# Comparison of clustering algorithms

Raúl Maximiliano Urrutia Hernández

November 7th, 2019

# Introduction

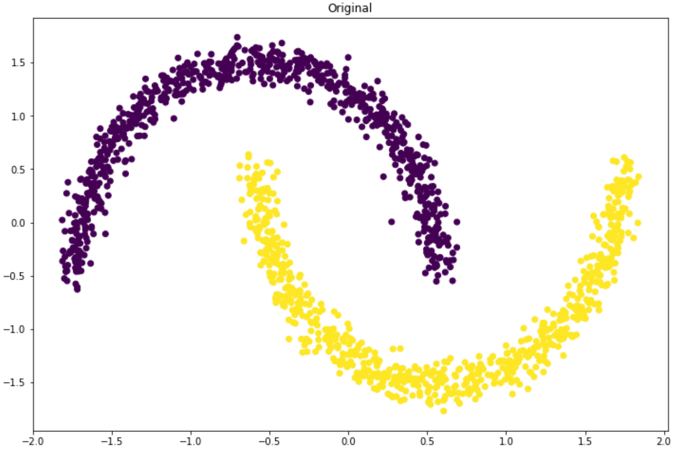
Here I compare the algorithms

- Affinity Propagation (AP)
- Spectral Clustering (SC)
- DBSCAN
- K-means
- Hierarchical Clustering (HClust)

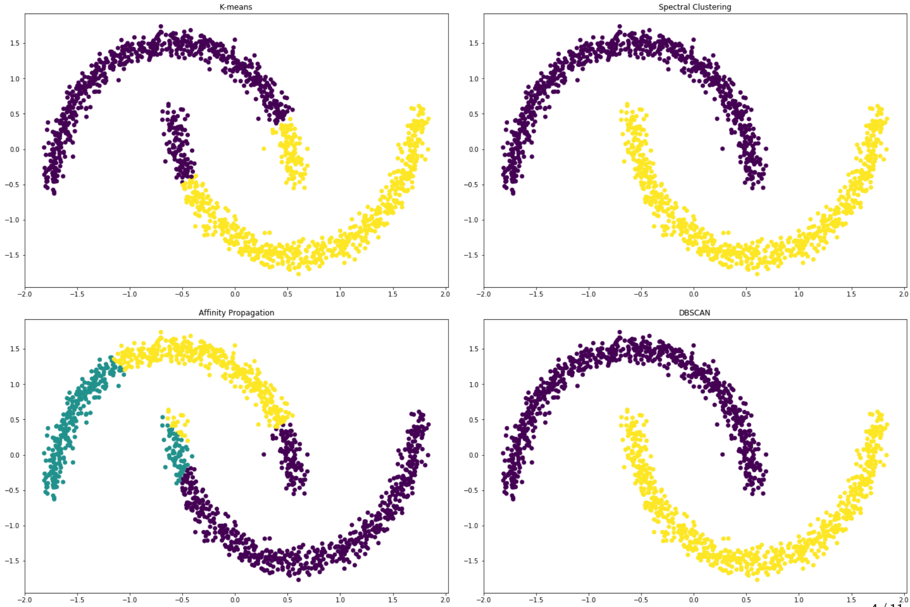
using the scikit-learn library and the make_moons dataset.
I use the **Rand Index** and the **Mutual Information** to measure the performance of the algorithms.

# Original Data

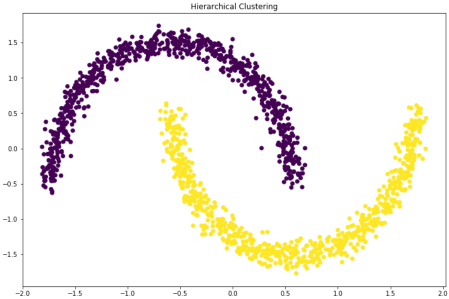
Here is the plot of the make_moons dataset with 1500 samples and 0.05 of noise:



# Clustered Data



# Clustered Data



Apparently DBSCAN, Spectral and Hierarchical Clustering give the best results.

The running times (in seconds) of these algorithms are

| Algorithm | K-means | AP | SC | DBSCAN | HClust |
|---|---|---|---|---|---|
| Time (sec) | 0.0313 | 11.633 | 1.234 | 0.0156 | 0.0469 |

