Alma Mater Studiorum · University of Bologna

Statistical and Mathematical Methods for Artificial Intelligence

Classification of MNIST Digits with SVD Decomposition.

Lorenzo Cellini (lorenzo.cellini3@studio.unibo.it)

February 5, 2022

Contents

1 Summary

Object classification is one of the most common task that artificial intelligence systems should be able to handle, at least with some degree of accuracy. In this work the digit classification task over the MNIST dataset has been addressed. The MNIST dataset consists of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. Although the dataset size counts 70000 examples, this work focuses on a small subset (given by teachers) of 1707 examples, where each one is a 16×16 grayscale image.

The classification task is carried out by leveraging the SVD decomposition of a digit's space, projecting an unknown sample into that digit's space and computing the distance with respect to its known class. The digit's space that minimizes that distance is output as the predicted class. In general this is a multiclass (or multinomial) classification task, where an unknown sample has to be classified as one among several (ten in this case) different classes. However, in the first exercise the problem has been restricted to be a binary classification task, while in the second exercise it has been generalized to the multinomial classification.

2 Background theory

In statistics, classification is the problem of identifying which, of a set of categories an observation belongs to. Classification can be thought of as two separate problems: binary classification and multiclass classification. In binary classification, only two classes are involved, whereas multiclass classification involves assigning an object to one of several classes. Since many classification methods have been developed specifically for binary classification, multiclass classification often requires the combined use of multiple binary classifiers. The algorithm that implements classification is called "classifier".

Usually, in machine learning, it is a good practice to split the dataset into two (or three) statistically balanced partitions. One is used to train the classifier, while the other one (or two) is used to test (or finetune and test) its performance on new samples. The same approach has been taken here.

In this work, the classifier is a function $f(\vec{x}, U_1, ..., U_n)$ that takes as input the unknown example $\vec{x}$ and the vector space representation of available classes $U_1, ..., U_n$, and outputs the predicted class.

The vector space representation of a class is given by the SVD decomposition of the matrix which has as columns all the training dataset vectors, associated with the same digit, and as rows the 256 pixels that make up the digit image (by taking the 16×16 image and reshaping it as a 256×1 vector). In particular, denoting as $X_i \in \mathbb{R}^{n \times m}$ the matrix given by all the vector associated with the digit i, its SVD decomposition is

$$X_i = U_i \Sigma_i V_i^T \tag{2.1}$$

By construction, the columns of U, whose same-numbered elements in Σ are non-zero, are an orthonormal set of basis vectors that span the column space of X. This can be proven by showing that $Im(X) = span(u_1, ..., u_r)$, where r is the index of the last non-zero element of Σ and $u_1, ..., u_r$ are ONB of $\mathbb{R}^n$ because they are eigenvectors of XX^T.

Columns of U_i can be thought of as a set of images that, when linearly combined, give an instance of the digit i. A sort of a set of features that describe the digit i.

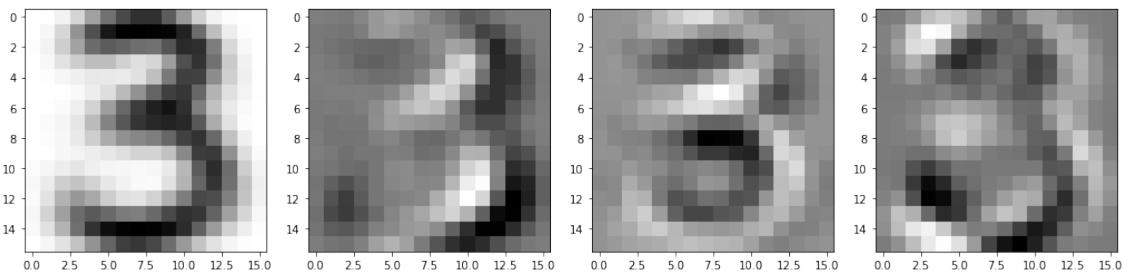

Figure 2.1: Examples of the first 4 columns of U_3.

