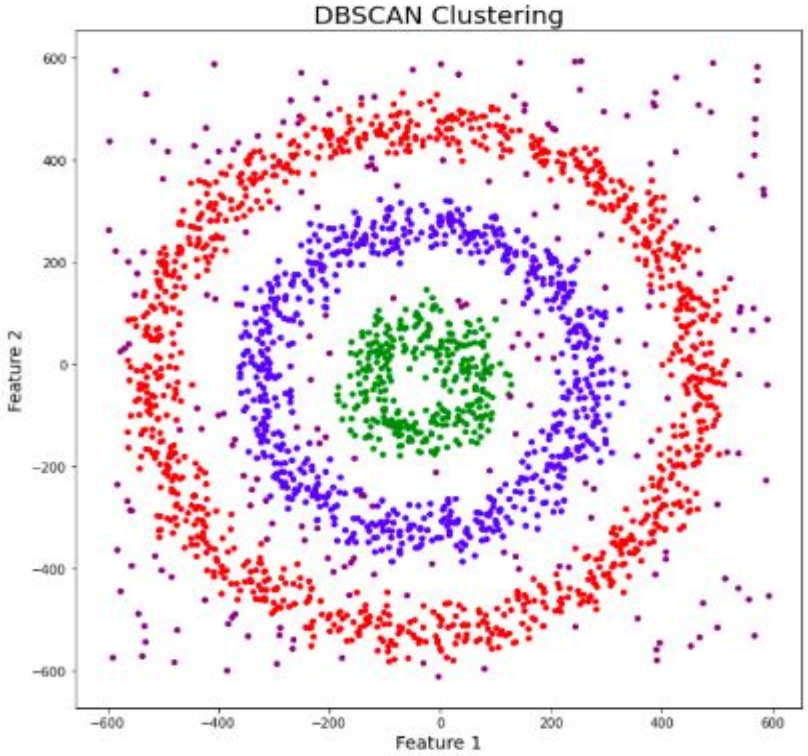
Clustering Algorithms

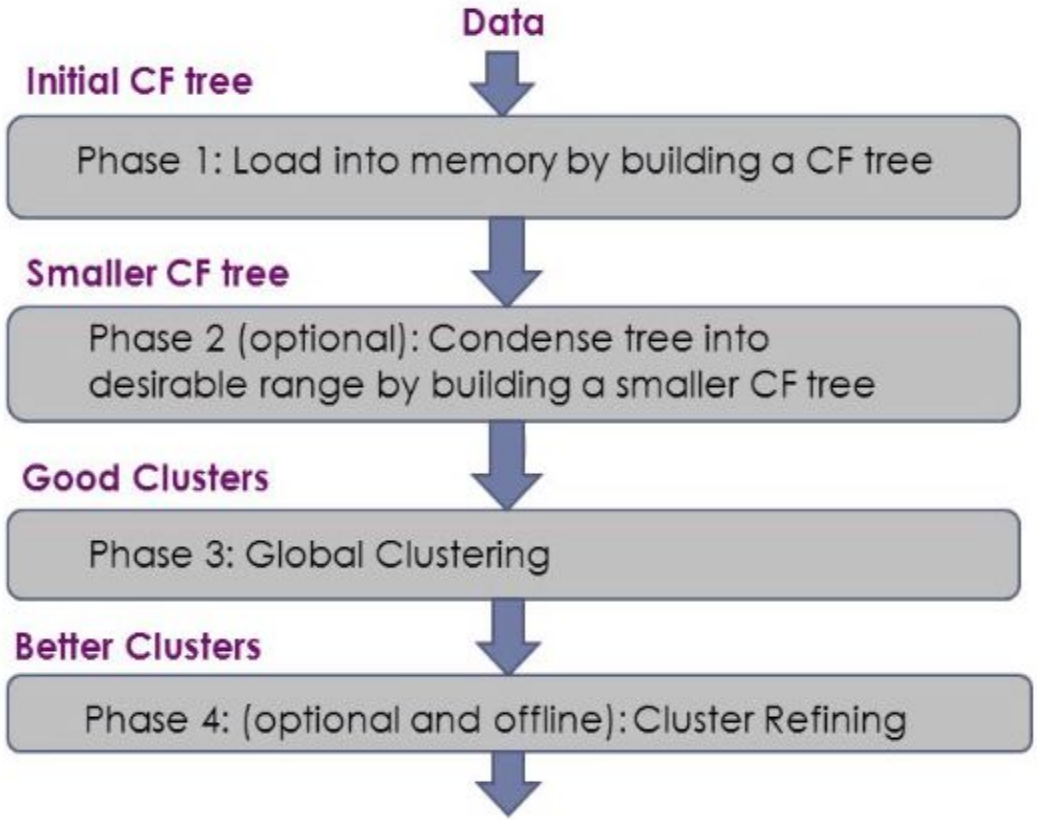
DBSCAN clustering algorithm

- DBSCAN stands for **density-based spatial clustering of applications with noise**. It's a density-based clustering algorithm, unlike k-means.
- This is a good algorithm for finding outliners in a data set. It finds arbitrarily shaped clusters based on the density of data points in different regions. It separates regions by areas of low-density so that it can detect outliers between the high-density clusters.
- This algorithm is better than k-means when it comes to working with oddly shaped data.


