CS5691 Pattern Recognition and Machine Learning
Programming Assignment 3

Team 16 - Akshith CS19B005, Shashank CS19B043

April 7, 2022

1 K-means, Gaussian Mixture Model

1.1 Image Dataset

We modelled each class using a GMM with k mixtures (for different values of k). We pooled all the blocks from every image of that particular class and estimated GMM parameters using them. If there are n images in a class, and 36 blocks per image, then there would be a total of $36 \cdot n$ 23-dimensional vectors for estimating the parameters of the GMM corresponding to that class. The initial values of priors, means and covariances are taken from the result of the k-means algorithm.

For a test image, we calculated the likelihood for every block belonging to each class, and classified the test image as belonging to that class which maximized the sum of log likelihoods of all the blocks.
If $p_1, p_2, \ldots, p_{36}$ are the probabilities of each block of the test image for a given class, then the class where $\log p_1 + \log p_2 + \cdots + \log p_{36}$ is maximized is chosen as the predicted class for that test image.

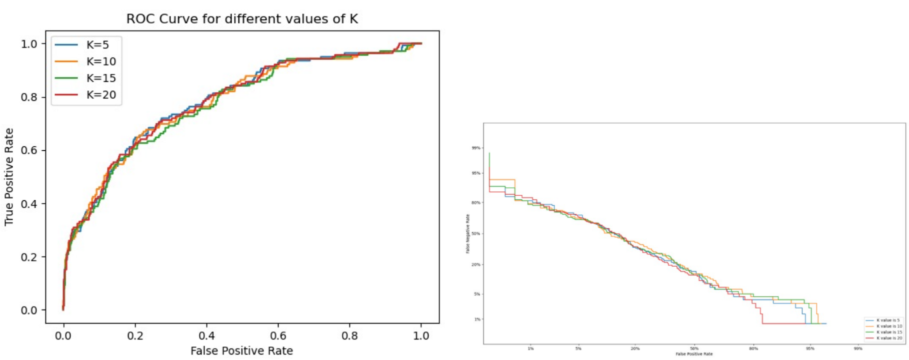
Fig1 shows the ROC and DET curves obtained for different values of k = [5, 10, 15, 20]. Math range errors come up for higher values of k.

1.2 Synthetic Dataset

The scatter plot of data given to our team is shown in Figure 2(a). Using the K-means algorithm, we separated the data into K clusters. Figure 2(b) shows various clusters for k = 5. To classify a test point, we model each class using k mixtures. The information from K-means clusters is used to compute initial priors, means and covariances in the EM algorithm for determining GMM parameters for each class. Figure 3 shows the variation of accuracy as k (no. of mixtures) varies for both diagonal and non-diagonal covariances. An accuracy of 100% is obtained on the development data for k = 20

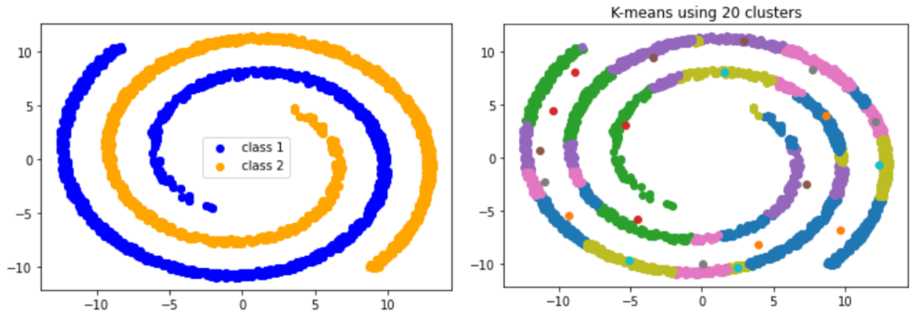
Figure 4 shows the contours of the mixtures (for both the classes, k = 15) on the data points. The Yellow and Red regions correspond to the higher probability regions of class 1 and class 2 respectively.

Figure 5 shows the contours of the mixtures for k = 20.



