Implementing classifiers for Face Recognition

Different combinations of classifiers and their parameters were tried out on the three data sets (data.mat; pose.mat and illumination.mat) and their accuracy results are reported. In particular, the following classifiers are implemented in MATLAB.

1. MLE with Gaussian Assumption followed by Bayes Rule
2. Nearest Neighbor Rule and its generalization to the kNN classifier
3. PCA followed by Bayes
4. PCA followed by kNN
5. LDA followed by Bayes
6. LDA followed by kNN

Results

A. Experiments with **data.mat**

1. **MLE with Gaussian Assumption followed by Bayes Rule**

The Naïve Bayes classifier with the Gaussian assumption was used to classify the faces in data.mat which had 200 subjects with 3 images each. 2 images were used for training and 1 for testing. The following combinations of the training and testing images was used.

a. Bayes with 2 training samples (3i-1, 3i-2) and 1 testing sample (3i) – Accuracy of 50 %
b. Bayes with 2 training samples(3i, 3i-1) and 1 testing sample (3i-2) – Accuracy of 83.5%
c. Bayes with 2 training, 1 testing. **Random selection** – Accuracy of 81%

It is seen that the classification accuracy is dependent on the choice of training samples. Random selection of 2 training and 1 testing sample gives an accuracy of 81 %.

Because the number of samples is less than the feature size, the covariance matrices are singular. This requires the use of some sort of regularization or shrinkage techniques to condition the matrix and make it stable[1]. The values for the discriminant functions for each of the training set vectors are evaluated and classified according to the maximum discriminant value.

2. **NN Rule and extension to kNN**

Training was done with 2 samples indexed by (3i-1, 3i-2) and testing with 1 sample indexed by (3i). The distance calculation was done using the 2 norm and 1 norm method and their corresponding classification accuracies are seen as follows. The concept of NN Rule is extended to k Nearest neighbors.

Number of Neighbors (k)	Distance Calculation Method	Accuracy
1	**2 norm**	**59.5 %**
2	2 norm	46.5 %

3	2 norm	45.5 %
4	2 norm	44.5 %
5	2 norm	46 %

Number of Neighbors	Distance Calculation Method	Accuracy
1	**1 norm**	**50.5 %**
2	1 norm	45 %
3	1 norm	41 %
4	1 norm	41 %
5	1 norm	38.5 %

3. PCA followed by Bayes

The dimensionality reduction offered by PCA is a very useful step for visualizing and processing high-dimensional datasets, while still retaining as much of the variance in the dataset as possible. Because PCA is translation variant, the faces must be frontal and well aligned on facial features such as the eyes, nose and mouth.

PCA is useful for dimensionality reduction if the size of the training set is too small for the number of dimensions of the data. But if you are using all of the principal components, PCA won't improve the results of your linear classifier - if your classes weren't linearly separable in the original data space, then rotating your coordinates via PCA won't change that.

The problem with image representation is its high dimensionality. Two-dimensional $p \times q$ grayscale images span a $m = pq$-dimensional vector space, so an image with 24 x 21 pixels lies in a 504-dimensional image space. The PCA method finds the directions with the greatest variance in the data, called principal components. By varying the number of principal components used, the dimensionality can be brought down as required.

The following table gives the classification accuracy for PCA followed by the Bayes Rule for different number of principal components. It is seen that the accuracy does not vary by a large amount by increasing the number of principal components which means that most of the energy of the signal is contained within the first 20 principal components.

Number of Principal Components	Accuracy
20	30 %
50	30 %
100	30 %
150	32.5 %
200	**32.55 %**
300	32.5 %

4. **PCA followed by kNN**

The following table gives the classification results after PCA is applied to the image vector and by using the kNN classifier. Results with k=1, 2 and 3 neighbors are reported with varying number of principal components. Here we see an increase in the classification accuracy as the number of principal components used in the data representation increase.

Number of Neighbors	Distance Calculation Method	Number of Principal Components	Accuracy
1	2 norm	20	13.5 %
	2 norm	100	29.5 %
	2 norm	200	32.5 %
	2 norm	300	33.5 %
	2 norm	**400**	**34 %**
2	2 norm	20	8 %
	2 norm	100	19 %
	2 norm	200	23 %
	2 norm	**300**	**25 %**
	2 norm	400	25 %
3	2 norm	20	9.5 %
	2 norm	100	21.5 %
	2 norm	200	24 %
	2 norm	300	24.5 %
	2 norm	**400**	**25 %**

The deficiencies of the eigenface method are as follows:

- Very sensitive to lighting, scale and translation; requires a highly controlled environment.
- Eigenface has difficulty capturing expression changes.
- The most significant eigenfaces are mainly about illumination encoding and don't provide useful information regarding the actual face.

