Project 2: Dimensionality Reduction and Manifold Learning

EEL5934: Applied Machine Learning Project 2

Nathan Oliver Noronha
Electrical and Computer Engineering
University of Florida
Gainesville, USA
n.noronha@ufl.edu

***Abstract*—This report is made for the purpose of Project 2 of the course EEL5934: Applied Machine learning systems. The data set used for the project are images of handwritten digits. According to the mentioned requirements, the appropriate are dimensionality techniques and estimators are used in the project and the corresponding decisions and explanations are given.**

I. Data Preprocessing

Each image in the dataset has a resolution of 300 x 300 which implies there are 90,000 features present in the dataset. The image size is resized to 50 x 50 to decrease the computation time which brings us down to 2500 pixels. These 2500 features are further reduced to lower dimensions by using appropriate techniques.

II. Dimensionality Reduction using Recursive Feature Elimination

Recursive Feature Elimination (RFE) was run on the dataset with estimators as Logistic Regression and Random Forest Classifier.

Method	Number of Feature selected	Accuracy
Random Forest	1000	100% (Overfitting)
Logistic Regression	700	61.01%

A. Selected Pixels and mask examples

The binary mask can be observed in 1 where it can be seen that the mask is concentrated at the center of the image, which implies that the most informative pixels are present in the middle which is true if we compare our image dataset. The borders are not considered since it almost no information.

RFE is a feature selection algorithm which iterates over the present features and then eliminates the least important features till the given number of features is reached. It is useful to reduce the dataset with a negligible change in performance but it can take a considerable amount of time for RFE to run select the desirable features.

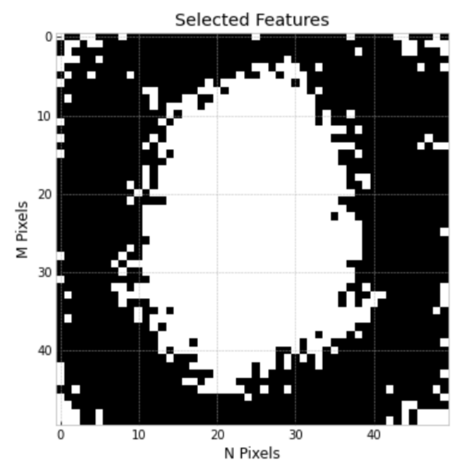

Fig. 1. Selected Pixels/ Binary Mask for RFE

We can also look at some of the mask examples from the training set in 2. The middle area is the mask area which is considered to be the most informative after performing RFE.

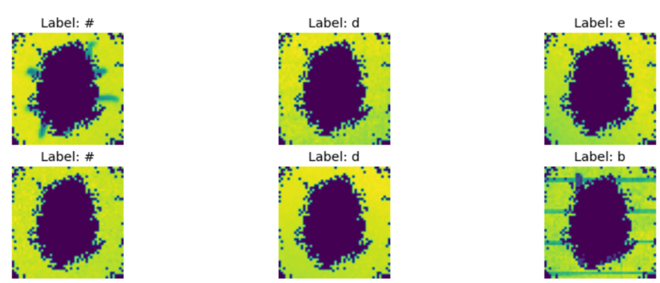

Fig. 2. Mask Examples from the training set

III. Dimensionality Reduction using Principal Component Analysis

The main goal of PCA is to transform the original features of a dataset into a new set of uncorrelated variables called principal components. These principal components capture the most variance across the data.

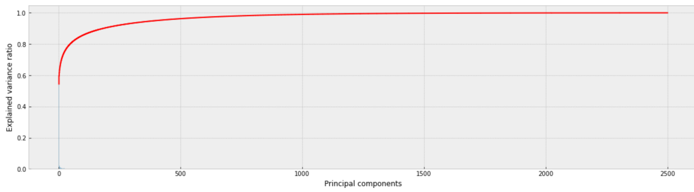

Fig. 3. Cumulative Explained Variance: It shows how the variance increases with increase in the Principal Components

Running PCA on the given dataset, I found the following cumulative explained variance. This plot suggests that 90% of the variance is present when you take 182 principal components which will not have a huge impact on performance if we only take 182 components. We can see the the cumulative variance increasing gradually with an increase in the Principal Components.

