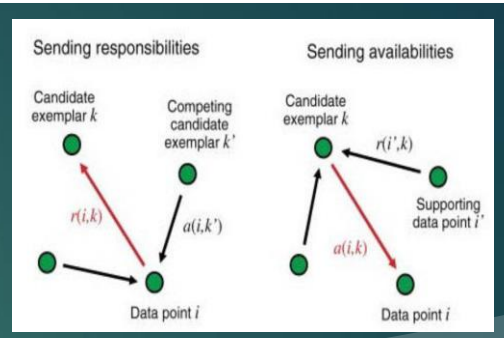
Machine Learning
Unsupervised Learning
Clustering

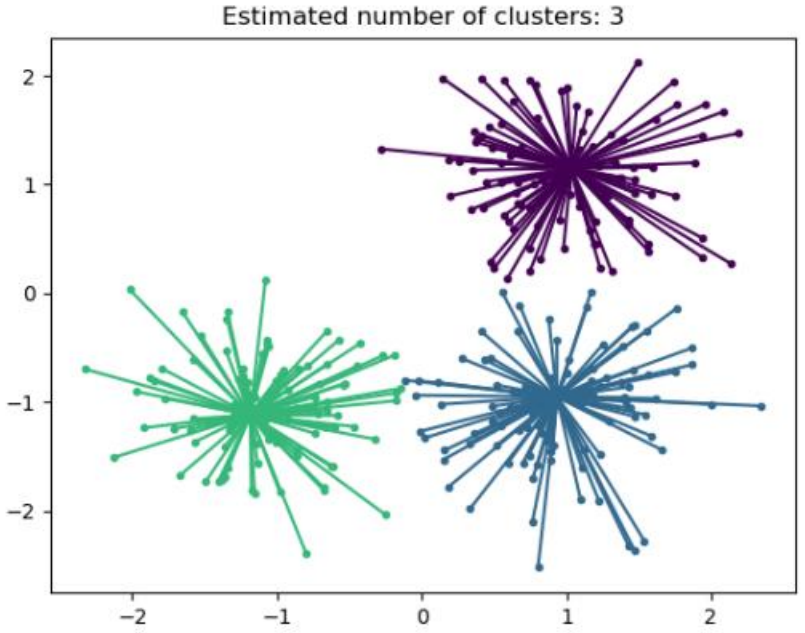
Clustering all algorithms with python codes

Affinity Propagation




```
class sklearn.cluster.AffinityPropagation(*, damping=0.5, max_iter=200,
convergence_iter=15, copy=True, preference=None, affinity='euclidean', verbose=False,
random_state=None) ¶
```

- Affinity Propagation is based on the concept of "message-passing" between data points to identify cluster centres and assign data points to these centres automatically.
- The objective of the algorithm is to identify the most representative exemplars of the overall data and employ them to cluster the data into compatible groups
- This algorithm is especially suitable for data with numerous clusters or data exhibiting intricate, non-linear distribution patterns.



Advantages of Affinity Propagation

- Does not require the number of clusters to be specified beforehand, making it a more flexible clustering algorithm.
- Can produce high-quality clusters even when the data points have different densities or sizes.
- Can be used to cluster data with complex relationships and non-linear structures.
- Can be used in a wide range of applications, including image segmentation, customer segmentation, and gene expression analysis.

