Lecture E1. MDP with Model 1

Baek, Jong min

2021-01-30

차 례

policy_eval()

```
gamma = 1
states = np.arange(0,80,10)
p_normal = pd.DataFrame(np.array([
0,1,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,0,1,0,0,0,0,
0,0,0,0,1,0,0,0,
0,0,0,0,0,1,0,0,
0,0,0,0,0,0,1,0,
0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,1
]).reshape(8,8),index=states, columns=states)
p_speed = pd.DataFrame(np.array([
.1,0,.9,0,0,0,0,0,
.1,0,0,.9,0,0,0,0,
0,.1,0,0,.9,0,0,0,
0,0,.1,0,0,.9,0,0,
0,0,0,.1,0,0,.9,0,
0,0,0,0,.1,0,0,.9,
0,0,0,0,0,.1,0,.9,
0,0,0,0,0,0,0,1,
]).reshape(8,8),index=states, columns=states)
```

```
def transition(given_pi,states,p_normal,p_speed):
  p_out = pd.DataFrame(np.zeros(shape=(len(states),len(states))),index=states, columns=states)
  for s in range(len(states)) :
    action_dist = given_pi.iloc[s]
    p = action_dist['normal']*p_normal + action_dist['speed']*p_speed
    p_out.iloc[s] = p.iloc[s]
  return p_out
```

```
R_s_a = np.array([[-1,-1,-1,-1,0.0,-1,-1,0],[-1.5,-1.5,-1.5,-1.5,-0.5,-1.5,-1.5,0]]).T
R_s_a = pd.DataFrame(R_s_a,columns=['normal','speed'],index=[states])
R_s_a
```

```
##     normal  speed
## 0     -1.0   -1.5
## 10    -1.0   -1.5
## 20    -1.0   -1.5
```

```
## 30    -1.0   -1.5
## 40     0.0   -0.5
## 50    -1.0   -1.5
## 60    -1.0   -1.5
## 70     0.0    0.0
```

```
def reward_fn(given_pi,R_s_a):
  R_pi = np.sum(given_pi*R_s_a,axis=1)
  return np.array(R_pi).reshape(8,1)
```

```
def policy_eval(given_pi):
  R = reward_fn(given_pi,R_s_a=R_s_a)
  p = transition(given_pi,states=states,p_normal=p_normal,p_speed=p_speed)
  gamma = 1.0
  epsilon = 10**(-8)
  v_old = np.zeros(shape=(8,1))
  v_new = R + np.dot(gamma*p,v_old)
  while np.max(np.abs(v_new - v_old)) > epsilon :
    v_old = v_new
    v_new = R + np.dot(gamma*p,v_old)
  return v_new
```

```
pi_speed = pd.DataFrame(np.array([np.repeat(0,len(states)),np.repeat(1,len(states))]).T,columns=['normal','sp
pi_speed
```

```
##     normal  speed
## 0        0      1
## 10       0      1
## 20       0      1
## 30       0      1
## 40       0      1
## 50       0      1
## 60       0      1
## 70       0      1
```

```
policy_eval(pi_speed).T
```

```
## array([[-5.80592905, -5.2087811 , -4.13926239, -3.47576467, -2.35376031,
##         -1.73537603, -1.6735376 ,  0.        ]])
```

```
pi_50 = pd.DataFrame(np.array([np.repeat(0.5,len(states)),np.repeat(0.5,len(states))]).T,columns=['normal','s
policy_eval(pi_50).T
```

```
## array([[-5.96923786, -5.13359222, -4.11995525, -3.38922824, -2.04147003,
##         -2.02776769, -1.35138838,  0.        ]])
```

Implementation

```
# opolicy evaluation
V_old = policy_eval(pi_speed)
pi_old = pi_speed
q_s_a = R_s_a + np.c_[np.dot(gamma*p_normal,V_old),np.dot(gamma*p_speed,V_old)]
q_s_a
```

```
##       normal     speed
## 0  -6.208781 -5.805929
## 10 -5.139262 -5.208781
## 20 -4.475765 -4.139262
## 30 -3.353760 -3.475765
## 40 -1.735376 -2.353760
## 50 -2.673538 -1.735376
## 60 -1.000000 -1.673538
## 70  0.000000  0.000000
```

```
# r - apply (data,direction,function)
pi_new_vec=q_s_a.apply(np.argmax,axis=1)
pi_new = pd.DataFrame(np.zeros([len(q_s_a.index),len(q_s_a.columns)]),
columns=['normal','speed'],index=[states])
for i in range(len(pi_new_vec)):
  pi_new.iloc[i,pi_new_vec.iloc[i]] = 1
pi_new
```

```
##     normal  speed
## 0      0.0    1.0
## 10     1.0    0.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

