D1_손민상

Son Min Sang

2021-01-13

차 례

page 10

```
import numpy as np

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

    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_N=10000
spending_records=np.zeros((MC_N,))
for i in range(MC_N):
    path='c' # coke today (day-0)

    for t in range(9):
        this_state=path[-1]
        next_state=soda_simul(this_state)
        path+=next_state

    spending_records[i]=cost_eval(path)

print(spending_records)
```

```
## [12.  14.5 15.  ... 14.  14.  13.5]
```

page 11

```
#MC evalutaion for state-value function
#with state s, time 0, reward r, time horizon H

num_episode = 10000
episode_i = 0
cum_sum_G_i = 0
while episode_i < num_episode:
  path = 's'

  for t in range(9):
    this_state = path[-1]
    next_state = soda_simul(this_state)
    path = path+next_state

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

print(V_t)
```

```
## 11.7217
```

page 17

For general t,

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

page 20

```
import numpy as np
P=np.array([[0.7,0.3],[0.5,0.5]])
R=np.array([[1.5],[1.0]])
H=10 #time-horizon
v_t1 = np.array([[0],[0]]) # v_{t+1}

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

```
## [[13.35937498]
##  [12.73437504]]
```