Unit 4
Clustering

- DBScan
- Hierarchical Clustering
- Fuzzy Clustering
- Graph Based Clustering

Objective

- Clustering Algorithms
  - Centroid based
  - Density based
  - Hierarchical
  - Fuzzy
  - Graph based

Evaluation of Clustering

- Challenging due to
  - In many real-world datasets, the boundaries between clusters are not clear-cut.
    - Some data points might sit at the boundary of two clusters and could be reasonably assigned to both.
  - Different applications might prioritize different aspects of clustering.
    - For example, in one application, it might be essential to have tight, well-separated clusters, while in another, capturing the overall data structure might be more important.

Types of Evaluation

- Two types of clustering evaluation measures (or metrics)

- **Internal measures** do not require any ground truth to assess the quality of clusters. They are based solely on the data and the clustering results.

- **External measures** compare the clustering results to ground truth labels.

Internal Evaluation Measures

- Most internal validation measures are based on the following two criteria:

- **Compactness measures** how closely related objects in the same cluster are.
  - Compactness can be measured in different ways, such as by using the variance of the points within each cluster, or computing the average pairwise distance between them.
- **Separation measures** how distinct or well-separated a cluster is from other clusters.
  - Examples for measures of separation include pairwise distances between cluster centers or pairwise minimum distances between objects in different clusters.

External Evaluation Measures

- External evaluation measures are used when the true labels of the data points are known.
- These measures compare the the results of the clustering algorithm against the ground truth labels.

Other Clustering Algorithms

Hierarchical Clustering

- Hierarchical clustering organizes data into a hierarchy of clusters, represented as a tree-like structure known as a dendrogram
  - This algorithm builds a hierarchy of clusters by iteratively merging or splitting clusters based on their similarity.

Hierarchical Clustering

- Types:
  - There are two main types of hierarchical clustering:
    - agglomerative (bottom-up) and
    - divisive (top-down).
- Strengths
  - Hierarchical clustering can discover clusters of arbitrary shapes and sizes, and it provides a visual representation of the hierarchical relationships between clusters.
- Weaknesses
  - Hierarchical clustering can be computationally expensive, especially for large datasets. It is also sensitive to the initial ordering of the data points and the choice of the distance metric.

Hierarchical Clustering - Types

- Agglomerative Clustering:
  - Bottom-up approach merging similar clusters sequentially.

- Divisive Clustering:
  - Top-down approach dividing clusters iteratively.

Basic Agglomerative Hierarchical Algo.

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

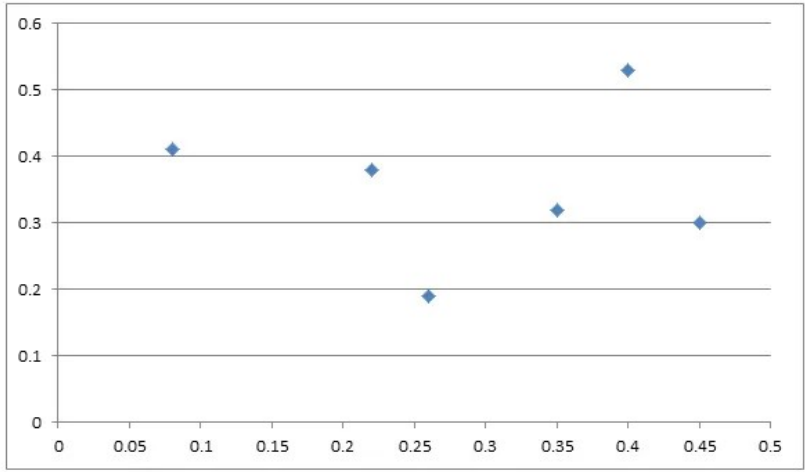
Hierarchical Agglomerative Clustering (HAC)

- Types
  - single link
    - For the single link or MIN version of hierarchical clustering, the proximity of two clusters is defined as the minimum of the distance (maximum of the similarity) between any two points in the two different clusters
  - complete link
    - For the complete link or MAX version of hierarchical clustering, the proximity of two clusters is defined as the maximum of the distance (minimum of the similarity) between any two points in the two different clusters

Hierarchical Agglomerative Clustering (HAC)

- Use the distance matrix in table below to perform single link and complete link hierarchical clustering. Show your results by drawing a dendogram.
- The dendogram should clearly show the order in which the points are merged.

