强化学习——多臂赌博机

强化学习区别于其他类型的学习，它使用训练数据不仅能够产生正确的行为指令，并且能够评价该行为。由此产生了显示搜索有利行为的主动探索需求。纯评价是反馈指明了行为值（对于行为收益的评估），而不是单纯的行为好坏性评价。另一方面，纯指示型反馈则指明应该采取的正确行为，独立于实际采取的行动。这种反馈基于监督式学习，他包含了大量的模式匹配、人工神经网络和系统识别。在这种纯粹的形式下，这两种反馈是完全不同的：评价式反馈依赖于所采取的动作，而指示型反馈则独立于实际采取的动作。

在此我们研究增强学习在简化场景下的评价方面，所谓简化场景就是说它不涉及多个场景的学习，这种非关联场景已经有许多相关工作涉及到评价式反馈，而且它避免了复杂的完全增强学习问题。学习这些例子有助于我们理解评价式反馈以及与之相区别的指示型反馈。

我们将探索的这种特殊的非关联性评价式反馈问题是 k-armed bandit problem 的简化版本。我们用多臂赌博机问题来引出后续章节中药介绍的完全增强学习的基本方法。

多臂赌博机问题介绍

考虑以下问题:你重复面对着 K 个不同的选项或者说是动作，在每个动作选择之后，你会从固定的概率分布中得到相应得动作数值奖励。你的目标是在一段时间内最大化期望总奖赏。比如，超过 1000 次动作选择或者时间步。这就是最初的多臂赌博机问题。

在 K-臂赌博机问题中，每个动作的选择都有一个期望奖赏或者平均奖赏。我们用 A_t 表示在时间 t 采取的动作.与之对应的奖赏表示为 Rt.而任意一个动作 a 的值表示为 $q_*(a)$，$q_*(a)$是选择 a 的期望奖赏。

$$q_*(a)=\mathrm{E}[R_t|A_t=a]$$

在此，我们假设不知道每个动作的真实值，虽然你可以估计其值，我们用 $Q_t(a)$表示在时间 t 下采取动作 a 的值，$Q_t(a)$是一个实验值，服从一定的概率分布。我们需要看到的效果是 $Q_t(a)$ 收敛于 $q_*(a)$，即 $Q_t(a)\approx q_*(a)$。

探索（exploration） VS 利用（exploitation）

贪婪动作（greedy action）：在已有的动作估计值中，在任意一个时间不长中必定至少存在一个最佳动作，这个最佳动作具有最大的估计值，选择估计值最大的动作便是贪婪动作。

利用(exploit)：根据当前的动作值集合，选择贪婪动作便是利用。

探索（explore）：选择非贪婪动作。

利用在具体的时间不长中会获得一个最大的期望奖赏值（一个动作的奖赏值最大），但探索在长期的运作中或许能产生更好的总奖赏。所以强化学习的问题之一就是平衡探索和利用以获取最大的总奖赏。

行为值估计法

对行为值的估计是为了更好的选择行为。行为的值为每次执行该行为所得奖励的期望。因此可以用 t 时刻前行为已得到的奖励作为行为值的估计，即：

$$Q_t(a) = \frac{t\text{时刻前}a\text{行为的奖励之和}}{t\text{时刻前}a\text{行为出现的次数}} = \frac{\sum_{i=1}^{t-1} R_i \cdot 1_{A_i=a}}{\sum_{i=1}^{t-1} 1_{A_i=a}}$$

上式中如果分母为 0 ，则 $Q_t(a)$为 0。当试验次数接近无穷时，$Q_t(a)$将收敛于 $q_*(a)$，这种方法称为**样本平均(sample-average)法**，当然，这只是估计行为值得一种方法，但并不是最好的一种。并在 t 时刻选择行为时，使用贪心策略选择行为值最大的行为，即

$$A_t = \arg\max Q_t(a)$$

贪心算法：上式描述了贪心算法，即每次选择具有最大值的行为。这种方法具有一定的缺陷，即只有利用并不探索，在某种程度上来说，贪心算法可以得到立即奖赏最大值，但从长远角度来看，它并不一定能得到最大的奖赏总值。由此产生了ε-greedy 方法:在大多数情况下选择贪婪动作，而在ε概率下从所有的动作中等概率选择（所有包括贪心动作也包括非贪心动作，或许有的资料上除去了贪心动作，但在笔者的参考资料上是所有的动作）。ε-greedy 方法的一个优势在于它能在有限的时间内更行行为的估计值，已达到 $Q_t(a)$收敛于 $q_*(a)$的效果。

