DATA603 - Project 1

Shrey Patel | UID: 117583915

Shrey Nair | UID: 113160714

Introduction

This project serves as a research and evaluation for popular techniques and methods used in modern face recognition such as Bayesian classification, k-NN classification and different transformations such as Principal Component Analysis, Multiple Discriminant Analysis to enhance the performance of base classifiers at every task.

Two types of data sets were used to perform tasks such as binary classification and multiclass subject classification, respectively. After dividing the datasets into training and testing, we calculated the accuracy.

Neural vs Smiley faces

- Here, Data.mat was used as the data set. A split of 0.25:0.75 (test-to-train) was kept. Each run sources random samples for the training and testing data sets while maintaining the split ratio. Before the split, we reshape the data to the total dimensions of each image for each class.
- So data.mat is a M x N x S is reshaped to (M*N,1) for training data and same with testing data.
- Labels for training and testing data sets are generated as well during preprocessing.
- Training data is fed to the user selected classifier for learning.
- The classifiers used for the experiment were Bayesian classification, PCA + Bayes classifier and PCA + kNN Classifier.
- For performing PCA on training data, we compute the eigenvectors of the covariance matrix and approximate the original data by a linear combination of leading eigenvectors.

In MATLAB, we are using svd() Singular Value Decomposition to find the singular values and singular vectors.

- Using the generated 'U', we calculate the new transformed training and testing dataset by multiplying it to the original respective datasets; the new training and testing datasets are then fed to the Bayesian classifier for decision.
- Bayesian classifier was run 10 times on randomized training and testing data set.
- kNN classifier was run from 1 to 20 nearest neighbors for a random training and testing data set. Refer figure for accuracy of the classifiers mentioned above over iterations and kNN number.

1. Bayesian classification for Neutral vs Facial Expressions:

- Calculate the mean (μ) per class for the training dataset.
- Calculate the variance ($\sum$) and variance inverse ($\sum -1$) for every class.
- We centre the data for each class to compute the covariance matrices
- We define the discriminant matrices based on the training data.
- To make a decision among two classes, w1 and w2; we calculate $g(x)$. $g(x) = X\ TWX + w\ TX + w0$
  where,
  $W = -\ 1\ 2 \sum -1\ w = \sum -1\mu\ w0 = -\ 1\ 2\ \mu\ T\sum -1\mu - 1\ 2\ ln|\sum -1\ | + ln(P(x))$
- Calculate these for all values in the test dataset and find the maximum value of $g(x)$. As we find the max value, we assign the label of that class value, thus making a decision per test sample.
- Evaluate various elements in the matrices and run the testing data through the defined discriminant function and record accuracy if the testing data matches the labeled data. (Fig-1)

