# Numerical Optimization and Linear Algebra

## Assignment 2

Mouselinos Spyridon – Part Time

## 1) Explanatory Analysis

In this exercise we are required to create a classification mechanism based on the Singular Value Decomposition method. The method is further investigated regarding the number and the importance of our singular values / basis vectors and finally applied to a dataset of grayscale handwritten digits.

Our data live in a csv file where each column represents the flattened 16 x 16 = 256 pixels of an image, resulting in 1707 training examples – images. The final training input matrix is thus of size 256 x 1707.

The same applies for the test data with 2007 examples and thus a matrix of size 256 x 2007. Finally, the correct labels (0-9) for each image are stored as a single column of the same (1707 – train / 2007 – test) size.

Before proceeding, we decide to pack our data into 4 different NumPy Matrices:

- X_train: Matrix of training images/examples.
- X_test: Matrix of test images/examples.
- Y_train: Array of Correct Training Labels.
- Y_test: Array of Correct Test Labels.

However, data are still yet to be normalized correctly in order to be further processed by our classifier. For this purpose we use the Sci-Kit Learn MinMaxScaler to enforce each Image pixels in the desirable range [-1,1].

Furthermore, as requested the method `scale_and_plot` is implemented that reshapes an image array of columnar shape 256 x 1 into a square matrix of dimensions 16x16 and rescales its values in the range [0,20].

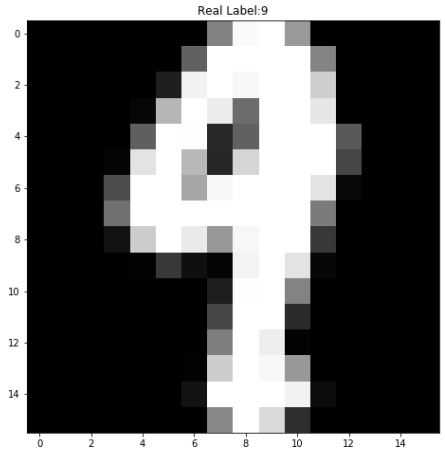
The method works in two steps:

- `Subtract the minimum value of the matrix in order to scale it to [0, max-min]`
- `Multiply by 20 and divide by max-min to rescale once again to [0,20]`

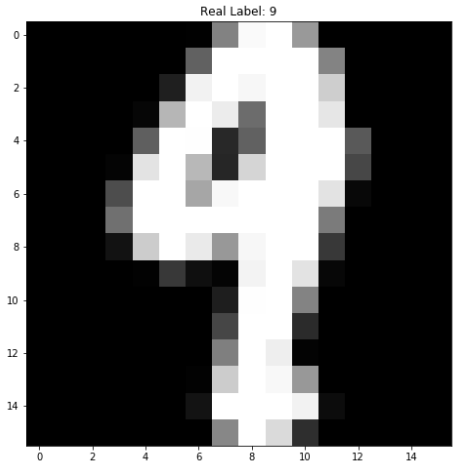
`Images Produced by plotting the train set and through scale_and_plot are presented below:`

Random Image Plotted with:

A) Imshow




B) Scale_and_plot




## 2) SVD Analysis

The next step was to perform SVD on each class separately and obtain the respective $U, S, V$ matrices. By grouping based on real label we created 10 trainset matrices that consisted of the vector images belonging to one of the (0-9) classes. After the application of the SVD decomposition we obtain the $U, S, V$ components., where:

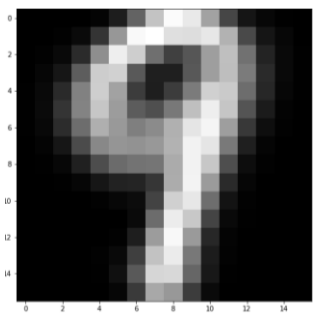
- $U$ is made of the left unitary vectors.
- $S$ contains the ordered singular values.
- $V$ is made of the right unitary vectors.

of each class.

