Machine learning homework

Hong Pengfei 1002949

November 2019

1 Q1

Q is set of all possible state, and V is set of all possible observation:

$$\begin{aligned} &Q = \{q_1, q_2, ..., q_n\} = \{< sos >, X, Y, Z, < eos >\} \\ &V = \{v_1, v_2, ..., v_m\} = \{a, b, c\} \end{aligned} \tag{1}$$

The parameters associated with HMM are:
Transition matrix A:

$$\begin{aligned} &A = [a_{ij}]_{N \times N} \\ &a_{ij} = P(i_{t+1} = q_j | i_t = q_i), i = 1, 2, ..., N; j = 1, 2, ..., N \end{aligned} \tag{2}$$

Emission matrix B:

$$\begin{gathered} B = [b_j(k)]_{N \times M} \\ b_j(k) = P(o_t = v_k | i_t = q_j), k = 1, 2..., M; j = 1, 2, ...N \end{gathered} \tag{3}$$

1.1 Calculation - Transition Matrix

number_of_count for transition $count(u, v)$:

transition_from\transition_to	<eos>	X	Y	Z
<sos>	0	2	0	3
X	1	0	3	2
Y	4	1	0	1
Z	0	3	3	0

number of count for each state $count(u)$:

<sos>	<eos>	X	Y	Z
5	5	6	6	6

using maximum loglikelihood we get: transition matrix $a_{u,v} = \frac{count(u,v)}{count(u)}$:

transition_from \transition_to	<eos>	X	Y	Z
<sos>	0	2/5	0	3/5
X	1/6	0	3/6	2/6
Y	4/6	1/6	0	1/6
Z	0	3/6	3/6	0

1.2 Calculation – Emission Matrix

emission count $count(u \to v)$:

state \observance	a	b	c
X	1	3	2
Y	2	0	4
Z	1	2	3

emission matrix $b_u(o) = \frac{count(u \to o}{count(u)}$

state \observance	a	b	c
X	1/6	3/6	2/6
Y	2/6	0	4/6
Z	1/6	2/6	3/6

2 Q2

viterbi decoding equation: $\pi(j+1, u) = max_v\{\pi(j, v) * a_{v,u}\}$ to calculate score matrix:

step \state	<sos>	X	Y	Z	<eos>
START	1	0	0	0	0
1	0	0.2	0	0.2	0
2	0	0.25	0	0.02222	0
STOP	0	0	0	0	0.042

Therefore by decoding the path from the scores using $y_n^* = argmax_j\{\pi(j, u) \cdot a_{u,y_{j+1}^*}\}$
We can decode the path to be: $START \to Z \to X \to STOP$

3 Q3

By changing the emission score $b_u(x_j)$ to $b_u(x_j) * p_u(x_j)$ can solve the problem. where $p_u(x_j)$ is the prior probability that **State u** generates **Word** x_j. After timing this, each word will have a new emission score that consider the prior distribution, for example if $p_u(x_j) = 0$ then the emission score will be set to zero.

4 Q4

4.1 forward path

Defintion: $\alpha_u(j) = P(x_1, ..., x_{j-1}, y_1, ..., y_{j-1}, z = u)$
Base case: $\alpha_u(1) = a_{start,u}$
iteration: $\alpha_u(j+1) = \sum_v \alpha_v(j) a_{v,u} b_v(x_j) b_v(y_j)$

4.2 backward path

Definition:$\beta_u(j) = P(x_j, ..., X_n, y_j, ..., y_n | z_j = u)$
base case: $\beta_u(n) = \alpha_{u,stop} b_u(x_n) b_u(y_n)$
iteration: $\beta_u(j) = \sum_v a_{u,v} b_u(x_j) b_u(y_j) \beta_v(j+1)$

The time complexity of this algorithm is:
$O(nN^2)$
Analyses:
To calculate score for each $\beta_u(j)$ we need to loop through all the number of tags N, since we have nN beta scores to calculate, the total time complexity is $O(N) * O(nN) = O(nN^2)$