# Project 1 - Statistical Pattern Recognition

Sandeep Kota Sai Pavan
University of Maryland
College Park, MD - 20740
Email: skotasai@umd.edu

## I. Introduction

This project is a demonstration of the different machine learning algorithms for applications on facial datasets provided. There are two main classification problems that are solved addressed in this project - face recognition and binary classification of face expressions. An Kernel SVM and a boosted SVM algorithms were used for the task of binary classification of facial expressions. While for the task of the facial recognition is done using a Maximum Likelihood Bayes Classifier and a K-Nearest neighbor Classifier. While these conventional machine learning algorithms perform well with smaller dimension inputs, it is more often than not to get good outputs with high dimensional data like in the case of images. Results were compared after performing dimensionality reduction techniques like MDA and PCA.

## II. Dataset

There are 3 datasets used for analysis namely *Data, Pose and Illumination*. The data sets all contain the labelled images of the people, with expressions, different expressions and in different illuminations respectively. The datasets are read using the functions mentioned in the *utils.py* script in the code. The data format for the face recognition task is $[labels, featuredimages, imwidth \times imheight]$, while for facial recognition SVM the data format is $[labels, imwidth \times imheight]$. The $Data$ dataset contains faces of 200 faces of 3 images features for each person, the $Pose$ dataset contains faces of 68 people with 13 different poses, and the illumination data consists faces of 68 people with 21 different illuminated faces.

## III. Dimensionality Reduction

As explained in the previous sections, the number of features available in the given datasets is quite a lot for these basic classifiers to generalize the task of face recognition. Hence there are 2 dimensionality reduction techniques used in the work.

### A. Principal Component Analysis (PCA)

Principal component analysis can be performed by simply finding the $m$ eigen vectors of the matrix containing stacked up flattened images. The eigen vectors of this stacked up matrix are also referred to as eigenfaces. These eigen faces can be used in linear combination to generate any image in the given training dataset. PCA reduces the dimension of an image to $m$ coefficients corresponding to the $m$ eigenfaces corresponding to the $m$ largest eigen values. A visual representation of the eigen-faces is given shown in figure 1. The implementation of the MDA can be found in the script MDA.py in the submitted code.

### B. Multiple Discriminant Analysis (MDA)

MDA is also another dimensionality reduction technique that uses the within class and between class covariance matrices to find reduce the feature dimension to $m$. It is to be noted that the computation of the transformation involves performing matrix inverse on the within class covariance matrix. In practice, the within class covariance matrix can be a singular matrix to which a regularizing coefficient of value 0.5 is added. The $m$ eigen vectors of the Fisher's discriminant are shown in the figure 2.

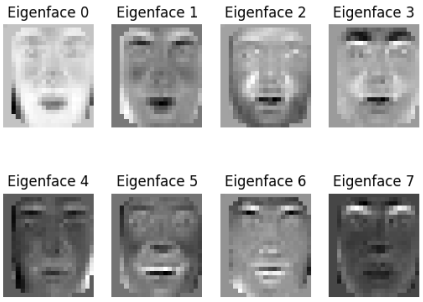
MDA tries to reduce to specific patterns in the images that are to be searched for the specific task whereas PCA tries to reduce to general patterns in the image. This is clearly evident in the PCA and MDA outputs for the face expression task on the Data dataset, where MDA tries to emphasise the pixels around skin wrinkles, while PCA tries to reduce to a generic smiling face.

## IV. Classifiers for Face Recognition

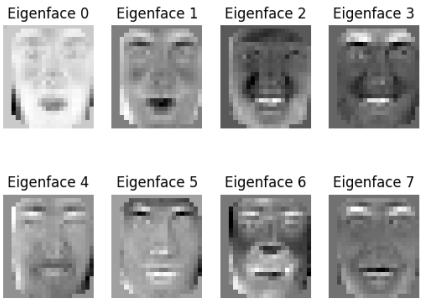
### A. Bayes Classifier without dimensionality reduction

The Bayes classifier is a simple classifier that performs maximum likelihood estimation of the test image with the train dataset. The prior values for each label can be calculated from the number of samples in the dataset. The prediction is based on the label that gives the maximum a posteriori value. In the calculation of the mean parameters from the training dataset, the mean covariance matrix was added with a regularization term in order to eliminate the cases with singular matrices, and the regularizing term value is set to $\lambda = 0.96$.

