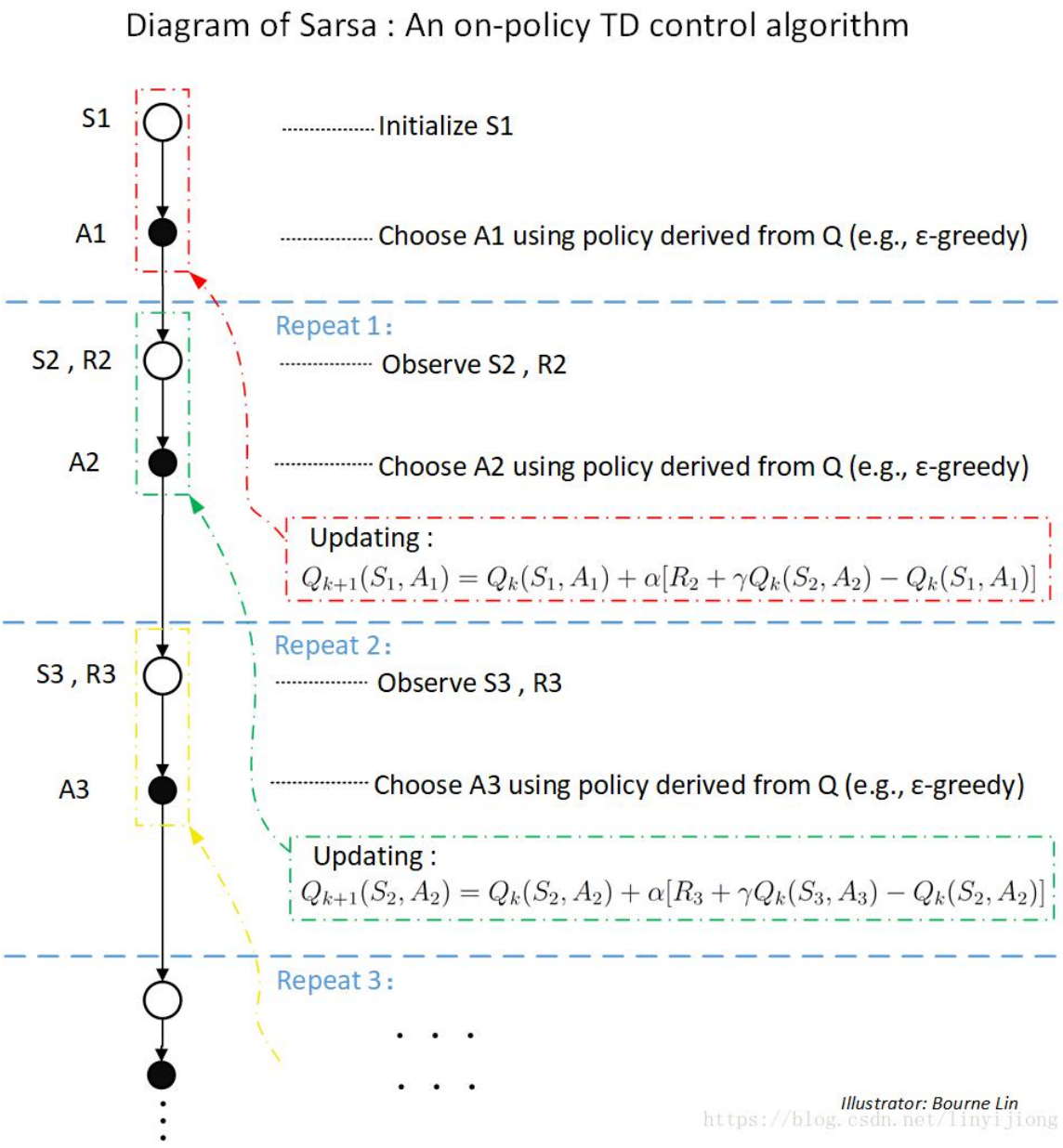
Sarsa 和 Q-learning 教程

视频教程：https://www.bilibili.com/video/av45015646/?p=7
https://www.bilibili.com/video/av45015646/?p=8
https://www.bilibili.com/video/av45015646/?p=9

