Lecture D1. Markov Reword Process1

Reinforcement Learning Study

2021-01-13

차 례

Recap

```
import numpy as np

def soda_simul(this_state):
    u= np.random.uniform()

    if(this_state=="c"):
        if(u<=0.7):
            next_state="c"
        else:
            next_state="p"

    else:
        if(u<=0.5):
            next_state="c"
        else:
            next_state="p"

    return(next_state)


def cost_eval(path):
    cost_one_path=path.count('c')*1.5+path.count('p')*1
```

```
    return cost_one_path
```

MC simulation for estimating state-value function

```
# MC evaluation for state-value function

#with state s, time 0, reward r, time-horizon H


def MC_V_t(initial_state, num_episode, time_horizon):

    episode_i = 0

    cum_sum_G_i = 0

    while(episode_i<num_episode) :
        path=initial_state
        for n in range(time_horizon-1):
            this_state=path[-1]
            next_state=soda_simul(this_state)
            path+=next_state

        G_i=cost_eval(path)
        cum_sum_G_i+=G_i
        episode_i+=1
    V_t=cum_sum_G_i/num_episode
    return V_t


print(MC_V_t('c',100000,10))
```

```
## 13.358845
```

```
print(MC_V_t('s',100000,10))
```

```
## 11.734395
```

iterative Solution Excercise

For general t,

$$
\begin{aligned}
V_t(s) &= \mathbb{E}[G_t|S_t = t] \\
&= \mathbb{E}[r_t + r_{t+1} + r_{t+2} \cdots + r_k|S_t = s] \\
&= \mathbb{E}[r_t|S_t] + \mathbb{E}[r_{t+1} + r_{t+2} \cdots + r_k|S_t = s] \\
&= R(s) + \mathbb{E}[r_{t+1} + r_{t+2} \cdots + r_k|S_t = s] \\
&= R(s) + \mathbb{E}[G_{t+1}|S_t = s, S_{t+1} = s'] \\
&= R(s) + \mathbb{E}[G_{t+1}|S_{t+1} = s'](\because Markovproperty) \\
&= R(s) + \sum_{s \in s'} P_{ss'} V_{t+1}(s')
\end{aligned}
$$

Backward induction for estimating state-value function

```
import numpy as np

P=np.array([[0.7,0.3],[0.5,0.5]])

R=np.array([1.5,1])[:,None] # [:,None] retrun Column vector

H=10

v_t1=np.array([0,0])[:,None]
```

```
print('P :\n',P)
```

```
## P :
##  [[0.7 0.3]
##  [0.5 0.5]]
```

```
print('R :\n',R)
```

```
## R :
##  [[1.5]
##  [1. ]]
```

```
print('v_t1 :\n',v_t1)
```

```
## v_t1 :
##  [[0]
##  [0]]
```

```
t=H-1

while(t>=0):
    v_t = R+np.dot(P,v_t1)
    t = t-1
    v_t1 = v_t
v_t
```

```
## array([[13.35937498],
##        [12.73437504]])
```

```
"Done, Lecture D1. Markov Reword Process1 "
```

```
## [1] "Done, Lecture D1. Markov Reword Process1 "
```