

소프트웨어특강2 : 강화 학습 개론

Assignment 2: Model-Free Reinforcement Learning

1. 목표

- Open AI Gym Taxi-v2 환경에서의 Model-Free RL 방법으로 문제 해결 하기
* Taxi-v3 환경으로 진행해도 무방합니다.

2. 개발 환경

- Python (version 3.5+)

3. 템플릿 파일 및 제출 파일

- 제공되어지는 템플릿 파일 2개
 - taxi.py: Taxi 환경 설정 및 기초 템플릿 코드 (제출하지 않음)
 - agent.py: q-learning 구현이 들어가야 할 프로그램 (완성 후 제출용)

4. Open AI Gym Taxi-v2 환경 소개

* 먼저 Open AI Gym 을 설치 합니다.

"There are 4 locations (labeled by different letters), and our job is to pick up the passenger at one location and drop him off at another. We receive +20 points for a successful drop-off and lose 1 point for every time-step it takes. There is also a -10 point as the penalty for illegal pick-up and drop-off actions."

https://github.com/openai/gym/blob/master/gym/envs/toy_text/taxi.py

```

+-----+
|R: | : :G|
| : : : : |
| : ■ : : : |
| | : | : |
|Y| : |B: |
+-----+

```

- 5x5 Map

```
env = gym.make('Taxi-v2')
```

```
action_size = env.action_space.n
```

```
space_size = env.action_space.n
```

- State Space = $5 \times 5 \times 5 \times 4 = 500$.

State 는 taxi locations (5×5), 5 possible passenger locations (4 corresponding to RGBY with 1 additional of being in the taxi), and 4 possible destinations (corresponding to RGBY).

- The Action Space = 6

0. down, 1. Up, 2. Right, 3. Left, 4. Pickup, 5. Dropoff

- 위와 같이 환경이 reset() 되면, R, G, Y, B 지점에 승객이 기다리고, 상하좌우로 움직여서 해당 위치로 이동한 후 pickup 한 후 목적지로 가서 dropoff 하면 20 점 보상 발생.

- next_state, reward, done, _ = env.step(action)

reward 는 -1, -10, 또는 20

done 은 올바른 목적지에 승객을 내려주어 20 점 보장을 받을 때 True,

또는 200 번 이상 action 을 수행하면 True, 그 이외는 False.

5. taxi.py 설명

*** taxi.py 파일은 수정하지 않으며 제출하지도 않습니다.**

해당 파일을 실행하면 아래와 action space 크기와, state space 크기를 출력한 후 메뉴가 출력됨.

```
1. Testing without learning
2. MC-Control
3. Q-learning
4. Testing after learning
5. Exit
select:
```

그림 1. 실행 옵션 설정

1 번은 학습과 무관하게 맵을 이동하고 승객을 태우고 내리는 행동을 수동으로 해볼 수 있음.

2 번은 Monte Carlo control 방식으로 학습 후 최근 100 번 에피소드의 평균 reward 가 **6.0** 이상이 되도록 함.

3 번은 Q_learning 방식으로 학습 후 최근 100 번 에피소드의 평균 reward 가 **8.0** 이상이 되도록 함.

4 번은 학습 하기 전에는 음수 값이 출력됨. MC-control 또는 Q-learning 학습으로 업데이트된 q-table 을 활용하여 테스트 평균 reward 로 각각 **6.0** 또는 **8.0** 이상 나오는지 확인.

6. agent.py 설명

*** agent.py 파일명 그대로 제출합니다.**

```
class Agent:
```

```
    def __init__(self, Q, mode="mc_control", nA=6, alpha = 0.1, gamma = 0.99):
```

생성자로 넘겨받는 Q 변수가 q-table 에 해당함. mode 로 넘겨받는 값이 “mc_control” 또는 “q_learning” 에 따라 적절한 학습 방법을 수행하도록 개발함.

템플릿 코드는 자유롭게 수정할 수 있음. 단 taxi.py 파일 수정 없이 연동되어야 함.

7. 제출 관련

- 제출 마감일: 11.17 (일) 23:59.
 - 지연 제출은 24시간마다 -15 점, 72시간 초과는 받지 않습니다.
 - 표절은 양쪽 모두 0점 처리 되며, 두 번 이상 반복 시에는 F 입니다.
- "agent.py" 파일명을 그대로 사용하고 해당 파일 1개만 iCampus 로 제출합니다.

8. 채점 기준

- Total 100 points
 - 40 points: 2. mc-control 학습 및 최종 100회 평균 reward 가 6.0 이상 출력.
 - 10 points: 3. Testing after learning 실행하여 mc-control 로 업데이트 된 Q-table 로 테스트 평균 reward 가 6.0 이상 출력.
 - 40 points: 2. q-learning 학습 및 최종 100회 평균 reward 가 8.0 이상 출력.
 - 10 points: 3. Testing after learning 실행하여 q-learning 으로 업데이트 된 Q-table 로 테스트 평균 reward 가 8.0 이상 출력.