Programming Assignment3

CS669: Pattern Recognition (Feb-Jun 2018)

Sukesh Kumar Das (Roll no. d17025)

Joe Johnson (Roll no. d17024)

Group 2

School of Computing and Electrical Engineering

Indian Institute of Technology Mandi

Email:skd_sentu@rediffmail.com

joe4robotics@gmail.com

---

1 Introduction

In the given assignment, two real world dataset type of scene image and speech with both having three classes are given. 24 dimensional color histogram features are extracted from scene image data corresponding to three different classes and 39 dimensional MFCC feature has been given for speech dataset. Decisions should be based fundamentally on task or cost specific. To achieve minimum cost path optimal alignment between two different length sequence is always essential which is achieved by dynamic time warping or hidden markov model. For all the datasets, the classifier used is Bayes classifier which is known statistical pattern recognition. As per the given asignment, two classifiers are to be built-(i) bayes classifier using k-nearest neighbour(kNN) method for class-conditional density estimation and (ii) bayes classifier using discrete HMM(DHMM). Experiments are performed with different values of k for kNN, different states(N) and different number of codebooks(M) for DHMM. kNN is a non- parametric learning algorithm which is used for classification as well as density estimation. Since kNN is non-parametric, it can perform density estimation for arbitrary distributions and unlike other parametric methods. It does not require any knowledge of distribution apriori. Also, there is no explicit training phase in this algorithm and all training data is required for testing which makes it computationally burden.

In k-nearest neighbour(kNN) method for density estimation, dynamic time warping(DTW) distance is used. The DTW alignment is carried out between a test feature and every train features. The distances are then sorted in

asending order and first k- nearest examples are choosed from sorted list. For all experiments in kNN, class label is done by using eq1.

$$Class\ label\ for\ \bar{x} \equiv arg\ \max_i p(\bar{x}/D_i) \equiv arg\ \max_i \frac{k_i}{N_i V} \tag{1}$$

Where k_i is the number of examples of class i out of k-nearest neighbours. N_i is the no of examples of class i. V is the volume that contains first k no of examples from the sorted list.

DTW is a template matching technique to find the distance between two different length continuous valued feature vectors. At event level it gets compressed or expanded but not always uniform. This technique find the best alignment by warping. In algorithm, if the two sequence having length of T_m and T_n are aligned by following the algorithm.

$$g(i,j) = \begin{cases} dist(1,1); & \text{for i=1 and j=1} \\ g(i-1,1) + dist(i,1); & \text{for i=2 to } T_m \text{ and j=1} \\ g(1,j-1) + dist(1,j); & \text{for i=1 and j=2 to } T_n \\ dist(i,j) + min\{g(i-1,j), g(i-1,j-1), g(i,j-1)\}; & \text{for i=2 to } T_m \text{ and j=2 to } T_n \end{cases} \tag{2}$$

$$Normalizeddistance, DTW(T_m, T_n) = \frac{1}{T_m T_n} g(T_m, Tn). \tag{3}$$

Hidden markov models(HMM) is a better way of DTW matching [1].One of the most powerful properties of HMM is their ability to exhibit some degree of invariance to local warping (compression and stretching) of the time axis. In this technique we take information from every clip/file and build a model of an event. Each of event gives some kind of state specifically transition state(i.e within state or one state to another transition). In HMM, sequence of observations is known but state sequence is unknown i.e. in which state the observations belongs to are not exactally known. Three things describe the HMM model- State transition probability(a_{ij}), state observation symbol probability or emission probability($b_j(v_k)$) and initial probability(π_i). HMM model for class c is denoted by $\lambda_c = \{N_c, M_c, A_c, B_c, \bar{\pi}_c\}$. Where N is the no of states, M is no of observation symbols, state probability matrix,$A = [a_{ij}]_{NxN}$. a_{ij} is the probability of transition to state j being at state i, emission probability matrix,$B = [b_j(v_k)]_{NxM}$. a_{ij}, initial state probability vector,$\bar{\pi} = [\pi_1, \pi_2 ... \pi_N]^T$

There are two types of HMM- (i) Ergodic HMM and (ii) Left to right HMM. Ergodic HMM includes forward and backward transitions. Left to right HMM includes only forward transitions. It is mainly used in Speech and video classification.

