Advanced Machine Learning - Assignment 2

Ragja Palakkadavath - SC18M003

January 2019

1. **Solution:**
   Given

   Model TM

   $\pi_{1=} \begin{bmatrix} 0.7 & 0.3 \end{bmatrix}$

   $A_{TM} = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}$

   Model EG

   $\pi_{2=} \begin{bmatrix} 0.6 & 0.4 \end{bmatrix}$

   $A_{EG} = \begin{bmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{bmatrix}$

   $P(HHLHH/\lambda_{TM}) = 0.7 \times 0.6 \times 0.2 \times 0.4 \times 0.6 = 0.02016$

   $P(HHLHH/\lambda_{EG}) = 0.6 \times 0.7 \times 0.6 \times 0.3 \times 0.7 = 0.05292$

   Model EG has higest probability to generate the given sequence.

2. **Solution:**
   Given Model $\lambda = (A, B, \pi)$

   $A = \begin{bmatrix} .2 & .2 & .6 \\ .3 & .3 & .4 \\ .2 & .5 & .3 \end{bmatrix} B = \begin{bmatrix} .7 & .3 \\ .5 & .5 \\ .8 & .2 \end{bmatrix} \pi = \begin{bmatrix} .3 & .4 & .3 \end{bmatrix}$

   (a) State sequences to be evaluated for exhaustive search: S1 S1, S1 S2, S1 S3, S2 S1, S2 S2, S2 S3, S3 S1, S3 S2, S3 S3
   Using exhastive search
   $P(v_1 v_2/\lambda) = \sum_{i=1}^{3} \sum_{j=1}^{3} \pi_i b_{s_i}(v_1)\, a_{s_i s_j} b_{s_j}(v_2)$
   $P(RG/\lambda) = \sum_{i=1}^{3} \sum_{j=1}^{3} \pi_i b_{s_i}(R)\, a_{s_i s_j} b_{s_j}(G)$
   $= 0.0126 + 0.021 + 0.0252 + 0.018 + 0.03 + 0.016 + 0.0144 + 0.06 + 0.0144 = 0.2116$

   (b) Forward Procedure:

   $\alpha_1(1) = P[v_2, q_1 = s_1/\lambda] = 0.3 * 0.3 = 0.09$

   $\alpha_1(2) = P[v_2, q_1 = s_2/\lambda] = 0.4 * 0.5 = 0.2$

   $\alpha_1(3) = P[v_2, q_1 = s_3/\lambda] = 0.3 * 0.2 = 0.06$

   $\alpha_2(1) = P[v_2 v_2, q_2 = s_1/\lambda] = 0.09 * 0.2 * 0.3 + 0.2 * 0.3 * 0.3 + 0.06 * 0.2 * 0.3 = 0.027$

   $\alpha_2(2) = P[v_2 v_2, q_2 = s_2/\lambda] = 0.09 * 0.2 * 0.5 + 0.2 * 0.3 * 0.5 + 0.06 * 0.5 * 0.5 = 0.054$

   $\alpha_2(3) = P[v_2 v_2, q_2 = s_3/\lambda] = 0.09 * 0.6 * 0.2 + 0.2 * 0.4 * 0.2 + 0.06 * 0.3 * 0.2 = 0.0304$

   $P[v_2 v_2/\lambda] = 0.027 + 0.04 + 0.0304 = 0.1114$

Backward Procedure:

$\beta_2(1) = P[O_3/q_1 = s_1, \lambda] = 1$

$\beta_2(2) = P[O_3/q_1 = s_2, \lambda] = 1$

$\beta_2(3) = P[O_3/q_1 = s_3, \lambda] = 1$

$\beta_1(1) = P[v_2/q_2 = s_1, \lambda] = 1.0 * 0.2 * 0.3 + 1.0 * 0.2 * 0.5 + 1.0 * 0.6 * 0.2 = 0.28$

$\beta_1(2) = P[v_2/q_2 = s_2, \lambda] = 1.0 * 0.3 * 0.3 + 1.0 * 0.3 * 0.5 + 1.0 * 0.4 * 0.2 = 0.32$

