实验一

潘李凡

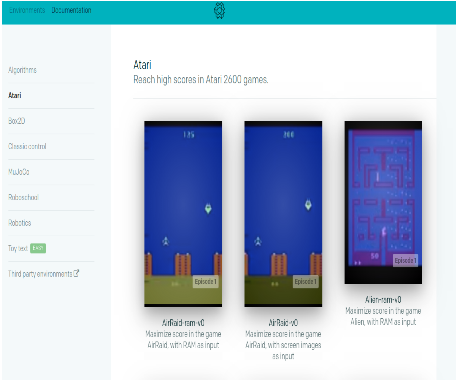
USTC
lifanpan@mail.ustc.edu.cn

2019 年 10 月 16 日

Table of Contents

介绍 gym

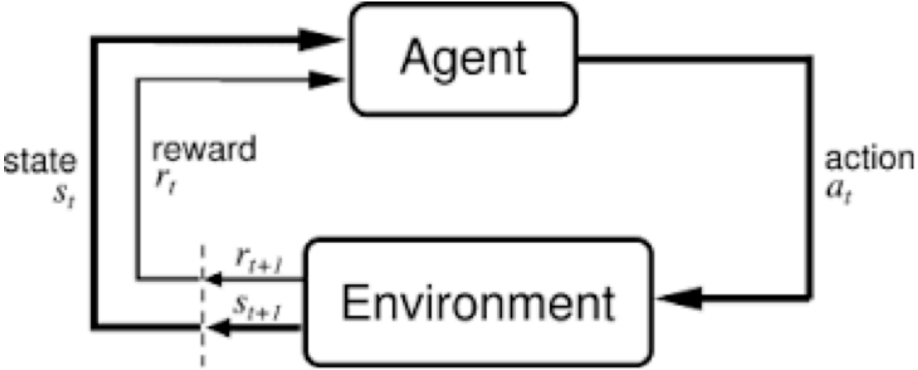
- https://gym.openai.com/



ubuntu16.04 安装 gym

- python 3.5.2 安装 gym
  sudo pip3 install gym matplotlib==2.0 pandas==0.23.0
- 其它
  sudo apt-get install python3-tk
- 如果上面报错呢？一般原因是少了一些必要的模块
  sudo apt-get install golang python3-dev python-dev libcupti-dev libjpeg-turbo8-dev make tmux htop chromium-browser git cmake zlib1g-dev libjpeg-dev xvfb libav-tools xorg-dev python-opengl libboost-all-dev libsdl2-dev swig

如何使用 gym 环境



如何使用 gym 环境

- 创建一个 environment
  - $env = gym.make(game\_name)$
- 初始状态 s_0
  - $s_0 = env.reset()$
- agent 和 environment 之间的交互
  - $next\_state, reward, done, \_ = env.step(action)$

如何使用 gym 环境

- 显示画面
  - *env.render*()
- spaces
  - *env.action_space*
  - *env.observation_space*

Table of Contents

black jack 环境

- reinforcement learning,sutton p_{74}
- black jack 也叫 21 点游戏
  - 目标：游戏者的目标是使手中的牌的点数之和不超过 21 点且尽量大，本次实验只有你和庄家玩
  - action： 叫牌 (hit) 和停止叫牌 (stick)
  - 计算：2 至 9 牌，按其原点数计算；K、Q、J 和 10 牌都算作 10 点；A 牌既可算作 1 点也可算作 11 点

black jack 环境

- 浏览环境

```
plf@plf-pc:~/Desktop/exp1/exp1/assignment1/mc$ python3 BlackjackEnv.py
Player Score: 17 (Usable Ace: False), Dealer Score: 1
Taking action: Hit
Player Score: 20 (Usable Ace: False), Dealer Score: 1
Taking action: Stick
Player Score: 20 (Usable Ace: False), Dealer Score: 1
Game end. Reward: 1.0
```

mc 算法

- 在 mc.py 中实现函数 mc(env, num_episodes, discount_factor, epsilon)

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\quad\pi \leftarrow$ an arbitrary ε-soft policy

$\quad Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$\quad Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

$\quad$Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$\quad G \leftarrow 0$

$\quad$Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$\quad\quad G \leftarrow G + R_{t+1}$

