SMAI Assignment 2

Swetanjal Dutta
International Institute of Information Technology, Hyderabad
Gachibowli, Telangana
swetanjal.dutta@research.iiit.ac.in

Abstract

This is the report for SMAI Assignment 2. It presents a detailed analysis of the experiments run for the problems of classification and verification for three popular face data sets. The report contains five major sections. The introduction lays down the aim of this assignment and the expectations fulfilled after completing the assignment. The next three sections, each section for each data set presents a detailed analysis of the experiments run for that data set. The last section presents extension of experiments run to do gender classification.

1. Introduction

The objective of this assignment was to get myself familiarise with problems of classification and verification using popular face data sets. After completing this assignment

- I find myself comfortable with the problem space and it's nuances
- Got some exposure to data visualization
- Got a flavour of trying various classifiers to solve Machine Learning Tasks

2. Indian Movie Face Database(IMFDB)

This is a data set containing images of faces of 8 Indian Actors. There are a total of 400 samples in this Indian data set, 50 samples for each actor. Each image is of size 32 x 32 and contains three channels(RGB). Flattening each sample gives us a 3072 dimensional vector to work with.

It takes first 124 eigen vectors to capture 95% of information from this data set. We do PCA, and do dimensional reduction of each vector to 124 dimensions and then reconstruct images back to get an average reconstruction error of 0.0384.

Amitabh Bachchan's image is hardest to represent compactly using less eigen vectors. Intuitively, this is because his images have a lot of variation not only in terms of his orientation, position and size of face/head but also his expressions(especially mouth). Different lighting conditions, presence of a black and white images and presence of blurred images are other reasons which add to the difficulty of compactly representing this actor's face with few eigen vectors.

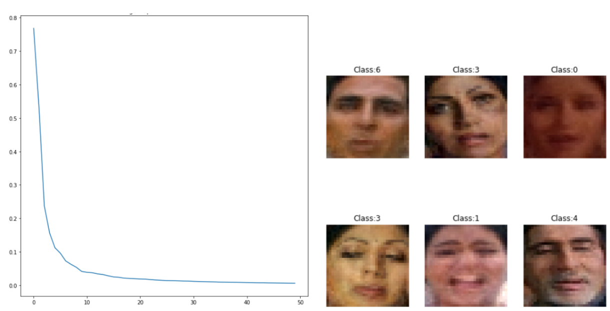

Figure 1. The image on the left shows eigen spectrum and the figure on right shows images reconstructed after dimensional reduction to 124 dimensions

2.1. Classifier

We trained few classifiers namely SVM, Multi Layer Perceptron, Logistic Regression and Decision Trees for the purposes of classifying faces of the actors. The train test split was done in the ratio 9:1 with a seed of 23. Instead of using the raw pixel values as features we tried various feature extraction methods namely PCA, Kernel PCA, LDA, Kernel LDA, ResNet and VGG-19. The table below gives a detailed comparison of the performance of each method.

We find when ResNet and VGG features are combined we get really good results showing the sophistication of these feature extractors which make them so famous in the area of Computer Vision. A classifier as simple as Decision Trees also gives an accuracy of 95% when ResNet + VGG features are combined and used.

Method	Dimension	Error	F1 Score
PCA + MLP	124	10.0	0.8994
PCA + SVM	124	17.5	0.8205
PCA + LR	124	15.0	0.8374
LDA + MLP	5	5.0	0.9497
LDA + SVM	7	2.5	0.9747
LDA + LR	7	7.5	0.9244
KPCA + MLP	20	10.0	0.8976
KPCA + LR	40	45.0	0.5426
KLDA + MLP	5	5.0	0.9497
VGG + MLP	4096	12.5	0.8717
VGG + SVM	4096	10.0	0.8964
VGG + LR	4096	10.0	0.9011
ResNet + MLP	2048	5.0	0.9494
ResNet + SVM	2048	5.0	0.9513
ResNet + LR	2048	7.5	0.9207
ResNet + VGG + MLP	6144	2.5	0.9747
ResNet + VGG + SVM	6144	0.0	1.0000
ResNet + VGG + LR	6144	0.0	1.0000
All features + DTree	6310	5.0	0.9513

Table 1. Comparison of various classification methods

Method	Dimension	Error	F1 Score
PCA	124	32.5	0.686546
LDA	5	2.5	0.974747
KPCA	20	32.5	0.675237
KLDA	5	2.5	0.974747
VGG	4096	7.5	0.924495
ResNet	2048	10.0	0.901015
ResNet + VGG	6144	0.0	1.000000
All features	6310	0.0	1.000000

Table 2. Comparison of various features extractors for verification using KNN

2.2. Verification

Similarly, a KNN classifier with k as 5 is trained for the purpose of face verification. Again, we get a perfect accuracy score when using ResNet + VGG for verification purposes.