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30



HAC

- Calculate Euclidean distance, create the distance matrix.

$$Euclidean\ Distance[(x,y),(a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$$

$$Distance(p1,p2) = \sqrt{(0.40-0.22)^2 + (0.53-0.38)^2}$$

$$= \sqrt{(0.18)^2 + (0.15)^2}$$

$$= \sqrt{0.0324 + 0.0225}$$

$$= \sqrt{0.0549}$$

$$= 0.23$$

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

HAC

- Find the minimum value element from distance matrix.
  - The minimum value element is (p3,p6)and value is 0.11
  - i.e. our 1st cluster **(p3,p6)**

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

HAC

- Recalculate or update the distance matrix for cluster(p3,p6)
- Same formula can be used for p2,p4,p5
- Updated distance matrix

$$Min[dist(point1, point2)]$$

$$Min[dist((p3, p6), p1)]$$

$$Min[dist((p3, p1), (p6, p1))]$$

$$Min[dist(0.22, 0.23)]$$

$$= 0.22$$

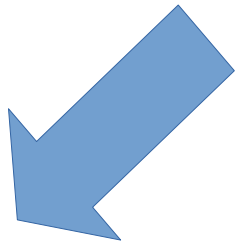
	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

HAC

- Repeat to get another pair
  - The minimum value element is (p2,p5)and value is 0.14
  - i.e. our 2nd cluster (p2,p5)
- Recalculate or update the distance matrix for cluster (p2,p5)
  - Same formula can be used for (p3,p6),p4
  - Updated distance matrix:

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0



	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

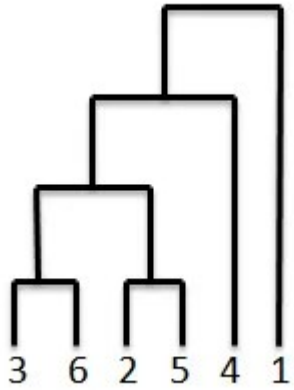
HAC

- Repeat
  - 1st cluster (p3,p6)
  - 2nd cluster (p2,p5)
  - 3rd cluster (p2,p5,p3,p6)
  - 4th cluster (p2,p5,p3,p6,p4)
  - 5th cluster (p2,p5,p3,p6,p4,p1)

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0

HAC

- Repeat
  - 1st cluster (p3,p6)
  - 2nd cluster (p2,p5)
  - 3rd cluster (p2,p5,p3,p6)
  - 4th cluster (p2,p5,p3,p6,p4)
  - 5th cluster (p2,p5,p3,p6,p4,p1)



MAX version of HAC

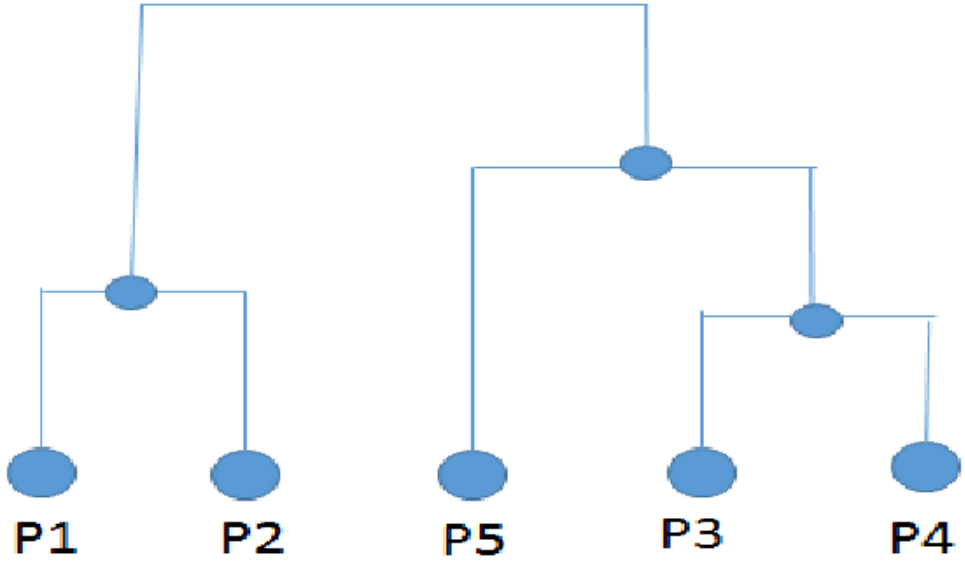
- Find the min distance and group them
- Update the distance
- $$\begin{aligned} dist(\{P1, P2\}, \{P3\}) &= max(dist(P1, P3), dist(P2, P3)) \\ &= max(0.41, 0.64) \\ &= 0.64 \\ dist(\{P1, P2\}, \{P4\}) &= \mathtt{max}(dist(P1, P4), dist(P2, P5)) \\ &= \mathtt{max}(0.55, 0.98) \\ &= 0.98 \\ dist(\{P1, P2\}, \{P5\}) &= \mathtt{max}(dist(P1, P5), dist(P2, P5)) \\ &= \mathtt{max}(0.35, 0.98) \\ &= 0.98 \end{aligned}$$