In the given assignment we implement the discrete HMM where events are in sequential order. It involves three problems- (i) estimating parameters-$\{A, B, \bar{\pi}\}$,(ii) Evaluation problem - probability of observation sequence for a given model($p(O/\lambda_c)$). (iii) optimal state sequence-viterbi alignment.

1.1 Forward Procedure

Forward variable,$\alpha_t(j)$ which is the probability of generating partial observation sequence untill time t and being at state j at time t for a given model, is calculated as

$$\alpha_t(j) = P(O_1O_2....O_T, q_t = j/\lambda) = \begin{cases} \pi_j b_j(t); & \text{if t=1} \\ (\sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij})b_j(O_t); & \text{if j=1,2...N and 1}\leq\text{t}\leq\text{T} \end{cases} \tag{4}$$

Termination: $P(O/\lambda) = \sum_{j=1}^{N} \alpha_T(j)$

1.2 Backward Procedure

Backward variable,$\beta_t(i)$ which is the probability of generating partial observation sequence($O_{t+1}O_{t+2}...O_T$) given state at time t and the model λ, is calculated as

$$\beta_t(i) = P(O_{t+1}O_{t+2}....O_T/q_t = i, \lambda) = \begin{cases} 1; & \text{if t=T and } \forall \text{ i=1,2...N} \\ (\sum_{j=1}^{N} a_{ij}b_j(O_{t+1})\beta_{t+1}(i); & \forall \text{ i=1,2...N. and } \forall \text{ t=T-1,T-2,...1} \end{cases} \tag{5}$$

Termination: $P(O/\lambda) = \sum_{j=1}^{N}(\pi_j b_j(O_1)\beta_1)\beta_1(i)$.

1.3 Optimal sequence

Otherway it is called viterbi alignment. For a observation sequence $O = (O_1O_2..O_M)$ and the modelλ, how to choose a corresponding state sequence $Q = (q_1, q_2...q_T)$. It better explains the observation in som sense. Log of the best probability(score) along the single path at time t, which accounts for the first t observations and ends at state i, is given as

$$\tilde{\delta}_t(j) = \begin{cases} log\{\delta_1(j)\} = \tilde{\pi}_j + \tilde{b}_j(O_1); & \forall \text{ j and t=1} \\ log\{\delta_t(j)\} = \max_i[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij} + \tilde{b}_j(O_t); & \text{Where } \forall \text{ j=1,2...N} \end{cases} \tag{6}$$

Termination: $\tilde{p}^* = arg\ \max_i\{\tilde{\delta}_T(j)\}$.

Optimal state at time t, is given as

$$\psi_t(j) = \begin{cases} 0; & \text{when t=1} \\ arg\ \max_c\ [\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}]; & \forall \text{ j=1,2...N} \end{cases} \tag{7}$$

$Class\ for\ O = arg\ \max_c\ P_c^*$.

1.4 E-M Method for HMM

It is also called Baum-Welch re-estimation procedure. In E-step, probability($\zeta_{st}^l(i,j)$) of being at state i at time t and state j at time (t+1) given observation sequence(for all observation) and model(λ), is claculated as

$$\zeta_{st}^l(i,j) = \frac{\alpha_t^l(i) a_{ij} b_j(O_{t,t+1}) \beta_{t+1}^l(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t^l(i) a_{ij} b_j(O_{t,t+1}) \beta_{t+1}^l(j)}. \tag{8}$$

posterior probability($\gamma_t^l(i)$) of being at state i at time t given observation(all l observation) squence and model(λ), is calculated as

$$\gamma_{st}^l(i) = \frac{\alpha_t^l(i)\beta_t^l(i)}{\sum_{i=1}^N \alpha_t^l(i)\beta_t^l(i)}.\forall \text{ l=1,2...L} \tag{9}$$

In M- step re-estimation of the parameters(λ^{new}) using current values($\zeta_{st}^l(i,j) and$) is done.

$$\hat{\pi}_{st}^l(i,j) = \frac{1}{L}\sum_{l=1}^{L} \gamma_1^l(i).\forall \text{ i=1,2...N.} \tag{10}$$

$$\hat{a}_{ij}^{new} = \frac{\sum_{l=1}^L \sum_{t=1}^{T-1} \zeta_{st}^l(i,j)}{\sum_{l=1}^L \sum_{t=1}^{T-1} \gamma_t^l(j)}. \tag{11}$$

$$\hat{b}_j^{new}(V_k) = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T}\gamma_{t|O_t=V_k}^{l}(j)}{\sum_{l=1}^{L}\sum_{t=1}^{T}\gamma_t^l(j)}.\forall \text{ j=1,2...N} \tag{12}$$

Finally likelihood $P(D/\lambda^{new})$ is computed and convergence criteria is checked. If it is not satisfied then $\lambda^{old} \leftarrow \lambda^{new}$ and again repeated from step2.

