

Analyse et Traitement de l'Information

TP3: Principal Component Analysis and k -Nearest-Neighbour Classification on the MNIST Dataset.

The MNIST dataset of handwritten digits ¹ has a training set of 60,000 examples, and a test set of 10,000 examples. All images are grayscale and normalized to have the same resolution 28×28 . The data file `mnist_all.mat` along with a Matlab[®] function `traintestMNIST.m` can be downloaded from the course web page at <http://moodle.unige.ch> following the link “TP3. PCA and k-NN classification” → “Data”. To load the data, issue the Matlab[®] command

```
[trainImages, trainLabels, testImages, testLabels] =  
    traintestMNIST(labels, ntrain, ntest);
```

`labels` is a row vector containing the digits to be selected (e.g. using `labels = [3, 4]` only returns digits which are threes or fours), and `ntrain` and `ntest` specify the total number of training and test instances to select, respectively. If `ntrain` and `ntest` are not given, the method `traintestMNIST` will select all the corresponding images. `trainImages` and `testImages` are row vectors representing training and test images, respectively. `trainLabels` and `testLabels` are the corresponding column vectors of labels

1 Principal Component Analysis (PCA) (50%)

Randomly sample 5000 images of the digit “2” from the MNIST dataset. Denote this subset as $\mathcal{X} = \{x_i\}_{i=1}^n$, where each x_i is a 784-dimensional vector corresponding to a 28×28 image of the digit “2”, and $n = 5000$. Perform PCA on \mathcal{X} . Denote the first m ($1 \leq m \leq 784$) principal components (with the largest variance) of \mathcal{X} as $\{\text{PC}_1, \dots, \text{PC}_m\}$. Note, each PC_j is also a 784 dimensional vector.

1. Any image $x_i \in \mathcal{X}$ can be reconstructed using $\{\text{PC}_1, \dots, \text{PC}_m\}$ by

$$r_m(x_i) = \sum_{j=1}^m \langle x_i - \bar{x}, \text{PC}_j \rangle \cdot \text{PC}_j + \bar{x}$$

where $\bar{x} = \sum_{i=1}^n x_i / n$ is the average image, and $\langle \cdot, \cdot \rangle$ denotes the inner product. The corresponding reconstruction error of \mathcal{X} can be defined by

$$\text{err}(\mathcal{X}, m) = \frac{1}{n} \sum_{i=1}^n \|x_i - r_m(x_i)\|^2.$$

- Plot $\text{err}(\mathcal{X}, m)$ as a function of m where $m = 1, \dots, 784$.
- Find m that corresponds to accuracy 50%, 95% and 100%.

¹<http://yann.lecun.com/exdb/mnist/>

2. Sample 5 random images of “2” from MNIST (not necessarily in the subset \mathcal{X}). For each image \mathcal{I} in the 5 samples, and for each $m = 1, 2, \dots, 10$, repeat the following task:
 - Compute the reconstruction of \mathcal{I} using the first m principal components of \mathcal{X} .

In total you should have 50 reconstructed images plus 5 original images. Plot these images.

3. Based on your results in the above tasks, discuss the principles of choosing a proper k (the number of principal components) for data analysis on the MNIST dataset.

2 k -Nearest-Neighbour (k -NN) Classification (50%)

Randomly sample 5000 images of “0”, “1”, “2”, “3” and “4” as the training set. For the testing set, use all available testing images. Using the Matlab[®] interface we provided, this can be done with the following call.

```
[trainImages, trainLabels, testImages, testLabels] =
    traintestMNIST([0,1,2,3,4], 5000);
```

A 1-NN classifier can be implemented as follows. To classify a sample x in the testing set, select the nearest sample from x in the *training set*, then predict the class label of x to be the same as this nearest sample.

1. Compute the baseline 1-NN classification accuracy, which is defined as the percentage of samples correctly classified in the *testing set*;
2. (PCA+1-NN) Repeat the following task for each $m = 10, 20, 30, 40, 50, 100, 150, 200, \dots, 750$.
 - Perform PCA on the training set and reconstruct it using the first m principal components $\{\text{PC}_1, \dots, \text{PC}_m\}$. To classify a testing sample x , compute its reconstruction $r_m(x)$ using $\{\text{PC}_1, \dots, \text{PC}_m\}$, then output the class label of the nearest reconstructed sample in the training set.

Plot the classification accuracy with respect to m . Explain your observations on the effect of m to the classification performance.

Submission

Please archive your report and codes in “Prénom Nom.zip” (replace “Prénom” and “Nom” with your real name), and upload to “Upload TP3. PCA and k-NN classification” on <https://moodle.unige.ch> before **Monday, October 31 2016, 23:59 PM**. Note, the assessment is mainly based on your report, which should include your answers to all questions and the explanations of your experimental results.