

Designing a classifier for handwritten digits recognition

Nicolas Brissonneau , UTEID:nb24488 , email:nicolasb@utexas.edu

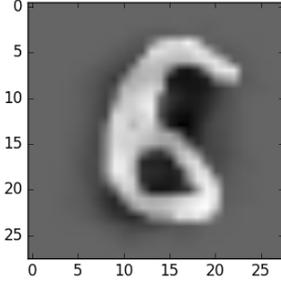


Fig. 1. handwritten digit

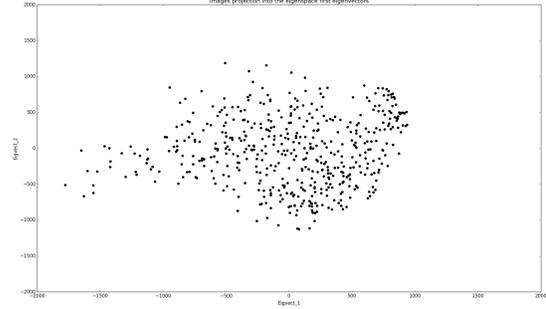


Fig. 2. Projection

I. INTRODUCTION

I am considering the problem of identifying a "testing sample" S_{test} composed of n_{test} handwritten digits similar to Fig.1, each originally encoded as a 28x28 gray-scaled picture (where brightness corresponds to the pixel value) which will be named V_{im} and of dimensions 784x1, we will denote $V_{im}(j)$ the j-indexed image from the samples list.

II. METHOD

In order to differentiate digit $V_{im}(j)$ from digit $V_{im}(k)$ ($j \neq k$) I am using the Principal Component Analysis (PCA) approach which relies on the derivation of the direction principal variations in the space of digits from a training set S_{train} . We will use these directions to allocate coordinates to the handwritten digit brightness patterns which will ultimately allow us to discriminate different samples based on a quantification of the similarities between them. Using a known set of labels from the training samples, we aim to design an appropriate metrics for quantifying the similarities between $V_{im}(j)$ and $V_{im}(k)$, as well as a suitable choice of eigenvectors to better discriminate between digits. The combination of these two goals would more efficiently predict the testing set's digits class (from 0 to 9).

A. Computing the direction principal variations

We first need to get rid of the bias among the digits from the training set S_{train} and thus introduce $W_{im}(j)$:

$$V_{average} = \frac{\sum_{j=1}^M V_{im}(j)}{M} \quad (1)$$

$$W_{im}(j) = V_{im}(j) - V_{average} \quad (2)$$

We can now consider the concatenated matrix A of $W_{im}(j)$ for $j \in [1, n_{train}]$ to get the sample covariance matrix Σ which leads to the digits eigenspace:

$$\Sigma = AA^T \quad (3)$$

However, we notice that depending on the resolution of the data we can face a high computing cost, so we also consider another approach to find the digits eigenspace, we will instead use Σ' to identify the set of eigenvectors v of the smaller dimension system:

$$\Sigma' = A^T A \quad (4)$$

$$A^T A v = \mu v \quad (5)$$

And by conveniently pre-multiplying by A we have the long set of eigenvectors U of Σ :

$$AA^T A v = \mu A v \Rightarrow \Sigma U = \mu U \quad (6)$$

We can now use U to project the image $W_{im}(j)$ into the eigenspace coordinates to compare it with another image $W_{im}(k)$.

B. k-nearest neighbours

In order to quantify the similarities between different images projections into the eigenspace, we can look at their euclidean distance in that space. The k-nearest neighbours algorithm suggests that if an unknown digit's projected image is close enough to numerous other images projections belonging to a common class (meaning that the other images' label are similar), the unknown digit might share the same class. This methods thus looks at the k closest projections and associates the unknown label to the most represented one among the k closest projections.

One can have a better intuition regarding this algorithm by considering the distance between the projections of images on only two eigenvectors as in Fig.2.