2 Bayes Classifier: kNN using DTW distance

Here in building all kNN classifiers, distances are measured using DTW template matching technique which is common technique to calculate the distances between variable length sequences. Classifier is implemented using different no of k. When k=1, it is called nearest neighbour method and for $k = \infty$,it acts as Bayes classifier. The classifier is used for two types of dataset- (i) Scene image data from which histogram feature is extracted and (ii) MFCC features representing speech data.

2.1 Scene image dataset

For given three class scene image data, every image is resized to a nearest size which is multiple of 32. It is done because 32x32 patch is taken to extract 24 dimentional histogram features from color image.

2.1.1 Performance Analysis of kNN classifier for scene image data

Table 1 shows the confusion matrices(for k=3) and the table 2 shows the total performance of the kNN classifier for different values of k and higest performance is highlighted.

Table 1

Confusion matrix when histogram of scene image data is considered for kNN classifier for k=3.

		Actual			
		Coast	Kennel	V court	Total
Prediction	Coast	44	10	7	61
	Kennel	6	39	13	58
	V court	0	1	30	31
	Total	50	50	50	

The performance measure of the optimal value of k(3) interms of accuracy, recall, precesion and F-score is obtained by using eq(13-24).

$$Accuracy\ of\ the\ system = \frac{No\ of\ examples\ correctly\ classified}{Total\ no\ of\ test\ samples} \times 100\% == 75.33\% \tag{13}$$

$$Recall\ for\ class\ Coast = \frac{True\ Positive\ for\ Class\ Coast}{Total\ Actual\ Coast} = 0.88. \tag{14}$$

$$Precision\ for\ class\ Coast = \frac{True\ Positive\ for\ Class\ Coast}{Total\ Predicted\ Coast} = 0.721. \tag{15}$$

$$F - measure\ for\ class\ Coast = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.792. \tag{16}$$

$$Recall\ for\ class\ Kennel = \frac{True\ Positive\ for\ Class\ Kennel}{Total\ Actual\ Kennel} = 0.78. \tag{17}$$

$$Precision\ for\ class\ Kennel = \frac{True\ Positive\ for\ Class\ Kennel}{Total\ Predicted\ Kennel} = 0.7213. \tag{18}$$

$$F - measure\ for\ class\ Kennel = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.722. \tag{19}$$

$$Recall\ for\ class\ VC = \frac{True\ Positive\ for\ Class\ VC}{Total\ Actual\ VC} = 0.60. \tag{20}$$

$$Precision\ for\ class\ VC = \frac{True\ Positive\ for\ Class\ VC}{Total\ Predicted\ VC} 0.97. \tag{21}$$

$$F - measure\ for\ class\ VC = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.74. \tag{22}$$

$$Mean\ recall\ = \frac{1}{M}\sum_{i=1}^{M} Recall for\ class\ C_i = 0.75. Where\ M = \ No.\ of\ classes. \tag{23}$$

$$Mean\ precision\ = \frac{1}{M}\sum_{i=1}^{M} Precision for\ class\ C_i = 0.79. Where\ M = \ No.\ of\ classes. \tag{24}$$

Table 2

Performance Analysis of kNN classifier for scene image data with different k:

Sl. No	k	Accuracy	Recall(Coast,Kenne,VC,Mean)	Precesion(Coast,Kenne,VC,Mean)	F-measure(Coast,Kenne,VC)
1	1	74.67	0.86,0.72,0.66,0.75	0.75,0.61,0.97,0.78	0.80,0.66,0.79
2	3	75.33	0.88,0.78,0.60,0.75	0.72,0.67,0.97,0.79	0.79,0.72,0.74
3	5	73.33	0.86,0.84,0.50,0.73	0.75,0.64,0.93,0.77	0.80,0.72,0.64
4	9	72	0.88,0.90,0.38,0.72	0.83,0.59,0.90,0.78	0.85,0.71,0.53
5	13	74	0.90,0.88,0.44,0.74	0.85,0.60,0.92,0.79	0.87,0.72,0.59

2.1.2 Observation

kNN yields highly non-linear boundary and hence over all it works well in all kind of data classification. But it has to bear with lot of computational complexity. For the given scene image data the classifier works well while k=3 by achiving the accuracy 75.33%.