(a) ROC for different values of K (b) DET for different values of K

Figure 1: Plots for the Image Dataset



(a) Actual Dataset (b) Clustering using K-means

Figure 2: Plots for the synthetic dataset

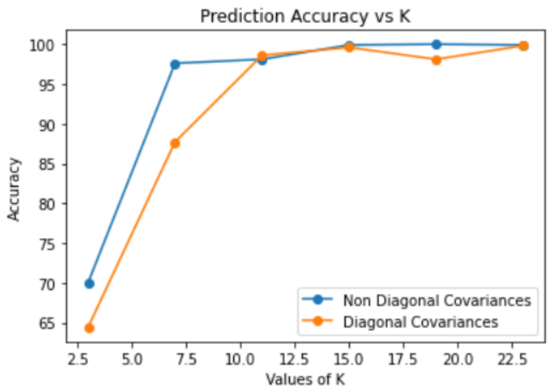


Figure 3: Prediction Accuracy for different values of K

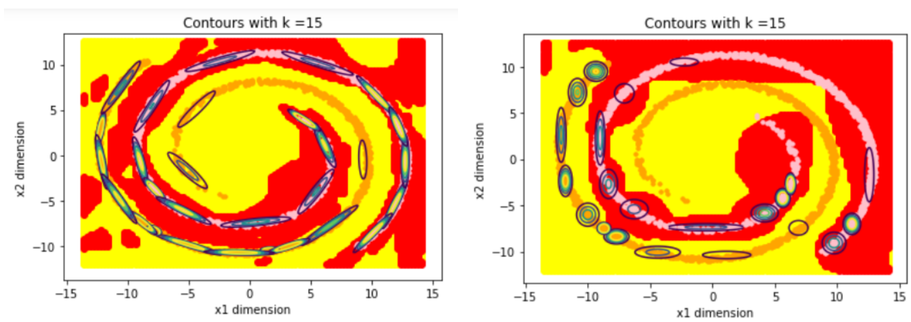
Figure 6 shows the ROC curves for different values of k when non-diagonal & diagonal covariances are considered. Figure 7 shows the DET curves for different values of k when non-diagonal & diagonal covariances are considered.

We restricted the number of K-means iterations to 20 (or convergence, whichever is achieved first), and the no. of EM iterations (for GMM) to 4. We observed that as the number of K-means iterations is increased further, there is no significant improvement in the accuracy, only the computation becomes harder.

2 DTW and Discrete HMM

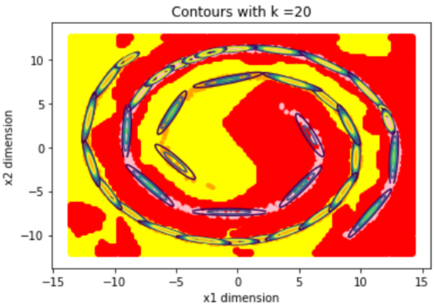
In DTW, to predict the class of a test vector, we find the DTW distance of the test vector from each example of a class and sort these values in ascending order. Choose the least K (empirical) values and find their average. For each test vector, we get an average score from each of the classes. Classify the test point into the class having the lowest average distance.

The reciprocals of these per-class averages are considered as thresholds to plot the ROC and