```
def policy_improve(V_old,pi_old,R_s_a,gamma,p_normal,p_speed):
  q_s_a = R_s_a + np.c_[np.dot(gamma*p_normal,V_old),np.dot(gamma*p_speed,V_old)]
  pi_new_vec=q_s_a.apply(np.argmax,axis=1)
  pi_new = pd.DataFrame(np.zeros([len(q_s_a.index),len(q_s_a.columns)]),columns=['normal','speed'],index=[sta
  for i in range(len(pi_new_vec)):
    pi_new.iloc[i,pi_new_vec.iloc[i]] = 1
```

```
    return pi_new
```

One step improvement from pi_speed

```
pi_old = pi_speed
V_old = policy_eval(pi_old)
pi_new = policy_improve(V_old,pi_old,R_s_a,gamma,p_normal,p_speed)
```

Policy iteration

Step 0

```
pi_old = pi_speed
print(pi_old)
```

```
##     normal  speed
## 0        0      1
## 10       0      1
## 20       0      1
## 30       0      1
## 40       0      1
## 50       0      1
## 60       0      1
## 70       0      1
```

Step 1

```
V_old = policy_eval(pi_old)
pi_new = policy_improve(V_old,pi_old,R_s_a,gamma,p_normal,p_speed)
pi_old = pi_new
print(pi_old)
```

```
##     normal  speed
## 0      0.0    1.0
## 10     1.0    0.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

Step 2

```
V_old = policy_eval(pi_old)
pi_new = policy_improve(V_old,pi_old,R_s_a,gamma,p_normal,p_speed)
pi_old = pi_new
pi_old
```

```
##     normal  speed
## 0      0.0    1.0
## 10     0.0    1.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

Step 3

```
V_old = policy_eval(pi_old)
pi_new = policy_improve(V_old,pi_old,R_s_a,gamma,p_normal,p_speed)
pi_old = pi_new
pi_old
```

```
##     normal  speed
## 0      0.0    1.0
## 10     0.0    1.0
## 20     0.0    1.0
## 30     1.0    0.0
## 40     1.0    0.0
## 50     0.0    1.0
## 60     1.0    0.0
## 70     1.0    0.0
```

```
(pi_new == pi_speed).all(axis=1)
# pi_new.equals(pi_speed)
# if pi_new.equqals(pi_speed).all() :
#   print('yes')
```

```
## 0      True
## 10     True
## 20     True
```

```
## 30    False
## 40    False
## 50     True
## 60    False
## 70    False
## dtype: bool
```

Policy iteration process (from pi_speed)

```
pi_old = pi_speed
cnt = 0
while True :
  print(str(cnt)+'-th iteration')
  print(pi_old.T)
  V_old = policy_eval(pi_old)
  pi_new = policy_improve(V_old,pi_old,R_s_a,gamma,p_normal,p_speed)

  if pi_new.equals(pi_old) == True:
    break
  pi_old= pi_new
  cnt = cnt+1
```

```
## 0-th iteration
##        0  10 20 30 40 50 60 70
## normal  0  0  0  0  0  0  0  0
## speed   1  1  1  1  1  1  1  1
## 1-th iteration
##          0    10   20   30   40   50   60   70
## normal  0.0  1.0  0.0  1.0  1.0  0.0  1.0  1.0
## speed   1.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0
## 2-th iteration
##          0    10   20   30   40   50   60   70
## normal  0.0  0.0  0.0  1.0  1.0  0.0  1.0  1.0
## speed   1.0  1.0  1.0  0.0  0.0  1.0  0.0  0.0
```

```
print(policy_eval(pi_new))
```

```
## [[-5.1077441 ]
##  [-4.41077441]
##  [-3.44107744]
##  [-2.66666667]
##  [-1.66666667]
##  [-1.66666667]
##  [-1.        ]
##  [ 0.        ]]
```

E2.Rmd

```
"Hello"
```

```
## [1] "Hello"
```