Inotes2_MDP

Jaemin Park

2021-02-03

차 례

Implementation

```
gamma=0.5
states = range(1,8)
P_TL = np.matrix([[1,0,0,0,0,0,0],[1,0,0,0,0,0,0],
[0,1,0,0,0,0,0],[0,0,1,0,0,0,0],[0,0,0,1,0,0,0],
[0,0,0,0,1,0,0],[0,0,0,0,0,1,0]])
P_TR = np.matrix([[0,1,0,0,0,0,0],
[0,0,1,0,0,0,0],[0,0,0,1,0,0,0],[0,0,0,0,1,0,0],
[0,0,0,0,0,1,0],[0,0,0,0,0,0,1],[0,0,0,0,0,0,1]])

pi_50 = np.hstack((np.repeat(0.5,len(states)).reshape(7,1),np.repeat(0.5,len(states)).reshape(7,1)))
pi_50 = pd.DataFrame(pi_50,states,["Left","Right"])

pi_TL = np.hstack((np.repeat(1,len(states)).reshape(7,1),np.repeat(0,len(states)).reshape(7,1)))
pi_TL = pd.DataFrame(pi_TL,states,["Left","Right"])

pi_TR = np.hstack((np.repeat(0,len(states)).reshape(7,1),np.repeat(1,len(states)).reshape(7,1)))
pi_TR = pd.DataFrame(pi_TR,states,["Left","Right"])


def transition(given_pi, states, P_TL, P_TR):
    P_out = pd.DataFrame(np.zeros((len(states),len(states))),states,states)
    for i,s in enumerate(states):
        action_dist = given_pi.loc[s]
        P = action_dist["Left"]*P_TL + action_dist["Right"]*P_TR
        P_out.loc[s] = P[i,:]

    return P_out

R_s_a = np.matrix([[1,1,0,0,0,0,0],[0,0,0,0,0,10,10]]).T
R_s_a = pd.DataFrame(R_s_a,states,["Left","Right"])

def reward_fn(given_pi):
    R_pi = np.matrix(given_pi*R_s_a).sum(axis=1)
    R_pi = pd.DataFrame(R_pi,states)
    return(R_pi)

print(reward_fn(pi_TL).T)
```

```
##    1  2  3  4  5  6  7
## 0  1  1  0  0  0  0  0
```

```
def policy_eval(given_pi):
    R=reward_fn(given_pi).values.reshape(7,1)
    P=transition(given_pi, states, P_TL, P_TR)
    gamma=0.5
    epsilon=10**(-8)
    v_old=np.repeat(0,7).reshape(7,1)
    v_new=R+gamma*np.dot(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

V_old=policy_eval(pi_TL)
pi_old=pi_TL
q_s_a=R_s_a+np.hstack((np.dot(gamma*P_TL,V_old),np.dot(gamma*P_TR,V_old)))
print(q_s_a.T)
```

```
##           1    2     3      4       5         6         7
## Left    2.0  2.0  1.00  0.500  0.2500   0.12500   0.06250
## Right   1.0  0.5  0.25  0.125  0.0625  10.03125  10.03125
```

Policy Improve

```
pi_new_vec=q_s_a.idxmax(axis=1)
pi_new=pd.DataFrame(np.zeros(pi_old.shape), index=pi_old.index, columns=pi_old.columns)

for i in range(len(pi_new_vec)):
    pi_new.iloc[i][pi_new_vec.iloc[i]]=1

def policy_improve(V_old, pi_old = pi_old, R_s_a = R_s_a, gamma = gamma,
 P_TL = P_TL, P_TR = P_TR):
    q_s_a=R_s_a+np.hstack((np.dot(gamma*P_TL,V_old),np.dot(gamma*P_TR,V_old)))
    pi_new_vec=q_s_a.idxmax(axis=1)
    pi_new=pd.DataFrame(np.zeros(pi_old.shape), index=pi_old.index, columns=pi_old.columns)

    for i in range(len(pi_new_vec)):
        pi_new.iloc[i][pi_new_vec.iloc[i]]=1
    return pi_new

pi_old = pi_TL
V_old = policy_eval(pi_old)
pi_new = policy_improve(V_old, pi_old = pi_old, R_s_a = R_s_a, gamma = gamma, P_TL = P_TL, P_TR = P_
print(pi_old.T)
```

```
##        1  2  3  4  5  6  7
## Left   1  1  1  1  1  1  1
## Right  0  0  0  0  0  0  0
```

```
print(pi_new.T)
```

```
##          1    2    3    4    5    6    7
## Left   1.0  1.0  1.0  1.0  1.0  0.0  0.0
## Right  0.0  0.0  0.0  0.0  0.0  1.0  1.0
```