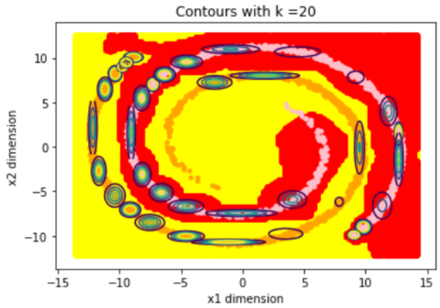


(a) Non-diagonal Covariances (b) Diagonal Covariances

Figure 4: Contour Plots of Mixtures for k = 15

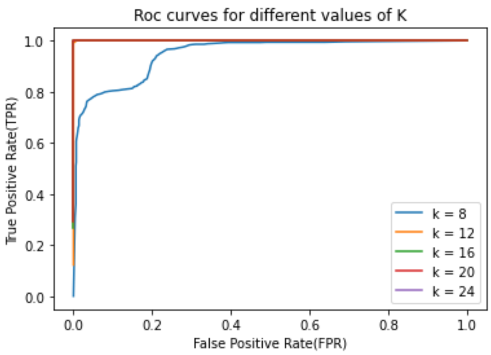


(a) Non-diagonal Covariances

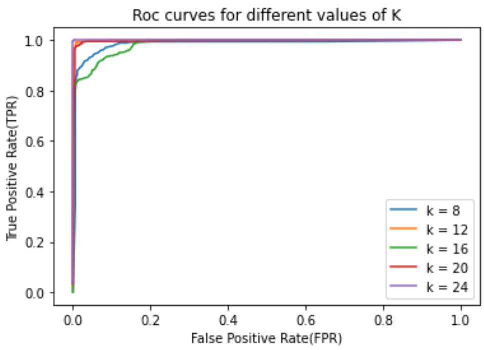


(b) Diagonal Covariances

Figure 5: Contour Plots of Mixtures for k = 20

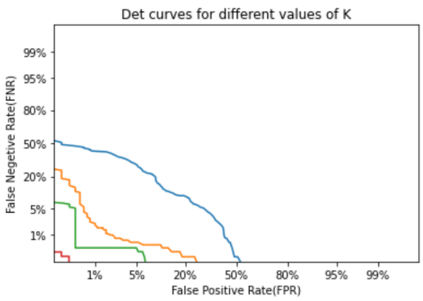


(a) ROC for non-diagonal covariances

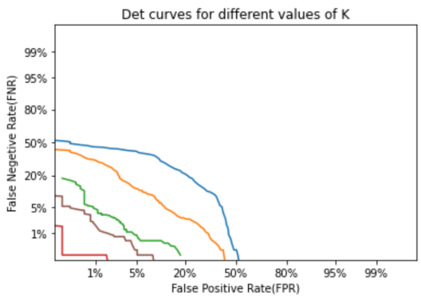


(b) ROC Curve for diagonal covariances

Figure 6: ROC Curves

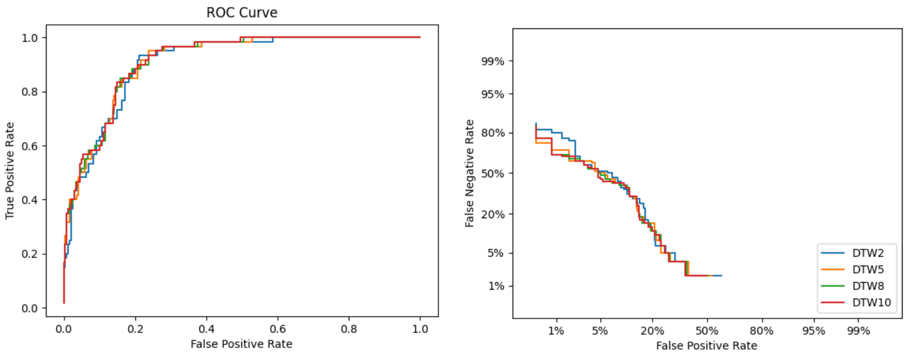


(a) DET curve (non-diagonal covariances)



(b) DET curve (diagonal covariances)

Figure 7: DET Curves



(a) ROC curve (b) DET curve

Figure 8: Plots for Spoken Digit Datset using DTW

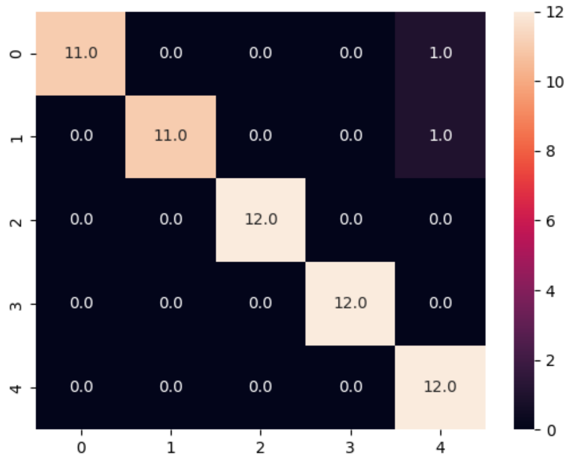


Figure 9: Confusion Matrix for Spoken Digit Dataset using DTW (k = 5)

DET curves. Please refer to the code for more clarity.