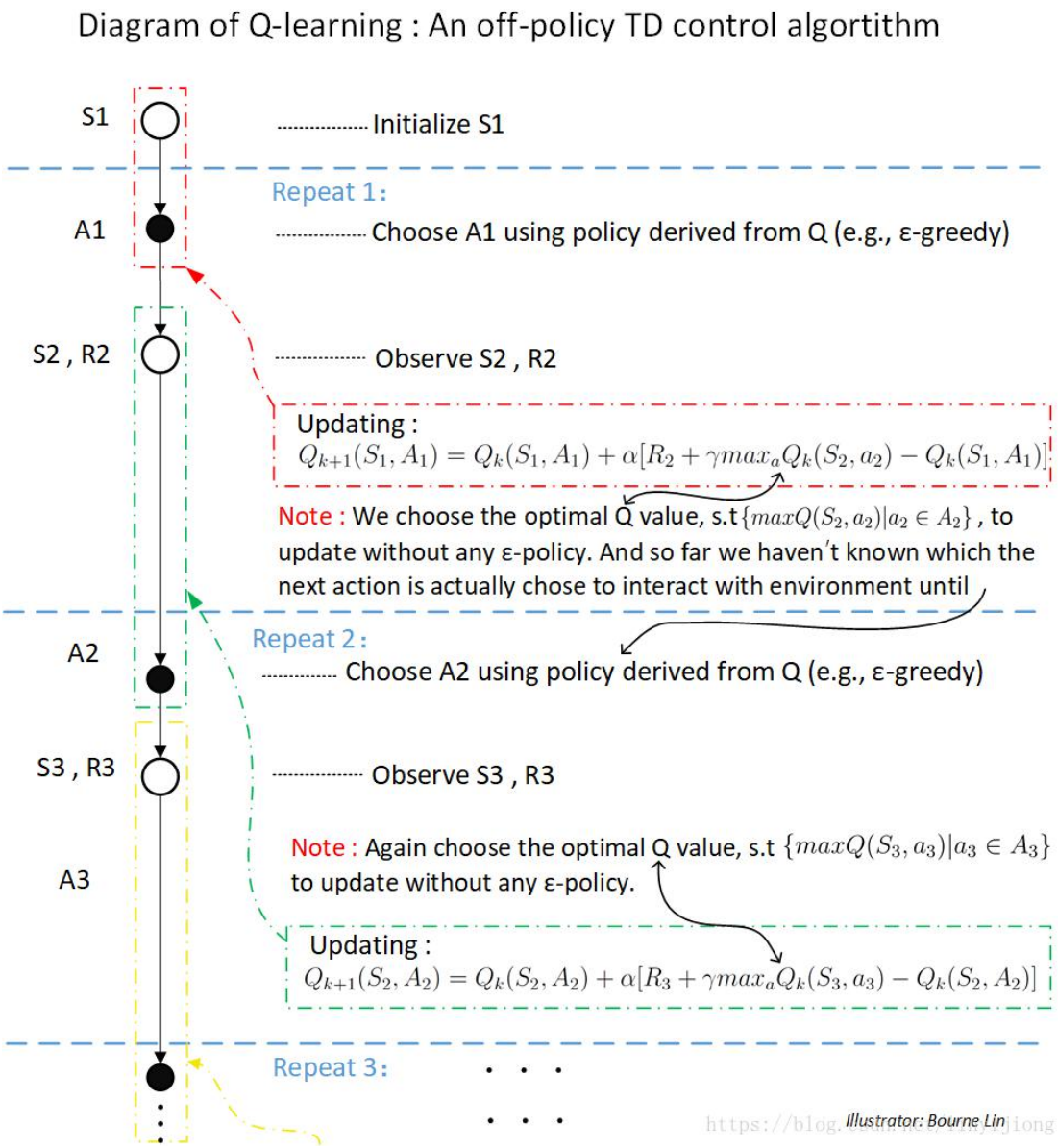
Sarsa



1.1) 一个回合（Episode）开始，随机选择（初始化）第一个状态S_1。并基于 ε-greedy策略在状态S_1中选择动作，有两种情况，一是有$(1-\varepsilon)$的概率直接选择具有最大值Q的动作，二是有εε 概率随机选择 S_1 下的任意动作（在第二种情况下每个动作的概率均为$\varepsilon/|A_1|$，其中$|A_1|$为S_1下的动作总个数）。

