EE503 HW7 Chih-Cheng Hsieh

1. Let R: rain

N: not rain

There are several states:

(RRR), (RRN), (RNR), (RNN), (NRR), (NRN), (NNR), (NNN).

Total 8 states #

2. $\because$ Prob. depends on time n

$\therefore$ not a time-homogeneous

To transform it into a time-homogeneous

Let E: even, D: odd

Let $i = \{(E,0), (E,1), (E,2), (D,0), (D,1), (D,2)\}$

#

3. From induction

when $\boxed{n=1}$

$$P^{(1)} = \begin{bmatrix} \frac{1}{2}+\frac{1}{2}(2p-1)^1 & \frac{1}{2}-\frac{1}{2}(2p-1)^1 \\ \frac{1}{2}-\frac{1}{2}(2p-1)^1 & \frac{1}{2}+\frac{1}{2}(2p-1)^1 \end{bmatrix} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} = P \checkmark$$

when $\boxed{n=n}$

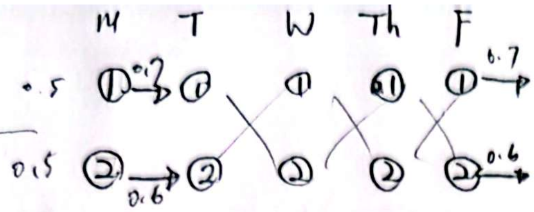
$$P^{(n)} = \begin{bmatrix} \frac{1}{2}+\frac{1}{2}(2p-1)^n & \frac{1}{2}-\frac{1}{2}(2p-1)^n \\ \frac{1}{2}-\frac{1}{2}(2p-1)^n & \frac{1}{2}+\frac{1}{2}(2p-1)^n \end{bmatrix}$$

when $\boxed{n=n+1}$

$$P^n \cdot P = \begin{bmatrix} \frac{1}{2}+\frac{1}{2}(2p-1)^n & \frac{1}{2}-\frac{1}{2}(2p-1)^n \\ \frac{1}{2}-\frac{1}{2}(2p-1)^n & \frac{1}{2}+\frac{1}{2}(2p-1)^n \end{bmatrix} \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} = \begin{bmatrix} A & B \\ B & A \end{bmatrix}$$

$$A = \cancel{\tfrac{1}{2}p} + \frac{1}{2}(2p-1)^n p + \frac{1}{2} - \cancel{\tfrac{1}{2}p} - \frac{1}{2}(2p-1)^n + \frac{1}{2}(2p-1)^n p$$

$$= \frac{1}{2} + \frac{1}{2}(2p-1)^n + (2p-1)^n p = \frac{1}{2} + (p-\frac{1}{2})(2p-1)^n = \frac{1}{2}+\frac{1}{2}(2p-1)^{n+1}$$



$B = \frac{1}{2} - \cancel{\frac{1}{2}p} + \frac{1}{2}(2p-1)^n - \frac{1}{2}(2p-1)^n p + \cancel{\frac{1}{2}p} - \frac{1}{2}(2p-1)^n p$

$= \frac{1}{2} + \frac{1}{2}(2p-1)^n - (2p-1)^n p = \frac{1}{2} + (\frac{1}{2} - p)(2p-1)^n$

$= \frac{1}{2} - \frac{1}{2}(1-2p)^{n+1}$

$$\therefore P^n \cdot P = \begin{bmatrix} A & B \\ B & A \end{bmatrix} = \begin{bmatrix} \frac{1}{2} + \frac{1}{2}(1-2p)^{n+1} & \frac{1}{2} - \frac{1}{2}(1-2p)^{n+1} \\ \frac{1}{2} - \frac{1}{2}(1-2p)^{n+1} & \frac{1}{2} + \frac{1}{2}(1-2p)^{n+1} \end{bmatrix} = P^{(n+1)}$$

※

4. Coin 1 : $P(H) = 0.7$

Coin 2 : $P(H) = 0.6$

$$\begin{array}{r} 0.49 \\ 0.12 \\ \hline .61 \end{array}$$

$$P = \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

$$P^2 = \begin{bmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{bmatrix}$$