Our verification process involves finding the K Nearest Neighbours of the face to be verified. If the ground truth class label is same as the label of the majority nearest neighbours, then verification succeeds. Else, it fails.

The table below shows comparison of various verification methods. The variation in the methods is only in the way features are extracted.

2.3. Data Visualisation

The TSNE plot using ResNet + VGG features show how the data samples of same classes come together in high dimensional space. Similarly, the Fisher Linear Discriminant Analysis does a decent job in cluttering samples with same label in 3 dimensional space. Refer to the figures below for the plots.

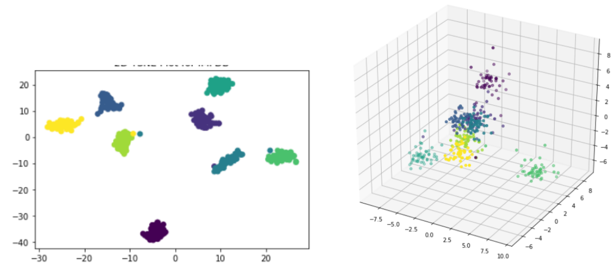

Figure 2. The image on the left shows TSNE plot(in 2D) and the figure on right shows Kernel LDA plot in 3D

3. IIIT Cartoon Face Dataset

This is a data set containing images of cartoon faces of 8 subjects. There are a total of 800 samples in this data set, 100 samples for each subject. Each image is of size 32 x 32 and contains three channels(RGB). Flattening each sample gives us a 3072 dimensional vector to work with.

It takes first 309 eigen vectors to capture 95% of information from this data set. We do PCA, and do dimensional reduction of each vector to 309 dimensions and then reconstruct images back to get an average reconstruction error of 0.0676.

We notice the high reconstruction error for this data set and also the unusually high number of eigen vectors being required to capture 95% of the information. This is because cartoons show a lot of variety and the eigen vectors can't capture patterns like mouth, eyes etc. at certain pixel locations so easily. More the variety in the images, more difficult it is to represent the data set using few eigen vectors. In contrast something like the IMFDB database is easier to represent compactly because faces are more or less of similar size and eigen vectors can capture eyes, mouth etc at certain pixel locations.

Narendra Modi's facial images are hardest to represent compactly with few eigen vectors in the IIIT-CFW Dataset. This is because the cartoon images of the Prime Minister shows a variety of colors, starting from realistic skin color to black and white and also pink color(that of a pig). The sizes of some of the images are extremely small and most of the faces do not have a fixed dimension and are either too big or too small. These reasons make it difficult for eigen vectors to capture these pixel features precisely. Besides, Narendra Modi's images do not have something very unique in every image. This is in contrast to someone like Manmohan Singh whose images' top few pixels are blue because of his turban or someone like Obama or Putin who have a specific head structure. Narendra Modi's cartoons are definitely the wildest of the lot(for example, the sample

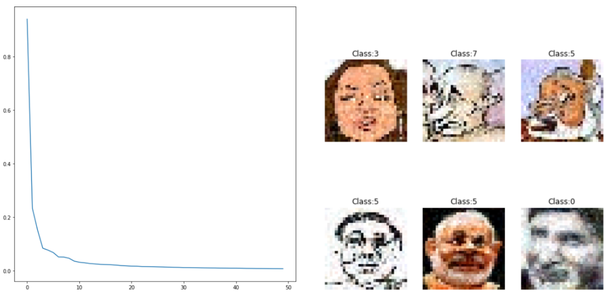

Figure 3. The image on the left shows eigen spectrum and the figure on right shows images reconstructed after dimensional reduction to 309 dimensions

where two horns are coming out of his head and the sample where he is shown to swallow ghosts and also the sample where he is blindfolded)

3.1. Classifier

We trained few classifiers namely SVM, Multi Layer Perceptron, Logistic Regression and Decision Trees for the purposes of classifying faces of the actors. The train test split was done in the ratio 9:1 with a seed of 23. Instead of using the raw pixel values as features we tried various feature extraction methods namely PCA, Kernel PCA, LDA, Kernel LDA, ResNet and VGG-19. The table below gives a detailed comparison of the performance of each method.