2.1 Isolated Spoken Digit dataset

In the spoken digit dataset, we obtained an accuracy of 98.3% using the least 5 values to compute the class average.

The ROC and DET curves are shown in Figure 8. The confusion matrix is shown in Figure 9 (for top 5 values).

2.2 Online Handwritten-Character dataset

In the handwritten character dataset, we obtained an accuracy of 100% on the development data, using the top 5 values to compute the class average.

Figure 10 shows the ROC and DET curves for various k values, Figure 11 shows the confusion matrix.

3 Discrete HMM

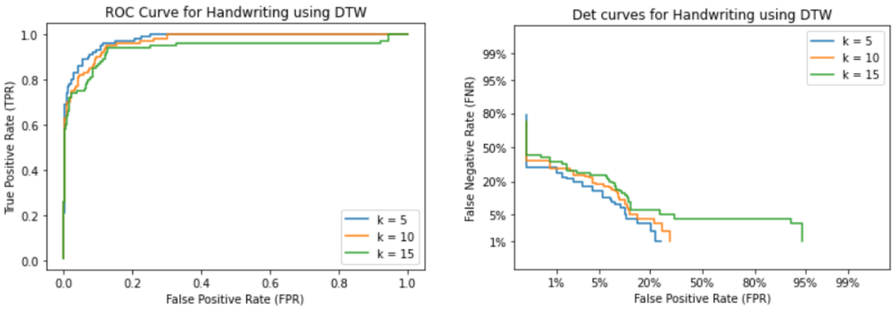
3.1 Isolated Spoken-Digit dataset:

Given dataset consists of spoken utterances.we need to claasify the development data.

STEPS:-

we need to do the vector quantization for the vectors present in the observation sequence of each digit.First creating mixture using all vectors present in all classes.

Now,Assign the observation for each mixture created using kMM.



(a) ROC curve

(b) DET curve

Figure 10: Plots for Handwritten-Character Datset using DTW

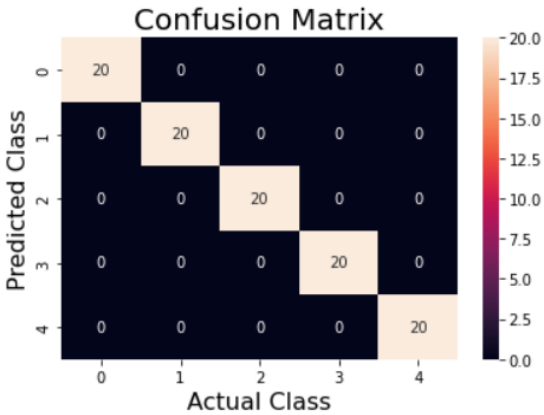
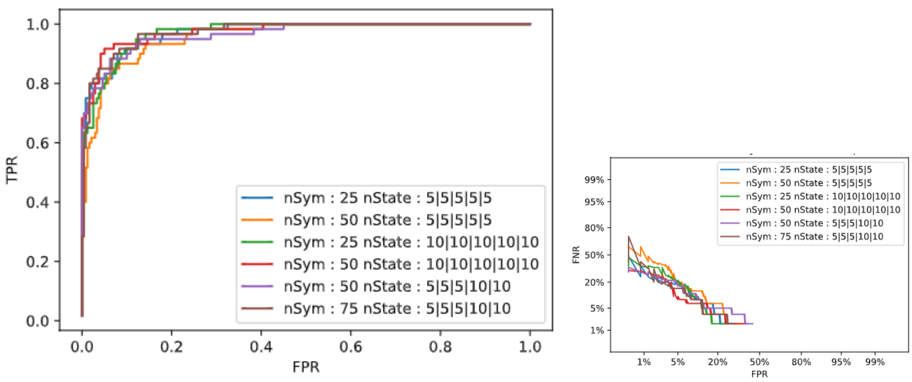


