强化学习

Reinforcement learning

第9节　试探/探索与开发/利用

Exploration & Exploitation

张世周

Outlines

- 1.1 介绍
- 1.2 多臂赌博机
- 1.3 Contextual Bandits
- 1.4 MDPs

探索和利用的困局

探索和利用的困局：

- 利用是做出当前信息下的最佳决定；
- 探索则是尝试不同的行为继而收集更多的信息。
- 最好的长期战略通常包含一些牺牲短期利益举措。通过搜集更多或者说足够多的信息使得个体能够达到宏观上的最佳策略。因此探索和利用是一对矛盾。

举例

饭店选择

利用：去你最喜欢的饭店

探索：去一家新饭店

线上广告：

利用：选择最受欢迎的广告

探索：选择不同的广告

石油钻探：

利用：选择最出名的地点

探索：选择一个新地点

玩游戏：

利用：选择你认为最好的动作

探索：随机选一个动作

几个基本的探索方法

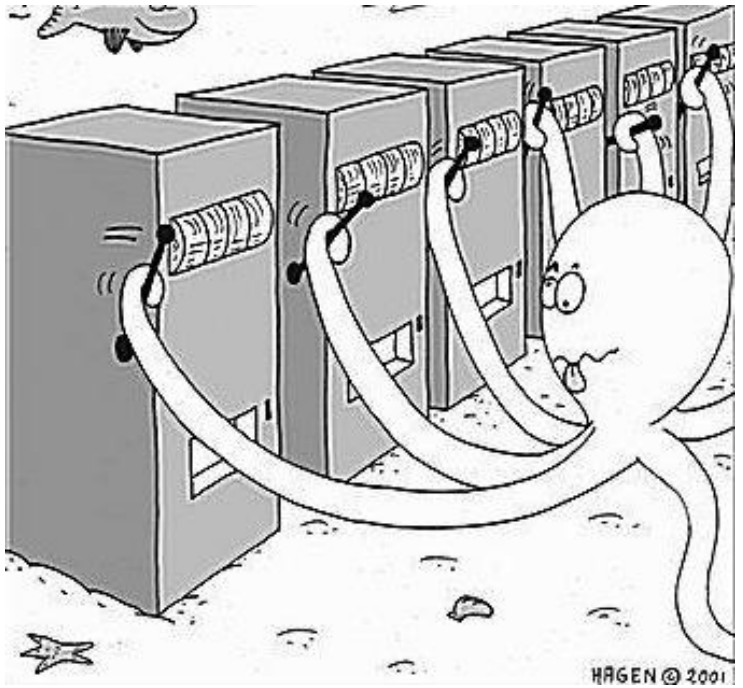
- 朴素探索(Naive Exploration): 在贪婪搜索的基础上增加一个ε以实现朴素探索；
- 乐观初始估计(Optimistic Initialization): 优先选择当前被认为是最高价值的行为，除非新信息的获取推翻了该行为具有最高价值这一认知；
- 不确定优先(Optimism in the Face of Uncertainty): 优先尝试不确定价值的行为；
- 概率匹配（Probability Matching): 根据当前估计的概率分布采样行为；
- 信息状态搜索(Information State Search): 将已探索的信息作为状态的一部分联合智能体的状态组成新的状态，以新状态为基础进行前向探索。

几个基本的探索方法

- 依据状态-行为空间的探索：针对每一个当前的状态，以一定的方法尝试之前该状态下没有尝试过的行为。
- 参数化探索：针对策略的函数近似，此时策略用各种形式的参数表达，探索即表现为尝试不同的参数设置。优点在于，得到基于某一策略的一段持续性行为；缺点是对智能体曾经到过的状态空间毫无记忆，也就是智能体也许会进入一个之前曾经进入过的状态而并不知道其曾到过该状态，不能利用“已经到过这个状态”的信息。

多臂赌博机

- 一个多臂赌博机是一个元组<A,R>
- A是已知的m个动作
- $\mathcal{R}^a(r) = \mathbb{P}[r|a]$ 是一个对奖励的未知概率分布
- 在每个步骤t中，智能体都会选择一个动作 $a_t \in \mathcal{A}$
- 环境会生成一个奖励 $r_t \sim \mathcal{R}^{a_t}$
- 目标是使累积奖励最大化 $\sum_{\tau=1}^{t} r_\tau$



后悔值

- 动作价值是对行动a的期望奖励，$Q(a) = \mathbb{E}[r|a]$
- 最优动作价值V^*为 $V^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a)$
- 后悔值为每一步的机会损失：$l_t = \mathbb{E}[V^* - Q(a_t)]$
- 总后悔值为总机会损失：$L_t = \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right]$
- 最大化累积奖励=最小化总后悔值

