Pattern Recognition
Assignment 4
Group No - 36

ED13D000 Bhargava Sai Ramu
CS17S008 Nitesh Methani

November 8, 2017

Contents

1 Objectives

1. Recognize Digits : Given the MFCC feature vector of a digit uttered by a speaker, classify it as 1, 2 or 3 using HMM and DTW.

2. Recognize hand written Telugu characters : Given the X and Y coordinates of a Telugu character written by someone, recognize it as 'a', 'ta' or 'da'.

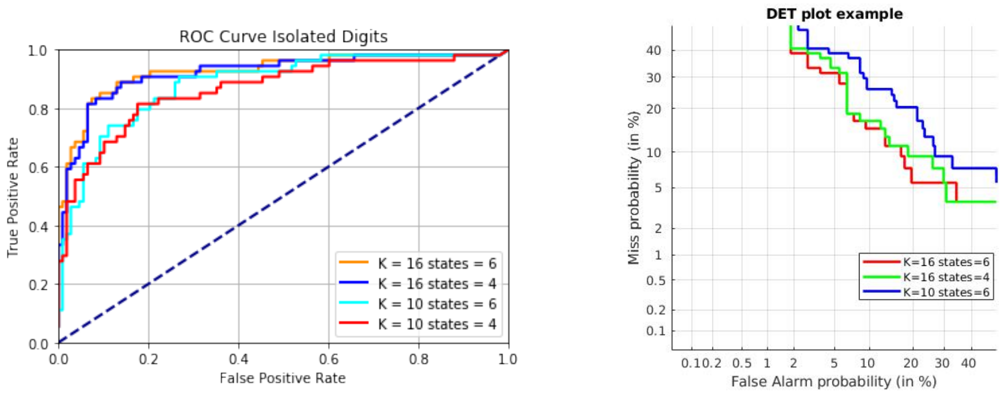
2 Hidden Markov Model

2.1 Isolated Digits

Table 1 is the summary of class wise accuracy of digits 1, 2, 3 for different number of states and symbols.

Symbols	States	1	2	3
16	6	100%	100%	97%
16	4	100%	100%	97%
16	2	100%	57%	97%
10	6	100%	97%	93%
10	4	97%	97%	90%
10	2	100%	97%	90%
6	6	100%	97%	90%
6	4	100%	97%	97%

Table 1: Digits Class wise Accuracy



(a) ROC Curve for K=16, Q=6 (b) DET Curve

Figure 1: Plots for Isolated Digits

2.2 Connected Digits

Table 2 is the prediction of our model for the given test data. Initially we had 3 HMM models for 3 digits. We concatenated them with different permutations to get 27 different HMM models. We didn't use dummy state. We modified the transition probabilities of the last state of the individual models such that probability to remain in that state is 0.5 and probability to transit to first state of the next model is also 0.5. The symbol emission probabilities were adjusted accordingly.

Test 1	Predicted	Test 2(files)	Predicted
11322a	d2_d3_d2_	342	d2_d2_d1_
132a	d3_d2_d1_	343	d3_d2_d1_
13a	d3_d2_d2_	344	d1_d3_d1_
212a	d2_d1_d1_	345	d1_d2_d1_
21a	d2_d1_d2_	346	d3_d1_d2_
22123a	d3_d3_d2_	347	d3_d2_d1_
22a	d2_d2_d1_	348	d1_d3_d1_
233a	d2_d2_d2_	349	d1_d2_d1_
23a	d1_d1_d1_	350	d3_d1_d2_
311a	d1_d1_d2_	351	d1_d2_d2_
31a	d1_d1_d3_	352	d1_d1_d2_
322a	d1_d2_d1_	353	d2_d1_d2_
331a	d1_d2_d2_	-	-
33a	d3_d1_d2_	-	-

Table 2: Connected Digits Recognition

3 Recognition of Hand Written data

3.1 Feature Extraction

Size Normalization: Size of the characters varies from writer to writer due to different writing styles. Therefore to make all the samples of uniform size we did size normalization to improve the recognition performance. Now the character data are normalized to fit into a box of unit length. Other features extracted are as follows:

1. Features

   (a) Normalized Coordinate Values : Normalized X and Y coordinates are used as features. Reason : They represent the shape of the character in abstract format.

   (b) Deviation Features : Characters vary in size and shape with the variation in character trajectories either in horizontal or vertical directions. So deviation features are extracted.

   (c) Zero Mean Feature : Distance of the horizontal and vertical coordinate values from the horizontal and vertical means is calculated. Since data points are subtracted from the data mean, the resulting data is named as Zero mean features.

   (d) Trajectory Features : While writing a character handwriting forms a trajectory which comprises of angle and distance parameters. Distance and angle parameter from the origin of the Cartesian Coordinate System and the horizontal axis respectively. Distance and angle parameter from the centroid. Distance and angle parameter between the consecutive points.