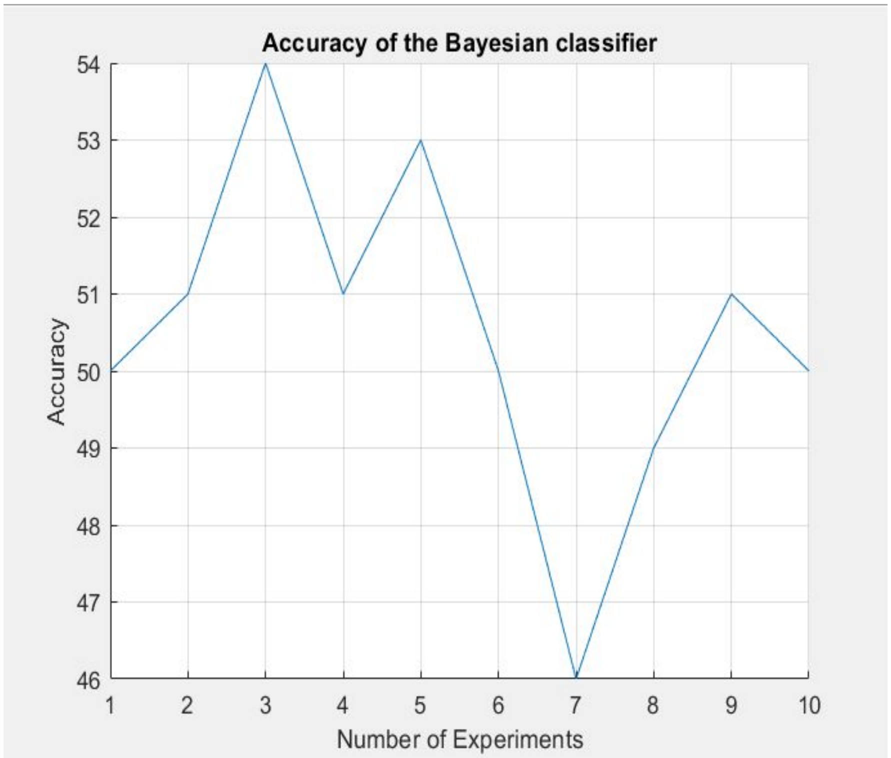


Fig-1

- Further experiments were performed to find a correlation between the number of dimensions (nPCA) post-PCA, and the structure of the training set. Below table-1 has the accuracy scores from the series of experiments performed manually.

	Training set size						
nPCA	130	140	145	150	170	175	180
20	82.8%	83.34%	85.5%	82%	83.34%	84%	80%
50	77%	78.34%	83.36%	78%	85%	**88%**	85%
10	82.85%	81.66%	80%	80%	75%	76%	77.50%
100	73.6%	74%	74.54%	74%	75%	74%	70%

Table-1

- Two things are clear from the table-1:
    i. There’s a noticeable drop in accuracy score as we decrease the nPCA,
    ii. While accuracy score is the highest midway between training sizes 130 to 180 for any given nPCA.

Case Analysis

- Following is the confusion matrix from the combination that resulted in the highest accuracy in table-1.

Accuracy: 88%

nPCA = 50	Predicted		
	Split 175/25	Neutral	Smiley
Actual	Neutral	21	4
	Smiley	2	23

- Below are the 4 neutral faces that were misclassified as smiley:

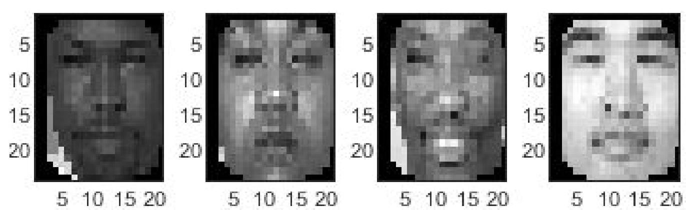


Fig-2

- Below are the two smiley faces that were misclassified as neutral:

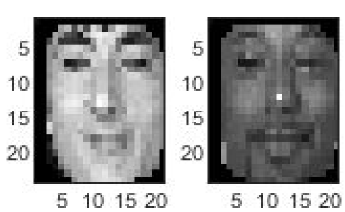


Fig-3

- All six images seem to be ambiguous in terms of neutral vs smiley classification problem. Although, the model does seem to give center pixels utmost importance. The center pixels include areas such as eyes and lips. In fig-2, the third neutral picture that was misclassified as smiley clearly has center pixels around lips/teeth with high values.

2. k-NN classification for Neutral vs Facial Expression:

- We define the number of k-nearest neighbors we would like to evaluate over the training dataset.
- We calculate the 'Euclidean distance' to every sample in the training dataset
- Sort the distances by their indices
- Extract the indices of the k-nearest neighbors
- Finally, we compare the labels we assigned to the labels of the test dataset. Thus, we can find the accuracy of our classifier.

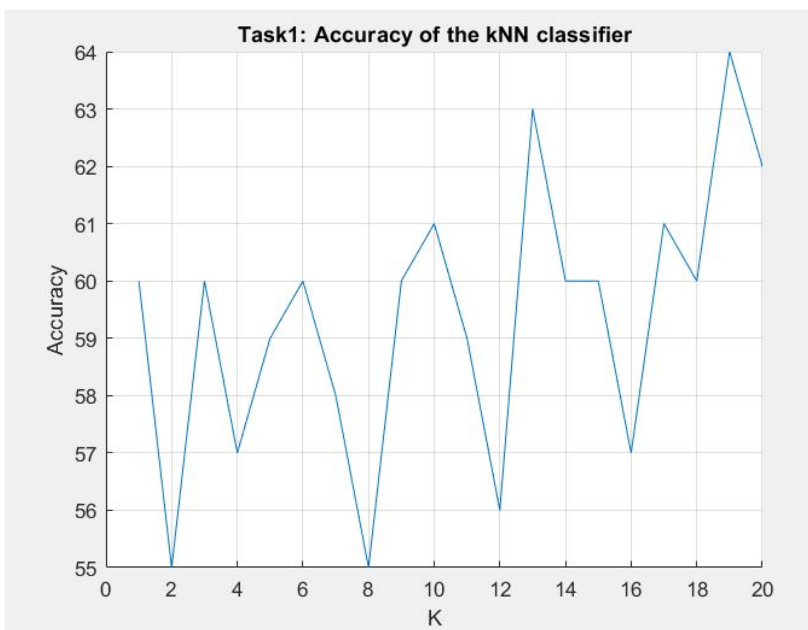


Fig-4

- Further experiments were performed to find a correlation between the number of nearest neighbors considered (k) and the structure of the split, while keeping the nPCA constant. Hence, for nPCA = 50. Following are the accuracy scores for the experiments performed manually.

