强化学习入门三之SARSA和 Q-learning

-----By 老老老童

本系列视频理论部分很多都是参考西湖大学赵世钰老师在B站的视频，因此大家看完老师对应章节，再来看这部分偏代码实践的教程更好。

赵世钰老师课程视频地址（强推，讲的非常好）：
https://www.bilibili.com/video/BV1sd4y167NS/?spm_id_from=333.999.0.0&vd_source=6701b0e4f68084bbd3ea4661adf42933

针对赵世钰老师视频，有位大佬开源了其代码，具体源码我没仔细看，不过代码整体风格还是非常优雅（大家根据自己情况来选择性的参考）：
https://github.com/jwk1rose/RL_Learning
B站：
https://www.bilibili.com/video/BV1HX4y1H7uR/?vd_source=6701b0e4f68084bbd3ea4661adf42933

如果上述基础过完之后，推荐另外一位UP主的强化学习视频，可以继续进阶一下：
https://www.bilibili.com/video/BV1X94y1Y7hS/?spm_id_from=333.999.0.0&vd_source=6701b0e4f68084bbd3ea4661adf42933

目录

01 前言

在观看本视频之前，需要你对赵世钰老师如下视频中的内容有了解（包括前面的课程），否则你直接上来看本视频可能会

P24	第6课-随机近似与随机梯度下降（通过例子介...	10:27
P25	第6课-随机近似与随机梯度下降（Robbins-Mo...	10:37
P26	第6课-随机近似与随机梯度下降（Robbins-Mo...	14:05

P31	第7课-时序差分方法（例子）	08:33
P32	第7课-时序差分方法（TD算法介绍）	13:04
P33	第7课-时序差分方法（TD算法收敛性、与MC的...	15:39
P34	第7课-时序差分方法（Sarsa）	15:42
P35	第7课-时序差分方法（Expected Sarsa 和n-ste...	13:40
P36	第7课-时序差分方法（Q-learning介绍、on-po...	17:07
P37	第7课-时序差分方法（Q-learning伪代码与例子）	08:29
P38	第7课-时序差分方法（TD算法的统一形式和总...	09:48

赵世钰老师课程视频地址（强推，讲的非常好）：
https://www.bilibili.com/video/BV1sd4y167NS/?spm_id_from=333.999.0.0&vd_source=6701b0e4f68084bbd3ea4661adf42933

理论基础----通俗易懂

01 理论基础—简单示例

对于一个随机变量X，需要估计其期望 $E(X)$

解答：

假设我们在独立同分布的情况下采集到一组数据 $\{x_i\}_{i=1}^{N}$ 。X的期望可由如下式子估计出

$$E(X) \approx \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (N\text{趋向于无穷大})$$

同时假设 $w_{k+1} = \frac{1}{k}\sum_{i=1}^{k} x_i \quad k = 1,2,3,4...$ ，因此有 $w_k = \frac{1}{k-1}\sum_{i=1}^{k-1} x_i \quad k = 2,3,4...$

我们得到如下式子：

$$w_{k+1} = \frac{1}{k}\sum_{i=1}^{k} x_i = \frac{1}{k}(\sum_{i=1}^{k-1} x_i + x_k)$$

$$= \frac{1}{k}((k-1)w_k + x_k) = w_k - \frac{1}{k}(w_k - x_k)$$

更一般化，我们可以将上述式子转写为 $w_{k+1} = w_k - \alpha_k(w_k - x_k)$ ，其中 $\alpha_k > 0$ ，在应实际应用中一般取(0,1)之间的一个数。这是一种随机逼近（Stochastic Approximation，SA）算法的特殊形式，也是一种随机梯度下降算法的特殊形式。

01 理论基础—Robbins-Monro (RM) 算法

罗宾斯门罗算法，随机逼近领域的一个开创新工作。

问题：求解 $g(w)=0$ 的根。

解答：基于RM算法可以解决上述问题。求解公式如下

$$w_{k+1}=w_k-\alpha_k\tilde{g}(w_k,\eta_k)$$

其中参数含义如下

w_k：是根的第k次的估计值；

$\tilde{g}(w_k,\eta_k)=g(w_k)+\eta_k$：是第k次带有噪声的观测值；

α_k：是一个正系数；

因为我们不知道$g(w)$数学表达式，该算法依赖数据，将其视为黑盒，我们输入一组数据，得到一组输出序列。通过这组序列基于RM算法就可以估算出 $g(w)=0$的根。

输入：$\{w_1,w_2,....w_k,w_{k+1}\}$

输出：$\{\tilde{g}(w_1,\eta_1),\tilde{g}(w_2,\eta_2),.....,\tilde{g}(w_k,\eta_k),\tilde{g}(w_{k+1},\eta_{k+1})\}$