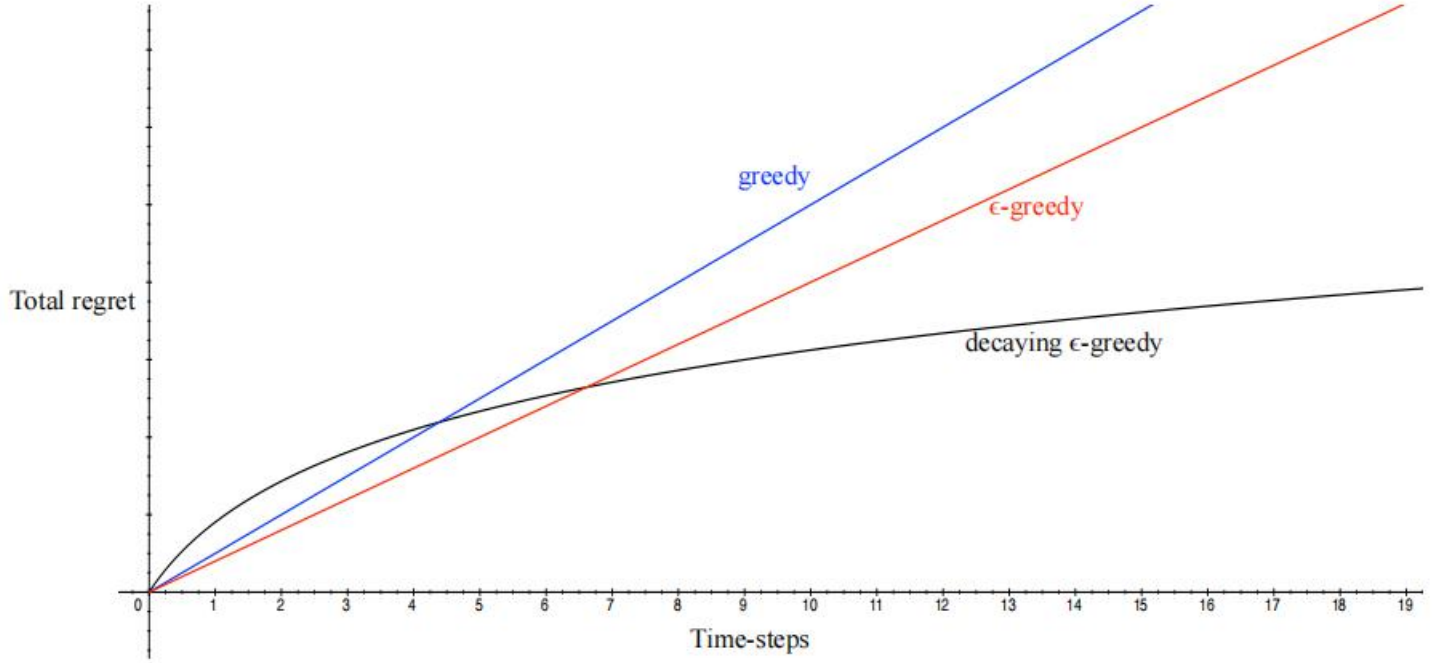
计数后悔值

- 定义计数 $N_t(a)$ 为到 t 时刻时已执行行为A的次数；
- 定义差距 Δ_a 为最优价值 a^* 与行为a的价值之间的差；
- 那么后悔值就是差距和计数的一个方程：

$$
\begin{aligned}
L_t &= \mathbb{E}\left[\sum_{\tau=1}^{t} V^* - Q(a_\tau)\right] \\
&= \sum_{a \in \mathcal{A}} \mathbb{E}\left[N_t(a)\right](V^* - Q(a)) \\
&= \sum_{a \in \mathcal{A}} \mathbb{E}\left[N_t(a)\right] \Delta_a
\end{aligned}
$$

- 一个好的算法应该尽量减少那些差距较大的行为的次数
- 问题是：差距是未知的！

线性和次线性的后悔值



- 如果一个算法永远地探索，它将会有线性的总后悔值（锁死在次优动作）
- 如果一个算法从未被探索过，它也将会有线性的总后悔值（总是有随机行为）
- 是否有可能实现次线性的总后悔值？

贪婪算法

- 我们提出一个算法近似该行为的实际价值 $\hat{Q}_t(a) \approx Q(a)$ ，通过蒙特卡罗评价来估计每个动作的值：

$$\hat{Q}_t(a) = \frac{1}{N_t(a)} \sum_{t=1}^{T} r_t \mathbf{1}(a_t = a)$$

- 贪婪算法选择值最高的动作：

$$a_t^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \hat{Q}_t(a)$$

- 贪婪算法将锁定一个次优的动作，因此贪婪算法有线性的总后悔值。

Ɛ-greedy算法

- Ɛ-greedy算法将一直探索：

  有1-Ɛ的概率选择 $a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \hat{Q}(a)$

  有Ɛ的概率选择一个随机动作

- 有一个固定小的几率采取完全随机的行为，如采取随机行为，那将一直会带来一定后悔值，如果持续以虽小但却固定的几率采取随机行为，那么总的后悔值会一直递增，导致呈现与时间之间的线性关系。

$$l_t \geq \frac{\varepsilon}{A} \sum_{a \in A} \Delta_a$$

乐观初始估计(Optimistic Initialization)

- 这是一个简单却使用的方法：将Q(a)初始化为一个较高的值，通过递增蒙特卡罗评估来更新动作价值：

$$\hat{Q}_t(a_t) = \hat{Q}_{t-1} + \frac{1}{N_t(a_t)}(r_t - \hat{Q}_{t-1})$$

- 可以看出，某行为的价值会随着实际获得的即时奖励在初始设置的较高价值基础上不断得到更新，这在一定程度上达到了尽可能尝试所有可能的行为。但是该方法仍然可能锁死在次优行为上。理论上，该方法与greedy或Ɛ-greedy结合带来的结果同样是线性增加的总后悔值。
- 但是实际应用效果却非常好！

衰减ε-greedy(Decaying ε-greedy)

- 即随着时间的延长，ε值越来越小。假设我们现在知道每一个行为的最优价值 V^*，那么我们可以根据行为的价值计算出所有行为的差距 Δ_a。算法可设置为：如果一个行为的差距越小，则尝试该行为的机会越多；如果一个行为的差距越大，则尝试该行为的几率越小。数学表达如下：

$$c > 0$$

$$d = \min_{a|\Delta_a>0} \Delta_i$$

$$\epsilon_t = \min\left\{1, \frac{c|\mathcal{A}|}{d^2 t}\right\}$$

- 惊奇的是它能够使得总的后悔值呈现出与时间步长的次线性（sublinear）关系：对数关系。不巧的是，该方法需要事先知道每个行为的差距 Δ_a，实际上是不可行的。

总后悔值下限

- 任何算法的性能都是由最优臂和其他臂之间的相似性决定的，比较困难的问题
  比如：多个单臂给出的奖励值有很多时候非常接近，但其均值却差距较大。
- 因此，可以通过比较两个单臂价值（均值）的差距Δ以及描述其奖励分布的相似程度的KL散度 $KL(\mathcal{R}^a||\mathcal{R}^a*)$ 来判断总的后悔值下限：差距越大，后悔值越大；奖励分布的相似程度越高，后悔值越低。

