D2 Python Code

Kang, Eui Hyeon

2021-01-15

차 례

17p

```
P=np.array([0.7,0.5,0.3,0.5]).reshape(2,2,order='F')
R=np.array([1.5,1.0]).reshape(2,1)

gamma=.9

V=np.dot(np.linalg.inv(np.eye(2)-gamma*P),R)
V
```

```
## array([[13.35365854],
##        [12.74390244]])
```

23p

```
R=np.array([1.5,1.0]).reshape(2,1)
P=np.array([0.7,0.5,0.3,0.5]).reshape(2,2,order='F')
gamma=.9
epsilon=10**(-8)

v_old=np.array(np.zeros(2,)).reshape(2,1)
v_new=R+np.dot(gamma*P,v_old)

while True:
    v_old=v_new
    v_new=R+np.dot(gamma*P,v_old)

    if np.max(np.abs(v_new-v_old))>epsilon:
        continue
    break

print(v_new)
```

```
## [[13.35365845]
##  [12.74390235]]
```

The full iteration process - 24p

```
R=np.array([1.5,1.0]).reshape(2,1)
P=np.array([0.7,0.5,0.3,0.5]).reshape(2,2,order='F')
gamma=0.9
epsilon=10**(-8)

v_old=np.array(np.zeros(2,)).reshape(2,1)
v_new=R+np.dot(gamma*P,v_old)

results=v_old.T
results=np.vstack((results,v_new.T))

while True:
    v_old=v_new
    v_new=R+np.dot(gamma*P, v_old)
    results=np.vstack((results,v_new.T))
    if np.max(np.abs(v_new-v_old))>epsilon:
        continue
    break

results=pd.DataFrame(results, columns=['coke','pepsi'])

print(results.head())
```

```
##        coke    pepsi
## 0  0.000000  0.00000
## 1  1.500000  1.00000
## 2  2.715000  2.12500
## 3  3.784200  3.17800
## 4  4.742106  4.13299
```

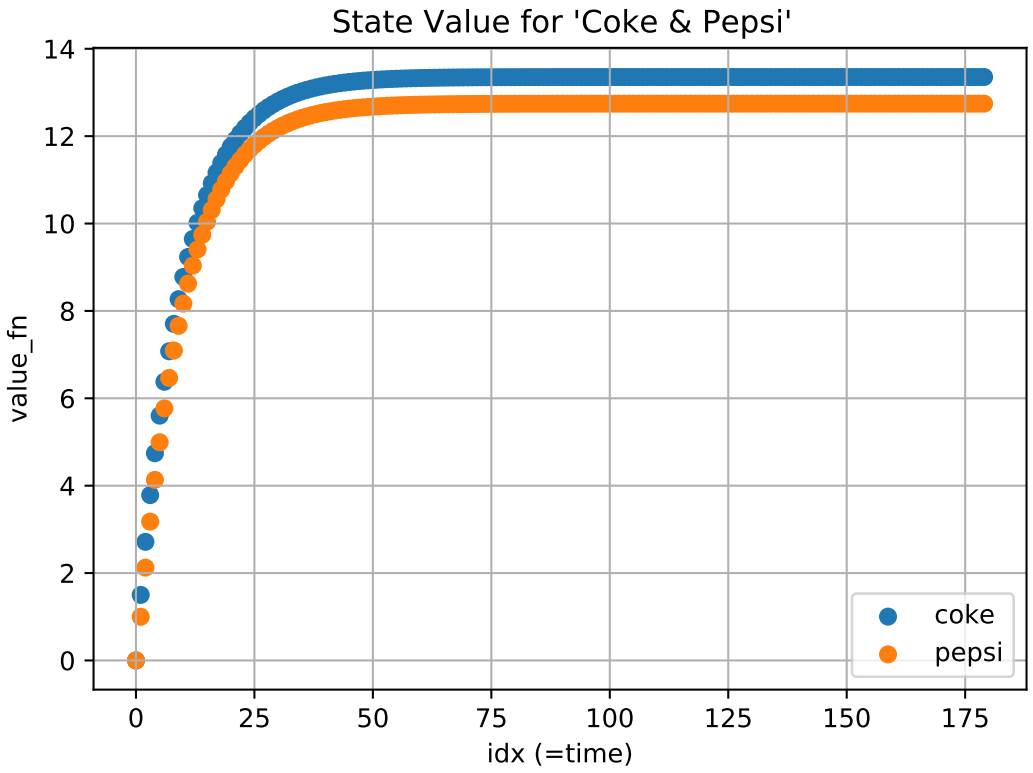
```
print(results.tail())
```

```
##           coke      pepsi
## 175  13.353658  12.743902
## 176  13.353658  12.743902
## 177  13.353658  12.743902
## 178  13.353658  12.743902
## 179  13.353658  12.743902
```