在ε-greedy 方法中：

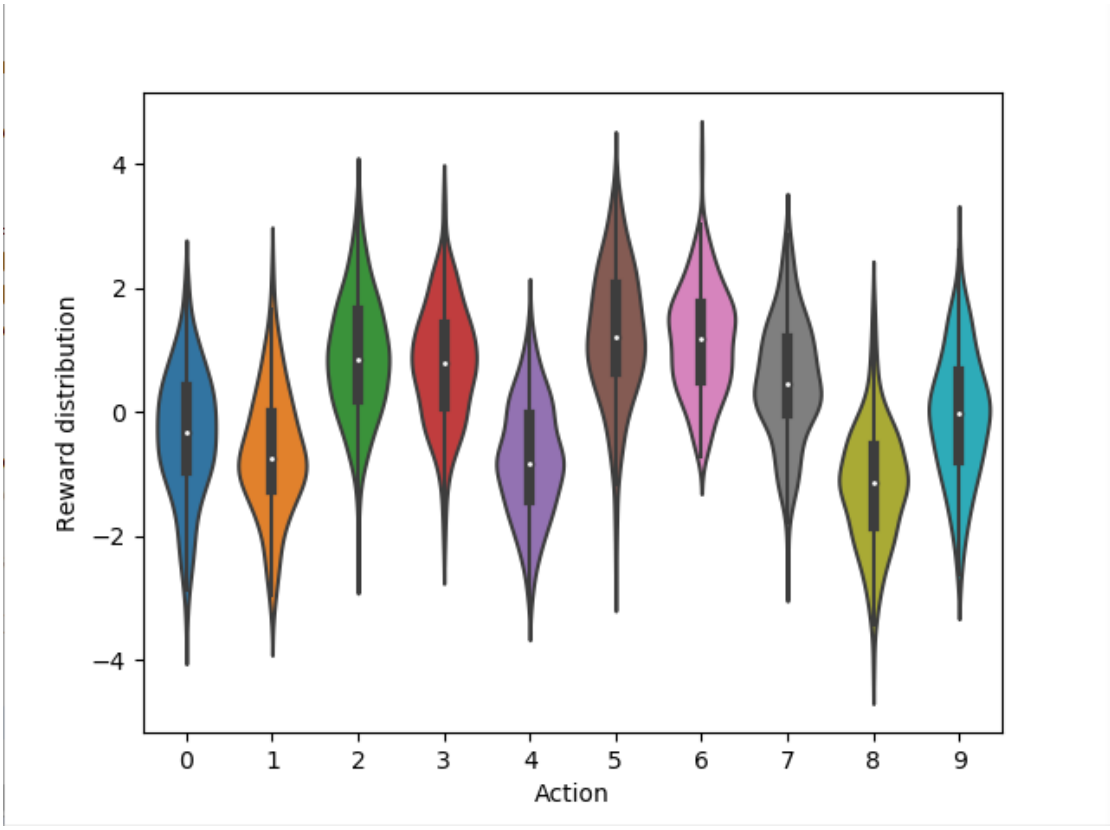
- 贪心动作概率：$A_t = 1 - \varepsilon + \frac{\varepsilon}{|A|}$
- 其他动作概率：$\frac{\varepsilon}{|A|}$

(公式推导自证)