2.2 Consonant Vowel (CV) segment dataset (Speech)

Extracted MFCC features from three classes(TI,tI,ti) of data are provided for executing the assignment. kNN classifier is built using DTW distance and performances are measured for different values of k.

2.2.1 Performance Analysis of kNN classifier for speech data

Table 3 shows the confusion matrices(for k=13) and the table 4 shows the total performance of the kNN classifier for different values of k and higest performance is highlighted.

Table 3

Confusion matrix when speech data is considered for kNN classifier for k=13.

		Actual			
		TI	iT	ti	Total
Prediction	TI	10	4	0	14
	iT	5	27	8	40
	it	6	15	28	49
	Total	21	46	36	

The performance measure of the optimal value of k(13) interms of accuracy, recall, precesion and F-score is obtained by using eq(25-36).

$$Accuracy\ of\ the\ system = \frac{No\ of\ examples\ correctly\ classified}{Total\ no\ of\ test\ samples} \times 100\% = 63.10\% \tag{25}$$

$$Recall\ for\ class\ TI = \frac{True\ Positive\ for\ Class\ tI}{Total\ Actual\ ti} = 0.48. \tag{26}$$

$$Precision\ for\ class\ TI = \frac{True\ Positive\ for\ Class\ TI}{Total\ Predicted\ TI} = 0.71. \tag{27}$$

$$F - measure\ for\ class\ TI = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.57. \tag{28}$$

$$Recall\ for\ class\ tI = \frac{True\ Positive\ for\ Class\ tI}{Total\ Actual\ tI} = 0.59. \tag{29}$$

$$Precision\ for\ class\ tI = \frac{True\ Positive\ for\ Class\ tI}{Total\ Predicted\ tI} = 0.68. \tag{30}$$

$$F - measure\ for\ class\ tI = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.63. \tag{31}$$

$$Recall\ for\ class\ ti = \frac{True\ Positive\ for\ Class\ ti}{Total\ Actual\ ti} = 0.78. \tag{32}$$

$$Precision\ for\ class\ ti = \frac{True\ Positive\ for\ Class\ ti}{Total\ Predicted\ ti} = 0.57. \tag{33}$$

$$F - measure\ for\ class\ ti = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.69. \tag{34}$$

$$Mean\ recall\ = \frac{1}{M}\sum_{i=1}^{M} Recall for\ class\ C_i = 0.61. Where\ M = \ No.\ of\ classes. \tag{35}$$

$$Mean\ precision\ = \frac{1}{M}\sum_{i=1}^{M} Precision for\ class\ C_i = 0.65. Where\ M = \ No.\ of\ classes. \tag{36}$$

Table 4

Performance Analysis of kNN classifier for speech data:

Sl. No	k	Accuracy	Recall(TI,tI,ti,Mean)	Precesion(TI,tI,ti,Mean)	F-measure(TI,tI,ti)
1	1	52.43	0.62,0.37,0.67,0.55	0.68,0.55,0.45,0.56	0.65,0.44,0.54
2	3	57.28	0.67,0.37,0.78,0.60	0.54,0.65,0.55,0.58	0.60,0.47,0.64
3	5	61.17	0.57,0.52,0.75,0.61	0.60,0.65,0.59,0.61	0.59,0.58,0.66
4	9	57.28	0.43,0.50,0.75,0.60	0.60,0.59,0.55,0.58	0.50,0.54,0.64
5	13	63.10	0.48,0.59,0.78,0.61	0.71,0.68,0.57,0.65	0.57,0.62,0.66

