# Numerical Optimizations and Linear Algebra

# Assignment 2: Singular Value Decomposition

## Introduction

In the current document the outcomes of 2nd Assignment are reported. The purpose of this Assignment is the creation of a classification algorithm and the use of it for classification of handwritten digits (0 – 9). Classification Algorithm presented is implemented in Python Programming Language. The Singular Value Decomposition (SVD) method that is used in the Assignment is imported from NumPy.

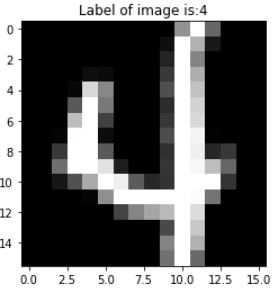
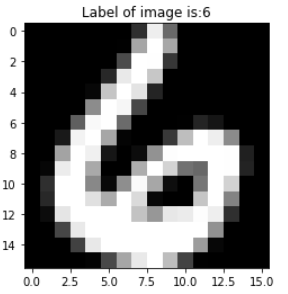
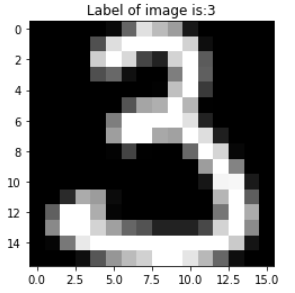
## Data

Data are imported from Excel file “data.xlsx”. The matrices that represent the images are stored as NumPy arrays, x\_train having 256 rows and 1707 columns and x\_test having 256 rows and 2007 columns. Each column represents a grayscale image of 256 pixels. The vectors that represent the class labels are stored as Pandas Data Frames, y\_train and y\_test, having 1707 and 2007 elements respectively.

Image vectors are normalized to having values in the range [-1,1] using SkLearn’s MinMaxScaler method. Note that this method treats each column as an independent feature.

A method for plotting vectors as images, called v2im, is implemented. This method reshapes a vector of dimensions (256,1) to a matrix of dimensions (16,16) and scales its values in the range [0,20], by subtracting the minimum value of the matrix from every element and then multiplying every element by 20/(maximum value of the matrix).

Below are plotted some images of the training set:

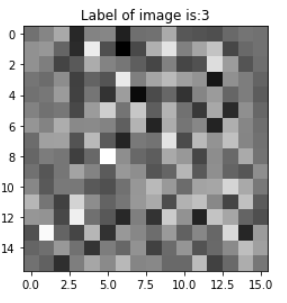
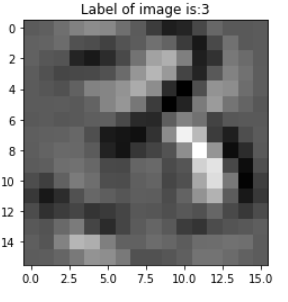
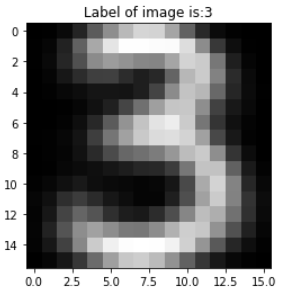


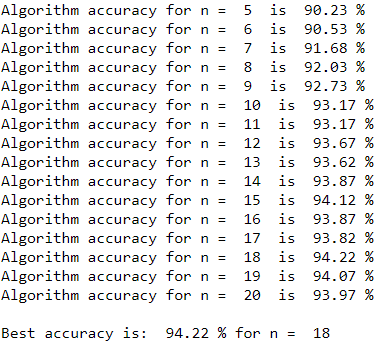
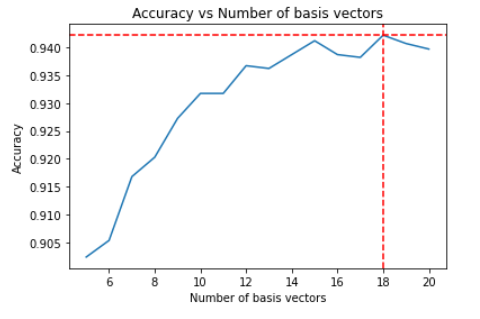
## SVD on classes

For the SVD step, 10 matrices are created using the training set, each one containing vectors (as columns) that represent images belonging to a particular class. Then, SVD is applied to the 10 matrices. After decomposition, each one of the matrices is represented by three matrices: . Matrix contains the left unitary (basis) vectors of each class, matrix the singular values of each class and matrix the right unitary (basis) vectors of each class. Matrices and are stored in lists and this concludes the training phase of the algorithm.

