强化学习经典入门书的读书笔记系列--第二篇（下）。

2017-05-03 小吕 神经网络与强化学习

这一篇接之前的第二篇（上），完成整个第二章的内容。

2.5 Tracking a Nonstationary problem

到目前为止为们讨论的sample-average算法针对stationary environment是适用的，然而当我们面对Nonstationary environment时，就需要加以改变。首先说明一下什么叫stationary，什么叫nonstationary。所谓stationary，在这个例子中就是指赌博机的臂是不随着时间改变的，也就是说我们可以认为其真实的value值是保持不变的。相对应的，nonstationary，就是说赌博机的臂会随着时间改变，也就是其真实value值是随时间变化的。尽管在我们的场景想象中认为赌博机怎么会随时间变化，但是这里讨论的是更广泛的强化学习的问题情形所常见的nonstationary现象。

回归到nonstationary话题，在这种情况下，我们需要更加依赖近期的reward信息，而不太依赖很远时间前的reward信息，因为赌博机的臂总是变化，近期的更有参考意义。因此，我们可以把stepsize变成固定值，而非之前的$\frac{1}{k}$。为什么stepsize变成固定值就能起到上述效果呢？看下式：

$$\begin{aligned} Q_{k+1} &= Q_k + \alpha[R_k - Q_k] \\ &= \alpha R_k + (1-\alpha)Q_k \\ &= \alpha R_k + (1-a)[\alpha R_{k-1} + (1-a)Q_{k-1}] \\ &= (1-\alpha)^k Q_1 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} R_i \end{aligned}$$

我们把这种形式叫做weighted average，权重平均，注意看每一个R_i的权重都依赖于它离当前选择的时间长短。有时候，这种形式也被叫做exponential,recency-weighted average。

当stepsize随着时间变化时，比如用$\alpha_k(a)$表征第k次选择动作a时的stepsize，如果要保证随着实验次数的增加，选择某一个stepsize能使最终estimate value收敛于real value，那么stepsize需要满足一点条件。如下：

$$\sum_{k=1}^{\infty} \alpha_k(a) = \infty$$
$$\sum_{k=1}^{\infty} \alpha_k^2(a) < \infty$$

这两个条件需要同时被stepsize满足。第一个条件保证了最终stepsize足够大，可以克服任何初始条件或者随机波动；第二个条件保证了最终stepsize变得足够小可以收敛。

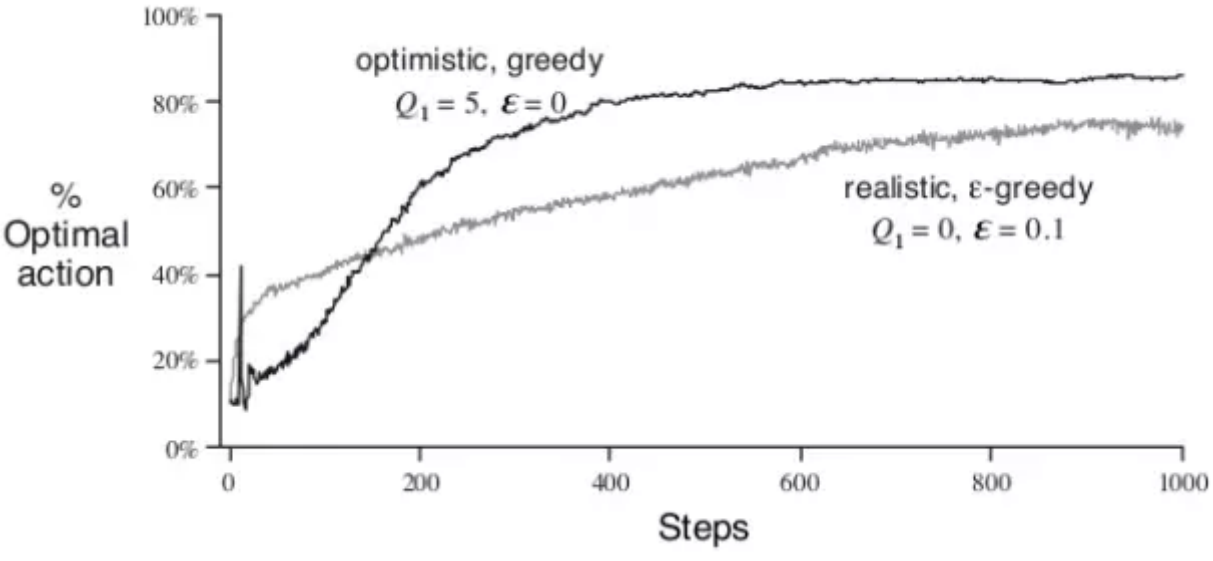
注意到我们在之前的sample-average中使用的stepsize-- ，它同时满足这两个条件，但是后来的这个constant stepsize--- 却不满足第二个条件，这表明了在这种stepsize情况下，estimate value永远不会收敛到real value，并且会随着近期的reward信息而不断变化。但是在nonstationary情形下，这种性质是我们需要的，因为我们每个动作的真实value都在变。另外，满足上述两个条件的stepsize收敛通常都非常慢，并且需要大量的调参，才能使得其收敛效果令人满意。尽管这种条件上的限制性在理论上很重要，在实际应用中其实大家都不怎么在意这些限制条件。