kNN	**Training set size**				
K	130	145	150	170	175
5	82.14%	81.8%	82%	88.34%	88%
11	82.25%	78.18%	79%	86.67%	**88%**
21	83.57%	81.8%	79%	81.67%	82%
51	80.17%	78.18%	76%	83.33%	84%
91	75%	76.36%	75%	76.67%	76%

Table-2

- With the same combination that was optimal for the Bayes' classifier, nPCA = 50 and split 172/25, once again, it resulted in an accuracy score of 88% when k=11 nearest neighbors are considered.

Case Analysis

- Following is the confusion matrix from the combination that resulted in the highest accuracy in table-2.

Accuracy: 88%

K = 11	Predicted		
Actual	Split 175/25	Neutral	Smiley
	Neutral	25	0
	Smiley	6	19

- All neutral faces that were correctly classified as smiley.
- Whereas the following six smiley faces were misclassified as neutral.

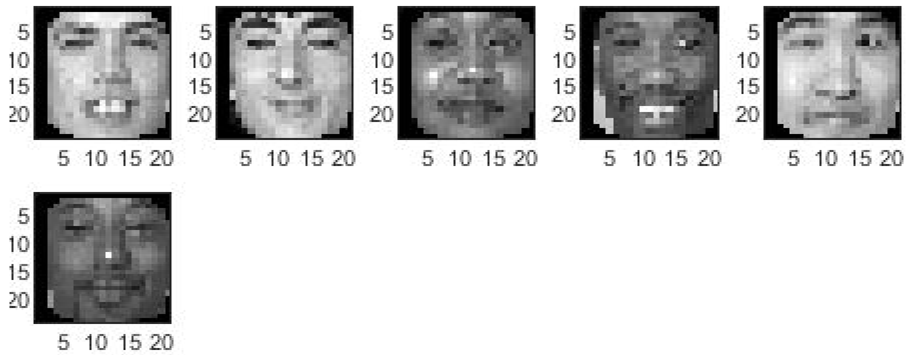


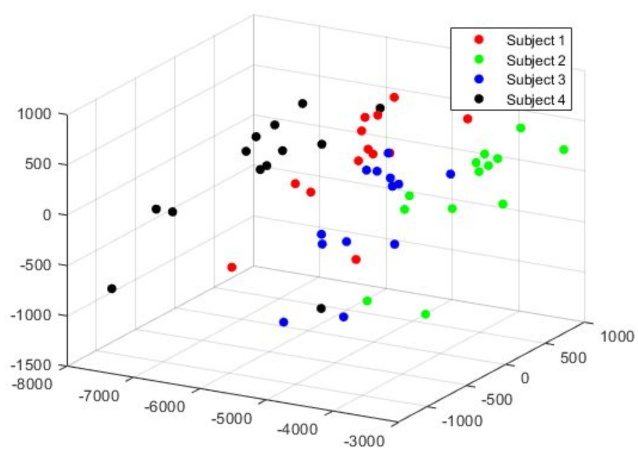
Fig-5

- Since all neutral faces were classified correctly, while almost 1/4th (6/25) smiley faces were misclassified as neutral, it seems like the model performs better on neutral faces than smiley. In other words, it has high sensitivity but low specificity.

Identifying Subjects

- Used pose.mat dataset, we split the dataset into 10 subjects for training and 3 subjects for testing.
- Before splitting the dataset, we reshape it to the total dimensions as a column vector.
- For this task, after the split we perform PCA/MDA before feeding it to the classifiers viz. Bayesian classifier and k-NN classifier.
- Post-PCA, we visualized the first four subjects in three dimensions.

