Lecture E1. MDP with Model 1

Baek, Jong min

2021-01-29

차 례

Policy evaluation 1(Page21)

```
import numpy as np
import pandas as pd
R = np.hstack((np.repeat(-1.5, 4), -0.5, np.repeat(-1.5, 2), 0)).reshape(-1,1)
states = np.arange(0, 80, 10)
p = np.matrix([
[0.1,0,0.9,0,0,0,0,0],
[0.1,0,0,0.9,0,0,0,0],
[0,0.1,0,0,0.9,0,0,0],
[0,0,0.1,0,0,0.9,0,0],
[0,0,0,0.1,0,0,0.9,0],
[0,0,0,0,0.1,0,0,0.9],
[0,0,0,0,0,0.1,0,0.9],
[0,0,0,0,0,0,0,1.0]
])
p = pd.DataFrame(p,index=states,columns=states)
print(R)
```

```
## [[-1.5]
##  [-1.5]
##  [-1.5]
##  [-1.5]
##  [-0.5]
##  [-1.5]
##  [-1.5]
##  [ 0. ]]
```

```
print(p)
```

```
##      0    10   20   30   40   50   60   70
## 0   0.1  0.0  0.9  0.0  0.0  0.0  0.0  0.0
## 10  0.1  0.0  0.0  0.9  0.0  0.0  0.0  0.0
## 20  0.0  0.1  0.0  0.0  0.9  0.0  0.0  0.0
## 30  0.0  0.0  0.1  0.0  0.0  0.9  0.0  0.0
## 40  0.0  0.0  0.0  0.1  0.0  0.0  0.9  0.0
## 50  0.0  0.0  0.0  0.0  0.1  0.0  0.0  0.9
## 60  0.0  0.0  0.0  0.0  0.0  0.1  0.0  0.9
## 70  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
```

rewritten with intermediate saving(Page22)

```
gamma=1.0
epsilon=10**-8
v_old = np.zeros(8).reshape(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)
print(v_new.T)
```

```
## [[-5.80592905 -5.2087811  -4.13926239 -3.47576467 -2.35376031 -1.73537603
##   -1.6735376   0.        ]]
```

```
gamma = 1.0
epsilon = 10**-8
v_old = np.zeros(8).reshape(8,1)
v_new =  R + np.dot(gamma*p,v_old)
results = v_old.T
results = np.vstack([results,v_new.T])
while np.max(np.abs(v_new-v_old)) > epsilon :
  v_old = v_new
  v_new = R + np.dot(gamma*p,v_old)
  results = np.vstack([results,v_new.T])
print(v_new.T)
```

```
## [[-5.80592905 -5.2087811  -4.13926239 -3.47576467 -2.35376031 -1.73537603
##   -1.6735376   0.        ]]
```

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

```
##       0      10      20     30     40      50     60   70
## 0  0.000  0.0000  0.0000  0.000  0.000  0.0000  0.000  0.0
## 1 -1.500 -1.5000 -1.5000 -1.500 -0.500 -1.5000 -1.500  0.0
## 2 -3.000 -3.0000 -2.1000 -3.000 -2.000 -1.5500 -1.650  0.0
## 3 -3.690 -4.5000 -3.6000 -3.105 -2.285 -1.7000 -1.655  0.0
## 4 -5.109 -4.6635 -4.0065 -3.390 -2.300 -1.7285 -1.670  0.0
```

```
print(results.tail())
```

```
##           0         10        20        30       40        50        60   70
## 18 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 19 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 20 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 21 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
## 22 -5.805929 -5.208781 -4.139262 -3.475765 -2.35376 -1.735376 -1.673538  0.0
```

