ch1: Markov Decision Process (static concepts)

state(S) - policy(pi) - action(A) - model(p) - state(S')
reward(R') from transition
return(G) from trajectory
value(V+Q)

策略与价值:
reward、return、value，用来评估一个策略的好坏。
策略与价值一一对应。
价值比较，策略比较，策略改进定理。
强化学习的终极目标，求取最优策略，
最优策略不唯一，最优价值唯一。
r线性变换，V+Q线性变换，改变最优价值，不改变greedy最优策略。
最优动作价值，意味着选取这个动作，未来回报的期望最大。
迭代时，最优策略可能已经稳定了，但是对应的最优价值还没稳定。
从终止状态反向迭代更新价值，速度更快。但是哪里是终止状态? 上帝视角。

$$p\left(s^{\prime}, r \mid s, a\right)=\operatorname{Pr}\left\{S_{t}=s^{\prime}, R_{t}=r \mid S_{t-1}=s, A_{t-1}=a\right\}$$

$$\sum_{s^{\prime} \in S} \sum_{r \in R} p\left(s^{\prime}, r \mid s, a\right)=1$$

$$p\left(s^{\prime} \mid s, a\right)=\sum_{r \in R} p\left(s^{\prime}, r \mid s, a\right)$$

$$r(s, a)=\sum_{s^{\prime} \in S} \sum_{r \in R}\left(p\left(s^{\prime}, r \mid s, a\right) * r\right)$$

ch2: Bellman Equations (static relations)

实质：描述状态值之间的静态关系（单项形式、矩阵形式）
求解：（矩阵求逆、数值迭代）—（policy-evaluation）

$$
\begin{aligned}
v_\pi(s) &= E_\pi\left[G_t \mid S_t = s\right] \\
&= E_\pi\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \sum_{a \in A} \pi(a \mid s) \sum_{s', r} p\left(s', r \mid s, a\right) * \left[r + \gamma E_\pi\left[G_{t+1} \mid S_{t+1} = s'\right]\right] \\
&= \sum_{a \in A} \pi(a \mid s) \sum_{s', r} p\left(s', r \mid s, a\right) * \left[r + \gamma v_\pi\left(s'\right)\right] \\
&= \sum_{a \in A} \left(\pi(a \mid s) * q_\pi(s, a)\right)
\end{aligned}
$$

$$
\begin{aligned}
q_\pi(s, a) &= E_\pi\left[G_t \mid S_t = s, A_t = a\right] \\
&= E_\pi\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a\right] \\
&= \sum_{s', r} p\left(s', r \mid s, a\right) * \left[r + \gamma E_\pi\left[G_{t+1} \mid S_{t+1} = s'\right]\right] \\
&= \sum_{s', r} p\left(s', r \mid s, a\right) * \left[r + \gamma v_\pi\left(s'\right)\right] \\
&= \sum_{s', r} p\left(s', r \mid s, a\right) * \left[r + \gamma \sum_{a' \in A} \left(\pi\left(a' \mid s'\right) * q_\pi\left(s', a'\right)\right)\right]
\end{aligned}
$$

policy-comparison:

$$\pi' \geq \pi \quad \longleftrightarrow \quad v_{\pi'}(s) \geq v_{\pi}(s) \quad \forall s \in S$$

policy-improvement:

$$E_{\pi'}\left[q_{\pi}\left(s, \pi'(s)\right)\right] \geq v_{\pi}(s) = E_{\pi}\left[q_{\pi}\left(s, \pi(s)\right)\right] \quad \forall s \in S$$

Bellman Optimal Equations:

$$\begin{aligned} v_*(s) &= \max_{\pi} v_{\pi}(s) \\ &= \max_{\pi}\left(r_{\pi} + \gamma P_{\pi} v\right) \\ &= \max_{a \in A} q_{\pi*}(s, a) \quad \forall s \in S \end{aligned}$$

Contraction Mapping Theorem (迭代收敛至唯一不动点)
贝尔曼最优方程的收缩迭代过程，即是value iteration算法

ch3: Dynamic Programming (dynamics with model p)

(1) Value Iteration:

$$v_{k+1} = \max_{\pi} (r_\pi + \gamma P_\pi v_k)$$

Policy Update:

$$\pi_{k+1} = \arg\max_{\pi} (r_\pi + \gamma P_\pi v_k)$$

Value Update:

$$v_{k+1} = r_{\pi+1} + \gamma P_{\pi+1} v_k$$

(2) Policy Iteration:

Policy Evaluation: (matrix solution vs. iteration solution)

$$v_{\pi_k} = r_{\pi_k} + \gamma P_{\pi_k} v_{\pi_k}$$

Policy Improvement:

$$\pi_{k+1} = \arg\max_{\pi} (r_\pi + \gamma P_\pi v_k)$$

(3) Turncated Iteration:

1 Monte Carlo

(1) 基于数据 trajectory 的价值估计、策略改进:

2 Temporal Difference

(1) 基于数据 transition 的价值估计、策略改进: