**What, why and how of Spectral Clustering!**

[**Keerthana V**](https://www.analyticsvidhya.com/blog/author/keerthz/)**— May 24, 2021**

[Classification](https://www.analyticsvidhya.com/blog/category/classification/) [Clustering](https://www.analyticsvidhya.com/blog/category/clustering/) [Intermediate](https://www.analyticsvidhya.com/blog/category/intermediate/) [Machine Learning](https://www.analyticsvidhya.com/blog/category/machine-learning/) [Project](https://www.analyticsvidhya.com/blog/category/project/) [Python](https://www.analyticsvidhya.com/blog/category/python-2/) [Structured Data](https://www.analyticsvidhya.com/blog/category/structured-data/) [Unsupervised](https://www.analyticsvidhya.com/blog/category/unsupervised/)

*This article was published as a part of the*[*Data Science Blogathon*](https://datahack.analyticsvidhya.com/contest/data-science-blogathon-8/)

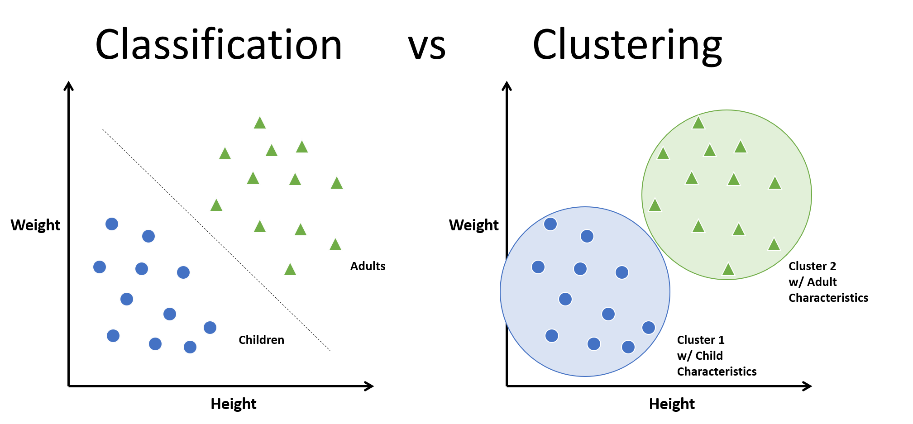
When given a bunch of movies to organize, you might sort them based on similarity measures such as genre, year of release, casting, revenue generated, and so on. While you categorize them based on a measure, it doesn’t mean that a measure used by others is of no good.

Our perception of data is greatly influenced by the clustering measure being used. When you believe that two sci-fi movies belong to the same cluster, someone else might consider two movies released in the same year to be in the same cluster. Because you might have used the genre as a measure to cluster and the other person might have used the year of release as a measure.

Classifications vs Clustering

As humans, in machine learning, a widely used unsupervised algorithm to group unlabeled data points by similarity and distance measures is clustering. If the data points are labeled, grouping is known as classification. Clustering algorithms have their application in many places including anomaly detection, image segmentation, search result grouping, market segmentation, and social network analysis. Clustering is one of the initial steps done in exploratory data analysis to visualize the similarity and to identify the pattern lying hidden in data points. The motive of clustering is to find the similarity within a cluster and the difference between two clusters.

