Q. 2 H.W.

a)

$$P(S_{t+1}=j \mid S_t=i, O_1 \ldots O_T)$$

$$= \frac{P(S_t=i, S_{t+1}=j, O_1 \ldots O_T)}{P(S_t=i, O_1 \ldots O_T)} \quad \because \text{Prod rule}$$

For num:-

$$P(S_t=i, S_{t+1}=j, O_1 \ldots O_T)$$

$$= P(S_t=i, O_1, \ldots O_t) \times P(S_{t+1}=j, O_{t+1} \ldots O_T \mid S_t=i, O_1 \ldots O_t)$$

$$= \alpha_{it}\, P(S_{t+1}=j, O_{t+1} \ldots O_T \mid S_t=i)$$

$\because$ CI by d(i)

$$= \frac{\alpha_{it}}{P(S_t=i)} \times P(S_t=i, S_{t+1}=j, O_{t+1} \ldots O_T)$$

$$= \frac{\alpha_{it}}{P(S_t=i)}\, P(S_{t+1}=j)\, P(O_{t+2} \ldots O_T \mid S_{t+1}=j)\, P(S_t=i, O_{t+1} \mid S_{t+1}, O_{t+2} \ldots T)$$

$$= \frac{\alpha_{it}}{P(S_t=i)} \times P(S_{t+1}=j)\, \beta_{j,t+1}\, P(S_t=i, O_{t+1} \mid S_{t+1}=j)$$

$\because$ CI by d(ii)

$$= \frac{\alpha_{it}}{P(S_t=i)} \times \beta_{j,t+1}\, P(S_t=i, O_{t+1}, S_{t+1}=j)$$

$\because$ Prod. rule

$$= \frac{\alpha_{it}}{P(S_t=i)} \times P(S_t=i)\, P(S_{t+1}=j \mid S_t=i)\, P(O_{t+1} \mid S_{t+1}=j) \times \beta_{j,t+1}$$

$\because$ CI by d(i)

$$= \alpha_{it}\, \beta_{j,t+1}\, a_{ij}\, b_{j\,O_{t+1}} \quad (1)$$

Scanned with CamScanner

a) For den :-

$$P(S_t = i, O_1, O_2, \cdots O_T)$$

$$= P(S_t = i, O_1 \cdots O_t) \times P(O_{t+1} \cdots O_T \mid S_t = i, O_1 \cdots O_t)$$

$$= \alpha_{it} \times P(O_{t+1} \cdots O_T \mid S_t = i)$$

$\because$ CI by d(i)

$$= \alpha_{it} \cdot \beta_{it} \quad - (2)$$

$\therefore$ Using (1) & (2),

$$\frac{\cancel{\alpha_{it}}\ \beta_{jt+1}\ a_{ij}\ b_{jO_{t+1}}}{\cancel{\alpha_{it}}\ \beta_{it}}$$

$$= \frac{\beta_{jt+1} \times a_{ij} \times b_{jO_{t+1}}}{\beta_{it}}$$

//

//

b) $P(S_t = i \mid S_{t+1} = j, O_1, O_2 \cdots O_T)$

$$= \frac{P(S_t = i, S_{t+1} = j, O_1 \cdots O_T)}{P(S_{t+1} = j, O_1, \cdots O_T)}$$

Numerator from part a — 1

Scanned with CamScanner

=) For denom:-

$$P(S_{t+1}=j, O_1 \cdots O_{t+1})$$
$$\times P(O_{t+2} \cdots O_T \mid S_{t+1}=j, O_1, \cdots O_{t+1})$$

$$= \alpha_{j\,t+1} \times P(O_{t+2} \cdots O_T \mid S_{t+1}=j)$$

$\because$ CI

ly $d(i)$

$$= \alpha_{j\,t+1} \times \beta_{j\,t+1} \quad (2)$$

$\therefore$ Using (1) & (2),

$$\text{Ans} = \frac{\alpha_{i\,t}\ \cancel{\beta_{j\,t+1}}\ a_{ij}\ b_j O_{t+1}}{\alpha_{j\,t+1} \times \cancel{\beta_{j\,t+1}}}$$

