HAI 1.0B Assignment 1

Abhiroop Talasila

9th June 2021

1. First Question

1.1. What are Eigenfaces?

Eigenfaces are nothing but the eigenvectors obtained from PCA when used for human face recognition. Since images are high dimensional data, we need components which capture most variability in the dataset. Thus, all faces in the dataset can be represented as linear combinations of these eigenfaces. The information loss is inversely related to the number of top **k** eigenfaces we select.

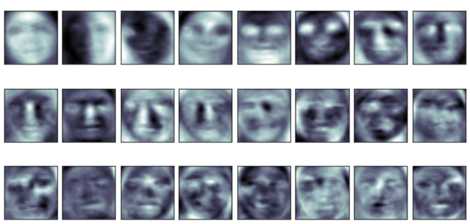

Figure 1. Eigenfaces from IMFDB Dataset

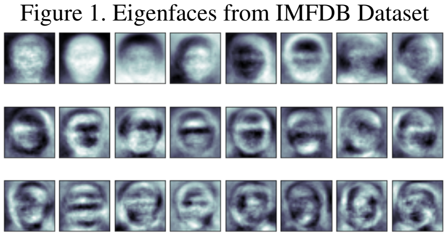

Figure 2. Eigenfaces from IIIT-CFW Dataset

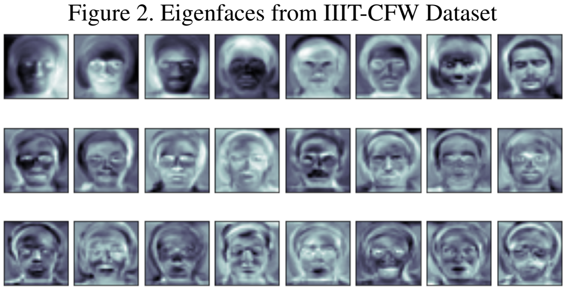

Figure 3. Eigenfaces from Yale Dataset

1.2. How many eigen vectors/faces are required to "satisfactorily" reconstruct a person in the three datasets?

Using eigenvectors which capture around 90% of the total variance worked best to "satisfactorily" reconstruct faces. For IMFDB, first 40 eigenvectors are taken, 40 for IIIT-CFW, and 30 for Yale Dataset. The corresponding graphs are given below the `get_pca()` cell in the notebook. Refer Figures 14-16 at the end.

1.3. Reconstruct the image back for each case

First, the data is centered and covariance matrix is calculated. The mean of the dataset is added to the top **k** eigenvectors and these are our final eigenvectors. Now, the dot product of these eigenvectors and Principal Component projections gives us the reconstructed images. The process can be described as:

Image = PC Projections · Eigenvectors^T **+ Mean**

Mean Squared Error is chosen to measure reconstruction error. Table 1 shows error with each feature for each dataset.

1.4. Which person/identity is difficult to represent compactly with fewer eigen vectors? Why is that? Explain with your empirical observations and intuitive answers

For IMFDB, performance of eigenfaces from PCA is worst for Kajol. This is due to a combination of complex facial expressions and variations in illumination. For IIIT-CFW, Amitabh Bachchan is much harder to represent than others. This might probably be due to complex facial features which require more eigenvectors to represent. For Yale, Class 6 is slightly harder to represent. The confusion matrices of each dataset are shown in Figures 4-6.

2. Second Question

2.1. Use any classifier and find the classification accuracy. Write code from scratch making classifier as a class and functions of training, validation and confusion matrix etc.

The custom `SVM()` class defined in the notebook has 'rbf', 'poly', 'sigmoid' and 'linear' kernel instances. The input data is normalized using `StandardScaler()`. The class reports mean accuracy and weighted F1 score using `StratifiedKFold()` validation. If required, `classification_report()` and `confusion_matrix()` are printed.