$\beta_1(3) = P[v_2/q_2 = s_3, \lambda] = 1.0 * 0.2 * 0.3 + 1.0 * 0.5 * 0.5 + 1.0 * 0.3 * 0.2 = 0.37$

$P[v_2 v_2/\lambda] = 0.3 * 0.3 * 0.28 + 0.4 * 0.5 * 0.32 + 0.3 * 0.2 * 0.37 = 0.1114$

(c) Vieterbi Algorithm: for sequence v_1, v_1, v_2

$\delta_1(1) = \pi_1 * b_1(v_1) = 0.3 * 0.7 = 0.21$

$\psi_1(1) = 0$

$\delta_1(2) = \pi_2 * b_2(v_1) = 0.4 * 0.5 = 0.2$

$\psi_1(2) = 0$

$\delta_1(3) = \pi_3 * b_3(v_1) = 0.3 * 0.8 = 0.24$

$\psi_1(3) = 0$

$\delta_2(1) = max_i \delta_1(i) * a_{ij} b_j(v_1) = max(0.21 * 0.2 * 0.7, 0.2 * 0.3 * 0.7, 0.24 * 0.2 * 0.7) = 0.042$

$\psi_2(1) = 2$

$\delta_2(2) = max_i \delta_1(i) * a_{ij} b_j(v_1) = max(0.21 * 0.2 * 0.5, 0.2 * 0.3 * 0.5, 0.24 * 0.5 * 0.5) = 0.06$

$\psi_2(2) = 3$

$\delta_2(3) = max_i \delta_1(i) * a_{ij} b_j(v_1) = max(0.21 * 0.6 * 0.8, 0.2 * 0.4 * 0.8, 0.24 * 0.3 * 0.8) = 0.1008$

$\psi_2(3) = 1$

$\delta_3(1) = max_i \delta_2(i) * a_{ij} b_j(v_1) = max(0.042 * 0.2 * 0.3, 0.06 * 0.3 * 0.3, 0.1008 * 0.2 * 0.3) = 0.006048$

$\psi_3(1) = 3$

$\delta_3(2) = max_i \delta_2(i) * a_{ij} b_j(v_1) = max(0.042 * 0.2 * 0.5, 0.06 * 0.3 * 0.5, 0.1008 * 0.5 * 0.5) = 0.0252$

$\psi_3(2) = 3$

$\delta_3(3) = max_i \delta_2(i) * a_{ij} b_j(v_1) = max(0.042 * 0.6 * 0.2, 0.06 * 0.4 * 0.2, 0.1008 * 0.3 * 0.2) = 0.006048$

$\psi_3(3) = 3$

Optimal δ value $= 0.0252$

Best Sequence: $[1 \rightarrow 3 \rightarrow 2]$

3. **Solution:**
Given Model $\lambda = (A, B, \pi)$

$$A = \begin{bmatrix} .5 & .5 \\ .3 & .7 \end{bmatrix} B = \begin{bmatrix} .6 & .2 & .2 \\ .2 & .5 & .3 \end{bmatrix} \pi = \begin{bmatrix} .8 & .2 \end{bmatrix}$$

(a) State sequences to be evaluated for exhaustive search: S1 S1, S1 S2, S2 S1, S2 S2
Using exhaustive search
$P(v_1 v_2/\lambda) = \sum_{i=1}^{2} \sum_{j=1}^{2} \pi_i b_{s_i}(v_1)\, a_{s_i s_j} b_{s_j}(v_2)$
$P(RG/\lambda) = \sum_{i=1}^{2} \sum_{j=1}^{2} \pi_i b_{s_i}(R)\, a_{s_i s_j} b_{s_j}(G)$
Each of the state sequence probabilites are given by:
$pi(S_1) * b(S_1 R) * a(S_1 S_1) * b(S_1 G) = 0.048$
$pi(S_1) * b(S_1 R) * a(S_1 S_2) * b(S_2 G) = 0.12$

$pi(S_2) * b(S_2R) * a(S_2S_1) * b(S_1G) = 0.0024$
$pi(S_2) * b(S_2R) * a(S_2S_2) * b(S_2G) = 0.014$
$P(RG/\lambda) = 0.048 + 0.12 + 0.0024 + 0.014 = 0.1844$