1.2) 进入第一次循环(Repeat 1 / Step 1)：执行 A_1之后（与环境互动），观察下一个状态 S_2，并马上得到 S_2的即时回报 R_2。此时，再次基于 ε-greedy策略，在状态S_2中选择动作A_2。得到 A_2 后，即可进行Q函数的更新（Update），更新中的 $Q_k(S_2,A_2)$ 为ε-greedy策略下所随机选取的动作，这是与Q-learning的不同之处！（下标 k 或 i 表示最近一次更新的Q值，是一个迭代序数而非时间步（step）序数，在此可先忽略。）

Q-Learning



2.1) 一个回合（Episode）开始，随机选择（初始化）第一个状态S_1。

2.2) 进入第一次循环（Repeat 1 / Step 1）：首先基于ε-greedy策略在状态S_1中选择动作。选择并执行A_1之后（与环境互动），观察下一个状态S_2，并马上得到S_2的即时回报R_2。此时，立即进行Q函数的更新（Update），更新中的$max_a Q_k(S_2, a_2)$为我们人为直接选择S_2下所有动作中具有最大Q值的动作，这就是与Saras根本区别！

2.3) 更新完毕后，进入第二次循环：基于ε-greedy策略，在状态S_2中选择动作与环境互动（此前在状态S_2时候并未采取动作与环境互动）。值得注意的是，我们在循环1中更新（Update）时所选取S_2的动作a_2是唯一的(人为强制选择)，即最具有最大价值Q的动作$max_a Q_k(S_2, a_2)$；而循环2中作为需要与环境互动的第二次动作A_2则是基于ε-greedy策略（即在此时究竟选取$max_a Q_k(S_2, a_2)$对应的动作a_2还是其他动作完全根据是随机选择，听天由命吧 0.0 ）！因此，基于ε-greedy策略，与环境互动、做学习训练时做动作选择的决策（在off-policy中这被称为行为策略）与Sarsa是一致的。

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

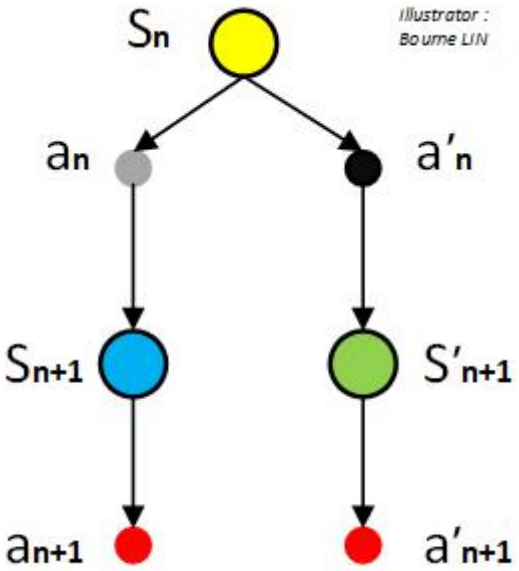
Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

大写的 A 表示集合，比如 则表示 下的所有动作，而 则表示具体的一个动作，它们之间的关系为：。回到流程图中，可以发现出现 a 都在 Q-learning 的 update 公式中，这是因为我们在更新时，人为指定选择具有最大值 Q 的 a，这是具有确定性的事件（Deterministic）。而在 Q-learning 中与环境互动的环节、在 Sarsa 中更新 Q 值的环节与环境互动的环节时，动作的选择是随机的（ ε -greedy），因此所有动作都有可能被选中，只不过是具有最大值 Q 的动作被选中的概率大。

Q-learning 如何实现更加有效的探索？

清楚整个流程之后，我们来具体看看，Q-learning 到底是怎么实现有意义的探索，如何在环境中发掘出更有价值的动作？（即一个当前估值（evaluate）不高但潜力巨大的动作的逆袭之路）