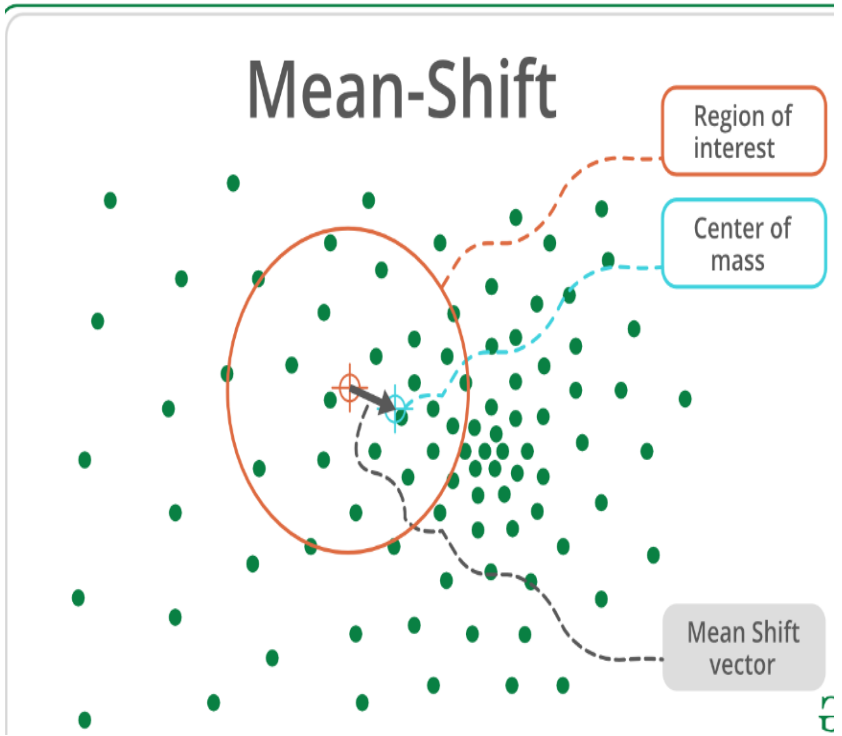
Disadvantages of Affinity Propagation

- Can be computationally expensive, especially for large datasets, making it unsuitable for large-scale clustering problems
- May not always produce the best results compared to other clustering algorithms, such as K-Means or Gaussian Mixture Models.
- Can be sensitive to the choice of similarity metric used to measure the similarities between data points.
- Can be sensitive to the choice of similarity metric used to measure the similarities between data points.

Mean Shift

```
class sklearn.cluster.MeanShift(*, bandwidth=None, seeds=None, bin_seeding=False, min_bin_freq=1, cluster_all=True,
n_jobs=None, max_iter=300) ¶                                                                                [source]
```

- **Meanshift** is falling under the category of a clustering algorithm in contrast of Unsupervised learning that assigns the data points to the clusters iteratively by shifting points towards the mode
- mode is the highest density of data points in the region, in the context of the Meanshift
- As such, it is also known as the **Mode-seeking algorithm**
- Mean-shift algorithm has applications in the field of image processing and computer vision.



Advantage of Mean Shift

- Finds variable number of modes
- Robust to outliers
- General, application-independent tool
- Model-free, doesn’t assume any prior shape like spherical, elliptical, etc. on data clusters
- Just a single parameter (window size h) where h has a physical meaning (unlike k-means)