A. Using a Classifier with PCA

Further, a Logistic Regression classifier was trained on the reduced dataset and the original dataset. Naturally, training was faster on the reduced dataset as we are taking only 182 Principal components as opposed to the 2500 features in the original dataset. Below are the performances of both of them.

Method	**Accuracy**	**Confidence Interval**
Without PCA	35%	[0.750, 0.785]
With PCA	40%	[0.758, 0.791]

There is less accuracy without PCA even though it should have more; is because the training took more time, hence it reached the max iterations and could not totally fit the data. Hence, with PCA model was able to complete the training as it had only 182 features as compared to the 2500 features without PCA model.

B. Eigenvector Visualizations

Fig shows the top 10 visualized eigenvectors. Eigenvectors give us intuition on the direction of maximum variance of data and can help us to interpret the information in the features.

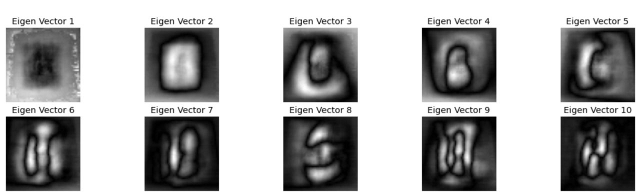

Fig. 4. Visualization of the top 10 Eigen Vectors

Among all the visualizations, the first eigen vector tells us that the most informative pixels are present in the middle area of the image. Other visualizations also tell us about the most prominent areas of the handwriting.

C. Image Reconstruction from PCA projections

Fig shows the original images and the reconstructed images from the training set.

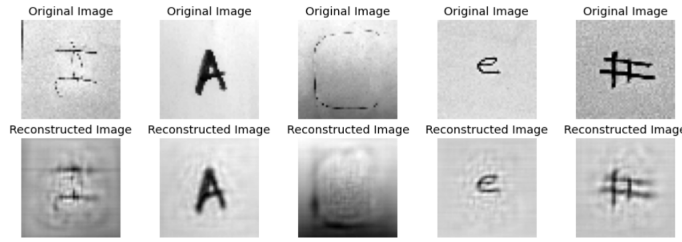

Fig. 5. Image Reconstruction from PCA projections

As seen from the Fig. the reconstructed image is blurry at the borders as those are the pixels which were found to be non informative by PCA. The pixels in the center are more informative hence they can be easily distinguished. Also, The quality of the reconstruction depends on the number of selected principal components. If we increase the number of principal components, the number of features will tend to the original number and hence we will approach the original image.

IV. Dimensionality Reduction using Fisher's Linear Discriminant Analysis and t-SNE

Fisher's LDA tries to maximize the between-class variance and minimize the in class variance while projecting the data onto a lower dimension. It can be seen in the Fig, since it is projected onto 2D, the data points overlap but it can be seen that the data points of a particular class are brought closer together.

A. 2D Dataset Visualization

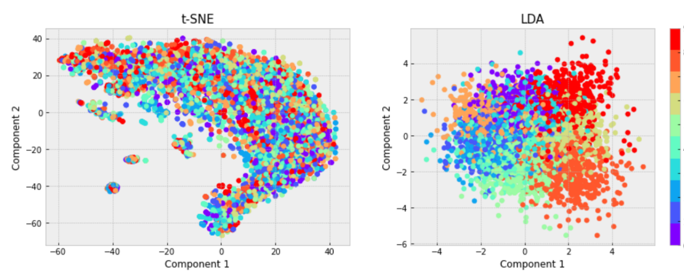

Fig. 6. 2D Visualizations of t-SNE and LDA

B. Selection of Number of Features

By running Grid Search on LDA, the best number of features to select for LDA which give a better performance are 8 features.

t-SNE is used for visualization and not for dimension reduction as it does not preserve distances, but it basically estimates probability distributions. The input space is assumed to be a Gaussian distribution and the map space a t-distribution. The heavy tails of the t- distribution are not able to preserve the local structure of the data at high dimensions. Hence, we can we can select 2 or 3 features to visualize in 2D or 3D.

