IIIT-H Assignment1

Anonymous CVPR submission

Paper ID

1. Eigen Faces

1.1. What are Eigen Faces?

To understand what eigen faces are, one must first understand PCA (Principal Component Analysis). PCA is a dimensionality reduction method which creates new independent features to represent the data. The features can be combined to get the old features. These new features are the eigen vectors of the covariance matrix of the data. Suppose we have 100 images of size 32*32, after flattening them we get a matrix X of size (100,1024). Along with computing the eigen vectors of the covariance matrix(XX^T), eigen values which tell us the importance of the eigen vector to recreate the image are also computed. Importance of the eigen vectors is in terms of how much variance is explained by each vector. It can be easily identified that more than 90% variance can be explained by a handful of eigen vectors. These handful of eigen vectors are also termed as the eigen faces. Each image can represented in the lower feature space by loosing only a small fraction of information.

1.2. How many eigen vectors/faces are required to satisfactorily reconstruct a person?

A graph between the cumulative variance and the number of features can be plotted and the optimal number of eigen vectors/faces can be determined. Here I have decided to take the features that explain 97% of the variance in the data.

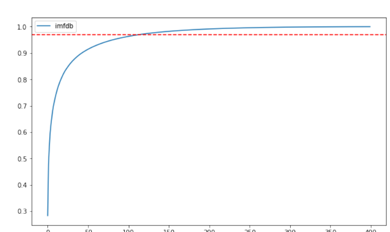

Figure 1. Variance vs number of features for IMFDB dataset

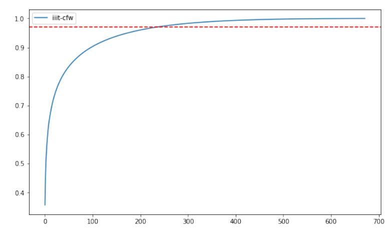

Figure 2. Variance vs number of features for IIIT-CFW dataset

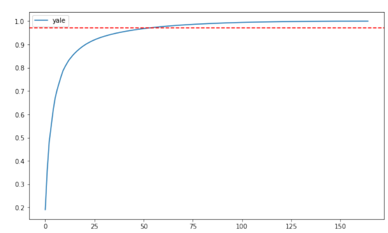

Figure 3. Variance vs number of features for Yale face dataset

1.3. Reconstruction of image

When an image is transformed to a lower feature space each image is represent by r datapoints (r being the number of vectors that explain 97% variability, in case of IMFDB dataset it is 115). It can be transformed back to its original size by taking the dot product of the transpose of the image and the eigen vectors and then adding the mean of all images to it.

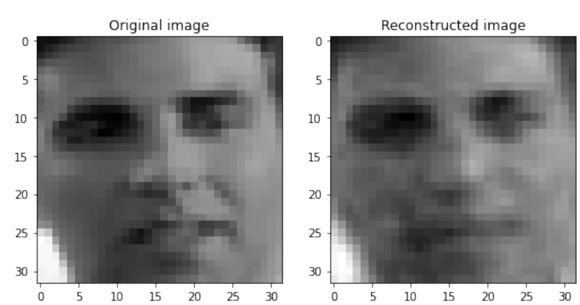

Figure 4. Original Image and Reconstructed image

1.4. Which person/identity is difficult to represent compactly with fewer eigen vectors?

To approach this problem, I chose to use reconstruction loss. From the graphs above it is clear that to explain 97% variability the IIIT-CFW dataset would require the most number of features or eigen vectors.The mean reconstruction loss for Manmohan Singh is 2.34*10-3 which is the highest for any class in the dataset.

2. Comparitive study of Classifiers

In this paper four different classifiers along with 6 different dimensionality reduction methods and show the best result achieved on the three datasets. Unlike PCA which is used for linear transformations, kernel PCA is applied for non-linear transformations. The points are mapped into a feature space and then linear dimensionality reduction is performed on it. This can be achieved easily by getting the eigen vectors of the kernel matrix (X^TX), and instead of dot product K function is used between two points (rbf/poly). Linear Discriminant Analysis (LDA) maximizes the distance between means of the classes while minimizing the variation within each class. Kernel LDA is the non linear transformation for LDA. VGG-19[1] and Resnet-50[2] are pre-trained deep CNN models whose last layer features are taken. It was found that Logistic regression performed the best for various combinations surpassed only by MLP when using kernel dimensionality reduction methods.

2.1. Decision Tree

A decision tree as the name suggests is a tree like structure with branches and nodes. Each node helps split the data and finally the leaf nodes are classes, this can tell us which datapoint belongs to which class. I have used the Gini index or impurity measure to identify the features on which to split the data.

2.2. Logistic Regression

Unlike its name logistic regression is used for classification tasks. It squishes the output y between 0 and 1 by applying a sigmoid function and hence we start dealing in probabilities, P(y=1—X). Maximum likelihood Estimation is used to find the parameters that maximise the log-likelihood.