十臂测试平台（实例）

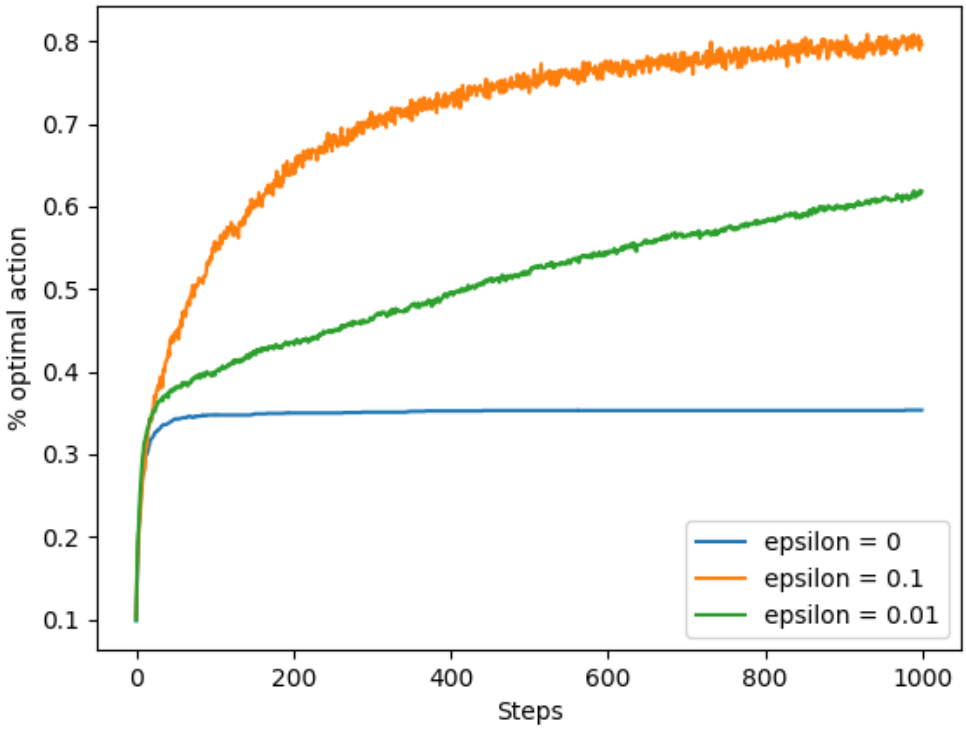
为了系统的评估贪心算法和ε贪心算法的有效性。我们做了一个恰当的实验:
设定 k-armed bandit 的 k=10，并且随机运行 2000 次。

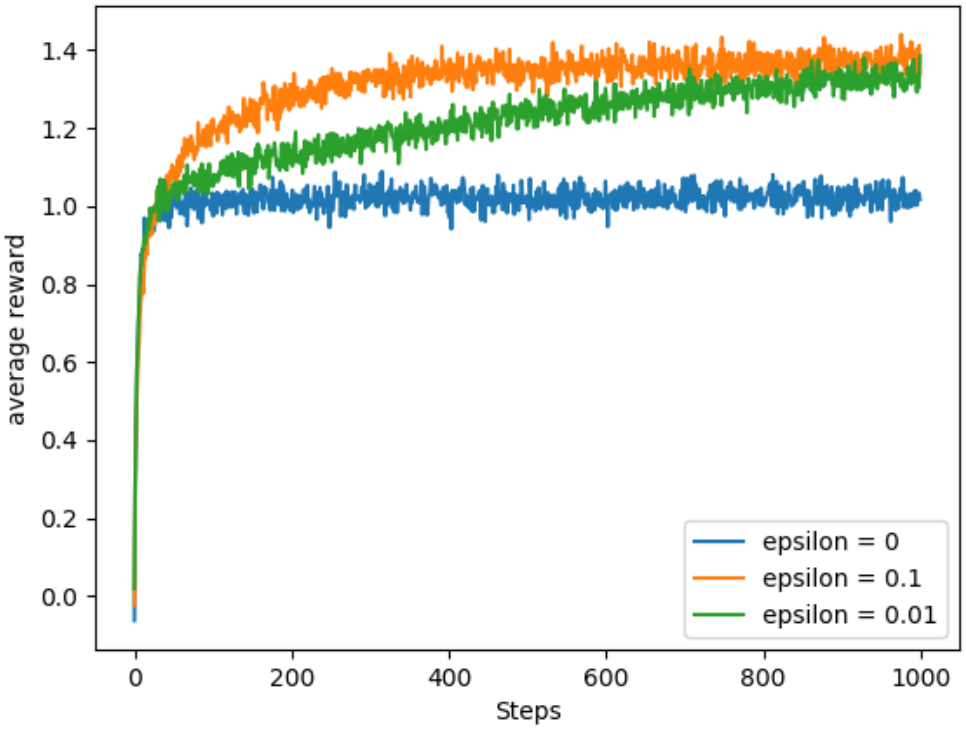
在每个赌博机问题中，动作值（action value）服从期望为 0，方差为 1 的高斯分布。当一个具体的学习方法应用于在时间步 t 选择动作问题上时，实际奖赏 R_t 服从期望 $q_*(A_t)$方差 1 的正态分布。如图 2.1 所示，我们把这种实验称为十臂测试平台。