In order to classify unknown samples, a notion of distance, between that sample and the vector space representation of digits is needed. So, when receiving the unknown sample, the algorithm porjects it onto the spaces of digits and compute the distance between the original sample and its projections. At the end, the class with the smaller distance, with respect to the sample, is chosen as the predicted one.

Sample projection would be computed as $\vec{y}^{||} = U(U^T U)^{-1} U^T \vec{y}$, but because the $(u_1, ..., u_r)$ are ONB it simplifies to

$$\vec{y}_i^{||} = U_i U_i^T \vec{y} \tag{2.2}$$

At the end, the distance $d_i = ||\vec{y} - \vec{y}_i^{||}||$ is computed for each class i and the chosen one is

$$pred\_class = \underset{i}{\operatorname{argmin}}(d_1, ...d_n) \tag{2.3}$$

All this reasoning applies to both the binary and multiclass classification task.

3 Exercise 1 - Binary classification

This exercise consists of a binary classification task, at first on digits 3 and 4, and then experimenting with different digit is required. 3.1 shows instances of digits 3 and 4 in the dataset.

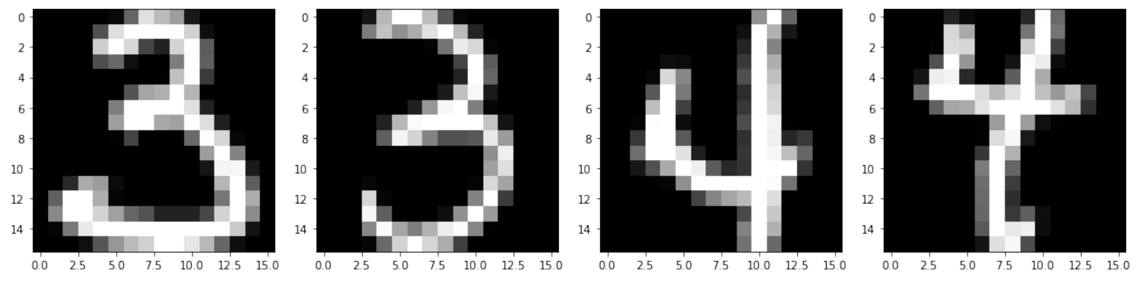

Figure 3.1: Instances of digits 3 and 4 in the dataset.

The procedure is:

- load the dataset and the true class labels
- split them into training and test set (to do this I have used the function *train_test_split* from *sci-kit learn* with a 80% training set)
- extract the training and test partitions relative to digits 3 and 4
- compute the SVD decomposition for each digit $X_i = U_i \Sigma_i V_i^T$, where $i = 3, 4$
- for each unknown sample from the test set compute its projection onto the spaces spanned by the relative columns of U_3 and U_4
- compute the distances between the projections and the original vector and choose as predicted class the one that minimize that distance

The evaluation metrics for this classifier are the misclassification error and the confusion matrix. As shown below, this classifier has a 0% misclassified samples in case of digits 3 and 4.

Other experiments have been run over different combination of digits in order to investigate if visual similarity affects classification. In particular, pairs tested are (3,8), (3,5), (6,9), (1,7). Figure 3.3 shows the results, while comments are reported in 5.

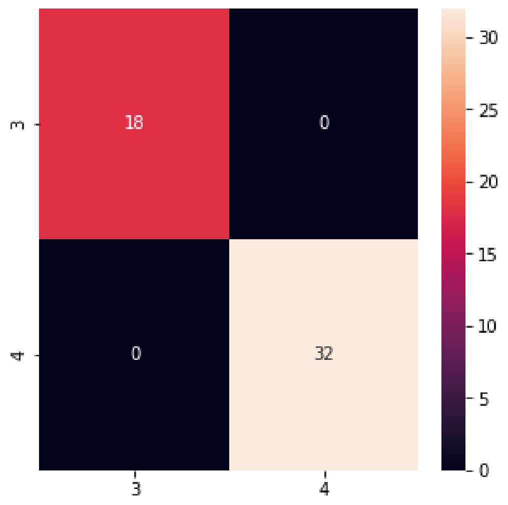

Figure 3.2: Confusion matrix for digits 3 and 4.