$P(X_2 = 1) = \pi_1 \cdot 0.61 + \pi_2 \cdot 0.52 = 0.5\,(0.61 + 0.52) = 0.565$

(Solution of third day) ※

$$P^3 = \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0.583 & 0.417 \\ 0.556 & 0.444 \end{bmatrix} \end{array}$$

$\pi_1 = 0.5 \times 0.7 = 0.35$

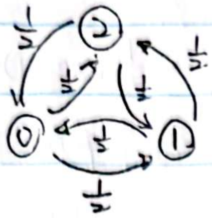
$\pi_2 = 0.5 \times 0.6 = 0.3$

$P(\text{Friday} = H) = ((\pi_1 \times 0.583) + (\pi_2 \times 0.556)) \times 0.7$

$\quad + (\pi_1 \times 0.417 + \pi_2 \times 0.444) \times 0.6$

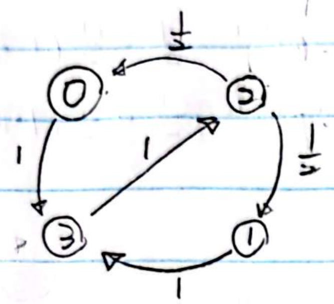
$= 0.4271$ ※ (Solution of Friday's flip = H

5. P_1 :



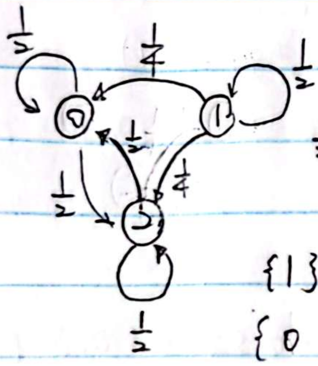
recurrent #

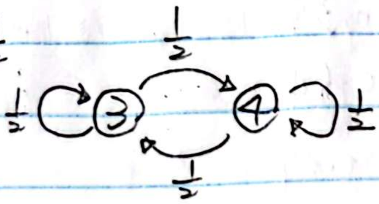
P_2 :



recurrent #

P_3 :



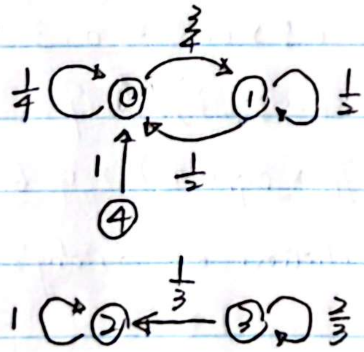


$\{1\}$: transient

$\{0\ 2\}$: recurrent

$\{3\ 4\}$: recurrent

P_4 :



$\{0, 1\}$: recurrent

$\{2\}$: recurrent

$\{3\}$, $\{4\}$: transient

6. $P = \begin{array}{c|c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix} \end{array}$

a) $P(\text{flip coin } 1) = \dfrac{0.6+0.5}{0.6+0.5+0.4+0.5} = 0.55$ #

b) $P^5 = \begin{array}{c|c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0.55556 & 0.44444 \\ 0.55555 & 0.44445 \end{bmatrix} \end{array}$

$P_{12} = 0.44444$ #

7. Limiting prob. : $\begin{cases} \Pi = \Pi P & ① \\ \sum_j \Pi_j = 1 & ② \end{cases}$

① $\Pi P = \Pi \Rightarrow \sum_i \Pi_i P_{ij} = \sum_i \frac{1}{M+1} P_{ij} = \frac{1}{M+1} \sum_i P_{ij} = \frac{1}{M+1} \cdot 1 =$ #

② $\sum_j \Pi_j = 1 \Rightarrow \sum_j \Pi_j = \sum_j \frac{1}{M+1} = \sum_{j=0}^{M} \frac{1}{M+1} = \frac{1}{M+1} \times (M+1) = 1$ #

8. $\Pi = [\Pi_1, \Pi_2, \ldots, \Pi_n]$

$\Pi(I - P + 11^T) = 1^T \Rightarrow \Pi - \Pi P + [\sum_{i=1}^{n} \Pi_i, \sum_{i=1}^{n} \Pi_i \cdots]$

If $\Pi P = \Pi$, $\sum_{i=1}^{n} \Pi_i = 1$

$\Rightarrow \Pi(I - P + 11^T) = \Pi - \Pi P + [\sum_{i=1}^{n} \Pi_i, \sum_{i=1}^{n} \Pi_i \ldots]$