DBSCAN Clustering
Feature 2
Feature 1
600
400
200
0
−200
−400
−600
−600
−400
−200
0
200
400
600

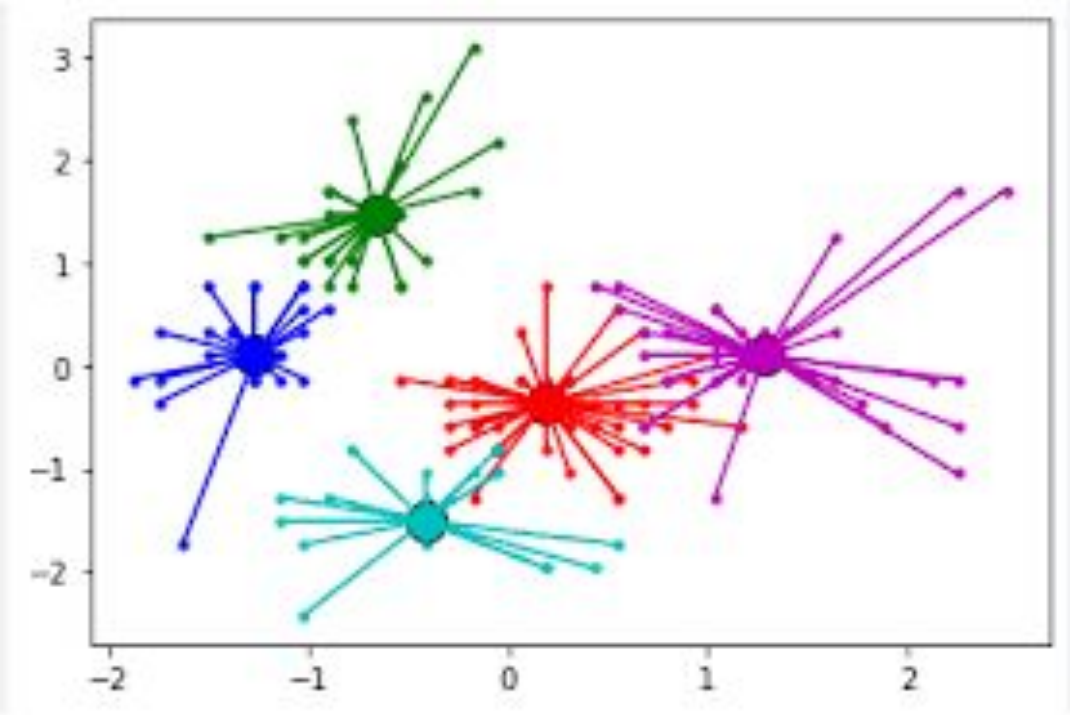
BIRCH algorithm

- The **Balance Iterative Reducing and Clustering using Hierarchies** (BIRCH) algorithm works better on large data sets than the k-means algorithm.
- It breaks the data into little summaries that are clustered instead of the original data points. The summaries hold as much distribution information about the data points as possible.
- This algorithm is commonly used with other clustering algorithm because the other clustering techniques can be used on the summaries generated by BIRCH.