//

Scanned with CamScanner

c) $P(S_{t-1}=i, \; S_t=k, \; S_{t+1}=j \mid O_1, \cdots O_T)$

$$= \frac{P(S_{t-1}=i, S_t=k, S_{t+1}=j, O_1 \cdots O_T)}{P(O_1, \cdots O_T)}$$

Num:-

$\alpha_{i\,t-1} \; P(S_t=k, \; S_{t+1}=j, O_t \cdots O_T \mid S_{t-1}=i, O_1 \cdots O_{t-1})$

$= \alpha_{i\,t-1} \; P(S_t=k, S_{t+1}=j, O_t \cdots O_T \mid S_{t-1}=i)$

$\because$ CIly d(i)

$$= \frac{\alpha_{i\,t-1}}{P(S_{t-1}=i)} P(S_t=k, \; S_{t+1}=j, \; O_t \cdots O_T \; S_{t-1}=i)$$

$$= \frac{\alpha_{i\,t-1}}{P(S_{t-1}=i)} \times P(S_{t+1}=j) \; P(O_{t+2} \cdots O_T \mid S_{t+1}=j)$$

$P(S_{t-1}=i, S_t=k, O_t, O_{t+1} \mid S_{t+1}=j)$

$$= \frac{\alpha_{i\,t-1}}{P(S_{t-1}=i)} \times \beta_{j\,t+2}\; P(S_{t-1}=i,\, S_t=k,\, S_{t+1}=j,\, O_t,\, O_{t+1})$$

$$= \frac{\alpha_{i\,t-1}}{\cancel{P(S_{t-1}=i)}} \times \beta_{j\,t+2} \times \cancel{P(S_{t-1}=i)}\; P(S_t=k \mid S_{t-1}=i)\; P(O_t \mid S_t=k)\; P(S_{t+1}=j \mid S_t=k)\; P(O_{t+1} \mid S_{t+1})$$

$\because$ C I by d(i) //

P.T.O

$$= \alpha_{i\,t-1}\,\beta_{j\,t+1}\,a_{ik}\,b_{k o_t}\,a_{kj}\,b_{j o_{t+1}} \quad -(1)$$

For denom :-

$$P(O_1 \cdots O_T)$$

$$= \sum_k P(S_t = k, O_1 \cdots O_T)$$

$$= \sum_k P(O_1 \cdots O_t, S_t = k)\, P(O_{t+1} \cdots O_T \mid S_t = k)$$

$$= \sum_k \alpha_{kt} \cdot \beta_{kt} \quad -(2)$$

Using (1), (2),

$$\text{Ans} = \frac{\alpha_{i\,t-1} \times \beta_{j\,t+1}\, a_{ik}\, b_{k o_t}\, a_{kj}\, b_{j o_{t+1}}}{\sum_k \alpha_{kt} \cdot \beta_{kt}}$$

//

d) $P(S_{t+1}=j \mid S_{t-1}=i, O_1 \cdots O_T)$

$$= \frac{P(S_{t-1}=i, S_{t+1}=j, O_1 \cdots O_T)}{P(S_{t-1}=i, O_1 \cdots O_T)}$$

=) For den:-

From prev ques:-

$$P(S_{t-1}=i, O_1 \cdots O_T) = \alpha_{i\,t-1}, \beta_{i\,t-1} \quad -(1)$$

c) For num:-

$$P(S_{t-1}=i, S_{t+1}=j, O_1 \cdots O_T)$$

$$\sum_k P(S_{t-1}=i, S_{t+1}=j, S_t=k, O_1 \cdots O_T)$$

From C:-

$$\sum_k \alpha_{i\,t-1} \beta_{j\,t+2} a_{ik} b_{k O_t} a_{kj} b_{j(O_{t+1})} \quad -(2)$$

Using (1) & (2),

∴ Ans:-

$$= \frac{\sum_k \cancel{\alpha_{i\,t-1}} \beta_{j\,t+1} a_{ik} b_{k O_t} a_{kj} b_j(O_{t+1})}{\cancel{\alpha_{i\,t-1}} \beta_{i\,t-1}}$$

$$= \frac{\sum_k \beta_{j\,t+1} a_{ik} b_{k O_t} a_{kj} b_j(O_{t+1})}{\beta_{i\,t-1}}$$

