Clustering Algorithms

What is clustering?

-Forming groups among data based on similarity

-A similarity metric can be defined by one or two features for easy comparison

-For our experiments, we use coordinate position as the similarity metric

REAL WORLD USE CASES:

-Social network analysis

-Market analysis

-Image segmentation

-Taxonomy

-GIS

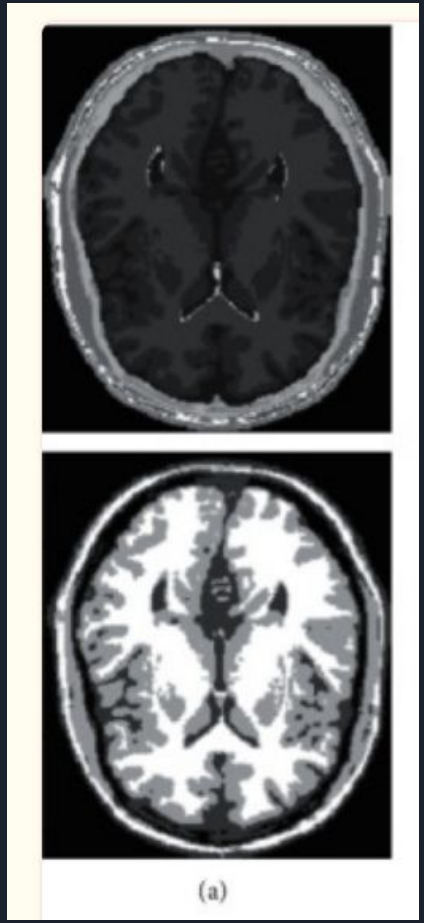
Why clustering algorithms?

Clustering is very important in computer science as it is used heavily by Machine Learning systems.

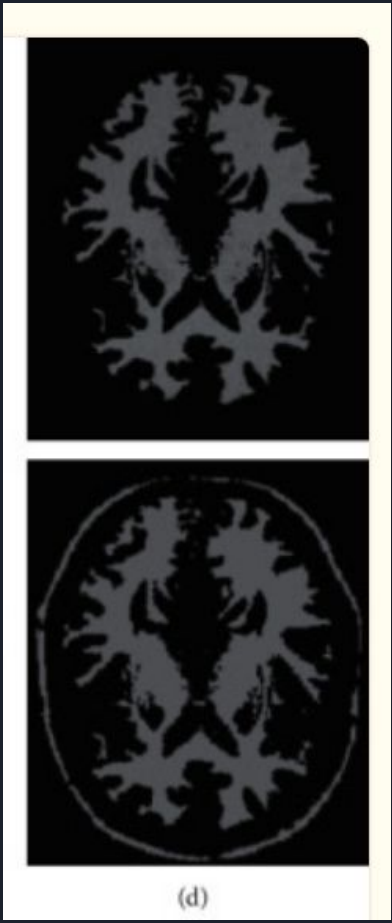
There is so much data available for use that it can be more efficient to group it based on clusters.

Working with colorful graphs is fun!
#matplotlib #sklearn #allmyhomieshatenumpy

Also almost halloween so brains :)



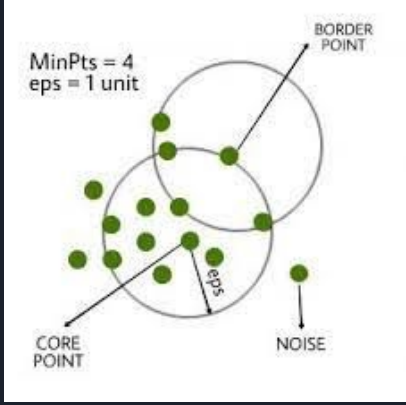
a) FCM Cluster results



d) White matter

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9365576/

DBSCAN



Density Based Spatial Clustering Applications with Noise:

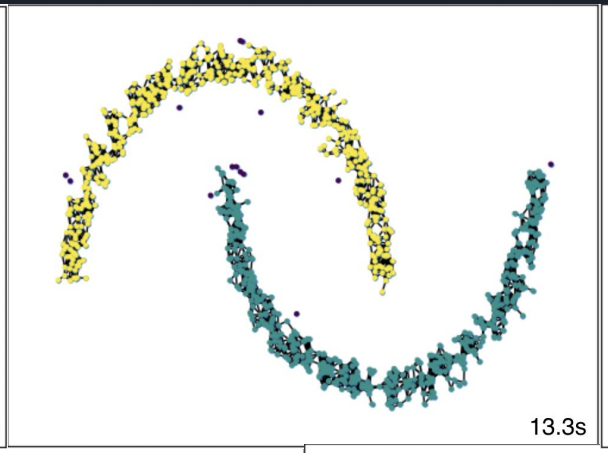
- Clusters points depending on density
  - Density is determined by two metrics: the radius ε in which to check for neighboring points, and minpts, which is how many points must be in this radius for a region to be considered dense
- For a given point, the density is checked. If the region is dense enough, neighboring points are checked for density. This continues until all points are either in a neighborhood, or are considered noise.

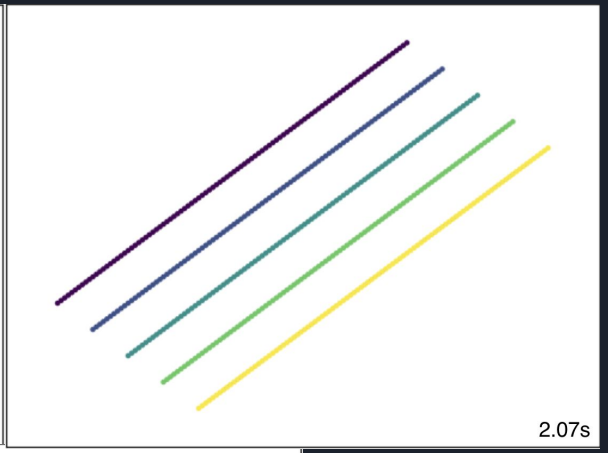
DBScan - Advantages and Disadvantages

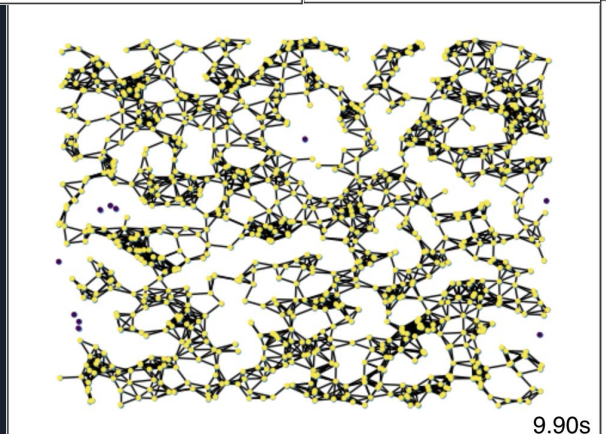
Advantages - Automatically computes amount of groups, ignores “noise”, O(n) + O(m) space complexity since only edges and nodes need to be stored.

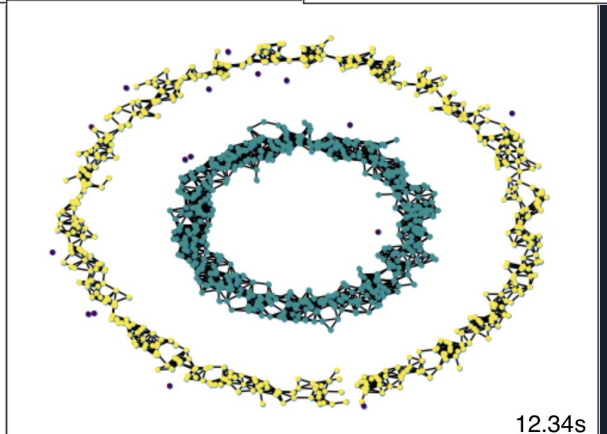
Disadvantages - $O(n^2)$ runtime, runtime increases drastically with higher density, must manually calibrate radius and minpts constraints

Graph Visualization for DBScan









k-Means!

