Lab Clustering n°1

A good clustering algorithm sorts objects and returns clusters where the objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible.

Exploring and cleaning

1. Import the dataset from github (under the Lab clustering n°1 folder)

2. In order to cluster objects, a metric is necessary to calculate differences between objects along each feature. So, to avoid having to create a metric between non numeric values, conserve only the numeric columns of the dataset.

3. Prepare the resultant dataset for analysis.

PCA

Dimension reduction is essential to clustering, as it gives the most important features affecting the distribution of the objects to classify.

1. Using the standard scaler, standardise the numerical dataset.

2. Observe the correlation matrix (you can use **seaborn**'s **heatmap** visualisation to have a nice readable plot). What conclusion can we drive from the corr matrix?

3. Get the eigen vectors and eigen values of the dataset (valeurs propres et vecteur propres)

4. In order to decide which dimension we hold on to cluster the dataset's objects, we're going to use the cumulative explained variance (method explained is at the end of this document)

5. Visualize the individual explained variance of each eigen value and the cumulative explained variance.

6. We want to hold on only to the minimum number of dimensions which explain at least 90% of the total variance. How many dimension will our dataset have?

7. Now that the number of dimension to work on to has been established, use the **PCA** function from the **sklearn** module to apply the pca method to our dataset. NB: PCA function accepts a 'n_component' parameter.

Clustering

Now that we conserved only the most important dimensions describing our datasets, it's time to cluster the objects along these dimensions.
In order to do that, we'll start by visualize the some plots to guess how many clusters will we obtain (We will be using the K-mean clustering algorithm).

1. Visualize the scatter between the first and the second dimension values of our dataset. According to the obtained plot, will the clustering be effective if we limit ourselves to two clusters?

2. Using the Kmean function of the sklearn module, cluster the dataset's objects into 2 clusters.

3. Visualize same scatter of question 1 after changing the colors of each point according to the clusters their in (use the c parameter of the scatter function). Describe the cluster.

4. Do the same after clustering the dataset's objects into 3 clusters. Describe the difference between the two approaches.

the cumulative explained variance

The cumulative explained variance is a method designed to decide which dimension conserve and which one throw away when applying a dimensions reduction.

Let M be a matrix and $\lambda_1, .., \lambda_n$ its eigen values.

Then the individual explained variance of the eigen value λ_i is given by:

$$expvar(\lambda_i) = \frac{\lambda_i}{\sum \lambda_k}$$

Then the cumulative explained variance of the eigen values $(\lambda_1, \lambda_2, .., \lambda_i)$ is given by:

$$cummvar = \sum_{j=1}^{i} expvar(\lambda_j)$$

Example

Let M be a 3 dimension matrix which eigen values are $\lambda_1 = 5$, $\lambda_2 = 4$ and $\lambda_3 = 1$.

Then the individual explained variance of each eigen value is :

$$expvar(\lambda_1) = \frac{5}{10} = 0.5 \quad expvar(\lambda_2) = \frac{4}{10} = 0.4 \quad and\ expvar(\lambda_3) = 0.1$$

If we want to conserve the minimum number of dimensions which describe the matrix's variance by at least 90%, we can reject the 3 dimension on the basis of:

$$cummvar(\lambda_1, \lambda_2) = 0.9 = 90\%$$

This means that the first two dimensions of our system explain 90% of the data variance