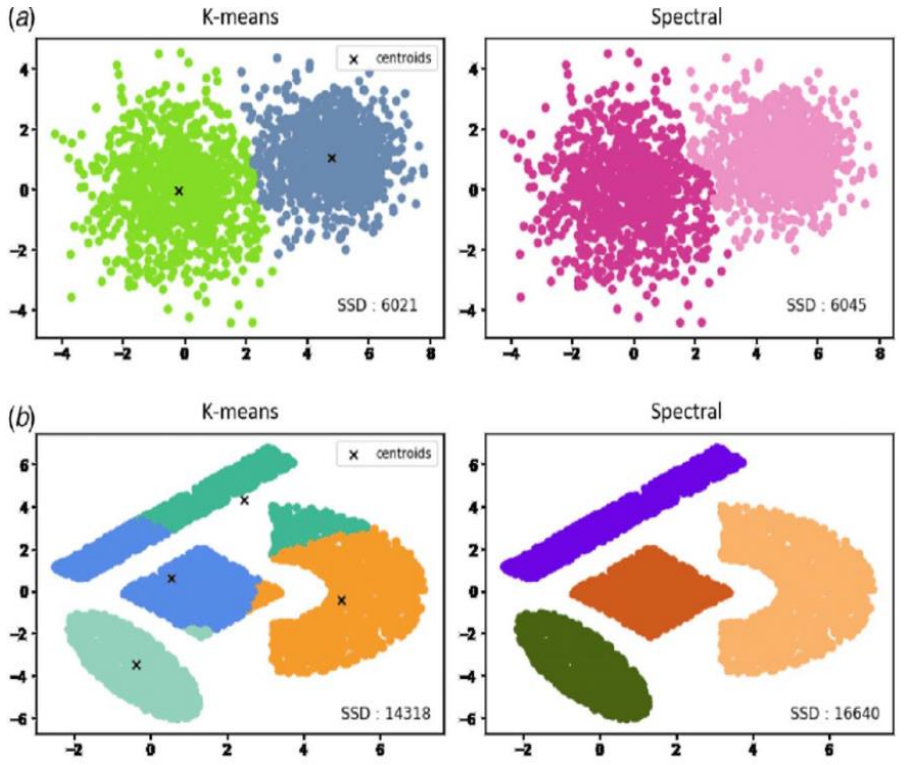
Disadvantages of Mean Shift

- Output depends on window size
- Window size (bandwidth) selecHon is not trivial
- Computationally (relatively) expensive (approx 2s/image)
- Doesn’t scale well with dimension of feature space

Spectral Clustering

class `sklearn.cluster.`**`SpectralClustering`**(*n_clusters=8, *, eigen_solver=None, n_components=None, random_state=None, n_init=10, gamma=1.0, affinity='rbf', n_neighbors=10, eigen_tol='auto', assign_labels='kmeans', degree=3, coef0=1, kernel_params=None, n_jobs=None, verbose=False*) [source]

- Spectral Clustering is a variant of the clustering algorithm that uses the connectivity between the data points to form the clustering.
- It uses eigenvalues and eigenvectors of the data matrix to forecast the data into lower dimensions space to cluster the data points.
- It is based on the idea of a graph representation of data where the data point are represented as nodes and the similarity between the data points are represented by an edge



Advantages of Spectral Clustering

- Scalability: Spectral clustering can handle large datasets and high-dimensional data, as it reduces the dimensionality of the data before clustering.
- Flexibility: Spectral clustering can be applied to non-linearly separable data, as it does not rely on traditional distance-based clustering methods
- Robustness: Spectral clustering can be more robust to noise and outliers in the data, as it considers the global structure of the data, rather than just local distances between data points.

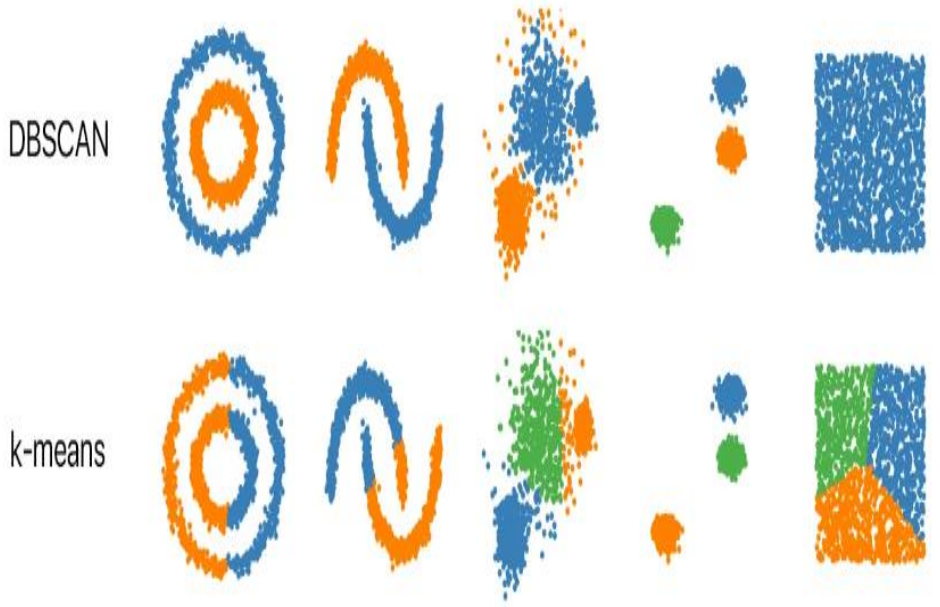
Disadvantage of Spectral Clustering

- Complexity: Spectral clustering can be computationally expensive, especially for large datasets.
- As it requires the calculation of eigenvectors and eigenvalues
- Model selection: Choosing the right number of clusters and the right similarity matrix can be challenging and may require expert knowledge or trial and error.