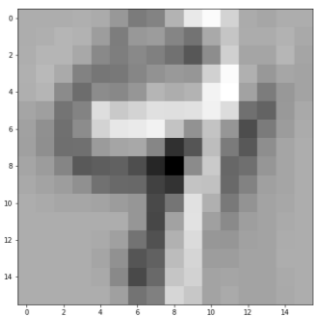
Each of the singular vectors "explain" some part of the original image information. The usage of all singular values will eventually use all singular vectors and reconstruct the original image as a whole.

In order to demonstrate this we will present the 1st the 5th and the 30th left unitary vectors of a random digit:

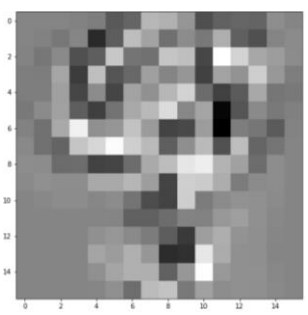
1st U Component



5th U Component



30th U Component



As we can see most of the information is stored in the 1st Unitary vector that seems a lot like a 9, some more high-frequency attributes are stored in the following vectors e.g in the 5th component, while at larger indices e.g in the 50th component the image vaguely looks like a 9.

## 3) Classification Analysis

In order to use the SVD analysis as a starting point in order to perform classification we must define a loss criterion. For this we decide to use the relative residual in least squares problem given by:

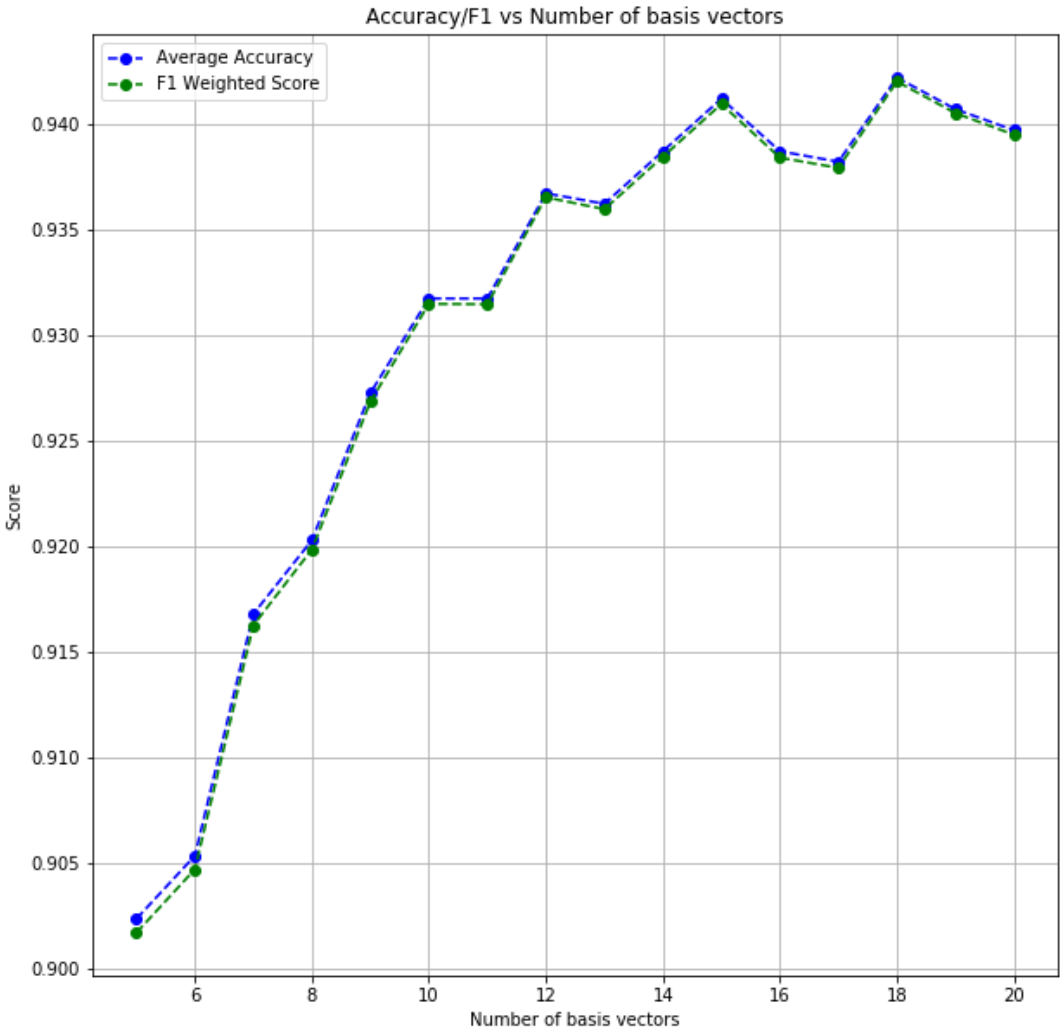
$$\frac{\left|\left(I - U_k U_k^T\right)z\right|_2}{|z|_2}$$