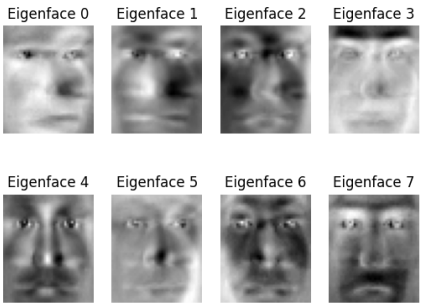
Bayes classifier was applied on all the 3 given datasets for the task of face recognition. The dataset was divided in such a way that approximately 80 % of the dataset is used for training, while the remaining dataset is available for testing. The testing set includes 15 images from the remaining dataset that is available for testing. The time taken for prediction is high when dimensionality reduction has not been applied on the data, and is expected due to the high dimensionality of the data samples in all the datasets. Looking at the results of the classifier as shown in table I applied on the images directly,



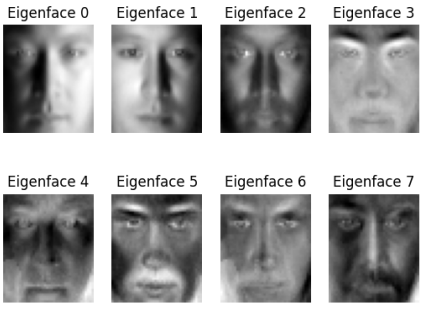

(a) Data dataset for face recognition task




(b) Data dataset for face expression task

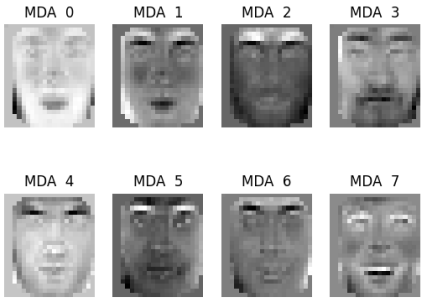



(c) Pose dataset

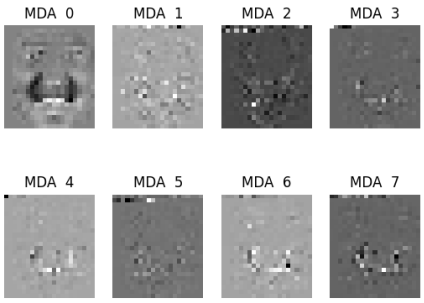



(d) Illumination dataset

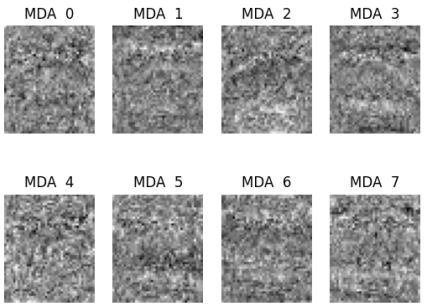
Fig. 1: PCA visualization



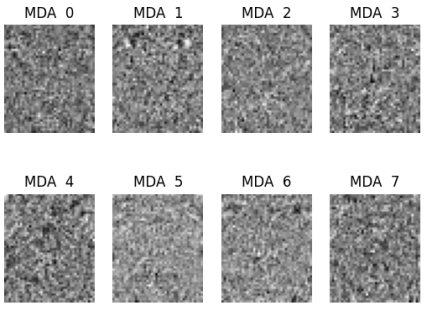

(a) Data dataset for face recognition task




(b) Data dataset for face expression task




(c) Pose dataset




(d) Illumination dataset

Fig. 2: MDA visualization

it can be seen that the number of images per label is very low in the dataset $Data$ and hence the bayes classifier fails to predict the labels correctly on test set. For the $Pose$, and $Illum$ datasets, the first 7 and 11 rows were used respectively for training or finding the mean parameters. However, by reducing the regularization coefficient to $\lambda = 0.4$, the performance classifier seemed to perform better than than the one with a higher regularization coefficient as shown in table I.