Density-Based Spatial Clustering of Applications with Noise -DBSCAN

```
class sklearn.cluster.DBSCAN(eps=0.5, *, min_samples=5, metric='euclidean', metric_params=None, algorithm='auto',
leaf_size=30, p=None, n_jobs=None)                                                                    [source]
```

- Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense.
- It comprises many different methods based on differential evolution.
- The goal is to identify dense regions, which can be measured by the number of objects close to a given point
- Two important parameters are required for DBSCAN: epsilon (“eps”) and minimum points (“MinPts”).



Advantages of DBSCAN

- DBSCAN can discover clusters of arbitrary shape, unlike k-means.
- it is robust to noise, as it can identify points that do not belong to any cluster as outliers.
- It does not require the number of clusters to be specified in advance.

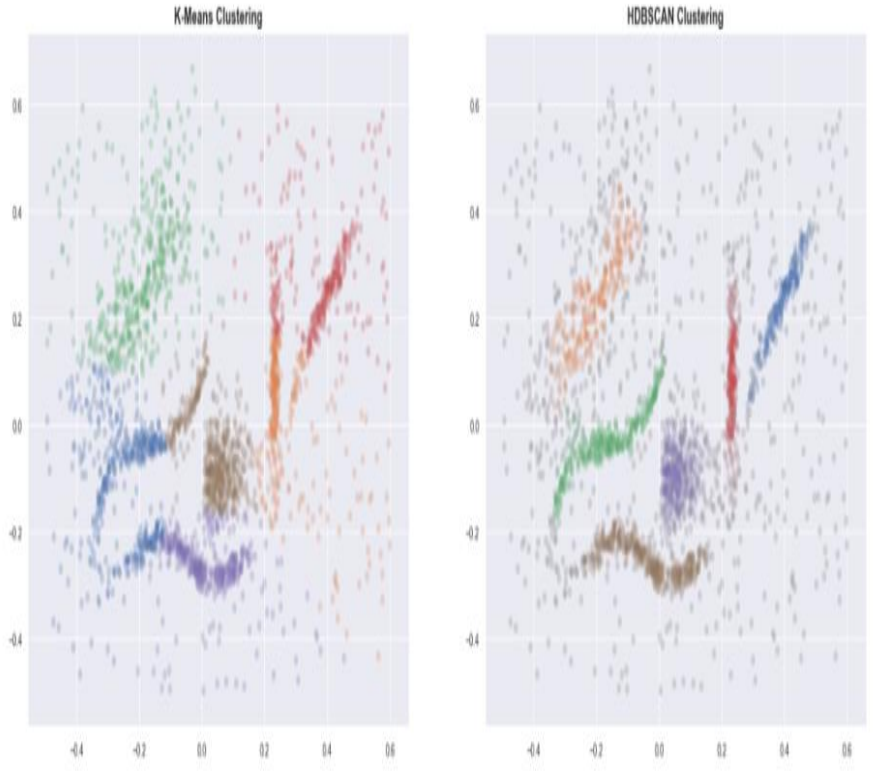
Disadvantages of DBSCAN

- It is sensitive to the choice of the Eps and MinPts parameters.
- It does not work well with clusters of varying densities.
- It has a high computational cost when the number of data points is large.
- It is not guaranteed to find all clusters in the data.

