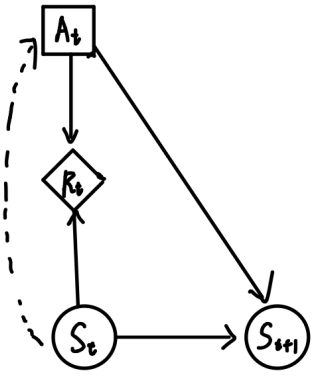
课后练习4-6章

人工智能学院 181220076 周韧哲

4.1

- 马尔科夫假设是指状态S_{t+1}和奖赏R_t仅依赖于时刻t的状态和行动，与之前的状态和行动无关。
- 马尔科夫决策过程由状态空间、行动空间、奖赏空间和动力函数构成。
- 稳态MDP是指动力函数不随时间发生变化的MDP，从而状态转移函数和奖赏函数也不随时间变化。
- 稳态MDP的决策网络表示：



4.2

（以下用π_l,π_r分别表示π_{left},π_{right}）

- 当$\gamma=0$时，策略是短视的，仅考虑当前奖赏，有$U^{\pi_l}=1,U^{\pi_r}=0$，故π_l为最优策略
- 当$\gamma=0.5$时
  - 在π_l下，$G_t=\sum_{k=0}^{\infty}0.5^k\times1$，$U^{\pi_l}=E_{\pi_l}[G_t]=\lim_{k\to\infty}2-\frac{1}{2^k}=2$。
  - 在π_r下，$G_t=\sum_{k=0}^{\infty}0.5^k\times2$，$U^{\pi_r}=E_{\pi_l}[G_t]=\lim_{k\to\infty}4-\frac{1}{2^{k-1}}=4$。

  所以π_r为最优策略
- 当$\gamma=0.9$时
  - 在π_l下，$G_t=\sum_{k=0}^{\infty}0.9^k\times1$，$U^{\pi_l}=E_{\pi_l}[G_t]=\lim_{k\to\infty}10\times(1-0.9^{k+1})=10$。
  - 在π_r下，$G_t=\sum_{k=0}^{\infty}0.9^k\times2$，$U^{\pi_r}=E_{\pi_l}[G_t]=\lim_{k\to\infty}20\times(1-0.9^{k+1})=20$。

  所以π_r为最优策略

4.3

- 由

$$U^{\pi}(s)=\sum_{a}\pi(a|s)\sum_{s',r}p(s',r|s,a)[r+\gamma U^{\pi}(s')]$$

容易写出每个状态的状态值函数：

$$\begin{aligned}U^{\pi}(h) &= \pi(s|h)[\alpha(r_s+\gamma U^{\pi}(h))+(1-\alpha)(r_s+\gamma U^{\pi}(l))]+\\ &\quad \pi(w|h)[1\times(r_w+\gamma U^{\pi}(h))]\\ &= \pi(s|h)[\alpha\gamma U^{\pi}(h)+(1-\alpha)\gamma U^{\pi}(l)+r_s]+\\ &\quad \pi(w|h)[r_w+\gamma U^{\pi}(h)]\end{aligned}$$

$$\begin{aligned}U^{\pi}(l) &= \pi(s|l)[\beta(r_s+\gamma U^{\pi}(l))+(1-\beta)(-3+\gamma U^{\pi}(h))]+\\ &\quad \pi(w|l)[1\times(r_w+\gamma U^{\pi}(l))]+\\ &\quad \pi(re|l)[0+\gamma U^{\pi}(h)]\\ &= \pi(s|l)[\beta\gamma U^{\pi}(l)+(1-\beta)\gamma U^{\pi}(h)+\beta r_s-3(1-\beta)]+\\ &\quad \pi(w|l)[r_w+\gamma U^{\pi}(l)]+\\ &\quad \pi(re|l)[\gamma U^{\pi}(h)]\end{aligned}$$