Where:

- $z$ is the image vector to be classified, flattened to 1D.
- $U_k$ the n-most significant left unitary vectors of a class (0-9), packed into a matrix.

After calculating the residuals for each class (0-9), our classifier predicts the most prominent one by choosing the class with the lowest residual. This means that the input image contains information that can be reconstructed by using the first n $U_{predictedclass}$ vectors, better that the other classes U matrices.

As requested a grid-search was performed for value n, using 5 to 20 U vectors. The results are shown below in terms of both accuracy and f1 score:

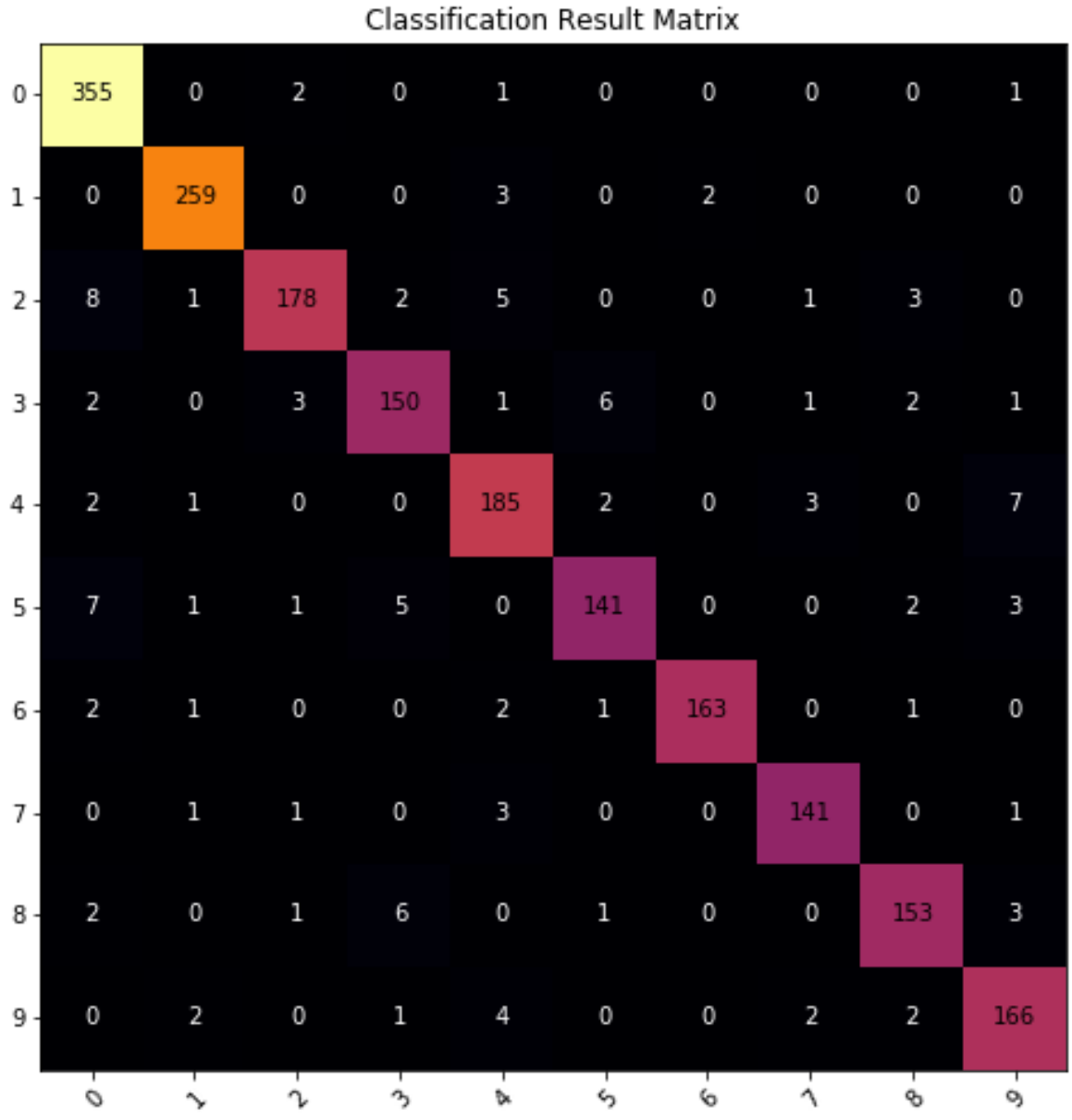



As we can see the best score is achieved at n = 18 basis vectors, with score 94.22% in terms of accuracy and 94.20% in terms of f1 score.