Q.3)

1) ~~True~~ False

2) True

3) False

4) False

5) True

6) False

7) True

8) True

9) False

10) Fase

11) True

12) True

Q.7.4)

a)

$$q_{jt} = P(S_t = j \mid O_1 \cdots O_t)$$

$$= \frac{P(S_t = j, O_1 \cdots O_t)}{P(O_1 \cdots O_t)} \quad \because \text{Prod Rule}$$

$$= \frac{\sum_i P(S_t = j, S_{t-1} = i, O_1 \cdots O_t)}{\sum_i \sum_j P(S_t = j, S_{t-1} = i, O_1 \cdots O_t)} \quad \because \text{Marginaliz}^n$$

$$= \frac{\sum_i P(S_t = j, S_{t-1} = i, O_t \mid O_1 \cdots O_{t-1})}{\sum_i \sum_j P(S_t = j, S_{t-1} = i, O_t \mid O_1 \cdots O_{t-1})} \quad \because \text{Prod. Rule}$$

For Den:-

$$P(S_t = j, S_{t-1} = i, O_t \mid O_1 \cdots O_{t-1})$$

$$= P(S_{t-1} = i \mid O_1 \cdots O_{t-1}) \times P(S_t = j \mid S_{t-1} = i) \times P(O_t \mid S_t = j) \quad \because \text{C I lly d(i)}.$$

$$= b_j(O_t)\, a_{ij}\, q_{it-1}. \quad -①$$

||ly for Num too.

$$\therefore q_{jt} = \frac{\sum_i b_j(O_t)\, a_{ij}\, q_{it-1}}{\sum_i \sum_j b_j(O_t)\, a_{ij}\, q_{it-1}}$$

$$= \frac{1}{Z_t} \times b_j(O_t) \sum_i a_{ij}\, q_{it-1}$$

//

b)
$$P(x_t \mid y_1, y_2, \cdots y_t)$$

$$= \frac{P(x_t, y_1, \cdots y_t)}{P(y_1 \cdots y_t)}$$

$$= \frac{\int dx_{t-1} \, P(x_t, x_{t-1}, y_1 \cdots y_t)}{\int dx_t \int dx_{t-1} \, P(x_t, x_{t-1}, y_1 \cdots y_t)}$$ ∵ "Marginaliz"

$$= \frac{\int dx_{t-1} \, (x_{t-1} \mid y_1 \cdots y_{t-1}) \, P(x_t \mid x_{t-1}) \, P(y_t \mid x_t)}{\int dx_t \int dx_{t-1} \, P(x_{t-1} \mid y_1 \cdots y_{t-1}) \, P(x_t \mid x_{t-1}) \, P(y_t \mid x_t)}$$

∵ Prod Rule
∵ C I by d(i)

$$= \frac{1}{Z_t} P(y_t \mid x_t) \int dx_{t-1} \, P(x_t \mid x_{t-1}) \, P(x_{t-1} \mid y_1 \cdots y_{t-1})$$

//

HP

Reason :-

If we use real values and integrate over it, it would be really diff - to keep track of it.
In contrast, when we use G V, it is easier to integrate

```
import numpy as np
from tqdm.notebook import tqdm
import matplotlib.pyplot as plt
%matplotlib inline
```

Q1 Viterbi

```
n = 27
# S = range(1,28)
# O = range(0,2)
aij = np.loadtxt("transitionMatrix.txt")
```

```
aij.shape
```

```
(27, 27)
```

```
Ot = np.loadtxt("observations.txt",dtype = "int")
```

```
Ot.shape, Ot[0]
```

```
((430000,), 0)
```

```
PIi = np.loadtxt("initialStateDistribution.txt")
```

```
PIi.shape, PIi[0]
```

```
((27,), 0.037037037037)
```

```
bik = np.loadtxt("emissionMatrix.txt")
```

```
bik.shape,bik[0]
```

```
((27, 2), array([0.96428571, 0.03571429]))
```

```
lsit = np.zeros((27,len(Ot)))
```