则其最优状态值函数的Bellman方程为：

$$\begin{aligned}U^{*}(h) = \max\{&\alpha\gamma U^{*}(h)+(1-\alpha)\gamma U^{*}(l)+r_s,\\ &r_w+\gamma U^{*}(h)\}\end{aligned}$$

$$\begin{aligned}U^{*}(l) = \max\{&\beta\gamma U^{*}(l)+(1-\beta)\gamma U^{*}(h)+\beta r_s-3(1-\beta),\\ &r_w+\gamma U^{*}(l),\\ &\gamma U^{*}(h)\}\end{aligned}$$

4.4

- $r=100$的最优策略为:

u	l	.
u	l	d
u	l	l

- $r=-3$的最优策略为：

r	r	.
r	r	u
r	r	u

- $r=0$的最优策略为：

r	r	.
u	u	u
u	u	u

- $r=3$的最优策略为：

u	l	.
u	l	d
u	l	l

4.5

- 证明：

$$\begin{aligned}
\|U^*(s)-U_k(s)\|_\infty &= \max_s |U^*(s)-U_k(s)| \\
&= \gamma \max_s |\max_a \sum_{s'} T(s'|s,a)U^*(s') - \max_a \sum_{s'} T(s'|s,a)U_{k-1}(s')| \\
&\le \gamma \max_s \max_a |\sum_{s'} T(s'|s,a)|U^*(s') - \sum_{s'} T(s'|s,a)U_{k-1}(s')| \\
&= \gamma \max_s |\sum_{s'} T(s'|s,a^*)(U^*(s') - U_{k-1}(s'))| \\
&\le \gamma \sum_{s'} T(s'|s,a^*) \max_{s'} |U^*(s') - U_{k-1}(s')| \\
&= \gamma \|U^*(s) - U_{k-1}(s)\|_\infty
\end{aligned}$$

同时，有：

$$\begin{aligned}
\|U^*(s)-U_{k-1}(s)\|_\infty - \|U^*(s)-U_k(s)\|_\infty &= \max_{s'} |U^*(s') - U_{k-1}(s')| - \max_{s'} |U^*(s') - U_k(s')| \\
&\le \max_{s'} |U_k(s') - U_{k-1}(s')| \\
&= \|U_k(s') - U_{k-1}(s')\|_\infty
\end{aligned}$$

综合可得

$$\begin{aligned}
\|U^*(s)-U_{k-1}(s)\|_\infty - \|U^*(s)-U_k(s)\|_\infty &= (\frac{1}{\gamma}-1)\|U^*(s)-U_k(s)\|_\infty \\
&\le \|U_k(s)-U_{k-1}(s)\|_\infty \\
&< \delta \\
&= \epsilon(\frac{1}{\gamma}-1)
\end{aligned}$$

同时除以$\frac{1}{\gamma}-1$，即得证：

$$\|U^*(s)-U_k(s)\|_\infty < \epsilon$$

4.6

- 动态规划适用于离散动作空间和离散状态空间都不太大的情形，能够快速获得最优解；而在大规模或连续空间MDP问题中，动态规划难以进行，近似动态规划能够帮助我们找到一个近似的最优策略；在线方法把计算限制在从当前状态可达的状态上，比整个状态空间小很多，可以显著减少近似最优行动选择所需的存储空间和计算时间，避免了维数灾难，在一些状态空间和动作空间巨大的情绪下能有效获得近似最优解。

5.1

- 在RL问题中，探索帮助Agent充分了解它的状态空间，利用则帮助Agent找到最优的动作序列，通常状态空间和动作空间是巨大的，为了在可接受的时间复杂度内获得近似最优解，必须要平衡探索与利用，因此探索与利用是RL的重要的基本问题和基本概念。
- n摇臂赌博机问题：有n个摇臂，赌徒在投入一个硬币后可选择拉下其中一个摇臂；每个摇臂以一定的概率吐出硬币，但赌徒并不知道这个概率；总共能拉h次摇臂。目标是通过一定的策略最大化自己的奖赏，即获得最多的硬币。
- 探索：我们只知道其中一个摇臂会以0.9的概率输出$1，而另一个摇臂会输出什么并不知道，探索就是去探索另一个摇臂，看看另一个摇臂会不会比当前摇臂好。

