





听见下雨的声音

-  首页
-  分类
-  关于
-  归档
-  标签

【David Silver强化学习公开课之九】探索与利用

 发表于 2016-08-05 |  分类于 [project experience](#) |  |  909

本文是David Silver强化学习公开课第九课的总结笔记。这一课主要讲了因为存在Exploration和Exploitation矛盾的问题，从而需要考虑如何达到exploration的目的，提出了三种思路。

【转载请注明出处】chenrudan.github.io

本文是David Silver强化学习公开课第九课的总结笔记。这一课主要讲了因为存在Exploration和Exploitation矛盾的问题，从而需要考虑如何达到exploration的目的，提出了三种思路。

本课视频地址:

[RL Course by David Silver - Lecture 9: Exploration and Exploitation](#)

本课ppt地址:http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/dyna.pdf

文章的内容是课程的一个总结和讨论，会按照自己的理解来组织。个人知识不足再加上英语听力不是那么好可能会有一些理解不准的地方，欢迎一起讨论。

建了一个强化学习讨论qq群，有兴趣的可以加一下群号595176373或者扫描下面的二维码。



1.内容回顾

第一课中讲过Exploration和Exploitation即探索和利用，简单来说就是前者会尝试新的选择而后者会从已有的信息中选择最好的。例如在线投放的广告，前者会投放一些新的广告，而后者会投放最受欢迎的广告。

有三种方式可以达到exploration的目的，第一种就是类似 ϵ -greedy算法在一定概率基础下随机选择一些action，第二种是更加倾向选择更加具有不确定性的状态/动作，这种方法就需要一种方法来衡量这种不确定性，第三种就是收集一些状态信息来确定是否值得到达这个状态。

2. ϵ_t greedy

© 2017 ♥ Rudan Chen

例如k-摇臂赌博机，随机摇动一个摇臂会获得固定的奖励，[N个摇臂会获得不同的奖励](#)，[N个摇臂会获得不同的奖励](#)，动作集是多个摇臂，奖赏是动作的概率分布 $R^a(r) = P[r|a]$ ，在t时刻选择一个摇臂 a_t ，然后environment产生reward r_t ，目标就是最大化累积从开始到t

时刻的 reward $\sum_{\tau}^t r_{\tau}$ 。t时刻动作值函数取值是每次 reward 的均值即 $Q(a) = E[r|a]$ ，最好的是 $V^* = Q(a^*) = \max_{a \in A} Q(a)$ 。在试验过程中t时刻与最优值的差的期望称为regret为 $l_t = E[V^* - Q(a_t)]$ ，累积的total regret则是 $L_t = E[\sum_{\tau=1}^t V^* - Q(a_{\tau})]$ ，并将这个式子表达成两项之积，第一项叫count意义是这个动作的期望次数，第二项叫gap代表这个动作在当前值函数下取值与最优值之差， $L_t = \sum_{a \in A} E[N_t(a)] \Delta_a$ 。因此最大化累积奖赏就等于最小化total regret，引入regret是为了衡量算法好到什么程度。

考虑之前的greedy算法，它永远在选择已知信息中最好的，所以很容易陷入次优值中，并且会导致线性regret，而gap一直都没变。而 ϵ -greedy算法，有 ϵ 的概率会选择任意一个action，因为是任意选择一个，所以最终的regret仍然是个线性的，因为gap也没变，只是可能比greedy的方法变小了一点。所以为了保证尽可能让各种action都能被选择，就可以将值函数的初始值设的比较大，让gap变小，这样就有更多的可能取尝试那个动作。但是这种式仍然改变不了两种方法的regret呈线性，因此做一个改变，将 ϵ 的值不固定，而是采用逐渐减小的 ϵ_t ，这种让regret呈现对数函数形式。

