Assignment : 1

Team Member: Jash Patel (M22CS061)
Asmita Rani (M22CS059)

First 3 Problem:

- Import required file
- Make assumption of State distribution matrix, transition matrix, emission matrix using problem statement guideline

```
self.mat_A = np.array([[0.2, 0.3, 0.1, 0.2, 0.2],
                       [0.2, 0.3, 0.3, 0.1, 0.1],
                       [0.1, 0.1, 0.2, 0.5, 0.1],
                       [0.5, 0.1, 0.1, 0.2, 0.1],
                       [0.2, 0.3, 0.1, 0.1, 0.3]])

# self.mat_B = np.array([[0.00, 0.00, 0.40, 0.00, 0.00, 0.00, 0.20, 0.00, 0.00, 0.00, 0.40, 0.00],
#                        [0.25, 0.00, 0.00, 0.00, 0.25, 0.00, 0.00, 0.00, 0.50, 0.00, 0.00, 0.00],
#                        [0.00, 0.00, 0.00, 0.30, 0.00, 0.00, 0.00, 0.30, 0.00, 0.00, 0.00, 0.40],
#                        [0.00, 0.00, 0.00, 0.30, 0.00, 0.00, 0.00, 0.30, 0.00, 0.00, 0.00, 0.40],
#                        [0.00, 0.25, 0.00, 0.00, 0.00, 0.50, 0.00, 0.00, 0.00, 0.25, 0.00, 0.00]])

self.mat_B = np.array([[0.00, 0.00, 0.70, 0.00, 0.00, 0.00, 0.15, 0.00, 0.00, 0.00, 0.15, 0.00],
                       [0.15, 0.00, 0.00, 0.00, 0.15, 0.00, 0.00, 0.00, 0.70, 0.00, 0.00, 0.00],
                       [0.00, 0.00, 0.00, 0.15, 0.00, 0.00, 0.00, 0.15, 0.00, 0.00, 0.00, 0.70],
                       [0.00, 0.00, 0.00, 0.15, 0.00, 0.00, 0.00, 0.15, 0.00, 0.00, 0.00, 0.70],
                       [0.00, 0.15, 0.00, 0.00, 0.00, 0.70, 0.00, 0.00, 0.00, 0.15, 0.00, 0.00]])

self.PI = np.array([0.0, 0.0, 0.5, 0.5, 0.0])
```

- Convert state, transition enum in index using `DatasetToState()` Function

Estimation step

Forward

1. $\alpha_i(1) = \pi_i b_i(y_1)$,
2. $\alpha_i(t+1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) a_{ji}$.

Backward

1. $\beta_i(T) = 1$,
2. $\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_j(y_{t+1})$.

Update

$$\gamma_i(t) = P(X_t = i | Y, \theta) = \frac{P(X_t = i, Y|\theta)}{P(Y|\theta)} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

equals α (forward) × β (backward) for state i at time t

$$\xi_{ij}(t) = P(X_t = i, X_{t+1} = j | Y, \theta) = \frac{P(X_t = i, X_{t+1} = j, Y|\theta)}{P(Y|\theta)} = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}$$

equals α for state i at time t × tranistion prob. between i and j × β for state j at time $t+1$ × observe y_{t+1} for state j

Maximization step

$$\pi_i^* = \gamma_i(1)$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$P(X_t = i, X_{t+1} = j | Y, \theta)$

$P(X_t = i | Y, \theta)$

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T} 1_{y_t = v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

(sum γ over all time steps where the observation y_t is the same as v_t at time t)

where

$$1_{y_t = v_k} = \begin{cases} 1 & \text{if } y_t = v_k, \\ 0 & \text{otherwise} \end{cases}$$

- Apply learn model algorithm
- Calculate likelihood

1. initialize

initial state distribution

$$\alpha_1(j) = \pi_j b_j(y_1)$$

probability of observing y_1 given state j

2. For each time step

transition probabity

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(y_t)$$

sum over all states

probability of observing y_t given current state $= j$

probability of all previous observations give last state i

3. Result

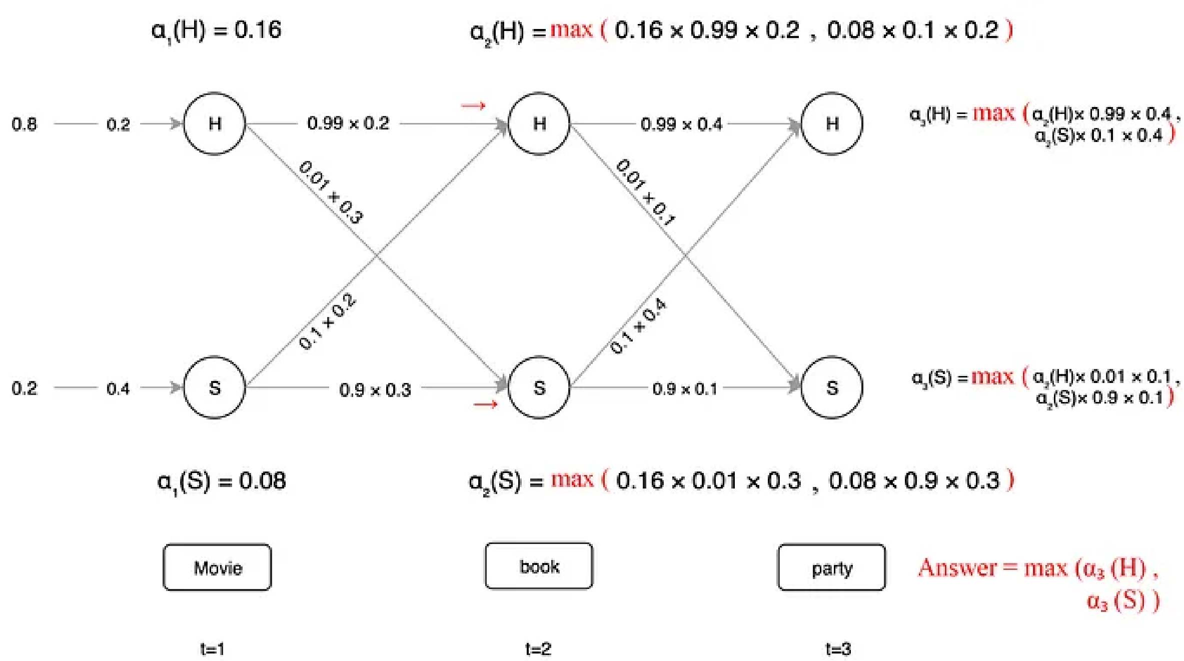
$$P(Y|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

all observations

final step

sum over all possible state

- Calculate max are using below algorithm



- Result

```
⊗ (base) jash@jash-Iaptop:~/WorkSpace/MTech/sem2/AAI/Assignment/A1-boilerplate-code/Python$ python tests.py
  Unit test #7 failure: Basic assumption test failed
  1 tests failed!
○ (base) jash@jash-Iaptop:~/WorkSpace/MTech/sem2/AAI/Assignment/A1-boilerplate-code/Python$
                                                                    Ln 376, Col 19 (14 selected)    Spaces: 4
```

- Here only one test case failed.