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`, or the data file `mnist.npz` along with a python ® script `traintestMNIST.py` 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);
```

or the python 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)

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 $\{\mathtt{PC}_1, \ldots, \mathtt{PC}_m\}$. Note, each $\mathtt{PC}_\jmath$ is also a 784 dimensional vector.

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

$$r_m(x_i) = \sum_{\jmath=1}^{m} \langle x_i - \bar{x},\ \mathtt{PC}_\jmath \rangle \cdot \mathtt{PC}_\jmath + \bar{x}$$

   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

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

[1] `http://yann.lecun.com/exdb/mnist/`

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

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 $m = 1, 2, \ldots, 10$, 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

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 $m = 10, 20, 30, 40, 50, 100, 150, 200, \ldots, 750$.

   - Perform PCA on the training set and reconstruct it using the first m principal components $\{\texttt{PC}_1, \ldots, \texttt{PC}_m\}$. To classify a testing sample x, compute its reconstruction $r_m(x)$ using $\{\texttt{PC}_1, \ldots, \texttt{PC}_m\}$, 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.

3 Factorial Correspondence Analysis (FCA) *

For this exercise, you have to use the python code and data provided in the file "fca.py". In this file you will find two datasets as instances of the pandas.DataFrame class:

- "df1" is the dataframe with the data of the slide 9 of ATI.04

- "df2" is the dataframe with the data of the slide 3 of ATI.04

The code gives the 2D projections of the row and columns profiles for these 2 datasets.

Questions

1. Modify the code such for each dataset, row and column data, it should also give the correlation matrix (you could use the "seaborn.heatmap" function to display them nicely).
2. Comment on these correlations.
3. For the first dataset "df1" only, find which row of data you could delete such that the 2D projection of the column profiles will be the most similar to the original one with all rows of data. In order to compare, please display for each deleted row the overlaying 2D projections of the original and truncated data (as many figures as there are rows in "df1").
4. why it was this raw that we delete (use corr matrix)
   Give an interpretation for the result of question 3, and then delete 2 and 3 rows of data such that the 2D projection of the column profiles will be the most similar to the original one. Display two more figures for the overlaying 2D projections of the original and truncated data, when 2 and 3 rows are deleted.

le on delete celle enlevée du point 3, puis on refait le meme exo.

ou on peut utiliser euclidian interpretation pour y deleted

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, k-NN classification, FCA`" on `https://moodle.unige.ch` before Monday, October 26 2020, 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.

Supplements - not evaluated

1. Define the PCA and present the steps to perform it.
2. Define the variance and inertia of a set of samples, and the links with the co-variance matrix of it and with the PCA.
3. Present the link between PCA and the scalar product.
4. What can you achieve with PCA?
5. Explain in general the context of classification algorithms and present the k-NN classification.