2. Other Features

   (a) Curvature

   (b) Cosine Distances

   (c) Slope

   (d) First Derivative

   (e) Second Derivative

   (f) Normalized First Derivative

   (g) Normalized Second Derivative

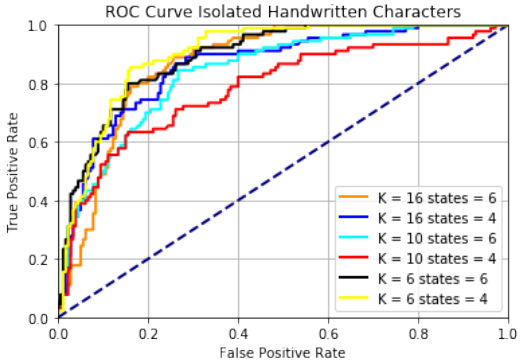
The values in the extracted feature are normalizd to fit in a window size of [0, 1].

3.2 Isolated Handwritten Characters

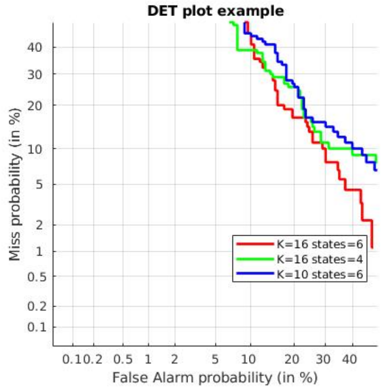
Table 3 is the summary of class wise accuracy of handwritten characters a, dA, tA for different number of states and symbols.

Symbols	**States**	**a**	**dA**	**tA**
16	6	100%	94%	100%
16	4	100%	94%	100%
16	2	100%	94%	100%
10	6	94%	100%	100%
10	4	89%	94%	100%
10	2	89%	89%	100%
6	6	94%	94%	94%
6	4	94%	83%	100%
6	2	94%	89%	100%

Table 3: Handwritten Characters Class wise Accuracy



(a) ROC Curve



(b) DET Curve

Figure 2: Plots for Isolated Handwritten Characters

3.3 Connected Handwritten Characters

Table 4 is the prediction of our model for the given test data. Initially we had 3 HMM models for 3 characters. We concatenated them with different permutations to get 27 different HMM models. We didn't use dummy state. We modified the transition probabilities of the last state of the individual models such that probability to remain in that state is 0.5 and probability to transit to first state of the next model is also 0.5. The symbol emission probabilities were adjusted accordingly.

Given Characters	Predicted
dA tA dA	tA tA a
dA tA a	a tA a
dA dA dA	a dA a

Table 4: Connected Handwritten Characters Recognition

4 Dynamic Time Warping

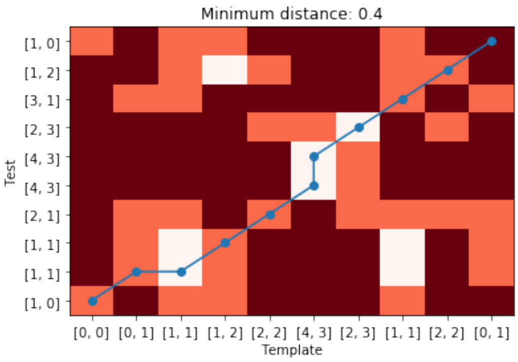
4.1 Observations

Table 5 is the summary of accuracy of normalized and raw data using brute force and representative method.

DTW	Accuracy	Time
Raw Data	100%	00:11:53:41
Zero-mean	38.89%	00:09:32:57
Min-Max Normalization	42.52%	00:09:44.80
Using Representatives	98.84%	00:01:49:37
Vector Quantization	48.14%	00:11:38.18

Table 5: Recognition Rates

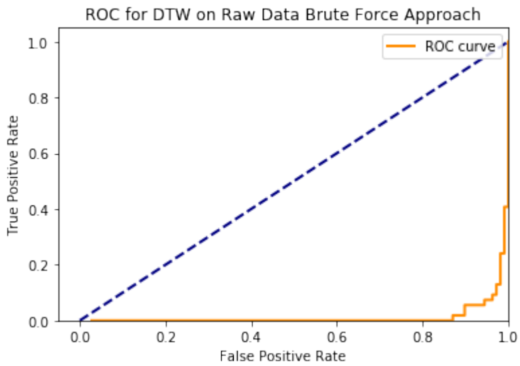
4.2 Observations



(a) Example DTW

```
Confusion matrix, without normalization
[[18  0  0]
 [ 0 18  0]
 [ 0  0 18]]
Normalized confusion matrix
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]]
```

(b) Confusion Matrix on Raw Data



(c) ROC Curve

Figure 3: Plots for DTW

Representatives are as follows:
Class 1 : 30^{th} and 31^{st} template
Class 2 : 31^{st} and 22^{nd} template
Class 3 : 15^{th} and 8^{th} template

5 Coding Style Improvement

- Writing Object Oriented Code for re-usability.
- Writing shell scripts to avoid writing commands again and again on the terminal.
- Added modularity while writting code.
- Practiced multi-file programming style and some regular expressions for file handing.
- Better handling images and tables in latex.

6 Conclusion

In this assignment we generated HMM models for different number of symbols and states and trained them on digits 1, 2 and 3 and characters a, tA and dA for the respective questions. After testing them independently we then connected them to recognize the connected strings. We also did the recognition task using DTW method. One of the mistakes that we were doing was to do K-Means on the test data again for assigning cluster labels.