## 4) Classification Results / Misclassification Analysis

After finalizing on the usage of 18 basis vectors we decide to evaluate our algorithm of the test set and present the results as well as any striking misclassification evidence.

The confusion matrix of our predictions is the following:



Here the rows correspond to the True Classes while the columns correspond to the predicted classes. For example, for the digit 0 we see that we have:

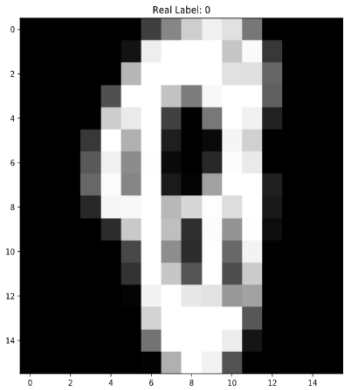
- 355 predictions for 0.
- 2 predictions for 2.
- 1 prediction for 4.
- 1 prediction for 9.

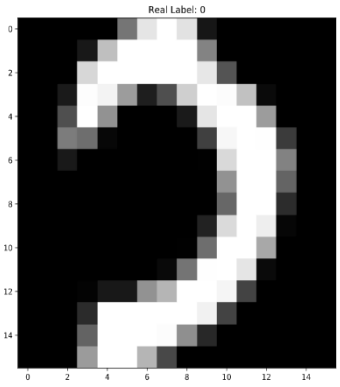
The total accuracy is calculated as the sum of the trail of the matrix divided by the number of the test examples. The F1 score describes the tradeoff between the precision and the recall of each classifier in one-vs-all fashion and we report the weighted variant of it that takes into the account the class priors in order to give results closer to reality.

