lec05 ppt版

这个版本的符号与详细版的不同，详细版是根据 Reinforcement Learning 2021 版记录的，ppt 版本则是根据课程 ppt。想要深入了解多臂赌博机问题可以参考详细版，但是对于考试来说这个版本足够了（大概）

多臂赌博机问题

符号：

- a_t：一次动作（拉一次摇臂）
- r_t：动作 a_t 的 reward；r_t 的分布仅仅取决于 a_t
- $E\{r_t \mid a_t\} = Q^*(a_t)$：动作 a_t 的 reward r_t 的期望是 $Q^*(a_t)$
- $Q_t(a) \approx Q^*(a)$：action value 估计，所有动作 a 回报的平均
- a_t^*：贪婪行动
    - $a_t^* = \underset{a}{\operatorname{argmax}}\, Q_t(a)$：能使得 $Q_t(a)$ 最大的 a 就是 a_t^*
    - $a_t = a_t^* \Rightarrow$ Exploitation
    - $a_t \neq a_t^* \Rightarrow$ Exploration

Action-Value 方法

$$Q_t(a) = \frac{r_1 + \ldots, +r_{k_a}}{k_a} = \frac{1}{k_a}\sum_{i=1}^{k_a} r_i \qquad \text{(Sample Average)}$$

$$\lim_{k_a \to \infty} Q_t(a) = Q^*(a)$$

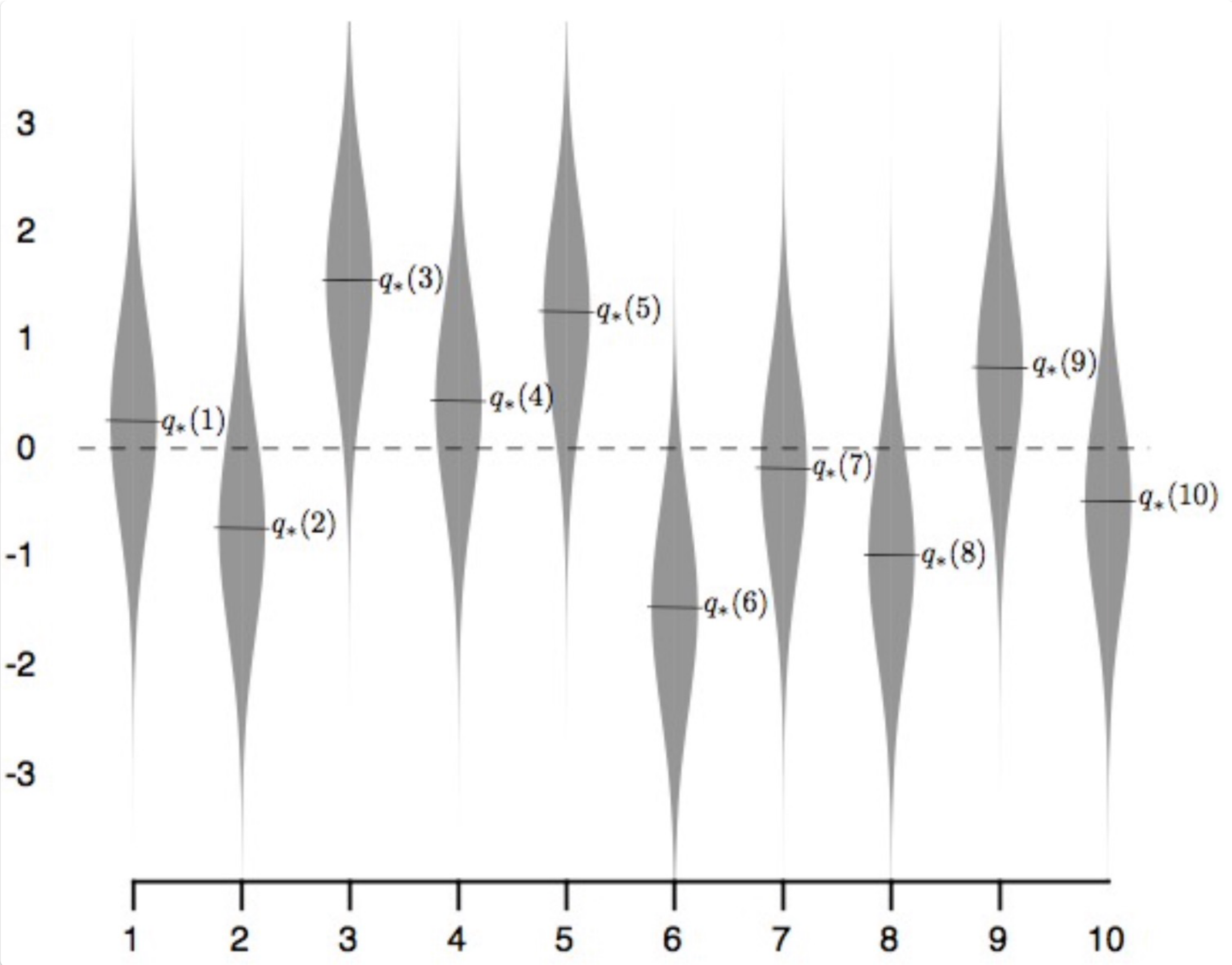
其中，k_a 表示次数
值得注意的是，这是动作 a 的次数，不是所有动作的次数；即，动作 $b, c, \ldots$ 需要另外计算