| Param | $\lambda = 0.96$ | | $\lambda = 0.4$ | |
|---|---|---|---|---|
| Dataset | Accuracy(%) | No. Wrong | Accuracy(%) | No. Wrong |
| Data | 33.33 | 10/15 | 66.66 | 5/15 |
| Pose | 93.33 | 1/15 | 93.33 | 1/15 |
| Illum | 100 | 0/15 | 100 | 0/15 |

TABLE I: Accuracy Score of Bayes Classifier without Dimensionality Reduction.$\lambda$ corresponds is the regularization coefficient of the mean covariance matrix of ML estimator

### *B. Bayes Classifier with MDA*

By applying MDA, the time taken for prediction reduced significantly depending on the number of dimensions to which the data was reduced to. This is the main benefit of having reduced dimensions, but at the cost of some accuracy performance. It is expected as dimensionality reduction techniques only approximate the given input image in the reduced dimension. The performance results on the 3 datasets after applying the dimensionlaity reduction techniques is shown table II . Unlike in the previous experiments, this time the training set is set larger at 50 samples and the performance was analyzed.

| Param | $\lambda = 0.4$ | | | $\lambda = 0.96$ | | |
|---|---|---|---|---|---|---|
| MDA Dims (m) | 100 | 150 | 200 | 100 | 150 | 200 |
| Data | 66 | 66 | 58 | 62 | 70 | 62 |

TABLE II: Accuracy score of Bayes Classifier with MDA. The parameter $\lambda$ corresponds to the regularization coefficient added to the mean covariance matrix of maximum likelihood estimator, $m$ is the dimension of the transformed data samples

Observing the results, it can be found that applying bayes classifier on the $Data$ dataset does not show a significant improvement in accuracy results compared to the non-dimensionality reduction model. It is observed that the accuracy score doesn't increase proportional to the number of dimensions in the reduced data. It showed better accuracy score at $m = 150$. Playing with two different values of the regularization coefficient $\lambda$ shows that a higher value corresponds to a better model. Although the number of test samples is low to draw a conclusive result, it was a common trend observed in the given datasets. Also the prediction score in the $Illum$ dataset is high due to the large training samples per label in the dataset.

### *C. Bayes Classifier with PCA*

As with the case with MDA, the execution time for prediction is reduced by a significant amount, depending on the dimension of the reduced data $m$. A similar range of experiments were performed for PCA and the results were documented in the table III

| Param | $\lambda = 0.4$ | | | $\lambda = 0.96$ | | |
|---|---|---|---|---|---|---|
| $\eta$ | 10 | 15 | 20 | 10 | 15 | 20 |
| Data | 56 | 72 | 62 | 64 | 58 | 66 |
| Pose | 94 | 92 | 88 | 100 | 92 | 90 |
| Illum | 100 | 100 | 100 | 100 | 100 | 100 |

TABLE III: Accuracy score of Bayes Classifier with PCA. The parameter $\lambda$ corresponds to the regularization coefficient added to the mean covariance matrix of maximum likelihood estimator, $\eta$ is the error coefficient used in the selection of $m$ dimensions in PCA

### *D. K - Nearest Neighbor (KNN) without dimensionality reduction*

KNN classifier selects the label of a test image based on the labels of its K nearest neighbors based on its distance. The metric to compute the distance between its neighbors is a simple norm distance between the two images. The classification performance for different values of K is shown in the table IV. The accuracy values shown are an average accuracy computed over 5 trails of KNN.

| Param(K) | K=1 | K=3 | K=5 | K=7 | K=9 | K=11 |
|---|---|---|---|---|---|---|
| Data | 56.3 | 46.5 | 47 | 43 | 38 | 35.2 |
| Pose | 89.4 | 78.8 | 67.7 | 60.2 | 56.99 | 47.6 |
| Illum | 100 | 87.4 | 96.4 | 82 | 76.4 | 57.2 |

TABLE IV: Accuracy score of KNN classifier without applying any dimensionality reduction.