$\quad\quad$Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

$\quad\quad\quad$Append G to $Returns(S_t, A_t)$

$\quad\quad\quad Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

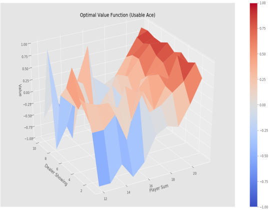
$\quad\quad\quad A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

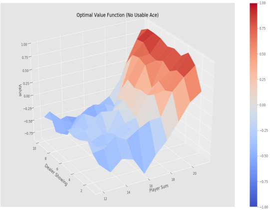
$\quad\quad\quad$For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

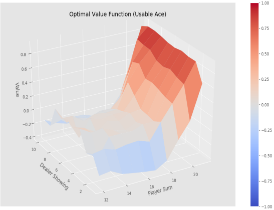
mc 算法结果

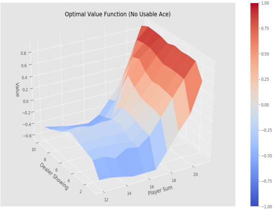
- 10000 episodes





- 500000 episodes



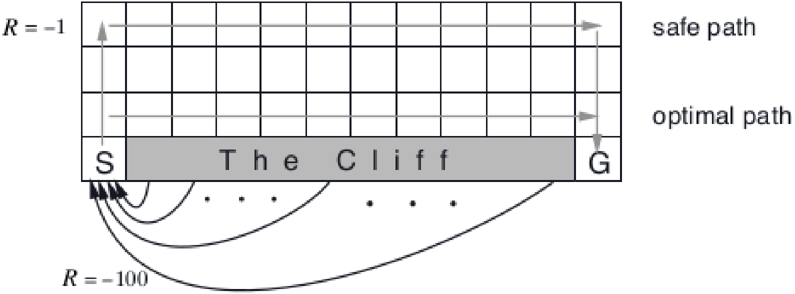


可选做

- 额外加分
  实现 every-visit 版本，并且对比 first-visit 和 every-visit

Table of Contents

cliff walk 环境



cliff walk 环境

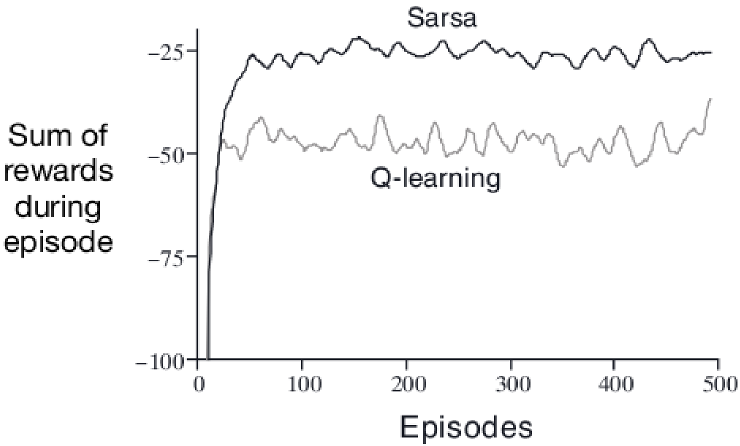
- 运行 td 文件下的 cliff_walk.py 浏览环境

```
plf@plf-pc:~/Desktop/exp1/exp1/assignment1/td$ python3 cliff_walk.py
36
o   o   o   o   o   o   o   o   o   o   o   o
o   o   o   o   o   o   o   o   o   o   o   o
o   o   o   o   o   o   o   o   o   o   o   o
x   C   C   C   C   C   C   C   C   C   C   T

(24, -1.0, False, {'prob': 1.0})
o   o   o   o   o   o   o   o   o   o   o   o
o   o   o   o   o   o   o   o   o   o   o   o
x   o   o   o   o   o   o   o   o   o   o   o
o   C   C   C   C   C   C   C   C   C   C   T
```

