Exploration *and* Exploitation

老虎机(bandit)任务中的探索利用问题

问题定义

强化学习中有一个经典的问题叫Mult-armed bandit问题，即我们常说的多臂老虎机。假设我们有一个K臂老虎机，每一个臂(action)的回报(reward_i)都是固定的，但是agent并不知道这个回报率是多少，agent如何在T回合内最大化自己的回报(注意，这里的T通常是远远大于K的)。

MAB问题可以看做是一个真实分布$B = \{R_1, \ldots, R_K\}$的集合，这里的分布R_i分别为$K \in \mathbb{N}^+$个臂的reward distribution。$\mu_1, \ldots, \mu_K$是reward distribution的均值，每一轮赌徒选择某一个臂并观测获得的reward。H 是剩下的要玩的轮数，bandit的问题相当于one-state MDP。

- Starting in state $s \sim d(s)$
- Terminating after one time-step with reward $r = \mathcal{R}_{s,a}$

常用方法

random

纯随机的方法，$U(1, K)$

naive

就是先尝试一定次数，然后当每个action都能得到一个均值以后就开始一直地选均值最大的那个action，也就是探索有限次数就一直greedy。

ϵ-greedy

以这个ε概率去随机选动作，以1-ε的概率去执行当前为止已知的最好的动作。 即DQN等中常用的方法。

Optimism-based exploration : Upper confidence bounds(UCB)

Optimism的核心思想就是"对于未知事物保持乐观"，它们认为没见过的就是好的，因为不熟悉的区域可能蕴含着巨大的开发潜力（当然也冒着被打脸的风险）。

由于ϵ-greedy的算法在随机探索的时候仍然是按照均匀分布随机采样的，但是事实上我们有可能已经有足够多的采样证明了action a比action b产生的收益期望更大。这种时候我们应该更多的采取行动a而不是均匀随机了。虽然在DRL中我们采取逐渐decay这个ϵ 来解决这个问题，但是在针对不同的任务中，decay schedule是需要不同定义的，这并不是一个通用的方法。而UCB恰好可以解决这个问题。

UCB的主要思想就是我们要增加对自己把握不准动作的探索而相应的减小对自己已经熟悉的动作的探索，继而提高探索的效率。具体做法：当我们对某个动作采样无穷多次的时候，其样本分布的期望接近于真实分布的期望，但是对于某一个动作来说，其采样不可能有无穷多次，因此估计出的价值$\hat{Q}(a)$ 和真实的价值$Q(a)$ 总有一个差值$U_t(a)$ ，即 $\hat{Q}(a)-U_t(a) \leq Q(a) \leq \hat{Q}(a)+U_t(a)$ 。我们每次估计reward为$\hat{Q}(a)+U_t(a)$ 。而这里的$U_t(a)$ 是被不同动作的采样次数影响的，对于被选的action来说，每多一次采样就会使其差值$U_t(a)$ 更小；对于没被选的action来说，每多一次采样就会使其差值$U_t(a)$ 变大。

对于分布的有界性，有一个定理Hoeffding's Inequality:

Let $X_1, \ldots, X_t$ be i.i.d. (independent and identically distributed) random variables and they are all bounded by the interval [0, 1]. The sample mean is $\overline{X}_t = \frac{1}{t}\sum_{\tau=1}^{t} X_\tau$. Then for $u > 0$, we have $\mathbb{P}\left[\mathbb{E}[X] > \overline{X}_t + u\right] \leq e^{-2tu^2}$

因此真实分布的期望大于样本期望加差值的概率是充分小的，也侧面验证了我们刚才的说法。在这里我们不妨令$e^{-2tU_t(a)^2} = p$ ，那么我们就可以得到$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$ 。由于我们刚才提到每次采样不同动作会影响到$U_t(a)$，因此我们如果令$p = t^{-4}$ ，那么就可以得到**UCB1**算法：

$U_t(a) = \sqrt{\frac{2\log t}{N_t(a)}}$ ，并且$a_t^{UCB1} = \arg\max_{a\in\mathcal{A}} Q(a) + \sqrt{\frac{2\log t}{N_t(a)}}$

在**UCB1**算法中，由于我们没有对reward函数做任何假设，因此我们必须依靠Hoeffding's Inequality去得到比较general的评价，如果我们能够提前知道reward的先验（比如是高斯分布的），那么我们可以根据置信度为95%时候设置$\hat{U}_t(a)$ 为2倍的std。（概率论中的内容）

Posterior matching exploration: Thompson sampling/posterior sampling

在每个time step上，我们都想要根据某个动作a 最佳的概率去选择动作a，即

这里$\pi(a|h_t)$ 是在given历史信息h的情况下挑选action a 的概率。

对于常规的老虎机来说，拍一次所获得的奖励无非就是两种，要么是1，要么是0，这也是我们所说的Bernoulli bandit。那么基于这个性质，我们很自然的可以假设$Q(a)$服从Beta分布。Beta分布有两个参数(α，β)，分别代表着输赢。那么算法开始的时候我们需要将这两个参数进行初始化，这里边我们可以加入些个人的经验在里边。举个栗子，如果我们初始化时候令$\alpha = \beta = 1$，那么就意味着我们认为胜负概率均为50%，单我们其实也不是很确定；如果我们取$\alpha = 1000$，$\beta = 9000$，那么就是说我们还是比较确定这个奖励概率在10%。总体流程就是我们从先验的$Beta(\alpha_i, \beta_i)$ 中去sample出来reward $\tilde{Q}(a)$ ，然后根据$a_t^{TS} = \arg\max_{a\in\mathcal{A}} \tilde{Q}(a)$ 进行选择action，得到真正的reward，来更新我们认知中的收益分布参数α，β：$\alpha_i \leftarrow \alpha_i + r_t 1\left[a_t^{TS} = a_i\right]$ $\beta_i \leftarrow \beta_i + (1-r_t)\,1\left[a_t^{TS} = a_i\right]$

在实际应用中，可能难以用贝叶斯推断的方法去估计后验分布，在这种情况下，我们可以采用 Gibbs sampling, Laplace approximate和bootstraps等方式近似后验分布，来使用Thompson sampling。详细的关于TS的内容可以参照 https://arxiv.org/abs/1707.02038v2"Tutorial"

参考文献