t = 1

```
for i in range(len(lsit)):
  lsit[i][0] = np.log(PIi[i]) + np.log(bik[i][Ot[0]])
# print(lsit)
```

t > 1

```
len(lsit)
```

```
27
```

```
lsit = np.zeros((27,len(Ot)))
phitj = np.zeros((27,len(Ot)))
for i in range(len(lsit)):
  lsit[i][0] = np.log(PIi[i]) + np.log(bik[i][Ot[0]])
# print(lsit)
T = len(Ot)
# T=10000
# argMaxi = [0]*T
for t in tqdm(range(1,T)):
  # tmp = np.zeros((27,1))
  for j in range(27):
    argMaxi = lsit[:,t-1] + np.log(aij[:,j])
    # print(phit1j.shape)
    maxi = np.argmax(argMaxi)
    phitj[j,t] = maxi

    ft = np.amax(argMaxi)
    st = np.log(bik[j,Ot[t]])
    lsit[j][t] = ft+st

  # lsit = np.hstack((lsit,tmp))

  # break


    #
    #



```

100% 429999/429999 [03:46<00:00, 1473.49it/s]

```
lsit.shape
```

```
(27, 10000)
```

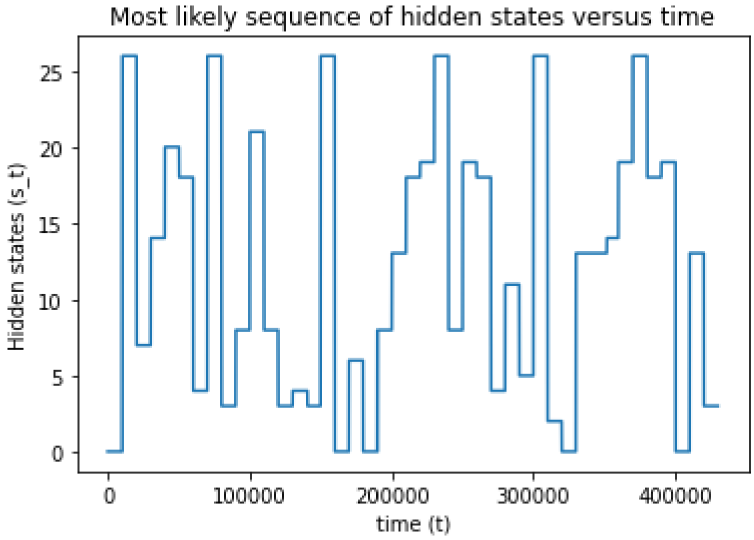
```
St = [0]*T
for i in tqdm(range(len(St)-1,-1,-1)):
  if i == T-1:
    St[i] = np.argmax(lsit[:,T-1])
  else:
    # tmpp = St[i+1]
    # tmp = phitj[i+1,St[i+1]]
    St[i] = phitj[int(St[i+1]),int(i+1)]

```

100% 430000/430000 [00:00<00:00, 762863.29it/s]

```
plt.plot(St)
plt.title('Most likely sequence of hidden states versus time')
plt.xlabel('time (t)')
plt.ylabel('Hidden states (s_t)')
```

```
Text(0, 0.5, 'Hidden states (s_t)')
```




```
st = St[0]
lmt = 0
while(True):
  lmt+=1
  if St[lmt]!=st:
    break
print(lmt)
```

```
10000
```

```
dick = {}
cnt = 0
for i in range(ord("a"),ord("z")):
  dick[cnt] = chr(i)
  cnt+=1
dick[26] = " "
verify = ""
print(dick.keys())
cnt = 0
lmt = 8000
for t in range(T-1):
  if St[t] == St[t+1]:
    cnt += 1
  else:
    cnt = 0
  if cnt > 8000:
    verify += (dick.get(St[t]))
    cnt = 0
```

```
  # if i%lmt==0:
  #   # print(i)
  #   verify+=dick[int(St[i+1])]
  #   cnt = 1



# verify+=dick[int(St[T-1])]
print(verify)
# set(verify)

```

```
dict_keys([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
a house divided against itself cannot stand
```

```

```