Figure 11: Confusion Matrix for Handwritten-Character Dataset using DTW



(a) ROC curve

(b) DET curve

Figure 12: Plots for Spoken-Digit Datset using HMM

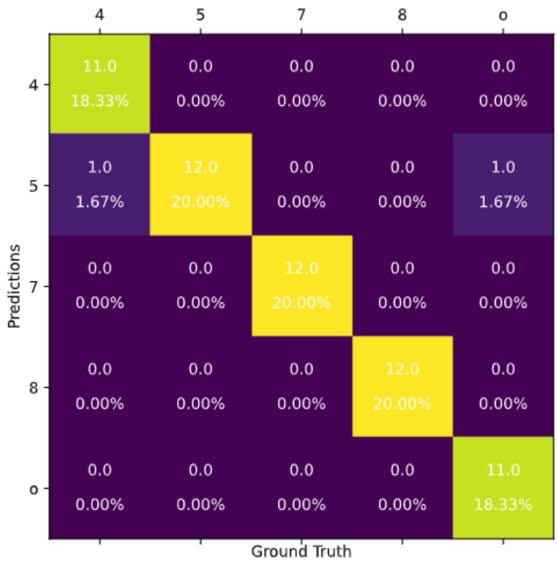


Figure 13: Spoken-Digit Dataset using HMM

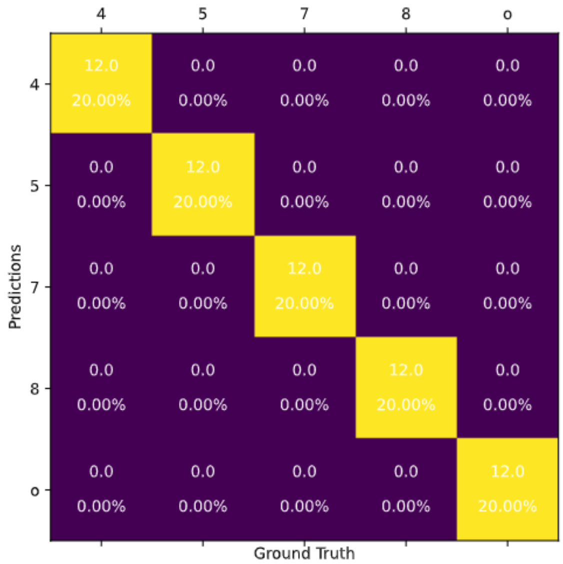


Figure 14: Confusion Matrix with high accuracy for spoken-Digit Dataset

Now using HMM-Code given the training sequences,we can get the transition probability from one state to another state and as well probability of emiting the observation over the edge from one state to another state.

Read the output given by HMM-code for each class and create a HMM model for each class.

Now compute the Likelihood w.r.to each class using backward method and pick the maximum likelihood which is the predicted class.

Now as number of states and symbols increases accuracy increases.

Results:- we can acheive almost 100% accuracy in case of symobols = 75 and states = [5,5,5,10,10] in each class can be seen in figure 14.