利用：就是利用我们已知的这个摇臂，可以在期望上较为稳定地获得非0的奖赏。

5.2

- ρ_i为点估计，θ_i为区间估计。使用ϵ-贪心策略时，有0.5的概率随机选择一个摇臂，有0.5的概率选择第1个摇臂（即行动值最大的摇臂）。使用一个95%置信区间的区间探索策略，会选择拉上置信界最大的摇臂，即第2个摇臂。

5.3

- 将增量估计方程写为：

$$\hat{x}_n = (1-\alpha)\hat{x}_{n-1} + \alpha x_n$$

下面使用数学归纳法证明$\hat{x}_n = (1-\alpha)^n\hat{x}_0 + \sum_{i=1}^{n}\alpha(1-\alpha)^{n-i}x_i$。

- 当$n=1$时，有

$$\hat{x}_1 = (1-\alpha)\hat{x}_0 + \alpha x_1$$

显然成立

- 假设$n=k-1, k>2$时有

$$\hat{x}_{k-1} = (1-\alpha)^{k-1}\hat{x}_0 + \sum_{i=1}^{k-1}\alpha(1-\alpha)^{k-1-i}x_i$$

- 则当$n=k$时，由增量估计公式得：

$$\hat{x}_k = (1-\alpha)\hat{x}_{k-1} + \alpha x_k$$

将$\hat{x}_{k-1} = (1-\alpha)^{k-1}\hat{x}_0 + \sum_{i=1}^{k-1}\alpha(1-\alpha)^{k-1-i}x_i$代入，即可得：

$$\hat{x}_k = (1-\alpha)^{k}\hat{x}_0 + \sum_{i=1}^{k}\alpha(1-\alpha)^{k-i}x_i$$

得证。

5.4

- 增量估计方程为$\hat{x}_n = \hat{x}_{n-1} + \alpha(n)(x_n - \hat{x}_{n-1})$，学习率为$\alpha(n)$。
- 学习率为0.1时，新估计为$3 + 0.1 \times (7-3) = 3.4$。
- 学习率为0.5时，新估计为$3 + 0.5 \times (7-3) = 5$。
- 可以看出，学习率控制着增量估计中每一次估计的“步长”，可以通过调整学习率来调整每次估计的变化幅度。

5.5

- 由行动值函数的定义：

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}[G_t|S_t=s, A_t=a] \\ &= E_{\pi}[\sum_{k=0}^{\infty}\gamma^k R_{t+k}|S_t=s, A_t=a] \\ &= E_{\pi}[R_t + \gamma Q^{\pi}(S_{t+1}, A_{t+1})|S_t=s, A_t=a] \end{aligned}$$

再由增量估计方程：

$$\hat{x}_n = \hat{x}_{n-1} + \alpha(n)(x_n - \hat{x}_{n-1})$$

TD目标值即为$R_t + \gamma Q(S_{t+1}, A_{t+1})$，TD误差为$R_t + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$，从而得到Sarsa更新公式：

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_t + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

由Bellman最优方程：

$$Q^*(s,a) = E_\pi[R_t + \gamma \max_{a'} Q^*(S_{t+1}, a') | S_t = s, A_t = a]$$

TD目标值即为$R_t + \gamma \max_{a'} Q^*(S_{t+1}, a')$，TD误差为$R_t + \gamma \max_{a'} Q^*(S_{t+1}, a') - Q(S_t, A_t)$，从而得到Q-learning更新公式：

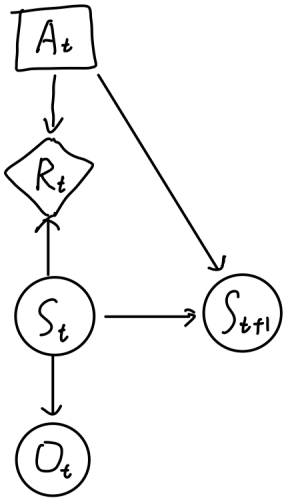
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_t + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

