Analyse et Traitement de l'Information

Extra TP1: Hierarchical Clustering (20 pts).

The MNIST dataset of handwritten digits [1] consists of 28×28 grayscale images. To download the dataset use this code:

```
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split

images, labels = fetch_openml('mnist_784', \
            return_X_y=True, as_frame=False, parser='auto')
train_images, test_images, train_labels, test_labels = \
            train_test_split(images, labels, random_state=42)
```

To filter images of the specific class you can use the following method:

```
import numpy as np

def select_with_label(images, labels, desired_labels):
    mask = np.isin(labels, desired_labels)
    return images[mask], labels[mask]

images_of_two, labels_of_two = \
        select_with_label(train_images, train_labels, desired_labels=['2'])
images_of_odd, labels_of_odd = \
        select_with_label(train_images, train_labels, \
        desired_labels=['1', '3', '5', '7', '9'])
```

You can copy-paste from this link.

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

Clustering

Classification is a *supervised* problem, where labeled (training) data is used to label the unlabeled (testing) data.
Clustering is an *unsupervised* problem, where the objective is to group a given (unlabeled) dataset $\{\mathbf{x}\}_{i=1}^{n}$ into k clusters, so that similar samples appear in the same cluster, and dissimilar samples are in different clusters.

Hierarchical Clustering

We will take a look at another useful type of Clustering – Agglomerative Clustering. The idea is to unite the clusters with the shortest distance between them. This method has a high computational complexity so it is applied only **for small datasets**.

Linkage is a method which will be used to compute the distance between the clusters.

- *average* - the distance between two clusters is the average distance between all possible pairs of points where one belongs to one cluster and another one to another
- *single* - the distance between two clusters is the minimal distance between two points where one belongs to one cluster and another one to another
- *complete* - the distance between two clusters is the maximal distance between two points where one belongs to one cluster and another one to another

```
class AgglomerativeClustering:
    def __init__(self, n_clusters=16, linkage="complete"):
        raise NotImplementedError()

    def fit_predict(self, X, y=None):
        #this function returns a cluster number to each element of X
        raise NotImplementedError()

```

Questions

1. Implement AgglomerativeClustering with *complete* distance using *numpy* package. Explain the algorithm in the report.
2. Take 200 instances of "3"s and "7"s from the MNIST dataset. For the case k = 2, give the confusion matrix for both Agglomerative and k-means clustering. (you can use sklearn implementation of KMeans for this task)
3. Sample 200 instances of "3"s and "5"s. Perform both clustering approaches with k = 2. Build the confusion matrix. Based on your results, explain the difference between these two clustering problems:
   - "3" vs. "7"
   - "3" vs. "5".

Submission

Please archive your report and codes in "Prénom Nom.zip" (replace "Prénom" and "Nom" with your real name), and upload to "`Upload Extra TP1 - Hierarchical Clustering.`" on `https://moodle.unige.ch` before Monday, December 1 2023, 21:59 PM. Note, that the assessment is mainly based on your report, which should include your answers to all questions and the experimental results. *Importance is given on the mathematical explanations of your works and your codes should be commented*

Please use the template from the first TP.