第一个简单的栗子

在这个例子中，我们将更新黄色状态的动作价值$Q(S = yellow, A)$。

假设已知黄色状态下只有两个动作可选：灰动作和黑动作。而蓝色、绿色状态下最大价值动作均为红动作，其他动作暂不列出（因为在Q-learning中更新时，人为强制选择下一状态中最大价值的动作，因此同一状态下的其他动作在更新环节没有任何体现）。

在某个回合（episode）中，在时间步为n的时候（time step = n），所处状态为黄色。并且在第k-1次更新 $Q(S = yellow, A)$时，（ $Q_1(S_n, a_n)$ 通过即时奖励获取，不纳入更新迭代次数k），已知灰动作价值比黑动作大，即有

$$Q_k(S_n, a_n = gray) > Q_k(S_n, a'_n = black)$$

所以基于 ε-greedy策略选择动作，会出现情况①或②：

①有 $(1-\varepsilon)+\varepsilon/2=1-\varepsilon/2$ 的可能性选择当前最大价值Q的灰动作

而另一黑动作没有更新，即 $Q_{k+1}(S_n,a'_n)=Q_k(S_n,a'_n)$ 。

其中， $S_n=yellow$ S_n=yellow 、 $S_{n+1}=blue$ 。

②有 $\varepsilon/2$ ε /2 的可能性选择当前较小价值Q的黑动作 a'_n a_n' ：

$$Q_{k+1}(S_n,a'_n)=Q_k(S_n,a'_n)+\alpha[R_{n+1}+maxQ_i(S'_{n+1},a'_{n+1})-Q_k(S_n,a'_n)]$$

而另一灰动作没有更新，即 $Q_{k+1}(S_n,a_n)=Q_k(S_n,a_n)$ 。

其中， $S_n=yellow$ S_n=yellow 、 $S'_{n+1}=green$ 。

无论发生情况①或是②，黄色状态下的灰动作与黑动作的价值的大小关系都可能发生变化！！我们通过取最大值（即greedy思想）来更新目标策略 π\pi （target policy），：

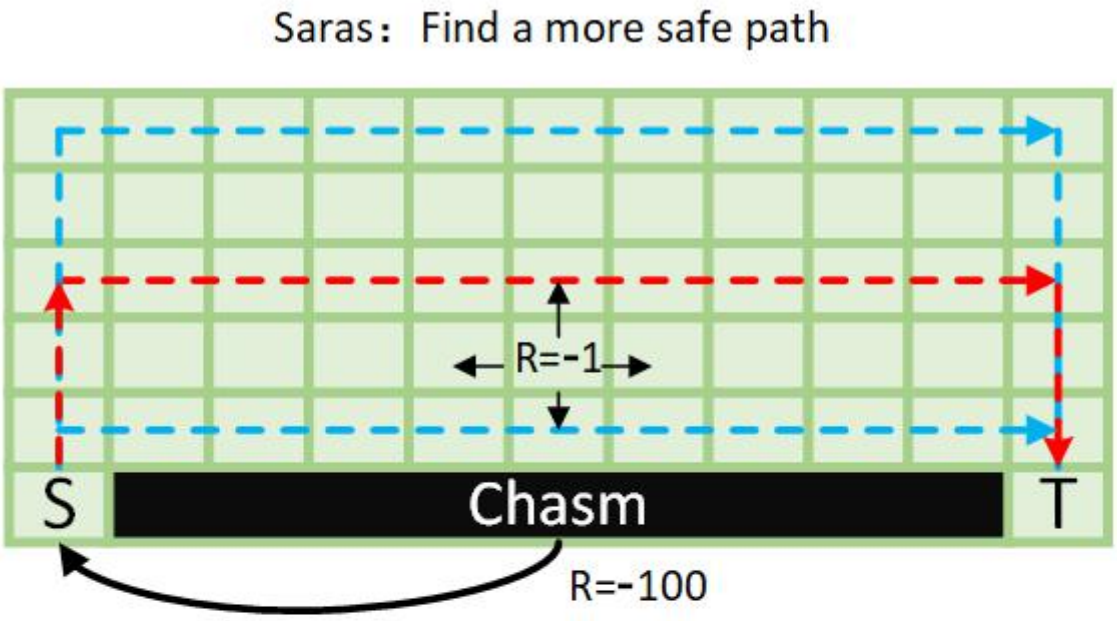
$$Q^{\pi}_{k+1}(S=yellow,A)=max\left\{Q_{k+1}(S_n,a_n),Q_{k+1}(S_n,a'_n)\right\}$$