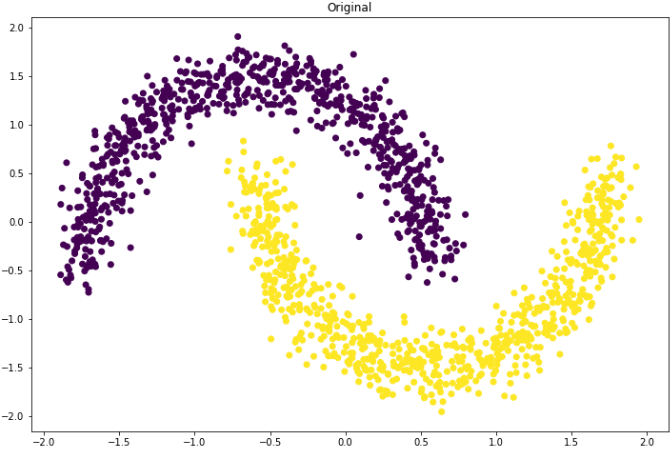
# Performance Evaluation

The Rand Index is a measure of the similarity of two assignments, meanwhile the Mutual Information measures the agreement of two assignments. Here we compare the assignments of each algorithm with the original one. In both cases, close values to 1 means better performance.

| Algorithm | Ad. Rand Ind. | Ad. M.I. |
|---|---|---|
| K-means | 0.4953 | 0.3949 |
| AP | 0.4069 | 0.2986 |
| SC | 1.0 | 1.0 |
| DBSCAN | 1.0 | 1.0 |
| HClust | 1.0 | 1.0 |

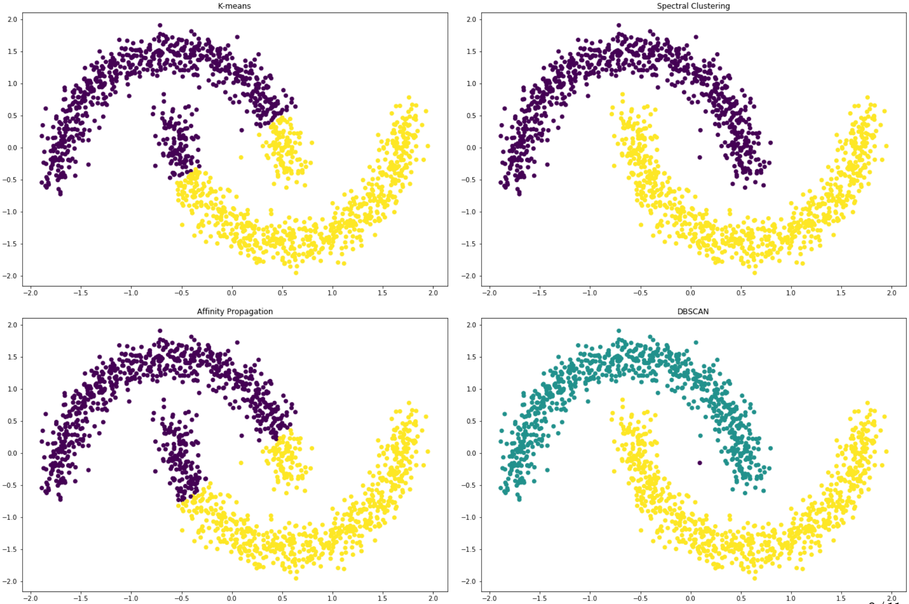
# Original Data

Now lets check the dataset with 0.09 of noise:



I made another test with this dataset without changing the hyperparameters.

# Clustered Data



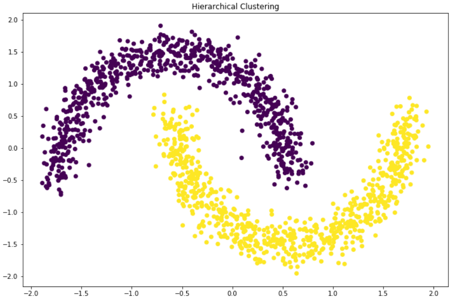
K-means

Spectral Clustering

Affinity Propagation

DBSCAN

# Clustered Data



The running times (in seconds) of these algorithms are

| Algorithm | K-means | AP | SC | DBSCAN | HClust |
|---|---|---|---|---|---|
| Time (sec) | 0.0157 | 15.2239 | 0.4999 | 0.0155 | 0.0312 |

# Performance Evaluation

| Algorithm | Ad. Rand Ind. | Ad. M.I. |
|---|---|---|
| K-means | 0.4915 | 0.3915 |
| AP | 0.4339 | 0.3702 |
| SC | 1.0 | 1.0 |
| DBSCAN | 0.9987 | 0.9927 |
| HClust | 1.0 | 1.0 |

# Conclusions

- DBSCAN and K-means are the fastest, followed by Hierarchical clustering.
- The slowest algorithm is Affinity Propagation, and it is very sensitive to the parameters.
- In terms of the performance metrics that I used, Hierarchical, Spectral and DBSCAN are the best (in both, noise 0.05 and 0.09), so in this special case these three algorithms are very suitable.
- In general there is no best algorithm, since some algorithms can perform better with certain problems than others. For example, Spectral clustering performs well with a few number of clusters, meanwhile hierarchical can be used with many clusters; k-means can be used if we have a large number of entries and if even cluster sizes are wanted, and DBSCAN can deal with uneven cluster sizes.