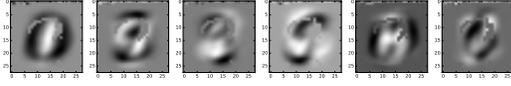


Fig. 3. Eigenvectors

III. RESULTS

For each of my experiments and data shown in this section, I have selected a random set of images from S_{train} , and when accuracy is shown in a graph it means that I have also selected a random set of images from S_{test} to test the classifier upon. In order to make sure I was not lucky/unlucky in my data, for each data point shown in a graph corresponding to a very specific choice of parameters, I have run several iterations with random selection of images each time and I have kept the mean performance.

A. Eigenspace representation

We can see the eigenvectors representation in Fig.3, these are normalized and can be understood as the axis of a coordinate system.

B. Image reconstruction

Fig.8 shows how the images are represented through their reconstruction using 3 different eigenvector sizes. One can notice that the number m of eigenvectors has an impact on the reconstruction.

C. Influence of the number of eigenvectors

Fig.4 shows how the accuracy of the digits labels classification increases as m increases, until it seems to remain constant. We learn that the minimum value of m is important, but it does not require to increase much as it does not significantly impact the accuracy afterward.

D. Influence of the number of images for training

Fig.5 shows how the accuracy of the digits labels classification varies over the number of training images. The evolution is similar to subsection C's, even though we still observe a noticeable growth as the training set's size increases.

E. Influence of the number of neighbours to consider

Fig.6 shows how the accuracy of the digits labels classification varies over the number of neighbours counted. We will discuss it more in the conclusion.

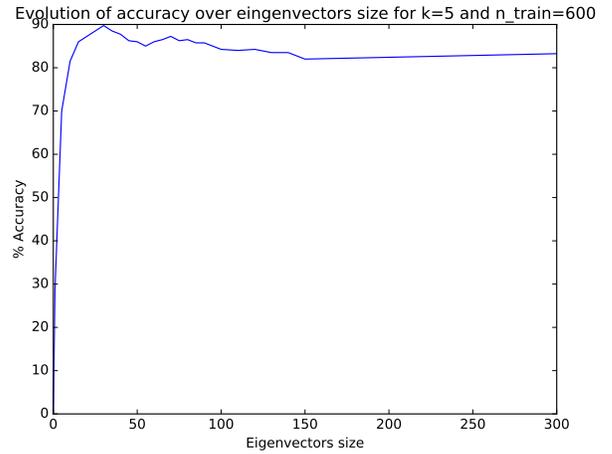


Fig. 4. Influence of m

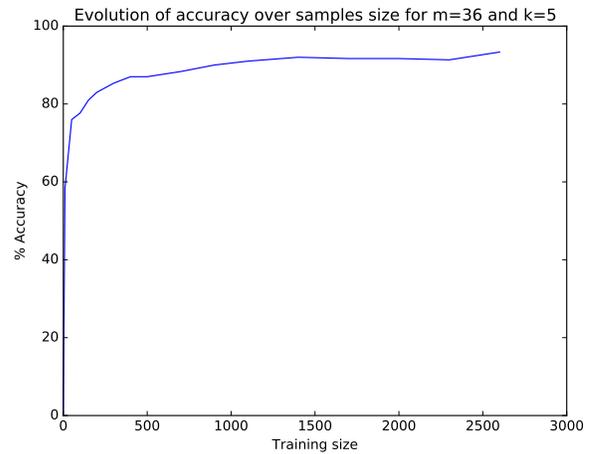


Fig. 5. Influence of sample size

IV. CONCLUSION

The data indicate that the choice of k and m plays a big part in improving the accuracy of the classifier and this is rather intuitive, however I was surprised to see how easily the k parameter could decrease the accuracy when being overestimated. This parameter's sensitivity can partially be solved by counting the closest neighbours while weighting them inversely proportionally to their distance with the projection of the image to classify. I chose to weight them as $\frac{1}{D^4}$ where D is the euclidean distance mentioned in section 2. The exponent 4 seems to decrease the loss of accuracy over k as seen in Fig.7. compared to Fig.6.