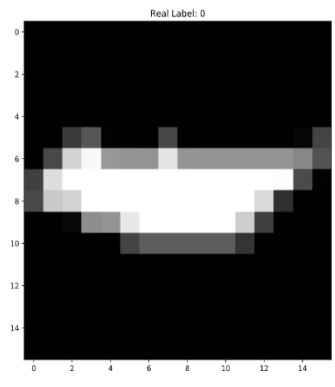
Let's have a closer look:

- Digit 0 has 4 misclassifications resulting in 1.114% of misclassification rate.
- Digit 1 has 5 misclassifications resulting in 1.894% of misclassification rate.
- Digit 2 has 20 misclassifications resulting in 10.101% of misclassification rate.
- Digit 3 has 16 misclassifications resulting in 9.639% of misclassification rate.
- Digit 4 has 15 misclassifications resulting in 7.5% of misclassification rate.
- Digit 5 has 19 misclassifications resulting in 11.875% of misclassification rate.
- Digit 6 has 7 misclassifications resulting in 4.118% of misclassification rate.
- Digit 7 has 6 misclassifications resulting in 4.082% of misclassification rate.
- Digit 8 has 13 misclassifications resulting in 7.831% of misclassification rate.
- Digit 9 has 11 misclassifications resulting in 6.215% of misclassification rate.

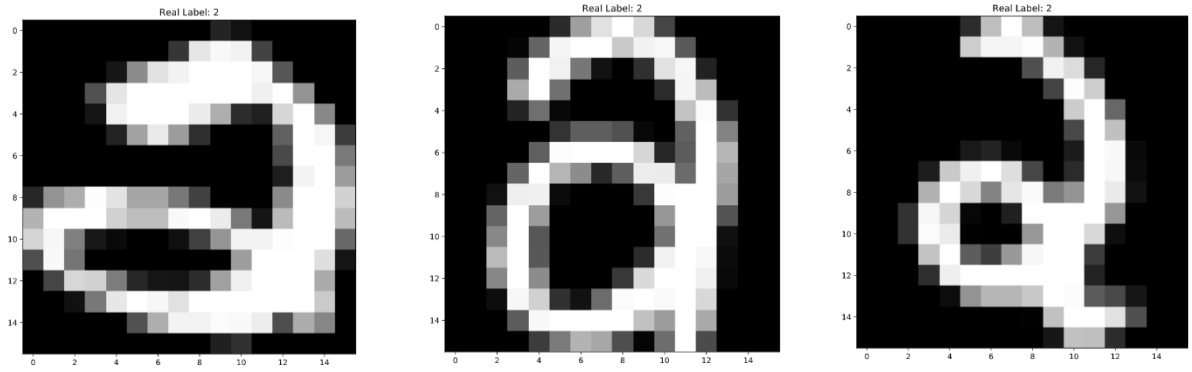
Here are 3 images misclassified with original label 0:



Here are 3 images misclassified with original label 2:

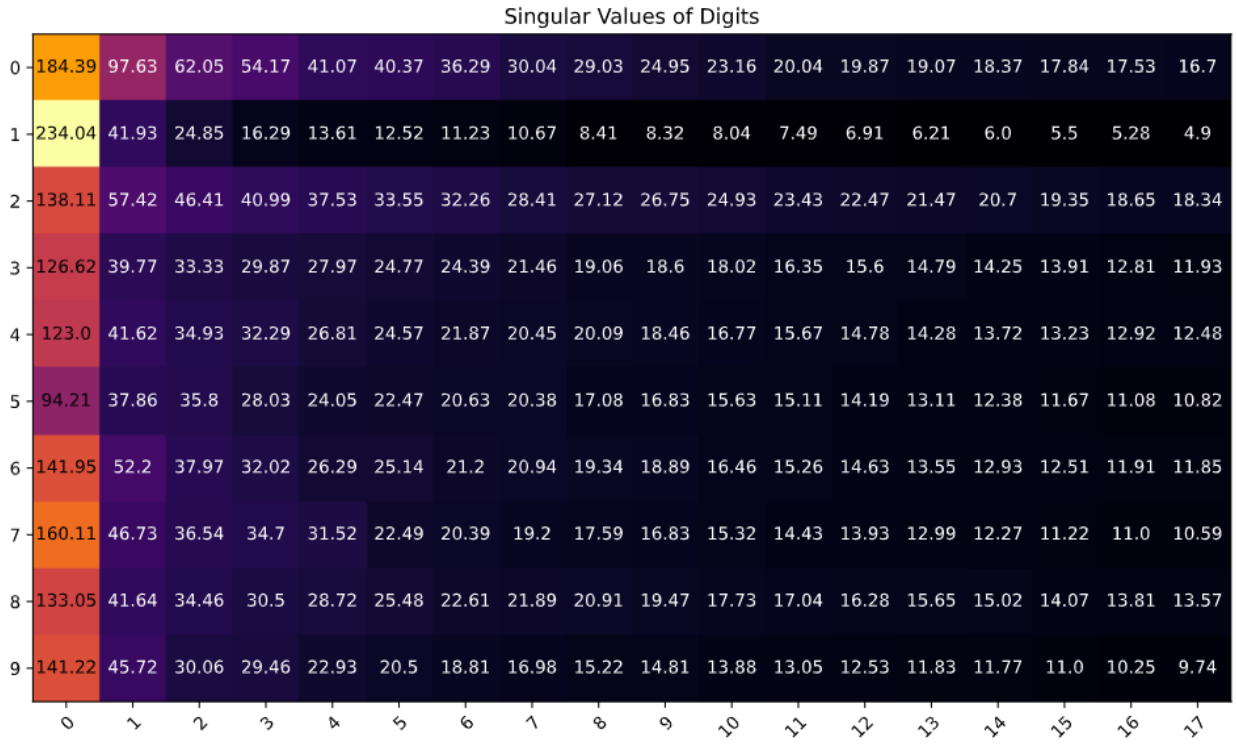



As we can see in this cases our classifier is faced with very badly written images that in the case of 0 tend to look like 8's / 2's or dots and in the case of 2's with inverse 6's or letter a's.

Generally speaking the most misclassified digits seem to be digits 2 and 5 with digit 2 being mistaken for 5 and digit 5 being mistaken for 0, as seen in the confusion matrix above.