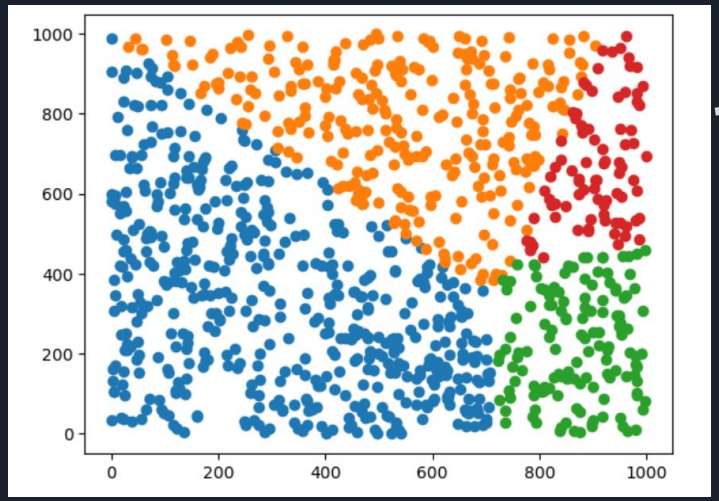
Clustering based on number of desired clusters and distance from centroid:

a. K number of centroids are randomly assigned
b. Every data point is assigned to the closest centroid
c. Mean of each cluster is calculated
d. Centroids are updated to be the mean of the cluster
e. Iterate through Steps a-d a certain number of times to minimize errors (10 iterations seems to be enough for tests of 1000 points)

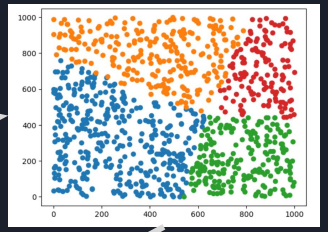
Runs fast! However, it must be run multiple times to get desirable results because randomly assigned centroids are not always evenly distributed.

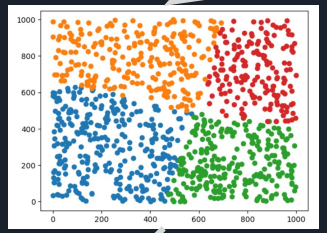
Amount of distance calculations done per iteration is n data points times k clusters because the distance from each point to each centroid is computed.

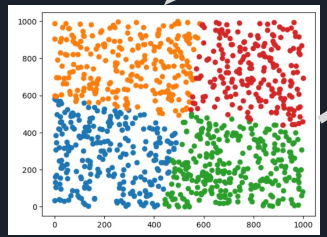
How the iteration works:



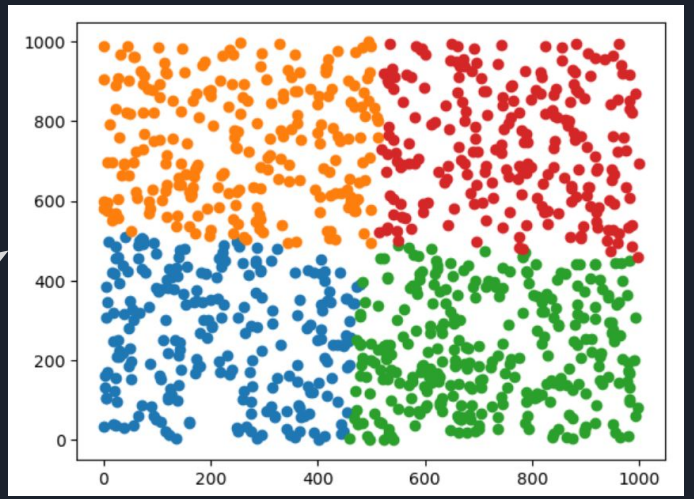
Start: uneven groups







End: symmetrical clusters



k-Means Advantages and Disadvantages

Advantages:

-Runtime is dependent on the amount of data points and the amount of clusters. Thus it can be said the average runtime is O(n) because the amount of clusters acts as a constant multiple, k.