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)

```
class sklearn.cluster.HDBSCAN(min_cluster_size=5, min_samples=None,
cluster_selection_epsilon=0.0, max_cluster_size=None, metric='euclidean', metric_params=None,
alpha=1.0, algorithm='auto', leaf_size=40, n_jobs=None, cluster_selection_method='eom',
allow_single_cluster=False, store_centers=None, copy=False)                    [source]
```

- HDBSCAN is a clustering algorithm that is designed to uncover clusters in datasets based on the density distribution of data points.
- Unlike some other clustering methods, it doesn't requires specifying the number of clusters in advance, making it more adaptable to different datasets.
- It uses high-density regions to identify clusters and views isolated or low-density points as noise.
- HDBSCAN is especially helpful for datasets with complex structures or varying densities because it creates a hierarchical tree of clusters that enable users to examine the data at different levels of granularity
- HDBSCAN is a clustering algorithm used in unsupervised learning to identify groups of similar data points, also known as clusters, within a dataset



Advantages of HDBSCAN

- **Automatic cluster Discovery:** It automatically determines the number of clusters in the dataset without requiring a prior specification, making it suitable for datasets with varying densities and complex structures.
- **Handling Cluster Shapes:** It can identify clusters of varying shapes and sizes, including clusters that are non-convex and have irregular shapes.
- **Hierarchical Clustering:** HDBSCAN constructs a hierarchical clustering structure allowing exploration of clusters at different levels of granularity, providing valuable insights into the data's underlying structure.

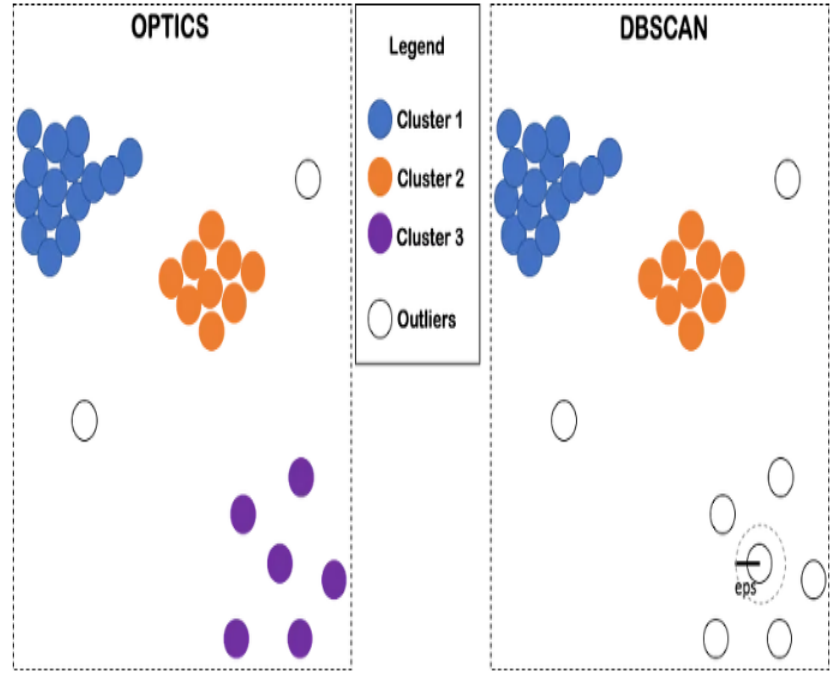
Disadvantages of HDBSCAN

- **Computationally Intensive:** HDBSCAN can be computationally expensive, particularly for large datasets, due to the construction of the minimum spanning tree and the calculation of mutual reachability distances.
- **Sensitive to Distance metric:** In HDBSCAN, the distance metric used can influence the clustering results. Some distance metrics may not accurately capture the data's underlying structure, resulting in suboptimal clustering results.
- **Parameter Sensitivity**: Although HDBSCAN is less sensitive to parameter settings than some other clustering algorithms, it still requires parameter tuning, particularly for the minimum cluster size and minimum sample parameters, which can influence clustering results.