Data
Initial CF tree
Phase 1: Load into memory by building a CF tree
Smaller CF tree
Phase 2 (optional): Condense tree into desirable range by building a smaller CF tree
Good Clusters
Phase 3: Global Clustering
Better Clusters
Phase 4: (optional and offline): Cluster Refining

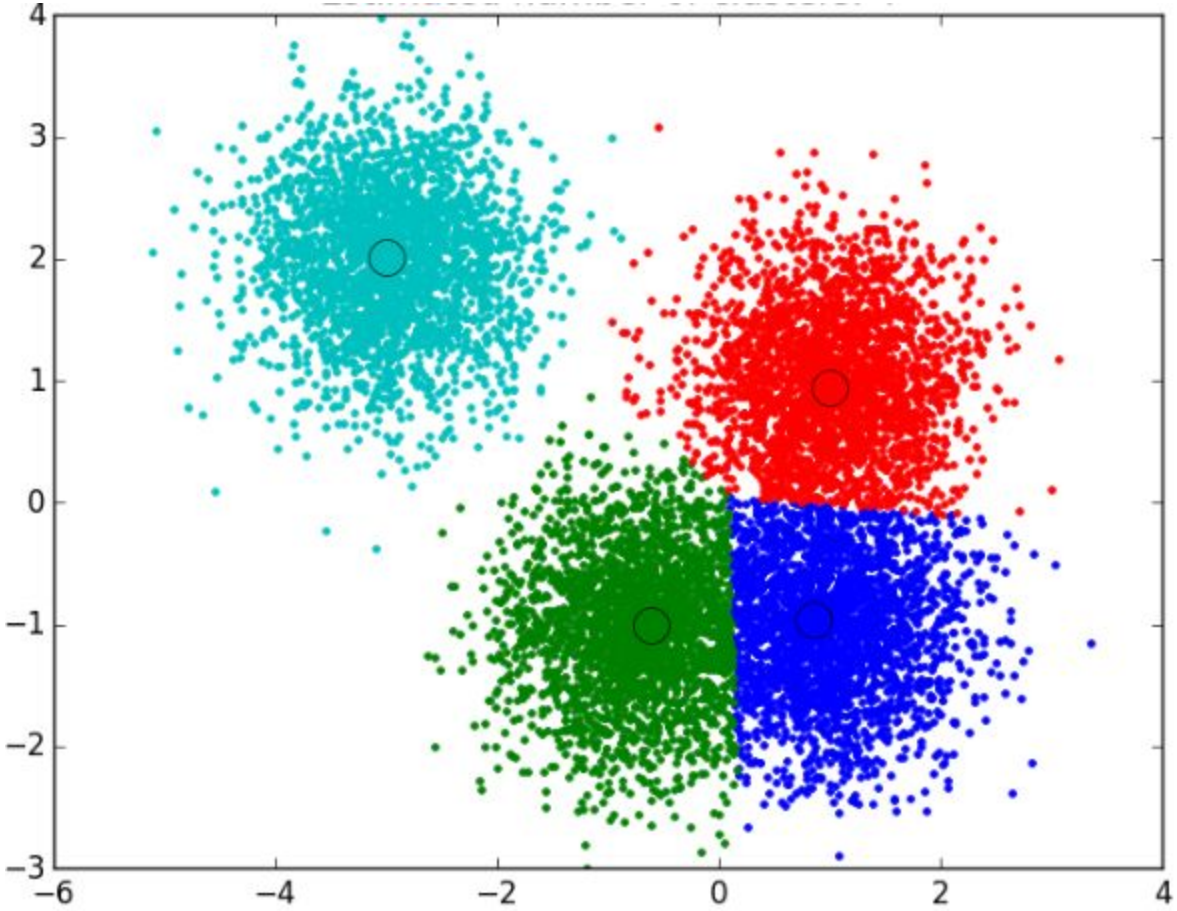
Affinity Propagation clustering algorithm

- This clustering algorithm is completely different from the others in the way that it clusters data.
- Each data point communicates with all of the other data points to let each other know how similar they are and that starts to reveal the clusters in the data. You don't have to tell this algorithm how many clusters to expect in the initialization parameters.
- As messages are sent between data points, sets of data called *exemplars* are found and they represent the clusters.