C. Comparison with PCA

LDA does a better job of minimizing the within class variance such that the datapoints of a class are close together in the 2D visualization.

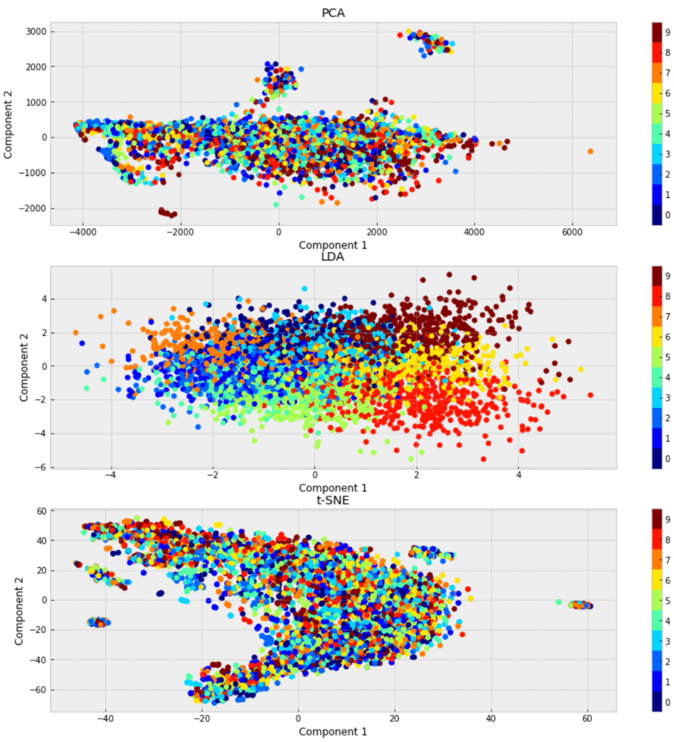

Fig. 7. 2D Visualization PCA

V. Dimensionality Reduction using Manifold Learning Algorithms

So far, we saw algorithms like PCA and LDA which are used for dimensionality reduction of linear datasets. Manifold Learning Algorithms are used for dimensionality reduction of non-linear datasets. Here we use three algorithms, namely:

- **Multi-Dimensional Scaling:** MDS focuses on preserving the pairwise distances or dissimilarities between data points rather than explicitly seeking linear combinations of variables. This makes MDS particularly useful when the relationships among data points are better captured by their relative positions rather than their absolute positions.
- **Isometric Mapping:** Isomap computes the geodesic distances, representing the true distances along the underlying manifold, rather than relying solely on the Euclidean distances in the original high-dimensional space, overcoming limitations of MDS.
- **Locally Linear Embedding:** LLE finds a low-dimensional representation of the data where the local relationships between neighboring points are maintained. It achieves this by first defining a neighborhood around each data point and then reconstructing that point as a linear combination of its neighbors.

A. Comparative analysis between implemented manifold algorithms

Multi-Dimensional Scaling: If using euclidean distance, MDS is a generalization of PCA as that will flatten the higher dimension data similar to PCA.

Isometric Mapping: Isomap is an extension of MDS with geodesic distances which is the shortest path between samples. It uses the Floyd Warshall's algorithm to calculate these distances.

Locally Linear Embedding: LLE is a good dimension technique when we want to preserve the local relationship of our non linear dataset.

