Unsupervised Learning:

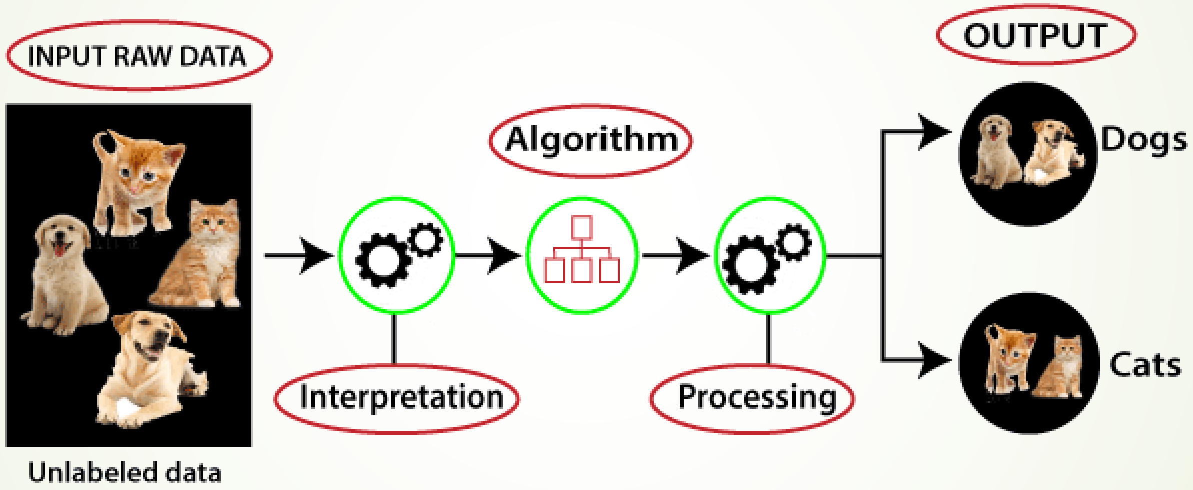
As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.

Why use Unsupervised Learning?

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

Working of Unsupervised Learning



Supervised vs Unsupervised Learning

In the table below, we've compared some of the key differences between unsupervised and supervised learning:

	Supervised Learning	Unsupervised learning
Objective	To approximate a function that maps inputs to outputs based out example input-output pairs.	To build a concise representation of the data and generate imaginative content from it.
Accuracy	Highly accurate and reliable.	Less accurate and reliable.
Complexity	Simpler method.	Computationally complex.
Classes	Number of classes is *known.*	Number of classes is *unknown.*
Output	A desired output value (also called the supervisory signal).	No corresponding output values.

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:

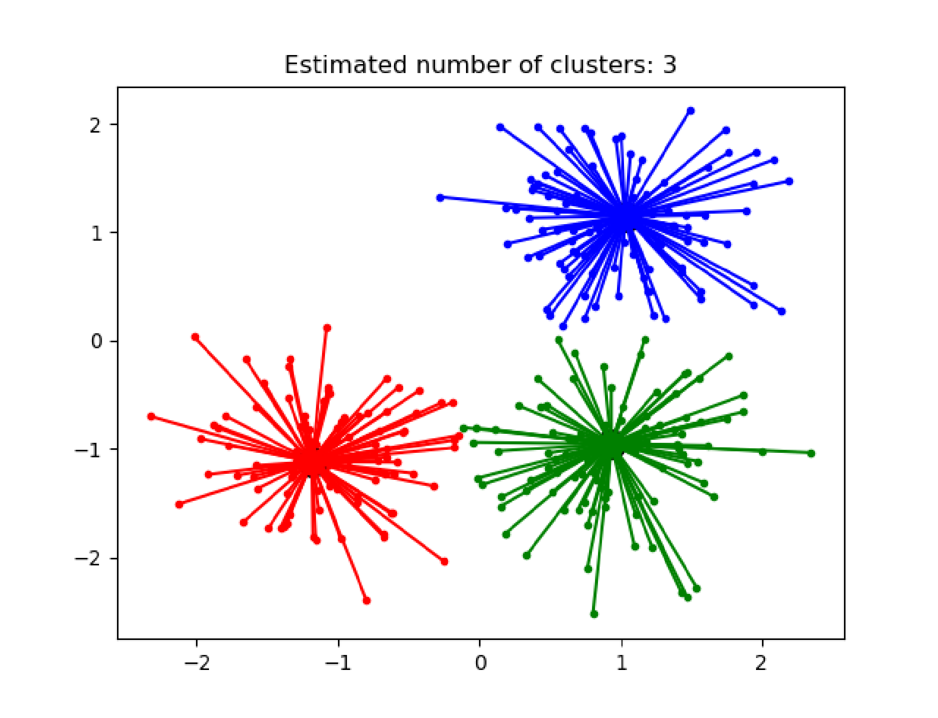
- **Clustering**: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.
- **Association**: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