2.6 Optimistic initial value

目前为止我们讨论的所有方法都是不依赖初始值的，用统计科学的话说，就是这些方法都有初始值偏差。对于sample-average算法，当所有的十个赌博机的臂都被选择过至少一次后，这种偏差就不存在了（读者可以想想问什么）。但是对于constant stepsize，这种偏差一直存在，虽然随着时间在降低。事实上，这种偏差通常都不是什么问题，反而有时会有帮助。初始值的缺点在于这些参数需要认为设置，优点在于它可以告诉我们一些关于reward水平的先验知识。

初始值有时也可以被用于促进exploration。比如，我们把之前实验中的estimate value的初始值都设置为5,而不是0。由于real value来源于均值为0,方差为1的正态分布（实际选择时会有一定的噪声），那么相对于real value，estiamate value的初始值就偏大了。这样的话，执行机构就会发现第一个动作选项的reward小于estimate value，于是转而尝试第二个选项，很大概率上执行机构会发现几乎所有选项的reward都小于estimate value，结果就是无形中把这些选项都exploration了一遍，就算是greedy算法也是如此。

对比是否使用这种optimistic intial value，做了一组实验。结果如下：



明显可以看出，使用了optimistic intial value的方法，在一开始由于exploration过多，使得效果不太好，但是在后期远远好于初始值为0的$\varepsilon - greedy$算法。我们可以把这个初始化的技巧看成一个小trick，它在stationary的情形下很不错。但是这个技巧在nonstationary的情形下就不适用了。因为它对exploration的促进是暂时性的。事实上，任何集中在初始情形的方法都不可能在广义上的nonstationary情形下有效。因为任务一旦变化，所有的初始值都不可能始终适用。这也给了我们一个教训证明一开始的sample-average算法是不太合理的。因为它把所有时间线上的reward都给予了相同的权重。然而这并不是把这些方法一棍子打死了，因为它们之间的结合使用在实际情况中往往是合适的。

2.7 Associtive search

这一章主要是提前预热完整的多situation的强化学习问题。

到目前为止，我们讨论的都是单一situation下的任务，然而更常见的是多situation下的任务。学习的目标是一个policy，就是一张映射表，用来关联某个situation下的某个最优的action。现在我们就适当把当前的多臂赌博机的问题扩展到多situation情形下。