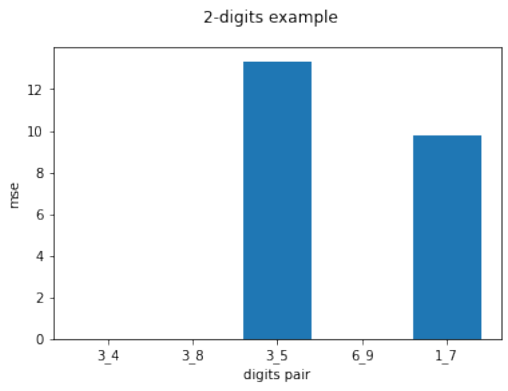

Figure 3.3: Misclassification error.

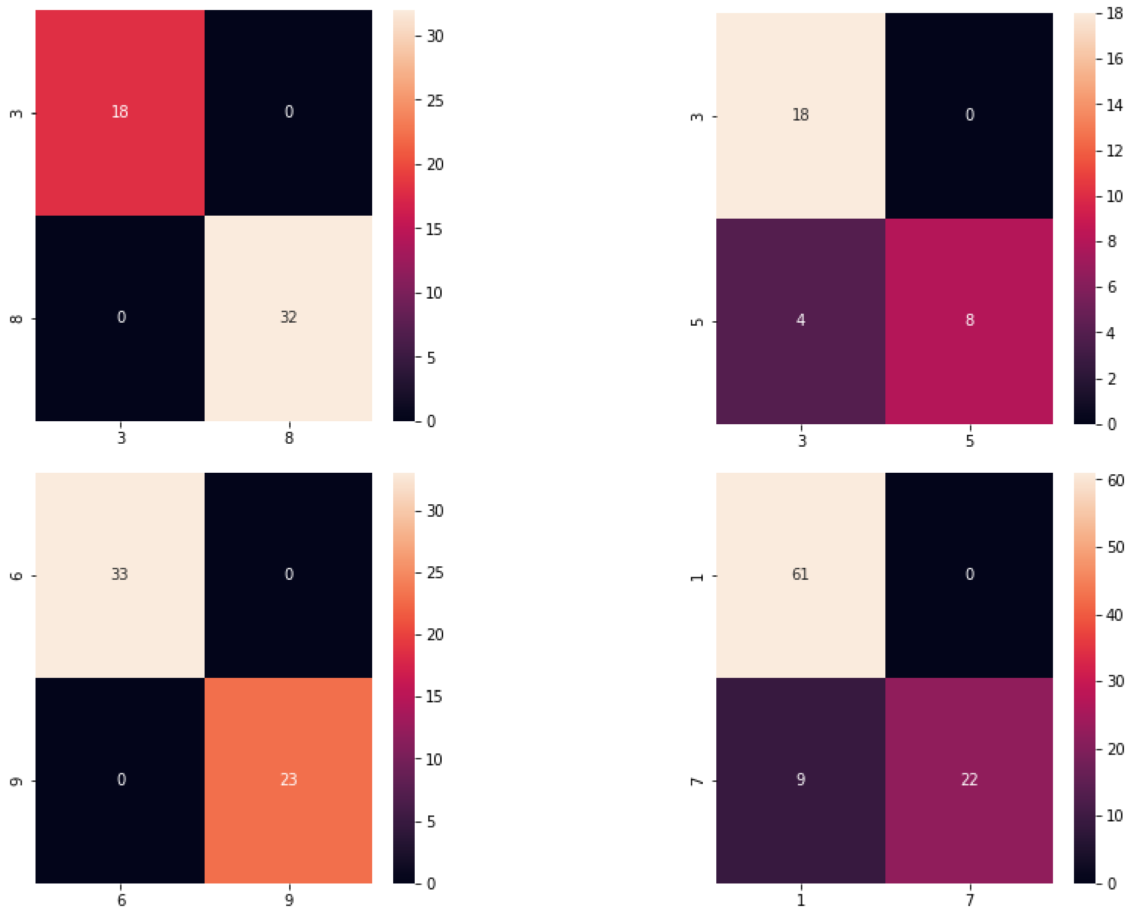

Figure 3.4: Confusion matrices for different experiments.