policy Iteration

```
pi_old = pi_50
cnt = 0
while True:
    print(cnt,"-th iteration")
    print(pi_old.T)
    V_old = policy_eval(pi_old)
    pi_new = policy_improve(V_old, pi_old = pi_old, R_s_a = R_s_a, gamma = gamma, P_TL = P_TL, P_TR
    if pi_new.equals(pi_old):
        break
    pi_old = pi_new
    cnt += 1
```

```
## 0 -th iteration
##          1    2    3    4    5    6    7
## Left   0.5  0.5  0.5  0.5  0.5  0.5  0.5
## Right  0.5  0.5  0.5  0.5  0.5  0.5  0.5
## 1 -th iteration
##          1    2    3    4    5    6    7
## Left   1.0  1.0  1.0  0.0  0.0  0.0  0.0
## Right  0.0  0.0  0.0  1.0  1.0  1.0  1.0
## 2 -th iteration
##          1    2    3    4    5    6    7
## Left   1.0  1.0  0.0  0.0  0.0  0.0  0.0
## Right  0.0  0.0  1.0  1.0  1.0  1.0  1.0
```

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

```
## [[ 2.        ]
##  [ 2.        ]
##  [ 2.49999999]
##  [ 4.99999999]
##  [ 9.99999999]
##  [19.99999999]
##  [19.99999999]]
```

Value Improvement

```
cnt = 0
gamma=0.5
epsilon = 10**(-8)
V_old = np.zeros(len(states)).T
V_old = pd.DataFrame(V_old,states)
results = V_old.T
while True:
    q_s_a=R_s_a+np.hstack((np.dot(gamma*P_TL,V_old),np.dot(gamma*P_TR,V_old)))
    V_new = np.matrix(q_s_a.apply(max,axis=1)).T
    V_new = pd.DataFrame(V_new,states)
    if(np.max(np.abs(V_new-V_old)).item()<epsilon or cnt>300):
        break
    results = np.vstack((results,V_new.T))
    V_old = V_new
    cnt+=1

results = pd.DataFrame(results, columns = states)
print(results.tail())
```

```
##       1    2    3    4     5     6     7
## 26  2.0  2.0  2.5  5.0  10.0  20.0  20.0
## 27  2.0  2.0  2.5  5.0  10.0  20.0  20.0
## 28  2.0  2.0  2.5  5.0  10.0  20.0  20.0
## 29  2.0  2.0  2.5  5.0  10.0  20.0  20.0
## 30  2.0  2.0  2.5  5.0  10.0  20.0  20.0
```

```
V_opt=results.tail(1).T
q_s_a=R_s_a+np.c_[np.dot(gamma*P_TL,V_opt), np.dot(gamma*P_TR, V_opt)]
print(q_s_a)
```

```
##     Left  Right
## 1   2.00   1.00
## 2   2.00   1.25
## 3   1.00   2.50
## 4   1.25   5.00
```

```
## 5   2.50  10.00
## 6   5.00  20.00
## 7  10.00  20.00
```

```
pi_opt_vec=q_s_a.idxmax(axis=1).T
print(pi_opt_vec)
```

```
## 1     Left
## 2     Left
## 3    Right
## 4    Right
## 5    Right
## 6    Right
## 7    Right
## dtype: object
```