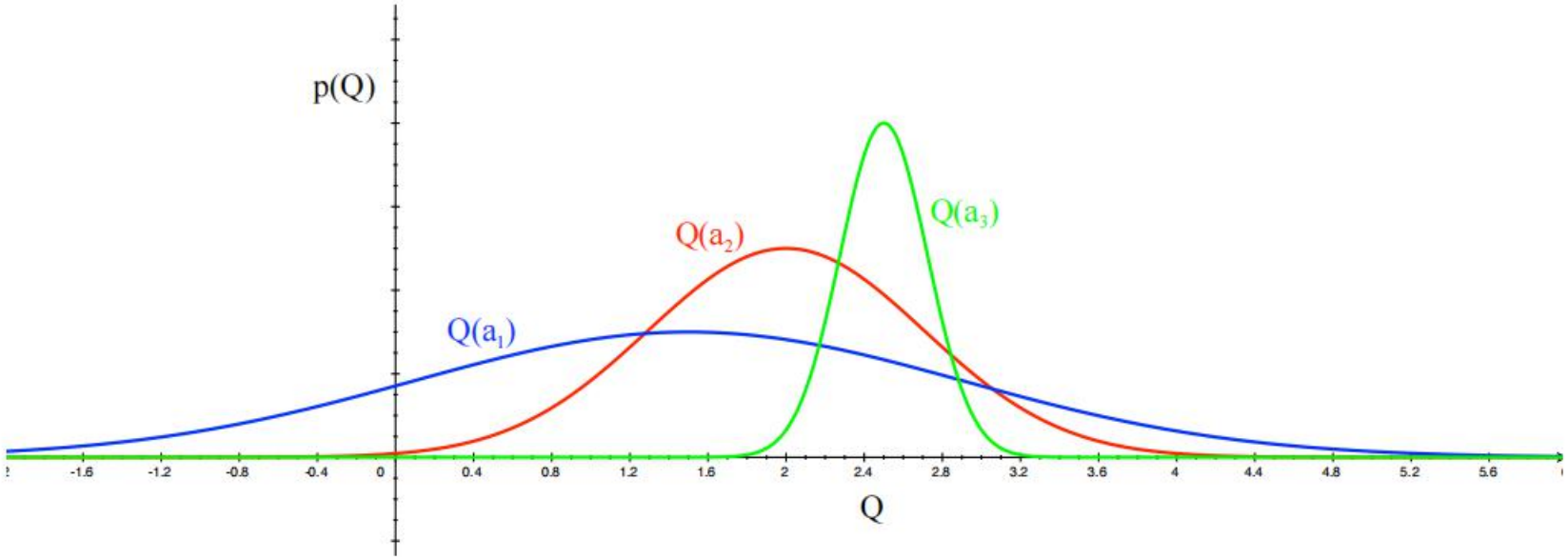
定理

存在一个总后悔值的下限，没有哪一个算法能够做得比这个下限更好

$$\lim_{t\to\infty} L_t \geq \log t \sum_{a|\Delta_a>0} \frac{\Delta_a}{KL(\mathcal{R}^a||\mathcal{R}^{a^*})}$$

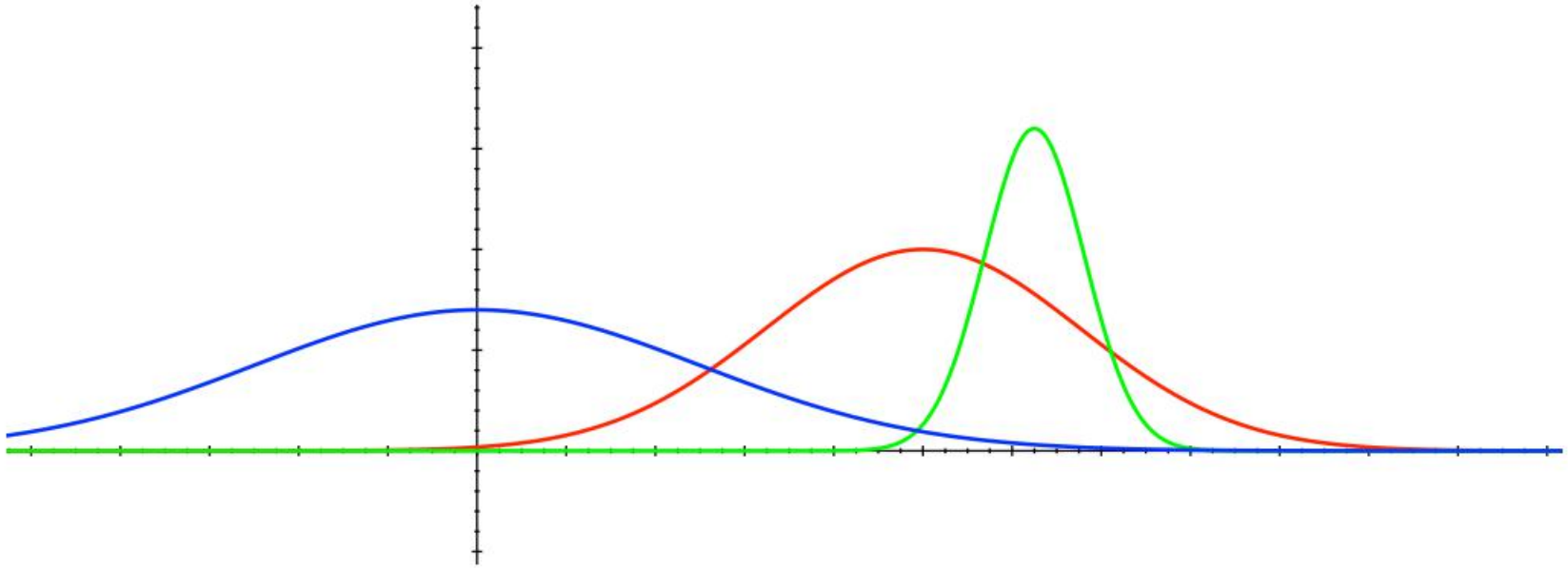
不确定行为优先搜索

- 想象一下现在由3个不同的单臂组成的多臂赌博机，现根据历史行为和奖励信息，绘制它们当前的奖励分布图。



- 我们对行动价值越不确定，探索行动就越重要，它可能是最好的行动
- 图中蓝色臂具有较高的不确定性

不确定行为优先搜索（2）



- 当我们选择蓝色的动作后，对这个价值就不那么不确定了。而且更有可能选择另一个行动，直到我们判断出最好的行动

置信区间上界(Upper Confidence Bound, UCB)

- 估计每个动作值的上置信度 $\hat{U}_t(a)$，这使得 $Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ 的概率较高。
- 当某一行为的计数较少时，该行为价值可信度上的价值上限将偏离均值较多；随着针对某一行为的奖励数据越来越多，该行为价值在某一可信度的上限将越来越接近均值。
- 因此我们可以用置信区间上界来指导行为的选择，令：

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \hat{Q}_t(a) + \hat{U}_t(a)$$

霍夫丁不等式(Hoeffding's Inequality)

Theorem (Hoeffding's Inequality)

