强化学习笔记

党所贵

2017 年 9 月 24 日

目录

第一部分 Multi-arm bandits

K-摇臂赌博机在选择一个动作后，有一定的概率获得奖励，但是这个概率是我们不知道的，设在时间 t 采取的动作为 A_t，获得的奖励为 R_t，使用 $p_*(a)$ 表示动作 a 预计的奖励则有

$$p_*(a) = \boldsymbol{E}[R_t|A_t = a]$$

假设我们得到动作 a 的奖励的估计，我们每次采取奖励最大的动作。我们记在时间 t，动作 a 的奖励为 $Q_t(a) \approx q_*(a)$，每次采取奖励最大的动作作为当前动作，即使用贪心算法。在试探的过程，有两种策略，一种称之为探索，每个动作均有机会被试探，能够获得每个动作的奖励的估计，一种为利用，即，每次采取奖励最高的动作，能够较为迅速的找到最优的动作策略。

1 Action-Value Method

我们需要获得每一个动作的奖励的估计，一种简单的方法就是使用平均数：

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbf{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbf{1}_{A_i=a}}$$

为了平衡探索和利用，我们在贪心算法中以概率 ϵ 随机选择动作，最终的决策为：

$$A_t = \begin{cases} \arg\max_a(Q_t(a)) & \text{possibility is } 1-\epsilon \\ random & \text{possibility is } \epsilon \end{cases}$$

算法见 Algorithm 1：

Upper-Confidence-bound Action Selection 使用上面的方法可能会导动作的选择局限在很小的范围之内，我们通过改变优化函数来避免这个问题的发生

$$A_t = \arg\max_a(Q_t(a) + c\sqrt{\frac{logt}{N_t a}})$$

Algorithm 1: $\epsilon - greedy$

Input: 摇臂数 K;

Input: 奖赏函数 R;

Input: 尝试次数 T;

Input: 探索概率 ϵ ;

Output: 累计奖励 r

$r = 0$;

$\forall a = 1, 2, \cdots, K : Q_0(a) = 0, count(a) = 0$;

For$t = 1; t \leq T; t++$ **if** *rand()*$\leq \epsilon$ **then**

  $k =$random in$1, 2, \cdots, T$;

else

  $k = \arg\max_a Q(a)$;

$v = R(k)$;

$r = r + v$;

$Q(k) = \frac{Q(k) \cdot count(k) + v}{count(k) + 1}$;

$count(k)++$;

2 Gradient Bandit Algorithm

我们可以考虑给每一个动作 a 学习一个值 $H_t(a)$，反应选择动作的时候对 a 的偏爱程度。使用 soft-max 分布：

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

使用梯度上升法，更新规则为：

$$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)(\mathbf{1}_{a=A_t} - \pi_t(a)), \quad \forall a$$