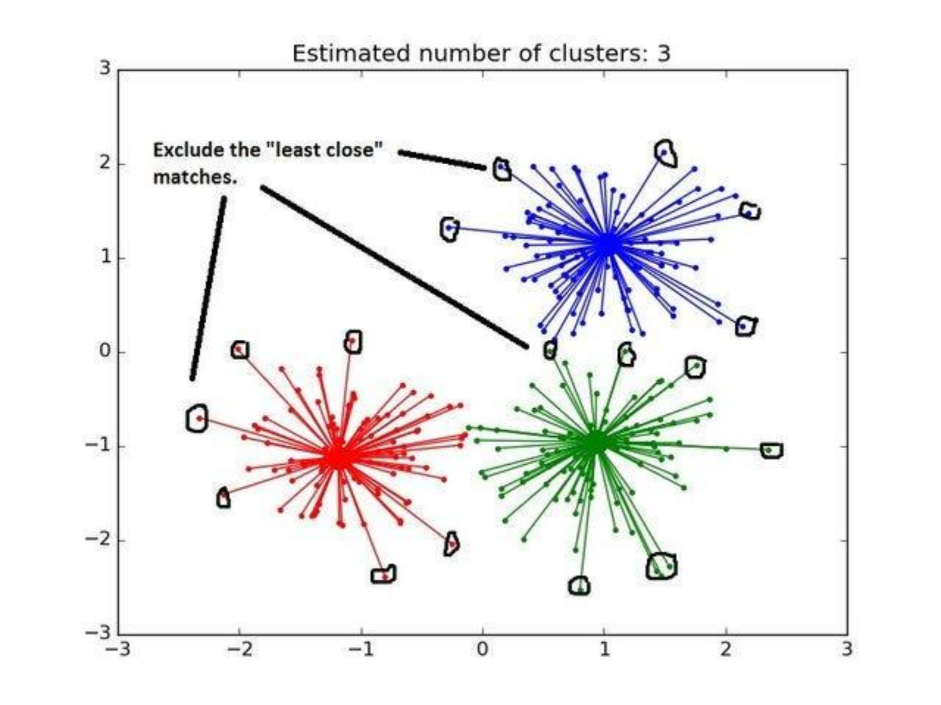
Affinity Propagation:

- Affinity propagation is an clustering algorithm based on the concept of “Message passing” between the data points. Unlike clustering algorithm’s such as k-means or k-medoids, Affinity propagation doesn’t require to number of cluster’s to determined or estimated before running the algorithm. Similar to k-medoids affinity propagation finds ‘Exemplar's’ numbers of input set that representative of clusters.

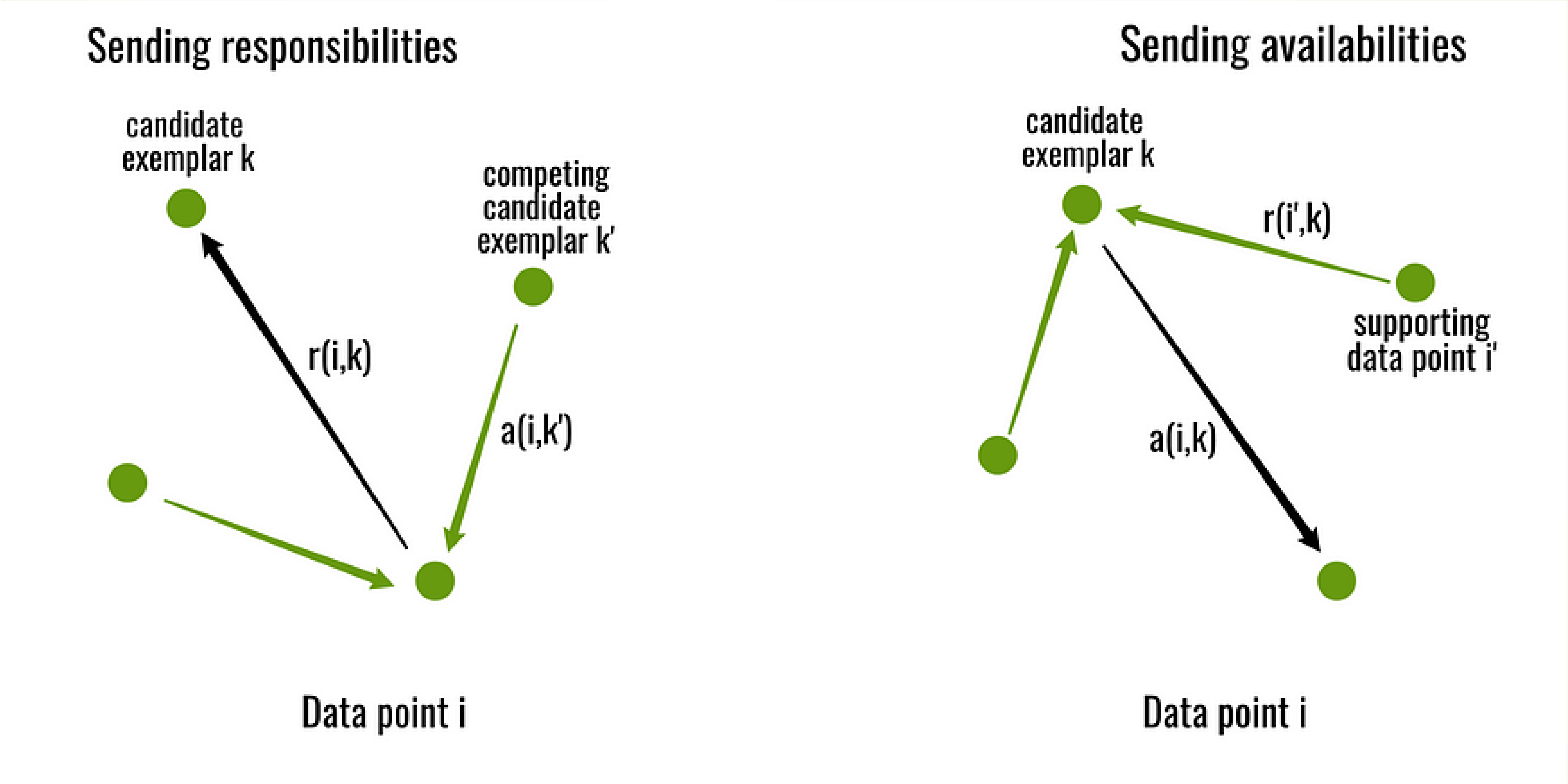
Why do we affinity propagation ?

- The inventors of affinity propagation showed it is better for certain computer vision and computational biology tasks, e.g. clustering of pictures of human faces and identifying regulated transcripts, than k-means, even when k-means was allowed many random restarts and initialized using PCA.





Principal behind affinity propagation:



Advantages:

- Affinity Propagation (AP) clustering does not need to set the number of clusters, and has advantages on **efficiency and accuracy**, but is not suitable for large-scale data clustering. ... The data set to be clustered is firstly divided into several subsets, each of which can be efficiently clustered by AP algorithm.