Optics-Ordering Points to identify the Clustering Structure

```
class sklearn.cluster.OPTICS(*, min_samples=5, max_eps=inf, metric='minkowski', p=2,
metric_params=None, cluster_method='xi', eps=None, xi=0.05, predecessor_correction=True,
min_cluster_size=None, algorithm='auto', leaf_size=30, memory=None, n_jobs=None)   [source]
```

- The main idea behind OPTICS is to extract the clustering structure of a dataset by identifying the density-connected points.
- The algorithm builds a density-based representation of the data by creating an ordered list of points called the reachability plot.
- Each point in the list is associated with a reachability distance,
- which is a measure of how easy it is to reach that point from other points in the dataset.
- Points with similar reachability distances are likely to be in the same cluster.
- finds core sample of high density and expands clusters from them [1]. Unlike DBSCAN, keeps cluster hierarchy for a variable neighborhood radius



Example data with varying density. OPTICS performs better than DBSCAN. (Image by author)

Advantages of Optics

- OPTICS clustering doesn’t require a predefined number of clusters in advance.
- Clusters can be of any shape, including non-spherical ones.
- Able to identify outliers(noise data)

Disadvantages Of Optics

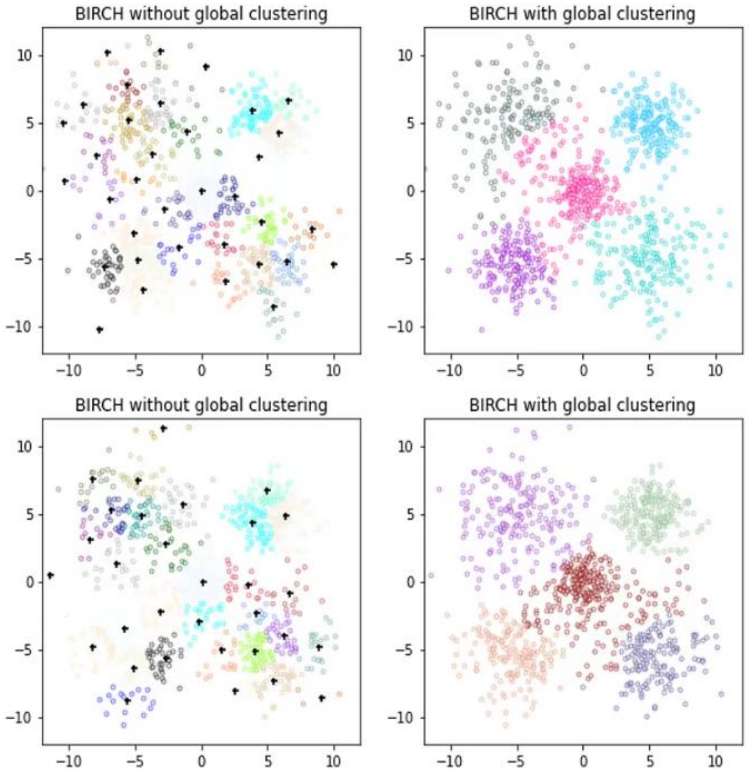
- It fails if there are no density drops between clusters.
- It is also sensitive to parameters that define density(radius and the minimum number of points) and proper parameter settings require domain knowledge.

Branch -Balance Iterative Reducing and Clustering using Hierarchies

class `sklearn.cluster.`**`Birch`**(*, *threshold=0.5*, *branching_factor=50*, *n_clusters=3*, *compute_labels=True*, *copy=True*) [source]

¶

- **Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)** is a clustering algorithm that can cluster large datasets by first generating a small and compact summary of the large dataset that retains as much information as possible.
- This smaller summary is then clustered instead of clustering the larger dataset. BIRCH is often used to complement other clustering algorithms by creating a summary of the dataset that the other clustering algorithm can now use.



Advantages of Brach

- Scalability: BIRCH is designed to efficiently handle massive datasets by utilizing memory-efficient data structures and a hierarchical clustering approach.
- Fast clustering: The CF tree structure allows for fast traversal and clustering, making it suitable for real-time applications.
- Automatic determination of cluster count: BIRCH can automatically determine the number of clusters by adaptively splitting them based on the specified threshold.

Disadvantages of Brach

- Sensitive to parameter tuning: Proper configuration of the branching factor, threshold, and other parameters is crucial to obtain optimal clustering results.
- Limited to spherical clusters: BIRCH performs well when dealing with spherical clusters but may struggle with clusters of complex shapes.