深度之眼
deepshare.net

强化学习专题:蒙特卡洛方法

导师：Alex

目录

深度之眼
deepshare.net

探索与应用

Exploration and Exploitation

探索与应用

Exploration and Exploitation

之前策略迭代的问题?

无法实现!

- 我们不知道$p(s', r|s, a)$
- 我们需要考虑很多实际过程中永远不会考虑的state

需要探索$p(s', r|s, a)$，但也要考虑探索的成本

探索与应用

Exploration and Exploitation

- Exploration:
  - 学习环境中的未知因素
  - state的transition
  - reward的distribution

- Exploitation
  - 根据已经积累，学习的经验，选择已知的优质action

- 强化学习需要考虑两者的平衡，注重于提升效率和稳定性

深度之眼
deepshare.net

蒙特卡洛方法

Monte Carlo Method

蒙特卡洛方法

Monte Carlo Method

本部分是重要的基础，包括:

- 更新$v_\pi(s)$和$q_\pi(s,a)$的基本方法和思想
- 部分理论分析
- 伪代码
- 例题

主要方法包括:

- 蒙特卡洛方法(Monte Carlo, MC)
- 时差方法(Temporal Difference, TD)

蒙特卡洛方法

Mento Carlo Method

Mento Carlo指基于根据随机的实际经历学习，推断的方法

思考：

根据一个随机轨迹(trajectory) $s_0, a_0, r_1, \dots, s_{T-1}, a_{T-1}, r_T, s_T$我们能对$v_\pi(s)$和$q_\pi(s,a)$做出的最好的估计是什么？

蒙特卡洛方法

Mento Carlo Method

- Monte Carlo: 用很多随机实现来估计一个值
- 什么是$v_\pi(s_0)$和$q_\pi(s_0,a_0)$的随机实现？

$$\hat{q}_\pi(s_0,a_0)=\hat{v}_\pi(s_0)=\sum_{t=1}^{T}\gamma^{t-1}r_t=r_1+\gamma r_2+\gamma^2 r_3+\cdots+\gamma^{T-1}r_T$$

- 我们还有什么的随机实现？

$$\hat{q}_\pi(s_i,a_i)=\hat{v}_\pi(s_i)=\sum_{t=i+1}^{T}\gamma^{t-i-1}r_t=r_{i+1}+\gamma r_{i+2}+\gamma^2 r_{i+3}+\cdots+\gamma^{T-i-1}r_T$$

蒙特卡洛方法

Mento Carlo Method

- 一次随机样本随机性太大，方差太大

- 使用多个trajectory的信息，计算平均值，能使方差变小

- 如何使用大量的随机轨迹的平均值？
  - 直接取平均
  - Incremental method

蒙特卡洛方法

Mento Carlo Method

Incremental Method:

假设原估计值是$\hat{v}$，新的估计值是v_{new}

$$\hat{v} \longrightarrow \hat{v} + \alpha(v_{new} - \hat{v}) = \alpha v_{new} + (1 - \alpha)\hat{v}$$

α可以理解为学习率

```
###### python 3.6.8    numpy  1.16.4
1.import numpy as np
2.
3.GAMMA = # usually 0.99, 0.995
4.STATE_SIZE = # depends on your environment
5.ACTION_SIZE = # depends on your environment
6.
7.def get_Q(experience, gamma):
8.    temp_Q = np.zeros((STATE_SIZE, ACTION_SIZE))
9.    G = 0
10.    T = len(experience)
11.    for t in range(T-1, -1, -1): # learn backward
12.        state, action, reward = experience[t]
13.        G = gamma*G + reward
14.        temp_Q[state, action] = G
15.    return temp_Q
16.
17.def update_Q(old_Q, experience, lr):
18.    old_Q += lr * (get_Q(experience, GAMMA) - old_Q)
19.    return old_Q
```