(b) Forward Procedure:

$\alpha_1(1) = P[v_1, q_1 = s_1/\lambda] = 0.8 * 0.6 = 0.48$

$\alpha_1(2) = P[v_1, q_1 = s_2/\lambda] = 0.2 * 0.2 = 0.04$

$\alpha_2(1) = P[v_1v_2, q_2 = s_1/\lambda] = 0.48 * 0.5 * 0.2 + 0.04 * 0.3 * 0.2 = 0.0504$

$\alpha_2(2) = P[v_1v_2, q_2 = s_2/\lambda] = 0.48 * 0.5 * 0.5 + 0.04 * 0.7 * 0.5 = 0.134$

$P[v_1v_2/\lambda] = 0.0504 + 0.134 = 0.1844$

Backward Procedure:

$\beta_2(1) = P[O_3/q_1 = s_1, \lambda] = 1$

$\beta_2(2) = P[O_3/q_1 = s_2, \lambda] = 1$

$\beta_1(1) = P[v_2/q_2 = s_1, \lambda] = 1 * 0.5 * 0.2 + 1 * 0.3 * 0.5 = 0.35$

$\beta_1(2) = P[v_2/q_2 = s_2, \lambda] = 1 * 0.5 * 0.2 + 1 * 0.7 * 0.5 = 0.41$

$P[v_1v_2/\lambda] = 0.8 * 0.6 * 0.35 + 0.2 * 0.2 * 0.41 = 0.1844$

(c) Vieterbi Algorithm: for sequence v_1, v_2

$\delta_1(1) = \pi_1 * b_1(v_1) = 0.8 * 0.6 = 0.48$

$\psi_1(1) = 0$

$\delta_1(2) = \pi_2 * b_2(v_1) = 0.2 * 0.2 = 0.04$

$\psi_1(2) = 0$

$\delta_2(1) = max_i \delta_1(i) * a_{ij} b_j(v_1) = max(0.48 * 0.5 * 0.2, 0.04 * 0.3 * 0.2) = 0.048$

$\psi_2(1) = 1$

$\delta_2(2) = max_i \delta_1(i) * a_{ij} b_j(v_1) = max(0.48 * 0.5 * 0.5, 0.04 * 0.7 * 0.5) = 0.12$

$\psi_2(2) = 1$

$Optimal \delta value = 0.12$

$BestSequence : [1 \rightarrow 2]$

4. **Solution:**

Given Model $\lambda = (A, B, \pi)$

$A = \begin{bmatrix} .7 & .3 \\ .2 & .8 \end{bmatrix} B = \begin{bmatrix} .1 & .2 & 0 & 0.5 & 0.2 \\ .3 & .2 & .4 & 0 & 0.1 \end{bmatrix} \pi = \begin{bmatrix} .5 & .5 \end{bmatrix}$

(a) Optimal Sequence of States that generates the sequence:
NARNRDCCRN is :- [H- H- H- H- H- H+ H+ H+ H- H-]

δ value obtained from Viterbi Algorithm for the optimal sequence: 1.47975045e-08

(b) Forward Procedure Evaluation:
$P[NARNRDCCRN/\lambda] = 9.501926400000006e - 08$

Backward Procedure Evaluation:
$P[NARNRDCCRN/\lambda] = 9.501926400000006e - 08$

(c) Diagrams of the given terms:

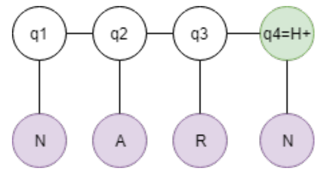

Figure 1: $\alpha_4(H+)$

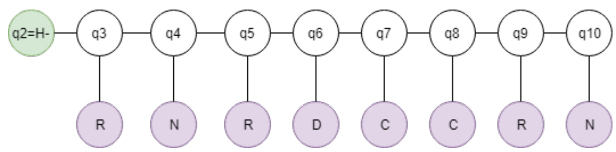

Figure 2: $\beta_2(H-)$

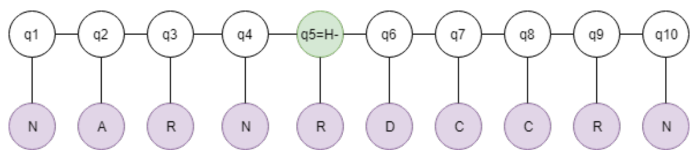

Figure 3: $\gamma_5(H-)$

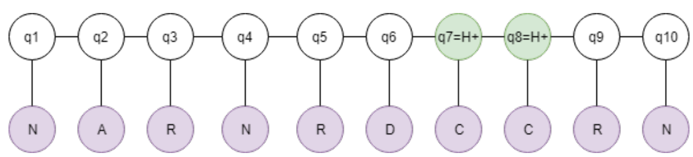

Figure 4: $\xi_7(H+H+)$

5. **Solution:**

Model that best describes the sequence: [HHTTTHHHH HHTHTHTHTTTTT] is :

$$A = \begin{bmatrix} 0.79 & 2.19e^{-04} & 0.22 & 5.48e^{-04} \\ 7.39e^{-05} & 6.72e^{-12} & 0.99 & 1.01e^{-29} \\ 0.17 & 0.823 & 5.58e^{-12} & 1.99e^{-16} \\ 8.69e^{-07} & .99 & 2.20e^{-15} & 1.43e^{-15} \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0 & 5.61e^{-15} \\ 0.54 & 0.46 \\ 1.36e^{-11} & 1.0 \\ 1.0 & 3.36e^{-17} \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.240 & 0.7.60 & 1.28e^{-14} & 4.71e^{-07} \end{bmatrix}$$

Solution was obtained using Baum-Weltch Algorithm.

6. **Solution:**

Short notes on Graph Laplacian: Given any weighted graph, $G = (V, W) with V = v_1, v_2 ... v_m$ and $W = [w_{ij}]$, i, j=1...m, the unnormalized Graph Laplacian of G is defined by:

$$L(G) = D(G) - W,$$

where $D(G) = diag(d_1, d_2, d_3 .... d_m)$ is the degree matrix of G where $d_i = \sum_{j=1}^{m} w_{ij}$

L is a positive semi definite matrix. The smallest eigenvalue of L is 0 and the corresponding eigenvector is [1 1....1 1].

If a graph has k disconnected components, then the dimension of the null space of L, N(L) is k and is spanned by $1_{A_1}, 1_{A_2} ...... 1_{A_k}$ where $A_1 .... A_k$ are the components of the graph

For any f $\in R^{\mathrm{N}}$, $f^T L f = \frac{1}{2} \sum_{i=1}^{N} w_{ij} (f_i - f_j)^2$

7. **Solution:**

Separate your data sets into the data sets for each class. Train one HMM on data from each of the class separately. On the test set compare the likelihood of each model to classify each of the test data point. Assign the test point to that label which has the produced the highest likelihood.

Example:

Assume that there is a set of V labels for classification. Set of N tokens of each of the label is available as the training set and an independent testing set.

HMM model is built for each of the different labels in the classification task. Using the observations from the training data, the optimum parameter for each label can be estimated by giving the model λ_v for the v^{th} label. Each unknown data point in the test set is characterized by an observation sequence, $O = o_1, o_2 ..... o_t$ and for each $P_v = P(O/\lambda_v)$ using Baum-Welch algorithm. Finally, choose the label for which the model probability is the highest for the given test observation.

8. **Solution:**

Unnormalized spectral clustering for Data 1

(a) Number of clusters=2 There was a sudden jump from the values of eigenvalues from the order of 10^{-14} to 4 from the 2^{nd} eigenvalue to the 3^{rd} eigenvalue when it was plotted.Hence number of clusters was selected to be 2.