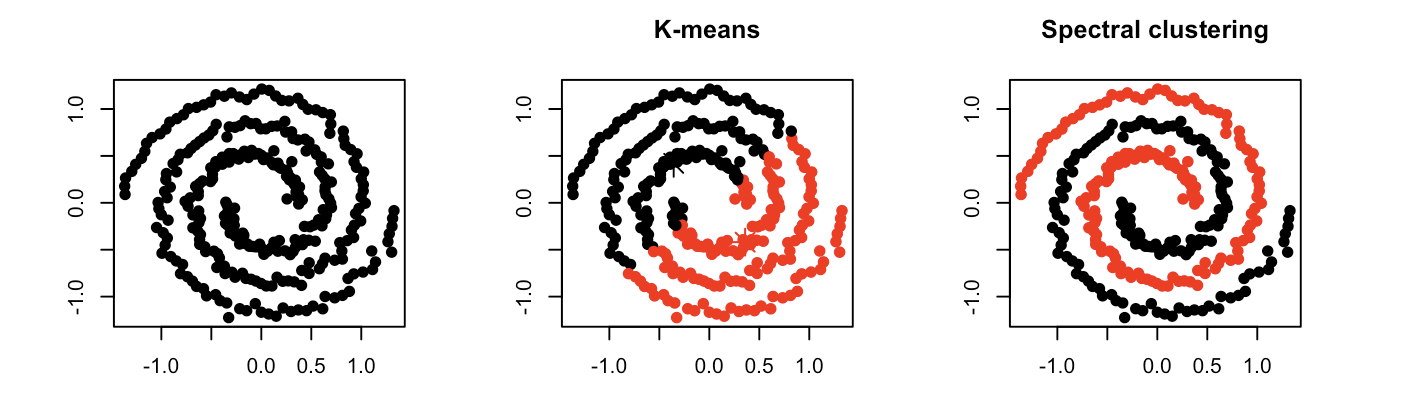
Support Vector Machines, Decision trees, Random forests, Linear classifiers and Neural networks are few classification algorithms whereas K means clustering, Fuzzy analysis clustering, Mean shift, DBSCAN and Spectral are clustering algorithms.



There are two major approaches in clustering. They are:

1. Compactness
2. Connectivity

In compactness, the points are closer to each other and are compact towards the cluster center. Distance is used as a measure to compute closeness. There are different types of distance metrics that are in use. A few of them are Euclidean distance, Manhattan distance, Minkowski distance, and Hamming distance. K-means algorithm uses the compactness approach. In connectivity, the points in a cluster are either immediately next to each other (epsilon distance) or connected. Even if the distance is less, they are not put in the same cluster. Spectral clustering is one of the techniques to follow this approach.



How to do Spectral Clustering?

The three major steps involved in spectral clustering are: constructing a similarity graph, projecting data onto a lower-dimensional space, and clustering the data. Given a set of points S in a higher-dimensional space, it can be elaborated as follows:

1. Form a distance matrix

2. Transform the distance matrix into an affinity matrix A

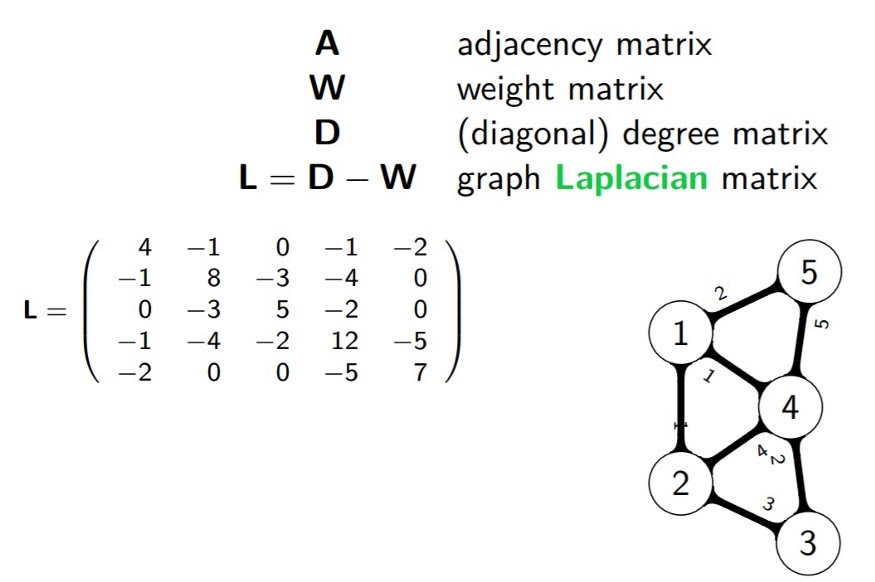
3. Compute the degree matrix D and the Laplacian matrix L = D – A.

