

强化学习系列之三:模型无关的策略评价

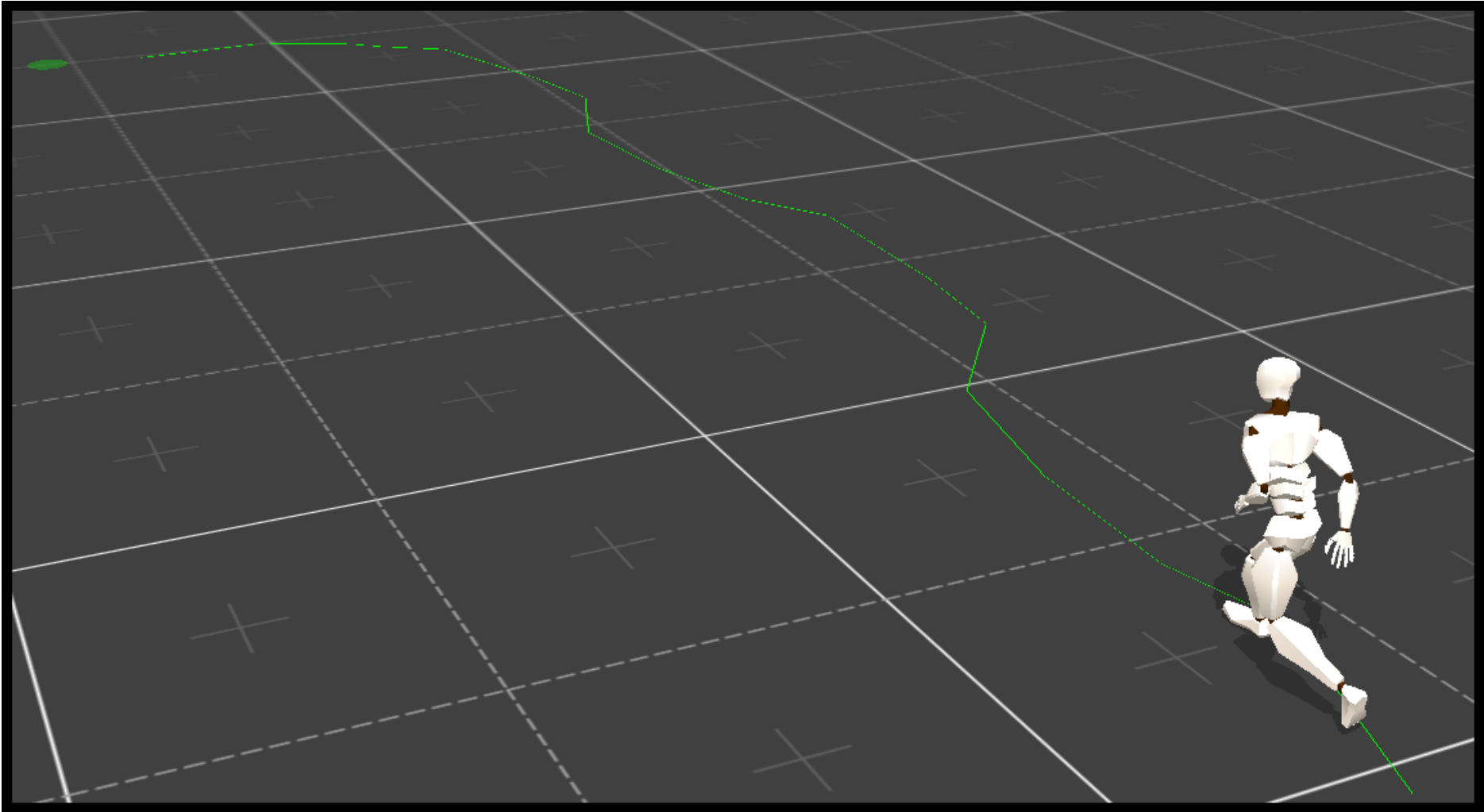
发表于2016年4月18日由lili

文章目录 [\[隐藏\]](#)

- 1. 蒙特卡罗算法
 - 2. 时差学习算法
 - 3. 一个例子
 - 4. 总结
- [强化学习系列系列文章](#)

上一章我们介绍了模型相关 (Model-based) 的强化学习。从现在开始我们要介绍模型无关 (Model-free) 的强化学习。

由于模型无关的强化学习比较复杂，今天先介绍其中一部分——模型无关的策略评价。模型无关的策略评价是，不知道马尔科夫决策过程转移概率和奖励函数的情况下，计算一个策略的每一个状态价值。模型无关的策略评价主要有两种算法，一个是蒙特卡罗算法，另一个叫时差学习算法。