Let $X_1, \ldots, X_t$ be i.i.d. random variables in [0,1], and let $\overline{X}_t = \frac{1}{\tau}\sum_{\tau=1}^{t} X_\tau$ be the sample mean. Then

$$\mathbb{P}\left[\mathbb{E}[X] > \overline{X}_t + u\right] \leq e^{-2tu^2}$$

该定理给出了位于区间[0,1]的两两随机变量其期望与均值之间满足的关系。结合该不等式，很容易得到:

$$\mathbb{P}\left[Q(a) > \hat{Q}_t(a) + U_t(a)\right] \leq e^{-2N_t(a)U_t(a)^2}$$

计算置信区间上界

- 假定我们设定行为的价值有p的概率超过我们设置的可信区间上界，即令：

$$e^{-2N_t(a)U_t(a)^2} = p$$

- 那么可以得到：

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

- 随着时间步长的增加，我们逐渐减少p值，比如 $p = t^{-4}$ ，那么随着时间步长趋向无穷大，我们据此可以得到最佳行为。

$$U_t(a) = \sqrt{\frac{2\log t}{N_t(a)}}$$

UCB1

- 给出实际应用时 $U_t(a)$ 和 a_t 的公式：

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}} \qquad U_t(a) = \sqrt{\frac{2 \log t}{N_t(a)}}$$

- 注：上式中，argmax是针对后两项整体的，式中Nt(a)是行为a的计数、Q(a)是根据历史数据获得的奖励的平均值。

Theorem

The UCB algorithm achieves logarithmic asymptotic total regret

$$\lim_{t \to \infty} L_t \leq 8 \log t \sum_{a | \Delta_a > 0} \Delta_a$$

例子: UCB vs. ε-Greedy On 十臂赌博机

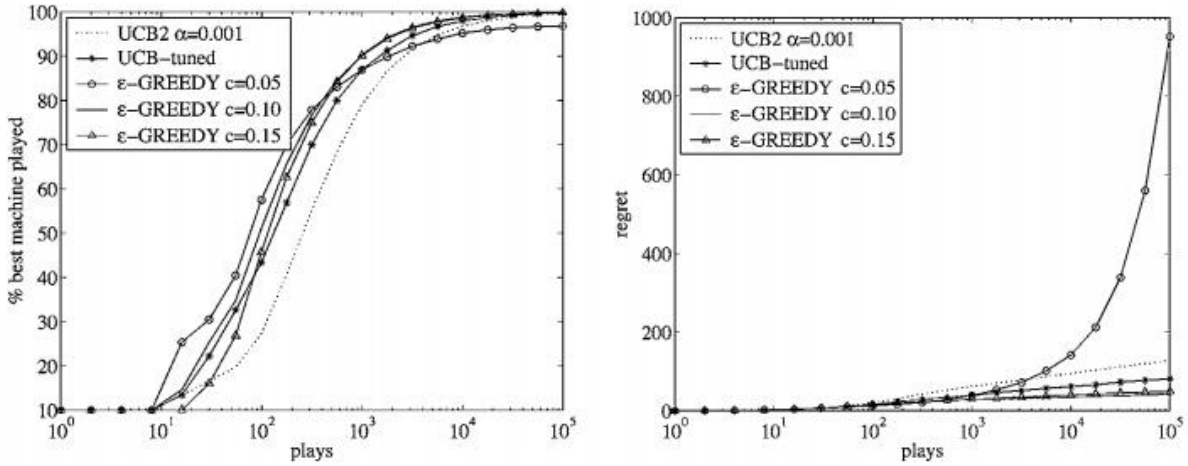


Figure 9. Comparison on distribution 11 (10 machines with parameters 0.9, 0.6, . . . , 0.6).

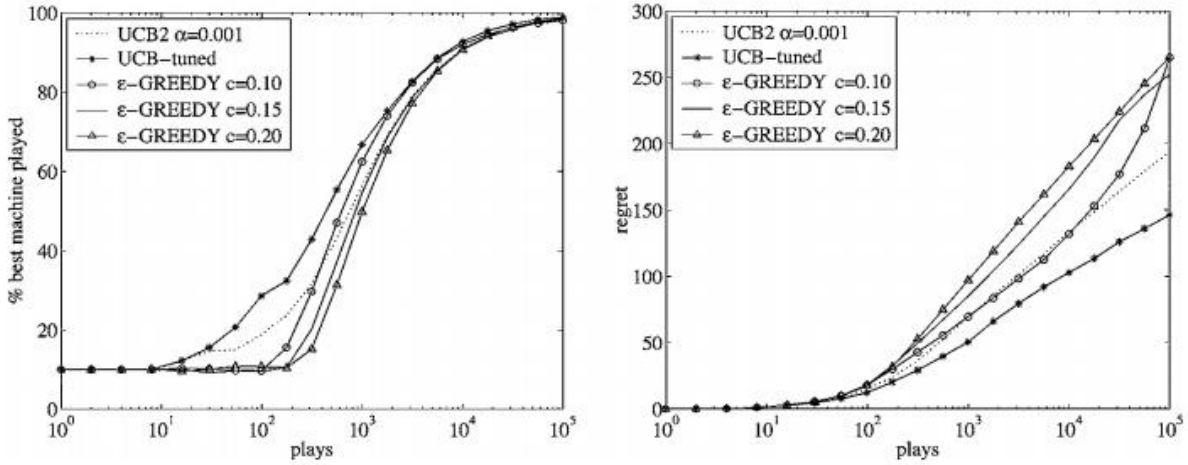


Figure 10. Comparison on distribution 12 (10 machines with parameters 0.9, 0.8, 0.8, 0.8, 0.7, 0.7, 0.7, 0.6, 0.6, 0.6).

