D3 - Exercise

Kang, Eui Hyeon

2021-01-20

차 례

Exercise 1

How would you generalize this game with arbitrary value of m_1 (minimum increment), m_2 (maximum increment), and N (the winning number)?

- State Space : $S = \{1, 2, 3...N\}$
- Action : $A = (m_1, m_2)$
- Reward : $R(N - m_1, m_1) = R(N - m_2, m_2) = 1$ and other $R(s, a) = 0$.
- Optimal Policy : $\pi^*(s) = N - k(m_1 + m_2)$, k=natural number

Exercise 2

Two players are to play a gam.e The two players take turns to call out integers. The rules are as follows. Describe A's winning strategy.

- A must call out an integer between 4 and 8, inclusive.
- B must call out a number by adding A's last number and an integer between 5 and 9, inclusive.
- A must call out a number by adding B's last number and an integer between 2 and 6, inclusive.
- Keep playing until the number larger than or equal to 100 is called by the winner of this game.

…

Exercise 3

There is only finite number of deterministic stationary policiy. How many is it? \

$|\prod| = |A|^{|S|}$

Exercise 4

Formulate the first example in this lecture note using the terminology including state, action, reward, policy, transition. Describe the optimal policy using the terminology as well.

- State Space : $S = \{1, 2, 3, ...31\}$
- Action : $A = (a_1, a_2)$
  - a_{1} = increments by 1
  - a_{2} = increments by 2
- Reward : $R(30, a_1) = R(29, a_2) = 1$ and all other $R(s, a) = 0$.
- Transition : $P^a_{ss'} = \begin{cases} 1, & s' = s + a_m, m \in [1, 2] \\ 0, & otherwise \end{cases}$
- Optimal Policy : $\pi^* = \begin{cases} a_1, & s = 3n \\ a_2, & s = 3n - 1 \end{cases}$, n=natural number

Exercise 5

From the first example,

- Assume that your opponent increments by 1 with prob. 0.5 and by 2 with prob. 0.5.
- Assume that the winning number is 10 instead of 31.
- Your opponent played first and she called out 1.
- Your current a policy π_0 is that
  - If the current state s<=5 then increments by 2.
  - If the current state s>5 then increments by 1.

Evaluate $V^{\pi_0}(1)$

```
import numpy as np

state=1

user=1
while state<=10:
    user*=-1

    if state<=5:
        state+=2
    else:
        state+=1

    if state>=10:
        break

    user*=-1
    prob=np.random.uniform(0,1)

    if prob<0.5:
        state+=1

    else:
        state+=2

    if state>=10:
        break

if user==1:
    print('winner : opponent')
```

```
else:
    print('winner : me')
```

```
## winner : me
```