Inotes2 Mars Rover Python

Jaemin Park

2021-01-13

차 례

Mars Rover Markov Chain **2**

Monte-Carlo simulation 2

Iterative solution 3

Mars Rover Markov Chain

Monte-Carlo simulation

```
MC_N = 10000
H=10
def mars_simul(state):
    u=np.random.rand()
    #print(u)
    i = state
    if(i==1 and u<=0.4):
        i=2
    elif(i==7 and u<=0.4):
        i=6
    elif(i>1 and i<7):
        if(u<=0.4):
            i-=1
        elif(u<=0.8):
            i+=1
    return i

def reward_calculate(result):
    reward = result.count(1)*1 + result.count(7)*10
    return reward

reward = np.array([])
result = []
start_state = 4
for i in range(MC_N):
    state = start_state
    for j in range(H):
        result.append(state)
        state = mars_simul(state)

    reward=np.append(reward, reward_calculate(result))
    result=[]

print(np.mean(reward))
```

```
## 8.1605
```

Iterative solution

```
P = np.matrix([[0.6,0.4,0,0,0,0,0],[0.4,0.2,0.4,0,0,0,0],[0,0.4,0.2,0.4,0,0,0],
[0,0,0.4,0.2,0.4,0,0],[0,0,0,0.4,0.2,0.4,0],[0,0,0,0,0.4,0.2,0.4],[0,0,0,0,0,0.4,0.6]])
R = np.matrix([1,0,0,0,0,0,10]).reshape(7,1)
H=10
v_t1 = np.matrix([0,0,0,0,0,0,0]).reshape(7,1)

t=H-1
while(t>=0):
    v_t = R + P*v_t1
    t-=1
    v_t1 = v_t

print(v_t)
```

```
## [[ 5.06485555]
##  [ 3.99416218]
##  [ 4.74779648]
##  [ 8.11882086]
##  [15.21931059]
##  [27.31909325]
##  [45.53596109]]
```

$\therefore S(4) = 8.1188$