We find when ResNet and VGG features are combined we get really good results showing the sophistication of these feature extractors which make them so famous in the area of Computer Vision. A classifier as simple as Decision Trees also gives an accuracy of 98% when ResNet + VGG features are combined and used.

3.2. Verification

Similarly, a KNN classifier with k as 5 is trained for the purpose of face verification. Again, we get a perfect accuracy score when using ResNet + VGG for verification purposes.

Our verification process involves finding the K Nearest Neighbours of the face to be verified. If the ground truth class label is same as the label of the majority nearest neighbours, then verification succeeds. Else, it fails.

The table below shows comparison of various verification methods. The variation in the methods is only in the way features are extracted.

3.3. Data Visualisation

The TSNE plot using ResNet + VGG features show how the data samples of same classes come together in high dimensional space. Similarly, the Fisher Linear Discriminant

Method	Dimension	Error	F1 Score
PCA + MLP	309	36.764706	0.629641
PCA + SVM	309	39.705882	0.600273
PCA + LR	309	48.529412	0.504505
LDA + MLP	5	4.411765	0.956326
LDA + SVM	7	4.411765	0.956275
LDA + LR	7	2.941176	0.970121
KPCA + MLP	20	36.764706	0.635744
KPCA + LR	40	55.882353	0.410622
KLDA + MLP	5	7.352941	0.928928
VGG + MLP	4096	32.352941	0.685771
VGG + SVM	4096	41.176471	0.574205
VGG + LR	4096	35.294118	0.650302
ResNet + MLP	2048	1.470588	0.984827
ResNet + SVM	2048	1.470588	0.984827
ResNet + LR	2048	1.470588	0.984827
ResNet + VGG + MLP	6144	1.470588	0.984827
ResNet + VGG + SVM	6144	0.000000	1.000000
ResNet + VGG + LR	6144	0.000000	1.000000
All features + DTree	6495	1.470588	0.984827

Table 3. Comparison of various classification methods

Method	Dimension	Error	F1 Score
PCA	309	64.705882	0.342077
LDA	5	0.000000	1.000000
KPCA	20	45.588235	0.527953
KLDA	5	0.000000	1.000000
VGG	4096	33.823529	0.645253
ResNet	2048	1.470588	0.980159
ResNet + VGG	6144	0.000000	1.000000
All features	6495	0.000000	1.000000

Table 4. Comparison of various features extractors for verification using KNN

Analysis does a decent job in cluttering samples with same label in 3 dimensional space. Refer to the figures below for the plots.

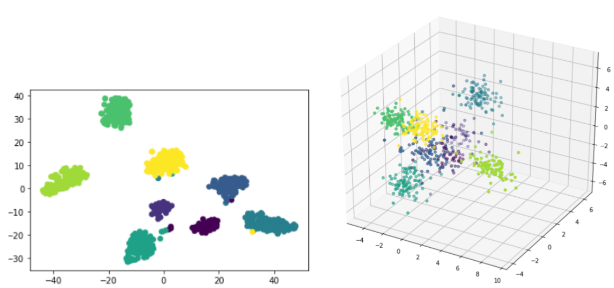

Figure 4. The image on the left shows TSNE plot(in 2D) and the figure on right shows Kernel LDA plot in 3D

4. Yale Face Database

This is a data set containing images of faces of 15 subjects. There are a total of 165 samples in this data set, 11 samples for each subject. Each image is of size 32 x 32 and contains three channels(RGB). Flattening each sample gives us a 3072 dimensional vector to work with.

It takes first 62 eigen vectors to capture 95% of information from this data set. We do PCA, and do dimensional reduction of each vector to 62 dimensions and then reconstruct images back to get an average reconstruction error of 0.0536.