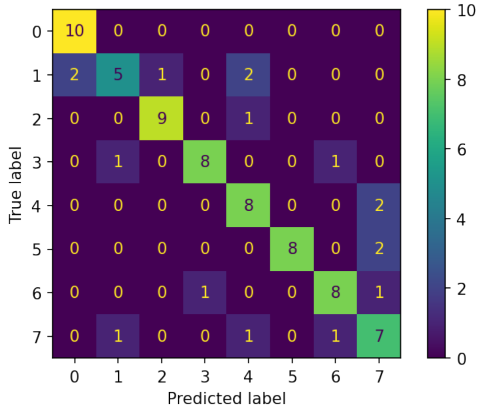

Figure 4. Confusion Matrix for IMFDB Dataset

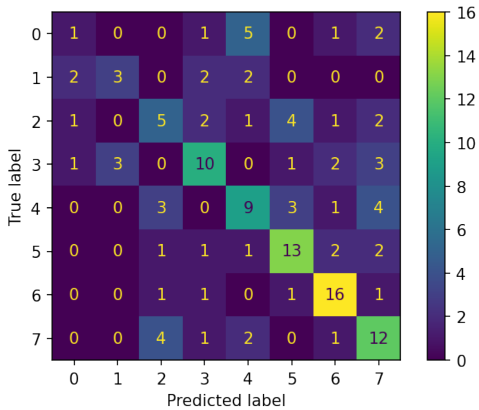

Figure 5. Confusion Matrix for IIIT-CFW Dataset

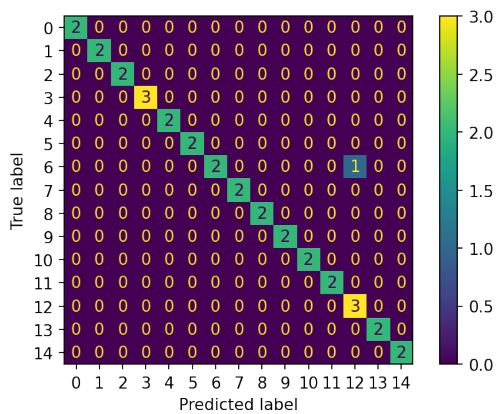

Figure 6. Confusion Matrix for Yale Dataset

2.2. Which method works well? Do a comparative study.

The results are shown in Table 2 and 3. We can see that vanilla LDA works best for our task. PCA reduces dimensions by focusing on the features with the most variation. LDA is like PCA, but it focuses on maximizing the separability among known categories. This leads to better performance for classification tasks.

3. Third Question

3.1. Similar to 1(b) use t-SNE based visualization of faces? Does it make sense? Do you see similar people coming together? Or something else? Can you do visualization dataset wise and combined?

Figures 7-10 visualize this. Best discrimination is obtained when vanilla LDA is used with tSNE. There is clear clustering of all the different classes of people in all datasets. The combined dataset is obtained by concatenating all images and we can again see good separation in classes with some reasonable amount of outliers.