-	P1	P2	P3	P4	P5
P1	0.00	0.10	0.41	0.55	0.35
P2	0.10	0.00	0.64	0.47	0.98
P3	0.41	0.64	0.00	0.44	0.85
P4	0.55	0.47	0.44	0.00	0.76
P5	0.35	0.98	0.85	0.76	0.00

	P12	P3	P4	P5
P12	0.00	0.64	0.98	0.98
P3	0.64	0.00	0.44	0.85
P4	0.98	0.44	0.00	0.76
P5	0.98	0.85	0.76	0.00

MAX version of HAC

	P12	P345
P12	0.00	0.98
P345	0.98	0.00



Complete Link Dendogram

Numericals

- https://aryabhattacollege.ac.in/samplepaper/637227449508725497DataMining(Chapter8).pdf
- https://www.analyticsvidhya.com/blog/2021/06/single-link-hierarchical-clustering-clearly-explained/
- https://medium.com/@rohanjoseph_91119/learn-with-an-example-hierarchical-clustering-873b5b50890c

Density Based Clustering

- Clustering based on density (local cluster criterion), such as density-connected points or based on an explicitly constructed density function
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters

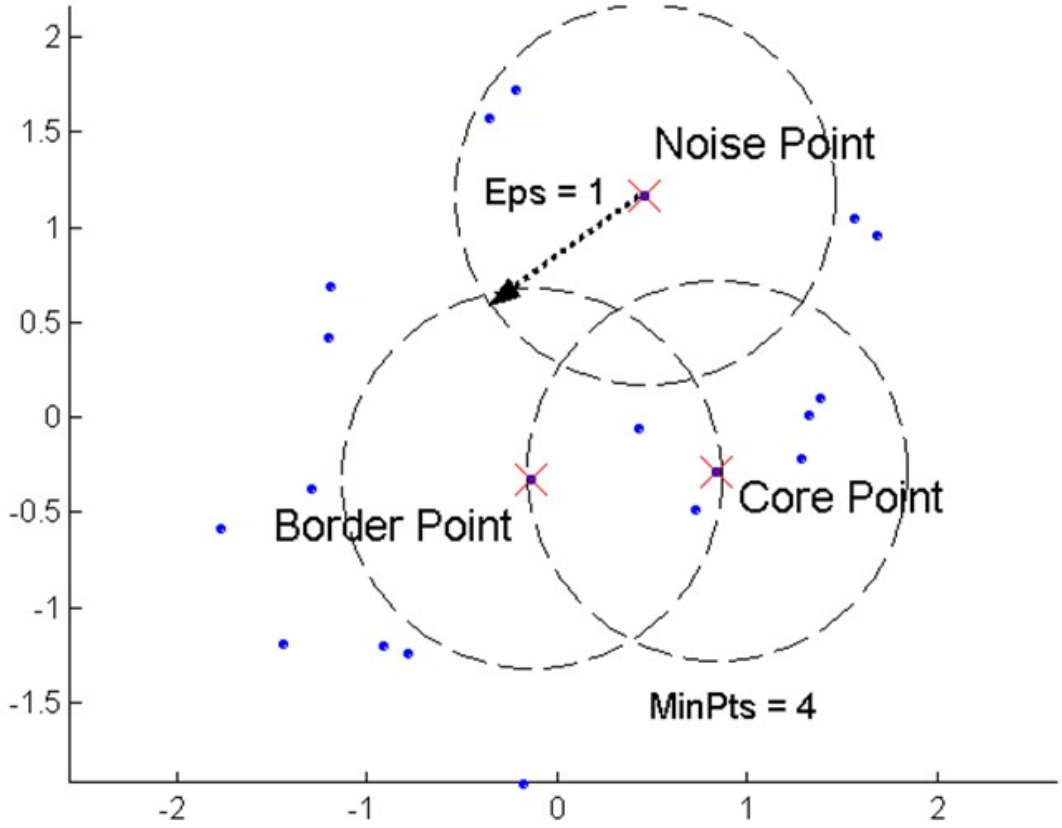
Density Based Clustering

- Algorithms
  - DBSCAN: Ester, et al. (KDD 96)
  - DENCLUE: Hinneburg & D. Keim (KDD 98/2006)
  - OPTICS: Ankerst, et al (SIGMOD 99).
  - CLIQUE: Agrawal, et al. (SIGMOD 98)

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius r (**Ɛ - EPS**)