ε-greedy action 的选择方法

- greedy
    - $a_t = a_t^* = \operatorname{argmax}_a Q_t(a)$
- ε-greedy：
    - $$a_t = \begin{cases} a_t^*, & 以 1-\epsilon 的概率 \\ 随机, & 以\ \epsilon 的概率 \end{cases}$$
- softmax：
    - $$\frac{\exp\left(\frac{Q_t(a)}{\tau}\right)}{\sum_{b=1}^{n} \exp\left(\frac{Q_t(b)}{\tau}\right)}$$

    其中，τ 是 computational temperature 计算温度

10-Armed Testbed



增量方法

我们讨论的是同一个动作，因此省略 k_a 下特指的动作 a
对于 sample average estimation 方法

$$Q_{k+1} = \frac{r_1 + \ldots, r_k}{k}$$

我们可以不存储所有奖励，而是逐步地进行更新（显然这种方法计算更快）

$$Q_{k+1} = Q_k + \frac{1}{k+1}[r_{k+1} - Q_k]$$

即

$$新估计 = 老估计 + 步长\,(目标 - 老估计)$$

Nonstationary Case 非平稳情况

在平稳问题中，选择 Q_k 作为 sample average 非常合适。但是在非平稳问题中，$Q^*(a)$ 会随时间变化，这时再进行和平稳问题中一样的选择就不合适了

我们可以

$$Q_{k+1} = Q_k + \alpha[r_{k+1} - Q_k]$$

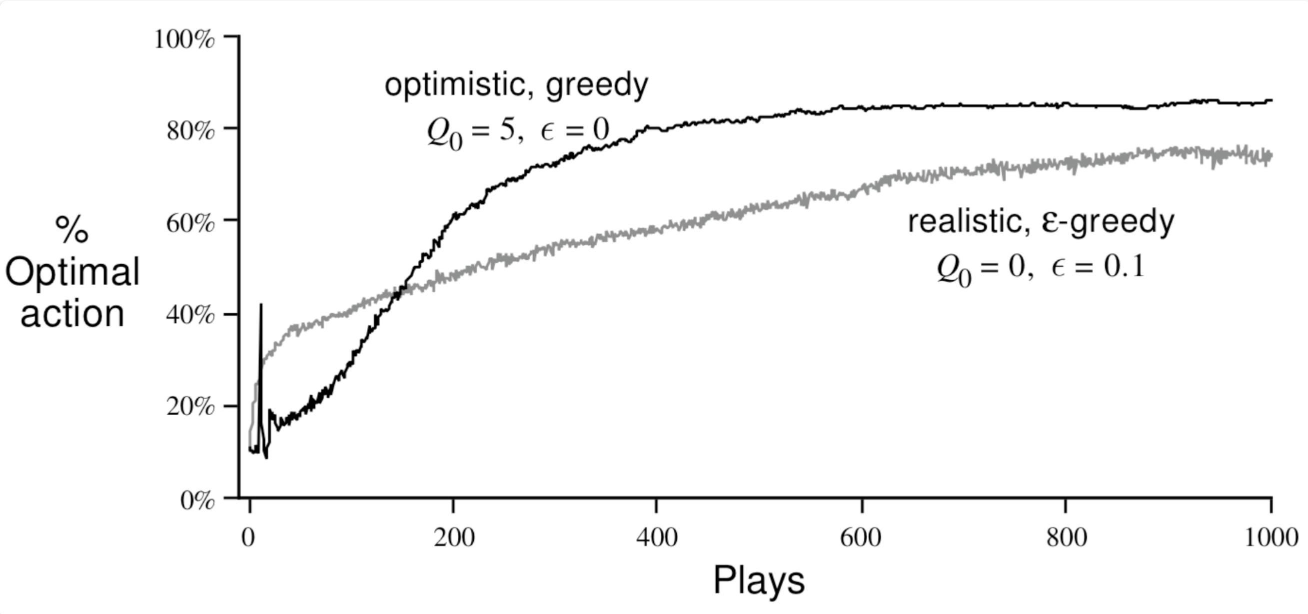
其中，α 是一个 $(0, 1]$ 区间内的常数
而其等价于

$$Q_{k+1} = \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} r_i$$

这是一个指数型的、具有近期加权平均的公式

Optimistic Initial Values 乐观的初始值

当前的所有方法都依赖于 $Q_0(a)$，即初始的动作值估计，这些方法是有偏的（biased）。我们将动作值乐观地初始化，例如在10臂赌博机测试中，将所有动作的初始值 $Q_0(a)$ 设为5（相对于之前设为0）



- 乐观初始化（黑色曲线）在早期阶段迅速提高了选择最优动作的比例
  - 通过将初始动作值设为较高值，算法会在早期阶段进行更多的探索，从而更快地找到最优动作
  - 这种方法可以克服早期对次优动作的过度依赖，促进更有效的学习
  - 在乐观初始值下，即使使用贪婪策略（不进行额外探索），也能获得较好的效果，因为初始的高值促使算法尝试更多的动作
- 相比之下，现实初始化（灰色曲线）虽然逐步提高了选择最优动作的比例，但速度较慢