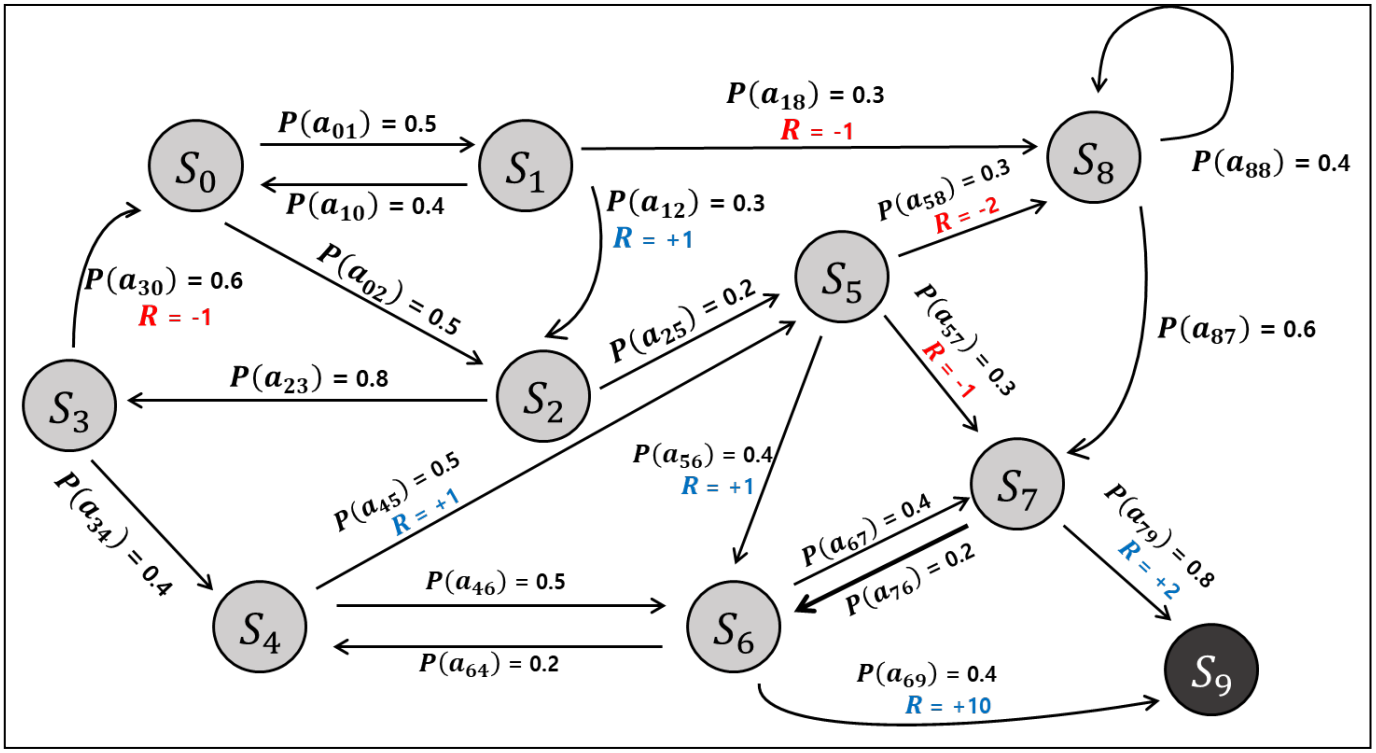
지능시스템

과제2 - 정책 이터레이션

2019305059

이현수

● 다음의 State diagram으로 나타난 MDP에 대해 정책 이터레이션을 이용하여 각 상태의 가치함수 및 최적 정책을 구하는 프로그램을 작성하고 결과를 제시하시오.