  - A point is a **core point** if it has more than a specified number of points (**MinPts**) within **Ɛ**
    - These are points that are at the interior of a cluster
  - A border point has fewer than **MinPts** within **Ɛ**, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.
- DBSCAN is a density-based algorithm.

DBSCAN- core, border, and noise points

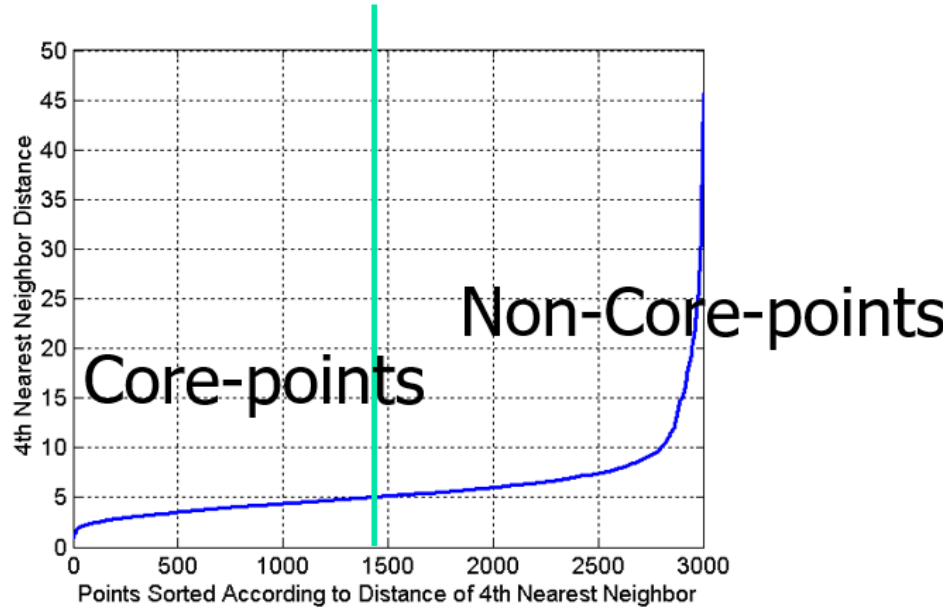


Simplified view of DBSCAN

- Create a graph whose nodes are the points to be clustered
- For each core-point c create an edge from **c** to every point **p** in the **ε-neighborhood** of **c**
- Set **N** to the nodes of the graph;
- If **N** does not contain any core points terminate
- Pick a core point **c** in **N**
- Let **X** be the set of nodes that can be reached from **c** by going forward;
  - create a cluster containing X U {c}
  - N=N / (X U {c})
- Continue with step 4

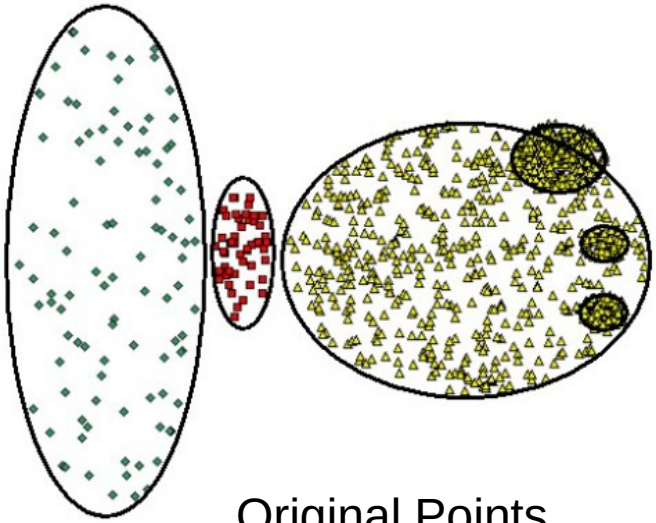
Estimating Ɛ and MinPts

- Idea is that for points in a cluster, their kth nearest neighbors are at roughly the same distance
  - Noise points have the kth nearest neighbor at farther distance
  - So, plot sorted distance of every point to its kth nearest neighbor
- Thus, eps=10