(b) Plot of Eigenvalues

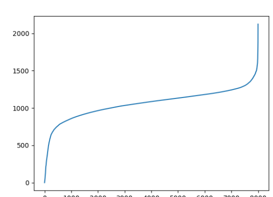

Figure 5: Eigenvalue plot of Data1

(c) First k Eigenvectors and Eigenvalues of Laplacian Matrix of Data 1 is present in : $data1\_eigenvalues.csv$ and $data1\_eigenvectors.csv$

(d) Plot of Cluster

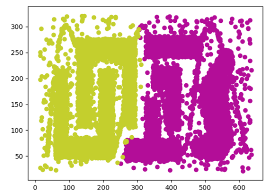

Figure 6: Cluster plot of Data1

Unnormalized spectral clustering for Data 2

(a) Number of clusters=3 There was a sudden jump from the values of eigenvalues from the order of 10^{-16} to 10^{-3} from the 3^{rd} eigenvalue to the 4^{th} eigenvalue when it was plotted.Hence number of clusters was selected to be 3.

(b) Plot of Eigenvalues

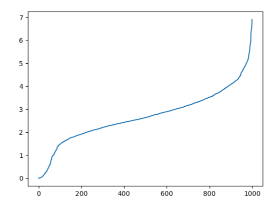

Figure 7: Eigenvalue plot of Data2

(c) First k Eigenvectors and Eigenvalues of Laplacian Matrix of Data 2 is present in : $data2\_eigenvalues.csv$ and $data2\_eigenvectors.csv$

(d) Plot of Cluster

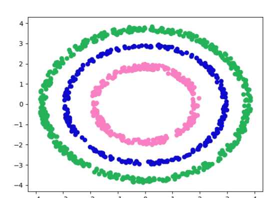

Figure 8: Cluster plot of Data2

Normalized spectral clustering for Data 1

(a) Number of clusters=2 There was a sudden jump from the values of eigenvalues from the order of 10^{-14} to 4 from the 2^{nd} eigenvalue to the 3^{rd} eigenvalue when it was plotted.Hence number of clusters was selected to be 2.

(b) Plot of Eigenvalues

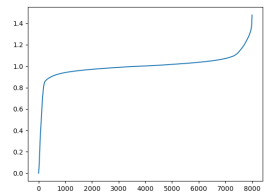

Figure 9: Eigenvalue plot of Data1

(c) First k Eigenvectors and Eigenvalues of Laplacian Matrix of Data 1 is present in : *data1_eigenvalues_normalized.csv* and *data1_eigenvectors_normalized.csv*

(d) Plot of Cluster

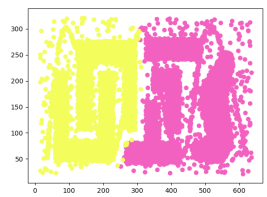

Figure 10: Cluster plot of Data1

Normalized spectral clustering for Data 2

(a) Number of clusters=3 There was a sudden jump from the values of eigenvalues from the order of 10^{-16} to 10^{-3} from the 3^{rd} eigenvalue to the 4^{th} eigenvalue when it was plotted.Hence number of clusters was selected to be 3.

(b) Plot of Eigenvalues

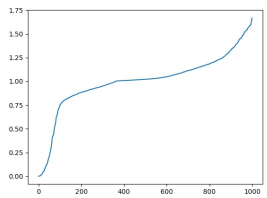

Figure 11: Eigenvalue plot of Data2

(c) First k Eigenvectors and Eigenvalues of Laplacian Matrix of Data 2 is present in : *data2_eigenvalues_normalized.csv* and *data2_eigenvectors_normalized.csv*

(d) Plot of Cluster

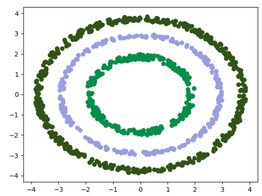

Figure 12: Cluster plot of Data2

9. **Solution:**

(a) Number of clusters=4 There was a sudden jump from the values of eigenvalues from the order of 10^{-16} to 10^{-3} from the 4^{th} eigenvalue to the 5^{th} eigenvalue when it was plotted.Hence number of clusters was selected to be 4.

(b) First k Eigenvectors and Eigenvalues of Laplacian Matrix of Data 2 is present in : $RNA\_eigenvalues.csv$ and $RNA\_eigenvectors.csv$