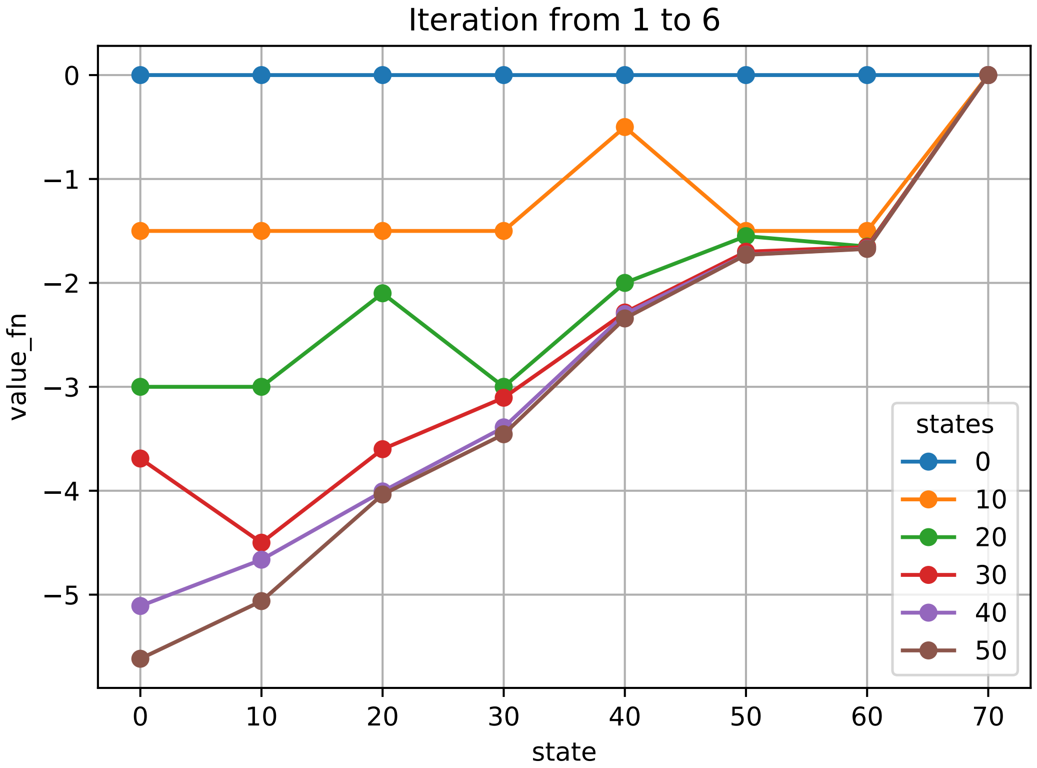
visualization

```
fig1=results[results.index < 6]
fig2=results[(results.index >= 7)&(results.index < 12)]
fig3=results[(results.index >= 13)&(results.index < 18)]
```

```
plt.plot(fig1.T,marker='o')
```

```
## [<matplotlib.lines.Line2D object at 0x000000002CB16278>, <matplotlib.lines.Line2D object at 0x000000002CB1632
```

