Statistical Methods in AI 04.11.2019

SMAI Assignment 2 Report

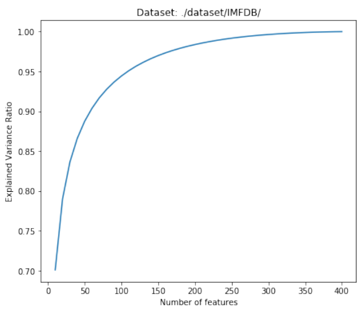
Zubair Abid 20171076

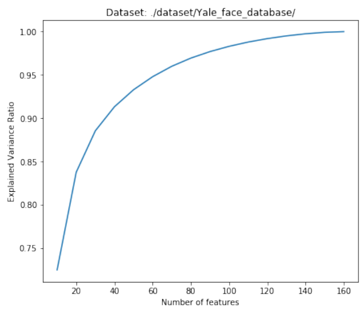
1 Basic Questions

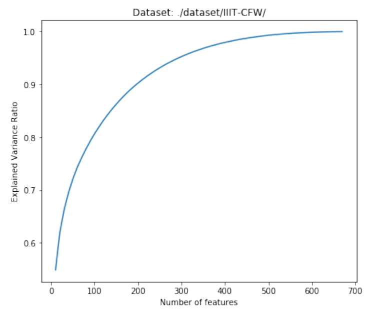
1. **What are eigen faces?**
   Eigenfaces is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition. It is essentially an eigendecomposition of faces to lower dimensions, using multiple methods such as basic PCA, LDA, Fisherface, etc.

2. **How many eigenvectors/faces are required to "satisfactorily" reconstruct a person in these three datasets?**
   We do not need too many features to reconstruct a person, per se. However, to be absolutely certain, it would be correct to claim that a representation that has about 95% of the features of the original image should be satisfactory to reconstruct a person in this three databases.

   Going with this logic, we attempt to calculate the number of features needed by a basic PCA eigenface representation to get this approximate 95%. IMFDB: 110 features; Yale face database: 70 features; IIIT CFW: 300 features







3. **Which person/identity is difficult to represent com-pactly with fewer eigen vectors? Why is that? Explain with your empirical observations and intuitive answers**
   Within each class, the hardest to represent compactly (by small margins) have been:

   (a) IMDB: 2

   (b) Yale: 0

   (c) IIIT-CFW: 3

2 Classification

Use any classifier(MLP, Logistic regression, SVM, Decision Trees) and find the classification accuracy. Which method works well? Do a comparitive study.
After running tests with all classifiers, and multiple dimensionality reduction tools, the best methods were evaluated.

Features	Red. Dims	Classif. Error	Accuracy	F1 Score
K-LDA Sigmoid + MLP	7	0.0500	0.9500	0.944892
K-LDA Polynomial + LR	7	0.0500	0.9500	0.944892
K-LDA Polynomial + MLP	7	0.0500	0.9500	0.944892
LDA/Fisherface + LR	7	0.0500	0.9500	0.944892
K-LDA RBF + LR	7	0.0500	0.9500	0.944892

Table 1: IMFDB Dataset, best results

Features	Red. Dims	Classif. Error	Accuracy	F1 Score
K-LDA RBF + MLP	14	0.000000	1.000000	1.000000
K-LDA Polynomial + SVC	14	0.000000	1.000000	1.000000
K-LDA Polynomial + MLP	14	0.000000	1.000000	1.000000
K-LDA RBF + LR	14	0.000000	1.000000	1.000000
K-LDA RBF + LR	7	0.0500	0.9500	0.944892

Table 2: Yale Face Dataset, best results

Features	Red. Dims	Classif. Error	Accuracy	F1 Score
Resnet + LR	2048	0.037037	0.962963	0.960506
K-LDA Polynomial + MLP	7	0.037037	0.962963	0.959481
K-LDA Polynomial + LR	7	0.037037	0.962963	0.959481
K-LDA RBF + LR	7	0.037037	0.962963	0.959481
K-LDA RBF + LR	7	0.0500	0.9500	0.944892

Table 3: IIIT CFW Dataset, best results

The confusion matrix of the best performing method is:

3 t-SNE

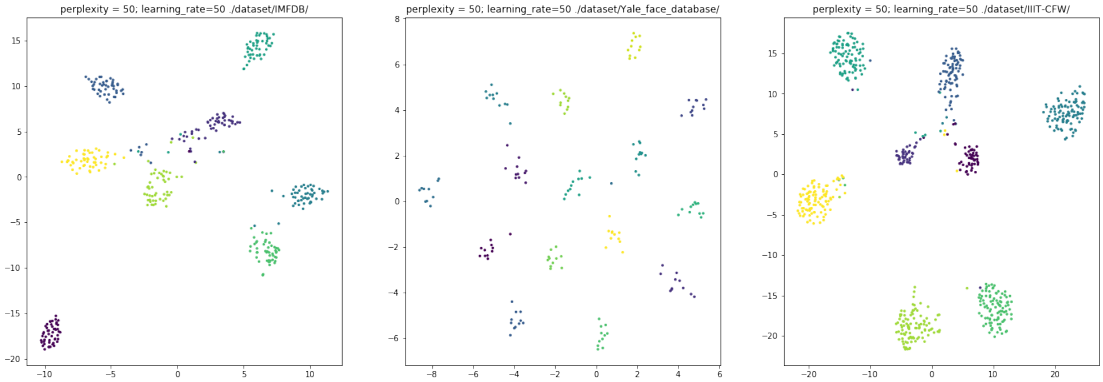
Similiar to 1(b) use t-SNE based visilization of faces? Does it makesense? Do you see similar people coming together?or something else? Can you do visualization dataset-wise and combined?
Due to the parameter tuning needed, I ran a script to calculate t-SNE across datasets to get a range of results.

First, I ran just t-SNE on the data provided. As we can see, there is not much conglomeration, similar people do not come together despite multiple tests.

We can conclude that TSNE does not make much sense in this context.

However, as an experiment, I then ran t-SNE on data that was reduced by LDA to 2 dimensions. Here, around perplexity = 30/50 onwards we see conglomeration, especially in the IIIT-CFW dataset. While full separation across all data is not achieved yet, the closest we can get by eyeball estimates are at perplexity = 50 and learning rate = 50.

Instead of LDA, I then decided to try using Resnet with the same parameters, and the results are promising. I have presented this as the solution.
We should note that despite this presentation, the clustering comes from the preceding feature extraction, and not due to TSNE.



4 face for verification

1. **How do we formulate the problem using KNN?**
   We model the problem in much the same way as in Question 2: use KNN for multiclass classification instead of binary. The key parameter here is to decide the value of k, which can be done by simple experimentation and observing performance.

2. **How do we analyze the performance ? suggest the metrics (like accuracy) that is appropriate for this task.**
   Analyzing the performance is a matter of simply splitting the dataset into train and test, training on the train set and then testing on test. We would look at accuracy: correct classification of the test data as our primary metric, but Precision is also a metric to keep track of.

3. **Show empirical results with all the representations**

Features	Red. Dims	Classif. Error	Accuracy	F1 Score
Kernel LDA + Sigmoid + KNN k=3	7	0.0500	0.9500	0.949800
Kernel LDA + Sigmoid + KNN k=5	7	0.0375	0.9625	0.959226
Kernel LDA + Sigmoid + KNN k=7	7	0.0375	0.9625	0.959226
Kernel LDA + Sigmoid + KNN k=9	7	0.0375	0.9625	0.960027
K-LDA RBF + LR	7	0.0500	0.9500	0.944892

Table 4: IMFDB Dataset, best results

Features	Red. Dims	Classif. Error	Accuracy	F1 Score
Kernel LDA + Sigmoid + KNN k=3	7	0.000000	1.000000	1.000000
Kernel LDA + Sigmoid + KNN k=5	7	0.030303	0.969697	0.980769
Kernel LDA + Sigmoid + KNN k=7	7	0.030303	0.969697	0.980769
Kernel LDA + Sigmoid + KNN k=9	7	0.030303	0.969697	0.980769

Table 5: Yale Dataset, best results

Features	Red. Dims	Classif. Error	Accuracy	F1 Score
Kernel LDA + Sigmoid + KNN k=3	7	0.029630	0.970370	0.976190
Kernel LDA + Sigmoid + KNN k=5	7	0.037037	0.962963	0.969190
Kernel LDA + Sigmoid + KNN k=7	7	0.037037	0.962963	0.970752
Kernel LDA + Sigmoid + KNN k=9	7	0.037037	0.962963	0.970752
K-LDA RBF + LR	7	0.0500	0.9500	0.944892

Table 6: IIIT-CFW Dataset, best results

5 Extension

5.1 The Problem

Distinguish cartoon faces from real faces.

5.2 Solution Utility

A use case for this solution would be to automatically filter out unnecessary information, say we need to annotate human face data but scraping leads to both human and cartoon faces being downloaded. In this scenario, we use the system to filter out the cartoon data first, and then annotate accordingly.

5.3 Experimental Pipeline

1. Creating Dataset: Take the IIIT-CFW Dataset and the IMFDB Dataset and combine the two. Create classification array accordingly based on previously available classification arrays for both datasets.

2. Classification: Create Sigmoid Kernel LDA features. Make train test splits with 20% test and use the MLP classifier with two hidden layers with 50 each. Train the classifier on the train sets and test on the test sets. For metrics, use classification_report provided by sklearn to get accuracy, precision, and F1 Scores.

3. Qualitative results: Plot out the data reduced by PCA, TSNE, and Isomap.