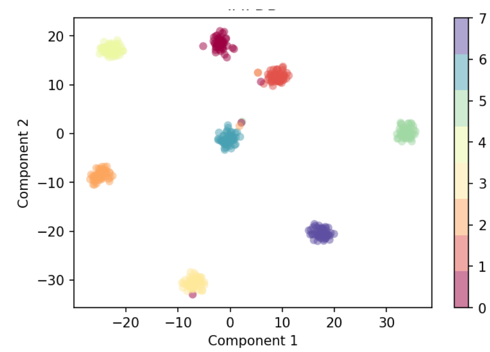

Figure 7. LDA + tSNE for IMFDB Dataset

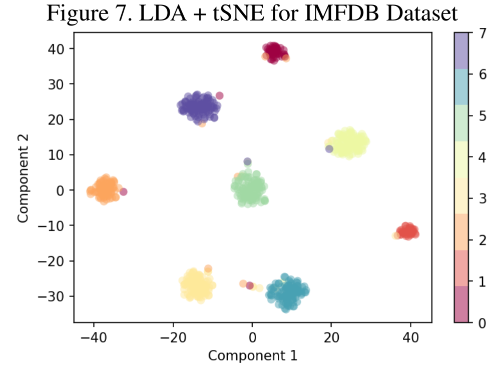

Figure 8. LDA + tSNE for IIIT-CFW Dataset

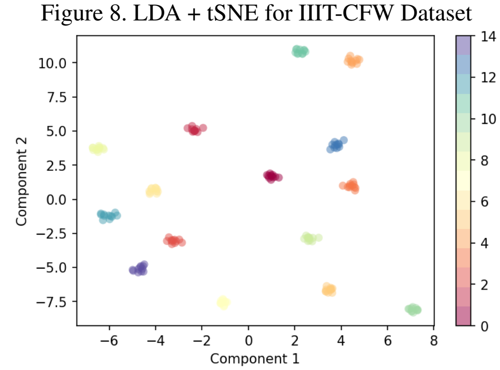

Figure 9. LDA + tSNE for Yale Dataset

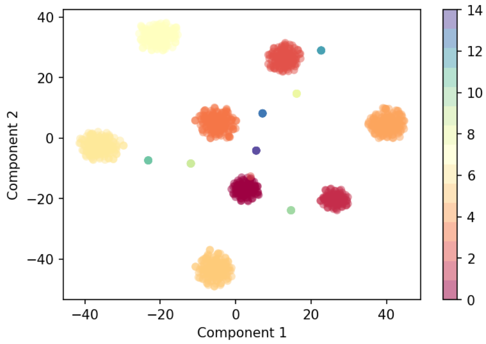

Figure 10. tSNE + LDA for Combined Dataset

4. Fourth Question

4.1. In practice "face" is used for verification. How do we formulate the problem using kNN?

We can extract any one of the given features and calculate **k** Nearest Neighbors for each testing face. This builds on the assumption that data points of similar classes are closer to each other. I think LDA will be better for this task since it focuses on maximizing the separability among known classes.

4.2. How do we analyze the performance ? suggest the metrics (like accuracy) that is appropriate for this task.

Generally, Manhattan distance or Euclidean distance are used with kNN to predict classes. But there was a paper by Alkasassbeh (CJPAS 2015) which showed how the Hassanat distance metric enhances the performance of the Nearest Neighbour classifier. They proved that the Hassanat distance metric is invariant to data scale, noise and outliers.

4.3. Show empirical results with all the representations

I've tried both Euclidean and Hassanat. The latter doesn't always perform better, but the authors do mention that Hassanat works better for larger datasets. Hassanat works better without normalized data, whereas Euclidean prefers it (implemented). We can compare performance using accuracy and F1 score. I've used two approaches: using raw dataset as input to kNN or using LDA features as input to kNN. The LDA + kNN approach achieves perfect scores on all datasets. Figures 11-13 show the variation of accuracy of plain kNN w.r.t **k** on the datasets.

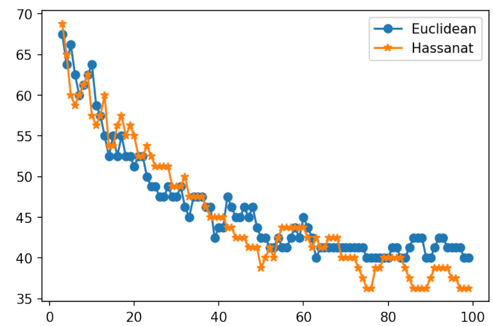

Figure 11. kNN accuracy on IMFDB Dataset

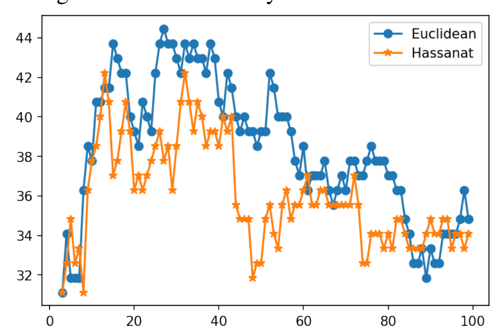

Figure 12. kNN accuracy on IIIT-CFW Dataset

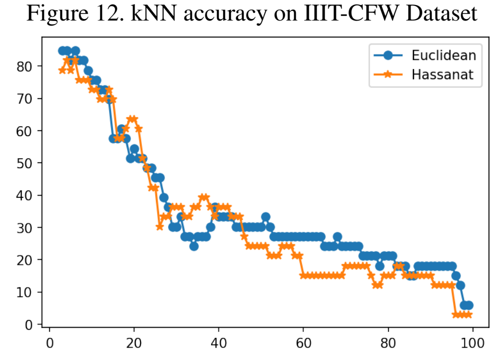

Figure 13. kNN accuracy on Yale Dataset

Dataset/Features	PCA	KPCA	PCA+LDA	LDA	KPCA+LDA
IMFDB	0.11	0.02	0.03	0.09	0.03
IIIT-CFW	0.14	0.07	0.08	0.13	0.07
Yale	0.15	0.05	0.49	0.50	0.27

Table 1. Reconstruction error (MSE)