Drawbacks:

1. It is hard to know the value of the parameter preferences which can yield an optimal **clustering** solution.
2. When oscillations occur, AP cannot automatically eliminate them.

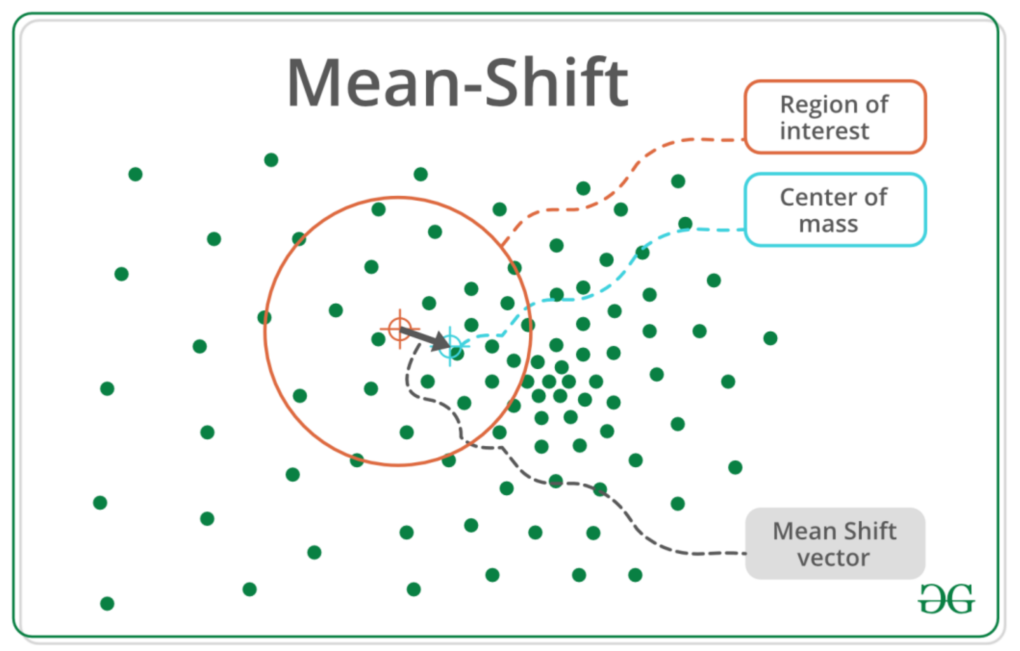
Mean Shift

- Mean Shift algorithm basically 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.
- Unlike the popular K-Means cluster algorithm, mean-shift does not require specifying the number of clusters in advance. The number of clusters is determined by the algorithm with respect to the data.

Mean Shift Steps

- On a high level, Mean Shift works as follows:

1. Create a sliding window/cluster for each data-point
2. Each of the sliding windows is shifted towards higher density regions by shifting their centroid (center of the sliding window) to the data-points' mean within the sliding window. This step will be repeated until no shift yields a higher density (number of points in the sliding window)
3. Selection of sliding windows by deleting overlapping windows. When multiple sliding windows overlap, the window containing the most points is preserved, and the others are deleted.
4. Assigning the data points to the sliding window in which they reside.



Advantages

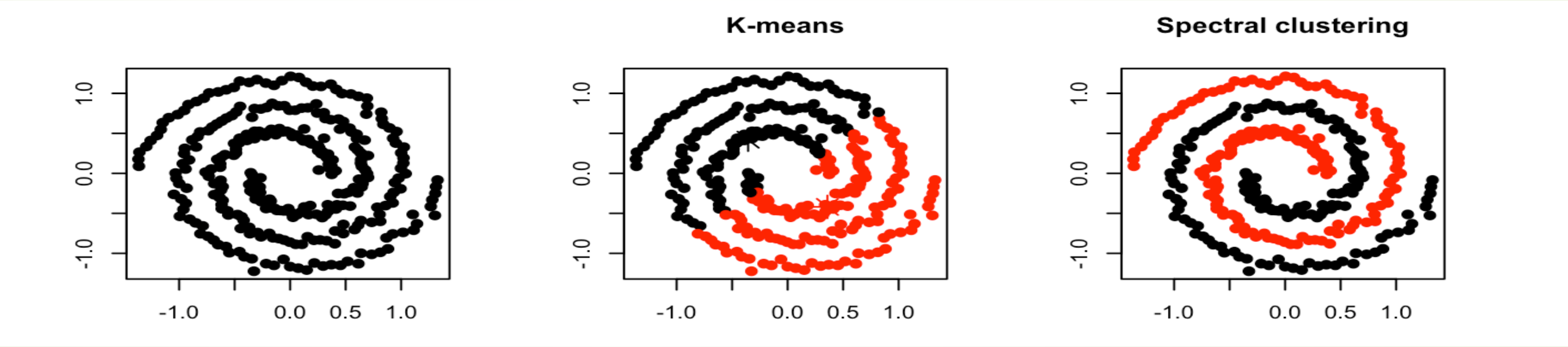
- Mean Shift is a simple cluster method that works very well on spherical-shaped data. Furthermore, it automatically selects the number of clusters contrary to other clustering algorithms like KMeans. Also, the output of Mean Shift is not dependent on the initialization since, at the start, each point is a cluster.

