Inotes2 - Mars Rover

Kang, Eui Hyeon

2021-01-13

차 례

3.1.1 Example of a Markov process : Mars Rover

- State Space : $S = \{S1, S2, S3, S4, S5, S6, S7\}$
- Transition Probabilities of the states

$$P = \begin{pmatrix} 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.2 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.2 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.2 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 \end{pmatrix}$$

- $recurrent$: $\{S1, S2, S3, S4, S5, S6, S7\}$
- $transient$: $\{\}$
- $absorb$: $\{\}$

3.2.4 Example of a Markov reward process : Mars Rover

```
import numpy as np

def mars_rover(state):
    prob=np.random.uniform(0,1)

    if state=='S1':
        if prob<=0.6:
            return 'S1'
        else:
            return 'S2'
    elif state=='S7':
        if prob<=0.6:
            return 'S7'
        else:
            return 'S6'
    else:
        if prob<=0.2:
            return state
        elif 0.2<prob and prob<=0.6:
            return 'S'+str(int(state[-1])-1)
        else:
            return 'S'+str(int(state[-1])+1)



def count_reward(path):
    size=len(path)
    value=0
    for i in range(0,size,2):
        if path[i:i+2][-1]=='1':
            value+=1
        elif path[i:i+2][-1]=='7':
            value+=10
        else:
            value+=0

    return value


reward_result=list()
```

```
num=10000

for _ in range(num):
    H=10
    path='S4'

    for _ in range(H):
        this_state=path[-2:]
        next_state=mars_rover(this_state)
        path+=next_state

    reward=count_reward(path)
    reward_result.append(reward)

print(np.mean(reward_result))
```

```
## 9.1413
```