HMM_7

April 17, 2019

1 HMM for bioinformatics

1.1 Definition

we introduce the notion of a hidden Markov model as a stochastic machine denoted by a 6-tuple:

$$M = (Q, \alpha, P_t, q_0, q_f, P_e)$$

where

: 1. state set Q**;** 2. alphabet α;

3. transition distribution $P_t : QQ \rightarrow \mathbf{R}$;
4. initial state q_0;
5. final state q_f;
6. emission distribution $P_e : Q\alpha \rightarrow \mathbf{R}$.

It was explained that 1. a machine M operates by starting in state q_0;

2. transitioning stochastically from state to state according to $P_t(y_i|y_{i1})$, for $\{y_i, y_{i1}\} \subseteq Q$;
3. Upon entering a state q, the machine emits a symbol s according to $P_e(s|q)$;
4. terminating in state q_f.

There are no transitions into q_0 , and none out of q_f , and neither state emits any symbols.

1.2 Conventions

:1. We reserve the symbol q for particular states in the model: $Q = \{q_0, \ldots, q_{m1}\}$, for $m = |Q|$;

:2. We denote the elements of the list using some generic variable (sequence of hidden states), such as y i.e., $\phi = (y_0, y_1, \ldots, y_{n1})$ for $n = |\phi|$.

:3. For convenience, we will always assume $q_f = q_0$ that is, the 0th state in Q will always serve the function of initial and final state for the HMM;

:4. Thus, we can now denote an HMM more compactly as:

$$M = (Q, \alpha, P_t, P_e).$$

:5. We reserve the letter s for the elements of the alphabet $\alpha = \{s_0, \ldots, s_{k1}\}$ for $k = |\alpha|$.

When dealing with an input sequence S we will use a generic variable such as x to denote the individual symbols in the sequence: $S = x_0, \ldots, x_{L1}$, for $L = |S|$. Since any particular symbol s

i may occur in a sequence S zero or more times, we may have $x_i = x_j$ for $i \neq j$, whereas $s_i = s_j$ will always imply that $i = j$, since s_i is taken to be the unique name for the ith symbol in α. Thus, for s_i we take i to be an index into the alphabet α, whereas for x_j we take j to be an index into a sequence.

1.3 Representing HMMs

An HMM can be represented very simply in software by utilizing two matrices, one for the emission probabilities E and one for the transition probabilities P.

For a state set $Q = \{q_0, q_1, \ldots, q_{n1}\}$ and alphabet $\alpha = \{s_0, s_1, \ldots, s_{m1}\}$, we can utilize an $n \times m$ emission matrix, E by establishing $E_{ij} = P_e(s_j|q_i)$

Similarly, we can designate an $n \times n$ transition matrix, P, such that $P_{ij} = P_t(q_j|q_i)$.

1.4 Simple Example

Let us consider a simple example:

$$M_1 = (\{q_0, q_1, q_2\}, \{Y, R\}, P_t, P_e)$$

where

$$P_t = \{(q_0, q_1, 1), (q_1, q_1, 0.8), (q_1, q_2, 0.15), (q_1, q_0, 0.05), (q_2, q_2, 0.7), (q_2, q_1, 0.3)\}$$

and

$$P_e = \{(q_1, Y, 1), (q_1, R, 0), (q_2, Y, 0), (q_2, R, 1)\}.$$

1.4.1 representation M_1

```
In [2]: P=matrix(c(0,1,0,0.05,0.8,0.15,0,0.3,0.7), nrow=3,byrow=TRUE)
        P
```