Below are plotted some basis vectors (1st, 11th and 101th) of class “3”. The first singular vector, as expected, looks like a 3. The following singular images represent the dominating variations of the training set around the first singular image.

## Classification

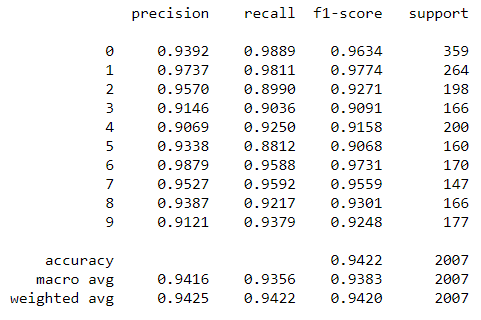


Classification is performed according to relative residual in least squares problem, calculated by the formula: , where is the vector that needs to be classified in a specific instance and a matrix consisting of basis vectors for class . More specifically, in order to classify a vector to one of the ten classes, ten residuals (one for every pair vector – class, for ten classes) are calculated using the formula given. Then the vector is classified to the class that gives the smallest value on the calculation motioned. Classification is performed for . Accuracy of classification for every value of is presented below:

Best accuracy is 94.22% and is accomplished when 18 basis vectors are used.

## Comparison Between Classes

Classification Report and Confusion Matrix are used in order to find out if all digits are equally easy classified. The metrics are calculated for 18 basis vectors, the value of parameter that gave the best accuracy score.

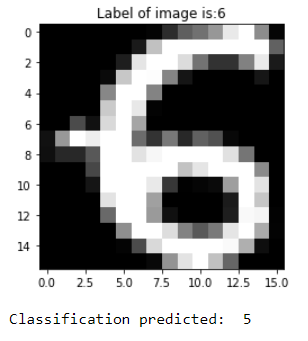
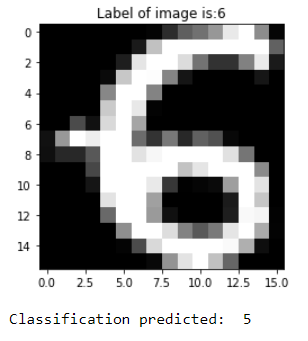
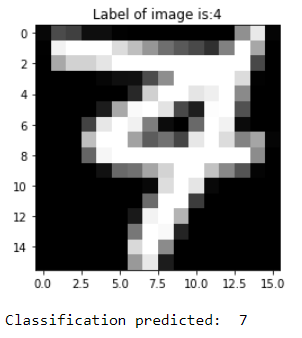
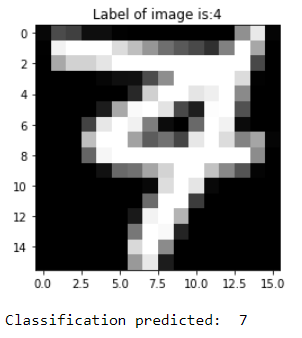
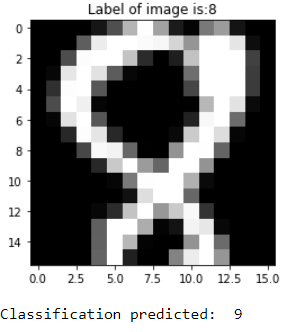
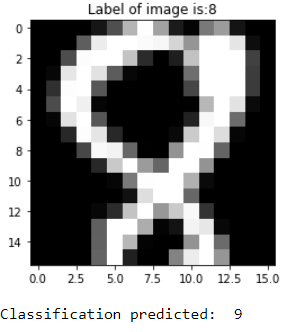
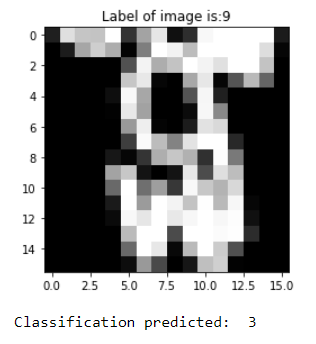
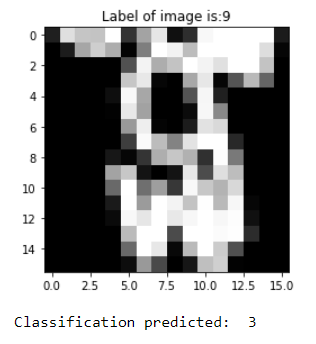


