

常用链接

[我的随笔](#)[我的评论](#)[我的参与](#)[最新评论](#)[我的标签](#)

最新随笔

1. matplotlib库笔记1
2. python语法的一些问题1
3. 高级系统项目管理师笔记1
4. python画图
5. 协同过滤代码---loadMovieLens.py文件
6. 协同过滤代码--getRating.py文件
7. Linux下安装Scala
8. Linux安装jdk
9. [转]切片
10. [转]递归函数

随笔分类(16)

[PHP\(1\)](#)[Python\(6\)](#)[笔记\(6\)](#)[翻译](#)[机器学习\(3\)](#)

随笔-16 文章-1 评论-0

增强学习贪心算法与Softmax算法

(一) ϵ -贪心算法

这个算法是基于一个概率来对探索和利用进行折中：每次尝试时，以 ϵ 的概率进行探索，即以均匀概率随机选取一个摇臂，以 $1-\epsilon$ 的概率进行利用，即以这个概率选择当前平均奖赏最高的摇臂（如有多个，则随机选取）。

论文
前端

随笔档案(16)

2016年7月 (6)
2016年6月 (6)
2016年5月 (4)

文章分类(1)

Python
机器学习(1)
其他

阅读排行榜

1. 增强学习----介绍(1661)
2. 增强学习贪心算法与Softmax算法(910)
3. 增强学习——K-摇臂赌博机(579)
4. 对MVC的理解(165)
5. python画图(140)

令 $Q(k)$ 记录摇臂 k 的平均奖赏. 若摇臂 k 被尝试了 n 次, 得到的奖赏为 v_1, v_2, \dots, v_n , 则平均奖赏为

$$Q(k) = \frac{1}{n} \sum_{i=1}^n v_i. \quad (16.1)$$

若直接根据式(16.1)计算平均奖赏, 则需记录 n 个奖赏值. 显然, 更高效的做法是对均值进行增量式计算, 即每尝试一次就立即更新 $Q(k)$. 不妨用下标来表示尝试的次数, 初始时 $Q_0(k) = 0$. 对于任意的 $n \geq 1$, 若第 $n-1$ 次尝试后的平均奖赏为 $Q_{n-1}(k)$, 则在经过第 n 次尝试获得奖赏 v_n 后, 平均奖赏应更新为

$$Q_n(k) = \frac{1}{n} ((n-1) \times Q_{n-1}(k) + v_n) \quad (16.2)$$

$$= Q_{n-1}(k) + \frac{1}{n} (v_n - Q_{n-1}(k)). \quad (16.3)$$

其中：小 k 表示第 k 个摇臂。因为大 K 表示摇臂总数； n 表示尝试的次数， v_n 表示第 n 次尝试的奖赏。

Q_{n-1}

Q_n 的直观意思为： Q_{n-1} 为前 $n-1$ 次的平均奖赏。当其与 $n-1$ 相乘所得是前 $n-1$ 次总奖赏。再加上第 n 次的奖赏，处于 n ，则为 n 次的平均奖赏。

$Q(i)$ 和 $\text{count}(i)$ 分别记录摇臂 i 的平均奖赏和选中次数。

在 $[0, 1]$ 中生成随机数。

本次尝试的奖赏值。

式(16.2)更新平均奖赏。

输入: 摇臂数 K ;
 奖赏函数 R ;
 尝试次数 T ;
 探索概率 ϵ 。

过程:

```

1:  $r = 0$ ;
2:  $\forall i = 1, 2, \dots, K : Q(i) = 0, \text{count}(i) = 0$ ;
3: for  $t = 1, 2, \dots, T$  do
4:   if  $\text{rand}() < \epsilon$  then
5:      $k =$  从  $1, 2, \dots, K$  中以均匀分布随机选取
6:   else
7:      $k = \arg \max_i Q(i)$ 
8:   end if
9:    $v = R(k)$ ;
10:   $r = r + v$ ;
11:   $Q(k) = \frac{Q(k) \times \text{count}(k) + v}{\text{count}(k) + 1}$ ;
12:   $\text{count}(k) = \text{count}(k) + 1$ ;
13: end for
输出: 累积奖赏  $r$ 

```

图 16.4 ϵ -贪心算法

其中： $\arg\max$ 为选取最优的 $Q(i)$ 。 count 是从0开始的，故 $\text{count}(k)+1$ 的值为 n ，算出的 $Q(k)$ 为 n 次的平均奖赏。

(二) Softmax算法

softmax算法是基于当前已知的摇臂平均奖赏来对探索和利用进行折中。若各摇臂的平均奖赏相当，则选取各摇臂的概率也相当；如果某些概率的平均奖赏明显高于其他奖赏，则它们被选的概率也明显高。

在贪心算法中, ϵ 的取值是由用户选取的。softmax算法中摇臂概率的分配是基于Boltzmann分布

$$P(k) = \frac{e^{\frac{Q(k)}{\tau}}}{\sum_{i=1}^K e^{\frac{Q(i)}{\tau}}},$$

<查>Boltzmann分布

其中, $Q(i)$ 记录当前摇臂的平均奖赏; $\tau > 0$ 称为“温度”, τ 越小则平均奖赏高的摇臂被选取的概率越高. τ 趋于 0 时 Softmax 将趋于“仅利用”, τ 趋于无穷大时 Softmax 则将趋于“仅探索”. Softmax 算法描述如图 16.5 所示.