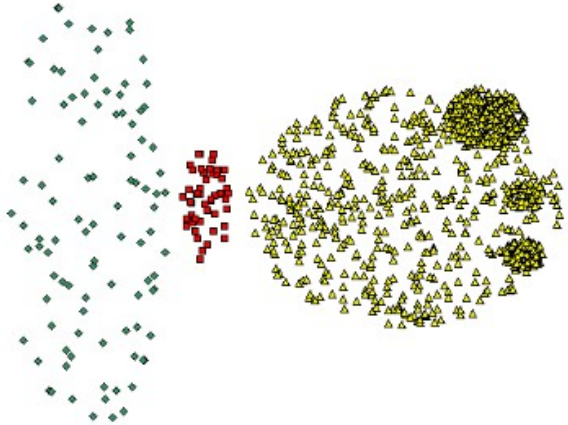
Where DBSCAN Fails

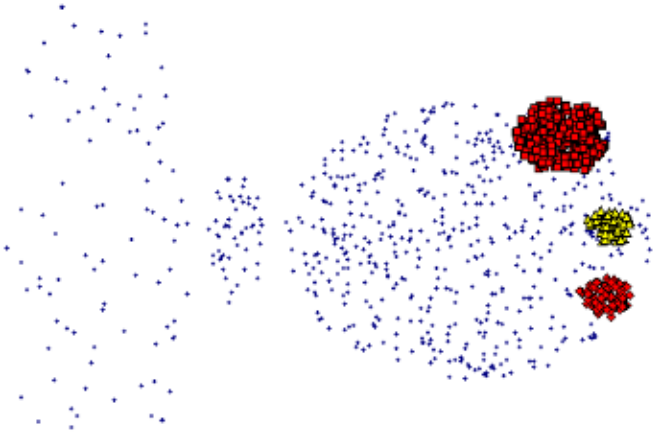
- Varying densities
- High-dimensional data



