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:
①reward、return、value，用来评估一个策略的好坏。
策略与价值一一对应。
②价值比较，策略比较，策略改进定理。
③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。
最优动作价值，意味着选取这个动作，未来回报的期望最大。
④r线性变换，V+Q线性变换，改变最优价值，不改变greedy最优策略。
⑤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。
⑥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:

值迭代有限次数，介于1次与无穷次之间；值也未稳定，就进行策略改进

ch4: Monte Carlo (model-free, dynamics with trajectory)

Sample:

$$v_\pi(s) = E_\pi\left[G_t \mid S_t = s\right]$$
$$q_\pi(s,a) = E_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

理解:
①采样进行估计，基于概率论的大数定理。
②episode长度（探索半径是否覆盖终点？）对估值影响，最优价值是否反向传播。
③估计的更新方式，非增长式（等着一起算）和增长式（来一个算一个）。
④epsilon关乎采样策略的探索性和最优性，
大则探索性强、最优性弱，小则探索性弱、最优性强，
⑤如果epsilon大到一定程度，可能会导致epsilon-greedy与最优greedy不一致。

(1) MC-Basic
二次循环，遍历所有(s, a)；某个策略下，每对采足够样，非增长式估计相应Q。
策略相应的，一套稳定Q值下，策略改进。
迭代。

(2) MC-Exploring-Starts
起始分布覆盖(s, a)全集。
Pi下，充分利用每一个trajectory里的所有(s, a)对，访问，即增长式估计相应Q。
每一个trajectory结束后，Q值未必稳定，都进行策略改进。
迭代。

(3) MC-epsilon-greedy
过程分布覆盖(s, a)全集。
e-Pi下，充分利用每一个trajectory里的所有(s, a)对， 访问，即增长式估计相应Q。
每一个trajectory结束后，Q值未必稳定，都进行策略改进，生成e-Pi。
迭代。

*stochastic approximation

(1) Incremental-Estimation:

$$w_k = \frac{1}{k}\sum_{i=1}^{k} x_i \qquad\qquad w_{k-1} = \frac{1}{k-1}\sum_{i=1}^{k-1} x_i$$

$$w_k = \frac{1}{k}\left[(k-1)w_{k-1} + x_k\right] \quad = w_{k-1} + \frac{1}{k}\left[x_k - w_{k-1}\right]$$

(2) Robbins-Monro:

$$g(w) = 0 \qquad \text{(g is unknown, w is input, 0 is output)}$$

$$g(w) = \nabla_w L(w) = 0 \quad \text{(w is parametres)}$$

$$g(w) = L(w) - C = 0$$

$$\text{w* is the solution (Convergence Condition)}$$

$$w_{k+1} = w_k + a_k\left[\tilde{g}\left(w_k, \eta_k\right) - 0\right]$$

$$= w_k + a_k\left(g(w_k) + \eta_k\right)$$

$$iteration:\ \{w_k\} + \{\tilde{g}_k\} + \{a_k\}$$

(3) Optimazation:

$$\min_w J(w) = E\left[f(w, X)\right]$$

$$\Rightarrow\Rightarrow \qquad \nabla_w E\left[f(w, X)\right] = 0$$

$$\Rightarrow\Rightarrow \qquad Gradient: \quad InputSpace, \quad direction + magnitude$$

GD + (Mini)Batch GD + Stochastic GD:

$$w_{k+1} = w_k - \alpha_k \nabla_w E\left[f(w_k, X)\right] = w_k - \alpha_k E\left[\nabla_w f(w_k, X)\right]$$

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$$

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k)$$

ch5: Temporal Difference

(1) 基于数据 transition 的价值估计、策略改进: