}}+\gamma * v_k(s_{t+1})]$
5. $\quad \pi_{k+1}(s)=argmax_a q_k(s,a)$
6. $Until \quad \pi_{k+1}(s)=\pi_k(s)$
7. 输出$\pi^*(s)=\pi_{k+1}(s)$

价值迭代

- 价值迭代

$$v(s_t)=\sum \pi(a|s_t)\sum p(s_{t+1}|s_t,a)[R_t+\gamma v(s_{t+1})]$$

$$q(s_t,a_t)=\sum_{s_{t+1}} p(s_{t+1}|s_t,a_t)[R_t+\gamma v(s_{t+1})]$$

$$\pi(s)=argmax_a q(s,a)$$

$$\pi(s)=argmax_a \sum_{s_{t+1}} p(s_{t+1}|s_t,a_t)[R_t+\gamma v(s_{t+1})]$$

价值迭代

- 由于最优策略的存在，实际上策略最终的选择是单一的，也就是说对于每一个状态，最优策略会采取某一种行动，这种行动不会比其他行动差。所以我们可以认为：

$$v(s)=max_a q(s,a)$$

即：

$$v(s)=max_a \sum_{s'} p(s'|s,a)[r(s,a,s')+\gamma \tilde{v}(s')]$$

采用动态规划求解

有模型学习

- 有模型学习小结
  - 强化学习任务可归结为基于动态规划的寻优问题
  - 与监督学习不同，这里并未涉及到泛化能力，而是为每一个状态找到最好的动作
- 问题：如果模型未知呢？

纲要

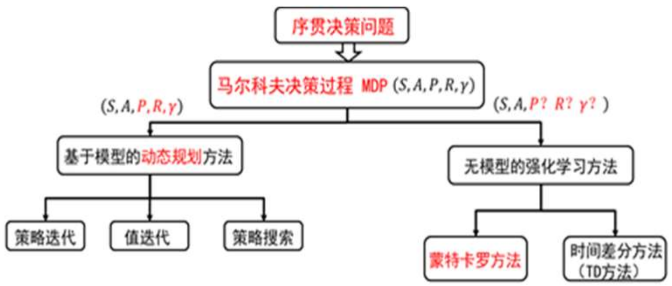
- 强化学习问题基本设置
- K-摇臂赌博机
- 有模型学习
- 免模型学习
- 值函数近似
- 模仿学习

免模型学习

- 免模型学习 (model-free learning)：更加符合实际情况
  - 转移概率，奖赏函数未知
  - 甚至环境中的状态数目也未知
  - 假定状态空间有限
- 免模型学习所面临的困难
  - 策略无法评估
  - 无法通过值函数计算状态-动作值函数
  - 机器只能从一个起始状态开始探索环境

免模型学习

- 免模型学习 (model-free learning)：更加符合实际情况
  - 转移概率，奖赏函数未知
  - 甚至环境中的状态数目也未知
  - 假定状态空间有限
- 免模型学习所面临的困难
  - 策略无法评估
  - 无法通过值函数计算状态-动作值函数
  - 机器只能从一个起始状态开始探索环境
- 解决困难的办法
  - 多次采样
  - 直接估计每一对状态-动作的值函数
  - 在探索过程中逐渐发现各个状态



免模型学习

- 蒙特卡罗强化学习：采样轨迹，用样本均值近似期望
  - 策略评估：蒙特卡罗法
    - 从某状态出发，执行某策略
    - 对轨迹中出现的每对状态-动作，记录其后的奖赏之和
    - 采样多条轨迹，每个状态-动作对的累积奖赏取平均

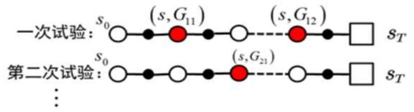
一条轨迹：

$$< x_0, a_0, r_1, x_1, a_1, r_2, \ldots, x_{T-1}, a_{T-1}, r_T, x_T >$$

- 我们可以利用策略产生很多次试验，每次试验都是从任意的初始状态开始直到终止状态，比如—-次试验(an episode)为：S_1，A_1，R_2,...,S_T计算- -次试验中状态s处的折扣回报返回值为：

$$G_t(s) = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$$

- 多次试验



- 第一次访问蒙特卡罗方法的计算公式为：

$$v(s) = \frac{G_{11}(s) + G_{21}(s) + \cdots}{N(s)}$$

每次访问蒙特卡罗方法是指，在计算状态 s 处的值函数时，利用所有访问到状态 s 时的回报返回值，即：$v(s) = \dfrac{G_{11}(s) + G_{12}(s) + \cdots + G_{21}(s) + \cdots}{N(s)}$

根据大数定律：$v(s) \to v_\pi(s)$ as $N(s) \to \infty$

免模型学习

- 蒙特卡罗强化学习：采样轨迹，用样本均值近似期望
  - 策略评估：蒙特卡罗法
    - 从某状态出发，执行某策略
    - 对轨迹中出现的每对状态-动作，记录其后的奖赏之和
    - 采样多条轨迹，每个状态-动作对的累积奖赏取平均
  - 策略改进（提升）：换入当前最优动作

一条轨迹：

$$< x_0, a_0, r_1, x_1, a_1, r_2, \ldots, x_{T-1}, a_{T-1}, r_T, x_T >$$

如何获得充足的经验是无模型强化学习的核心所在

- 探索性初始化蒙特卡罗方法

[1] 初始化所有：
$s \in S, a \in A(s), Q(s,a) \leftarrow arbitrary,$
$\pi(s) \leftarrow arbitrary, Returns(s,a) \leftarrow empty\ list$

[2] Repeat:
随机选择 $S_0 \in S, A_0 \in A(S_0)$ ，从 S_0, A_0 开始以策略 V
生成一个实验(episode)，对每对在这个实验中出现的状态和动作，s, a:

策略评估

[3] $G \leftarrow s,a$第一次出现后的回报
将G附加于回报Returns(s, a)上
$Q(s,a) \leftarrow average(Returns(s,a))$对回报取均值

[4] 对该实验中的每一个 s：
$\pi(s) \leftarrow \arg\max_a Q(s,a)$ 策略改进

免模型学习

- 蒙特卡罗强化学习：采样轨迹，用样本均值近似期望
  - 策略评估：蒙特卡罗法
    - 从某状态出发，执行某策略(每次试验)
    - 对轨迹中出现的每对状态-动作，记录其后的奖赏之和
    - 采样多条轨迹，每个状态-动作对的累积奖赏取平均
  - 策略改进：换入当前最优动作

一条轨迹：

$$< x_0, a_0, r_1, x_1, a_1, r_2, \ldots, x_{T-1}, a_{T-1}, r_T, x_T >$$

- 蒙特卡罗强化学习可能遇到的问题：轨迹的单一性

在探索性初始化中，迭代每一幕时，初始状态是随机分配的，这样可以保证迭代过程中每个状态行为对都能被选中。它蕴含着一个假设，即：假设所有的动作都被无限频繁选中。对于这个假设，有时很难成立，或无法完全保证。

- 解决问题的办法
  - ε-贪心法
    - 同策略：被评估与被改进的是同一个策略（on-policy）
    - 异策略：被评估与被改进的是同一个策略 (用重要性采样技术，off policy)

$$\pi^{\epsilon}(x) = \begin{cases} \pi(x), & \text{以概率 } 1-\epsilon; \\ A \text{ 中以均匀概率选取的动作}, & \text{以概率 } \epsilon. \end{cases}$$

$$\pi(a|s) \leftarrow \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|A(s)|} \; if \; a = arg\max_a Q(s,a) \\ \frac{\varepsilon}{|A(s)|} \; if \; a \neq arg\max_a Q(s,a) \end{cases}$$

免模型学习

- 蒙特卡罗强化学习：采样轨迹，用样本均值近似期望
  - 策略评估：蒙特卡罗法
    - 从某状态出发，执行某策略
    - 对轨迹中出现的每对状态-动作，记录其后的奖赏之和
    - 采样多条轨迹，每个状态-动作对的累积奖赏取平均
  - 策略改进：换入当前最优动作

一条轨迹：

$$< x_0, a_0, r_1, x_1, a_1, r_2, \ldots, x_{T-1}, a_{T-1}, r_T, x_T >$$

- 蒙特卡罗强化学习可能遇到的问题：轨迹的单一性

ϵ

免模型学习

- 同策略蒙特卡罗强化学习算法（on-policy）

[1] 初始化所有：$s \in S, a \in A(s), Q(s,a) \leftarrow arbitrary$

$Returns(s,a) \leftarrow empty\, list$

$\pi(s) \leftarrow arbitrary\ \varepsilon\text{-}soft$策略，

Repeat:

[2] 从 S_0, A_0 开始以策略 π 生成一次实验(episode)，

[3] 对每对在这个实验中出现的状态和动作，s, a：

$G \leftarrow s,a$第一次出现后的回报

将G附加于回报Returns(s, a)上

$Q(s,a) \leftarrow average(Returns(s,a))$对回报取均值

策略评估

[4] 对该实验中的每一个s：

策略改进

$$\pi(a|s) \leftarrow \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|A(s)|} & if\ a = \arg\max_a Q(s,a) \\ \frac{\varepsilon}{|A(s)|} & if\ a \neq \arg\max_a Q(s,a) \end{cases}$$