It is observed that as K increases, the classifier seems to perform poorly. And as a general trend, the performance is best on illum dataset followed by pose dataset and is worst on data dataset. KNN performance was a littlebad as compared to the bayes classifier, which is an expected trend.

### *E. K - Nearest Neighbor with MDA*

Repeating the same set of experiments after applying MDA and on different dimensions $m$, the results were compiled in the table V

| Param(K) | K=1 | K=3 | K=5 | K=7 | K=9 | K=11 |
|---|---|---|---|---|---|---|
| Data (m=100) | 57.6 | 44.8 | 44 | 46 | 33.2 | 32.4 |
| Data (m=150) | 56 | 46.7 | 44.4 | 46 | 35.6 | 41.2 |
| Data (m=200) | 61.2 | 44 | 46.4 | 41.6 | 33.2 | 37.2 |
| Pose (m=100) | 92.8 | 98.7 | 94.0 | 98 | 93.5 | 93.5 |
| Pose (m=150) | 98.8 | 91.6 | 81.6 | 97.2 | 93.9 | 93.6 |
| Pose (m=200) | 96.9 | 84.8 | 90 | 90.4 | 84 | 91.2 |
| Illum (m=100) | 100 | 100 | 100 | 100 | 100 | 99.6 |
| Illum (m=150) | 100 | 100 | 100 | 100 | 100 | 98.4 |
| Illum (m=200) | 100 | 100 | 100 | 100 | 99.2 | 99.2 |

TABLE V: Accuracy score of KNN classifier with MDA. The parameter $m$ is the number of dimensions.

It was observed that the performance on the data datset is relatively similar to the previous case. However a slight bump in the accuracy was observed on the pose dataset. No

specific trend was observed with the increase in the dimension to which the data was reduced. The results shown in the table are not exact values as the accuracies are evaluated as an average over 5 evaluation runs, with randomly generated test data. In general, KNN classifier with MDA performeed better than KNN without dimensionality reduction.

### *F. K - Nearest Neighbor with PCA*

Repeating the same set of experiments after applying MDA and on different dimensions $m$, the results were compiled in the table VI

| Param(K) | K=1 | K=3 | K=5 | K=7 | K=9 | K=11 |
|---|---|---|---|---|---|---|
| Data ($\eta = 10$) | 58 | 45.1 | 42 | 40.8 | 42.4 | 36 |
| Data ($\eta = 15$) | 56 | 40 | 44.4 | 42.4 | 36.8 | 33.9 |
| Data ($\eta = 20$) | 56 | 43.5 | 40.8 | 38 | 40.4 | 35.6 |
| Pose ($\eta = 10$) | 91.6 | 73.2 | 53.6 | 74.8 | 58.4 | 71.2 |
| Pose ($\eta = 15$ | 89.2 | 64.7 | 60.4 | 49.2 | 58.8 | 62.7 |
| Pose ($\eta = 20$) | 91.6 | 66.4 | 70.8 | 60.8 | 50.8 | 56.4 |
| Illum ($\eta = 10$) | 100 | 74.8 | 71.2 | 80 | 55.1 | 77.6 |
| Illum ($\eta = 15$) | 100 | 86 | 93.2 | 57.6 | 64 | 63.1 |
| Illum ($\eta = 20$) | 100 | 87.6 | 80.8 | 76 | 62.8 | 78.2 |

TABLE VI: Accuracy score of KNN classifier with PCA. The parameter $\eta$ is the error coefficient used in the selection of $m$ dimensions in PCA

KNN with PCA performed similar to the the KNN without any dimensionality reduction. In general increasing the error coefficient seemed to reduce the performance in general. A probavle reason why KNN-PCA did not perform as good as KNN-MDA is becaue PCA tries to reduce dimension into general features while MDA tries to reduce dimension to specific features. Because KNN is based on the distance between the features, finding distance between specific features helps in better classification.