贝叶斯赌博机（Bayesian Bandits）

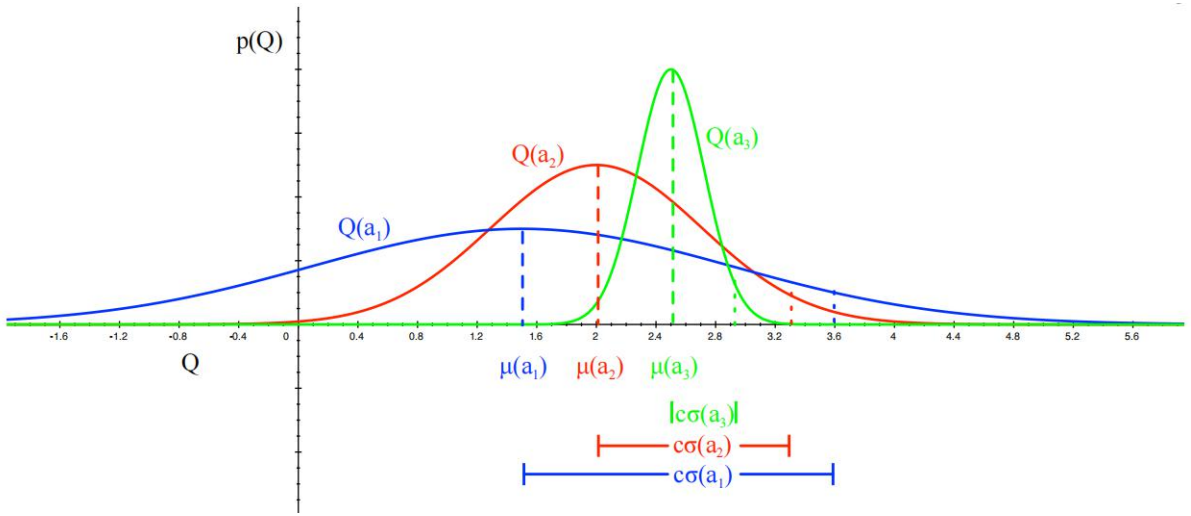
- 到目前为止，我们还没有对奖励分配R进行任何假设，除了奖励的边界。

贝叶斯赌博机利用了对奖励的先验知识 $p[\mathcal{R}]$，计算奖励的后验分布 $p[\mathcal{R} \mid h_t]$ 这里的 $h_t = a_1, r_1, \ldots, a_{t-1}, r_{t-1}$ 是历史。

- 使用后验去指导探索有两种方法：
  - 置信区间上界（贝叶斯UCB）
  - 概率匹配（汤普森抽样）
- 如果先验知识是准确的，则性能更好

贝叶斯UCB的例子：独立高斯分布

- 假设各单臂赌博机服从相互独立的高斯分布，可以用每一个单臂赌博机的均值和标准差参数化整体奖励分布：



- 计算 μ_a 和 σ_a^2 上的高斯后验（根据贝叶斯定律）

$$p\left[\mu_a, \sigma_a^2 \mid h_t\right] \propto p\left[\mu_a, \sigma_a^2\right] \prod_{t \mid a_t=a} \mathcal{N}(r_t; \mu_a, \sigma_a^2)$$

贝叶斯UCB的例子：独立高斯分布（2）

- 选择均值和一定比例的标准差之和来作为UCB算法中的置信区间上限，即依据下式选择后续行为：

$$a_t = \text{argmax}\ \mu_a + c\sigma_a/\sqrt{N(a)}$$

概率匹配 Probability Matching

- 概率匹配的想法先估计每一个行为可能是最佳行为的概率，然后依据这个概率来选择后续行为。

$$\pi(a \mid h_t) = \mathbb{P}\left[Q(a) > Q(a'), \forall a' \neq a \mid h_t\right]$$

- 该算法背后隐藏的思想是：越不确定价值的行为有着越高的几率被选择，这种被选择的目的是通过采样减少其不确定性，进而调整后续策略。

汤普森抽样

- **Thompson sampling算法是基于该思想的一种实际可行的算法，该算法实现起来非常简单，同时也是一个非常接近总后悔值对数关系的一个算法。该算法的步骤如下：**

1. 利用历史信息构建各单臂的奖励分布估计

2. 依次从每一个分布中采样得到所有行为对应即时奖励的采样值

3. 选取最大采样值对应的行为。

$$\pi(a \mid h_t) = \mathbb{P}\left[Q(a) > Q(a'), \forall a' \neq a \mid h_t\right]$$
$$= \mathbb{E}_{\mathcal{R}|h_t}\left[\mathbf{1}(a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(a))\right]$$

汤普森抽样

- 该算法的采样过程中利用到了历史信息得到的分布，同时行为得到的真实奖励值将更新该行为的分布估计。

- 汤普森采样达到了Lai和 Robbins的下界

信息价值 Value of Information

探索是有用的因为它获得了价值。

我们可以定义信息的价值吗？

- 在做出决定之前，决策者会为了获得这些信息而准备支付多少报酬？
- 获得信息后的获得的是长期奖励-亦或是即时奖励

在不确定的情况下，信息增益较高，因此，更多地探索不确定的情况是有意义的。

如果我们知道信息的价值，我们就可以最优地权衡探索和利用

信息状态空间

我们认为赌博机是一步决策问题（one-step decision-making problems）

可以将这个问题视为序贯决策问题吗？

在每一步中都有一个信息状态 $\tilde{s}$

$\tilde{s}$ 是历史的一个统计 $\tilde{s}_t = f(h_t)$，总结迄今为止积累的所有信息

每个操作a都会导致转换到新的信息状态 $\tilde{s}'$（通过添加信息），其概率为 $\tilde{\mathcal{P}}^a_{\tilde{s},\tilde{s}'}$

这定义了增强信息状态空间中的MDP $\tilde{\mathcal{M}}$

$$\tilde{\mathcal{M}} = \langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}, \gamma \rangle$$

例子：伯努利赌博机

- 考虑一个伯努利赌博机，比如 $\mathcal{R}^a = \mathcal{B}(\mu_a)$
- μ_a是指赢了或输掉一个游戏的概率
- 想找到哪个赌博机的臂有最高的μ_a
- 这个问题的信息状态为 $\tilde{s} = \langle \alpha, \beta \rangle$
- α_a记录的是拉动赌博臂收益为0的赌博臂
- β_a记录的是拉动赌博臂收益为1的赌博臂