(图 2.1)

在以上数据的基础上，在十臂赌博机基础上考虑一下三种情况来做一个实验：贪心算法、ε-贪心算法（ε=0.1）和ε-贪心算法（ε-0.01）其结果如下：





由图我们可以得到结论表示：短期内，贪婪算法显然更占优势，但从长远来看，适当的探索对我们更有利。

增量实现

目前我们所讨论的动作值方法中，所有估计的动作值都是所观察到的样本平均值。现在我们把问题焦点转为怎样使用更有效的计算方法计算出这些平均值，特别是利用连续记忆和持续时间步长计算。

为了简化我们所关注的单个动作，我么用 R_i 表示第 i 个动作选择之后得到的奖赏值。用 Q_n 表示选择 n-1 次的动作值。简化如下：

$$Q_n = \frac{R_1 + R_2 + R_3 + \cdots + R_n}{n-1}$$

显然，通过上式我们可以计算并记录任意时刻的奖赏估计值。但长期运作下，这对于计算机内存以及计算性能要求很高，并不明智。上式可化解如下：

$$\begin{aligned} Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\ &= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\ &= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\ &= \frac{1}{n}\left(R_n + (n-1)Q_n\right) \\ &= Q_n + \frac{1}{n}\left(R_n - Q_n\right) \end{aligned} \tag{2.3}$$

这样就只需要记录Q_n和 n 即可得到 Q_{n+1} 。扩展到一般形式如下：

NewEstimate ←OldEstimate + StepSize [Target – OldEstimate]

Target-OldEstimate：估计差
Target：前进的理想方向

追踪不稳定问题

上述讨论的平均法适用于稳定的赌博机问题，而赌博机问题的奖赏分布不会随时间的变化而变化。但强化学习问题又通常是不稳定的，在这种情况下，最近的奖赏相比于前面的奖赏通常占有更重的比值。解决这类问题的通常方法是固定步长参数，令α为步长因子，$\alpha = \frac{1}{n}$ 。则 2.3 式变为：

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n) \qquad \alpha \in (0, 1]$$

为了推导出步长因子与奖赏之间的权重关系，做如下变化：

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n)$$

$$= \alpha R_n + (1-\alpha) Q_n$$

$$= \alpha R_n + (1-\alpha) [\ \alpha R_{n-1} + (1-\alpha) Q_{n-1}\ ]$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1} + (1-\alpha)^2 Q_{n-1}$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1} + \alpha(1-\alpha)^2 R_{n-2} + \cdots + \alpha (1-\alpha)^{n-1} R_1 + (1-\alpha)^n Q_1$$

$$= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

步长因子在每一步中都会有变化，我们用$\alpha_n(a) = \frac{1}{n}$来表示步长因子，在 n 足够大的时候，能保证上式收敛于真实值。但这种收敛在所有的$\{\alpha_n(a)\}$序列中并不足以得到保证。随机逼近理论的一个知名结果给了我们保证上式收敛的条件：

$$\begin{cases} \sum_{n=1}^{\infty} \alpha_n(a) = \infty & \text{(不收敛)} \\ \sum_{n=1}^{\infty} \alpha_n(a)^2 < \infty & \text{(收敛)} \end{cases}$$