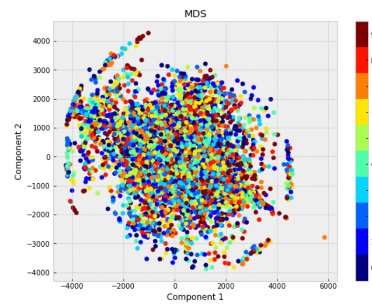

Fig. 8. Manifold 2D Visualization

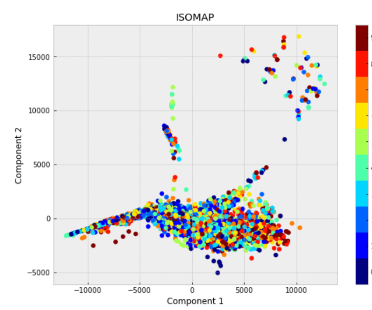

Fig. 9. ISOMAP 2D Visualization

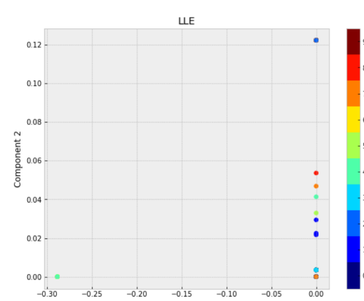

Fig. 10. LLE 2D Visualization

For our purpose, I would select LLE algorithm as we want to maintain the local structure of the data while projecting it to lower dimensions. Also, LLE does a good job for bring datapoints closer for ease of classification for an estimator.

B. Visualization and interpretation of the manifold learning algorithms in 2D

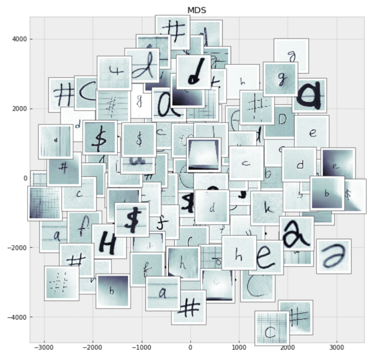

Fig. 11. 2D Visualization of MDS

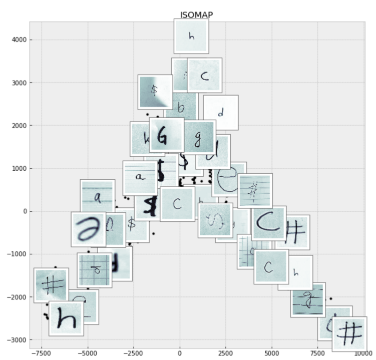

Fig. 12. 2D Visualization of ISOMAP

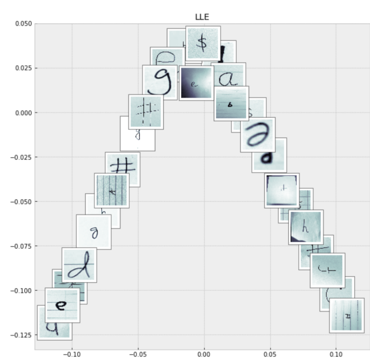

Fig. 13. 2D Visualization of LLE

For MDS the 2nd Component has the information for the size of the handwritten digit/letter. As you can see the letters at the bottom are tiny while the top of the Y-axis has letters which are huge. For ISOMAP, the 1st Component looks at the orientation and the stroke of the digit. For LLE, the 1st Component contains the information about the orientation of the digit while the 2nd Component looks for the stroke thickness and any loop present in that digit.

ACKNOWLEDGMENT

I would like to extend our sincere appreciation and gratitude to Dr Catia Silva and the Teaching Assistant's for their valuable support and feedback. I would also like to thank the department of Electrical and Computer Engineering at the University of Florida.

REFERENCES

[1] Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition (Aurélien Géron, 2019)

[2] Python Machine Learning - Third Edition (Sebastian Raschka , Vahid Mirjalili, 2019)

[3] https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/