解决信息状态空间赌博机

我们现在对信息状态有一个无限的MDP，这个MDP可以通过强化学习来解决。

比如：

1、模型无关的强化学习（Q-learning）

2、基于贝叶斯模型的强化学习，这种方法被称为贝叶斯自适应RL，发现贝叶斯的最优探索/利用权衡（Gittins indices）

贝叶斯自适应伯努利赌博机

从 $Beta(\alpha_a, \beta_a)$ 开始，先于奖励函数 $\mathcal{R}^a$ 。

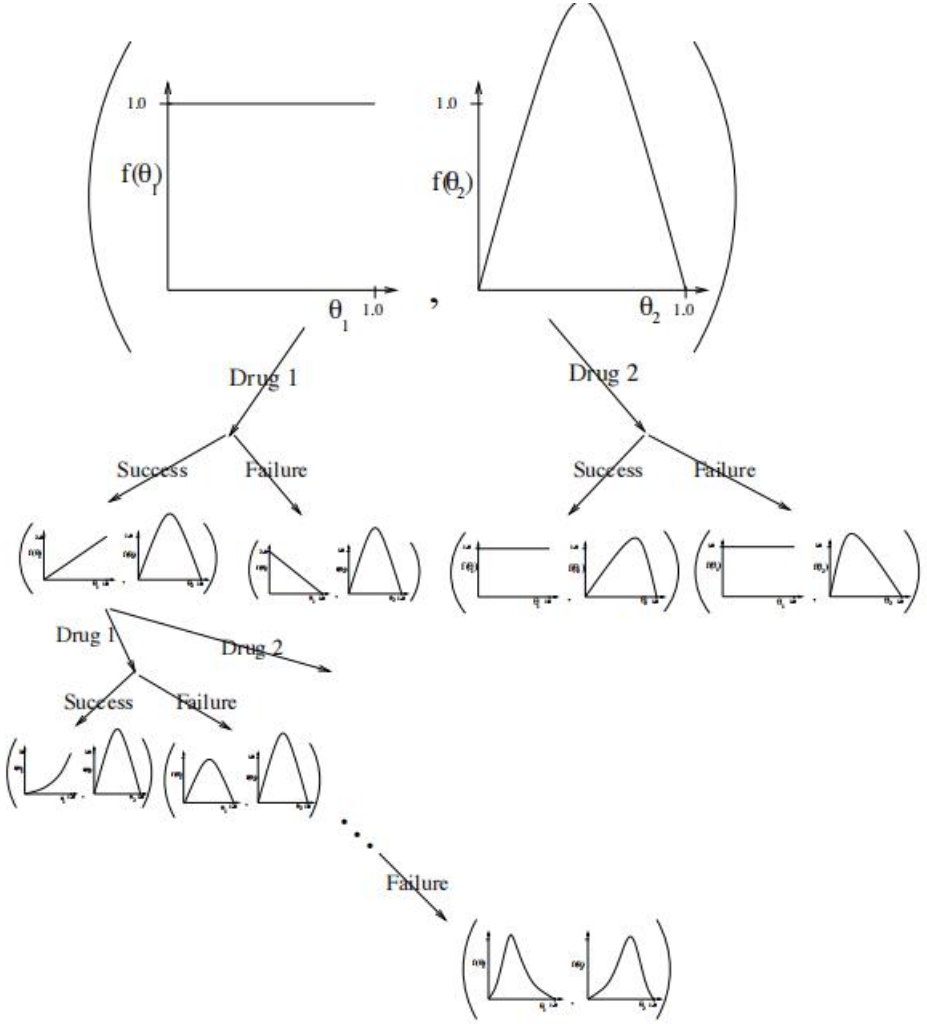
每次执行动作a之后更新 $\mathcal{R}^a$ 的后验：

$Beta(\alpha_a + 1, \beta_a)$ if $r = 0$
$Beta(\alpha_a, \beta_a + 1)$ if $r = 1$

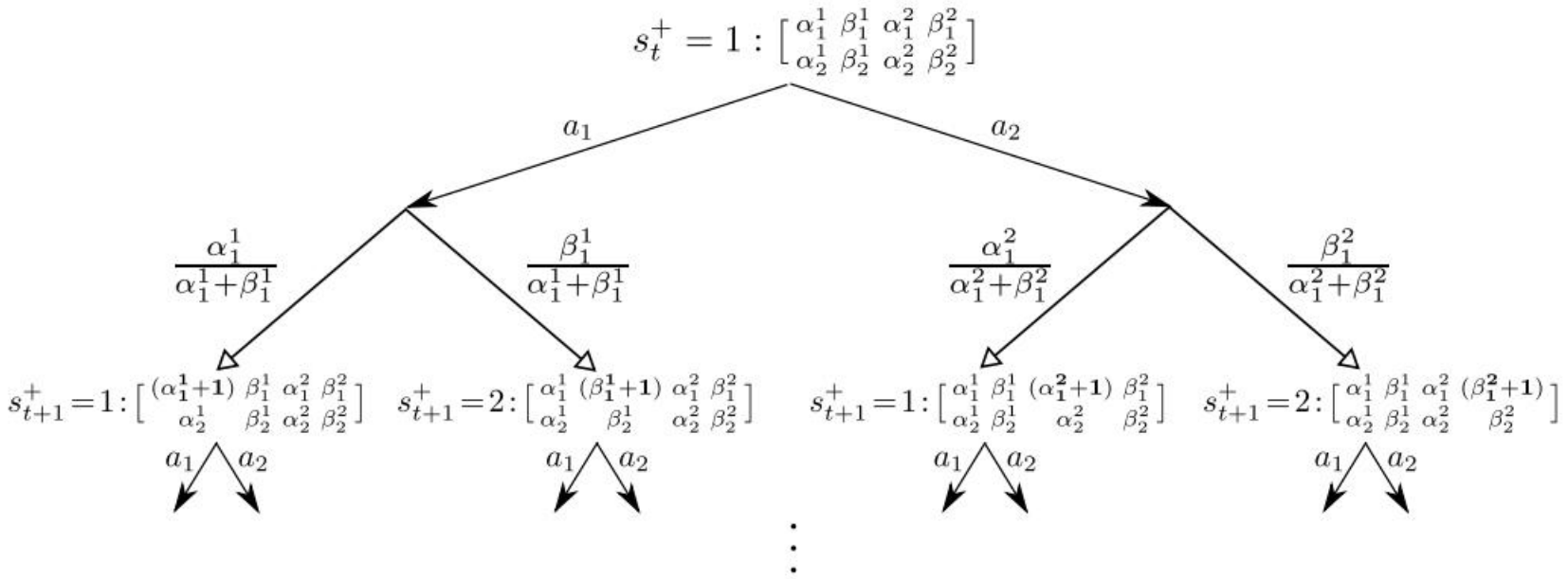
这定义了贝叶斯自适应MDP的转换函数 $\tilde{\mathcal{P}}$