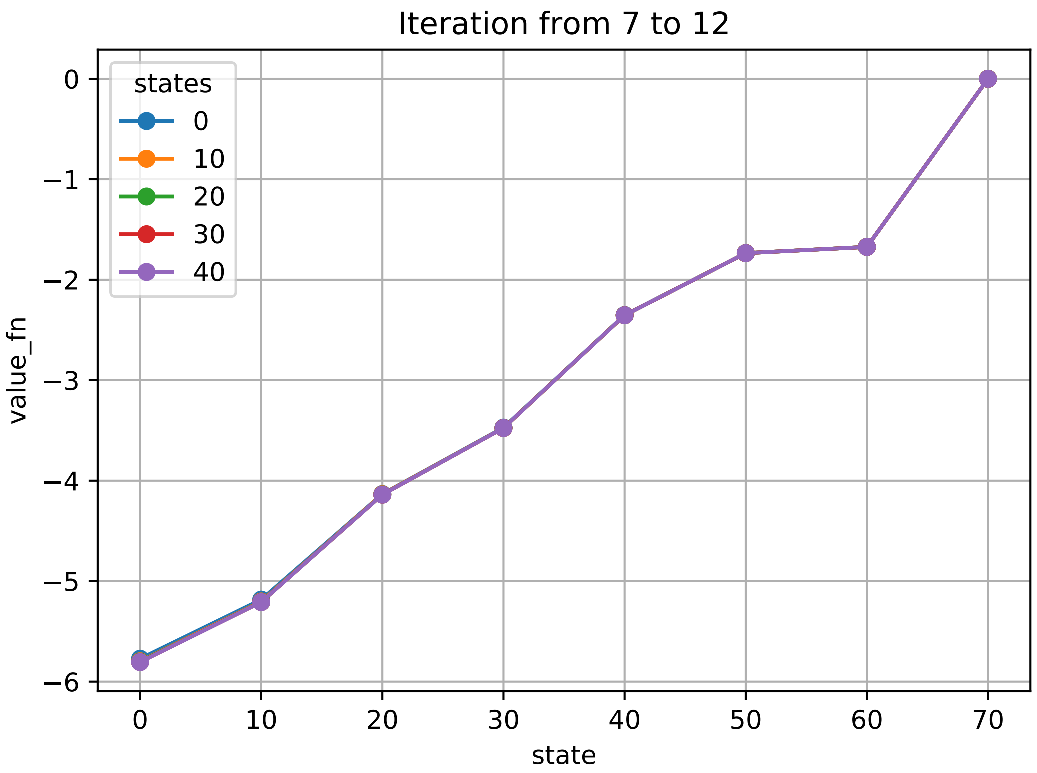
```
plt.legend(fig1.columns,title='states')
plt.grid(True)
plt.xlabel('state')
plt.ylabel('value_fn')
plt.title('Iteration from 1 to 6')
plt.show()
```



```
plt.plot(fig2.T,marker='o')
```

```
## [<matplotlib.lines.Line2D object at 0x000000002DC04F60>, <matplotlib.lines.Line2D object at 0x000000002DC04E
```

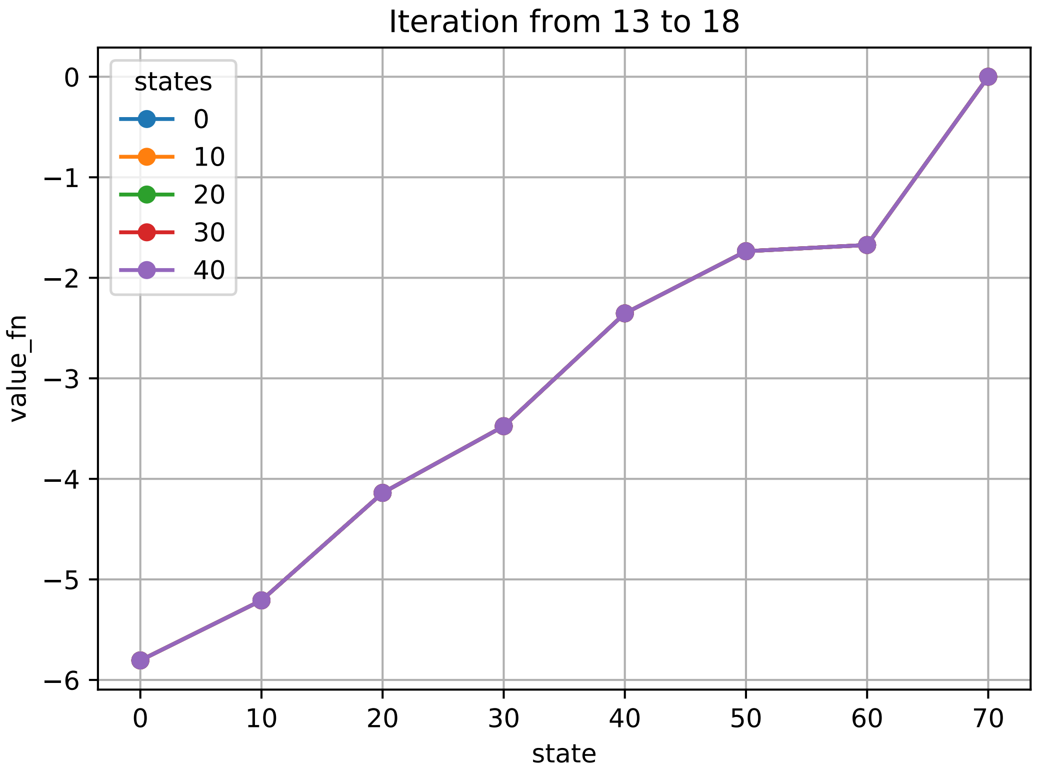
```
plt.legend(fig1.columns,title='states')
plt.grid(True)
plt.xlabel('state')
plt.ylabel('value_fn')
plt.title('Iteration from 7 to 12')
plt.show()
```



```
plt.plot(fig3.T,marker='o')
```

```
## [<matplotlib.lines.Line2D object at 0x000000002DBED978>, <matplotlib.lines.Line2D object at 0x000000002DBEDF
```

```
plt.legend(fig1.columns,title='states')
plt.grid(True)
plt.xlabel('state')
plt.ylabel('value_fn')
plt.title('Iteration from 13 to 18')
plt.show()
```


