TD 3

September 16, 2024

Exercise 1 : Wine data set

Python library *sklearn* provides several toy data sets that are described here. Use the following code for imports and to load Wine dataset :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine

#Loading data in a DataFrame:
data_f = load_wine()

#Loading wine features in a matrix:
X = data_f.data

#Loading wine classes in a vector:
y = data_f.target
```

1 Use *describe()* DataFrame method to get empirical mean and variance of each feature.

2 Code your own PCA function that takes matrix X as an input and return principal components, rotation matrix and eigenvalues.

3 Apply it first on raw data and plot the variance ratio by number of components. What do you observe ? How can you explain that ?

4 To avoid this problem, write a function for centering and scaling data. Then apply your PCA function on transformed data and plot the variance ratio.

Library *sklearn* also provides functions to center/scale data and to perform PCA.

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

#Standardize features
scaler = StandardScaler()
X_cs = scaler.fit_transform(X)

#Apply PCA
pca = PCA()
X_pca_components = pca.fit_transform(X_cs)

#Rotation matrix
rotation_matrix = pca.components_.T
```

```
#Variance ratio
print(pca.explained_variance_ratio_)
```

5 Verify that you get the same results using this method and plot data keeping only two principal components, using one color per class. What do you observe ?

6 Without implementing it, what would be your strategy to perform wine classification based on these two principal components ?

Exercise 2 : Written digits data set

This data set contains 8 × 8 images of written digits converted into vectors of size 64. This can be done using *img.flatten()* or *img.reshape(-1)* and images can be restored from vectors using *v.reshape(8,8)*.

```
from sklearn.datasets import load_digits

#Loading data in a DataFrame:
data_f = load_digits()

#Loading wine features in a matrix:
X = data_f.data

#Loading wine classes in a vector:
y = data_f.target
```

1 Apply PCA and plot cumulative variance ratio.

We would like to compress images by a factor 4 using principal components.

2 What quantity of information, from a variance point of view, can we expect to keep with such a compression ?

3 Project data on 16 principal components and write a function that takes as an input the rotation matrix and performs the inverse PCA transformation.

4 Display examples of original and recovered images using this method.

5 Create your own 8×8 image of a written digit and apply the two previous steps on it.

6 Compute norm-2 distance between your image and data set images. Do the same on their compressed versions.

7 Find, for both situations (original and compressed), the $N = 100$ nearest images of the data set from your own image.

8 Using data_f.target vector, plot histograms of labels contained in these N nearest neighbors and compare.