3
2
1
0
−1
−2
−2
−1
0
1
2

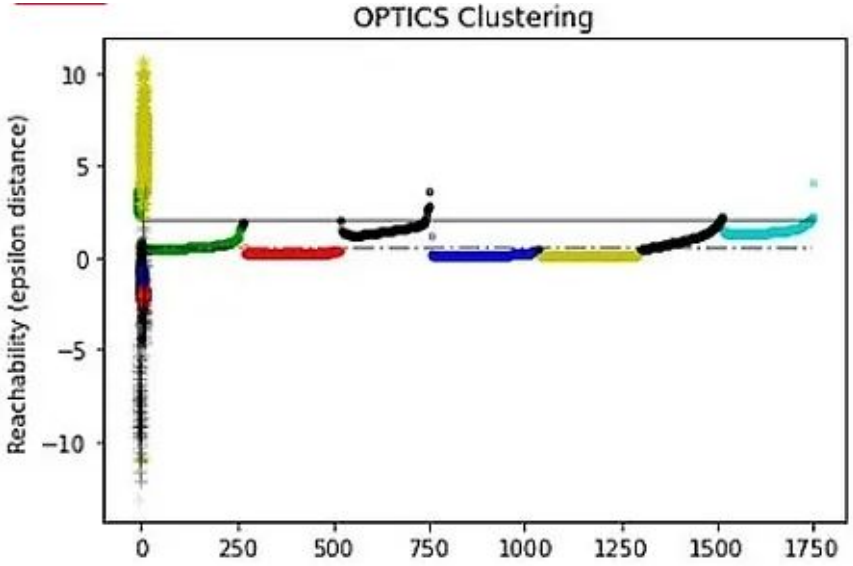
Mean-Shift clustering algorithm

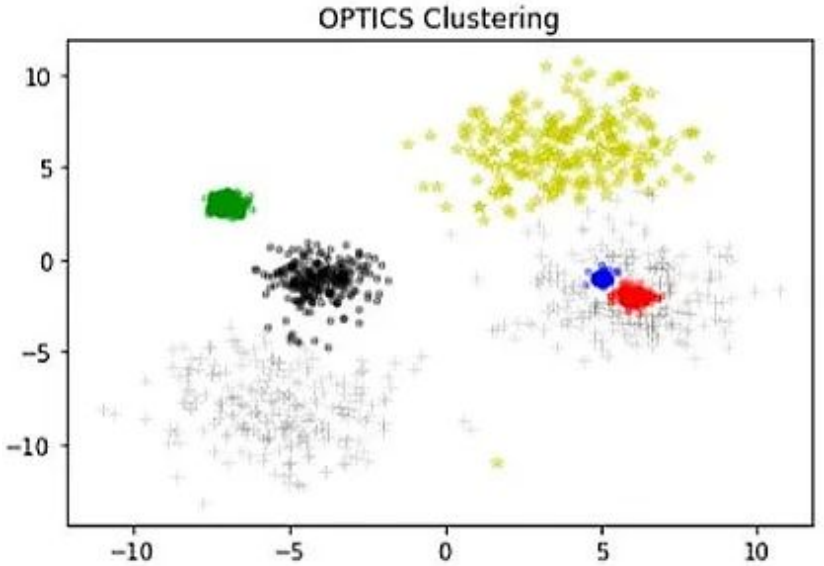
- Mean-shift is similar to the BIRCH algorithm because it also finds clusters without an initial number of clusters being set.
- It works by iterating over all of the data points and shifts them towards the mode. The mode in this context is the high density area of data points in a region.
- It will go through this iterative process with each data point and move them closer to where other data points are until all data points have been assigned to a cluster.
-