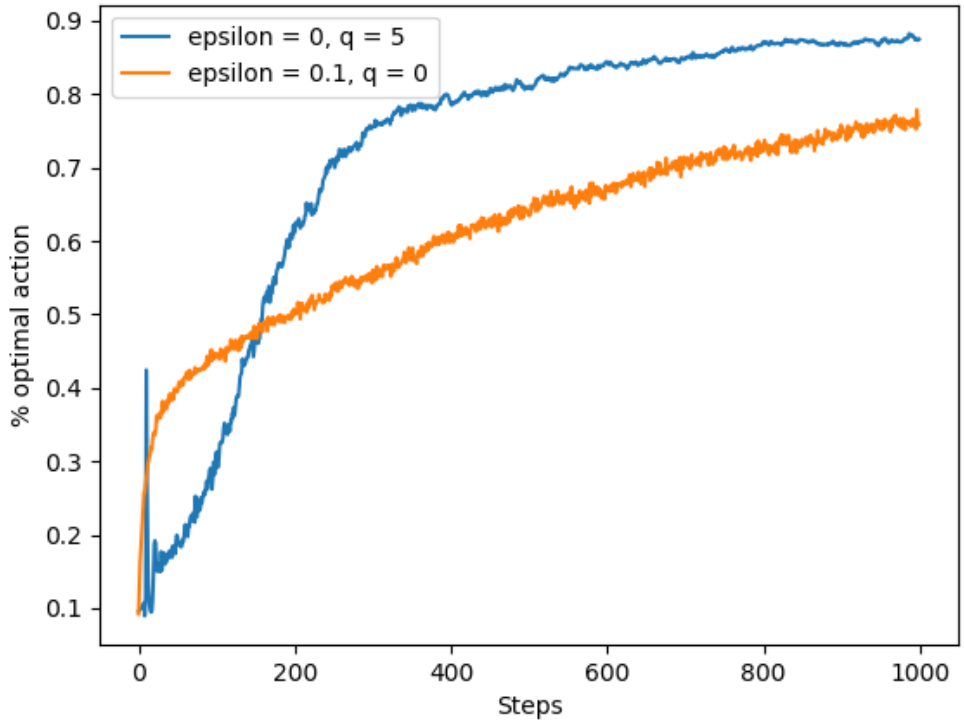
第一个条件用于保证不长足够大最终克服任何初始条件或者随机的起伏。第二个条件保证了最终的步数变得足够小保证收敛。

注意在样本平均值的情况下两个收敛条件都要成立，即 $\alpha_n(a) = \frac{1}{n}$ 。但对于固定步长参数的情况下却不是同时满足两个收敛条件，即 $\alpha_n(a) = \alpha$。后者不满足第二个收敛条件这表明了估计值不会完全收敛于真实值但会不断变化影响最近的奖赏。正如我们刚所提到的，这种做法在不稳定的环境中是可信赖的，并且此类问题在强化学习中不稳定问题是很常见。因此，一系列的步长参数在上述两个收敛条件下往往会收敛得比较慢或者需要适当的调整以达到一个满意的收敛率。

优化初始值

目前我们讨论过的所有方法都在某种程度上依赖于初始动作值的估计，$Q_1(a)$。在统计学中，这些方法在初值是估计的情况下结果是有偏差的。在样本平均方法中，只有在所有的动作都至少选择了一次之后这种偏差才会消失。但在使用到常量α的方法中，虽然这种偏差在时间步长增加的过程中会减弱，但这种偏差永恒存在。在实践中，这种偏差通常不是一个考虑问题甚至在某些时候这种偏差会很有帮助。坏处在于初始参数必须是由用户提供的一系列值集合，不然只能都初始化为 0，而好处是它以一种简单的方式提供了一些奖赏可期望程度的先

验知识。



初始动作值也可以作为一个简单的鼓励探索方法。在下图中，优化方法在最开始的时候因为探索更多而表现的不是很好，但随着探索的减少最终它表现的更好。我们称这种鼓励探索的方法为优化初始值。但这种方式并不适用于不稳定问题。因为在不稳定情景中，情景（task）一旦变化，又会重新探索，优化初始值并不会所帮助。