Original Points

Larger EPS , MinPts= 4





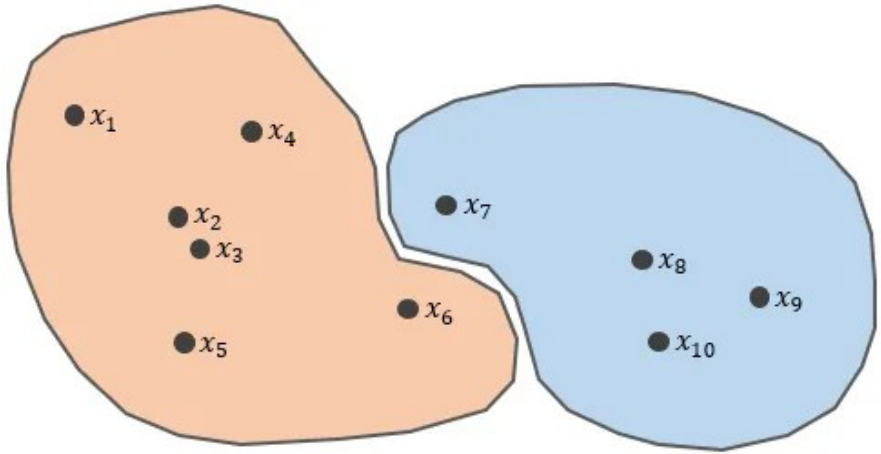
Smaller EPS , MinPts= 4

Resources

- https://cdn.aaai.org/KDD/1996/KDD96-037.pdf

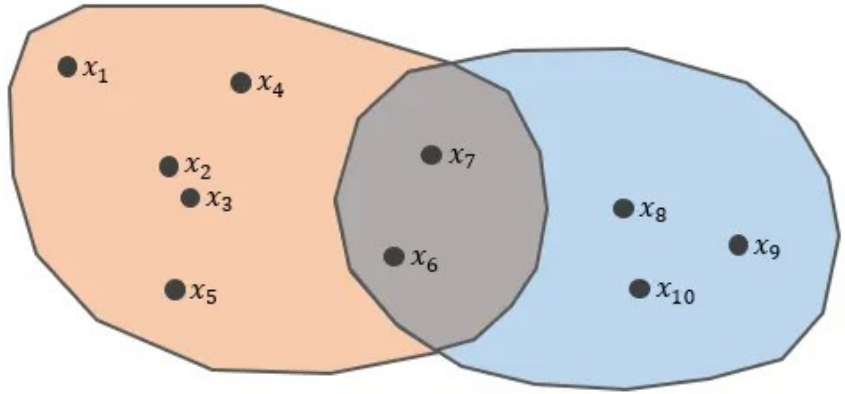
Hard partition

- Hard partition
  - where the data points are strictly allocated as a member of one cluster and are not a member of another cluster, assuming that the number of clusters is known.
  - The k-means is one of the algorithms that use a hard partition.



Soft Partition

- Every data point is given a probability of closeness [0, 1] for existing clusters, assuming that the number of clusters is known.
- One of the algorithms that use fuzzy partition is Fuzzy c-means that we will talk about it in depth.



Fuzzy Clustering

- Traditional clustering algorithms assign data points to exclusive clusters, where each point belongs to one and only one cluster.
- Fuzzy clustering, on the other hand, acknowledges the inherent ambiguity in real-world data and allows for a more flexible approach.
  - With fuzzy clustering, a data point can belong to multiple clusters to varying degrees, providing a richer representation of complex relationships.

Fuzzy Clustering

- The key idea behind fuzzy clustering lies in the concept of membership functions.
  - Instead of assigning binary membership values (0 or 1) as in traditional clustering, fuzzy clustering employs membership values ranging from 0 to 1.
  - These values represent the degree of belongingness of each data point to each cluster.
- A higher membership value indicates a stronger association with a particular cluster, while a lower value signifies a weaker association.

Application of fuzzy clustering

- Image Segmentation
  - In computer vision, fuzzy clustering can be used for image segmentation tasks.
  - By assigning membership values to pixels, it allows for a more flexible and granular delineation of image regions, enabling better object recognition and scene understanding.
- Pattern Recognition
  - Fuzzy clustering is useful in pattern recognition tasks, such as character recognition or speech recognition.
  - By capturing the uncertainty and variability inherent in patterns, it allows for more robust and accurate recognition algorithms.

Application of fuzzy clustering

- Customer Segmentation
  - Fuzzy clustering is valuable in customer segmentation for marketing purposes.
  - By considering the degree of membership to different segments, businesses can tailor personalized marketing strategies to effectively target customers based on their varying preferences and behaviors.
- Document clustering
  - Fuzzy clustering can be used to cluster documents based on their content, such as keywords, topics, and themes.
  - This can help in organizing and retrieving documents efficiently.

Fuzzy c-means (FCM)

- FCM is the most well-known and widely used fuzzy clustering technique.
  - It is an iterative algorithm that minimizes the sum of the weighted squared distances between each data point and the centers of the clusters.
  - The degree of membership of each data point to each cluster is calculated using a membership function, which assigns a probability value between 0 and 1 for each cluster.

Graph based Clustering

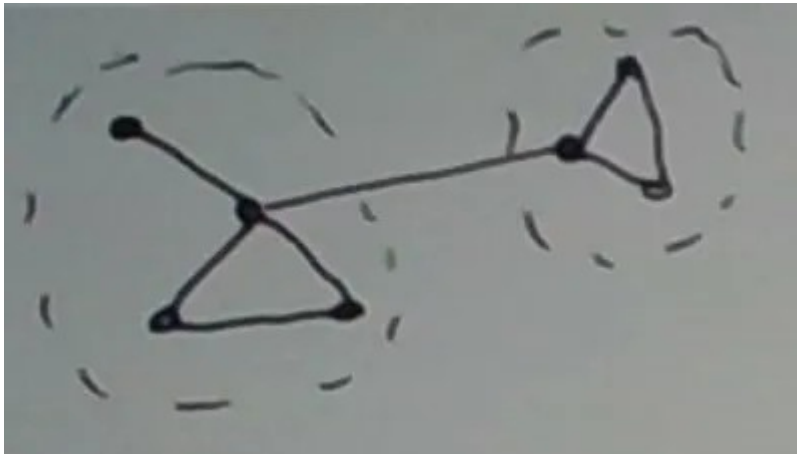
- Connected components clustering (CCC)
- HCS:
  - stands for highly connected subgraphs clustering