4
3
2
1
0
−1
−2
−3
−6
−4
−2
0
2
4

OPTICS algorithm

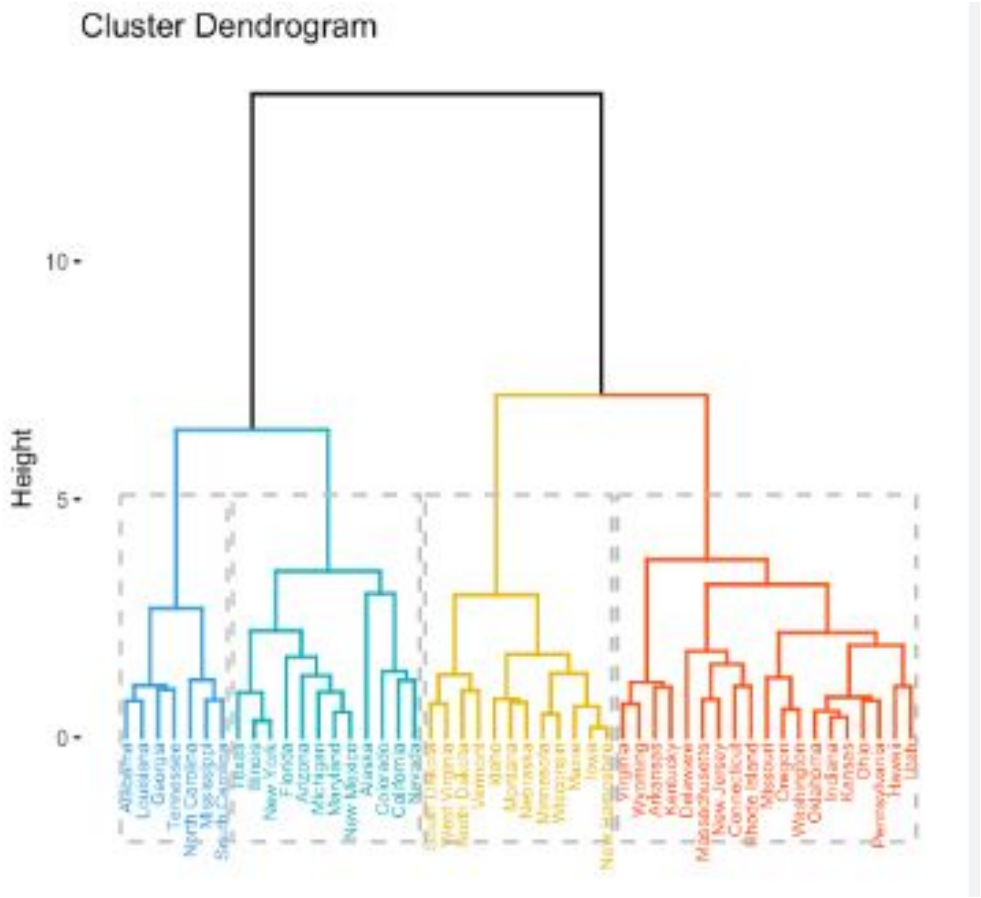
- OPTICS stands for Ordering Points to Identify the Clustering Structure. It's a density-based algorithm similar to DBSCAN, but it's better because it can find meaningful clusters in data that varies in density. It does this by ordering the data points so that the closest points are neighbors in the ordering.
- This makes it easier to detect different density clusters. The OPTICS algorithm only processes each data point once, similar to DBSCAN (although it runs slower than DBSCAN). There's also a special distance stored for each data point that indicates a point belongs to a specific cluster.


OPTICS Clustering
Reachability (epsilon distance)
10
5
0
−5
−10
0
250
500
750
1000
1250
1500
1750


