第三课 无模型价值函数估计和控制 下

1. $generalized\ policy\ iteration\ (GPI)$
2. $MC\ with\ exploring\ start$
    - 为保证PI收敛，要求$episode$有ES，即每个行为在$episode$行走无限次后总能被取到
    - 具体算法

$$
\begin{aligned}
&for\ all\ s \in S, initialize\ \pi(s) \in \mathcal{A}(s)\ arbitrarily\\
&initialize\ Q(s,a) \in \mathbb{R}\ for\ all\ s \in S\ and\ a \in \mathcal{A}(s)\\
&Returns(s,a) \leftarrow empty\ list\ for\ all\ s \in S\ and\ a \in \mathcal{A}(s)\\
&loop\ forever:\\
&\qquad \text{任取}S_0 \in S, A_0 \in \mathcal{A}(s)(\text{其中所有序对取到的概率大于}0)\\
&\qquad \text{以}\pi\text{为策略从}, S_0, A_0\text{开始生成一个}episode: S_0A_0R_1 \cdots S_{T-1}A_{T-1}R_T\\
&\qquad for\ each\ step\ of\ episode\ t = T-1, T-2 \cdots, 1, 0:\\
&\qquad\qquad G \leftarrow \gamma G + R_{t+1}\\
&\qquad\qquad \text{若}S_t, A_t\text{没出现在}S_0, A_0, \cdots, S_{t-1}, A_{t-1}\text{中：}\\
&\qquad\qquad\qquad append\ G\ to\ Returns(S_t, A_t)\\
&\qquad\qquad\qquad Q(S_t, A_t) \leftarrow average(Returns(S_t, A_t))\\
&\qquad\qquad\qquad \pi(S_t) \leftarrow \underset{a}{argmax}\ Q(S_t, a)
\end{aligned}
$$

3. $\epsilon - greedy\ exploration$
    - 确保不断的$explore$，即使用新的行为
    - 具体地：
        - 所有行动出现概率大于0
        - 有$1-\epsilon$的概率选择$greedy\ action$
        - 有ϵ的几率随机选择行为（$explore$）

$$
\pi(a|s) = \begin{cases} \frac{\epsilon}{|A|} + 1 - \epsilon & if\ a^\star = \underset{a \in A}{argmax}\ Q(s,a) \\ \frac{\epsilon}{|A|} & else \end{cases}
$$

    - $policy\ improvement$保证$\epsilon - greedy\ exploration$单调递增

$$
\begin{aligned}
q_\pi(s, \pi'(s,a)) &= \sum_{a \in A} \pi'(a|s) q_\pi(s,a)\\
&= \frac{\epsilon}{|A|} \sum_{a \in A} q_\pi(s,a) + (1-\epsilon) \underset{a}{max}\ q_\pi(s,a)\\
&\geq \frac{\epsilon}{|A|} \sum_{a \in A} q_\pi(s,a) + (1-\epsilon) \sum_{a \in A} \frac{\pi(a|s) - \frac{\epsilon}{|A|}}{1-\epsilon} q_\pi(s,a)\\
&= \sum_{a \in A} \pi(a|s) q_\pi(s,a)\\
&= v_{pi}(s)
\end{aligned}
$$

$$
\Rightarrow v_{\pi'}(s) = max\ q_\pi(s, \pi'(s)) \geq v_\pi(s)
$$

4. $MC\ with\ \epsilon - greedy\ exploration$

$$
\begin{aligned}
&initialize\ Q(S,A)=0, N(S,A)=0, \epsilon=1, k=1\\
&\pi_k=\epsilon-greedy(Q)\\
&loop\\
&\qquad generalize\ k_{th}\ episode\ (S_1A_1R_2\cdots S_T)\sim\pi_k\\
&\qquad for\ each\ S_t,A_t\ inepisode\ do:\\
&\qquad\qquad N(S_t,A_t)\leftarrow N(S_t,A_t)+1\\
&\qquad\qquad Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\frac{1}{N(S_t,A_t)}(G_t-Q(S_t,A_t))\\
&\qquad end\ for\\
&\qquad k\leftarrow k+1, \epsilon\leftarrow\frac{1}{k}\\
&\qquad \pi_k=\epsilon-greedy(Q)\\
&end\ loop
\end{aligned}
$$

5. 用TD代替MC，并使用$\epsilon-greedy$，得到$sarsa$算法（$on-policy\ TD\ control$）

$$
\begin{aligned}
&initialize\ Q(s,a)\ for\ all\ s\in S\ and\ a\in\mathcal{A}(s)\ and\ Q(terminal\ state,\cdot)=0\\
&for\ each\ episode:\\
&\qquad initialize\ S\\
&\qquad choose\ A\ through\ S\ and\ Q(s,a)\ and\ \epsilon-greedy\\
&\qquad for\ each\ step\ of\ episode:\\
&\qquad\qquad take\ action\ A\ and\ get\ R,S'\\
&\qquad\qquad choose\ A'\ through\ S'\ and\ Q\\
&\qquad\qquad Q(S,A)\leftarrow Q(S,A)+\alpha(R+\gamma Q(S',A')-Q(S,A))\\
&\qquad\qquad S\leftarrow S', A\leftarrow A'\\
&\qquad until\ terminal\ state
\end{aligned}
$$

6. $n-step\ sarsa$：调整$sarsa$向前步数（$n=1$即为普通$sarsa$）

$$
\begin{aligned}
&n=1: q_t^{(1)}=R_{t+1}+\gamma Q(S_{t+1},A_{t+1})\\
&n=2: q_t^{(2)}=R_{t+1}+\gamma R_{t+2}+\gamma^2 Q(S_{t+2},A_{t+2})\\
&\vdots\\
&n=\infty: q_t^{(\infty)}=R_{t+1}+\gamma R_{t+2}+\cdots+\gamma^{T-t-1}R_T\ (MC)
\end{aligned}
$$

其对于Q函数的改进对应地变成$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha(q_t^{(n)}-Q(S_t,A_t))$

7. $on-policy\ learning\ vs\ off-policy\ learning$
    - $on-policy$：通过从π中收到的经验学习π，为了$explore$所有行为，可能并非最佳，然后不断减少$explore$的程度
    - $off-policy$：通过从策略μ中收到的经验学习策略π
        - μ：行为策略，更加$explore$，且为生成轨迹的策略；π：目标策略，待学习并成为最优策略
        - 具体地：$S_1A_1R_2\cdots S_T\sim\mu$，用$S_1A_1R_2\cdots S_T$更新$\pi$
        - 优势
            - 跟随更加探索的策略学习得到最佳策略
            - 可以从人的观察或者其他智能体上进行学习
            - 可重复利用其它旧策略$\pi_1,\pi_2,\cdots$
8. $off-policy\ learning\ with\ Q-learning$
    - $target\ policy\ \pi$：$greedy\ on\ Q(S,A)$，即$\pi(S_{t+1})=\underset{A'}{argmax}\ Q(S_{t+1},A')$
    - $behavior\ policy\ \mu$：跟随$\epsilon-greedy$的$Q(S,A)$进行提升

- 具体地：

$$
\begin{aligned}
R_{t+1} + \gamma Q(S_{t+1}, A') &= R_{t+1} + \gamma Q(S_{t+1}, \underset{A'}{argmax}\ Q(S_{t+1}, A')) \\
&= R_{t+1} + \gamma \underset{A'}{max}\ Q(S_{t+1}, A')
\end{aligned}
$$

- 具体算法和上面的$on-policy$大致相同，区别在于如何更新Q函数，需要将$TD\ target$改成本标号下上面所述的$TD\ target$