输入: 摇臂数 K ;
 奖赏函数 R ;
 尝试次数 T ;
 温度参数 τ .

过程:

- 1: $r = 0$;
- 2: $\forall i = 1, 2, \dots, K : Q(i) = 0, \text{count}(i) = 0$;
- 3: **for** $t = 1, 2, \dots, T$ **do**
- 4: $k =$ 从 $1, 2, \dots, K$ 中根据式(16.4)随机选取
- 5: $v = R(k)$;
- 6: $r = r + v$;
- 7: $Q(k) = \frac{Q(k) \times \text{count}(k) + v}{\text{count}(k) + 1}$;
- 8: $\text{count}(k) = \text{count}(k) + 1$;
- 9: **end for**

输出: 累积奖赏 r

图 16.5 Softmax算法

从算法中并没有看出botlzmann分布的利用？

至于两个算法的取舍还要看实际情况。从下图看出，softmax当温度=0.01时，曲线与“仅利用”的曲线几乎重合。

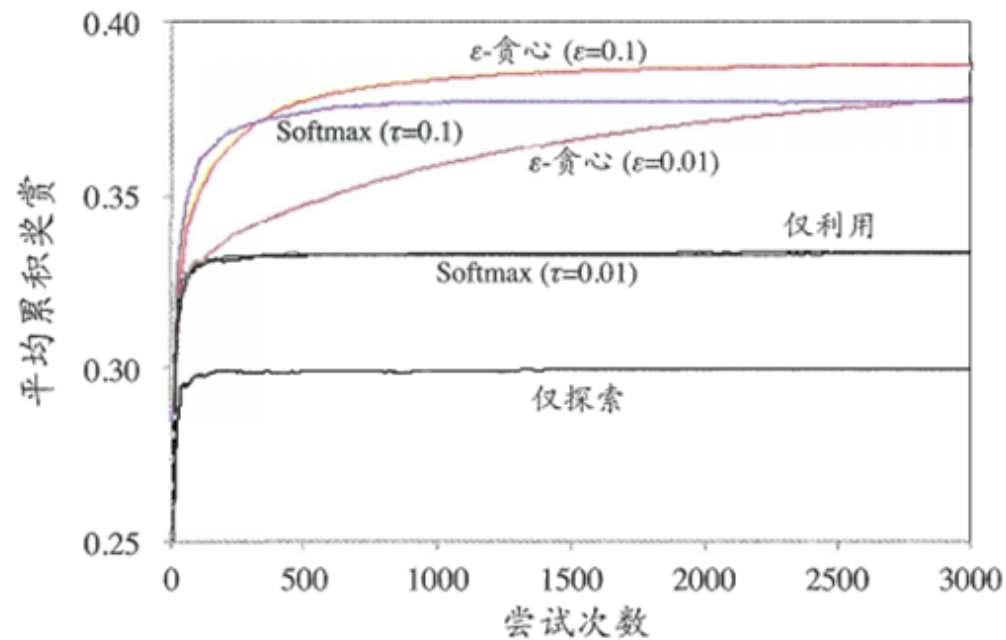


图 16.6 不同算法在 2-摇臂赌博机上的性能比较

分类: 机器学习

标签: 增强学习, 强化学习

好文要顶

关注我

收藏该文



YC_Yuan

关注 - 3

粉丝 - 0

+加关注

» 下一篇: 增强学习——K-摇臂赌博机

posted @ 2016-05-31 09:12 YC_Yuan 阅读(909) 评论(0) 编辑 收藏

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】搭建微信小程序 就选腾讯云

【推荐】报表开发有捷径：快速设计轻松集成，数据可视化和交互



最新IT新闻:

- 新浪和美国股东代理权之战升级 下月年度股东大会将见分晓
 - 乐视网：公司股票自10月17日起将继续停牌
 - 网易《我的世界》10月12日公测：全平台
 - 爱尔兰300多人举行集会 声援苹果建设全新数据中心
 - 再见了！Windows 10 Build 10586明起彻底停更
- » 更多新闻...



最新知识库文章:

- 实用VPC虚拟私有云设计原则
- 如何阅读计算机科学类的书
- Google 及其云智慧

- 做到这一点，你也可以成为优秀的程序员
- 写给立志做码农的大学生
- » [更多知识库文章...](#)