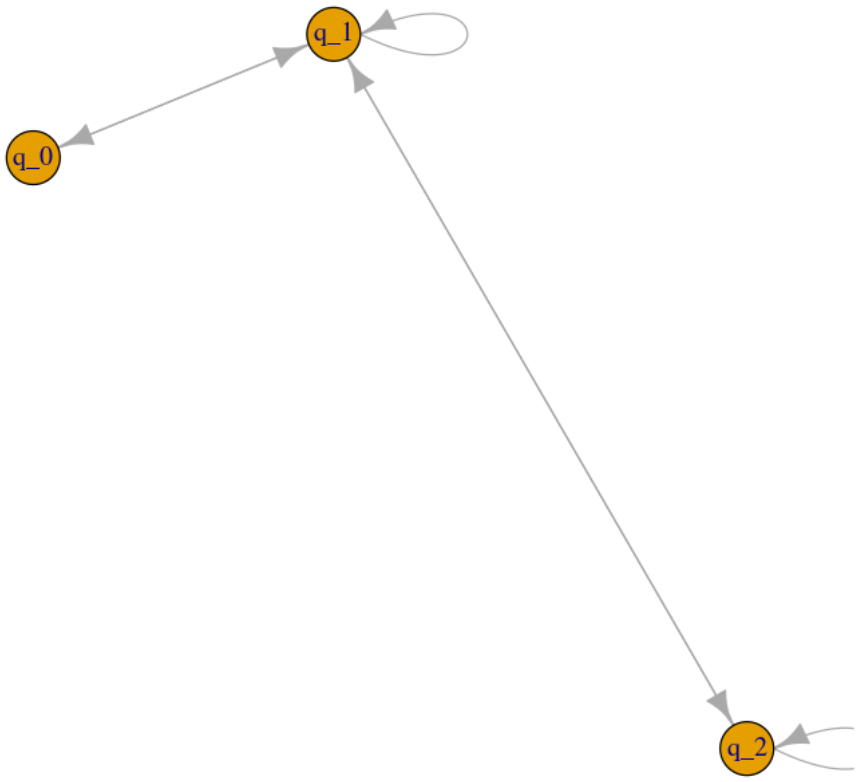
0.00	1.0	0.00
0.05	0.8	0.15
0.00	0.3	0.70

```
In [3]: E=matrix(c(1,0,0,1), nrow=2, byrow=TRUE)
        E
```

1	0
0	1

```
In [4]: source("hmm.R")
        H=c("q_0","q_1","q_2")
        O=c("Y","R")
```

```
In [6]: library("igraph")
        result <- graph_from_adjacency_matrix(P,mode="directed",weighted = TRUE)
        plot.igraph(result,vertex.label=H)
```




```
In [25]: sequence=HMM(P,E)
         sequence$hidden
         H[sequence$hidden+1]
         sequence$observed
         O[sequence$observed]
```

1. 0 2. 1 3. 1 4. 1 5. 1 6. 2 7. 1 8. 1 9. 2 10. 1 11. 1 12. 1 13. 0

1. ’q_0’ 2. ’q_1’ 3. ’q_1’ 4. ’q_1’ 5. ’q_1’ 6. ’q_2’ 7. ’q_1’ 8. ’q_1’ 9. ’q_2’ 10. ’q_1’ 11. ’q_1’ 12. ’q_1’ 13. ’q_0’

1. 1 2. 1 3. 1 4. 1 5. 2 6. 1 7. 1 8. 2 9. 1 10. 1 11. 1

1. ’Y’ 2. ’Y’ 3. ’Y’ 4. ’Y’ 5. ’R’ 6. ’Y’ 7. ’Y’ 8. ’R’ 9. ’Y’ 10. ’Y’ 11. ’Y’

1.5 Your own HMM

1.5.1 Transitions

In [56]:
```
nHidden_=5
P=matrix(sample(1:1000,25,replace=TRUE), nrow=nHidden_,byrow=TRUE)
P[1,]=c(0,1,0,0,0)
P[,1]=c(0,0,0,0,500)
temp_=rowSums(P)
#diag(1/temp_)
P=diag(1/temp_)%*%P
P
```

0.0000000	1.0000000	0.000000000	0.0000000	0.00000000
0.0000000	0.4040346	0.227089337	0.3383285	0.03054755
0.0000000	0.1017075	0.002227171	0.5820341	0.31403118
0.0000000	0.1372041	0.171597633	0.3313609	0.35983728
0.1684069	0.1748063	0.304816437	0.2000674	0.15190300

1.5.2 Emissions

In [63]:
```
nObserved_=4
E=matrix(sample(1:nObserved_^2,nObserved_^2,replace=TRUE), nrow=nObserved_, byrow=TRU
temp_=rowSums(E)
E=diag(1/temp_)%*%E
E
```