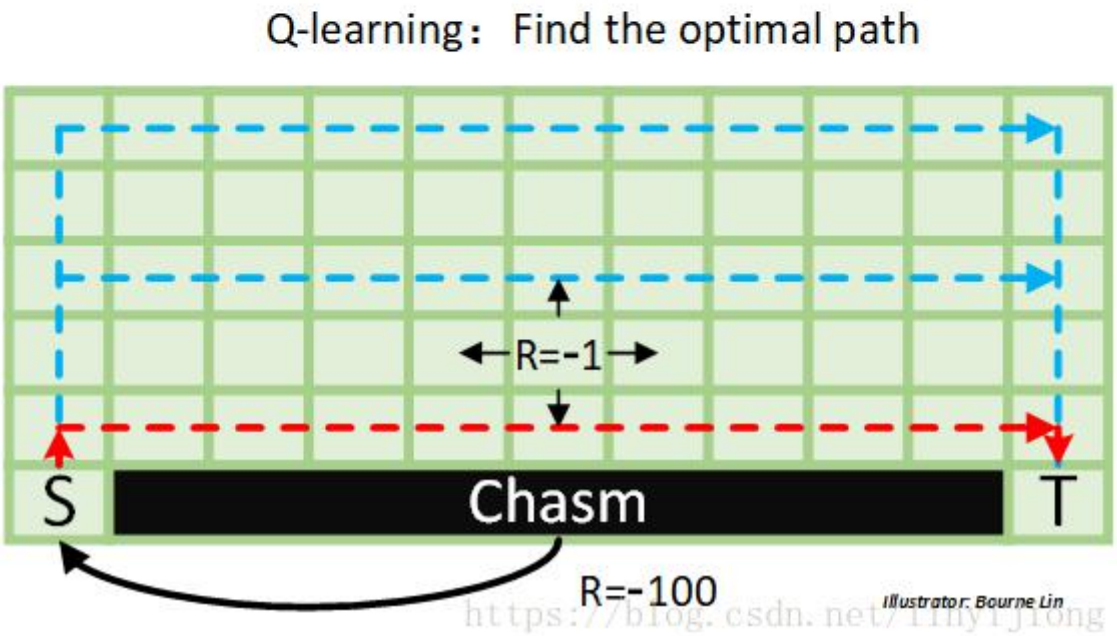
比如，当出现情况②的时候，即探索（explore）了黑动作，更新后有 $Q_{k+1}(S_n,a_n)<Q_{k+1}(S_n,a'_n)$ ，则此时黄色状态下的黑动作变为最优动作（颠覆了灰色动作有最大Q值的地位）。

一个例子

在此举一个非常直观的例子来帮助我们认识一下 Q-learning 和 Sarsa 的实际应用效果的区别。

在下面栅格化的小世界中，绿色区域为草地，在上面每移动一格子就会扣 1 分，而踏入黑色区域的悬崖（chasm），会扣一百分，并且回到起始点 S (Start)。我们希望能学习到一条得分最高的路径到达终点 T (Terminal)。分别使用 Sarsa 和 Q-learning 进行学习。结果如图所示，红色为相应算法的最优路径。





可以看到，Q-learning 寻找到一条全局最优的路径，因为虽然 Q-learning 的行为策略（behavior）是基于 ε-greedy 策略，但其目标策略（target policy）只考虑最优行为；而 Sarsa 只能找到一条次优路径，这条路径在直观上更加安全，这是因为 Sarsa（其目标策略和行为策略为同一策略）考虑了所有动作的可能性（ ε-greedy），当靠近悬崖时，由于会有一定概率选择往悬崖走一步，从而使得这些悬崖边路的价值更低。

总结

Q-learning 虽然具有学习到全局最优的能力，但是其收敛慢；而 Sarsa 虽然学习效果不如 Q-learning，但是其收敛快，直观简单。因此，对于不同的问题，我们需要有所斟酌。

参考文档：https://blog.csdn.net/linyijiong/article/details/81607691