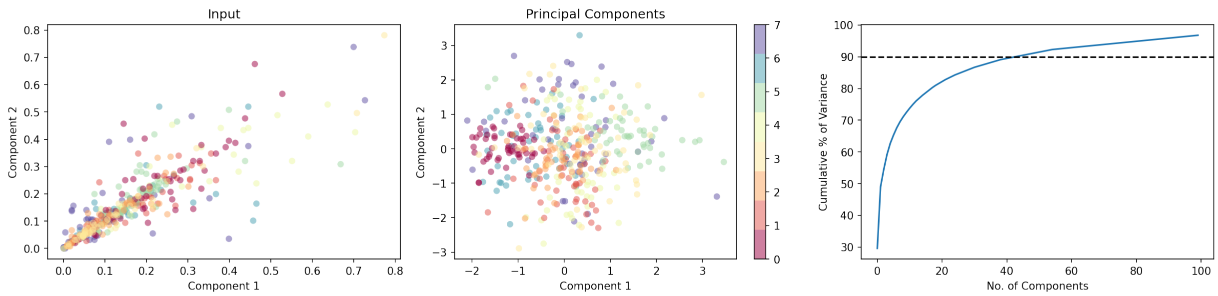

Figure 14. IMFDB Dataset

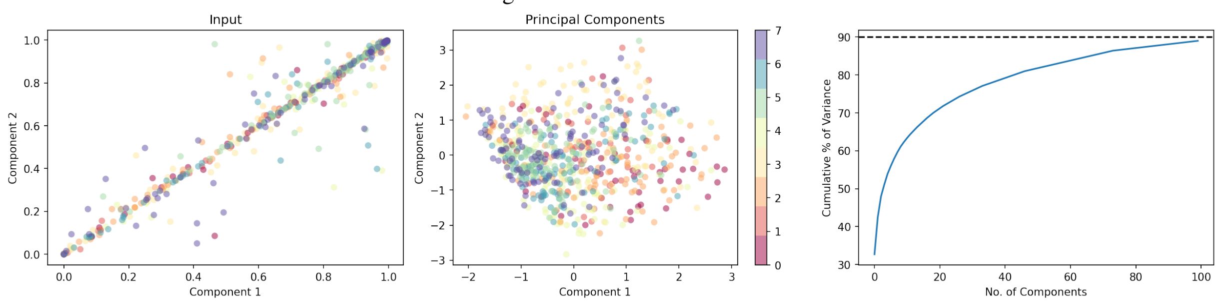

Figure 15. IIIT-CFW Dataset

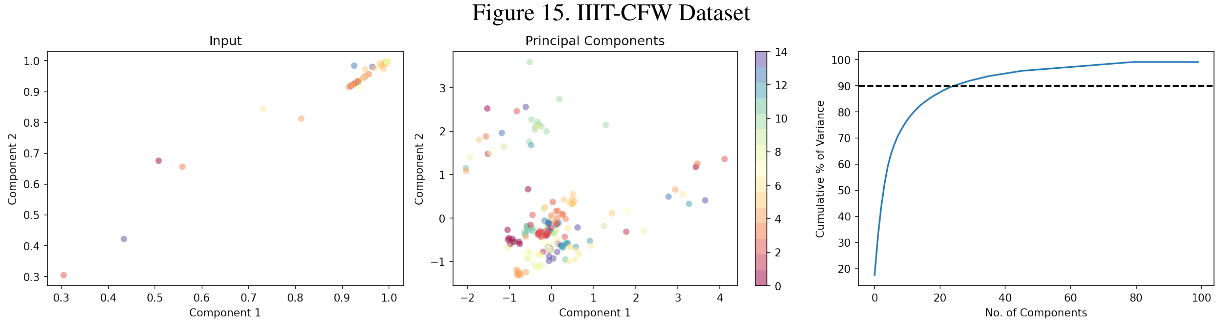

Figure 16. Yale Dataset

Dataset/Features	Raw	PCA (k)	KPCA (k)	PCA+LDA (k)	LDA (k)	KPCA+LDA (k)	VGG	Resnet
IMFDB	89%	79% (40)	79% (60)	86% (6)	100% (5)	85% (5)	89%	99%
IIIT-CFW	58%	51% (40)	54% (60)	60% (8)	98% (5)	61% (5)	72%	98%
Yale	100%	97% (30)	91% (60)	100% (8)	100% (8)	100% (7)	70%	100%

Table 2. Best Accuracy on given datasets

Dataset/Features	Raw	PCA (k)	KPCA (k)	PCA+LDA (k)	LDA (k)	KPCA+LDA (k)	VGG	Resnet
IMFDB	0.89	0.79 (40)	0.78 (60)	0.86 (6)	1.00 (5)	0.85 (5)	0.89	0.99
IIIT-CFW	0.58	0.50 (40)	0.54 (60)	0.58 (8)	0.98 (5)	0.60 (5)	0.72	0.98
Yale	1.00	0.97 (30)	0.90 (40)	1.00 (8)	1.00 (8)	1.00 (7)	0.63	1.00

Table 3. Best F1 Scores on given datasets