5.6

Algrithm Linear approximation Sarsa
  function LinearApproximationSarsa
    $t \leftarrow 0$
    $s_0 \leftarrow 0$
    Initialize θ
    loop
      Choose action a_t based on $\theta_a^T \beta(s_t)$ and some exploration strategy
      Observe new state s_{t+1} and reward r_t
      $\theta \leftarrow \theta + \alpha(r_t + \gamma \theta^T \beta(s_{t+1}, a_{t+1}) - \theta^T \beta(s_t, a_t))\beta(s_t, a_t)$
      $t \leftarrow t + 1$

6.1

- POMDP是Partially Observable MDP的缩写，与MDP的不同在于它是有观察模型的MDP，多了观察空间和观察函数，具有部分可观察性。POMDP的决策网络结构为：



与MDP决策网络的不同在于多了观察O_t。

6.2

- 对于左侧的栅格世界，可以用一个精确的值来表示状态；对于右侧的栅格世界，可以用一个分布来表示状态。这个分布就代表了在POMDPs中，我们对Agent所处的可能的状态有一个估计的概率值，当概率较大时，处在该状态的可能性就较大，也即对Agent处在该状态的信念越强，所以也叫信念状态MDPs。信念状态空间是连续的，所以求解信念状态MDPs是困难的。

6.3

- 更新POMDP的信念状态的方法有：
  - 离散状态滤波器
  - 线性高斯滤波器
  - 扩展的卡尔曼滤波器

    - 粒子滤波器：带拒绝与不带拒绝
- 使用时，若状态是离散的，则可使用离散状态滤波器；若状态转移模型和观察模型是连续的且有线性高斯形式，则可用线性高斯滤波器；若状态转移模型和观察模型有非线性高斯形式，则可用扩展的卡尔曼滤波器；若状态空间很大/连续，动力系统不能用线性高斯模型来很好近似时，则可以使用粒子滤波器。

6.4

$$
\begin{aligned}
b'(s') &= P(s'|o,a,b) \\
&\propto P(o|s',a,b)P(s'|a,b) && \text{贝叶斯规则} \\
&\propto O(o|s',a)P(s'|a,b)) && \text{观察函数定义} \\
&\propto O(o|s',a)\sum_s P(s'|a,b,s)P(s|a,b)) && \text{全概率公式} \\
&\propto O(o|s',a)\sum_s T(s'|s,a)b(s)) && \text{简化}
\end{aligned}
$$

6.5

- 这不是一个有效的策略，行动空间中的每一个动作对应一个阿尔法向量，每个行动不能有多个阿尔法向量。

6.6

- 离线求解POMDP，会在执行动作之前完成所有的或绝大部分的计算，大部分离线方法求解的是近似最优解，通常把策略表示为阿尔法向量或有限状态控制器。优点在于从全局求解最优策略而非局部空间，缺点在于计算复杂度较高，难以在高维空间中求解。而在线求解POMDP，会把计算限制在从当前状态可达的状态空间中来确定最优策略。优点是与总信念空间相比，从当前状态可到达的信念状态空间通常较小，约简了搜索空间，避免了维数灾难，有时更易于应用于高维问题。缺点是与离线方法相比，在线方法在执行过程中每个决策步骤需要更多的计算。
- QMDP是对完全可观察的近似，假设所有状态的不确定性在下一个时刻消失，用值迭代来计算阿尔法向量，优点是能给出最优值函数的上界与近似最优行动，缺点是在有信息收集行动的POMDP问题中性能不太好。FIB在一定程度上考虑部分可观察性来计算每个行动的阿尔法向量，优点是可以获得比QMDP更接近最优值函数的上界，缺点是计算复杂度比QMDP高。基于点的值迭代则是在信念状态空间中更新与有限个点关联的阿尔法向量。不同于QMDP和FIB，基于点的值迭代是获得最优值函数的下界，由两部分构成：选择信念点然后更新阿尔法向量，优点是通过更新信念状态的阿尔法向量，直至收敛，可以用这些阿尔法向量近似信念状态空间中任意点的函数值。