Fig-6



- We can see from figure-6 that post-PCA, the data points look much more classifiable.
- Measured Accuracies:

<table>
<tr><th>Classifiers</th><th>Analysis Type</th></tr>
<tr><td>Bayesian</td><td><table><tr><td>PCA only</td><td>MDA</td></tr><tr><td>20%</td><td>52.9412%</td></tr></table></td></tr>
<tr><td>k-NN</td><td><table><tr><td>PCA only</td><td>MDA (Average)</td></tr><tr><td>61% (max at k =2)</td><td>38.3824% (k=20)</td></tr></table></td></tr>
</table>

Table-3

1. **Bayesian classification for identifying subjects:**
    - After preprocessing the data using PCA/MDA, the means were estimated for each class. (mu1, mu2... mu68)
    - Then, the covariance matrices were estimated for each class. (S1,S2...S68)
    - Further, discriminant functions g(x) were defined for each class.
    - In order to compute g(x), the inverse of each covariance matrix was computed, and matlab threw a warning message that the covariance matrices were almost singular since there are just 10 data-points in the training set per subject. displayed in figure 6.

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
8.912695e-20.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
2.257485e-20.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
4.123790e-19.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
1.677475e-19.
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
```

Fig-6

- Hence, it was decided to use just 5 subjects instead of all 68.
- Means (mu1..mu5) and covariance matrices (S1...S5) were estimated again with the new training data.
- For each class/subject, a discriminant function was defined, with the following decision rule.

  For ith class, if gi(x)>gj(x) => decide i || where i != j || [0<i,j<6]

2. k-NN classification for identifying subjects using Euclidean metric:

- We define the number of k-nearest neighbors we would like to evaluate over the training dataset.
- We calculate the ‘Euclidean distance’ to every sample in the training dataset
- Sort the distances by their indices.
- Extract the indices of the k-nearest neighbors.
- Finally, we compare the labels we assigned to the labels of the test dataset. Thus, we can find the accuracy of our classifier.
- Below, figure-7 is a plot after pre-processing with PCA (nPCA=20) and feeding the test data into k-NN classifier.
- 25 experiments were performed ranging from k = 1 to 25.
- In this case, we see the highest accuracy score was obtained at mere k=2, and dropped from then on.
- Below is the figure-8 after pre-processing with MDA and feeding into k-NN classifier for 20 k - nearest neighbors.
- We can see that performance after preprocessing with MDA is much better than with just PCA.