HCS

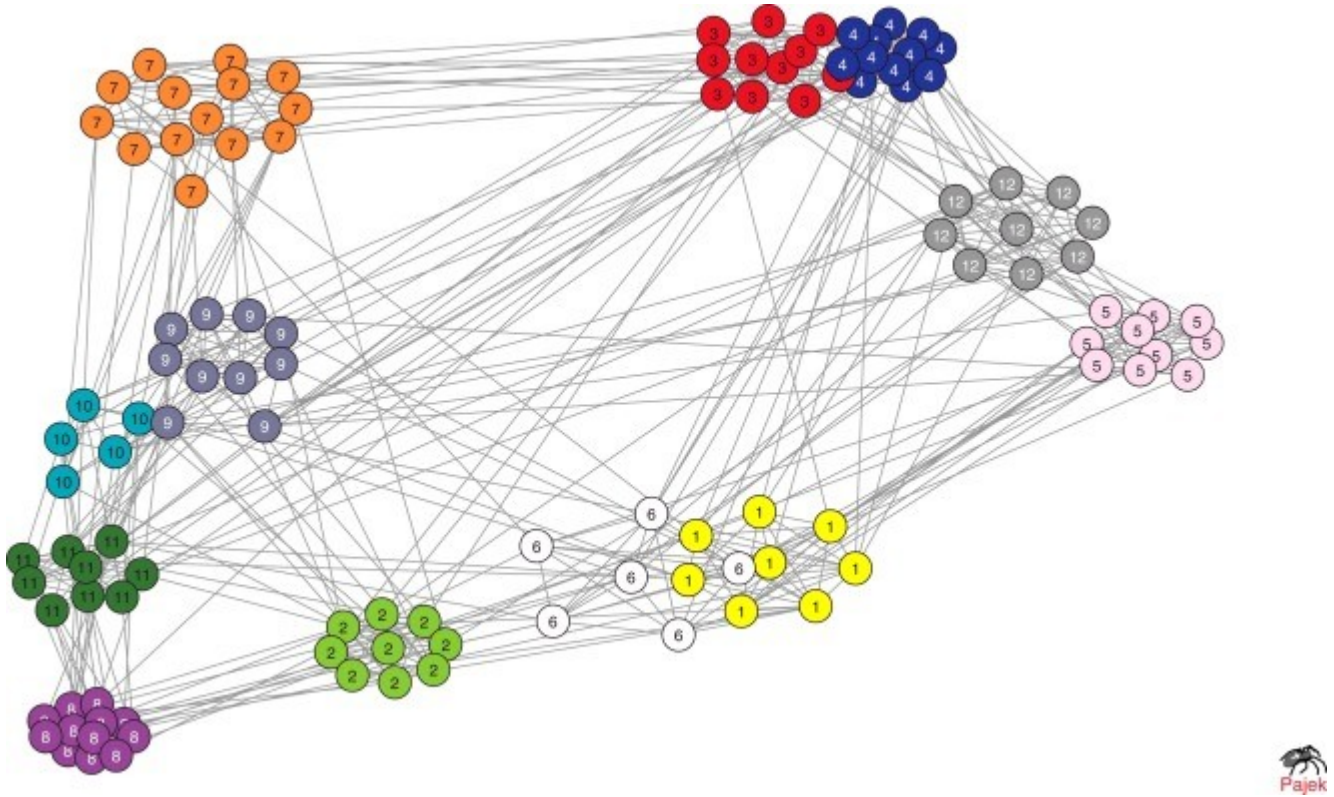
- It uses the notion of min-cuts.
- There are many heuristic algorithms to find min-cuts or approximations
- HCS starts with the entire graph and checks if it is highly connected.
  - We can define highly connected however we like so long as we can test for this condition efficiently.
  - If the graph is not highly connected, it partitions the graph into two non-empty subgraphs in such a way that the number of edges that cross the partition is minimized.
  - There are algorithms to find min-cuts under additional constraints.

HCS

- We then recurse on the two subgraphs.



HCS



Use of HCS

- Network based summarization
- Mapping application
- Use Case
  - Online Delivery
  - E-Commerce Store setup
  - Bus Route planning etc.

Thank you