4 Exercise 2 - Multiclass Classification

In the second exercise the problem of multinomial classification is addressed by generalizing the binary classifier of the previous exercise to the case of k different classes. The algorithm is exactly the same as before. Here I followed the idea to take the pairs tested in the previous exercise, add a new digit and observe how the distribution of predictions change relatively to the visual similarity.

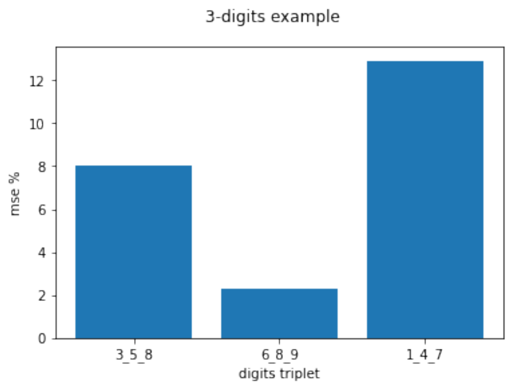

Figure 4.1: Misclassification error.

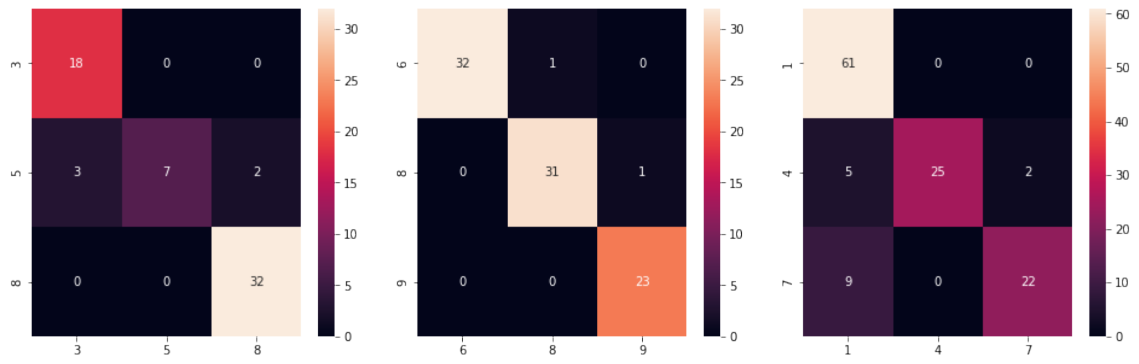

Figure 4.2: Confusion matrices for different experiments.

5 Conclusion

It is clear that a classifier based on SVD decomposition technique can perform very well on an "easy" task like digit recognition.

What is interesting is that visual similarity affects classification but only when also orientation is taken into account, at least for the 2- and 3-digits case: indeed, digits like 6 and 9, that from a human point of view could be judged "similar", differing only for a rotation factor, are well classified. Instead, digits like 1, 4 and 7, similar and with the same orientation, show a small, but non zero, classification error.

I tried to see if adding the digit 4 to the pair (1,7) could move some wrong prediction from 1 to 4 but it didn't happened. My idea is that the more the complexity of the digit, in terms of strokes and angles, the more the chance to fail the classification: indeed, the classification errors for the digit 7 are all in favour of the digit 1 (less complex) and never in favour of digit 4 (more complex). For the same reason, error classification for digit 4 are spread between digit 1 and 7.

Just for curiosity, I have implemented a general version of the classifier independent on the number of input digits. Results in 5.1 and 5.2 show that when all the digit, from 0 to 9, are included, the misclassification error reaches 60% and the vast majority of errors is in favour of 0 digit. I think this is due to the variance of shape when hand-writing digit 0: it is easy to draw shapes that share traits with other hand-written digits (example in 5.3). Indeed, when dropping digit 0 the misclassification error falls from 60% to around 24% and prediction are more spread in the confusion matrix.