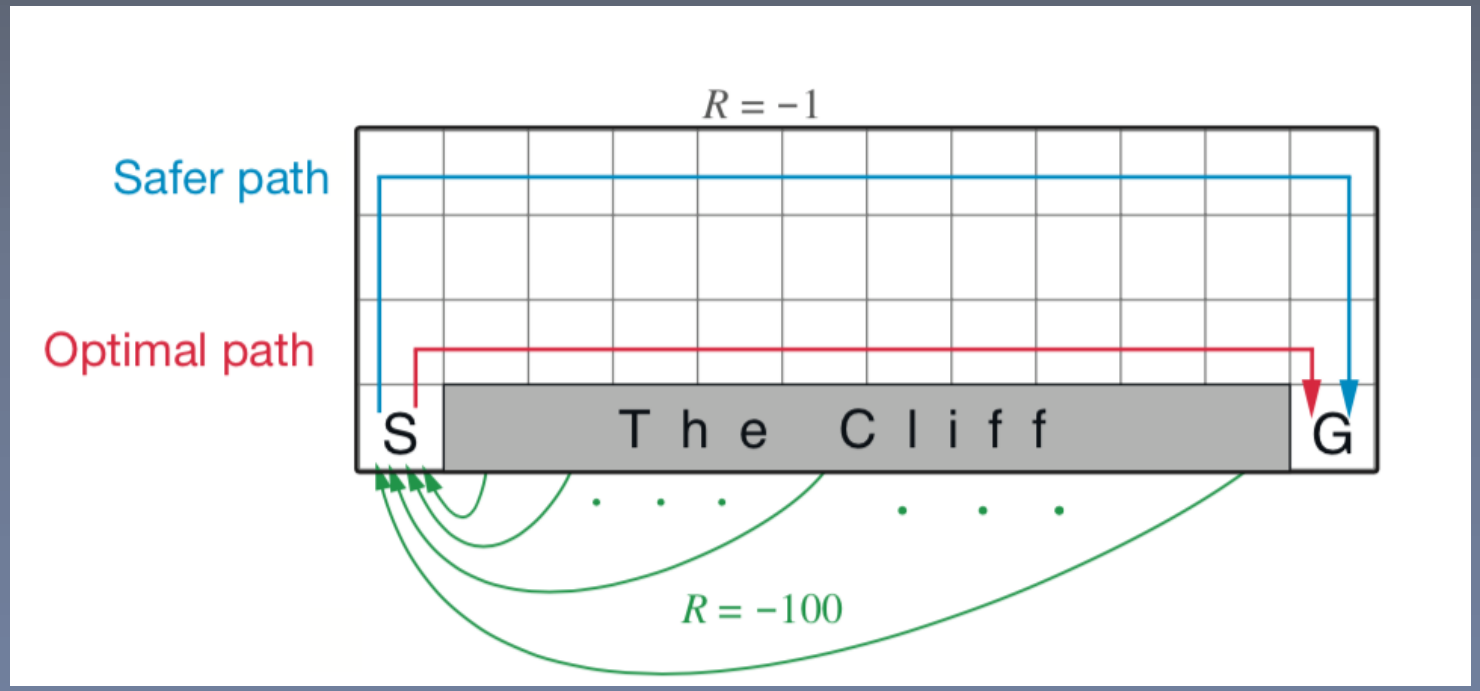
```
1.Q = np.zeros((STATE_SIZE, ACTION_SIZE))
2.
3.# assume we have the experience with tuples (s, a, r)
4.# experiences = [experience1, experience2, experience3, ...]
5.# experience should be [(s0, a0, r1), (s1, a1, r2), ... (s_{
T-1}, a_{T-1}, r_T)]
6.# Think about how to collect experiences
7.
8.for experience in experiences:
9.    Q = update_Q(Q, experience, 0.01)
10.
11.pi = {}
12.for i in range(STATE_SIZE):
13.    pi.update({i : np.argmax(Q[i, :])})
```

蒙特卡洛方法

Mento Carlo Method

环境介绍:

Cliff walking



- 起点为左下角
- 终点为右下角
- 四个行动：上下左右
- 每一步reward为-1
- 掉落悬崖reward为-100
- 每一步有一定概率无法人为选择行动，随机从四个行动中选择一个

蒙特卡洛方法

Mento Carlo Method

练习：

大家自己使用这种方法，模拟100000个episode，计算blackjack的最优策略

- 背景：每次游戏牌库的初始状态都是完整的一副牌，也只有玩家和庄家两个
- State：信息是二维的，就是庄家所持的牌，以及自己手中的牌点数之和
- Action：可以再加一张牌，或者不再加牌，和庄家比大小，或直接认输
- Reward：胜为1，负为-1，平为0，认输为-0.5
- $\gamma = 1$

深度之眼
deepshare.net

策略迭代

Policy Iteration

策略迭代

Policy Iteration

根据蒙特卡洛方法，我们可以在不知道$p(s', r|s, a)$的情况下进行策略迭代：

- 初始化所有的Q (s, a)，例如，全部设置为0，或者随机初始化
- 不断循环以下的步骤：
  - 更新策略$\pi(s) = \underset{a}{\operatorname{argmax}}\, Q(s, a)$，有相等则随机选择
  - 以下步骤进行多次循环以收集数据：
    - 获得初始状态s_0
    - 根据π，得到trajectory
  - 根据获得的数据，更新$Q(s, a)$
  - 如果收敛，或者循环到达一定次数，结束循环

策略迭代

Policy Iteration

之前的方法有一个严重的问题：

- 陷入局部最优，即无法寻找潜在的更好的策略，缺少exploration

解决思路：

- 不用greedy的方法更新策略，而是$\varepsilon - \mathrm{Greedy}$，强制要求exploration
- 每次行动，有$1-\varepsilon$的概率选择当前最优行动
- 剩下的ε概率随机选择一个行动

新的问题：对$v_\pi(s)$和$q_\pi(s,a)$的估计不再是最优的

我们在用另一个策略$b(a|s)$采样，估计与$\pi(a|s)$相关的值

深度之眼
deepshare.net

重要性采样

Importance Sampling

重要性采样

Importance Sampling

介绍两个概念：

- on-policy learning：从当前策略的trajectory中学习
  - 稳定，方差小，收敛快
  - 需要不断根据当前策略去实验，以获得数据
  - 需要大量的数据，因为数据使用过就会被抛弃
- off-policy learning：从其他策略的trajectory中学习
  - 不稳定，方差大，收敛慢
  - 更数据的使用更有效率
  - 可以使用任何来源的数据

重要性采样

Importance Sampling

使用$\varepsilon-\mathrm{Greedy}$寻找最优策略，是off-policy learning的一种，可以使用importance sampling修正估计值

importance sampling：从一个分部采样，对另一个分布进行估计

$$E_F(X)=\int xf(x)\mathrm{d}x=\int x\frac{f(x)}{g(x)}g(x)\mathrm{d}x=E_G[x\frac{f(x)}{g(x)}]$$

重要性采样

Importance Sampling

假如我们要估计的策略是$\pi(a|s)$，而实际采样使用的策略是$b(a|s)$

$$v_b(s) = E_b[G_t|S_t = s]$$

$$v_\pi(s) = E_b[\rho_{t:T-1}G_t|S_t = s], \quad \rho_{t:T-1} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

结 语

本次课程中，我们开始接触如何在不了解环境的情况下进行更新价值函数并进行策略迭代

下次课程中，我们将学习另一种更新价值函数的方法：Temporal Difference，并比较两者的不同

deepshare.net
深度之眼

联系我们:

电话: 18001992849

邮箱: service@deepshare.net

QQ: 2677693114

公众号

客服微信