1. 蒙特卡罗算法

一听到这个名字,我们就知道一个产生样本,通过样本计算状态价值的方法。首先,用当前策略探索产生一个完整的状态-动作-奖励序列。

$$s_1, a_1, r_1, \dots, s_k, a_k, r_k \sim \pi \quad (1)$$

然后,在序列第一次碰到或者每次碰到一个状态 s 时,计算其衰减奖励之后。

$$g_s = r_t + \gamma r_{t+1} + \dots + \gamma^{k-t} r_k \quad (2)$$

最后更新状态价值

$$\begin{aligned} S(s) &= S(s) + g_s \\ N(s) &= N(s) + 1 \\ v(s) &= \frac{S(s)}{N(s)} \end{aligned} \quad (3)$$

蒙特卡罗算法的代码如下所示。

```
#state_sample, action_sample, reward_sample 分别是状态、动作和奖励系列
def mc(gamma, state_sample, action_sample, reward_sample):
    vfunc = dict();
    nfunc = dict();
    for s in states:
        vfunc[s] = 0.0
        nfunc[s] = 0.0

    for iter1 in xrange(len(state_sample)):
        G = 0.0
        for step in xrange(len(state_sample[iter1])-1,-1,-1):
            G *= gamma;
            G += reward_sample[iter1][step];

        for step in xrange(len(state_sample[iter1])):
            s = state_sample[iter1][step]
            vfunc[s] += G;
            nfunc[s] += 1.0;
            G -= reward_sample[iter1][step]
            G /= gamma;

    for s in states:
        if nfunc[s] > 0.000001:
            vfunc[s] /= nfunc[s]
```

2. 时差学习算法

蒙特卡罗算法能够有效地求解模型无关的策略评估,但也存在一些问题。有时我们面临的强化学习问题是持续不断的。比如没有停止指令时,飞行器控制要求不停地根据姿势风向等因素调整,持续保持平稳飞行。这时我们得不到一个完整状态-动作-奖励系列,因此蒙特卡罗算法不适用。为了解决这个问题,人们提出了时差学习算法(Temperal Difference, TD)。时差学习算法利用马尔科夫性质,只利用了下一步信息。时差学习算法让系统按照策略指引进行探索,在探索每一步都进行状态价值的更新,更新公式如下所示。

$$v(s) = v(s) + \alpha(r + \gamma v(s') - v(s)) \quad (4)$$

s 为当前状态, s' 为下一步状态, r 是系统获得的奖励, α 是学习率, γ 是衰减因子。另外 $r + \gamma v(s')$ 被称为时差目标 (TD target), $r + \gamma v(s') - v(s)$ 被称为时差误差 $TDerror$ 。时差学习算法的代码。

```
#td 算法也可以输入状态-动作-奖励序列。
def td(alpha, gamma, state_sample, action_sample, reward_sample):
    vfunc = dict()
    for s in states:
        vfunc[s] = 0.0

    for iter1 in xrange(len(state_sample)):
        for step in xrange(len(state_sample[iter1])):
            s = state_sample[iter1][step]
            r = reward_sample[iter1][step]

            if len(state_sample[iter1]) - 1 > step:
                s1 = state_sample[iter1][step+1]
                next_v = vfunc[s1]
            else:
                next_v = 0.0;

            vfunc[s] += alpha * (r + gamma * next_v - vfunc[s]);
```

上面的时差学习算法对后续步骤不关心, 我们称这种时差学习算法为TD (0)。有时差学习算法关心后续个步骤, 我们称之为 TD(λ)。这里我们就不详细展开了, 有兴趣的同学可以看[这里](#)。

3. 一个例子

我们做了一个实验 (代码[在此](#)): 一个马尔科夫决策过程的状态集 $S=\{1,2,3,4,5,6,7,8\}$, 其中 5、6 和 8 是终止状态; 动作集合 $A=\{'n','e','s','w'\}$; 不知道奖励函数和转移概率; 衰减因子 $\gamma = 0.5$; 另外我们一千个状态-动作-奖励序列, 由一个策略探索得到。




```
1,e,0,2,e,0,3,s,1
1,s,-1
...
```

我们使用蒙特卡罗算法或者时差学习算法, 估算策略下不同状态的价值。经过计算, 我们可以得到这个策略下每个状态的价值。

```
mc result
{1: -0.321, 2: -0.002, 3: 0.306, 4: -0.011, 5: -0.357,}

td result
{1: -0.335, 2: -0.019, 3: 0.316, 4: -0.003, 5: -0.377}
```