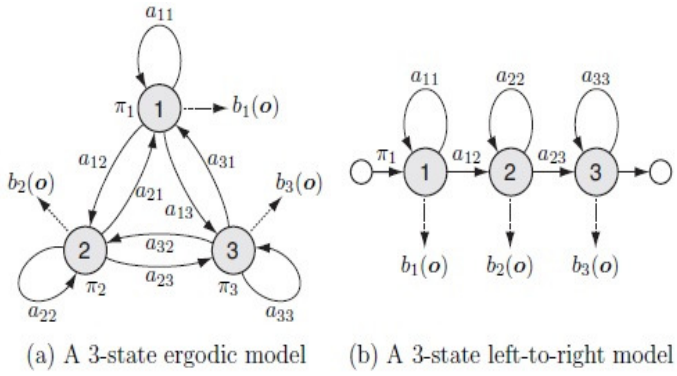
2.2.2 Observation

kNN yields highly non-linear boundary and hence over all it works well in all kind of data classification. But it has to bear with lot of computational complexity. For the given MFCC feature of speech data of three classes, the classifier

works well when k=13 by achiving the accuracy 63.10%. But kNN classifies scene image data than speech data.

3 Bayes classifier using Discrete HMM (DHMM)

Here in the assignment, for scene image data, ergodic HMM is implemented because it involves forward and backward transitions and for MFCC feature we implement left to right HMM because speech generally involves only forward transitions. For making the HMM ergodic, full transition matrix is used and to get left to right HMM, transition matrix is taken as an upper triangular matrix. Experiments are carrier out with different numbers of hidden states(N) and different numbers of visible symbols(M).The classifier is used for two types of dataset- (i) Scene image data from which histogram feature is extracted and (ii) MFCC features representing speech data. Figure 2 shows the ergodic and left to right DHMM and fig2 shows the block diagram of DHMM cassifier.



(a) A 3-state ergodic model (b) A 3-state left-to-right model

Fig. 1. Three state DHMM.

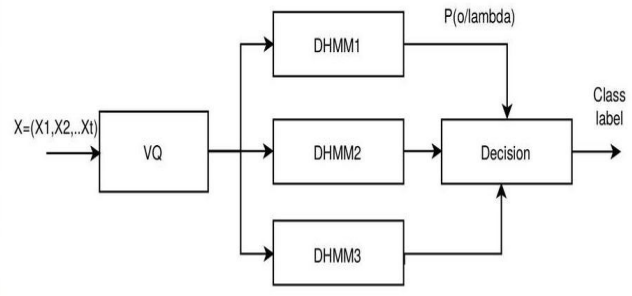


Fig. 2. Block diagram of DHMM classifier.

3.1 Scene image dataset

For given three class scene image data, every image is resized to the nearest size which is multiple of 32. It is done because 32x32 patch is taken to extract 24 dimentional histogram features from color image.

3.1.1 Performance Analysis of DHMM classifier for scene image data

Table 5 shows the confusion matrices(for M=2,N=3) and the table 6 shows the total performance of the DHHM classifier for different values of M and N and higest performance is highlighted.

Table 5

Confusion matrix when histogram feature of scene image data is considered for ergodic DHMM classifier for optimal value of N(3) and M(2):

		Actual			
		Coast	Kennel	V court	Total
Prediction	Coast	35	23	16	74
	Kennel	6	14	5	25
	V court	9	13	29	51
	Total	50	50	50	

The performance measure of the optimal value of N(3) and M(2) interms of accuracy, recall, precesion and F-score is obtained by using eq(37-48).

$$Accuracy\ of\ the\ system = \frac{No\ of\ examples\ correctly\ classified}{Total\ no\ of\ test\ samples} \times 100\% = 52\% \tag{37}$$

$$Recall\ for\ class\ Coast = \frac{True\ Positive\ for\ Class\ Coast}{Total\ Actual\ Coast} = 0.70. \tag{38}$$

$$Precision\ for\ class\ Coast = \frac{True\ Positive\ for\ Class\ Coast}{Total\ Predicted\ Coast} = 0.47. \tag{39}$$

$$F - measure\ for\ class\ Coast = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.56. \tag{40}$$

$$Recall\ for\ class\ Kennel = \frac{True\ Positive\ for\ Class\ Kennel}{Total\ Actual\ Kennel} = 0.28. \tag{41}$$

$$Precision\ for\ class\ Kennel = \frac{True\ Positive\ for\ Class\ Kennel}{Total\ Predicted\ Kennel} = 0.56. \tag{42}$$

$$F - measure\ for\ class\ Kennel = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.37. \tag{43}$$