01 理论基础—Robbins-Monro (RM) 算法（简单例子）

问题：求解$g(w)=w-10$ 的根。

解答：基于RM算法，我们取 $w_1=20, \alpha_k=0.5, \eta_k=0$ (假设没有观测误差)。

$$w_1=20 \Rightarrow g(w_1)=10$$
$$w_2=w_1-\alpha_1 g(w_1)=20-0.5*10=15 \Rightarrow g(w_2)=5$$
$$w_3=w_2-\alpha_2 g(w_2)=15-0.5*5=12.5 \Rightarrow g(w_3)=2.5$$
$$\ldots$$
$$w_k \to 10$$

那什么时候可以使用RM算法求解呢？？？ （我直接截屏的西湖大学赵世钰老师课件）

Theorem (Robbins-Monro Theorem)

In the Robbins-Monro algorithm, if

1) $0 < c_1 \leq \nabla_w g(w) \leq c_2$ *for all* w;
2) $\sum_{k=1}^{\infty} a_k = \infty$ *and* $\sum_{k=1}^{\infty} a_k^2 < \infty$;
3) $\mathbb{E}[\eta_k | \mathcal{H}_k] = 0$ *and* $\mathbb{E}[\eta_k^2 | \mathcal{H}_k] < \infty$;

where $\mathcal{H}_k = \{w_k, w_{k-1}, \ldots\}$, *then* w_k *converges with probability 1 (w.p.1) to the root* w^* *satisfying* $g(w^*) = 0$.

01 理论基础—Robbins-Monro (RM) 算法（求 $E(X)$）

问题：考虑函数 $g(w) = w - E(X)$ 的根，我们就可以得到 $E(X)$

解答：基于RM算法可以解决上述问题。

我们可以得到观测值

$$\tilde{g}(w, x) = w - x$$

注意

$$\begin{aligned}\tilde{g}(w, \eta) = w - x &= w - x + E(X) - E(X) \\ &= (w - E(X)) + (E(X) - x) \\ &= g(w) + \eta\end{aligned}$$

应用RM算法

$$w_{k+1} = w_k - \alpha_k \tilde{g}(w_k, \eta_k) = w_k - \alpha_k (w_k - x_k)$$

S A R S A

01 SARSA

在策略π 下有，$q_\pi(s,a) = E\left[R + \lambda q_\pi(S', A') \mid s,a\right]$ (具体证明见老师书第七章)

考虑式子 $g(q(s,a)) = q(s,a) - E\left[R + \lambda q_\pi(S', A') \mid s,a\right]$ 利用RM算法，我们有得到SARSA算法

$$q_{t+1}(s_t,a_t) = q_t(s_t,a_t) - \alpha_t(s_t,a_t)\left[q_t(s_t,a_t) - \left[r_{t+1} + \lambda q_t(s_{t+1},a_{t+1})\right]\right]$$
$$q_{t+1}(s,a) = q_t(s,a), \quad \forall(s,a) \neq (s_t,a_t)$$

在SARSA算法中，字母‘S、A、R、S、A’分别代表, s_t、a_t、r_{t+1}、s_{t+1}、a_{t+1}

解贝尔曼方程

问题：这里RM算法在这里承担的作用是什么呢？？？？？

其实就是在评估在策略π 下的 $q_{t+1}(s_t,a_t)$，就是求解在策略π下的贝尔曼方程

01 贝尔曼方程---具体细节见赵世钰老师第二课视频

给定策略π，我们可以得到如下贝尔曼方程（矩阵形式）：

$$v_\pi = r_\pi + \lambda P_\pi v_\pi$$

综合上一页PPT可知，在给定策略π的情况下，我们可以将矩阵变换一下，直接求得v_π。但在实际中，当状态数量很大时，矩阵维度也很大，运算效率低，因此我们一般使用值迭代的方式来求解。

值迭代方式如下：

给定策略π，随机初始化v_0 我们有如下迭代过程

$$v_1 = r_\pi + \lambda P_\pi v_0$$
$$v_2 = r_\pi + \lambda P_\pi v_1$$
$$v_3 = r_\pi + \lambda P_\pi v_2$$
$$\cdots\cdots$$
$$v_n = r_\pi + \lambda P_\pi v_{n-1}$$

当n趋于无穷时,v_n是收敛于v_π的。（具体证明见老师第二课PPT）

解贝尔曼方程

01 SARSA

Sarsa算法（详情见老师PPT）

Pseudocode: Policy searching by Sarsa

For each episode, do

  If the current s_t is not the target state, do

    Collect the experience $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$: In particular, take action a_t following $\pi_t(s_t)$, generate r_{t+1}, s_{t+1}, and then take action a_{t+1} following $\pi_t(s_{t+1})$.

    Update q-value:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\Big[q_t(s_t, a_t) - [r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1})]\Big]$$

    Update policy:

$$\pi_{t+1}(a|s_t) = 1 - \frac{\epsilon}{|\mathcal{A}|}(|\mathcal{A}| - 1) \text{ if } a = \arg\max_a q_{t+1}(s_t, a)$$

$$\pi_{t+1}(a|s_t) = \frac{\epsilon}{|\mathcal{A}|} \text{ otherwise}$$

ao

01 Expected-SARSA

Sarsa算法（详情见老师PPT）

Pseudocode: Policy searching by Sarsa

For each episode, do

If the current s_t is not the target state, do

Collect the experience $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$: In particular, take action a_t following $\pi_t(s_t)$, generate r_{t+1}, s_{t+1}, and then take action a_{t+1} following $\pi_t(s_{t+1})$.

