D1_Jeong,wonryeol

Jeong, wonryeol

2021-01-13

Contents

10_page

```
MC_N = 10000

spending_records = np.repeat(0 , MC_N)

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

  spending_records[i] = cost_eval(path)


np.mean(spending_records)
```

```
## 13.1087
```

11_page

while episode_i < num_episode

state-value-fn V_t(s) = cum_sum_G_i / num_episode

return V_t(s)

```
#MC evalutaion for state-value function

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

num_episode = 1000
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


V_t
```

```
## 11.7295
```

Ex7

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

20_page

```
P =  np.array([[0.7,0.3],[0.5,0.5]])
R =  np.array([[1.5,1.0]]).reshape(2,1)


H = 10 # time-horizon
v_t1 = np.array([0,0]).reshape(2,1)
v_t = np.array([[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]])
```

21__page

```python
#Backward induction for state-value function
#with transition prob mat P , reward vector R, time-horizon H, state-value vector v

def backward_induction(P,R,H):
  v = np.zeros(H) # zero column vector
  t = H-1
  v_t = np.array()
  while t >= 0 :

    v[t] = R+ np.dot(P,v[t+1])
    t = t-1

  return v
```