4. Find the eigenvalues and eigenvectors of L.

5. With the eigenvectors of k largest eigenvalues computed from the previous step form a matrix.

6. Normalize the vectors.

7. Cluster the data points in k-dimensional space.



**Pros and Cons of Spectral Clustering**

Spectral clustering helps us overcome two major problems in clustering: one being the shape of the cluster and the other is determining the cluster centroid. K-means algorithm generally assumes that the clusters are spherical or round i.e. within k-radius from the cluster centroid. In K means, many iterations are required to determine the cluster centroid. In spectral, the clusters do not follow a fixed shape or pattern. Points that are far away but connected belong to the same cluster and the points which are less distant from each other could belong to different clusters if they are not connected. This implies that the algorithm could be effective for data of different shapes and sizes.

When compared with other algorithms, it is computationally fast for sparse datasets of several thousand data points. You don’t need the actual dataset to work with. Distance or Though it might be costly to compute for large datasets as eigenvalues and eigenvectors need to be computed and then clustering is to be done. But the algorithms try to cut the cost. The number of clusters (k) needs to be fixed before starting the procedure.

**Where can I use spectral clustering?**

Spectral clustering has its application in many areas which includes: image segmentation, educational data mining, entity resolution, speech separation, spectral clustering of protein sequences, text image segmentation. Though spectral clustering is a technique based on graph theory, the approach is used to identify communities of vertices in a graph based on the edges connecting them. This method is flexible and allows us to cluster non-graph data as well either with or without the original data.

Python packages for spectral clustering: [spectralcluster](https://pypi.org/project/spectralcluster/" \t "_blank). SpectralCluster is a python library that has inbuilt code for spectral clustering. There are other packages with which we can implement the spectral clustering algorithm.

from spectralcluster import SpectralClusterer

clusterer = SpectralClusterer(

min\_clusters=2,

max\_clusters=100,

p\_percentile=0.95,

gaussian\_blur\_sigma=1)

labels = clusterer.predict(X)

where X is a NumPy array of shape (n\_samples, n\_features) and the returned labels is a NumPy array of shape (n\_samples, ).

Let us try an example with spectral clustering algorithm available from sklearn package.

Importing the required packages

from sklearn.cluster import SpectralClustering

from sklearn.datasets.samples\_generator import make\_blobs

import matplotlib.pyplot as plt

from numpy import random

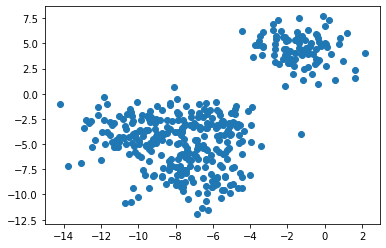
Preparing the data

random.seed(1)

x, \_ = make\_blobs(n\_samples=400, centers=4, cluster\_std=1.5)

plt.scatter(x[:,0], x[:,1])

plt.show()



Clustering with the spectral clustering and visualizing the data

sc=SpectralClustering(n\_clusters=4).fit(x)

SpectralClustering(affinity='rbf', assign\_labels='kmeans', coef0=1, degree=3,

eigen\_solver=None, eigen\_tol=0.0, gamma=1.0,

kernel\_params=None, n\_clusters=4, n\_components=None,

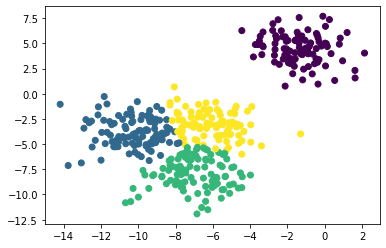
n\_init=10, n\_jobs=None, n\_neighbors=10, random\_state=None)

labels = sc.labels\_

plt.scatter(x[:,0], x[:,1], c=labels)

plt.show()

print(sc)



Let’s try changing the number of clusters.

f = plt.figure()

f.add\_subplot(2, 2, 1)

for i in range(2, 6):

sc = SpectralClustering(n\_clusters=i).fit(x)

f.add\_subplot(2, 2, i-1)

plt.scatter(x[:,0], x[:,1], s=5, c=sc.labels\_, label="n\_cluster-"+str(i))

plt.legend()

plt.show()