-Clusters are generally evenly distributed after multiple runs

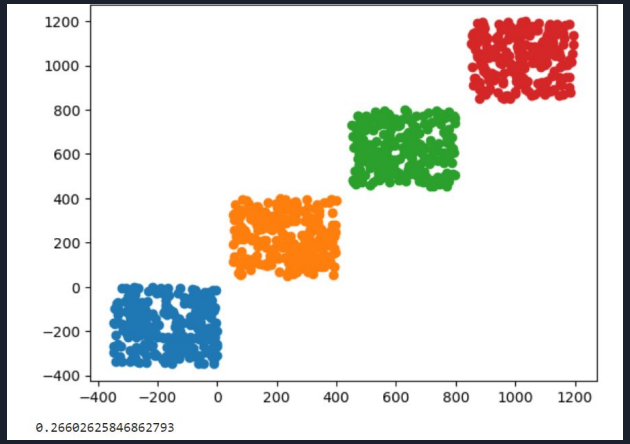
-Easy to change amount of clusters for any data set

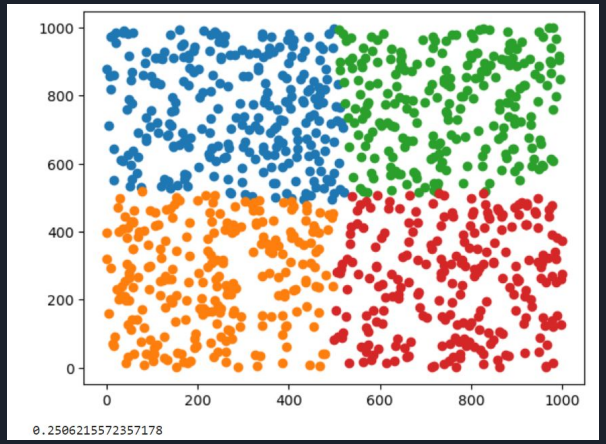
Disadvantages:

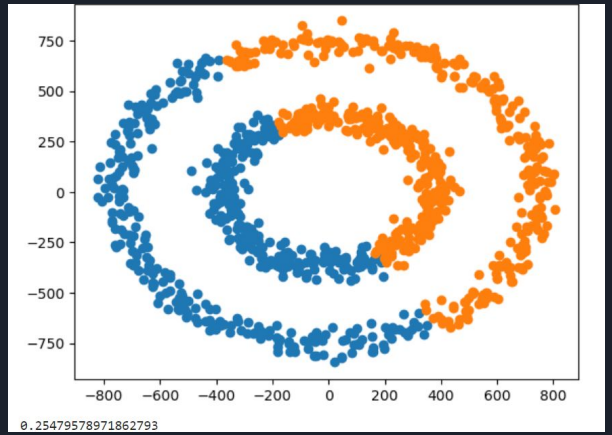
-Takes multiple runs to get desired output

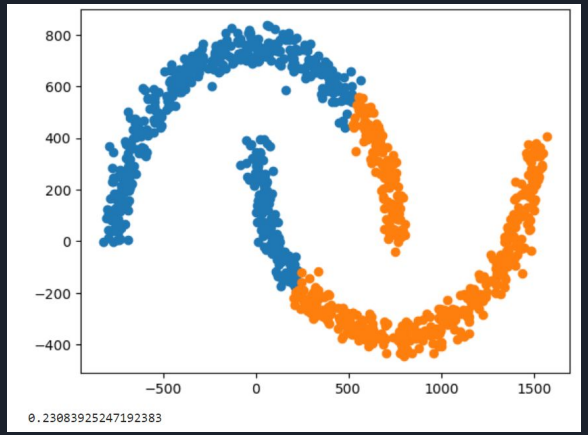
-Does not work well on data sets containing anything other than blob-shapes

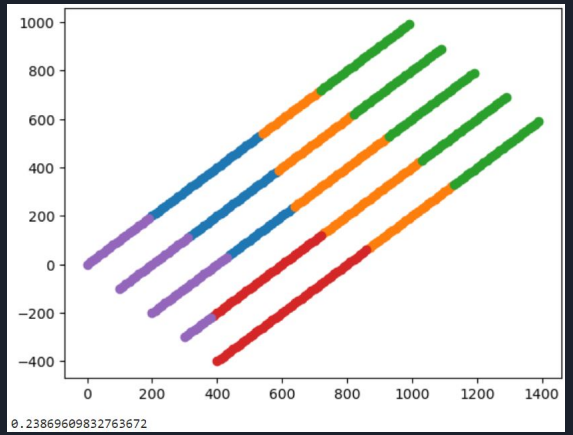
-Amount of clusters must be manually input