2.Upper Confidence Bounds Algorithm

根据第二种方法，需要选择更加具有不确定性的action，这种不确定性则可以根据该动作被选中的次数即count大小来衡量，如果count比较大说明这个action的信息已经多次被利用了，那么就比较确定这个action会带来reward，如果count小就说明不确定性比较大。(课中讲了一些推导这里略)令不确定性为 $\hat{U}_t(a) = \sqrt{\frac{2 \log t}{N_t(a)}}$ ，选择action的策略就变成了 $a_t = \underset{a \in A}{\operatorname{argmax}} Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$ ，第二项相当于加了一个额外的confidence和值估计一起来判断选择什么动作。

基于贝叶斯定理同样也能用来判断与选择action，reward有先验分布的假设 $p[R]$ ，将动作值函数表示为参数分布 $p[Q|w]$ ，计算w的后验概率 $p[w|R_1, \dots, R_t]$ ，再计算出值函数的后验概率 $p[Q(a)|R_1, \dots, R_t] = p[Q|w]p[w|R_1, \dots, R_t]$ 。可以根据 $\pi(a) = P[Q(a) = \max_{a'} Q(a')|R_1, \dots, R_t]$ (其中a是最优选择)来选择action。

3.Information State Search

根据第三种方法，需要考虑当前获取的information，引入每个时刻的信息状态 $\tilde{s} = f(h_t)$ ，它是history的函数并且也有状态转移概率 $\tilde{P}_{\tilde{s}, \tilde{s}'}^a$ 。例如赌博机的例子中，如果一次以概率 μ_a 赢，如何找到哪个摇臂有最大的概率赢，那么就能加入信息状态 $\tilde{s} = (\alpha, \beta)$ ，第一项记录摇臂输了的次数，第二项记录摇臂赢了的次数，所以信息状态记录了从开始到现在发生过的所有情况。从而可以根据这样的信息做出判断。

4.小结

这节课说实话没听太懂，课上提了很多方法，查了一下资料发现搞懂这些还需要看不少的论文...第十课因为视频的ppt不清晰，就没有看，但是感觉最后一课才是精华啊...因为前面这么多课都在介绍概念，很少有针对性来讲解具体的解法。总结一下这些内容，其实强化学习要解决就是马尔可夫决策过程(MDP)，即执行不同的行动到达不同的状态，它由 (S, A, P, R, γ) 组成，从而可以围绕着求解environment、policy、value function三者来求解的，有时已知environment有时未知environment，有时要显式的算出policy，有时要求的又是action-value function。我觉得与监督学习不同的地方在于MDP当前的某个状态下对最终的结果的影响是不定的，往往是经过过去的一段行为和将来的一段行为才能确定当前的这个状态的影响，而监督学习在求解过程中会认为当前这个状态对最终结果有确定的影响，所以监督学习无法直接解决MDP问题。

到现在才觉得有一点点入门，第九课的内容实在有点难度，希望日后能够把这些都搞懂。回头看写的这一系列笔记还不够完美，课上有很多精华难以全部摘录下来，写的内容也只是我自己能消化的，很多内容我尽可能的简写，希望看完这些的人能够对强化学习有一些认识就足够了，真正想搞懂的最好能够自己去看看David Silver的课，真的很厉害，课程安排循序渐进，每一课的内容都非常充实。这个系列到这里就结束了，感觉身体被掏空。

[文章目录](#) [站点概览](#)

[1. 1.内容回顾](#)

[2. 2. \$\epsilon_t\$ greedy](#)

[3. 2.Upper Confidence Bounds Algorithm](#)

[4. 3.Information State Search](#)

[5. 4.小结](#)

Learning and Planning(对Environment建立模型)

Disqus 无法加载。如果您是管理员，请参阅[故障排除指南](#)。

[文章目录](#) [站点概览](#)

- [1. 1.内容回顾](#)
- [2. 2. \$\epsilon_t\$ greedy](#)
- [3. 2.Upper Confidence Bounds Algorithm](#)
- [4. 3.Information State Search](#)
- [5. 4.小结](#)