3.2 Online Handwritten-Character dataset

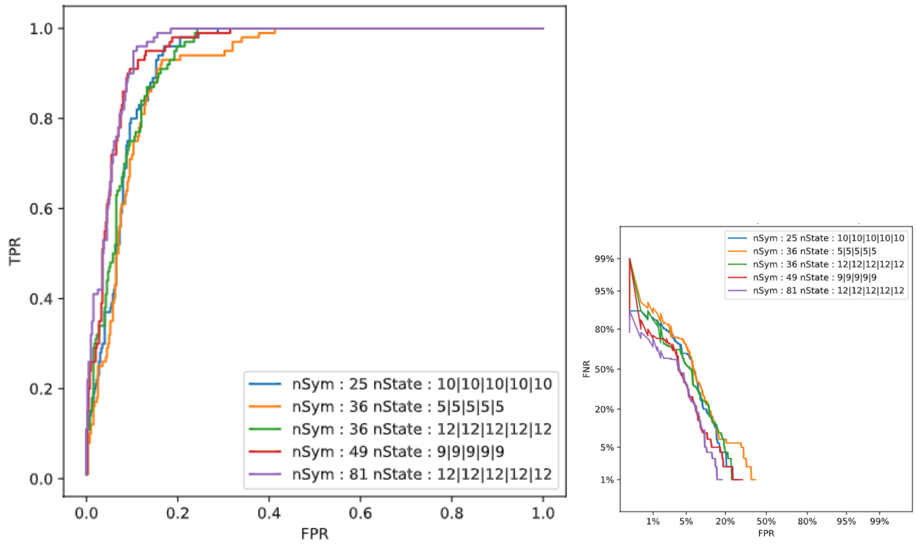
STEPS:- Given Handwritten Telugu characters.we need classify them.

In case of hand written telugu characters,we need to noramlize the data because we can write the same letter with scaling as well as relocation.

Repeat the same quantization as done in case of Isolated Spoken-Digit dataset.

Now,Assign the observation for each mixture created using kMM.

Now using HMM-Code given the training sequences,we can get the transition probability from one state to another state and as well probability of emiting the observation over the edge from



(a) ROC curve (b) DET curve

Figure 15: Plots for Handwritten-Character Datset using HMM

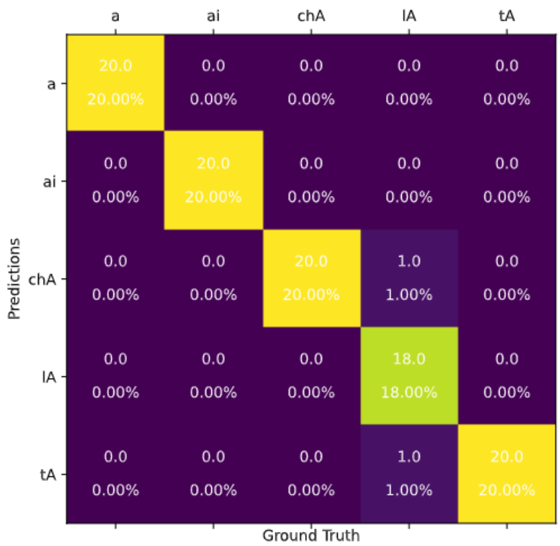


Figure 16: Handwritten-Character Dataset using HMM with symbols = 36 and states = 5 for each class

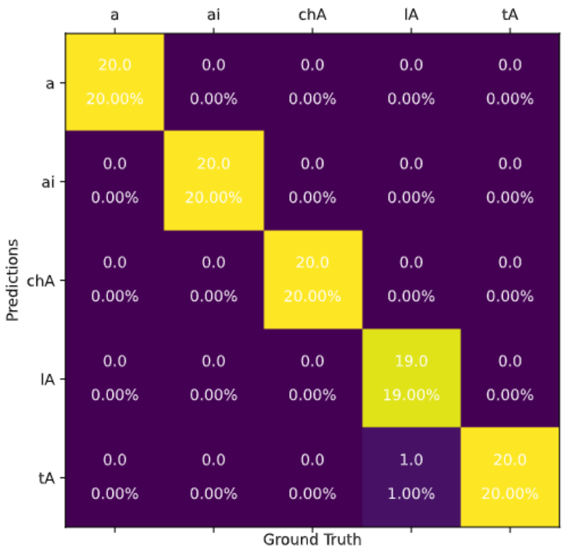


Figure 17: Confusion Matrix with high accuracy for Hand written

one state to another state.

Read the output given by HMM-code for each class and create a HMM model for each class.

Now compute the Likelihood w.r.to each class using backward method and pick the maximum likelihood which is the predicted class.

Now as number of states and symbols increases accuracy increases.

Results:- we can acheive almost 99% accuracy in case of symobols = 80 and states equal to 10 for each class can be seen in figure 17.