- Drawbacks

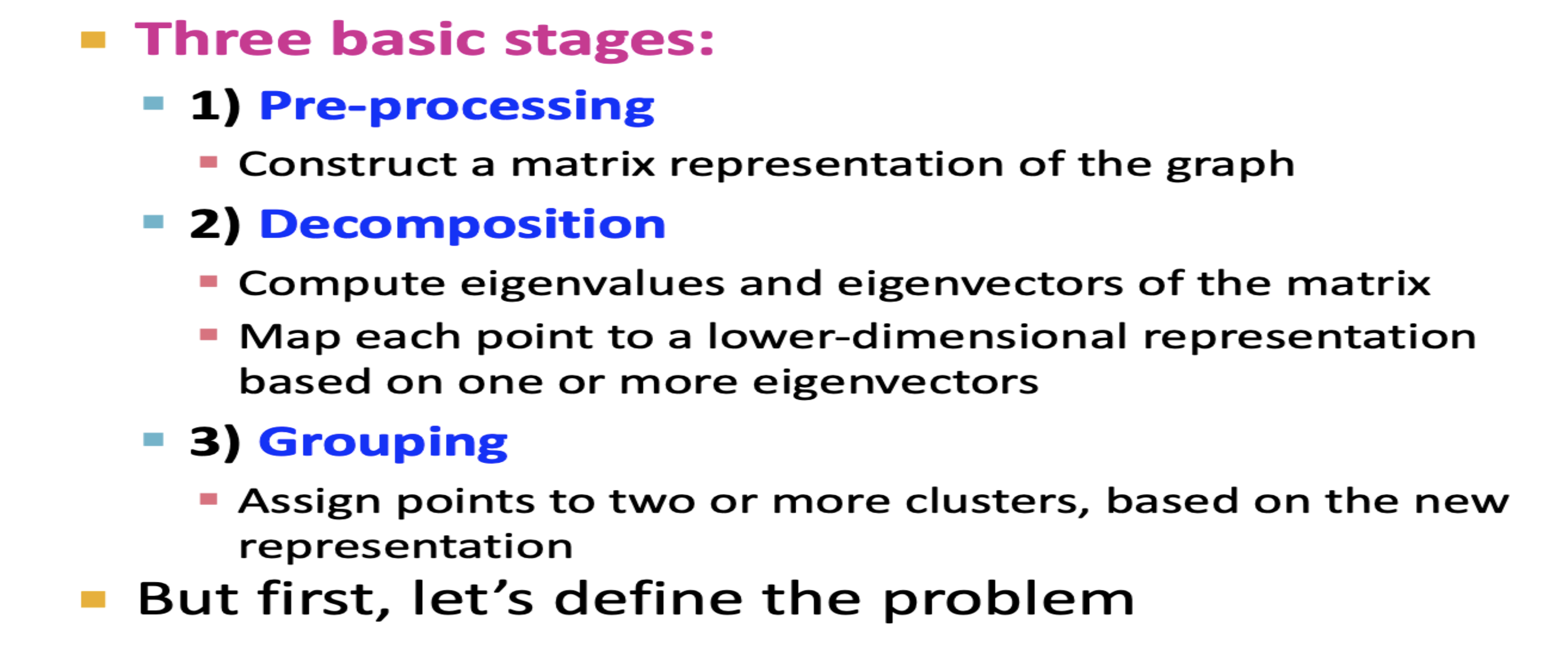
- The algorithm is not highly scalable, as it requires multiple nearest neighbor searches during its execution. It has a complexity of O(n^2). Furthermore, manually choosing the bandwidth can be non-trivial, and selecting a wrong value can lead to bad results.

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.



Spectral Clustering Algorithm:



- **Three basic stages:**
  - **1) Pre-processing**
    - Construct a matrix representation of the graph
  - **2) Decomposition**
    - Compute eigenvalues and eigenvectors of the matrix
    - Map each point to a lower-dimensional representation based on one or more eigenvectors
  - **3) Grouping**
    - Assign points to two or more clusters, based on the new representation
- But first, let’s define the problem

Advantages of Spectral Clustering:

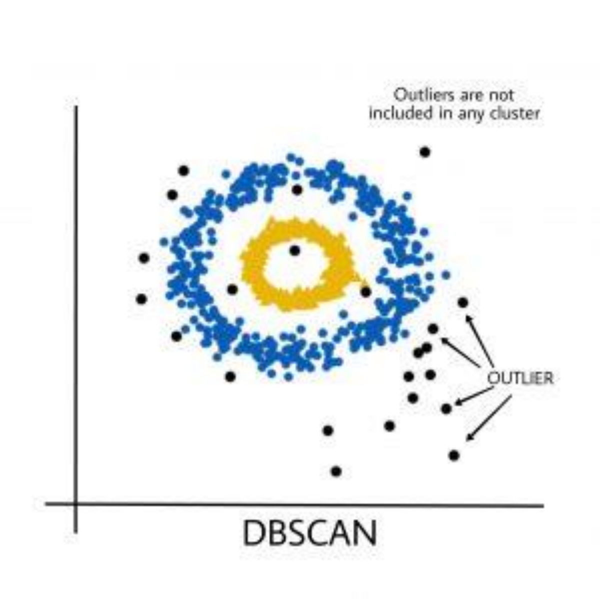
1. Scalability: Spectral clustering can handle large datasets and high-dimensional data, as it reduces the dimensionality of the data before clustering.
2. Flexibility: Spectral clustering can be applied to non-linearly separable data, as it does not rely on traditional distance-based clustering methods.
3. 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 of Spectral Clustering:

1. Complexity: Spectral clustering can be computationally expensive, especially for large datasets, as it requires the calculation of eigenvectors and eigenvalues.
2. 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.

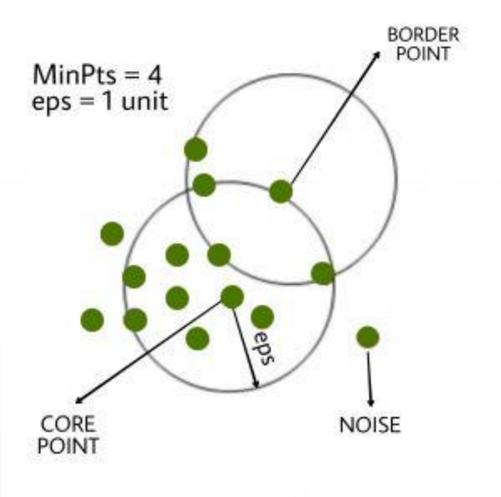
Density-Based Spatial Clustering Of Applications With Noise (DBSCAN)

- Clusters are dense regions in the data space, separated by regions of the lower density of points. The ***DBSCAN algorithm*** is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.