OPTICS Clustering
10
5
0
−5
−10
−10
−5
0
5
10

Agglomerative Hierarchy clustering algorithm

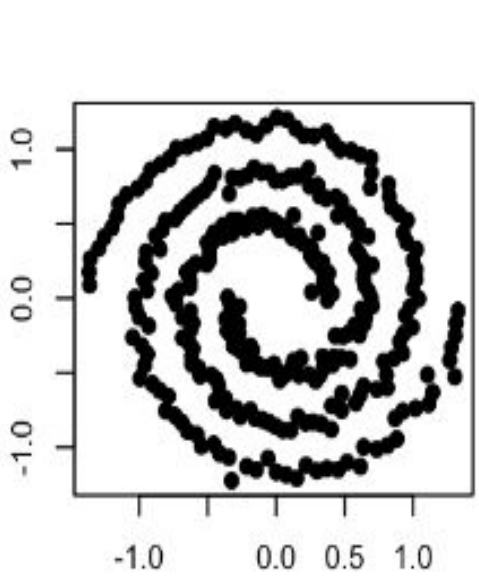
- This is the most common type of hierarchical clustering algorithm. It's used to group objects in clusters based on how similar they are to each other.
- This is a form of bottom-up clustering, where each data point is assigned to its own cluster. Then those clusters get joined together.
- At each iteration, similar clusters are merged until all of the data points are part of one big root cluster.The end result looks like a dendrogram so that you can easily visualize the clusters when the algorithm finishes.


Cluster Dendrogram
Height
10
5
0
Alabama
Louisiana
Georgia
Tennessee
North Carolina
Mississippi
South Carolina
Texas
Illinois
New York
Florida
Arizona
Michigan
Maryland
New Mexico
Alaska
Colorado
California
Nevada
Rhode Island
West Virginia
North Dakota
Vermont
Idaho
Montana
Minnesota
Wisconsin
Maine
Iowa
New Hampshire
Virginia
Wyoming
Arkansas
Kentucky
Delaware
Massachusetts
New Jersey
Connecticut
Rhode Island
Missouri
Oregon
Washington
Oklahoma
Indiana
Kansas
Ohio
Pennsylvania
Hawaii
Utah

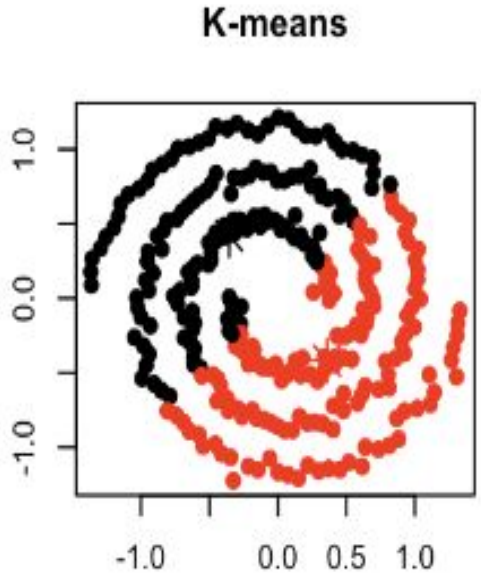
Spectral clustering algorithm

Spectral clustering has become increasingly popular due to its simple implementation and promising performance in many graph-based clustering. It can be solved efficiently by standard linear algebra software, and very often outperforms traditional algorithms such as the k-means algorithm.

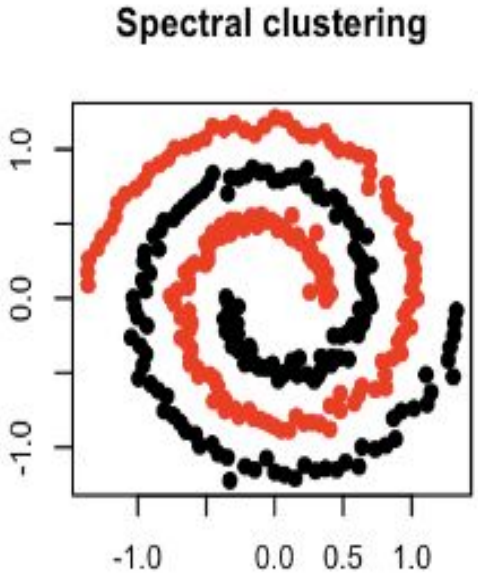
1. Create a similarity graph between our *N* objects to cluster.
2. Compute the first k eigenvectors of its Laplacian matrix to define a feature vector for each object.
3. Run k-means on these features to separate objects into k classes.



K-means



Spectral clustering



Thank you