Fig-7

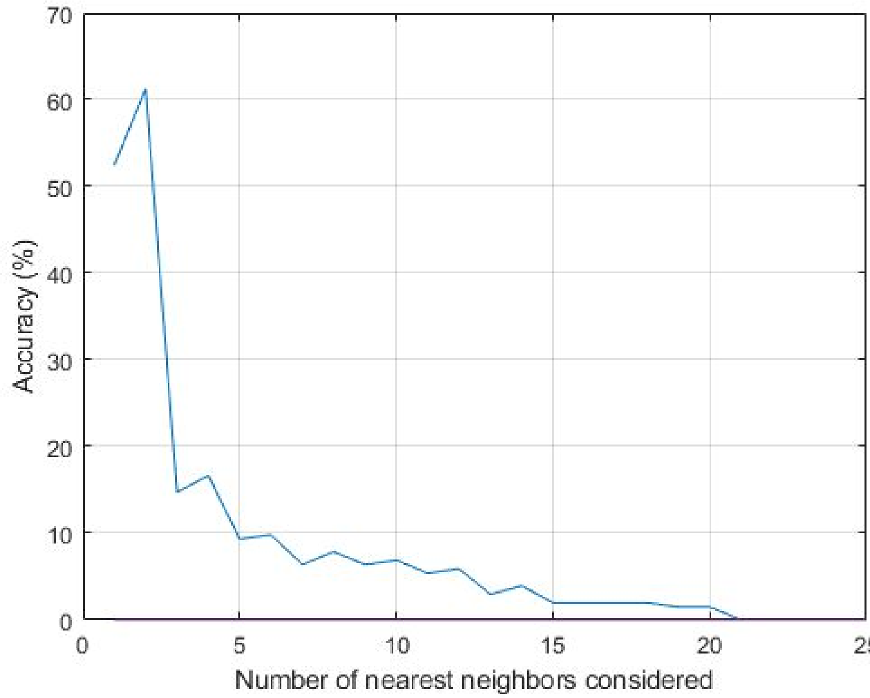
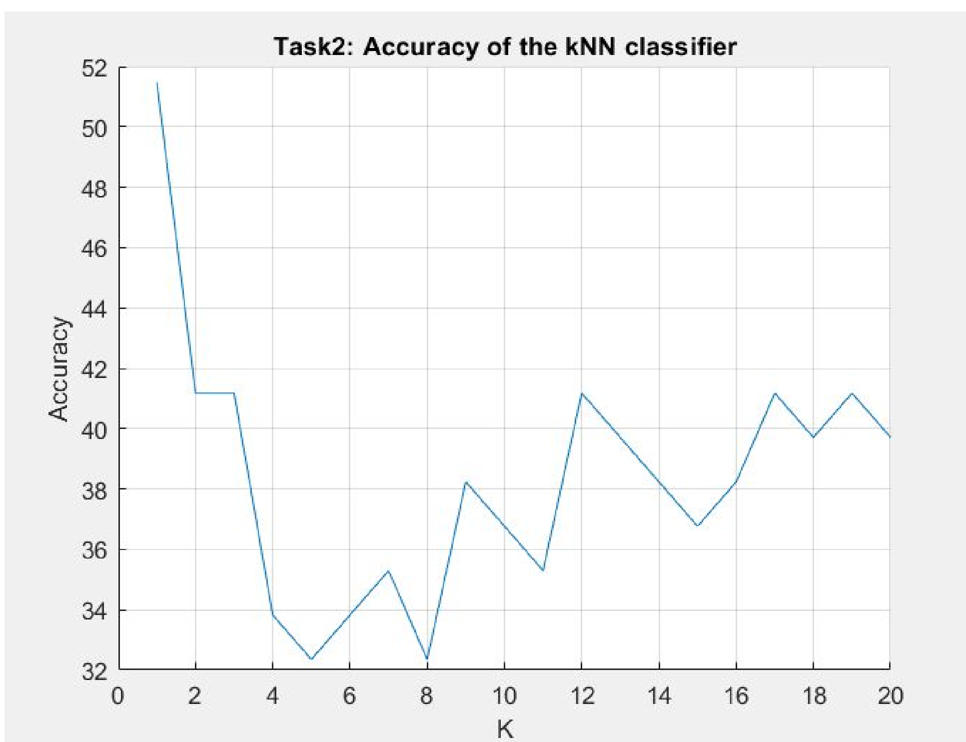

Accuracy (%)
Number of nearest neighbors considered

Fig-8


Task2: Accuracy of the kNN classifier
Accuracy
K

Conclusions

- In this project we implemented the Bayesian and KNN classifiers with dimensionality reduction (PCA and MDA) techniques.
  - Greater variance in training dataset results in higher accuracy.
- For Task 1:
  - For a random 25:75 test-to-train split with 10 iterations, the accuracy for Bayesian classification seems to be varying at a close range as shown in figure for picking randomized column indices as training dataset and testing dataset.
  - For Bayes', accuracy decreases as the nPCA increases. Although, the accuracy score is maximum when the train-test split is between 130 and 170.
  - For k-NN classification, as the number of k increases the accuracy of classification varies between the range of 55 - 65 for different values of k.
  - For neutral vs smiley face tasks, the most optimal combination is when the nPCA is 50 and the training set size is 175. Regardless of the classifier we pick, the accuracy score results in 88%.
  - In the optimal scenario, the kNN model performs better on neutral faces than smiley. Hence, it has high sensitivity but low specificity.
- For Task 2:
  - With a train/test split of 10/3, both models perform very poorly.
  - We noticed that the models were classifying the poses better than the individual themselves. For example, for a query point / test case of subject '2', the nearest training set subject almost never was a pose of subject '2', instead it almost always was a subject that somewhat looked like '2' but had the exact same pose.
  - Therefore, the models tend to give poses precedence over the individual subjects themselves. Hence, a high success rate.
  - With a train/test split of 12/1, we saw better results.
  - The accuracy was the highest for both types of classifiers
  - In Bayesian classification:
    - After PCA and feeding into Bayesian classifier it was observed that the accuracy was at 86.7647%

        - After MDA and feeding into Bayesian classifier, it was observed that the accuracy decreased 52.9412%
    - In kNN classification:
        - After PCA and feeding into the classifier it was observed that the Accuracy decreases with an increase in number of k
        - After MDA and feeding into the classifier, it was observed that the accuracy was deteriorating with increasing value of k and went up again.

MATLAB Code

Shrey Patel : Link to code

Shrey Nair : https://umd.app.box.com/folder/130277262140