Update q-value:

$$\boxed{q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\Big[q_t(s_t, a_t) - [r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1})]\Big]}$$

Update policy:

$\pi_{t+1}(a|s_t) = 1 - \frac{\epsilon}{|\mathcal{A}|}(|\mathcal{A}| - 1)$ if $a = \arg\max_a q_{t+1}(s_t, a)$

$\pi_{t+1}(a|s_t) = \frac{\epsilon}{|\mathcal{A}|}$ otherwise

01 N-step-SARSA

Sarsa算法（详情见老师PPT）

Pseudocode: Policy searching by Sarsa

For each episode, do

If the current s_t is not the target state, do

Collect the experience $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$: In particular, take action a_t following $\pi_t(s_t)$, generate r_{t+1}, s_{t+1}, and then take action a_{t+1} following $\pi_t(s_{t+1})$.

Update q-value:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\Big[q_t(s_t, a_t) - [r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1})]\Big]$$

Update policy:

$$\pi_{t+1}(a|s_t) = 1 - \frac{\epsilon}{|\mathcal{A}|}(|\mathcal{A}| - 1) \text{ if } a = \arg\max_a q_{t+1}(s_t, a)$$
$$\pi_{t+1}(a|s_t) = \frac{\epsilon}{|\mathcal{A}|} \text{ otherwise}$$

n-step Sarsa: can unify Sarsa and Monte Carlo learning

The definition of action value is

$$q_\pi(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a].$$

The discounted return G_t can be written in different forms as

$$\begin{aligned}
\text{Sarsa} \longleftarrow \quad & G_t^{(1)} = R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}), \\
& G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, A_{t+2}), \\
& \vdots \\
n\text{-step Sarsa} \longleftarrow \quad & G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n q_\pi(S_{t+n}, A_{t+n}), \\
& \vdots \\
\text{MC} \longleftarrow \quad & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots
\end{aligned}$$

It should be noted that $G_t = G_t^{(1)} = G_t^{(2)} = G_t^{(n)} = G_t^{(\infty)}$, where the superscripts merely indicate the different decomposition structures of G_t.

考虑N个状态动作对(s,a)来求动作价值

Q-learning

01 Q-learning

Q-learing算法核心，就是在SARSA基础上进一步做了改进

$$q_{t+1}(s_t,a_t)=q_t(s_t,a_t)-\alpha_t(s_t,a_t)\left[q(s_t,a_t)-\left[r_{t+1}+\lambda\max_{a\in A}q_t(s_{t+1},a)\right]\right]$$

$$q_{t+1}(s,a)=q_t(s,a),\quad \forall(s,a)\neq(s_t,a_t)$$

其求解的是如下的一个期望，Q-learing是在求解贝尔曼最优方程

$$q(s,a)=E\left[R+\lambda\max_{a}q(S_{t+1},a)\mid S_t=s,A_t=a\right],\quad \forall s,a$$

01 Q-learning

Q-learning算法（详情见老师PPT）

在线学习

Pseudocode: Policy searching by Q-learning $\boxed{\text{(on-policy version)}}$

For each episode, do

If the current s_t is not the target state, do

Collect the experience $(s_t, a_t, r_{t+1}, s_{t+1})$: In particular, take action a_t following $\pi_t(s_t)$, generate r_{t+1}, s_{t+1}.

Update q-value:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\Big[q_t(s_t, a_t) - [r_{t+1} + \gamma \max_a q_t(s_{t+1}, a)]\Big]$$

Update policy:

$$\pi_{t+1}(a|s_t) = 1 - \frac{\epsilon}{|\mathcal{A}|}(|\mathcal{A}| - 1) \text{ if } a = \arg\max_a q_{t+1}(s_t, a)$$

$$\pi_{t+1}(a|s_t) = \frac{\epsilon}{|\mathcal{A}|} \text{ otherwise}$$

离线学习

Pseudocode: Optimal policy search by Q-learning $\boxed{\text{(off-policy version)}}$

For each episode $\{s_0, a_0, r_1, s_1, a_1, r_2, \ldots\}$ generated by π_b, do

For each step $t = 0, 1, 2, \ldots$ of the episode, do

Update q-value:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\Big[q(s_t, a_t) - [r_{t+1} + \gamma \max_a q_t(s_{t+1}, a)]\Big]$$

Update target policy:

$$\pi_{T,t+1}(a|s_t) = 1 \text{ if } a = \arg\max_a q_{t+1}(s_t, a)$$

$$\pi_{T,t+1}(a|s_t) = 0 \text{ otherwise}$$

behavior policy：用于采样数据

target policy：最后我们实际用于生产环境的最优策略

离线学习：behavior policy和target policy不一样，最大的好处，behavior policy在设定的时候，可以使其探索性更强

在线学习：behavior policy和target policy相同

完结散花