Class 14 images are hardest to represent compactly with few eigen vectors in Yale Dataset. This is because the face is shown to wear glasses in some images and in some images he is shown without glasses. Also, he is seen to yawn and wink in some images. This makes it difficult for eigen vectors to capture specific pattern.

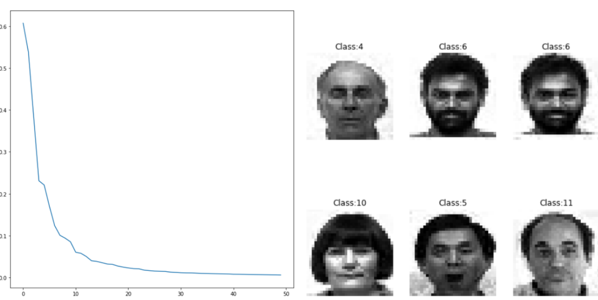

Figure 5. The image on the left shows eigen spectrum and the figure on right shows images reconstructed after dimensional reduction to 62 dimensions

4.1. Classifier

We trained few classifiers namely SVM, Multi Layer Perceptron, Logistic Regression and Decision Trees for the purposes of classifying faces of the actors. The train test split was done in the ratio 9:1 with a seed of 23. Instead of using the raw pixel values as features we tried various feature extraction methods namely PCA, Kernel PCA, LDA, Kernel LDA, ResNet and VGG-19. The table below gives a detailed comparison of the performance of each method.

We find when ResNet and VGG features are combined we get really good results showing the sophistication of these feature extractors which make them so famous in the area of Computer Vision. A classifier as simple as Decision Trees also gives an accuracy of 76.47% when ResNet + VGG features are combined and used.

4.2. Verification

Similarly, a KNN classifier with k as 5 is trained for the purpose of face verification. Again, we get a perfect accuracy score when using ResNet + VGG for verification purposes.

Method	Dimension	Error	F1 Score
PCA + MLP	62	5.882353	0.941176
PCA + SVM	62	17.647059	0.788235
PCA + LR	62	0.000000	1.000000
LDA + MLP	5	0.000000	1.000000
LDA + SVM	7	5.882353	0.941176
LDA + LR	7	0.000000	1.000000
KPCA + MLP	20	0.000000	1.000000
KPCA + LR	40	23.529412	0.705882
KLDA + MLP	5	0.000000	1.000000
VGG + MLP	4096	47.058824	0.470588
VGG + SVM	4096	41.176471	0.539216
VGG + LR	4096	41.176471	0.552941
ResNet + MLP	2048	0.000000	1.000000
ResNet + SVM	2048	0.000000	1.000000
ResNet + LR	2048	0.000000	1.000000
ResNet + VGG + MLP	6144	0.000000	1.000000
ResNet + VGG + SVM	6144	0.000000	1.000000
ResNet + VGG + LR	6144	0.000000	1.000000
All features + DTree	6248	23.529412	0.682353

Table 5. Comparison of various classification methods

Method	Dimension	Error	F1 Score
PCA	62	23.529412	0.744444
LDA	5	0.000000	1.000000
KPCA	20	23.529412	0.731111
KLDA	5	0.000000	1.000000
VGG	4096	41.176471	0.511111
ResNet	2048	0.000000	1.000000
ResNet + VGG	6144	0.000000	1.000000
All features	6248	0.000000	1.000000

Table 6. Comparison of various features extractors for verification using KNN

Our verification process involves finding the K Nearest Neighbours of the face to be verified. If the ground truth class label is same as the label of the majority nearest neighbours, then verification succeeds. Else, it fails.

The table below shows comparison of various verification methods. The variation in the methods is only in the way features are extracted.

4.3. Data Visualisation

The TSNE plots show how the data samples of same classes come together in high dimensional space. Similarly, the Fisher Linear Discriminant Analysis does a decent job in cluttering samples with same label in 3 dimensional space. Refer to the figures below for the plots.