置信上界选择

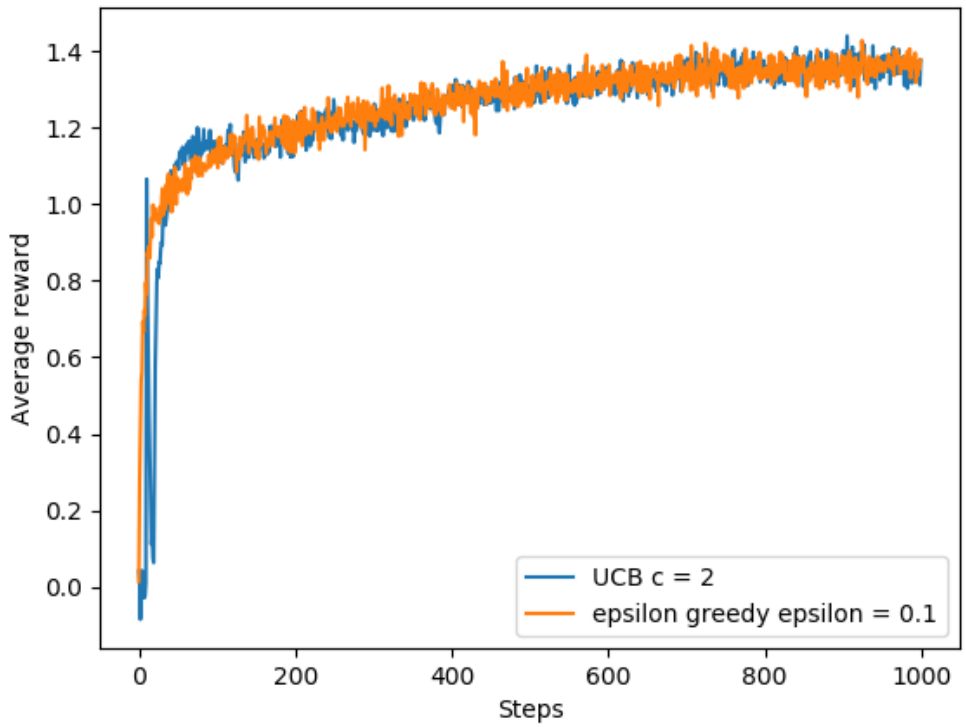
我们总是需要探索，因为动作值估计准确度的不确定性。贪心动作在目前的情况下看起来是最好的，但或许实际上一些其他的动作会更好。ε-greedy 贪心动作选择面临着非贪心动作的选取，但这种选取很弱智，它只会等概率的选择所有动作，而不会根据动作的潜力值（成为最优动作的潜力）来选取。一种有效的方式便是通过动作的潜力来选择动作：

$$A_t \doteq \text{argmax}\left[\, Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}\,\right]$$

$N_t(a)$:行为 a 在时间 t 之前已经被选择的次数
$\ln t$:时间 t 求对数，e≈2.71828
C :C>0,对探索力度的控制参数

置信上界选择核心思想是平方根是衡量动作值估计的准确度或方差。因此，数量上最大化可能是动作 a 值的真实值，而 c 决定了置信指数。一方面，每次选择 a 动作都会使得不确定性减少：$N_t(a)$增加，分母增大，总函数值减少。另一方面，每次选择 a 以外的动作，时间 t 增

大，$N_t(a)$不会增大，不确定性则增大。而使用自然对数的原因是为了随时间的增加让增长率变小一点，但不会有上界。 所有的动作都会被选择到，但是针对估计值低的动作，或者是被选择平凡的动作，会降低其选择频率。



梯度赌博算法

目前我们所讨论的方法都是估计动作值并且用这些估计的数值来选择动作。这确实是一种好的方法，但并不是最佳的方法。现在我们考虑学习对每个动作 a 的数值偏好，用 $H_t(a)$表示，qi 值越大，动作被选择的机会越大。但这种偏好在奖赏方面并不会有什么解释，不同动作之间只有相对偏好重要。如果我们把动作偏好增加 1000，那么对于选择动作的概率并没有什么影响，它是由 soft-max 分布决定：

$$\Pr\{A_t = a\} = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} = \pi_t(a)$$