信息状态 $\langle \alpha, \beta \rangle$ 对应于奖励模型 $Beta(\alpha, \beta)$

每个状态转换都对应于一个贝叶斯模型的更新



针对伯努利赌博机的贝叶斯自适应MDP



伯努利赌博机的Gittins Indices

贝叶斯自适应MDP可以通过动态规划来求解，这个解决方案被称为Gittins Indices。

但是因为信息状态空间太大，贝叶斯自适应MDP的精确解通常是难以解决的。

最近提出的idea：应用基于模拟的搜索(Guez et al. 2012)，其思路是：

- 在信息状态空间中进行前向搜索
- 使用当前信息状态的模拟

上下文赌博机

上下文赌博机是一个元组 $\langle \mathcal{A}, \mathcal{S}, \mathcal{R} \rangle$

总的目标是最大化累计收入 $\sum_{\tau=1}^{t} r_\tau$

A是一系列知道的动作

$\mathcal{S} = \mathbb{P}[s]$ 是未知的状态的分布

$\mathcal{R}_s^a(r) = \mathbb{P}[r|s,a]$ 是未知的收益可能性分布

在每个时间步t：

- 环境生成状态 $s_t \sim \mathcal{S}$
- 智能体选择动作 $a_t \in \mathcal{A}$
- 环境生成收益 $r_t \sim \mathcal{R}_{s_t}^{a_t}$

线性回归

动作值函数是对状态s和动作a的期望奖励

$$Q(s,a) = \mathbb{E}\left[r|s,a\right]$$

具有线性函数近似器的估计值函数

$$Q_\theta(s,a) = \phi(s,a)^\top \theta \approx Q(s,a)$$

用最小二乘回归法估计参数

$$A_t = \sum_{\tau=1}^{t} \phi(s_\tau, a_\tau)\phi(s_\tau, a_\tau)^\top$$

$$b_t = \sum_{\tau=1}^{t} \phi(s_\tau, a_\tau) r_\tau$$

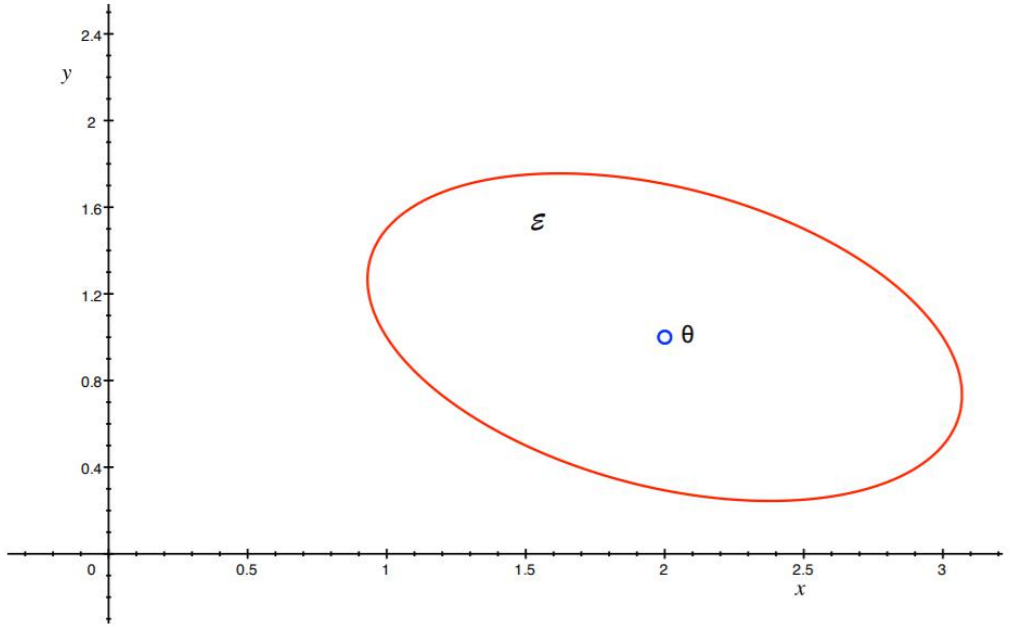
$$\theta_t = A_t^{-1} b_t$$

线性上置信度

最小二乘回归估计了平均动作值 $Q_\theta(s,a)$ ，同时它也可以估计动作值的方差 $\sigma^2_\theta(s,a)$

即由于参数估计误差而产生的不确定性，增加一个不确定度，$U_\theta(s,a)=c\sigma$ ，即定义UCB为高于平均值的c个标准差。

几何解释



定义参数 θ_t 周围的置信椭球 $\mathcal{E}_t$,使 $\mathcal{E}_t$ 包含真参数 θ^* 具有高概率
使用此椭球来估计操作值的不确定性
选择椭球内的参数，使行动价值最大化

$$\underset{\theta \in \mathcal{E}}{\operatorname{argmax}}\ Q_\theta(s, a)$$

计算线性上置信范围

对于最小二乘回归，参数协方差为 A^{-1}

作用值在特征上呈线性，$Q_\theta(s,a) = \phi(s,a)^\top \theta$

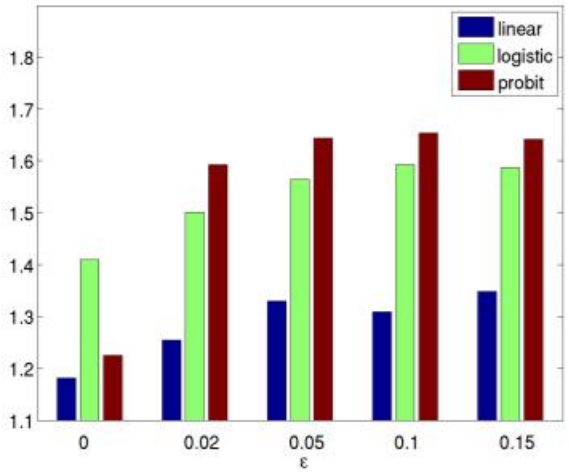
所以动作值方差是二次的，$\sigma_\theta^2(s,a) = \phi(s,a)^\top A^{-1} \phi(s,a)$

上置信界是 $Q_\theta(s,a) + c\sqrt{\phi(s,a)^\top A^{-1} \phi(s,a)}$

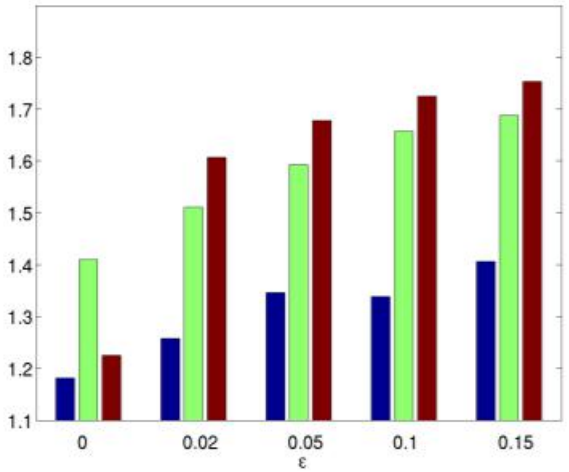
选择使置信上限最大化的行动

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q_\theta(s_t,a) + c\sqrt{\phi(s_t,a)^\top A_t^{-1} \phi(s_t,a)}$$