## V. Classifiers for Face Expression Task

Detection of face expression is a binary classification task and hence the classifiers used in the previous task can be used in this case also with a slight modification in the data reading pipeline. Additionally we also compare results with Kernel SVM and Boosted SVM.

### *A. Bayes Classifier without dimensionality reduction*

Bayes classifier with maximum likelihood parameter estimation was applied. A similar set of experiments were performed with and without dimensionality reduction. The test set consists of 50 faces. The results of bayes classifier without dimensionality reduction is compiled in table VII

| Param | $\lambda = 0.4$ | $\lambda = 0.96$ |
|---|---|---|
| Exp Data | 84 | 78 |

TABLE VII: Accuracy score of Bayes Classifier without dimensionality reduction for face expression task. The parameter $\lambda$ corresponds to the regularization coefficient added to the mean covariance matrix of maximum likelihood estimator

It can be observed that bayes classifier performs much better for the face expression task than face recognition task on the same data dataset.

### *B. Bayes Classifier with MDA*

When performing MDA with 3 different dimensions, the results are compiled in the table VIII

| Param | $\lambda = 0.4$ | | | $\lambda = 0.96$ | | |
|---|---|---|---|---|---|---|
| MDA Dims (m) | 100 | 150 | 200 | 100 | 150 | 200 |
| Data | 84 | 82 | 80 | 84 | 82 | 82 |

TABLE VIII: Accuracy score of Bayes Classifier with MDA. The parameter $\lambda$ corresponds to the regularization coefficient added to the mean covariance matrix of maximum likelihood estimator, $m$ is the dimension of the transformed data samples

From the results that were computed after an average of 5 trials, we can see that the reducing the dimension does not have a major change in performance in the case where $\lambda = 0.4$, but a good bump in the performance was observed in the case $\lambda = 0.96$ with dimensionality reduction. In general, bayes classifier performance is quite similar even with MDA.

### *C. Bayes Classifier with PCA*

When performing PCA with 3 different dimensions, the results are compiled in the table IX.

| Param | $\lambda = 0.4$ | | | $\lambda = 0.96$ | | |
|---|---|---|---|---|---|---|
| $\eta$ | 10 | 15 | 20 | 10 | 15 | 20 |
| Data | 86 | 84 | 82 | 82 | 80 | 78 |

TABLE IX: Accuracy score of Bayes Classifier with PCA. The parameter $\lambda$ corresponds to the regularization coefficient added to the mean covariance matrix of maximum likelihood estimator, $\eta$ is the error coefficient used in the selection of $m$ dimensions in PCA

The results dont show a significant increase in performance with dimensionality reduction. Also the number of trials used to compile the accuracy is quite low to extract useful patterns.

### *D. K-Nearest Neighbor*

Applying KNN classifier for the given task, the results were compiled in the table X

| Param(K) | K=1 | K=3 | K=5 | K=7 | K=9 | K=11 |
|---|---|---|---|---|---|---|
| Data | 71.6 | 78.8 | 80.8 | 77.6 | 74.4 | 76 |
| Data (m=100) | 69.6 | 80.8 | 81.6 | 75.6 | 77.59 | 73.6 |
| Data (m=150) | 56 | 46.7 | 44.4 | 46 | 35.6 | 41.2 |
| Data (m=200) | 70 | 77.2 | 81.9 | 77.2 | 76.8 | 74 |
| Data ($\eta$=10) | 69.2 | 80.4 | 76 | 78.7 | 76 | 68.8 |
| Data ($\eta$=15) | 71.6 | 76.8 | 81.6 | 76.4 | 76.5 | 73.2 |
| Data ($\eta$=20) | 73.6 | 80 | 80 | 77.6 | 78.8 | 74.8 |

TABLE X: Accuracy score of KNN classifier with PCA. The parameter $m$ is the number of dimensions.

