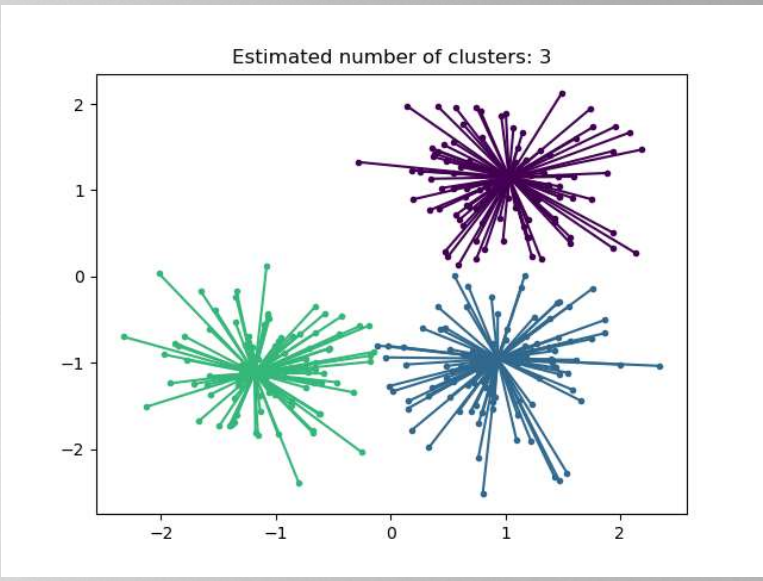
Machine learning clustering

Affinity Propagation

- Affinity Propagation creates clusters by sending messages between pairs of samples until convergence
- A dataset is then described using a small number of exemplars, which are identified as those most
- Representative of other samples.
- The messages sent between pairs represent the suitability for one
- Sample to be the exemplar of the other, which is updated in response to the values from other pairs.
- Affinity Propagation can be interesting as it chooses the number of clusters based on the data provided.
- All similarities between data points are represented in a similarity matrix.



Advantages :

- Affinity Propagation can identify the number of clusters automatically without specifying the number of clusters in advance.
- It can handle clusters of arbitrary shapes and sizes.
- It can handle datasets with noisy or incomplete data.
- It is relatively insensitive to the choice of initial parameters.

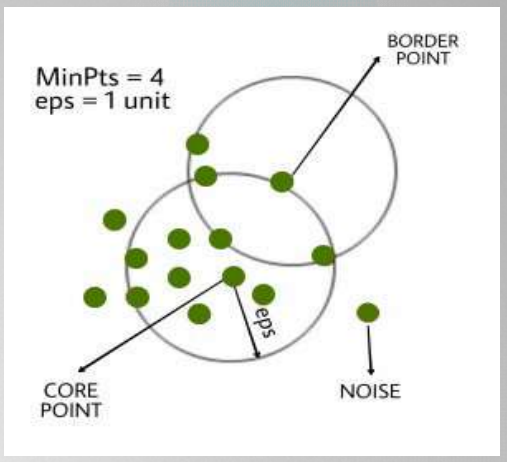
Disadvantages :

- The main disadvantage of Affinity Propagation is that it's quite slow and memory-heavy, making it difficult to scale to larger datasets.
- In addition, it also assumes the true underlying clusters are globular.
- It is hard to know the value of the parameter preferences which can yield an optimal **clustering** solutio

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

- DBSCAN is a popular clustering algorithm that is particularly useful for identifying clusters of varying shapes and sizes in data with noise and outliers.
- It works based on the density of data points in the feature space.
- Here's an overview of how DBSCAN works and its key features:



Steps Used In DBSCAN Algorithm :

❖Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.

❖For each core point if it is not already assigned to a cluster, create a new cluster.

❖Find recursively all its density-connected points and assign them to the same cluster as the core point.

A point *a* and *b* are said to be density connected if there exists a point *c* which has a sufficient number of points in its neighbors and both points *a* and *b* are within the *eps distance*. This is a chaining process. So, if *b* is a neighbor of *c*, *c* is a neighbor of *d*, and *d* is a neighbor of *e*, which in turn is neighbor of *a* implying that *b* is a neighbor of *a*.

❖Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

Advantages **:**

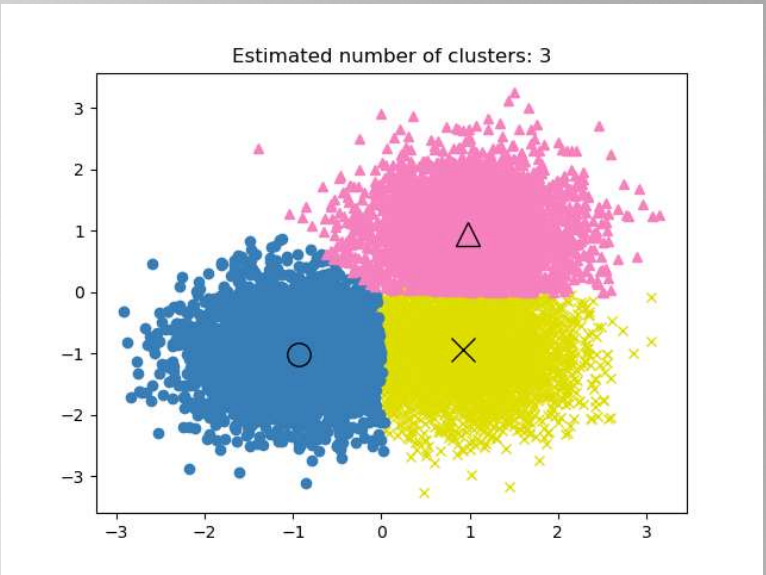
- Is great at separating clusters of high density versus clusters of low density within a given dataset.
- Is great for handling outliers within the dataset.

Disadvantages :

- While DBSCAN is great at separating high-density clusters from low-density clusters, DBSCAN struggles with clusters of similar density.
- Struggles with high dimensionality data. If given data with too many dimensions, DBSCAN suffers

Mean Shift

- Mean Shift is a non-parametric clustering algorithm that works by identifying dense regions in the data space.
- It is a centroid-based algorithm that iteratively shifts the centroids of clusters towards the maximum density of the data points until convergence
- Mean Shift is a mode-seeking algorithm, which means that it finds the modes (peaks) of the density distribution of the data.
- This is in contrast to centroid-based clustering algorithms, such as K-Means clustering, which find the centroids of the clusters.
- Mean Shift works by iteratively refining the positions of the data points, moving them towards the modes of the density distribution.
- This process is repeated until the data points converge to the modes of the density distribution.



Working of Mean-Shift Algorithm :

- First, start with the data points assigned to a cluster of their own
- Next, this algorithm will compute the centroids
- In this step, location of new centroids will be updated
- Now, the process will be iterated and moved to the higher density

  At last, it will be stopped once the centroids reach at position from where it cannot move further.

Advantages :

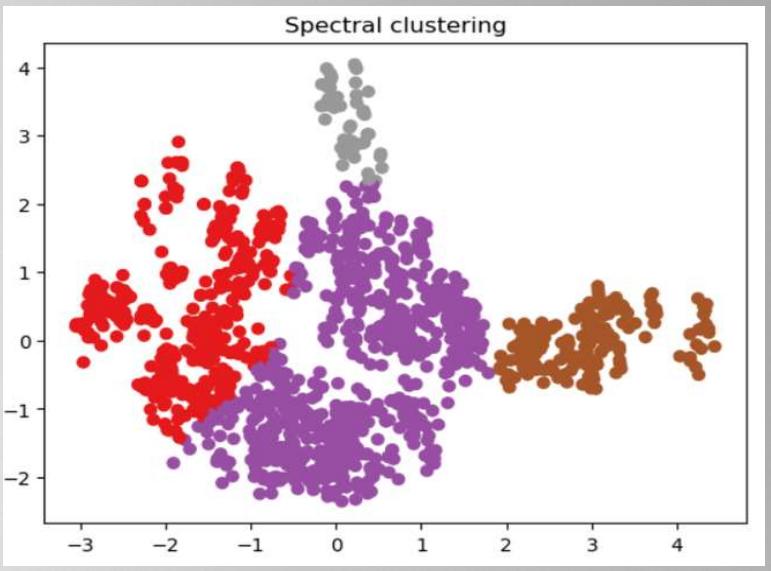
- The following are some advantages of Mean-Shift clustering algorithm −
- It does not need to make any model assumption as like in K-means or Gaussian mixture
-
- It can also model the complex clusters which have no convex shape.
- It only needs one parameter named bandwidth which automatically determines the number of clusters.
- There is no issue of local minima as like in K-means.
- No problem generated from outliers.