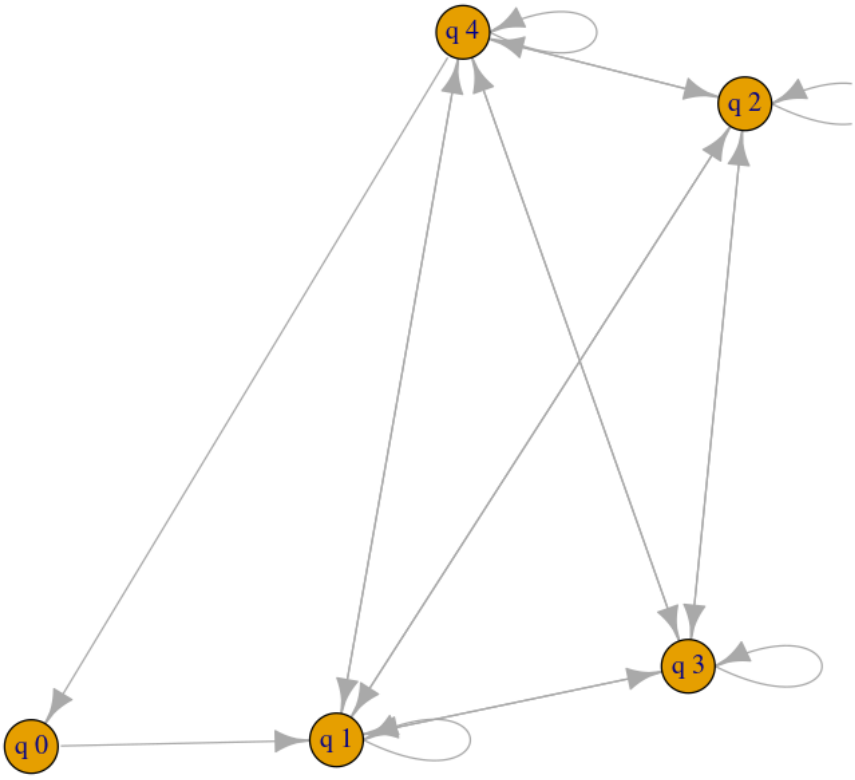
0.2647059	0.4411765	0.2058824	0.08823529
0.1200000	0.4400000	0.0800000	0.36000000
0.1875000	0.2500000	0.2500000	0.31250000
0.3478261	0.1521739	0.2826087	0.21739130

1.5.3 The graphical representation

In [72]:
```
library("igraph")
H=paste(rep("q",nHidden_),0:(nHidden_-1))
H

result <- graph_from_adjacency_matrix(P,mode="directed",weighted = TRUE)
plot.igraph(result,vertex.label=H)
```

1. 'q 0' 2. 'q 1' 3. 'q 2' 4. 'q 3' 5. 'q 4'



1.5.4 HMM sequence

```
In [79]: sequence=HMM(P,E)
         O=letters[1:nHidden_]
         sequence$hidden
         H[sequence$hidden+1]
         sequence$observed
         O[sequence$observed]
```

1. 0 2. 1 3. 1 4. 3 5. 1 6. 1 7. 1 8. 3 9. 4 10. 1 11. 1 12. 1 13. 1 14. 2 15. 3 16. 3 17. 4 18. 4 19. 3 20. 4 21. 1 22. 2 23. 3 24. 1 25. 2 26. 3 27. 3 28. 4 29. 1 30. 1 31. 3 32. 4 33. 0

1. ’q 0’ 2. ’q 1’ 3. ’q 1’ 4. ’q 3’ 5. ’q 1’ 6. ’q 1’ 7. ’q 1’ 8. ’q 3’ 9. ’q 4’ 10. ’q 1’ 11. ’q 1’ 12. ’q 1’ 13. ’q 1’ 14. ’q 2’ 15. ’q 3’ 16. ’q 3’ 17. ’q 4’ 18. ’q 4’ 19. ’q 3’ 20. ’q 4’ 21. ’q 1’ 22. ’q 2’ 23. ’q 3’ 24. ’q 1’ 25. ’q 2’ 26. ’q 3’ 27. ’q 3’ 28. ’q 4’ 29. ’q 1’ 30. ’q 1’ 31. ’q 3’ 32. ’q 4’ 33. ’q 0’

1. 4 2. 3 3. 3 4. 3 5. 1 6. 2 7. 2 8. 4 9. 2 10. 3 11. 1 12. 1 13. 4 14. 4 15. 4 16. 4 17. 4 18. 4 19. 4 20. 1 21. 2 22. 3 23. 3 24. 1 25. 4 26. 4 27. 4 28. 1 29. 3 30. 2 31. 1

1. 'd' 2. 'c' 3. 'c' 4. 'c' 5. 'a' 6. 'b' 7. 'b' 8. 'd' 9. 'b' 10. 'c' 11. 'a' 12. 'a' 13. 'd' 14. 'd' 15. 'd' 16. 'd' 17. 'd' 18. 'd' 19. 'd' 20. 'a' 21. 'b' 22. 'c' 23. 'c' 24. 'a' 25. 'd' 26. 'd' 27. 'd' 28. 'a' 29. 'c' 30. 'b' 31. 'a'

2 The three basic problems for HMMs

: 1. Given the observation sequence $S = x_1, x_2, \ldots, x_k$ and the model $M = (Q, \alpha, q_0, P_t, P_e)$ how do we efficiently compute $P(S|M)$, the probability of the observation sequence, given the model?