I believe a next step in improving the efficiency of the classifier would be to better choose the eigenvectors in the first place. As we proceeded, we selected the m highest eigenvalues and kept the associated eigenvectors, while we could have selected the eigenvectors who had the biggest impact in discriminating the digits from each others.

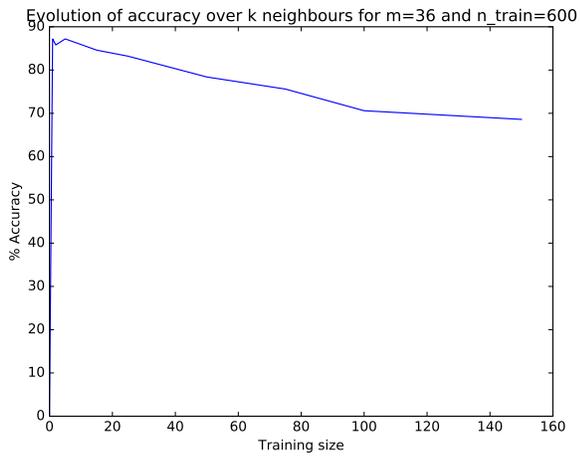


Fig. 6. Influence of k

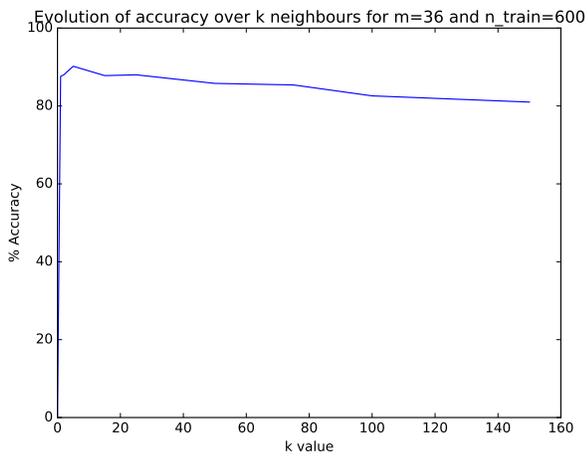
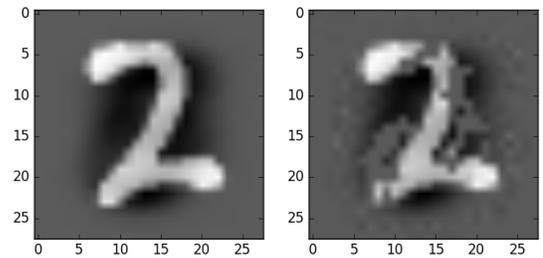
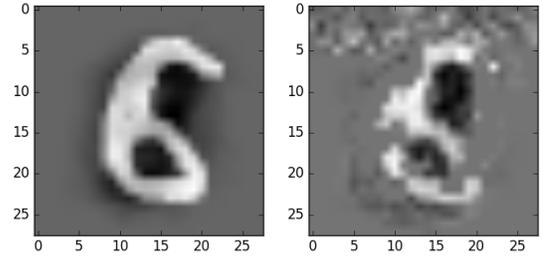


Fig. 7. Influence of k when the distance is taken into account

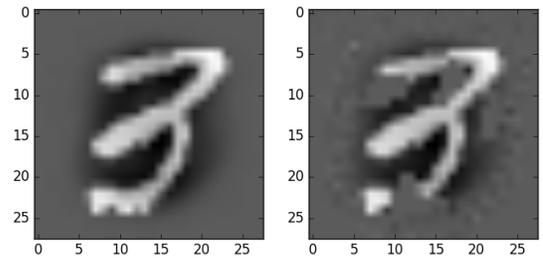


Fig. 8. Digit reconstructed