**ENEE633/ CMSC828C/ ENTS669E – PROJECT I REPORT**

**STATISTICAL PATTERN RECOGNITION**

**VARSHA ERANKI – UID 116204569**

1. **Introduction:** We are given a Fashion-MNIST dataset of 28 x 28 pixel grayscale images in a training set of these 60,000 thousand images belonging to 10 classes, and the testing set consists of 10,000 images.

* Using Maximum Likelihood followed by Bayesian rule for classification and Nearest neighbor rule classifier, the data is classified into its respective classes.
* Considering the high dimensional data, PCA and LDA methods are applied for dimensionality reduction and accuracy is checked.
* The entire project is programmed in Python.

1. **ML Estimation with Gaussian Assumption followed by Bayesian Classification: (METHOD 1)**

Data Preparation:

* The training dataset of 60,000 grayscale images with 28 x 28 pixels results in 784 pixels array each after flattening. These construct our training feature dataset, *X\_train,* with a 2-dimensional array of shape 60,000 x 784.
* The dataset is loaded into the MNIST dataset reader and separated into training and testing dataset features and label.
* To this array of flattened grayscale images, normalization was done to make it easier to compute and train the values between values 0 and 1.
* The necessary libraries for PCA, LDA and computation libraries for handling these datasets such as *numpy* is implemented.

Data Sorting:

* Data sorting was done according to the train and test labels (*y\_train and y\_test*) to sort the classes and their respective feature vectors for easy competence.

Gaussian Distribution and Probability density function:

* To Calculate the MLE, one must determine the probability density function for each class.
* This can be determined by calculation the Mean and Covariance matrix for each class which gives certain statistical data understanding behind every class data to recognize the pattern.
* The mean and covariance matrix are calculated by looping over the entire dataset for each class individually by calculating the mean of pixels belonging to that particular class.
* The Covariance is the calculated using an inbuilt function from one of the libraries mentioned.
* We finally arrive at a Mean matrix of shape 10 x 784, which depicts the mean pixel values for each of the 10 classes over the 784 normalized pixels.
* The Covariance matrix is also a 10 x 784 x 784 matrix, analogous to the mean matrix.

Determining MLE:

* The probability density function is calculated using Mean matrix and Covariance matrix followed by log application to calculate the highest value of probability from all the classes for each image to pinpoint to the true class and outputs it for prediction.
* To prevent for exceptions of Singular matrix or zero division, a small value of threshold was applied to the matrices and their features.

Applying PCA and LDA:

* The higher dimensionality dataset is difficult and time complex in nature, to make it simpler, we implement dimensionality reduction techniques.
* It is done by computing the highly variant features over the dataset for each class and considering only these features for classification.
* By this technique, the probability of belonging to a particular class is easily computed as the dataset has way less features to train and accuracy improves.

1. **K-Nearest Neighbor rule Classifier: (METHOD 2)**

Data Preparation, Data Sorting and Applying PCA and LDA are the same as used in ML Estimation followed by Bayesian Classification, what differs is the algorithm model of k-NN which is stated below.

K-NN Algorithm model:

* K-NN algorithm implements the distance parameter to calculate the probability of belonging to a class.
* In my problem, I implemented Minkowski equations to calculate the distance of feature points and their underlying class information.
* Implemented using K = 3 and computed for a better accuracy.

1. **Accuracies obtained:**

The accuracy of Bayesian classification task falls very low to 61% and the PCA and LDA accuracies go up to 77% and 81% respectively.

For PCA computation, The dimensionality reduction is observed from 61% to 77%.

The number of features to be detected for PCA is brought down to 50, whereas the LDA accuracy went sub-sequentially quite to 81% with 9 feature points only for each image.

LDA and PCA accuracies can be seen as follows for KNN where k = 3

In this case, the data visualization was seen for various values of ‘k’ at the same time.

The PCA and LDA accuracies turn out to be 85% and 81.4%.

Also, for many instances in the program, a small quantity was added to remove the exception of Singular matrix, and hence the above method was implemented.