: 2. Given the observation sequence $S = x_1, x_2, \ldots, x_k$ and the model $M = (Q, \alpha, q_0, P_t, P_e)$ how do we choose a corresponding hidden state sequence $y_1, y_2, \ldots, y_k$ which is optimal in some meaningful sense?

: 3. How do we adjust the model parameters $M = (Q, \alpha, q_0, P_t, P_e)$ to maximize $P(S|M)$?

2.1 The probability of $P(S|M_1)$

Because each nonsilent state in this HMM can emit only one of the two symbols in the alphabet, we can compute the probability that any given run of M_1 results in a given sequence by multiplying together the transition and emission probabilities. We have

$$P(YRYRY|M_1) = a_{0\to1} \times b_{1,Y} \times a_{1\to2} \times b_{2,R} \times a_{2\to1} \times b_{1,Y} \times a_{1\to2} \times b_{2,R} \times a_{2\to1} \times b_{1,Y}$$

where $a_{i\to j}$ denotes $P_t(q_j|q_i)$ whereas $b_{i,s}$ denotes $P_e(s|q_i)$.

```
In [7]: P=matrix(c(0,1,0,0.05,0.8,0.15,0,0.3,0.7), nrow=3,byrow=TRUE)
        P
```

0.00	1.0	0.00
0.05	0.8	0.15
0.00	0.3	0.70

```
In [8]: E=matrix(c(1,0,0,1), nrow=2, byrow=TRUE)
        E
```

1	0
0	1

```
In [9]: sequence=HMM(P,E)
        sequence$hidden
        H[sequence$hidden+1]
        sequence$observed
        O[sequence$observed]
```

1. 0 2. 1 3. 2 4. 1 5. 1 6. 1 7. 1 8. 1 9. 2 10. 2 11. 2 12. 2 13. 2 14. 2 15. 2 16. 1 17. 1 18. 1 19. 1 20. 1 21. 1 22. 1 23. 0

1. 'q_0' 2. 'q_1' 3. 'q_2' 4. 'q_1' 5. 'q_1' 6. 'q_1' 7. 'q_1' 8. 'q_1' 9. 'q_2' 10. 'q_2' 11. 'q_2' 12. 'q_2' 13. 'q_2' 14. 'q_2' 15. 'q_2' 16. 'q_1' 17. 'q_1' 18. 'q_1' 19. 'q_1' 20. 'q_1' 21. 'q_1' 22. 'q_1' 23. 'q_0'

1. 1 2. 2 3. 1 4. 1 5. 1 6. 1 7. 1 8. 2 9. 2 10. 2 11. 2 12. 2 13. 2 14. 2 15. 1 16. 1 17. 1 18. 1 19. 1 20. 1 21. 1

1. 'Y' 2. 'R' 3. 'Y' 4. 'Y' 5. 'Y' 6. 'Y' 7. 'Y' 8. 'R' 9. 'R' 10. 'R' 11. 'R' 12. 'R' 13. 'R' 14. 'R' 15. 'Y' 16. 'Y' 17. 'Y' 18. 'Y' 19. 'Y' 20. 'Y' 21. 'Y'

```
In [10]: sequence$observed
         Forward(P,E,sequence$observed)
```

1. 1 2. 2 3. 1 4. 1 5. 1 6. 1 7. 1 8. 2 9. 2 10. 2 11. 2 12. 2 13. 2 14. 2 15. 1 16. 1 17. 1 18. 1 19. 1 20. 1 21. 1

1.27903709999923e-06

```
In [31]: (0.8^3)*0.15*0.3*0.8*0.15*0.3*0.8*0.8*0.05
```

2.654208e-05

Where is the problem?

2.1.1 The Forward Algorithm

: 1. A procedure very similar to the Viterbi algorithm can be used to find the probability that a given model M

2. Because M may potentially emit S via any number of paths through the states of the model, to compute the full probability of the sequence we need to sum over all possible paths emiting S.

$$F(i,k) = \begin{cases} 1 & for\ k = 0,\ i = 0 \\ 0 & for\ k > 0,\ i = 0 \\ 0 & for\ k = 0,\ i > 0 \\ \sum_{j=0}^{|Q|-1} F(j,k-1) P_t(q_i|q_j) P_e(x_k|q_i) & for\ 1 \leq k \leq |S|, \\ & 1 \leq i \leq |Q| \end{cases}$$