Iteration from 13 to 18
states
0
10
20
30
40
value_fn
0
−1
−2
−3
−4
−5
−6
state
0
10
20
30
40
50
60
70

policy evalutation 2

$꩜(pi) : S →A $

```
states = np.arange(0,80,10)
pi_speed = np.array([np.repeat(0,len(states)),np.repeat(1,len(states))]).T
pi_speed = pd.DataFrame(pi_speed,columns=['normal','speed'],index=[states])
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
```

$R^{\pi} : S \to \mathbb{R}$

```
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 = 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_pi = np.sum(given_pi*R_s_a,axis=1)
  return R_pi
reward_fn(pi_speed)
```

```
## 0    -1.5
## 10   -1.5
## 20   -1.5
## 30   -1.5
## 40   -0.5
## 50   -1.5
## 60   -1.5
## 70    0.0
## dtype: float64
```

$P^\pi : S \times A \to S$

```
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)
```

```python
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
```

```python
transition(pi_speed,states,p_normal,p_speed)
```

```
##       0    10   20   30   40   50   60   70
## 0   0.1  0.0  0.9  0.0  0.0  0.0  0.0  0.0
## 10  0.1  0.0  0.0  0.9  0.0  0.0  0.0  0.0
## 20  0.0  0.1  0.0  0.0  0.9  0.0  0.0  0.0
## 30  0.0  0.0  0.1  0.0  0.0  0.9  0.0  0.0
## 40  0.0  0.0  0.0  0.1  0.0  0.0  0.9  0.0
## 50  0.0  0.0  0.0  0.0  0.1  0.0  0.0  0.9
## 60  0.0  0.0  0.0  0.0  0.0  0.1  0.0  0.9
## 70  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
```

Test 2

```
pi_50 = pi_speed = pd.DataFrame(np.array([np.repeat(0.5,len(states)),np.repeat(0.5,len(states))]).T,columns=[
pi_50
```

```
##     normal  speed
## 0      0.5    0.5
## 10     0.5    0.5
## 20     0.5    0.5
## 30     0.5    0.5
## 40     0.5    0.5
## 50     0.5    0.5
## 60     0.5    0.5
## 70     0.5    0.5
```

```
transition(pi_50,states,p_normal,p_speed)
```

```
##        0    10    20    30    40    50    60    70
## 0   0.05  0.50  0.45  0.00  0.00  0.00  0.00  0.00
## 10  0.05  0.00  0.50  0.45  0.00  0.00  0.00  0.00
## 20  0.00  0.05  0.00  0.50  0.45  0.00  0.00  0.00
## 30  0.00  0.00  0.05  0.00  0.50  0.45  0.00  0.00
## 40  0.00  0.00  0.00  0.05  0.00  0.50  0.45  0.00
## 50  0.00  0.00  0.00  0.00  0.05  0.00  0.50  0.45
## 60  0.00  0.00  0.00  0.00  0.00  0.05  0.00  0.95
## 70  0.00  0.00  0.00  0.00  0.00  0.00  0.00  1.00
```

E1.Rmd

```
"Hello"
```

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