$= \cancel{\Pi - \Pi} + [1\ 1 \ldots\ 1] = [1\ 1, \ldots 1] = 1^T$

#

EE503 HW7 Chih-Chung Hsieh

9. a)

$$P = \begin{bmatrix} p(1-p)+(1-p)^2 & p^2+(1-p)p & 0 & 0 \\ p(1-p) & p^2+(1-p)^2 & p(1-p) & 0 \\ 0 & p(1-p) & p^2+(1-p)^2 & (1-p)p \\ 0 & 0 & p(1-p) & p^2+(1-p) \end{bmatrix}$$

p: rain

b) $\begin{cases} \pi = \pi P \\ \sum_i \pi_i = 1 \end{cases}$, $\sum_j \pi_i = \sum_{j=0}^{3} \pi_j = \frac{q}{r+q} + \frac{1}{r+q} \cdot 3 = \frac{q+3}{3+q} = 1$

Let $\pi_0 = P(X_0 = 0)$

$\pi_{1,\dots,r} = P(X_0 = 1, \dots, r)$

$$P(X_1 = 0) = \pi_0\left(p(1-p) + (1-p)^2\right) + \pi_1\left(p(1-p)\right)$$

$$= \frac{q}{r+q}\left(p(1-p)+(1-p)^2\right) + \frac{1}{r+q}\left(p(1-p)\right)$$

$$= \frac{p(1-p)^2 + (1-p)^3 + p(1-p)}{r+q} = \frac{(1-p)\left(p(1-p) + (1-p)^2 + p\right)}{r+q}$$

$$= \frac{(1-p)\left(\cancel{p} - \cancel{p^2} + 1 - \cancel{2p} + \cancel{p^2} + \cancel{p}\right)}{r+q} = \frac{1-p}{r+q} = \frac{q}{r+q} = \pi_0$$

#

c)

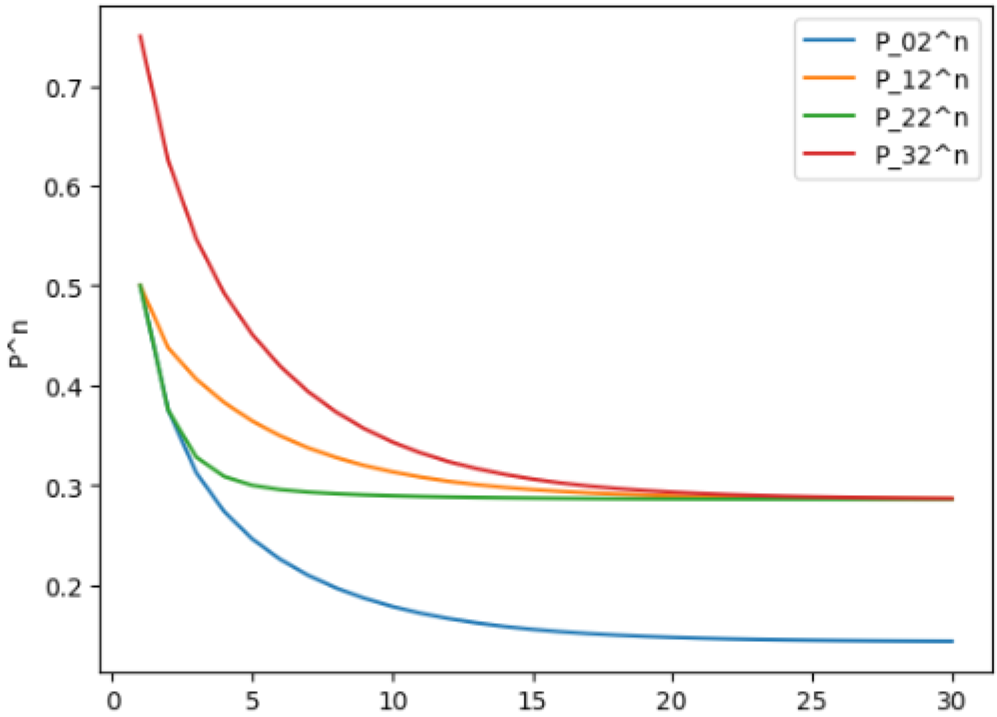
(c)

