增强学习（五）----- 时间差分学习(Q learning, Sarsa learning)

接下来我们回顾一下动态规划算法(DP)和蒙特卡罗方法(MC)的特点，对于动态规划算法有如下特性：

- 需要环境模型，即状态转移概率Psa
- 状态值函数的估计是自举的(*bootstrapping*)，即当前状态值函数的更新依赖于已知的其他状态值函数。

相对的，蒙特卡罗方法的特点则有：

- 可以从经验中学习不需要环境模型
- 状态值函数的估计是相互独立的
- 只能用于episode tasks

而我们希望的算法是这样的：

- 不需要环境模型
- 它不局限于episode task，可以用于连续的任务

本文介绍的**时间差分学习**(Temporal-Difference learning, TD learning)正是具备了上述特性的算法，它结合了DP和MC，并兼具两种算法的优点。

TD Learing思想

在介绍TD learning之前，我们先引入如下简单的蒙特卡罗算法，我们称为**constant-α MC**，它的状态值函数更新公式如下：

$$(1)V(st)\leftarrow V(st)+\alpha[Rt-V(st)]$$

其中Rt是每个episode结束后获得的实际累积回报，α是学习率，这个式子的直观的理解就是**用实际累积回报Rt作为状态值函数V(st)的估计值**。具体做法是对每个episode，考察实验中st的实际累积回报Rt和当前估计V(st)的偏差值，并用该偏差值乘以学习率来更新得到V(St)的新估值。

现在我们将公式修改如下，把Rt换成rt+1+γV(st+1)，就得到了TD(0)的状态值函数更新公式：

$$(2)V(st)\leftarrow V(st)+\alpha[rt+1+\gamma V(st+1)-V(st)]$$

为什么修改成这种形式呢，我们回忆一下状态值函数的定义：

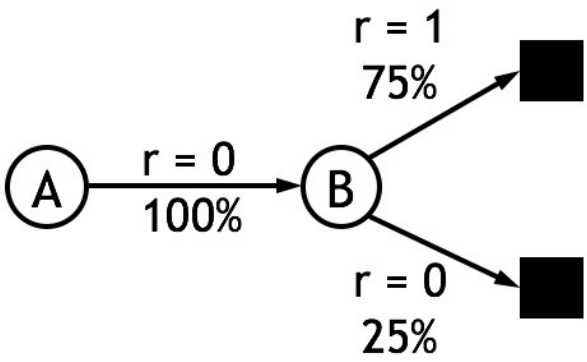
$$(3)V\pi(s)=E\pi[r(s'|s,a)+\gamma V\pi(s')]$$

容易发现这其实是根据(3)的形式，利用真实的立即回报rt+1和下个状态的值函数V(st+1)来更新V(st)，这种就方式就称为时间差分(temporal difference)。由于我们没有状态转移概率，所以要利用多次实验来得到期望状态值函数估值。类似MC方法，在足够多的实验后，状态值函数的估计是能够收敛于真实值的。

那么MC和TD(0)的更新公式的有何不同呢？我们举个例子，假设有以下8个episode, 其中A-0表示经过状态A后获得了回报0：

index	samples
episode 1	A-0, B-0
episode 2	B-1
episode 3	B-1
episode 4	B-1
episode 5	B-1
episode 6	B-1
episode 7	B-1
episode 8	B-0

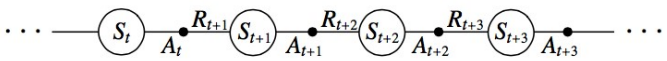
首先我们使用constant-α MC方法估计状态A的值函数，其结果是V(A)=0，这是因为状态A只在episode 1出现了一次，且其累计回报为0。



将式(2)作为状态值函数的估计公式后，前面文章中介绍的**策略估计**算法就变成了如下形式，这个算法称为TD prediction:

输入：待估计的策略π
任意初始化所有V(s)，(e.g.,V(s)=0,∀s∈s+)
Repeat(对所有episode):
    初始化状态 s
    Repeat(对每步状态转移):
        a←策略π下状态s采取的动作
        采取动作a，观察回报r，和下一个状态s′
        V(s)←V(s)+α[r+λV(s′)−V(s)]
        s←s′
    Until st is terminal
  Until 所有V(s)收敛
输出Vπ(s)

Sarsa算法



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

Q-learning

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

小结

本篇介绍了TD方法思想和TD(0),Q(0),Sarsa(0)算法。TD方法结合了蒙特卡罗方法和动态规划的优点，能够应用于无模型、持续进行的任务，并拥有优秀的性能，因而得到了很好的发展，其中Q-learning更是成为了强化学习中应用最广泛的方法。在下一篇中，我们将引入**资格迹(Eligibility Traces)**提高算法性能，结合Eligibility Traces后，我们可以得到Q(λ),Sarsa(λ)等算法

参考资料

[1] R.Sutton et al. Reinforcement learning: An introduction, 1998