0.26602625846862793

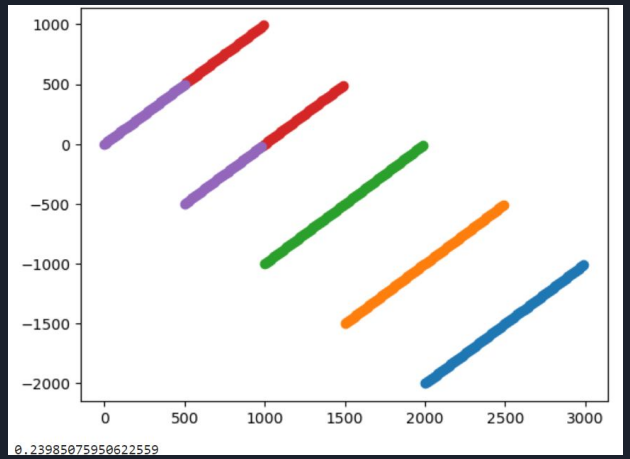

0.2506215572357178

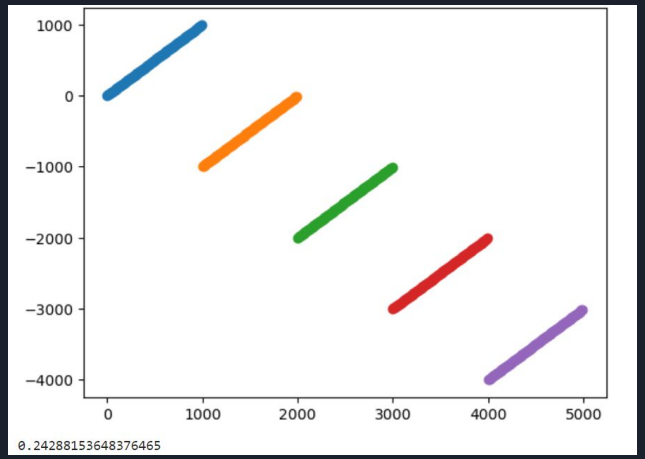

0.25479578971862793

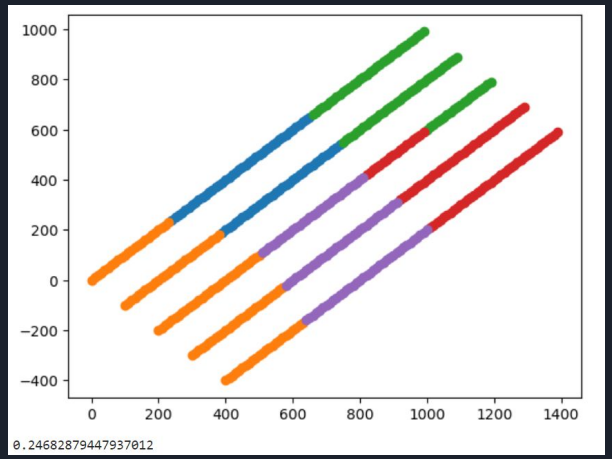

0.23083925247192383