Looking at the results of KNN classifier for the face expression task, it performs decently well with an overall accuracy of around 75%. Even with dimensionality reduction using either PCA or LDA showed a minute difference in performance with the existing parameters, upon tuning it with multiple trials for different values of $m$ and $\eta$ a better classifier can be achieved.

### E. Kernel - Support Vector Machines (SVM)

Support vector machine (SVM) are supervised learning models that work well for binary classification tasks. SVM's work by finding the maximum margin classifier that can separate the 2 class of data. In real world applications, the 2 classes of data my not be linearly separable. In such cases a regularization term with parameter $C$ is added to the Lagrangian dual problem of SVM to compensate to the number of mispredicions in the margin region. The SVM used in this project is a kernelized version of SVM where 3 different kernels namely RBF, Polynimoial and linear kernels are used to find the decision hyperplane. In all the results shown below, the accuracy values are averaged accuracy value of a 4 fold cross validation method.

| C | C=0.1 | C=1 | C=10 |
|---|---|---|---|
| Accuracy | 0.8125 | 0.8625 | 0.8625 |

TABLE XI: Accuracy score linear SVM and its change with parameter C.

| Param($\sigma$) | 2 | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| Accuracy(C=1) | 0.5025 | 0.775 | 0.89 | 0.87 | 0.8575 |
| Accuracy(C=10) | 0.637 | 0.78 | 0.895 | 0.89 | 0.895 |
| Accuracy(C=100) | 0.637 | 0.78 | 0.895 | 0.89 | 0.895 |

TABLE XII: Accuracy score RBF Kernel SVM classifier with different kernel parameters and regularization parameter C.

| Param ($r$) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Accuracy(C=0.1) | 0.8125 | 0.8825 | 0.8725 | 0.895 | 0.8525 |
| Accuracy(C=1) | 0.8625 | 0.88 | 0.877 | 0.9 | 0.845 |
| Accuracy(C=10) | 0.8625 | 0.8825 | 0.8725 | 0.9 | 0.8475 |

TABLE XIII: Accuracy score Poly Kernel SVM classifier with different kernel parameters and regularization parameter C.

A number of parameters were tested for the kernel parameters and the results were tabulated in the tables XI, XII and XIII. Looking at the accuracy results of SVM classifier, the Polynomial kernel seemed to be performing better for the task of recognizing facial expression. The best results were achieved by the polynomial kernel with the parameter $r = 4$ with the $C = 10$. As a general trend, increasing C value beyond a value had minimal effect in the performance of the classifier.

### F. Boosted SVM

Adaboost algorithm is a boosting technique where a number of weak classifiers can be used in linear combination and make it a more complex model. The downside of this is method being the chances of overfitting are rally high and that the performance may not be same with test set. In the training graph shown in figure 3 a linear SVM was chosen which has an intial accuracy of around 60%. This weak classifier was selected by increasing the threshold value for choosing the support vectors to $1e-2$. An increasing trend was observed in the training accuracy, while the test accuracy remained around the same.

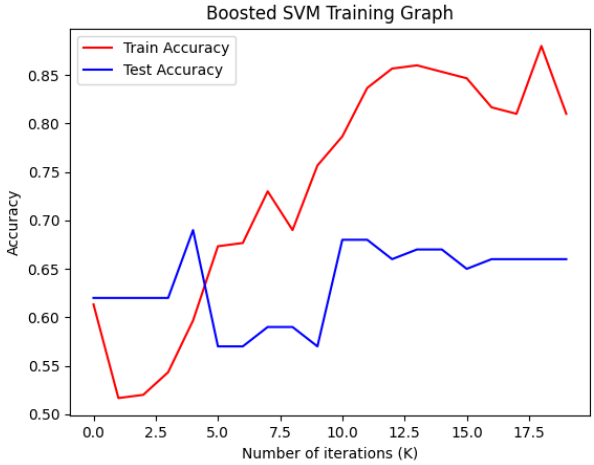



Fig. 3: Adaboost SVM Training Graph