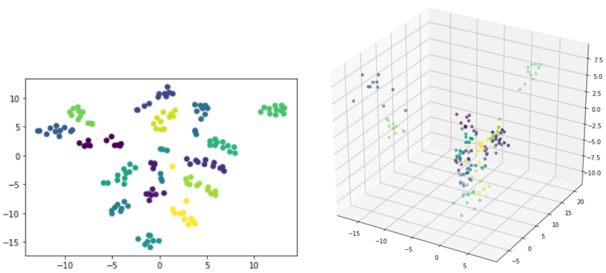

Figure 6. The image on the left shows TSNE plot(in 2D) and the figure on right shows Kernel LDA plot in 3D

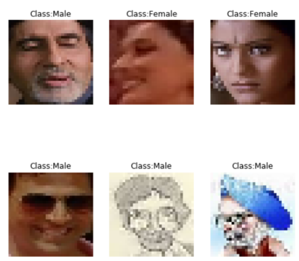

Figure 7. Few samples with their labels from the dataset

5. Gender prediction on combined IMFDB and IIIT-CFW Dataset

This part of the assignment is an extension where we were asked to choose one of the topics and extend the above solutions of classification and verification to solve a new problem. I chose to do gender prediction on the IMFDB and IIIT-CFW data set combined. What I learnt to appreciate is the fact that the basic classifiers taught in class can be used to solve a wide range of problems. Also, trying out various classifiers adjusting the hyper parameters can enable us to achieve state of the art performance.

I ran the SVM classifier using ResNet + VGG Features to do gender prediction. I achieved an accuracy of 99.069%. This shows the power of traditional methods like SVMs even in the era of deep learning.

What makes the problem hard is because the images are slightly blurred and we are using cartoon images as well(which shows a lot of variations). Even then our classifier was able to recognise patterns.

This sort of classifier can be used in Apps of saloons and beauty parlours when one wants to upload their face and try out various hair cuts and facials. The haircut/facial options depend on the gender and hence an automatic gender recognition will avoid the requirement of manually selecting gender.

It may also be used for security purposes where people try to hide/fake their genders.

The pipeline for any machine learning task is to load data, decide on the features, perform train-test split, train a classifier and validate results. We follow this pipeline and solve this problem.

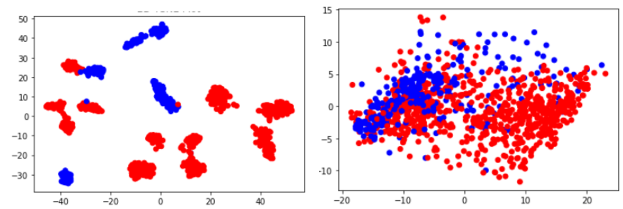

Figure 8. The plot on the left is TSNE plot in 2D, and the one on right is PCA plot in 2D

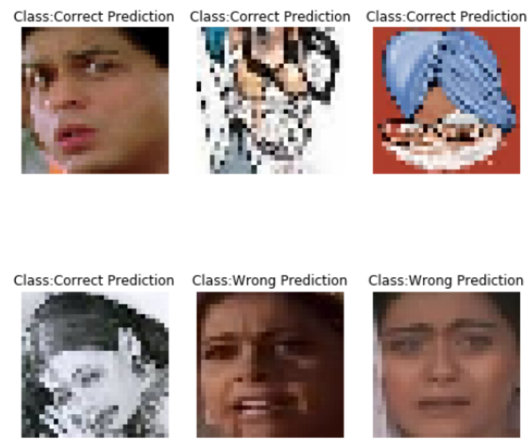

Figure 9. Test Results

We split training testing data in the ratio 9:1 with a seed of 3.

5.1. Data Visualization and Interpretation

The TSNE plot using ResNet + VGG features shows how the samples come together in high dimension space. The samples which belong to 'Male' class is in red and the samples which belong to 'Female' class is in blue. The cluttering of similar sample points in specific regions shows that there is some pattern in the data.

In contrast the PCA plot doesn't give us any information as all samples are spread through out the plot. In other words, there is no pattern that is getting captured in two dimensions which can be used for classification purposes.

5.2. Results

As already mentioned earlier, we achieve an accuracy of 99.069% on the training data set. We are wrongly classifying two images only and both of them are False positive of the class 'Male'. Closer analysis shows the misclassification happens in a scene where lighting is poor and face of the actress Kajol is without any make up. Also, her long hair is not visible in both the images. This clearly indicates features that are obviously getting picked up by the classifier. Below are some samples correctly classified and incorrectly classified.