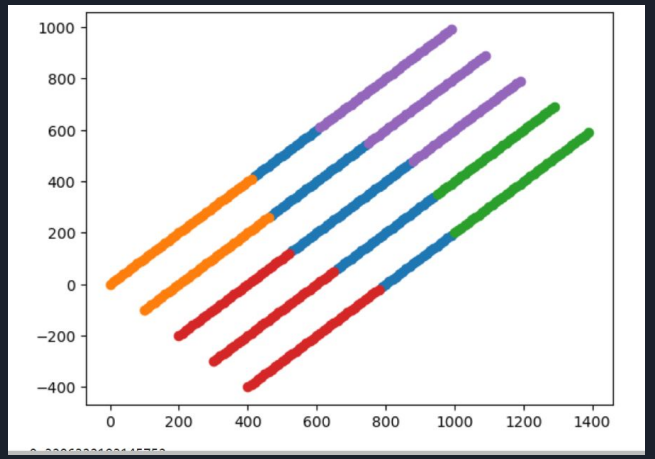

0.23869609832763672

Lines for comparison: K-means not ideal

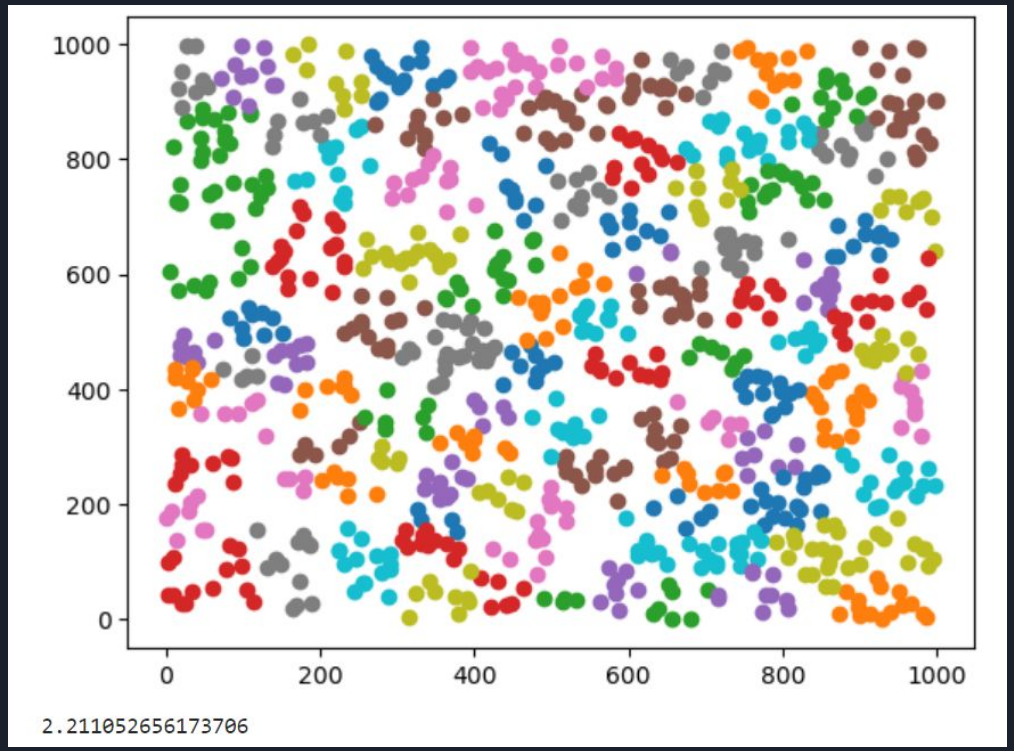


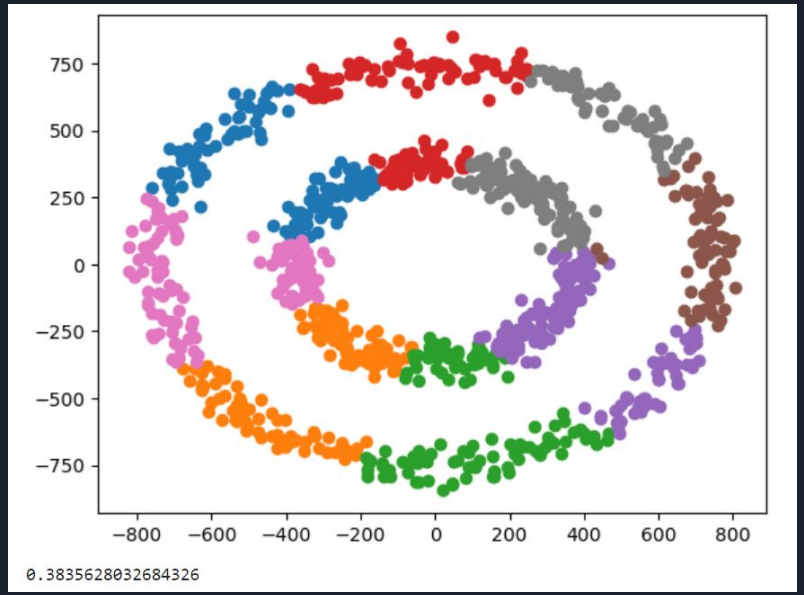






K-means can make lots of clusters!