Observe that classes 3 and 5 have the lowest F1 scores (F1 score gives a weighed glance at precision ad recall), making them the hardest to classify and that class 1 has the highest, indicating that this is the easiest to classify.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |  |  |  |
| 0 | 355 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |  |  |  | Predicted Class | |
| 1 | 0 | 259 | 0 | 0 | 3 | 0 | 2 | 0 | 0 | 0 |  |  |  | True Class | |
| 2 | 8 | 1 | 178 | 2 | 5 | 0 | 0 | 1 | 3 | 0 |  |  |  | Intersection | |
| 3 | 2 | 0 | 3 | 150 | 1 | 6 | 0 | 1 | 2 | 1 |  |  |  |  |
| 4 | 2 | 1 | 0 | 0 | 185 | 2 | 0 | 3 | 0 | 7 |  |  |  |  |
| 5 | 7 | 1 | 1 | 5 | 0 | 141 | 0 | 0 | 2 | 3 |  |  |  |  |
| 6 | 2 | 1 | 0 | 0 | 2 | 1 | 163 | 0 | 1 | 0 |  |  |  |  |
| 7 | 0 | 1 | 1 | 0 | 3 | 0 | 0 | 141 | 0 | 1 |  |  |  |  |
| 8 | 2 | 0 | 1 | 6 | 0 | 1 | 0 | 0 | 153 | 3 |  |  |  |  |
| 9 | 0 | 2 | 0 | 1 | 4 | 0 | 0 | 2 | 2 | 166 |  |  |  |  |

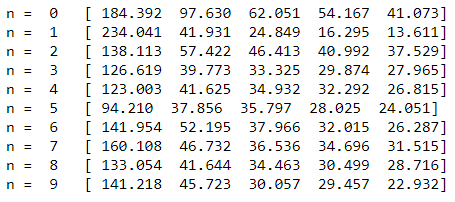
Also, looking at the Confusion matrix (yellow cells) it can be observed that the problem with these two classes is that digits that should be classified to 3 or 5 are put to other classes. Class 3 is frequently confused with class 5 and vise versa. Class 5 is confused with class 0 too.

Below are plotted some images of digits that were misclassified, their true class (above) and the class that the algorithm predicted (below). In many cases these digits are poorly written.



## Singular Values

Below are presented the five first singular values (matrix ) of each class.



Class 1 has a significantly larger first singular value compared to the other classes. It should be possible to use less basis vectors for this class, without it affecting the algorithm's accuracy. This means that the first basis vector of class 1 describes sufficiently the class. Class 1 has, also, the best scores as stated before.

The following experiments were performed:

|  |
| --- |
| Experiment 1 |
| Residual with class 1 is computed using 0 basis vectors – 18 for the rest  Accuracy: 81.56% |
| Experiment 2 |
| Residual with class 1 is computed using 1 basis vector – 18 for the rest  Accuracy: 92.18% |
| Experiment 3 |
| Residual with class 1 is computed using 2 basis vectors – 18 for the rest  Accuracy: 93.27% |
| Experiment 4 |
| Residual with class 1 is computed using 5 basis vectors – 18 for the rest  Accuracy: 93.72% |

Despite using such a small amount of basis vectors for class 1 the accuracy of the model is not affected that much.

## Two – Stage Algorithm with SVD (Optional)

In this section the use of a two – stage algorithm for classification is explored. In the first stage, residuals of vector to be classified with classes are computed using only one basis vector. The smallest residual is found and then the differences (as percentages) of the other residuals to this one are calculated. If every residual has a difference of {threshold}% or more to the smallest, then the digit is classified to the class with the smallest residual. The way to check that is, obviously, to check the two smallest residuals. If the threshold is not satisfied, then algorithm passes to second stage where classification is performed as explained to previous sections.

The following experiments were performed:

|  |
| --- |
| Experiment 1 - Threshold: 0.1% |
| Accuracy: 92.73%  Stage 2 was unnecessary: 63.13 % of times |
| Experiment 2 - Threshold: 0.2% |
| Accuracy: 94.02%  Stage 2 was unnecessary: 42.95 % of times |
| Experiment 3 - Threshold: 0.3% |
| Accuracy: 94.22%  Stage 2 was unnecessary: 30.44 % of times |
| Experiment 4 - Threshold: 0.1% |
| Accuracy: 94.22%  Stage 2 was unnecessary: 22.32 % of times |
| Experiment 5 - Threshold: 0.5% |
| Accuracy: 94.22%  Stage 2 was unnecessary: 16.79 % of times |

In the first 2 Experiments, while accuracy did not drop that much compared to the best achieved (94.22%), stage 2 was unnecessary for a large percentage of times, achieving savings in time and memory. For the rest of the Experiments the best accuracy is reached, but still, stage 2 remains unnecessary a lot of the times.