实际上, 我们实验使用的马尔科夫随机过程是之前介绍的机器人找金币, 策略是随机选择选择一个方向。随机策略下每个状态的价值如下图所示。大体上, 蒙特卡罗算法和时差学习算法能够得到状态价值。

-0.335	-0.008	0.283	-0.008	-0.335
				

4. 总结

我们在上一章介绍模型相关的策略评估的时候，已经介绍过一种策略评估。这种模型相关的策略评估利用了贝尔曼等式，其更新公式如下所示。

$$v(s) = \sum_{a \in A} \pi(s, a) (R_{s,a} + \gamma \sum_{s' \in S} T_{s,a}^{s'} v_t(s'))$$

根据公式，这种策略评估需要知道转移概率和奖励函数。而蒙特卡罗算法和时差学习算法不知道转移概率和奖励函数。

本文介绍了模型无关的策略评价，指我们不知道马尔科夫决策过程转移概率和奖励函数的情况，计算一个策略的每一个状态价值。模型无关的策略评价主要有两种算法，一个是蒙特卡罗算法，另一个叫时差学习算法。本文代码可以在 [Github](#) 上找到，欢迎有兴趣的同学帮我挑挑毛病。强化学习系列的下一篇文章将介绍如何在不知道马尔科夫决策过程的情况下学到最优策略，敬请期待。

文章结尾欢迎关注我的公众号 AlgorithmDog，每周日的更新就会有提醒哦~



欢迎关注
公众号讲述机器学习和系统研发的轶事，
希望讲得有趣，每周日更新~
扫描二维码即可关注。您，不关注下么？

强化学习系列系列文章

- [强化学习系列之一:马尔科夫决策过程](#)
- [强化学习系列之二:模型相关的强化学习](#)
- [强化学习系列之三:模型无关的策略评价](#)
- [强化学习系列之四:模型无关的策略学习](#)
- [强化学习系列之五:价值函数近似](#)
- [强化学习系列之六:策略梯度](#)
- [强化学习系列之九:Deep Q Network \(DQN\)](#)

此条目发表在[强化学习](#), [算法荟萃](#)分类目录，贴了[强化学习](#)标签。将[固定链接](#)加入收藏夹。

《强化学习系列之三:模型无关的策略评价》有 10 条评论



percy说:
2016年4月20日下午3:32

写的很好，学习了
[回复](#)



[上微博的猫](#)说:
2016年4月21日下午6:32

感谢您的认可～
[回复](#)



jeff说:
2016年5月10日下午4:26

我发现您的结果蒙特卡洛跟随机策略下计算的状态价值差不多，但是发现使用TD学习算法却有较大的误差，您有没有看到这
一点问题呢？
[回复](#)



[上微博的猫](#)说:
2016年5月11日下午8:56

恩恩，我当时注意到这个问题了，百思不得其解。
[回复](#)



jeff说:
2016年5月13日下午4:49

今天导师总结了一下MC和TD算法的一些利弊，TD算法虽然说是一种在线学习，但是它对状态初始价值更敏
感。而且对于您示例这样简单问题MC确实可以通过生成序列样本就可以获得很好的结果，可能对于复杂问题
来说就不同了吧。个人见解～
[回复](#)



[上微博的猫](#)说:
2016年5月13日下午11:24

恩恩，有可能，策略评价 MC 优于 TD。感谢～。
[回复](#)



Wolfgang说:
2016年7月11日上午10:41

博主，想请问下代码上，TD 的初始v可以设置成0吧
[回复](#)



[上微博的猫](#)说:
2016年7月16日下午8:56

感觉可以～，或者你可以试着设置成0，然后跑跑
[回复](#)



海格力斯说:
2016年11月1日上午10:56

博主，您好，想请教下，关于TD算法，在David Silver的习题答案中http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/exam-rl-answers.pdf

第2题，有A，B两个状态，已知两个事件： $A+3 \rightarrow A+2 \rightarrow B-4 \rightarrow A+4 \rightarrow B-3 \rightarrow \text{terminate}$ 和 $B-2 \rightarrow A+3 \rightarrow B-3 \rightarrow \text{terminate}$ ，应用TD(0)算法答案得出 $V(A)=2,V(B)=-2$ 是怎样得到的？我计算出来是 $V(A)=3,V(B)=-3$,用您的程序计算出来也是3,-3，求讲解。

[回复](#)



joyxiang 说:
2016年11月9日下午4:32

写的很好，学习了，还有个问题请教下。例子中还是使用了奖励函数，怎么处理奖励函数未知情况呢？谢谢。

[回复](#)