$$Recall\ for\ class\ VC = \frac{True\ Positive\ for\ Class\ VC}{Total\ Actual\ VC} = 0.58. \tag{44}$$

$$Precision\ for\ class\ VC = \frac{True\ Positive\ for\ Class\ VC}{Total\ Predicted\ VC} 0.57. \tag{45}$$

$$F - measure\ for\ class\ VC = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.56. \tag{46}$$

$$Mean\ recall\ = \frac{1}{M} \sum_{i=1}^{M} Recall for\ class\ C_i = 0.52. Where\ M = \ No.\ of\ classes. \tag{47}$$

$$Mean\ precision\ = \frac{1}{M} \sum_{i=1}^{M} Precision for\ class\ C_i = 0.53. Where\ M = \ No.\ of\ classes. \tag{48}$$

Table 6

Performance Analysis of ergodic DHMM classifier for scene image data with different N and M:

Sl. No	N,M	Accuracy	Recall(Coast,Kenne,VC,Mean)	Precesion(Coast,Kenne,VC,Mean)	F-measure(Coast,Kenne,VC)
1	2,2	26	0.02,0.76,0,0.26	0.08,0.28,-,-	0.03,0.41,0.03
2	2,4	19.33	0.04,0.54,0,0.19	0.06,0.23,-,-	0.05,0.33,0.05
3	2,8	22.66	0,0.54,0.14,0.23	0,0.23,0.37,0.20	-,0.33,-
4	2,16	42.40	0.58,0.26,0.46,0.43	0.43,0.62,0.38,0.47	0.49,0.37,0.49
5	4,2	26.67	0,0.8,0,0.27	0,0.29,-,-	-,0.27,-
6	4,4	14.42	0.04,0.26,0.16,0.15	0.06,0.18,0.20,0.14	0.05,0.21,0.04
7	4,8	20.00	0.04,0.54,0.02,0.20	0.11,0.23,0.08,0.14	0.06,0.32,0.06
8	4,16	43.33	0.54,0.36,0.40,0.43	0.47,0.31,0.59,0.46	0.50,0.33,0.50
9	4,32	44.66	0.74,0.16,0.44,0.45	0.42,0.73,0.43,0.53	-,-,0.51
10	6,8	35.33	0,0.3,0.76,0.35	0,0.28,0.46,0.25	-,0.29,-
11	6,16	39.33	0.82,0.2,0.16,0.39	0.38,0.77,0.29,0.48	0.52,0.37,0.52
12	6,32	34.66	0.62,0.40,0.02,0.35	0.41,0.32,0.11,0.28	0.49,0.35,0.49
13	8,8	40	0.02,0.22,0.96,0.40	0.06,0.39,0.46,0.30	0.03,0.22,0.96
14	8,16	32	0.34,0.16,0.46,0.32	0.30,0.25,0.37,0.30	0.32,0.20,0.32
15	8,32	50.40	0.7,0.28,0.58,0.52	0.47,0.56,0.57,0.53	0.56,0.37,0.56
16	10,16	47.33	0.22,0.32,0.88,0.47	0.34,0.64,0.47,0.49	0.27,0.43,0.27
17	10,32	37.33	0.76,0.3,0.06,0.39	0.35,0.48,0.33,0.39	0.48,0.37,0.47

3.1.2 Observation

HMM is very well known for modeling sequence informations. To extract feature from every image, we consider 32X32 patches which contains no much sequence information. Despite for the given scene image data, the classifier works well while N=8 and M=32 by achiving the accuracy 50.40%.

3.2 Consonant Vowel (CV) segment dataset (Speech)

Extracted MFCC features from three classes(TI,tI,ti) of data are provided for executing the assignment. Ergodic DHMM is built and performances are measured for different values of M and N.

3.2.1 Performance Analysis of DHMM classifier for speech data

Table 7 shows the confusion matrices(for N=3,M=2) and the table 8 shows the total performance of the DHMM classifier for different values of N and M and higest performance is highlighted.

