UNIVERSIDAD DE ANTIOQUIA
1803

Lab II - Clustering

Machine Learning II

Prerequisites

Python

Python

- Data structures (properties of lists, tuples, dicts, built-in modules…)
- Classes
- Packages and modules

NumPy

- Arrays
- Inner product
- Vector - Matrix product
- Distances

Linear Algebra

Class Concepts

- Vectors and matrices
- Properties of matrices
- Eigendecomposition
- Clustering

Workshop II

1. Research about the **Spectral Clustering** method, and answer the following questions:
    a. In which cases might it be more useful to apply?
    b. What are the mathematical fundamentals of it?
    c. What is the algorithm to compute it?
    d. Does it hold any relation to some of the concepts previously mentioned in class? Which, and how?

2. Research about the **DBSCAN** method, and answer the following questions:
    a. In which cases might it be more useful to apply?
    b. What are the mathematical fundamentals of it?
    c. Is there any relation between DBSCAN and Spectral Clustering? If so, what is it?

3. What is the elbow method in clustering? And which flaws does it pose to assess quality?

4. Remember the *unsupervised* Python package you created in the previous unit? 😀It's time for an upgrade.
    a. Implement the **k-means** module using Python and Numpy
    b. Implement the **k-medoids** module using Python and Numpy
    c. Remember to keep consistency with Scikit-Learn API as high as possible

5. Let's use the newly created modules in *unsupervised* to cluster some toy data.
    a. Use the following code snippet to create scattered data X

```
from sklearn.datasets import make_blobs
X, y = make_blobs(
    n_samples=500,
    n_features=2,
    centers=4,
    cluster_std=1,
    center_box=(-10.0, 10.0),
    shuffle=True,
    random_state=1,
)
```

    b. Plot the resulting dataset. How many clusters are there? How far are they from one another?
    c. For both k-means and k-medoids (your implementations), calculate the silhouette plots and coefficients for each run, iterating K from 1 to 5 clusters.
    d. What number of K got the best silhouette score? What can you say about the figures? Is this the expected result?

6. Use the following code snippet to create different types of scattered data:

```
import numpy as np
from sklearn import cluster, datasets, mixture

# ============
# Generate datasets. We choose the size big enough to see the scalability
# of the algorithms, but not too big to avoid too long running times
# ============
n_samples = 500
noisy_circles = datasets.make_circles(n_samples=n_samples, factor=0.5, noise=0.05)
noisy_moons = datasets.make_moons(n_samples=n_samples, noise=0.05)
blobs = datasets.make_blobs(n_samples=n_samples, random_state=8)
no_structure = np.random.rand(n_samples, 2), None
```

```
# Anisotropically distributed data
random_state = 170
X, y = datasets.make_blobs(n_samples=n_samples, random_state=random_state)
transformation = [[0.6, -0.6], [-0.4, 0.8]]
X_aniso = np.dot(X, transformation)
aniso = (X_aniso, y)

# blobs with varied variances
varied = datasets.make_blobs(
    n_samples=n_samples, cluster_std=[1.0, 2.5, 0.5], random_state=random_state
)
```

a. Plot the different datasets in separate figures. What can you say about them?
b. Apply k-means, k-medoids, DBSCAN and Spectral Clustering from Scikit-Learn over each dataset and compare the results of each algorithm with respect to each dataset.

Useful Resources

https://github.com/rushter/MLAlgorithms/blob/035e489a879d01a84fffff74885dc6b1bca3c96f/mla/kmeans.py
https://github.com/patchy631/machine-learning/blob/main/ml_from_scratch/KMeans_from_scratch.ipynb
https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html