정책 이터레이션 알고리즘

Policy iteration (using iterative policy evaluation)

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
      $\Delta \leftarrow 0$
      For each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))[r + \gamma V(s')]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      old-action $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

1. Initialization

```
S=[0,1,2,3,4,5,6,7,8]
V=[0,0,0,0,0,0,0,0,0,0]

a0=[0,1,1,0,0,0,0,0,0,0]
a1=[1,0,1,0,0,0,0,0,1,0]
a2=[0,0,0,1,0,1,0,0,0,0]
a3=[1,0,0,0,1,0,0,0,0,0]
a4=[0,0,0,0,0,1,1,0,0,0]
a5=[0,0,0,0,0,0,1,1,1,0]
a6=[0,0,0,0,1,0,0,1,0,1]
a7=[0,0,0,0,0,0,1,0,0,1]
a8=[0,0,0,0,0,0,0,1,1,0]
a9=[0,0,0,0,0,0,0,0,0,0]
a=[a0,a1,a2,a3,a4,a5,a6,a7,a8]

p0=[0,0.5,0.5,0,0,0,0,0,0,0]
p1=[0.4,0,0.3,0,0,0,0,0,0.3,0]
p2=[0,0,0,0.8,0,0.2,0,0,0,0]
p3=[0.6,0,0,0,0.4,0,0,0,0,0]
p4=[0,0,0,0,0,0.5,0.5,0,0,0]
p5=[0,0,0,0,0,0,0.4,0.3,0.3,0]
p6=[0,0,0,0,0.2,0,0,0.4,0,0.4]
p7=[0,0,0,0,0,0,0.2,0,0,0.8]
p8=[0,0,0,0,0,0,0,0.6,0.4,0]
p9=[0,0,0,0,0,0,0,0,0,0]
p=[p0,p1,p2,p3,p4,p5,p6,p7,p8]

r0=[0,0,0,0,0,0,0,0,0,0]
r1=[0,0,1,0,0,0,0,0,-1,0]
r2=[0,0,0,0,0,0,0,0,0,0]
r3=[-1,0,0,0,0,0,0,0,0,0]
r4=[0,0,0,0,0,1,0,0,0,0]
r5=[0,0,0,0,0,0,1,-1,-2,0]
r6=[0,0,0,0,0,0,0,0,0,10]
r7=[0,0,0,0,0,0,0,0,0,2]
r8=[0,0,0,0,0,0,0,0,0,0]
r9=[0,0,0,0,0,0,0,0,0,0]
r=[r0,r1,r2,r3,r4,r5,r6,r7,r8]

```

Action 초기화

Percentage 초기화

Reward 초기화

2. Policy Evaluation

```
policy_iteration_step = 0
while True:
    policy_iteration_step += 1
    count=1

    while True:
        print(f"[policy Evaluation] - policy_iteration_step: {policy_iteration_step} - {count}번 반복")
        delta = 0
        for s in S:
            v = V[s]
            value = 0
            i = 0
            for percentage, reward in zip(p[s], r[s]):
                value += percentage * (reward + 0.9 * V[i])
                i += 1
            V[s] = value
            delta = max(delta, abs(v - V[s]))
            print(f'V[s{s}] = {V[s]}')
        count+=1
        print()
        if delta < 0.001:
            break

```

3. Policy Improvement

```
    print(f"[policy Improvement] - policy_iteration_step: {policy_iteration_step}")
    policy_stable = True
    for s in S:
        old_action = a[s].index(1)
        q_list = []
        i = 0
        for percentage, reward in zip(p[s], r[s]):
            q_value = percentage * (reward + 0.9 * V[i])
            q_list.append(q_value)
            i += 1
        index = q_list.index(max(q_list))
        a[s][old_action] = 0
        a[s][index] = 1

        print(f'a[s{s}] = {a[s]}')
        if old_action != index:
            policy_stable = False

    if policy_stable == True:
        break
    print();print()
```

4. 최종결과

프로그래밍 실행 결과 policy_iteration_step은 총 3번 반복됐다.

policy_iteration_step: 1 → policy Evaluation 18번 발생.

policy_iteration_step: 2 → policy Evaluation 1번 발생.

policy_iteration_step: 3 → policy Evaluation 1번 발생.