reward

- reinforcement learning,sutton p_{106}



sarsa 算法

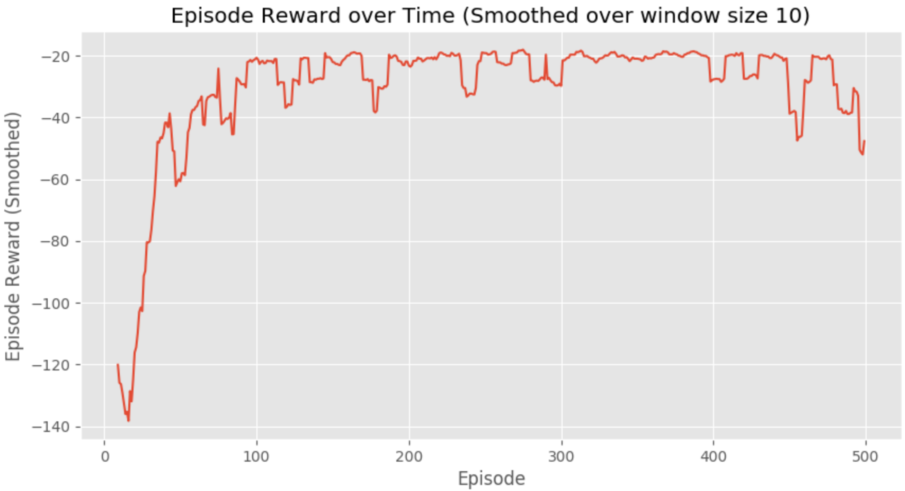
- 在 sarsa.py 中实现函数 sarsa(env, num_episodes, discount_factor, alpha, epsilon)

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

sarsa 算法


Episode Reward over Time (Smoothed over window size 10)
Episode Reward (Smoothed)
−20
−40
−60
−80
−100
−120
−140
0
100
200
300
400
500
Episode

q learning 算法

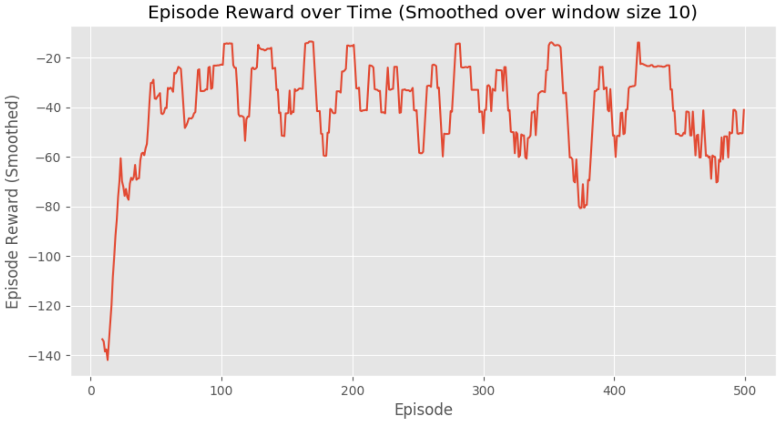
- 在 qlearning.py 中实现函数 q_learning(env, num_episodes, discount_factor, alpha, epsilon)

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\textit{terminal}, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using policy derived from Q (e.g., ϵ-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until S is terminal

qlearning 算法



可选做

- 额外加分
  实现 double q-learning, 对比 double q-learning 和 q-learning

Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$
Initialize $Q_1(s,a)$ and $Q_2(s,a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using the policy ε-greedy in $Q_1 + Q_2$
    Take action A, observe R, S'
    With 0.5 probabilility:
      $Q_1(S,A) \leftarrow Q_1(S,A) + \alpha\Big(R + \gamma Q_2\big(S', \operatorname{argmax}_a Q_1(S',a)\big) - Q_1(S,A)\Big)$
    else:
      $Q_2(S,A) \leftarrow Q_2(S,A) + \alpha\Big(R + \gamma Q_1\big(S', \operatorname{argmax}_a Q_2(S',a)\big) - Q_2(S,A)\Big)$
    $S \leftarrow S'$
  until S is terminal

提交要求

- 压缩文件命令格式：实验一
  - 源码
  - 实验结果，保存到 result 文件夹下面
- 截止日期：2019.10.31
- 提交地址：http://xzc.cn/Z55I5LPOxQ

联系

- 助教邮箱：lifanpan@mail.ustc.edu.cn
- 课程微信群组