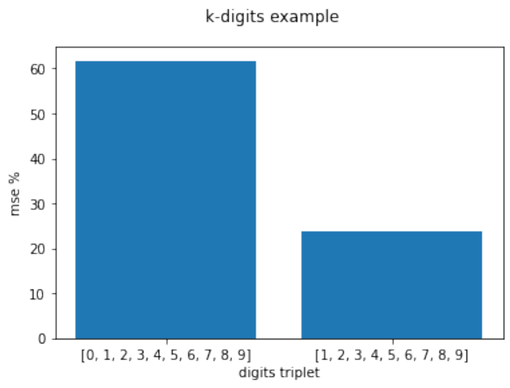

Figure 5.1: Misclassification error.

To conclude, SVD decomposition technique seems very powerful but also very sensitive to visual similarity. Neural network technique, although way more complex, have been proved to be very effective on this task reaching accuracies around 99%.

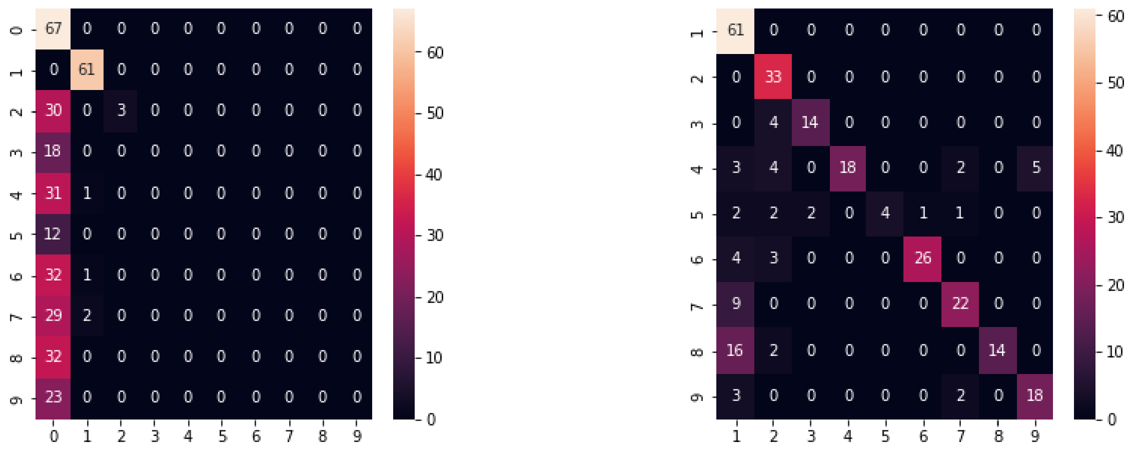

Figure 5.2: Confusion matrices for k-classifier with and without digit 0.

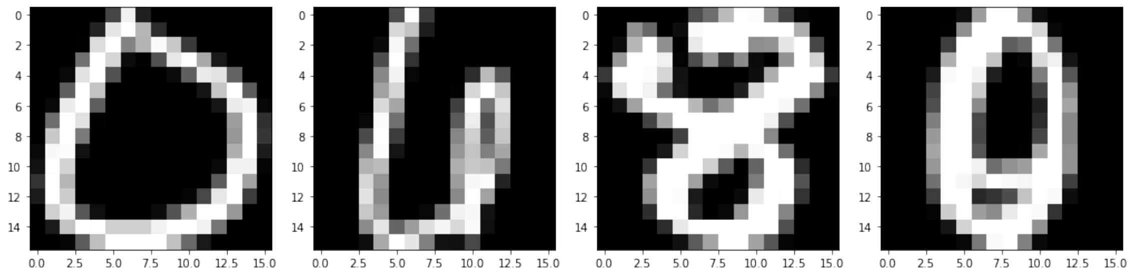

Figure 5.3: Visual similarity between different digits.