姓名: 葉冠宏　　學號: NTU_R11943113

結果:

Q1:

Find the probability: P(up, up, unchanged, down, unchanged, down, up|lamda:

ANS1:

0.0004967268975999999

作法:

$$\text{Induction } \alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}),$$

```
66  for i in range(1, obnum):
67      #print("round: {}".format(i))
68      newob =obseq[i]
69      ##################################################
70      #Discuss STATE1
71      #PART1: for total prob
72      P11=Probtill1 * TransPTo1From['1']
73      P12=Probtill2 * TransPTo1From['2']
74      P13=Probtill3 * TransPTo1From['3']
75
76      newProbtill1 = (P11+P12+P13) * EventP1[newob]
```

Q2:

Fnd the optimal state sequence of the model which generates the observation sequence: (up, up, unchanged, down, unchanged, down, up):

ANS2:

Optimal state sequence:

['1', '1', '3', '3', '3', '3', '1']

Largest sequence probability:

1.48176e-05

作法:

Find a best state sequence $\mathbf{s} = (s_1, s_2, .., s_T)$ for a given observation $O = (\mathbf{o}_1, \mathbf{o}_2, .., \mathbf{o}_T)$ ?

Define a new variable

$$\delta_t(i) = \max_{s_1, s_2, .., s_{t-1}} P[s_1, s_2, .., s_{t-1}, s_t = i, \mathbf{o}_1, \mathbf{o}_2, .., \mathbf{o}_t | \lambda]$$

= the best score along a single path at time $t$, which accounts for the first $t$ observation and ends in state $i$

By induction $\therefore \delta_{t+1}(j) = \left[\max_{1 \le i \le N} \delta_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1})$

$$\psi_{t+1}(j) = \arg \max_{1 \le i \le N} \delta_t(i) a_{ij}$$ .... For backtracing

We can backtrace from $s_T^* = \arg \max_{1 \le i \le N} \delta_T(i)$

```
78      #PART2: for largest prob
79      p11=LargeProb1 * TransPTo1From['1']
80      p12=LargeProb2 * TransPTo1From['2']
81      p13=LargeProb3 * TransPTo1From['3']
82
83      if (p11 > p12) and (p11> p13):
84          newMemory1 =Memory1.copy()
85          newMemory1.append('1')
86
87          newLargeProb1 = p11* EventP1[newob]
88      elif (p12 > p11) and (p12>p13):
89          newMemory1 =Memory2.copy()
90          newMemory1.append('1')
91
92          newLargeProb1 = p12* EventP1[newob]
93      else:
94          newMemory1 =Memory3.copy()
95          newMemory1.append('1')
96
97          newLargeProb1 = p13* EventP1[newob]
98
```