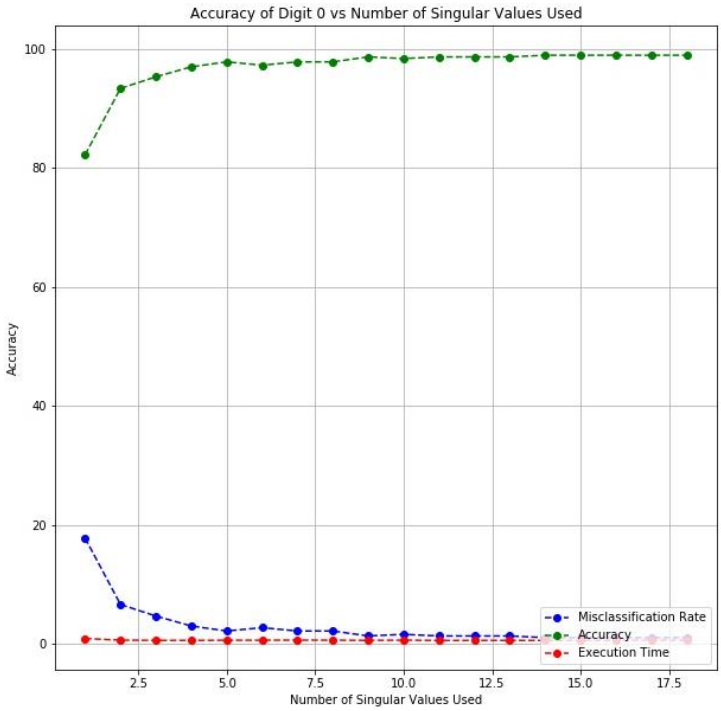
## 5) On the Subject of Singular Values

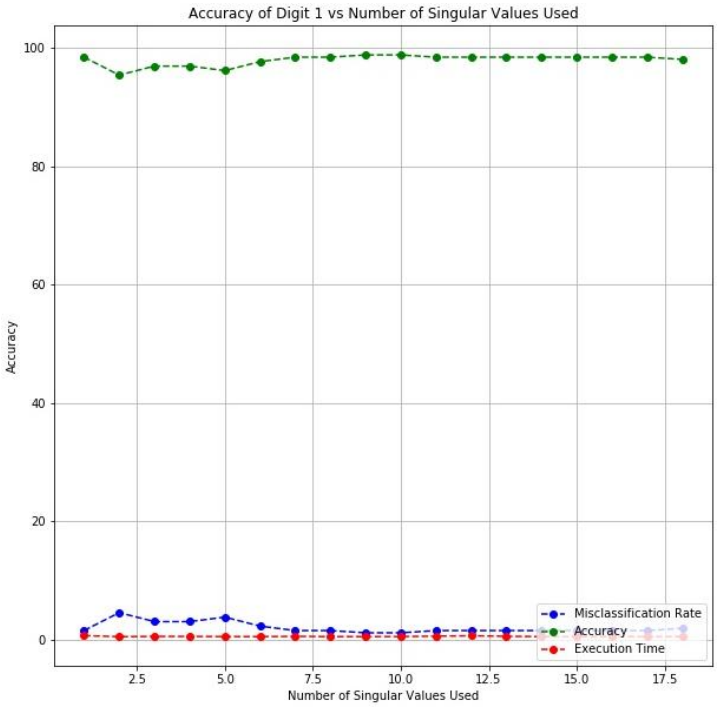
Let's have a better look on the first 18 singular values of each class:




As we can see in the case of classes 0,1 their first singular values are far greater than the rest one meaning the most information can be reconstructed using only the first basis vector. We can take advantage of this phenomenon and create a tradeoff between the algorithm's speed (fewer Flops in case of less singular vectors used) vs the overall accuracy of the model.

For those two digits, a grid search was performed to show the actual effect of this tradeoff




As we can see above for digit 0 the use of 3 singular values is close enough to the full method in terms of accuracy, and the use of 9 or more singular values seems to have miniscule effect.

A similar behavior can be also seen in digit 1 where the use of a single singular value is enough in terms of accuracy and the fastest method of all. Here it seems to be the best tradeoff choice.

## 6) Optional Question 1. Two-Stage SVD Algorithm

In this part of the exercise we will make use of the findings of the previous question, that not all the singular values/basis vectors are always required to correctly classify an input image. Here we will implement a two-stage algorithm that:

- Stage 1: Makes use of a single singular value and implements the SVD classifier up to the stage of calculating the relative residuals. Here a threshold value is implemented. Now we focus on:

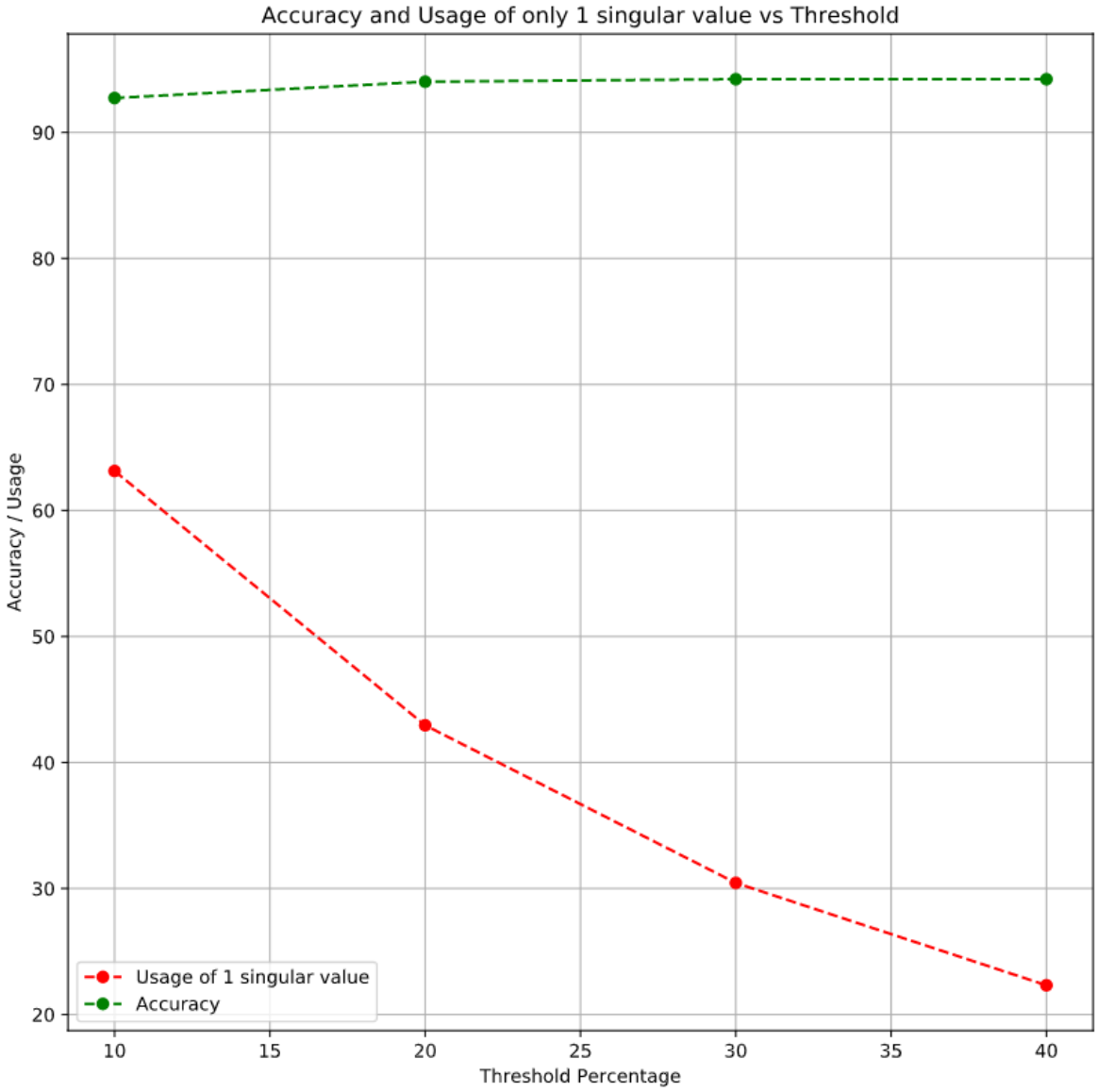
$$Residual_{smallest} - Residual_{nextsmallest} / Residual_{smallest}$$

  the relative magnitude of the first smallest to the second smallest residual. If this value is larger than the threshold, we have enough evidence to classify the image to the according class. If this value is below the threshold, we have little evidence to classify the image to the digit with the smallest residual. Then we move to stage 2.

- Stage 2: Makes use of the optimal eighteen singular values and implements the SVD classification fully, up to the point of decision.

Below we present the results of a grid search that uses different threshold values and plots:

- The Overall Accuracy achieved.
- The number of the times the times the Stage 1 was enough.




We can see here that at threshold equal to 0.1 or 10%, the full SVD method was required less than 40% of the images with no significant drop in the accuracy.

At 0.2 or 20% we can observe an optimum where the full SVD method is required more than 55% of the time while the accuracy is near the beast achieved value.

For threshold values over than 0.2 we don't observe any noticeable gain in terms of accuracy, while the use of the second stage is progressively becoming more often.

Closing, we can get the following remarks: That not all singular values / vectors are always required to correctly classify, as well as that we can adopt a dynamic policy as above to have a tradeoff between the accuracy and complexity of the models.