Disadvantages :

- The following are some disadvantages of Mean-Shift clustering algorithm −Mean-shift algorithm does not work well in case of high dimension, where number of clusters changes abruptly.
- We do not have any direct control on the number of clusters but in some applications, we need a specific number of clusters.
- It cannot differentiate between meaningful and meaningless modes.

Spectral Clustering

- 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 :

- 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.

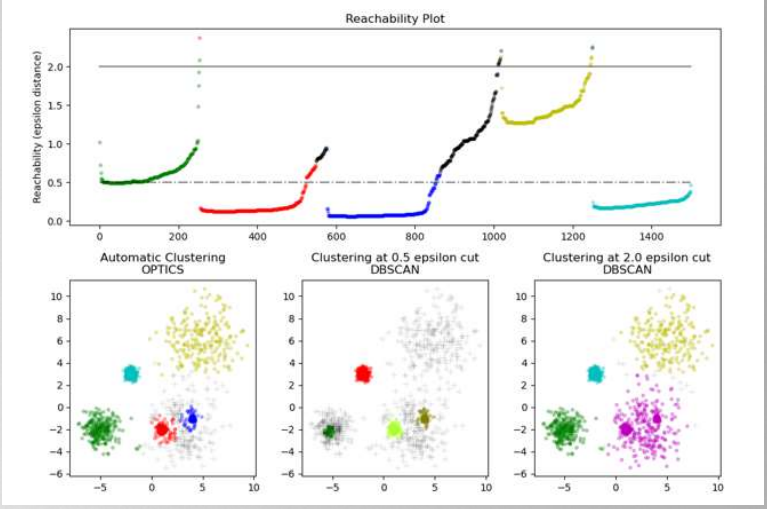
Disadvantages :

- 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

OPTICS

(Ordering Points To Identify the Clustering Structure)

- The OPTICS is a density-based clustering algorithm, similar to DBSCAN (but it can extract clusters of varying densities and shapes.
- It is useful for identifying clusters of different densities in large, high-dimensional datasets.
- 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.



OPTICS v/s DBSCAN :

- **Memory Cost :** The OPTICS clustering technique requires more memory as it maintains a priority queue (Min Heap) to determine the next data point which is closest to the point currently being processed in terms of Reachability Distance.
- It also requires more computational power because the nearest neighbor queries are more complicated than radius queries in DBSCAN.
- **Fewer Parameters:** The OPTICS clustering technique does not need to maintain the epsilon parameter and is only given in the above pseudo-code to reduce the time taken. This leads to the reduction of the analytical process of parameter tuning.
- **Handling varying densities:**
- OPTICS can handle varying densities by using the concept of reachability distance, which adapts to the local density of the data. This means that OPTICS can identify clusters of different sizes and shapes more effectively than DBSCAN in datasets with varying densities.
- **Cluster extraction:** While both OPTICS and DBSCAN can identify clusters, OPTICS produces a reachability distance plot that can be used to extract clusters at different levels of granularity.
- **Noise handling:** the OPTICS may be less then effective at identifying small clusters that are surrounded by noise points, as these clusters may be merged with the noise points in the reachability distance plot
- **Runtime complexity:** The runtime complexity of OPTICS is generally higher than that of DBSCAN, due to the use of a priority queue to maintain the reachability distances. However, recent research has proposed optimizations to reduce the computational complexity of OPTICS, making it more scalable for large datasets.

Advantages:

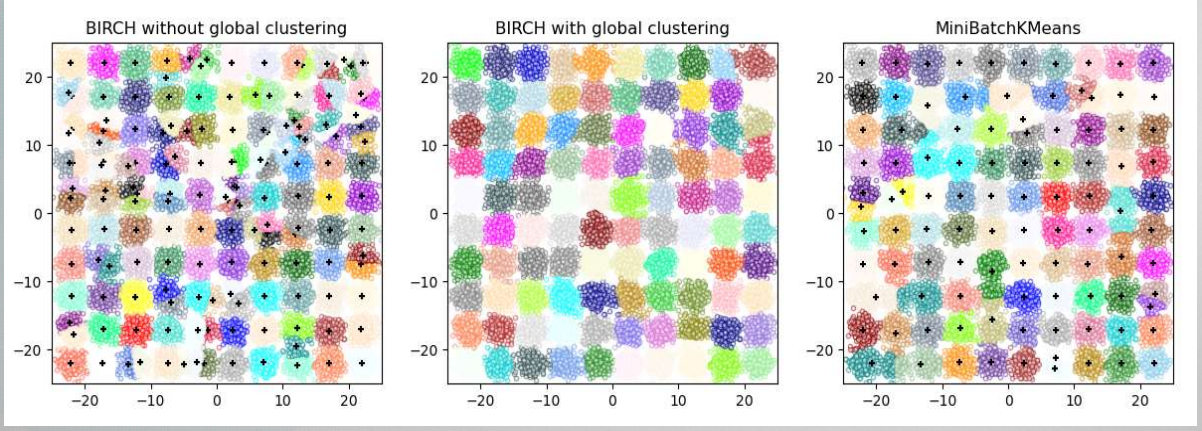
- OPTICS clustering doesn't require a predefined number of clusters in advance.
- Clusters can be of any shape, including non-spherical ones.
- Able to identify outliers(noise data)

Disadvantages:

- 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.

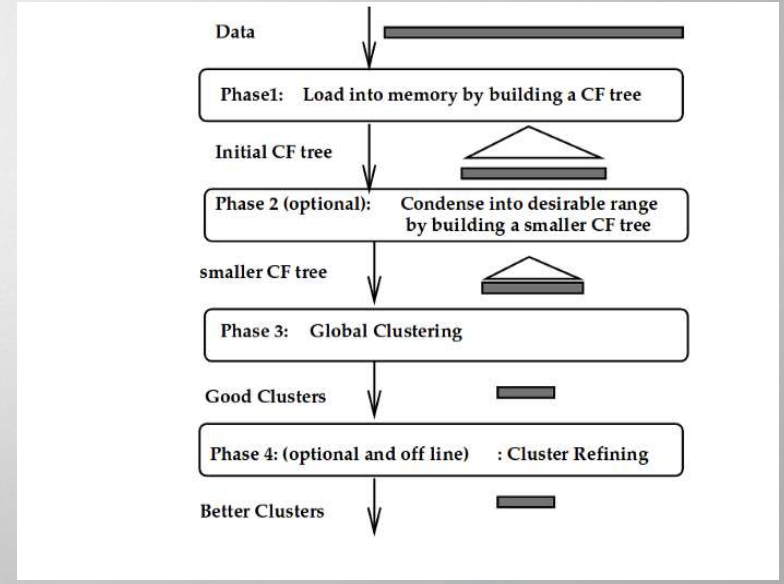
Birch

- Clustering algorithms like K-means clustering do not perform clustering very efficiently and it is difficult to process large datasets with a limited amount of resources (like memory or a slower CPU).
- So, regular clustering algorithms do not scale well in terms of running time and quality as the size of the dataset increases. This is where BIRCH clustering comes in



The BIRCH clustering algorithm consists of two stages:

- Building the CF Tree: BIRCH summarizes large datasets into smaller, dense regions called Clustering Feature (CF) entries. Formally, a Clustering Feature entry is defined as an ordered triple, (N, LS, SS) where 'N' is the number of data points in the cluster, 'LS' is the linear sum of the data points and 'SS' is the squared sum of the data points in the cluster. It is possible for a CF entry to be composed of other CF entries. Optionally, we can condense this initial CF tree into a smaller CF.
- Global Clustering: Applies an existing clustering algorithm on the leaves of the CF tree. A CF tree is a tree where each leaf node contains a sub-cluster.
- Every entry in a CF tree contains a pointer to a child node and a CF entry made up of the sum of CF entries in the child nodes. Optionally, we can refine these clusters.
- Global Clustering: Applies an existing clustering algorithm on the leaves of the CF tree. A CF tree is a tree where each leaf node contains a sub-cluster.
- Global Clustering: Applies an existing clustering algorithm on the leaves of the CF tree. A CF tree is a tree where each leaf node contains a sub-cluster.
- Every entry in a CF tree contains a pointer to a child node and a CF entry made up of the sum of CF entries in the child nodes. Optionally, we can refine these clusters.



Advantages:

- 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:

- 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.