Why DBSCAN?

- Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.
- ***In this algorithm, we have 3 types of data points.***
  Core Point*: A point is a core point if it has more than MinPts points within eps.*
  Border Point*: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.*
  Noise or outlier*: A point which is not a core point or border point.*



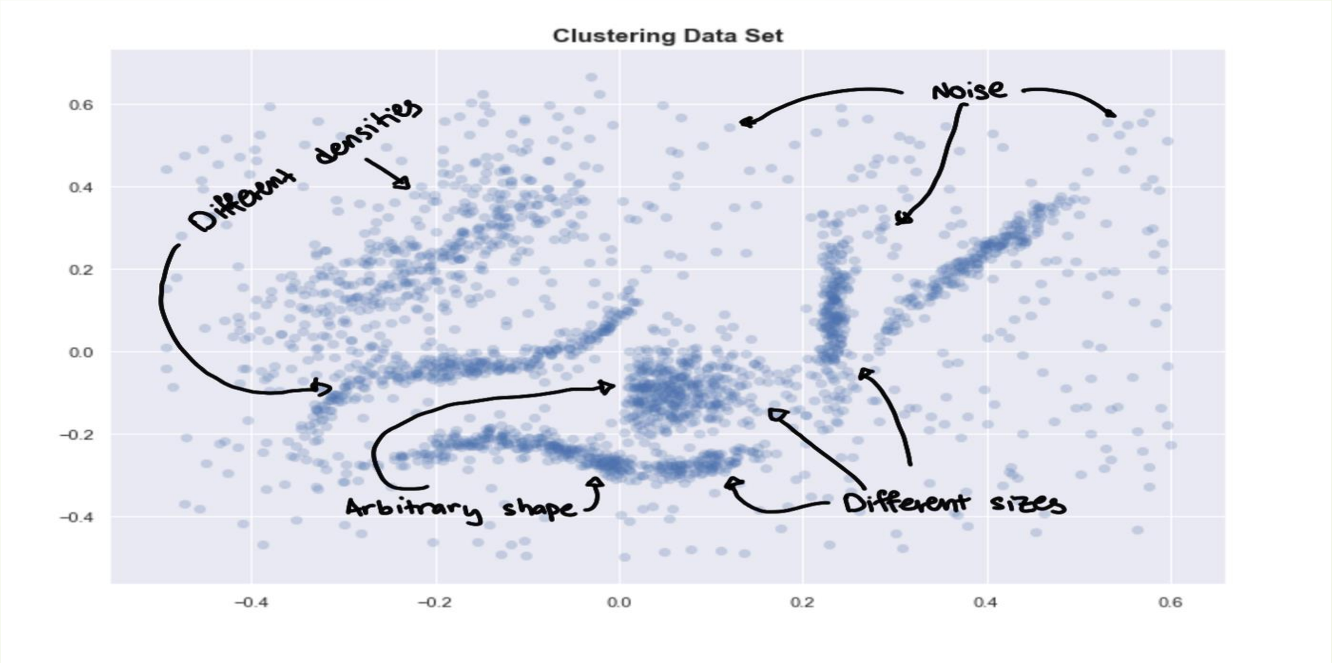
Steps Used In DBSCAN Algorithm

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. 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*.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

1. **Advantages**:
   1. **Outlier Detection**: DBSCAN excels at identifying outliers. It can distinguish noise points from actual clusters, making it useful for datasets with noisy or irregular data.
   2. **No Need for Specifying the Number of Clusters**: Unlike some other clustering algorithms (such as k-means), DBSCAN doesn’t require you to specify the number of clusters beforehand.
   3. **Ability to Identify Irregular Shapes**: DBSCAN can handle clusters with arbitrary shapes, which is particularly valuable when dealing with non-spherical or overlapping clusters.
   4. **Parameter Tuning Intuition**: If you understand your dataset well, you can set the parameters (such as the neighborhood radius and minimum points) based on your knowledge.
2. **Disadvantages**:
   1. **Sensitive to Parameters**: DBSCAN’s performance depends on the choice of parameters, especially the neighborhood radius (ε) and the minimum number of points (minPoints). Selecting appropriate values can be challenging.
   2. **Border Points Ambiguity**: When a point lies on the border between two clusters, DBSCAN may struggle to assign it to the correct cluster.
   3. **Difficulty with Varying Densities**: If your data has clusters with significantly different densities, DBSCAN might not perform well.

HDBSCAN - **Hierarchical Density-Based Spatial Clustering**

- HDBSCAN - **Hierarchical Density-Based Spatial Clustering** of Applications with Noise. Performs DBSCAN over varying epsilon values and integrates the result to find a clustering that gives the best stability over epsilon. This allows HDBSCAN to find clusters of varying densities (unlike DBSCAN), and be more robust to parameter selection.



DBSCAN or HDBSCAN is better option? and why?

- The main disavantage of **DBSCAN is that is much more prone to noise**, which may lead to false clustering. On the other hand, **HDBSCAN focus on high density** clustering, which reduces this noise clustering problem and allows a **hierarchical clustering** based on a decision tree approach.

How HDBSCAN works:

1. Transform the data: The algorithm first transforms the data space based on its density. This helps identify areas with higher concentrations of data points.
2. Build a minimum spanning tree: A graph is created connecting data points within a certain distance threshold. This forms the basis for identifying clusters.
3. Construct a cluster hierarchy: Connected components in the graph are identified, forming a hierarchical structure of potential clusters.
4. Condense the hierarchy: Based on the minimum cluster size parameter, smaller clusters are merged into larger, more stable ones.
5. Extract stable clusters: The final clustering solution is extracted from the condensed hierarchy, ensuring identified clusters are robust to parameter variations.