$\pi_t(a)$:在时间 t 选择动作 a 的概率
$H_t(a)$:对动作 a 的偏好程度

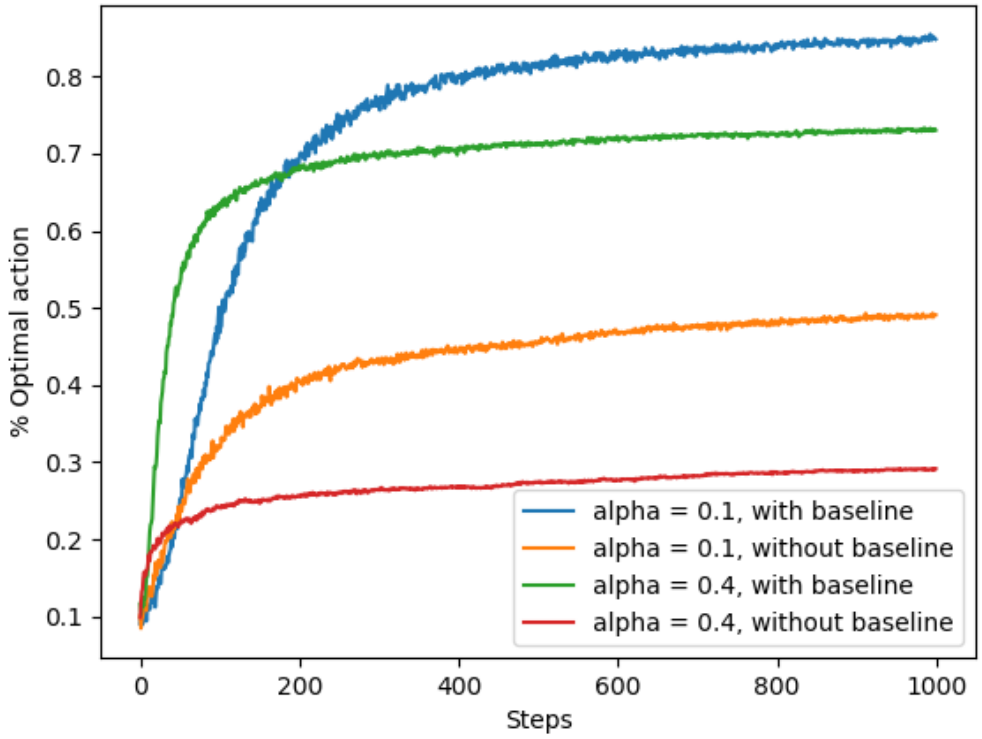
基于随机梯度上升的思想，有一个自然学习算法，在每一步中，选择动作A_t之后会收到一个奖赏R_t，通过如下函数更新动作偏好：

$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \overline{R_t})(1-\pi_t(A_t))$ a = A_t

$H_{t+1}(\alpha) = H_t(A_t) - \alpha(R_t - \overline{R_t})\pi_t(\alpha)$ for all a $\neq A_t$

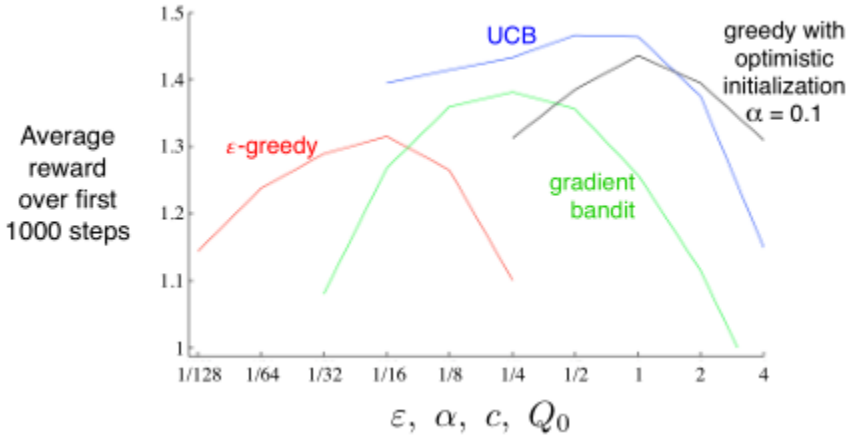
α：$\alpha > 0$,是一个步长参数，

$\overline{R_t}$：$\overline{R_t} \in R$，是时间 t 范围内动作的平均值，相当于一个基准线。



总结

这张我们介绍了几种简单的平衡探索和利用的方法，ε-greedy 方法以ε概率随机选择动作，UCB 方法选择确定动作但同时 UCB 又巧妙地设计了在每一步倾向于选择那些收到样本更少的动作进行探索。梯度赌博算法估计的不是动作值，而是动作偏好，更加倾向于选择统一分级的动作，而其分布概率方法则采用了 soft-max 概率分布方法。乐观初始值方法比贪心算法更加显著的进行探索。但很多人会产生一个疑问：这些方法中那个方法最好。我们在此通过 parameter study 图来给出解答：



在方法的选择中我们不仅应该评估其在最优参数下的表现，还要看其对参数的敏感度。所有的这些算法对参数都不敏感，参数在一个数量级内变化时这些算法表现都很好，但总的来说，UCB 表现最佳。

尽管本章提出的这些简单的方法可能是我们目前所能做到的最好的方法，但是他们离满意地解决探索与利用问题还有非常大的距离。目前在平衡和探索方面做的比较好的是贝叶斯方法，在贝叶斯方法中，我们甚至可以计算出探索与利用之间的平衡。感兴趣的读者可以自行研究。