比如，我们现在有多个多臂赌博机问题，我们随机决定玩哪一个。这样的话，游戏场景就随机变化。假定当你被随机分配到一个任务时，会有一些另外的线索提供给你，比如想象一个可以随着其action value的改变而改变颜色的赌博机（不同任务下的机器有不同的但是固定的颜色），这时你就能把当前看到的颜色和当前任务的最好的action关联起来（映射），这样下次当你被随机分配到另一个任务时，看到赌博机的颜色，依据之前的映射，就可以选择一个最优的action。这比你在不知道任何区分不同任务的线索的情况下要好的多。

这个例子是单一situation的多臂赌博机问题和真正的强化学习问题之间的过渡。如果我们的每个action还能影响后续的situation的话，那就是真正的强化学习问题了。我们会在后续章节陆续描述。

2.8 结论

我们在这一章里提出了一些简单的方法，用于平衡强化学习过程中的exploitation和exploration。$\varepsilon - greedy$算法在每经过一段时间后随机选择一个action；softmax-greedy算法根据每个action的estimate value来评估其被选中的概率，进行概率分级，而不是像$\varepsilon - greedy$那样概率均衡。我们怎么评价这些简单的方法呢？它们在实际的应用中是最好的方法吗？到目前为止来看，答案是：yes。尽管它们看起来很简单，但是它们的确可被认为是state-of-art的。的确是有一些更复杂的方法，然而受限于这些方法的复杂性和强假设前提，他们在实际应用中并没有太大优势。我们在第五篇会讨论一些部分基于本章思想的方法来解决实际的强化学习问题。之后的章节我们会讨论policy-based算法。

尽管我们现在能得到的方法距离真正能完美平衡exploration和exploitation的方法还很远，我们依然需要总结一下当前的思想，尽管不那么实用，但是对以后更好的方法会有启发性作用。

我们目前主要的思路是使用action value的estimate的uncertainty来控制exploration（这句话很绕）。为了更清楚的说明，举例如下：假设有两个action的estimate value均稍稍小于greedy action，但是它们的uncertainty差异较大。假设action 1不确定性较小，或许是因为这个动作的reward信息足够多以至于其estimate value接近确定，其真实value值出现大幅波动以至于超过其estimate value的可能性微乎其微。action 2的情况正相反。很明显，这种情况下explore action 1是更合理的。

顺着这个思路就引出了所谓“区间估计”的方法。这种方法对每个action的estimate value算出置信区间。置信区间的意思是相比于说这个action的value大约是10,区间估计的方法说这个action的estimate value在9和11之间的可能性是95%。最终被选中的action是有着最大的upper limit的那个，也就是区间上限最大的。在一些情况下，我们可以保证最优的action被选中的概率等于置信因子（95%）。然而，区间估计的方法在实际应用中受限于其统计方法的复杂性。另外，这种方法在统计学上的前提假设在实际案例中大多也不满足。

然而，类似区间估计这种，或者其他用于衡量估计值的不确定性的思路是值得研究的。

还有另外一种著名的算法叫做贝叶斯最优化方法。这种方法计算巨复杂，但是有一些很有效的办法去得到近似值。这个方法有个前提就是我们知道每一组可能的true action value的概率。这样的话，整个试验过程所包含的任何event chain的reward和possibility都可以知道，于是就可以选择最好的那个。但是这种可能性的增长速度是惊人的：就算只有两个action和两个reward，经过1000次动作选择，会产生2^{2000}种可能的event chain。

在多臂赌博机问题中的经典解法是计算一个Gittins indices方程。这个方法给出了某些更广义的多臂赌博机问题的最优解，但是也需要知道一些先验分布知识。不幸的是，无论从计算的可行性还是理论的易理解程度来说，这个方法都不适合推广到接下来我们讨论的广义强化学习问题。

作者的话：至此，第二章的内容基本结束。sutton在这一章给出了一个过渡性的强化学习案例讲解，也提出了一些重要的思想。这些思想虽然简单，但是对后续有效方法的提出有着很重要的作用。