2.3. Support Vector Machine

SVM tries to fit a hyperplane which divides the classes in the feature space. For a scenario with two classes, the hyperplane which maximizes the distances between nearest data point (either class) is chosen.

	PCA	Kernel PCA	LDA
IMFDB	11.25*	77.5*	99*
IIIT-CFW	53*	49.6*	97
Yale face	93.9	87.8*	100

Table 1. Accuracy comparision for PCA, Kernel PCA and LDA (* indicates MLP performed better than Logistic Regression)

	Kernel LDA	VGG-19	Resnet-50
IMFDB	27.5*	9.75	96
IIIT-CFW	25*	71.2	97.17
Yale face	48.4*	69.6*	100*

Table 2. Accuracy comparision for PCA, Kernel PCA and LDA (* indicates MLP performed better than Logistic Regression)

2.4. Multilayer Perceptron

MLP can be explained as an ANN with fully connected dense layers. In this case I have taken 2 hidden layers with 150 and 50 neurons respectively. I used adam optimizer and categorical cross-entropy loss to train the model.

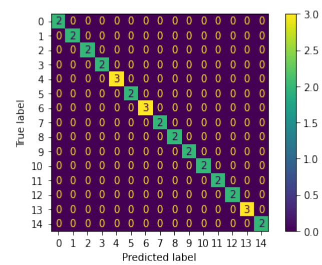

Figure 5. Confusion Matrix after LDA dimensionaltiy reduction on the Yale face dataset

3. t-SNE visualization

t-SNE is a dimensionality reduction technique that is highly useful in visualizing high dimensional data in lower dimensions. Note that the features extracted in the lower space after using t-SNE should not be used further for classification tasks, because it does not learn an explicit function to map the datapoint in the lower dimensionality space. It works on the concept of similarity, a point which is closer to certain points in the high dimensional space should be close to those points in the low dimensional space as well. In the Fig.6 and Fig.7, it is shown how t-SNE performs after applying PCA and LDA dimensionality reduction on the Yale dataset. I also observed that it did not form proper clusters when applying PCA to any other dataset but worked great when applying LDA which reinforces our faith in the results mentioned in the Table1 and Table2.

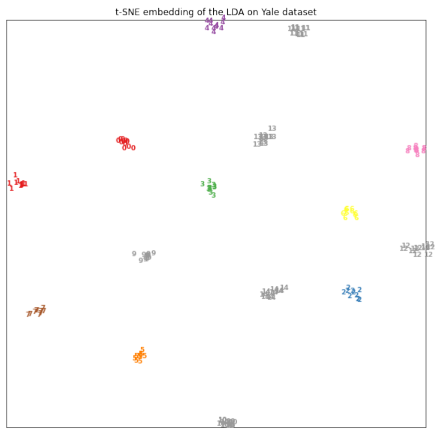

Figure 6. t-SNE visualization after LDA dimensionaltiy reduction

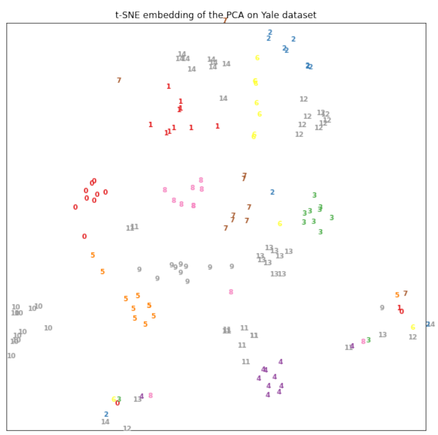

Figure 7. t-SNE visualization after PCA dimensionaltiy reduction

4. K-Nearest Neighbour

K-Nearest Neighbouror KNN algorithm is a simple classification algorithm which works on the concept that a new point in the dimension space will be assigned the class to whom it has most closest neighbours out of k. Suppose in a normal dog vs cat classifier having k as 21, a new image has 15 closest points of dog class and 6 of the cat class, the new image will be classified as a dog. Here k can be decided my drawing a graph between the K values and the accuracy of the model, after a certain K value the accuracy seems to be between particular values, this helps us decide the optimal value of k. Metrics like recall, precision and f1-score can be used to evaluate the performance of the model. I have shown the results of KNN only after performing LDA as it has shown to be superior for the tasks till now. The full classification report is available in the notebook and a mean accuracy of 98.25,97.6,100 in the 3 datasets was achieved respectively.

References

[1] Z. A. Simonyan K, “Very deep convolutional networks for large-scale image recognition.,” 2014, arXiv preprint arXiv:1409.1556.

[2] H. K. Z. X. R. S. S. J., “Deep residual learning for image recognition.,” pp. 770–778, 2016, Proceedings of the IEEE conference on computer vision and pattern recognition 2016.