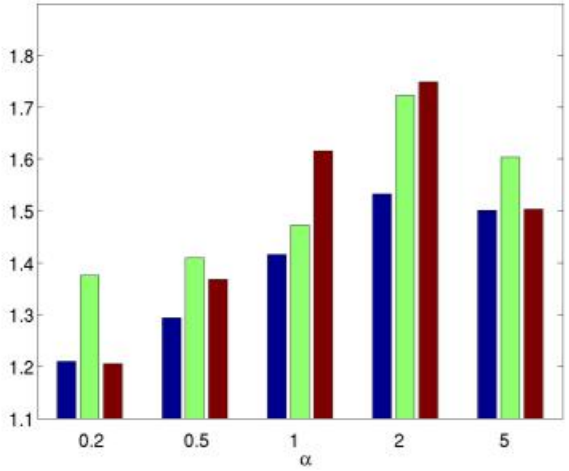
示例:选择头版新闻的线性UCB



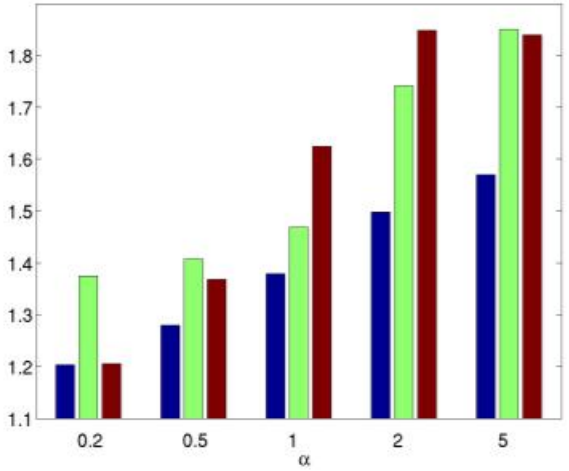
(a)



(b)



(c)



(d)

MDP的探索/开发原则

- 探索/利用的相同原则也适用于MDP:
- 朴素探索
- 乐观初始估计
- 不确定优先
- 概率匹配
- 信息状态搜索

乐观初始估计:无模型RL

- 初始化动作价值函数 $Q(s,a)$ to $\frac{r_{max}}{1-\gamma}$
- 运行最流行的无模型RL算法：
- **Monte-Carlo control**
- **Sarsa**
- **Q-learning**
- 鼓励系统地探索状态和行为

乐观初始估计:基于模型的RL

- 构建MDP的乐观模型
- 初始化过渡去 heaven(即转换到带有 r_{max} 奖励的终端状态)
- 利用偏好规划算法求解乐观MDP：
- policy iteration
- value iteration
- tree search
- ...
- 鼓励系统地探索状态和行为，比如：RMax算法(Brafman and Tennenholtz)

上置信范围:无模型RL

在动作价值函数 $Q^{\pi}(s,a)$ 上最大化UCB

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(s_t,a) + U(s_t,a)$$

- 估计策略评估中的不确定性(容易)
- 忽视策略改善带来的不确定性

在最优动作价值函数 $Q^{*}(s,a)$ 上最大化UCB

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(s_t,a) + U_1(s_t,a) + U_2(s_t,a)$$

- 估计策略评估中的不确定性(容易)
- 加上策略改进中的不确定性(困难)

贝叶斯基于模型的RL

在MDP模型上保持后验分布

估计过渡和奖励，$p[\mathcal{P}, \mathcal{R} \mid h_t]$ ，其中 $h_t = s_1, a_1, r_2, \ldots, s_t$ 是历史

使用后验引导探索

- 上置信范围(贝叶斯UCB)
- 概率匹配(汤普森抽样)

汤普森抽样:基于模型的RL

汤普森抽样实现概率匹配

$$\pi(s,a \mid h_t) = \mathbb{P}\left[Q^*(s,a) > Q^*(s,a'), \forall a' \neq a \mid h_t\right]$$
$$= \mathbb{E}_{\mathcal{P},\mathcal{R}|h_t}\left[\mathbf{1}(a = \underset{a \in \mathcal{A}}{\text{argmax}}\ Q^*(s,a))\right]$$

利用贝叶斯定律计算后验分布 $p\left[\mathcal{P},\mathcal{R} \mid h_t\right]$

采样MDP，从后验取样P, R

使用最喜欢的规划算法解决MDP，得到 $Q^*(s,a)$

为样本MDP选择最优行动， $a_t = \underset{a \in \mathcal{A}}{\text{argmax}}\ Q^*(s_t,a)$

信息状态搜索在MDP

MDP可以扩展到包含信息状态

现在扩增状态态是 $\langle s, \tilde{s} \rangle$

- 其中s是MDP和ndx中的原始状态。
- $\tilde{s}$ 是历史的一个统计数据(积累的信息)

每一个动作a都会引起一次转换

- 到一个新的状态 s' 概率为 $\mathcal{P}^a_{s,s'}$
- 到一个新的信息状态 $\tilde{s}'$

定义MDP $\tilde{\mathcal{M}}$ 在增强信息状态空间

$$\tilde{\mathcal{M}} = \langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}, \gamma \rangle$$

贝叶斯自适应MDP

MDP模型的后验分布是一种信息状态

$$\tilde{s}_t = \mathbb{P}\left[\mathcal{P}, \mathcal{R} | h_t\right]$$

扩增的MDP超过$\langle s, \tilde{s} \rangle$被称为贝叶斯自适应MDP，解决这个MDP以找到最佳的探索/开发平衡(相对于先前)

然而，贝叶斯自适应MDP通常是巨大的，基于模拟的搜索已被证明是有效的

(Guez et al.)

结论

我们的内容涵盖了探索/开发的几个原则：

- 朴素探索（ Ɛ-greedy）
- 乐观初始估计
- 不确定优先
- 概率匹配
- 信息状态搜索

每个原则都是在赌博机设定中发展起来的，但同样的原则也适用于MDP设置

The End