Plot - 25p

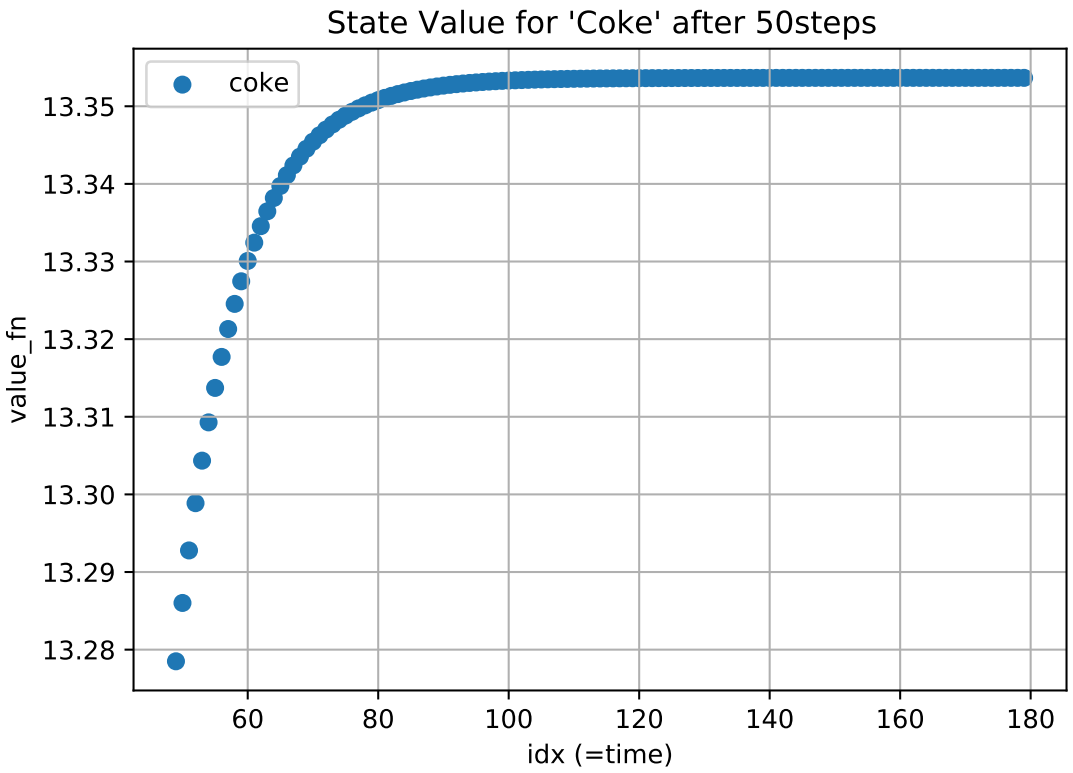
```
import matplotlib.pyplot as plt


plt.scatter(results.index, results['coke'],label='coke')
plt.scatter(results.index, results['pepsi'],label='pepsi')
plt.grid(True)
plt.title("State Value for 'Coke & Pepsi'")
plt.ylabel('value_fn')
plt.xlabel('idx (=time)')
plt.legend()
plt.show()
```



50 steps (coke only)

```
plt.scatter(results.index[49:], results['coke'][49:],label='coke')
plt.grid(True)
plt.title("State Value for 'Coke' after 50steps")
plt.ylabel('value_fn')
plt.xlabel('idx (=time)')
plt.legend()
plt.show()
```



100steps (coke only)

```
plt.scatter(results.index[99:], results['coke'][99:],label='coke')
plt.grid(True)
plt.title("State Value for 'Coke' after 100steps")
ax=plt.gca()
ax.get_yaxis().get_major_formatter().set_useOffset(False)
plt.ylabel('value_fn')
plt.xlabel('idx (=time)')
plt.legend()
plt.show()
```