数学 机器学习 运筹学 强化学习 (Reinforcement Learning)

关注者 571 被浏览 314,427

什么是汤普森采样（Thompson sampling）？

在看到一个解决Multi-armed bandit（多臂老虎机）问题时，提到Thompson sampling，谁能通俗的讲一下，维基百科上面讲的看不...显示全部

10 个回答 默认排序

覃含章
数学等 2 个话题下的优秀答主 已关注

本回答来自我的知乎专栏文章系列：在线学习(MAB)与强化学习(RL)[4]。这篇回答将主要谈谈**在Bandit情况下我们如何理解TS算法，以及它和在非贝叶斯情境下著名的UCB算法⁺的关系**。当然，实际上TS算法（也包括UCB算法等）在更一般的RL情境下仍然有广泛的应用。但这里为了简洁起见，我的讨论仅限于RL中非常特殊的一类最基本的bandit情形，对一般情形感兴趣的同学可以关注我的专栏文章[5]。对Bandit完全不熟悉的同学建议从专栏文章[1]系列看起（你至少需要理解bandit才能开始理解TS算法啊！）。

本回答主要的参考文献是：

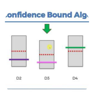
Russo D J, Van Roy B, Kazerouni A, et al. A tutorial on thompson sampling[J]. Foundations and Trends® in Machine Learning, 2018, 11(1): 1-96.

Slivkins教科书第三章：slivkins.com/work/MAB-b...。

一、贪心算法回顾

我们首先回顾一下贪心算法的思想，并引入TS算法的基本思想。基本的思想其实在非贝叶斯的情况下已经有比较详细的讨论，见：

覃含章：在线学习(MAB)与强化学习(RL)[2]：IID Bandit的一些算法
243 赞同 · 22 评论 文章



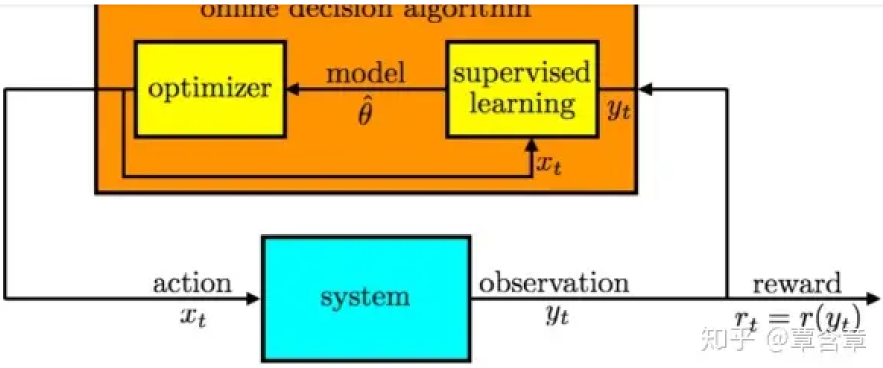
这边我们就再重新简单总结一下。

贪心算法（greedy algorithm）的思路非常直接，就是：

1. 使用过去的数据去估计（estimate）一个模型（model）
2. 选择能够optimize所估计模型的动作（action）

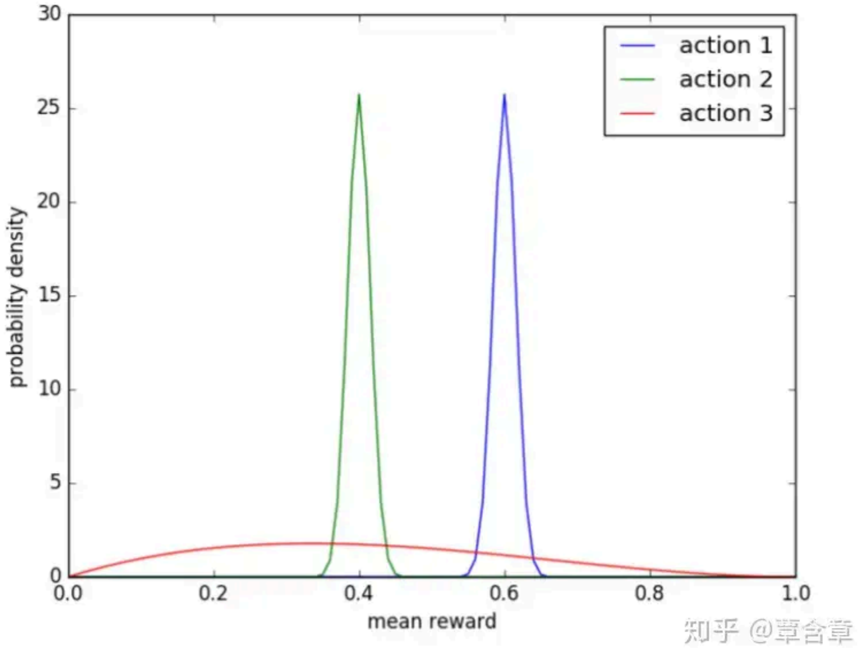
图例的话其实就是题图的这么一个流程，再贴一遍：

什么是汤普森采样（Thompson sampling）？



贪心算法图例

那么在非贝叶斯的情形下我们已经指出过贪心算法的不足之处，也就是**主动的探索（active exploration）**不足。我们这边再用一个直观的例子来说明这一点。



贪心算法的缺点：一个例子

我们考虑贝叶斯情形下的Bernoulli Bandit的一个例子。在这个例子中，一共有3个action（arm）可以选择，对应的mean reward $\theta \in \mathbb{R}^3$.（即每个action获得reward 1的概率，不然就得到reward 0） 自然，我们的算法事先是不知道 θ 的值的，我们能做的只是不断地根据观察到的样本去估计 θ 的值。具体来说，我们的算法需要时时刻刻有一个对 θ 的belief，或者说后验分布（posterior distribution）。

假设我们的belief如上图所示，对action 1和action 2可以认为我们已经有比较多的data，然后对 θ_1, θ_2 的分布估计地算是相对准确了，而action 3缺乏data，对 θ_3 估计的分布可能还不太准。那么我们马上知道基本上action 2应该是不用管了，因为大概率 $\theta_1 > \theta_2$. 然而，因为我们还不太确定 θ_1 和 θ_3 之间的大小关系，其实还是应该explore一下action 3的。然而，我们知道贪心算法不会做这个exploration，因为如果你贪心地来看目前的belief， θ_3 的估值是要小于 θ_1 的，也就是说**贪心算法在这种情形下只会不停地选择action 1**。这也就是我们前面说了贪心算法缺乏主动探索(active exploration)，在这个例子里，如果 $\theta_3 > \theta_1$ 那么贪心算法就远不是最优的了。

一种简单粗暴的解决方案就是所谓的 $\epsilon-$ greedy算法，这个我们在非贝叶斯情形下（系列文章[2]）里已经有过详细讨论，这里就不再多说了，只是再提一下 $\epsilon-$ greedy只是一种随机算法(randomized algorithm)，在纯贪心算法的基础上加入一定概率的uniform exploration（也就是randomize纯贪心算法和uniform exploration）。当然，在实际中这种算法对于纯贪心算法往往会有比较大的提高，但很多时候也是远非最优。因为比如说在上面的例子里面假使我们已经试了足够多次，且已经比较明确地得到了 $\theta_1,$
资源了，这个时候我们反而应该

什么是汤普森采样（Thompson sampling）？

二、Thompson Sampling

回顾完了贪心算法，我们还是沿用上面的例子，谈一谈TS算法，以及**为什么实际中它往往会比贪心算法好**。具体来说，我们就考虑Beta-Bernoulli Bandit，也就是说，对于 θ 我们的先验分布（prior distribution）是Beta分布，而每个arm reward的分布是以 θ 为参数的Bernoulli分布。容易知道，在这种情况下， θ 的后验分布仍然是Beta分布。

这里只用到了最基本的概率论和统计的知识，以防大家有些失忆，我写一些关键的公式出来。假设现在我们有 K 个arm，那么mean rewards $\theta = (\theta_1, \ldots, \theta_K)$ 事先是不知道的。在一开始，算法会选择一个action， a_1 ，然后会观察到reward $r_1 \in \{0, 1\}$ ，这是一个从Bernoulli分布draw的sample，即 $\mathbb{P}[r_1 = 1|a_1, \theta] = \theta_{a_1}, \mathbb{P}[r_1 = 0|a_1, \theta] = 1 - \theta_{a_1}$. 然后 $t = 2$ 的时候也是类似，算法根据历史数据选择action a_2 ，然后观察到跟 a_2, θ 所决定的 $r_2 \in \{0, 1\}$. 以此类推，一直持续到 $t = T$ 的时候算法停止。

除此之外，我们假设了一开始对 θ 的prior belief符合Beta分布（参数为 $\alpha = (\alpha_1, \ldots, \alpha_K), \beta = (\beta_1, \ldots, \beta_K)$ ），具体来说对每个arm k 的mean reward对应的先验分布为（ Γ 表示Gamma函数）

$$p(\theta_K) = \frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k - 1}(1-\theta_k)^{\beta_k - 1}.$$

容易知道，根据Baye's rule,后验分布也是Beta分布。具体来说，在time step t 我们可以这样更新关于 θ 的后验分布：

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k), & \text{if } a_t \neq k, \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t), & \text{if } a_t = k. \end{cases}$$

也就是说，如果我们选择了 arm k，那么如果得到reward 1就将相应的 α_k 加一（ β_k 不变），不然（reward 0）就将相应的 β_k 加一（ α_k 不变）。这个简单的更新规则也让Beta Bernoulli bandit成为基本上最适合当例子的贝叶斯bandit情形。

对Beta分布不熟悉的同学，其实在贝叶斯框架下理解起来也是比较直观的。注意到Beta(1,1)分布就等于[0,1]区间上的均匀分布（uniform distribution）。如果我们把这个当成prior distribution，那么我们可以把后验分布里的参数 α_k, β_k 当成“计数器”，即 α_k 是reward为1的次数， β_k 是reward为0的次数。我们的Beta分布，当 α_k 相比 β_k 比较大的时候则就会右倾（mean reward较大），反之则会左倾（mean reward较小）。比如在我们之前的图片里，action 1，2，3的分布就分别为Beta(601,401),Beta(401,601),Beta(2,3). 所以，这里我们也能定量化地看出action 3之前试验的次数比较少，而action 1,2之前试验的次数已经很多了。

Algorithm 1: $\text{BernGreedy}(K, \alpha, \beta)$

1. for $t = 1, 2, \ldots$do
2. *//estimate model*
3. for $k = 1, \ldots, K$do
4. $\hat{\theta}_k \leftarrow \alpha_k / (\alpha_k + \beta_k)$
5. end for
6. *//select and apply action:*
7. $a_t \leftarrow \arg\max_k \hat{\theta}_k$
8. Apply a_t and observe r_t
9. *//update distribution:*
10. $(\alpha_{a_t}, \beta_{a_t}) \leftarrow (\alpha_{a_t} + r_t, \beta_{a_t} + 1 - r_t)$
11. end for

Algorithm 2: $\text{BernTS}(K, \alpha, \beta)$

1. for $t = 1, 2, \ldots$do
2. *//sample model*

什么是汤普森采样（Thompson sampling）？

end for
//select and apply action:
$a_t \leftarrow \arg\max_k \hat{\theta}_k$
Apply a_t and observe r_t
//update distribution:
$(\alpha_{a_t}, \beta_{a_t}) \leftarrow (\alpha_{a_t} + r_t, \beta_{a_t} + 1 - r_t)$
end for

那么这边我们给出（见上）在Beta Bernoulli Bandit情形下的，之前贪心算法，和TS算法的伪代码，这样可以比较直接地进行比较。具体来说，主要的区别在于两个算法中贪心算法每个time step第一步是estimate model，而TS算法中第一步则是sample model。也就是说，$\hat{\theta}_k$ 决定的方法不同，一个是直接用sample average，即从sample中估计出来的成功率，$\alpha_k/(\alpha_k+\beta_k)$,而与此不同的就是TS算法是sample一个model，即 $\hat{\theta}_k$ 是直接从后验的 $\mathrm{Beta}(\alpha_k, \beta_k)$ 分布中采样出来。

乍看起来复杂度其实差不多：在Beta Bernoulli Bandit的情形下TS算法的复杂度看起来其实跟贪心算法差不多。那么TS算法的优势是什么呢？个人理解，TS算法是更**自然**的，也是天然randomized的，我们对于 θ 的估计不再是sample average，而是从我们当前的后验分布（belief）直接采样出来的。在这种情况下，TS算法天然就会同时完成exploitation和exploration这两个任务，因为如果一个arm还没有怎么被选择过，那么从这个arm采样出来的 $\hat{\theta}_k$ 会以近似均匀的概率落在整个区间上（相当于uniform exploration）。而一个arm如果被选择的次数多了，那么自然估计的就比较准了，如果这个arm比较"好"，则从它的后验分布里采样出来的 $\hat{\theta}_k$ 就有大概率是比较高的，这个arm也就比较容易会被选中（exploitation）。在非贝叶斯框架下，我们看到这也是UCB类算法相对于贪心算法的优势，而这边同样在贝叶斯框架下TS算法相对于贪心算法的优势。之后，我们再更细致地讨论一下UCB算法和TS算法的联系。

三、TS算法的一些分析，和UCB算法的联系

本回答的最后，我们在bandit情形下给出分析TS算法的一般思路，以及这个思路与之前分析UCB算法的联系。我们首先注意到，在贝叶斯bandit的情形下，我们一般考虑的目标是Bayesian regret，具体来说，我们定义

$$\mathrm{BR}(t) = \mathbb{E}_{\theta\sim\text{prior}}\left[\mathbb{E}\left[\theta^* \cdot t - \sum_{s=1}^{t} \theta_{a_s} \middle| \theta\right]\right]$$

为我们贝叶斯情形下的regret。和非贝叶斯情况的区别，主要就在内层的期望外层又套了一个对于 θ 的prior的期望。当然这其实看起来是比regret更强的一个东西，其实也确实如此，考虑Bayesian regret对于我们相比非贝叶斯情况下的regret分析是要有不少便利的。这里面，一个核心的假设就是，如果我们定义 t 时刻之前发生的事件（生成的 $\sigma-$ algebra）为 $\mathcal{F}_t$ ，那么，如果有个函数 $U_t(a_t)$ 关于（conditioned on） $\mathcal{F}_t$,是确定性（deterministic）的（假设 a_t, a_t^* 都是基于后验分布IID选取的），则我们有如下重要的关系式：（**这是贝叶斯bandit分析的精髓**）

$$\mathbb{E}[U_t(a^*)|\mathcal{F}_t] = \mathbb{E}[U_t(a_t)|\mathcal{F}_t].$$

注意这里，我们认为 a^* 就是对应 θ^* 的那个arm，即最优算法每个时刻 t 选择的action $a^* = \arg\max_{a\in\mathcal{A}} \theta_a$ ，而 a_t 是根据TS算法在每个时刻 t 所选择的action。**这是反应贝叶斯人信仰的重要假设**。因为实际上，我们有如下信仰：

当 $\mathcal{F}_t$ 固定，我们认为 a_t 和 a^* 是同分布的（$a_t \sim a^*$）。

至于这是为什么？注意到TS算法实质上就是从基于历史的 θ 后验分布中抓出一个样本并根据这个向量选择最好的arm，而 a^* 呢？同样应该是如此，因为我们当然应该相信，**基于历史 $\mathcal{F}_t$ ，我们的后验分布反应了当时对 θ 的真实信仰**。如果对这一点你没法信服，那你可能真的就只能去做个频率学家了，不然的话，欢迎成为贝叶斯人！

当然，前面也提到了，Bayesian r
最外面加上对prior的期望的分析，

什么是汤普森采样（Thompson sampling）？

我们这边就继续照着Bayesian regret这个目标说。注意到有了 $\mathbb{E}[U_t(a^*)|\mathcal{F}_t] = \mathbb{E}[U_t(a_t)|\mathcal{F}_t]$ 这个式子之后，我们其实就有对于任意满足前面条件的 U_t，

$$\begin{aligned}\mathbb{E}_{\theta\sim\text{prior}}[\mathbb{E}[\theta^* - \theta_{a_t}|\mathcal{F}_t]] =& \mathbb{E}_{\theta\sim\text{prior}}[\mathbb{E}[U_t(a_t) - U_t(a^*) + \theta^* - \theta_{a_t}|\mathcal{F}_t]] \\ =& \mathbb{E}_{\theta\sim\text{prior}}[\mathbb{E}[U_t(a_t) - \theta_{a_t}|\mathcal{F}_t]] + \mathbb{E}_{\theta\sim\text{prior}}[\mathbb{E}[U_t(a^*) - \theta_{a^*}|\mathcal{F}_t]] \\ =& \mathbb{E}[U_t(a_t) - \theta_{a_t}] + \mathbb{E}[U_t(a^*) - \theta_{a^*}].\end{aligned}$$

也就是说，我们把上式关于 t 加起来，就有

$$\text{BR}(T) = \sum_{t=1}^T \underbrace{\mathbb{E}[U_t(a_t) - \theta_{a_t}]}_{(*)} + \sum_{t=1}^T \underbrace{\mathbb{E}[\theta_{a^*} - U_t(a^*)]}_{(**)}.$$

也就是说其实我们的Bayesian regret有如上式这样非常简介的分解（decomposition）。**这个简洁的decomposition式就是贝叶斯bandit regret的分析核心**。不知道看过之前系列文章的你到这里会不会有点想法呢？嗯，我这边就直接往下讲了，这里注意我们 U_t 其实要求很宽，**我们是不是不妨就可以把它相应设作 θ_{a_t} 和 θ_{a^*} 的upper confidence bound（UCB）呢**？而且，这么设了之后，我们显然知道怎么把这两项bound住呢（参考系列文章[2]中对UCB算法的regret分析）。

其实到这边难度就不太大了。为了完整性，这边还是把对 $\text{BR}(T)$ 的证明思路大体捋一遍。

我们如果令 $\bar{\theta}_t(a)$ 表示到时刻 t 为止arm a 的reward的sample average。那么我们知道 $\forall\, a, t,$ 对每个 $\theta_t(a)$ 可以定义它的UCB和LCB如下：

$$U_t(a) = \bar{\theta}_t(a) + \sqrt{\frac{2\log(T)}{n_t(a)}},$$

$$L_t(a) = \bar{\theta}_t(a) - \sqrt{\frac{2\log(T)}{n_t(a)}},$$

注意和之前一样， $n_t(a)$ 代表的是 t 时刻以来arm a 被选择过的次数。那么，我们注意到对任意 $\gamma > 0,$ 如果有

$$\forall\, a, t,\ \mathbb{E}[(U_t(a) - \theta_a)^-] \le \frac{\gamma}{TK},$$

$$\forall\, a, t,\ \mathbb{E}[(\theta_a - L_t(a))^-] \le \frac{\gamma}{TK},$$

我们就分别有：

$$\begin{aligned}(**) \le& \mathbb{E}[(\theta_{a^*} - U_t(a^*))^+] \\ \le& \mathbb{E}\left[\sum_{a\in\mathcal{A}}(\theta_a - U_t(a))^+\right] \\ =& \sum_{a\in\mathcal{A}}\mathbb{E}[(U_t(a) - \theta_a)^-] \\ \le& \frac{\gamma}{T}.\end{aligned}$$