Functions of HDBSCAN:

- **Uncovering cluster structure**: HDBSCAN can effectively reveal the underlying cluster structure in a dataset, even if it contains clusters of different densities and noise points.
- **Pattern recognition:** By grouping similar data points together, HDBSCAN helps identify patterns and trends within the data, facilitating further analysis and interpretation.
- **Data segmentation**: It can be used to segment a dataset into meaningful groupings, making it easier to analyze and model different segments of the data.
- **Dimensionality reduction**: HDBSCAN can be used as a dimensionality reduction technique, by grouping data points that lie close together in the high-dimensional space. This can simplify further analysis and visualization.
- **Outlier detection**: While the algorithm focuses on clustering, it also identifies data points that are significantly different from any cluster, potentially representing outliers or anomalous data.

Advantages of HDBSCAN:

- Robust to varying densities: Unlike DBSCAN, HDBSCAN works well with datasets that have clusters of different densities. This makes it more flexible and adaptable to real-world data.
- Noise tolerant: Similar to DBSCAN, HDBSCAN can effectively handle outliers and noisy points, leading to cleaner and more accurate clusters.
- Intuitive parameters: With only the minimum cluster size to tune, HDBSCAN is easier to use for users unfamiliar with clustering algorithms. This reduces the need for extensive parameter tweaking.
- Scalable: The algorithm can efficiently handle large datasets, making it suitable for big data applications.
- Hierarchical clustering: HDBSCAN provides a hierarchical view of the data, allowing you to explore different levels of granularity within the clusters. This can be valuable for understanding the relationships between data points in more detail.
- Fast performance: HDBSCAN can be faster than DBSCAN, especially for large datasets.

Disadvantages of HDBSCAN:

- **Less control compared to DBSCAN:** Users have less control over the specific density parameters compared to DBSCAN. While this simplifies things, it may not be ideal for fine-tuning the clustering for specific scenarios.
- **Can be sensitive to the minimum cluster size**: Improperly choosing the minimum cluster size can lead to under- or over-fitting of the data, affecting the accuracy of the results.
- Not as well-established as DBSCAN: HDBSCAN is a newer algorithm compared to DBSCAN, and it may not be as widely adopted or have as extensive documentation and support available.
- **Limited interpretability**: While the hierarchical structure offers some insights, understanding the exact reasoning behind the clustering decisions can be less straightforward compared to other algorithms.

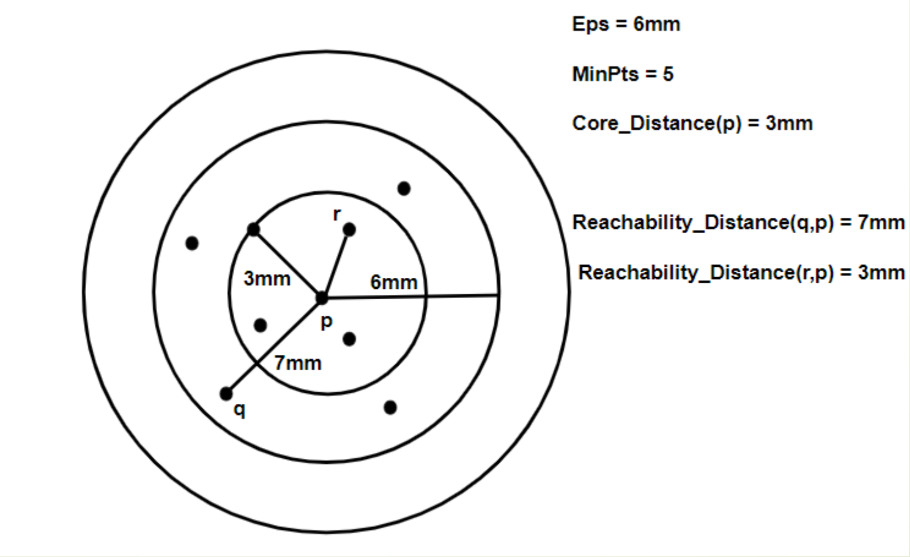
Ordering Points To Identify the Clustering Structure (OPTICS):

- 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 algorithm follows these main steps:

- It takes several parameters including the minimum density threshold (Eps), the number of nearest neighbors to consider (min_samples), and a reachability distance cutoff (xi).
- They are:-

1. **Core Distance:** It is the minimum value of radius required to classify a given point as a core point. If the given point is not a Core point, then it’s Core Distance is undefined.
2. **Reachability Distance:** It is defined with respect to another data point q(Let). The Reachability distance between a point p and q is the maximum of the Core Distance of p and the Euclidean Distance(or some other distance metric) between p and q. Note that The Reachability Distance is not defined if q is not a Core point.


Eps = 6mm
MinPts = 5
Core_Distance(p) = 3mm
Reachability_Distance(q,p) = 7mm
Reachability_Distance(r,p) = 3mm
r
3mm
6mm
p
7mm
q

Advantages:

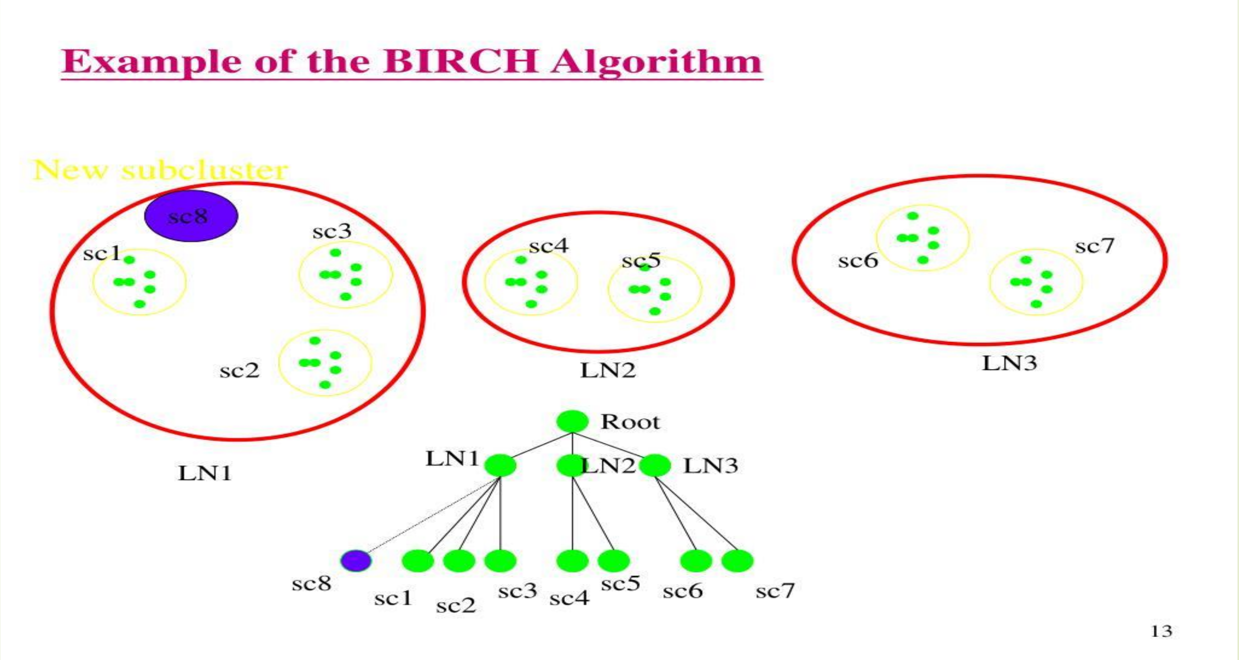
1. **Robustness against Noise or Outliers**: OPTICS is resistant to noise or outliers in the data, making it suitable for real-world datasets with irregularities.
2. **Handling Nonlinear Clusters**: It can handle clusters of varying shapes and sizes, including nonlinear ones.
3. **No Need to Specify Cluster Count**: Unlike some other clustering algorithms, OPTICS doesn’t require you to specify the number of clusters beforehand.

Disadvantages:

1. **Choosing the Proper ε Value**: Selecting the appropriate value for the epsilon (ε) parameter can be challenging, especially if the data and scale are not well understood.
2. **Complexity**: While OPTICS is powerful, it may be computationally expensive for very large datasets.

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 the large dataset that retains as much information as possible**.
- This smaller summary is then clustered instead of clustering the larger dataset. BIRCH does not directly cluster the dataset, but clusters the dataset first in small summaries, then after small summaries get clustered. BIRCH is often used with other clustering algorithms.



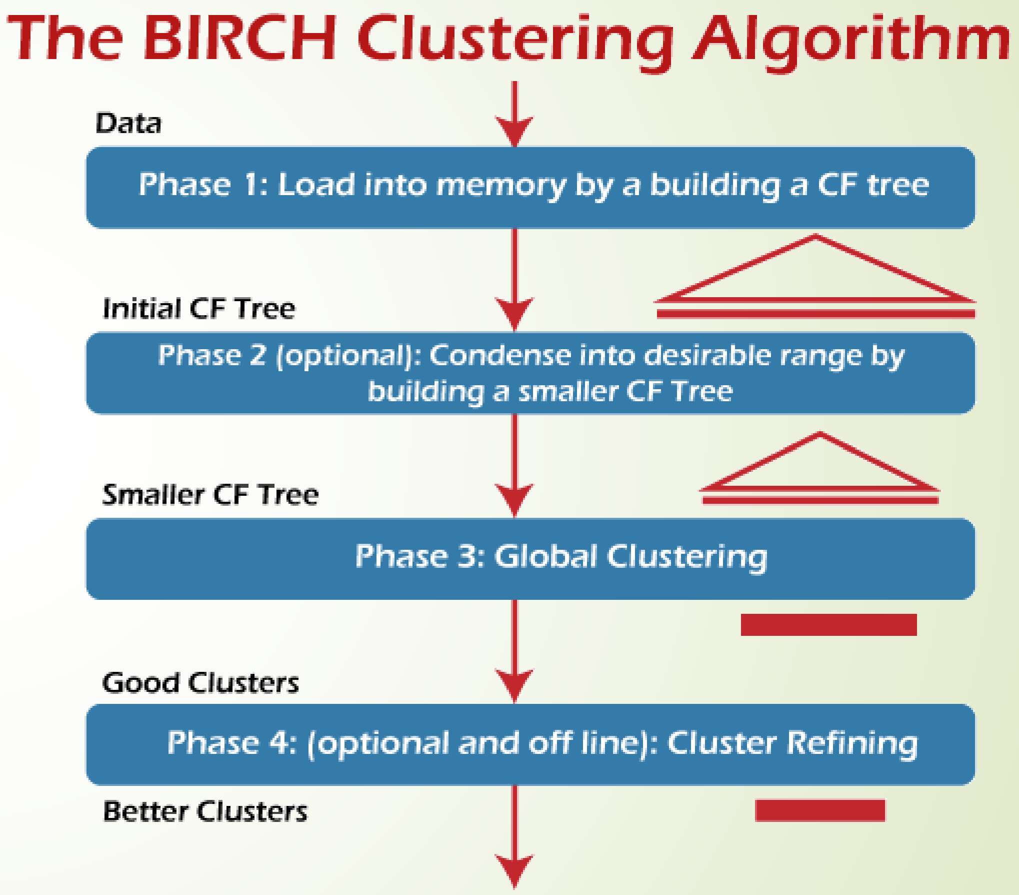
Example of the BIRCH Algorithm

BIRCH clustering algorithm consists of two stages:

1. **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. A CF entry can be composed of other CF entries. Optionally, we can condense this initial CF tree into a smaller CF.
2. **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.