1차과제 실행결과

V(S0) = 1.569871
V(S1) = 1.658858
V(S2) = 1.829745
V(S3) = 1.822560
V(S4) = 4.374526
V(S5) = 2.875012
V(S6) = 5.735046
V(S7) = 2.632308
V(S8) = 2.221010
V(S9) = 0

```
[policy Evaluation] - policy_iteration_step: 3 - 1번 반복
V[s0] = 1.5693936291497783
V[s1] = 1.6585426476047642
V[s2] = 1.829431695643747
V[s3] = 1.8223019424164908
V[s4] = 4.374526095058415
V[s5] = 2.875012360619108
V[s6] = 5.735045655109279
V[s7] = 2.6323082179196704
V[s8] = 2.2210100494013747

[policy Improvement] - policy_iteration_step: 3
a[s0] = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
a[s1] = [0, 0, 1, 0, 0, 0, 0, 0, 1, 0]
a[s2] = [0, 0, 0, 1, 0, 1, 0, 0, 0, 0]
a[s3] = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
a[s4] = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
a[s5] = [0, 0, 0, 0, 0, 0, 1, 1, 1, 0]
a[s6] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
a[s7] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
a[s8] = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0]

Process finished with exit code 0
```

[최적정책]
Policy[s0] => s2
Policy[s1] => s2
Policy[s2] => s3
Policy[s3] => s4
Policy[s4] => s6
Policy[s5] => s6
Policy[s6] => s9
Policy[s7] => s9
Policy[s8] => s7

정책 이터레이션 3회 반복 후 가치함수의 결과는 지난 1차과제 결과와 거의 비슷하게 나왔다.

5. 중간실행결과

policy_iteration_step: 1 → policy Evaluation 1번째

```
[policy Evaluation] - policy_iteration_step: 1 - 1번 반복
V[s0] = 0.0
V[s1] = 0.0
V[s2] = 0.0
V[s3] = -0.6
V[s4] = 0.5
V[s5] = -0.49999999999999994
V[s6] = 4.09
V[s7] = 2.3362000000000003
V[s8] = 1.2615480000000001
```

policy_iteration_step: 1 → policy Evaluation 6번째

```
[policy Evaluation] - policy_iteration_step: 1 - 6번 반복
V[s0] = 0.9079769041470969
V[s1] = 1.2163856347973239
V[s2] = 1.3640970802616152
V[s3] = 1.4406264776850135
V[s4] = 4.3514594876101
V[s5] = 2.8624039987910566
V[s6] = 5.7298999120550125
V[s7] = 2.631381984169902
V[s8] = 2.2120536264050377
```

policy_iteration_step: 1 → policy Evaluation 18번째

```
[policy Evaluation] - policy_iteration_step: 1 - 18번 반복
V[s0] = 1.5685092316959863
V[s1] = 1.657959445280024
V[s2] = 1.8288509167780385
V[s3] = 1.8218242546710584
V[s4] = 4.374525977384016
V[s5] = 2.8750122882556752
V[s6] = 5.735045629495005
V[s7] = 2.632308213309101
V[s8] = 2.2210099876961396
```

policy_iteration_step: 1 → policy Improvement

```
[policy Improvement] - policy_iteration_step: 1
a[s0] = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
a[s1] = [0, 0, 1, 0, 0, 0, 0, 0, 1, 0]
a[s2] = [0, 0, 0, 1, 0, 1, 0, 0, 0, 0]
a[s3] = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
a[s4] = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
a[s5] = [0, 0, 0, 0, 0, 0, 1, 1, 1, 0]
a[s6] = [0, 0, 0, 0, 0, 0, 0, 1, 0, 1]
a[s7] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
a[s8] = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0]
```

policy_iteration_step: 2 → policy Evaluation 1번째

```
[policy Evaluation] - policy_iteration_step: 2 - 1번 반복
V[s0] = 1.5690646629261282
V[s1] = 1.6583257228614343
V[s2] = 1.8292156752491837
V[s3] = 1.8221242698383553
V[s4] = 4.374526062987806
V[s5] = 2.8750123408896173
V[s6] = 5.735045648129082
V[s7] = 2.632308216663235
V[s8] = 2.221010032568757
```

policy_iteration_step: 2 → policy Improvement

```
[policy Improvement] - policy_iteration_step: 2
a[s0] = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
a[s1] = [0, 0, 1, 0, 0, 0, 0, 0, 1, 0]
a[s2] = [0, 0, 0, 1, 0, 1, 0, 0, 0, 0]
a[s3] = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
a[s4] = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
a[s5] = [0, 0, 0, 0, 0, 0, 1, 1, 1, 0]
a[s6] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
a[s7] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
a[s8] = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0]
```