$$\begin{aligned}(*) =& \mathbb{E}[U_t(a_t) - L_t(a_t) + L_t(a_t) - \theta_{a_t}] \\ \le& \mathbb{E}[U_t(a_t) - L_t(a_t)] + \mathbb{E}[(L_t(a_t) - \theta_{a_t})^+] \\ \le& \mathbb{E}[U_t(a_t) - L_t(a_t)] + \mathbb{E}\left[\sum_{a\in\mathcal{A}}(L_t(a_t) - \theta_{a_t})^+\right] \\ =& \mathbb{E}[U_t(a_t) - L_t(a_t)] + \sum_{a\in\mathcal{A}}\mathbb{E}\left[(\theta_{a_t} - L_t(a_t))^-\right] \\ \le& \mathbb{E}[U_t(a_t) - L_t(a_t)] + \frac{\gamma}{T}.\end{aligned}$$

也就是说我们就能有 $\text{BR}(T) \le 2\gamma + \sum_{t=1}^T \mathbb{E}[U_t(a_t) - L_t(a_t)].$

利用系列文章[2]里对UCB算法分析的基本技巧，我们容易证明 可以取 $\gamma = 2,$ 且 $\sum_{t=1}^T \mathbb{E}[U_t(a_t) - L_t(a_t)] = O(\sqrt{KT\log T})$ （留作练习），这样子，我们就得到 $\text{BR}(T) = O(\sqrt{KT\log T}).$ 这就是贝叶斯bandit对Bayesian regret基本的一个分析思路

什么是汤普森采样（Thompson sampling）？

接地使用UCB和LCB的值，但在分析中人为地引入UCB和LCB作为Bayesian regret的decomposition，会让我们的分析事半功倍。那么这两个看起来很迥异，且一般用在两个截然不同情景的算法，其之间的这些联系，就希望大家可以再好好体会了！

编辑于 2019-07-11 23:33

王腾云 + 关注

229 人赞同了该回答 ›

最近刚好看到这篇文章[0]，说一下我的理解，不一定对哈，望大神指正

Probability matching：

假设我有一个很奇葩的硬币，我知道每次抛它正面向上的概率是0.6，反面向上的概率是0.4。那么，给你预测10次抛硬币，你预测几次向上几次向下?

如果使用Probability matching的策略来预测，那么预测结果是6次向上，4次向下。

如果使用贝叶斯决策策略（Bayesian decision strategy）来预测，那么预测结果是10次向上。

这两种方案，预测正确数目的期望为：10*（0.6*0.6 + 0.4*0.4） = 5.2，和10*（0.6*1+0.4*0）= 6

Multi-armed bandit：

你面前有K台老虎机，假设每台老虎机收益的概率服从分布$P_k(\theta)$，每台老虎机每次消耗服从分布$P_k'(\theta')$。那么，你应该以什么样的顺序来玩老虎机，使得你的收益最大化。

这里主要的面临exploration-exploitation困境。exploitation可以理解为根据现有观测值，使下一次收益最大化；exploration理解为对未知老虎机概率分布的探索，当我们探索的越完备时，越容易取到全局最优解。

Thompson sampling：

假设我们有一个上下文环境$x \in X$，做出一个动作$a \in A$，得到一个回报$r \in R$，那么这个回报的似然函数为$p(r|\theta, a, x)$，其中$\theta \in \Theta$为回报分布的参数。

假设我们知道先验概率分布$p(\theta)$

假设我们有历史观测三元组$D = \{(x, a, r)\}$

所以后验分布可以被计算出来$p(\theta|D) \propto p(D|\theta)p(\theta)$

Thompson sampling consists in playing the action $a^* \in A$ according to the probability that it maximizes the expected reward .（这里有没有看到Probability matching 的影子？）

回报的期望$E(r(\theta, a, x)) = \int_\theta I[E(r|\theta, a, x) = \max_{a'} E(r|\theta, a', x)]p(\theta|D)d\theta$