- Due to this two-step process, BIRCH is also called ***Two-Step Clustering.***

- The tree structure of the given data is built by the BIRCH algorithm called the Clustering feature tree (CF tree). This algorithm is based on the CF (clustering features) tree. In addition, this algorithm uses a tree-structured summary to create clusters.



The BIRCH Clustering Algorithm

Data

Phase 1: Load into memory by a building a CF tree

Initial CF Tree

Phase 2 (optional): Condense into desirable range by building a smaller CF Tree

Smaller CF Tree

Phase 3: Global Clustering

Good Clusters

Phase 4: (optional and off line): Cluster Refining

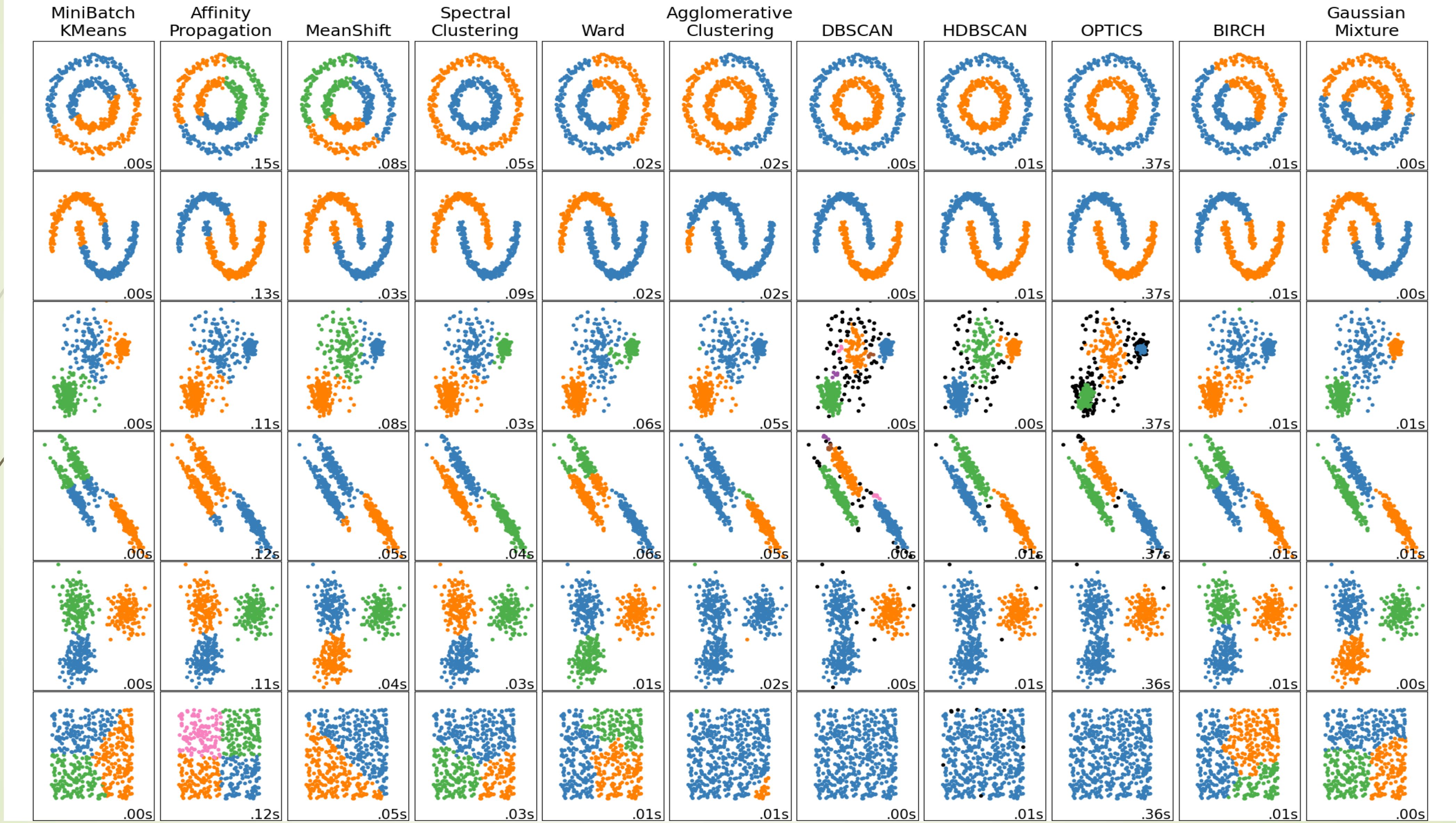
Better Clusters

Advantages of Birch Wood:

1. **Durability**: Birch wood is known for its durability, making it suitable for various applications.
2. **Workability**: It’s easy to work with both machine and hand tools.
3. **Finishing Qualities**: Birch finishes well, allowing for attractive surfaces.
4. **Light Appearance**: Birch’s natural light color gives a sense of spaciousness, making it ideal for indoor furniture and flooring.
5. **Shock Resistance**: Birch is medium-hardwood and shock-resistant, making it suitable for interior furniture.

Disadvantages of Birch Wood:

1. **Durability Issues**: Birch is less durable than some other hardwoods. It’s prone to rot, insect attacks, and decay.
2. **Spalting**: Birch can exhibit “spalting,” caused by certain fungi, which affects its appearance.
3. **Volume Loss During Curing**: Birch wood loses about 15% of its volume during drying, which can lead to warping if not properly handled


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
HDBSCAN
OPTICS
BIRCH
Gaussian Mixture
.00s .15s .08s .05s .02s .02s .00s .01s .37s .01s .00s
.00s .13s .03s .09s .02s .02s .00s .01s .37s .01s .00s
.00s .11s .08s .03s .06s .05s .00s .00s .37s .01s .01s
.00s .12s .05s .04s .06s .05s .00s .01s .37s .01s .01s
.00s .11s .04s .03s .01s .02s .00s .01s .36s .01s .00s
.00s .12s .05s .03s .01s .01s .00s .01s .36s .01s .00s