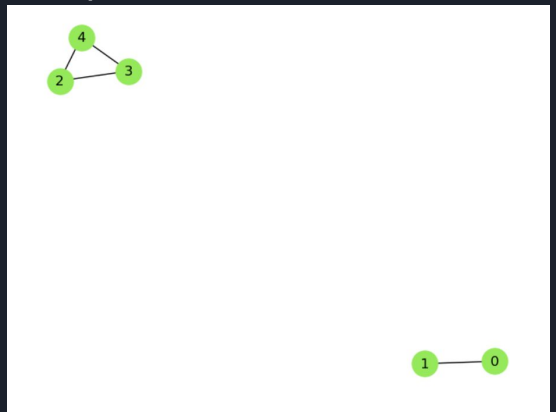


Spectral Clustering

Spectral clustering uses K-means clustering, but first modifies the data.

1. Make graph, A, based on density for the given dataset, build adjacency matrix
2. Find the Density Matrix, D, for graph A
3. Find the Laplacian Matrix for A. (L = D-A)
4. Find Eigenvalues and Eigenvectors for L.
5. Make matrix V with first k Eigenvectors of L where k is the specified amount of clusters
6. Use K-Means to cluster the rows of V

Graph:



Adjacency:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Degree:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

Laplacian:

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

Eigenvectors:

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Hierarchical Clustering

Clustering points and clusters by how much they relate to each other.

Closest points/clusters are identified, then merged into one. Process repeats until all points are in one cluster, graded with levels of relatedness. Visually, it is plotted with a dendrogram:

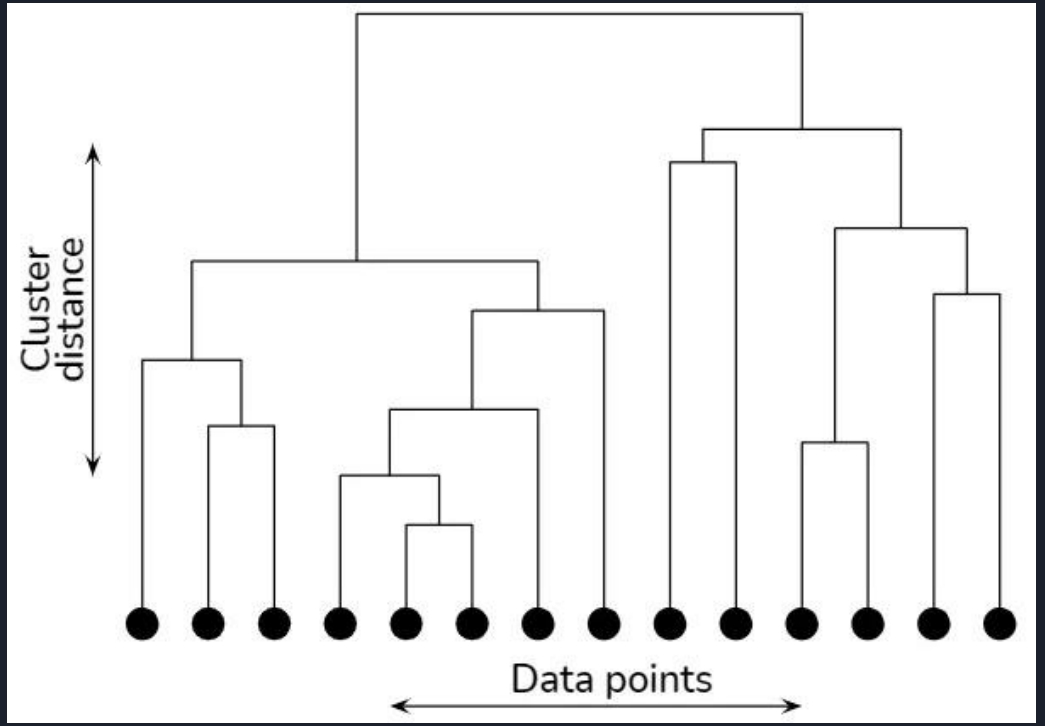


Image source: https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8

Advantages and Disadvantages

-Best used on datasets with hierarchical relationships or data points that relate to each other in varying degrees

