Homework3 Nov 20th

Fangyuan Li

```
import numpy as np
import matplotlib.pyplot as plt
```

1: Simulation of Markov Process

(a)

$$P = \begin{bmatrix} 0.2 & 0.7 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{bmatrix}$$

(b)

```
p = np.array([[0.2,0.7,0.1],[0.2,0.5,0.3],[0.2,0.4,0.4]])

X = 0
count = 1
state = []
cum_p = np.cumsum(p,axis=1)
while 1:
    state.append(X+1)
    count += 1
    u = np.random.uniform(size=1)
    if u<cum_p[X][0]:
        X = 0
    elif u<cum_p[X][1]:
        X = 1
```

```
        else:
            X = 2
        if count > 20:
            break
state
```

```
[1, 1, 2, 3, 3, 2, 3, 2, 3, 1, 3, 2, 3, 2, 3, 3, 1, 2, 3, 2]
```

I simulated one chain with length 20.

2: Stationary Distribution

(a)

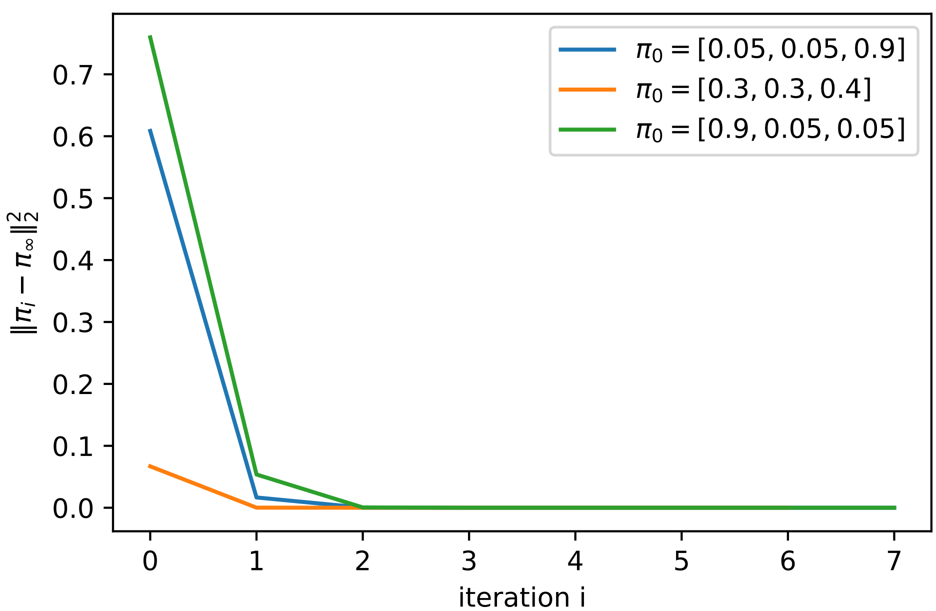
$$\begin{aligned}
\pi_1 &= 0.2\pi_1 + 0.2\pi_2 + 0.2\pi_3 \\
\pi_2 &= 0.7\pi_1 + 0.5\pi_2 + 0.4\pi_3 \\
\pi_3 &= 0.1\pi_1 + 0.3\pi_2 + 0.4\pi_3 \\
&\begin{bmatrix} -0.8 & 0.2 & 0.2 \\ 0.7 & -0.5 & 0.4 \\ 0.1 & 0.3 & -0.6 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = 0 \\
&\begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} \\ \frac{23}{45} \\ \frac{13}{45} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.51 \\ 0.289 \end{bmatrix}
\end{aligned}$$

(b)

```
def norm2(x):
    return np.sum(x**2)
def f(pi0):
    pi = np.array([9/45,23/45,13/45])
    plabel = pi0
    i = 0
    x = []
    threhold = 1e-13
    err = []
    while 1:
```

```
        x.append(i)
        tmp = norm2(pi-pi0)
        err.append(tmp)
        if tmp < threhold:
            break
        pi0 = pi0 @ p
        i += 1
    plt.plot(x,err,label=r'$\pi_0=[{}]$'.format(','.join(map(lambda x:
↪   str(x), plabel))))

pi0 = np.array([[0.05,0.05,0.9],[0.3,0.3,0.4],[0.9,0.05,0.05]])
for i in pi0:
    f(i)
plt.legend()
plt.xlabel('iteration i')
plt.ylabel(r'$\|\pi_{i}-\pi_{\infty}\|_{2}^{2}$')
plt.show()
```

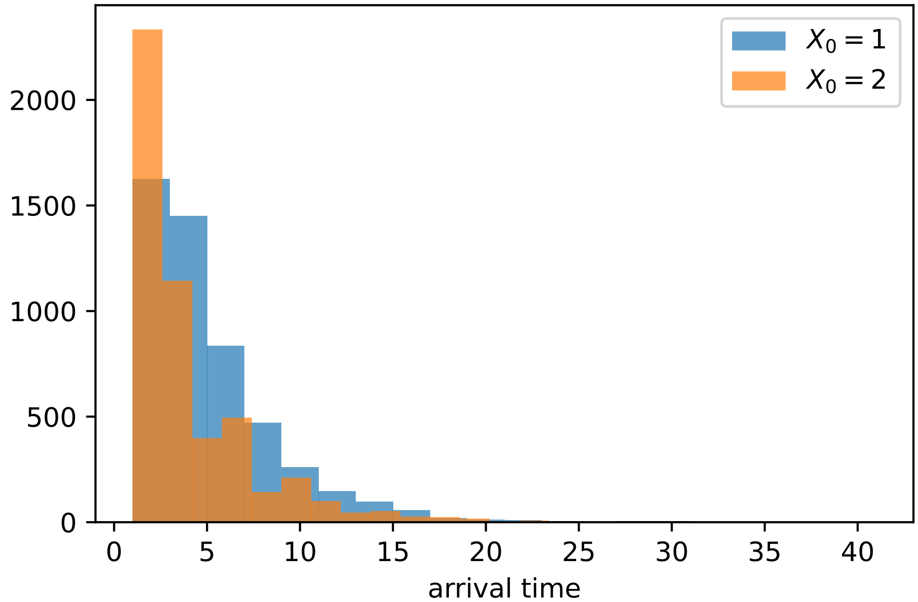


3: Absorbing state

(a)

```
def generate_state(X):
    state = []
    cum_p = np.cumsum(p,axis=1)
    while 1:
        state.append(X+1)
        if (X+1)==3:
            break
        u = np.random.uniform(size=1)
        if u<cum_p[X][0]:
            X = 0
        elif u<cum_p[X][1]:
            X = 1
        else:
            X = 2
    return len(state)-1

X1 = []
X2 = []
for i in range(5000):
    X1.append(generate_state(X=0))
    X2.append(generate_state(X=1))
plt.hist(X1,alpha=0.7,bins=20,label=r'$X_0=1$')
plt.hist(X2,alpha=0.7,bins=20,label=r'$X_0=2$')
plt.legend()
plt.xlabel('arrival time')
plt.show()
print('mean of X_0 = 1: {}'.format(np.mean(X1)))
print('mean of X_0 = 2: {}'.format(np.mean(X2)))
```




```
mean of X_0 = 1: 4.6734
mean of X_0 = 2: 3.8846
```

(b)

$$\mu_1 = 1 + 0.7\mu_2 + 0.2\mu_1$$
$$\mu_2 = 1 + 0.2\mu_1 + 0.5\mu_2$$
$$\mu_1 = \frac{60}{13} = 4.615$$
$$\mu_2 = \frac{50}{13} = 3.846$$
$$\mu_3 = 0$$