Table 7

Confusion matrix when speech data is considered for left to right DHMM classifier for N=3 and M=2:

		Actual			
		TI	iT	ti	Total
Prediction	TI	0	0	1	1
	iT	17	34	19	70
	it	4	12	16	32
	Total	21	46	36	

The performance measure of the optimal value of N(3) and M(2) interms of accuracy,recall,precesion and F-score is obtained by using eq(49-60).

$$Accuracy\ of\ the\ system = \frac{No\ of\ examples\ correctly\ classified}{Total\ no\ of\ test\ samples} \times 100\% = 48.54\% \tag{49}$$

$$Recall\ for\ class\ TI = \frac{True\ Positive\ for\ Class\ tI}{Total\ Actual\ ti} = 0. \tag{50}$$

$$Precision\ for\ class\ TI = \frac{True\ Positive\ for\ Class\ TI}{Total\ Predicted\ TI} = 0. \tag{51}$$

$$F - measure\ for\ class\ TI = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = NaN. \tag{52}$$

$$Recall\ for\ class\ tI = \frac{True\ Positive\ for\ Class\ tI}{Total\ Actual\ tI} = 0.74. \tag{53}$$

$$Precision\ for\ class\ tI = \frac{True\ Positive\ for\ Class\ tI}{Total\ Predicted\ tI} = 0.49. \tag{54}$$

$$F - measure\ for\ class\ C2 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.59. \tag{55}$$

$$Recall\ for\ class\ ti = \frac{True\ Positive\ for\ Class\ ti}{Total\ Actual\ ti} = 0.44. \tag{56}$$

$$Precision\ for\ class\ ti = \frac{True\ Positive\ for\ Class\ ti}{Total\ Predicted\ ti} = 0.50. \tag{57}$$

$$F - measure\ for\ class\ ti = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = 0.47. \tag{58}$$

$$Mean\ recall\ = \frac{1}{M}\sum_{i=1}^{M} Recall for\ class\ C_i = 0.39. Where\ M = \ No.\ of\ classes. \tag{59}$$

$$Mean\ precision\ = \frac{1}{M}\sum_{i=1}^{M} Precision for\ class\ C_i = 0.32. Where\ M = \ No.\ of\ classes. \tag{60}$$

Table 8

Performance Analysis of left to right DHMM classifier for speech data:

Sl. No	N,M	Accuracy	Recall(TI,tI,ti,Mean)	Precesion(TI,tI,ti,Mean)	F-measure(TI,tI,ti)
1	3,2	48.54	0,0.74,0.44,0.39	0,0.49,0.50,0.33	-,0.59,0.47
2	3,4	33.00	0.33,0.11,0.61,0.35	0.35,0.36,0.32,0.34	0.34,0.17,0.42
3	3,8	31.06	0.09,0,0.83,0.31	0.13,-,0.34,-	0.11,-,0.48
3	3,16	44.66	0,1,0,-	-,0.45,-,-	-,0.62,-
4	4,2	33.98	0,0,0.97,0.32	0,0,0.35,0.12	-,-,0.51
5	4,4	20.39	1,0,0,0.33	0.20,-,-,-	0.34,-,-
6	4,8	33.01	0,0.06,0.86,0.30	-,0.25,0.34,-	-,0.10,0.49
7	4,16	20.39	1,0,0,0.33	0.20,-,-,-	0.34,-,-
8	5,2	33.98	0,0.13,0.81,0.31	0,0.29,0.37,0.22	-,0.18,0.51
9	5,4	23.30	0.81,0.15,0,0.32	0.23,0.25,-,-	0.35,0.19,-
10	5,8	39.80	0,0.17,0.91,0.36	-,0.42,0.39,-	-,0.25,0.55
11	5,16	20.38	1,0,0,0.33	0.20,-,-,-	0.34,-,-

3.2.2 Observation

Left to right HMM is very well known in modelling the phonemes in speech. The classifier gives best result when number of states is 3 and nunber of visible symbols is 2. However the result is not as good as scene image case. It may be because of limited data for modelling at training period. There are many parameters are to be calculated which may optimally require a huge amount of training data. For DTW, size of the training dataset was sufficient enough to increase the confidence of classification.HMM also have other issues like it assumes the successive observations in a sequence to be independent. Although, this makes calculation of likelihood very simple as it is simply product of probabilities, the assumption may not be always true. Also, there is no formal method to determine whether the

model is ergodic or let-to-right. Apart from those limitations, discrete-HMM can model especially the phonemes in speech signals in a betterway. By considering good amount of data and CHMM, better result may be achieved.

References

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.