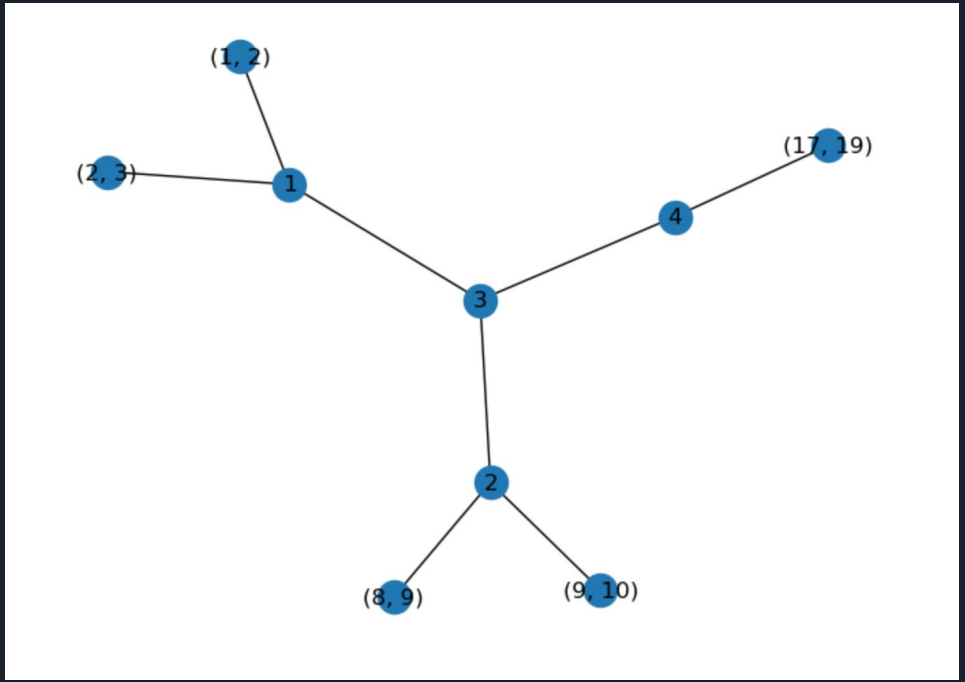
-Doesn’t require specification of number of clusters, as it will merge them all to one

-More time complex and less suitable for larger data sets

Complexity and Code Challenges

Iteration 1 Pseudocode:

1. Treat each data point as its own cluster
2. Calculate a pairwise distance matrix
3. Find the smallest distance and merge the two clusters together
4. Record the new cluster and update the matrix
5. Loop steps 3 & 4 until one cluster remains
6. Draw the “dendrogram”



Second Iteration Pseudocode

1. Treat all data points as unique clusters
2. Calculate a pairwise distance matrix, as well as arrays populated with a) each nodes' closest relative and b) the distances between them
3. Find the smallest element in the distance array and merge the closest clusters, updating all data structures.
4. Use the matrix to determine the new values of the distance and closest neighbor arrays
5. Repeat steps 3-4 until one cluster remains
6. Draw the dendrogram

Solving Recurrence Relations

Iteration 1:

$T(n)=t(n-1) + n^2$

$T(1)=1$

$T(n-1) = t(n-2) + (n-1)^2$

$T(n) = (t(n-2) + (n-1)^2) + n^2$

$T(n) = T(1) + n(n+1)(2n+1)/6$

$O(n^3)$

Iteration 2:

$T(n)=t(n-1) + \sim n$

$T(1)=1$

$T(n-1) = t(n-2) + n$

$T(n) = (t(n-2) + n) + n$

$T(n) = T(1) + n*n$

$O(n^2)$

Distance Comparisons for Clustering Algorithms

DBScan: n^2 distance comparisons to find which points are in radius ε

Spectral Clustering: n^2 distance comparisons to build adjacency matrix, plus distance comparisons for K-means clustering

K-Means:(k clusters) x (n points) x (m iterations) distance comparisons to determine clusters with minimum error

Hierarchical Clustering: n^2 distance comparisons/n^3 depending on implementation

Conclusion & Sources

Questions?

Sources:

NIH: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9365576/

Pai, Prasad. 2021: https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8

UCSB: https://sites.cs.ucsb.edu/~veronika/MAE/summary_SLINK_Sibson72.pdf

Springer: https://link.springer.com/article/10.1007/s40745-015-0040-1#Sec4

Towards Data Science: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

Medium: https://medium.com/udemy-engineering/understanding-k-means-clustering-and-kernel-methods-afad4eec3c11

Max-Planck Institute: https://www.cs.cmu.edu/~aarti/Class/10701/readings/Luxburg06_TR.pdf