To cope with illumination distraction in practice, the eigenface method usually discards the first three eigenfaces from the dataset. Since illumination is usually the cause behind the largest variations in face images, the first three eigenfaces will mainly capture the information of 3-dimensional lighting changes, which has little contribution to face recognition. By discarding those three eigenfaces, there will be a decent amount of boost in accuracy of face recognition, but other methods such as Fisherface and Linear space still have the advantage.

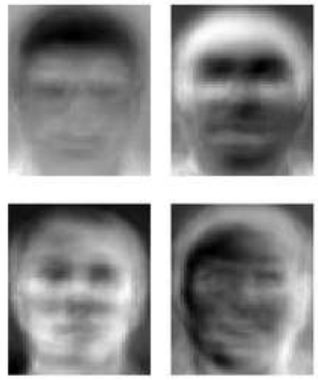

Fig 1. Example of Eigen faces corresponding to first 4 principal components

5. **LDA followed by Bayes**

Linear discriminant analysis is primarily used here to reduce the number of features to a more manageable number before classification. Each of the new dimensions is a linear combination of pixel values, which form a template. The linear combinations obtained using Fisher's linear discriminant are called *Fisher faces.* The number of dimensions are brought down from (24*21) to c-1 =199 where c is the number of classes. The accuracy of LDA combined with Bayes was found to be **83%**.

6. **LDA followed by kNN**

The following table gives the classification accuracy for LDA followed by kNN for feature length = c -1. It is seen that the NN rule performs very well in this case.

Number of Neighbors	Distance Calculation Method	Accuracy
1	**2 norm**	**100 %**
2	2 norm	51 %
3	2 norm	36.5 %
4	2 norm	28.5 %
5	2 norm	21 %

A set of similar experiments was performed with the other 2 datasets as well. The results are summarized below -

B. Experiments with **pose.mat**

(First) 9 images per subject (69% of data) are used for training and (last) 4 for testing (21% of data). In general, we can see that the classification accuracies for every method are more than the previous data set as the number of training/testing samples are more.

Classification Method	Parameters	Accuracy
MLE with Gaussian Assumption followed by Bayes Rule		82 %
kNN	k=1, 2norm	69.85 %
PCA + Bayes	#Principal components = 300;	46.5 %
PCA + kNN	#Principal components = 300; k=1, 2 norm	38.3 %
LDA + Bayes	#features = c -1 = 67	85.1 %
LDA + kNN	k=1, 2 norm, #features = 67	96.3 %

C. Experiments with **illumination.mat**

(First) 18 images per subject (85% of data) are used for training and (last) 3 for testing (15% of data).

Classification Method	Parameters	Accuracy
MLE with Gaussian Assumption followed by Bayes Rule		84.3 %
kNN	k=1, 2norm	99.3 %
PCA + Bayes	#Principal components = 300;	49.2 %
PCA + kNN	#Principal components = 300; k=1, 2 norm	42.1 %
LDA + Bayes	#features = c -1 = 67	87 %
LDA + kNN	k=1, 2 norm, #features = 67	98.1 %

Conclusion

In this project, we evaluated various machine learning algorithms for face recognition. The entire image was used as the feature vector for classification in the first 2 methods. We used PCA and LDA to extract features and then feed it to the machine learning algorithm in the subsequent methods. In general, it can be seen that as the amount of data increases, the classification accuracies increase. The accuracies are also dependent on the pose and illumination as can be seen by varying percentage accuracies for the pose and illumination datasets. As future work, some of the other commonly used feature extractors like – HOG (Histogram of Gradients) and SIFT (Scale Invariant Feature Transform) can also be used to boost accuracies.

MATLAB files

The following MATLAB files are used for implementing the classifiers –

1. BayesClassifier.m – Implements and tests the Naïve Bayes classifier
2. knnclassification.m – Describes the knnclassification function. Can test with kNNtestfile.m.
3. pcaP1.m – Implements the PCA algorithm. Can test with PCAandNNTest.m file
4. lda.m – Implements the LDA algorithm. Can test with LDAandNNTestfile.m

References

[1] www.doc.ic.ac.uk/~dfg/ProbabilisticInference/IDAPILecture17.pdf