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

def calculate_P(p, n):
    P = np.array([[p*(1-p)+(1-p)*(1-p), p*p+(1-p)*p, 0, 0],
                  [p*(1-p), p*p+(1-p)*(1-p), p*(1-p), 0],
                  [0, p*(1-p), p*p+(1-p)*(1-p), p*(1-p)],
                  [0, 0, p*(1-p), p*p+(1-p)]])
    result = np.linalg.matrix_power(P, n)
    return result

p = 0.5
n_val = range(1, 31)
for i in range(4):
    P_val = [calculate_P(p, n)[i][i] for n in n_val]
    plt.plot(n_val, P_val, label = f'P_{i}2^n')

plt.xlabel('n')
plt.ylabel('P^n')
plt.legend()
plt.show()
```



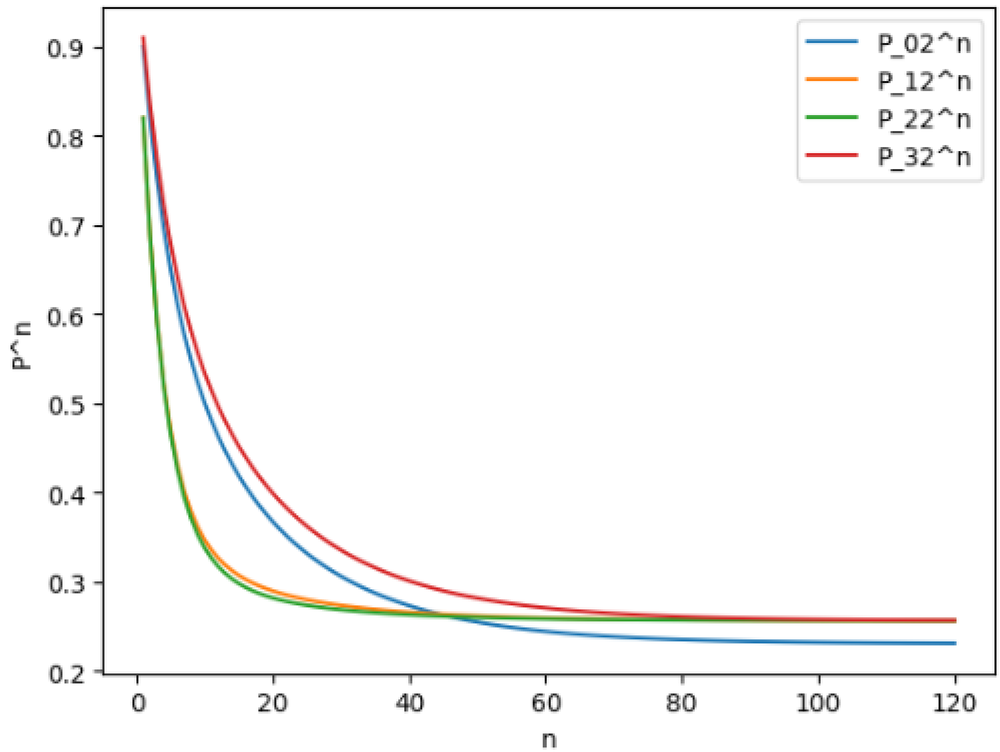
(d)

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

def calculate_P(p, n):
    P = np.array([[p*(1-p)+(1-p)*(1-p), p*p+(1-p)*p, 0, 0],
                  [p*(1-p), p*p+(1-p)*(1-p), p*(1-p), 0],
                  [0, p*(1-p), p*p+(1-p)*(1-p), p*(1-p)],
                  [0, 0, p*(1-p), p*p+(1-p)]])
    result = np.linalg.matrix_power(P, n)
    return result

p = 0.1
n_val = range(1, 121)
for i in range(4):
    P_val = [calculate_P(p, n)[i][i] for n in n_val]
    plt.plot(n_val, P_val, label = f'P_{i}2^n')

plt.xlabel('n')
plt.ylabel('P^n')
plt.legend()
plt.show()
```



When p = 0.5, P converges at n = 30, whereas when p = 0.1, it takes until n = 100 to converge. This indicates that convergence is faster when p = 0.5.