Machine Learning

Lecture 18-19: Clustering

COURSE CODE: CSE451

2021

Course Teacher

Dr. Mrinal Kanti Baowaly

Associate Professor

Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Bangladesh.

Email: baowaly@gmail.com

What is Unsupervised Learning?

- Unsupervised learning is where you only have unlabeled input data (X) and allow the algorithm to work on its own to discover the interesting structure or pattern in the data.
- These are called unsupervised learning because unlike supervised learning there is no correct answers and there is no teacher (i.e., learning from the labeled training data).
- Unsupervised learning problems can be further grouped into **clustering** and **association** rule mining.

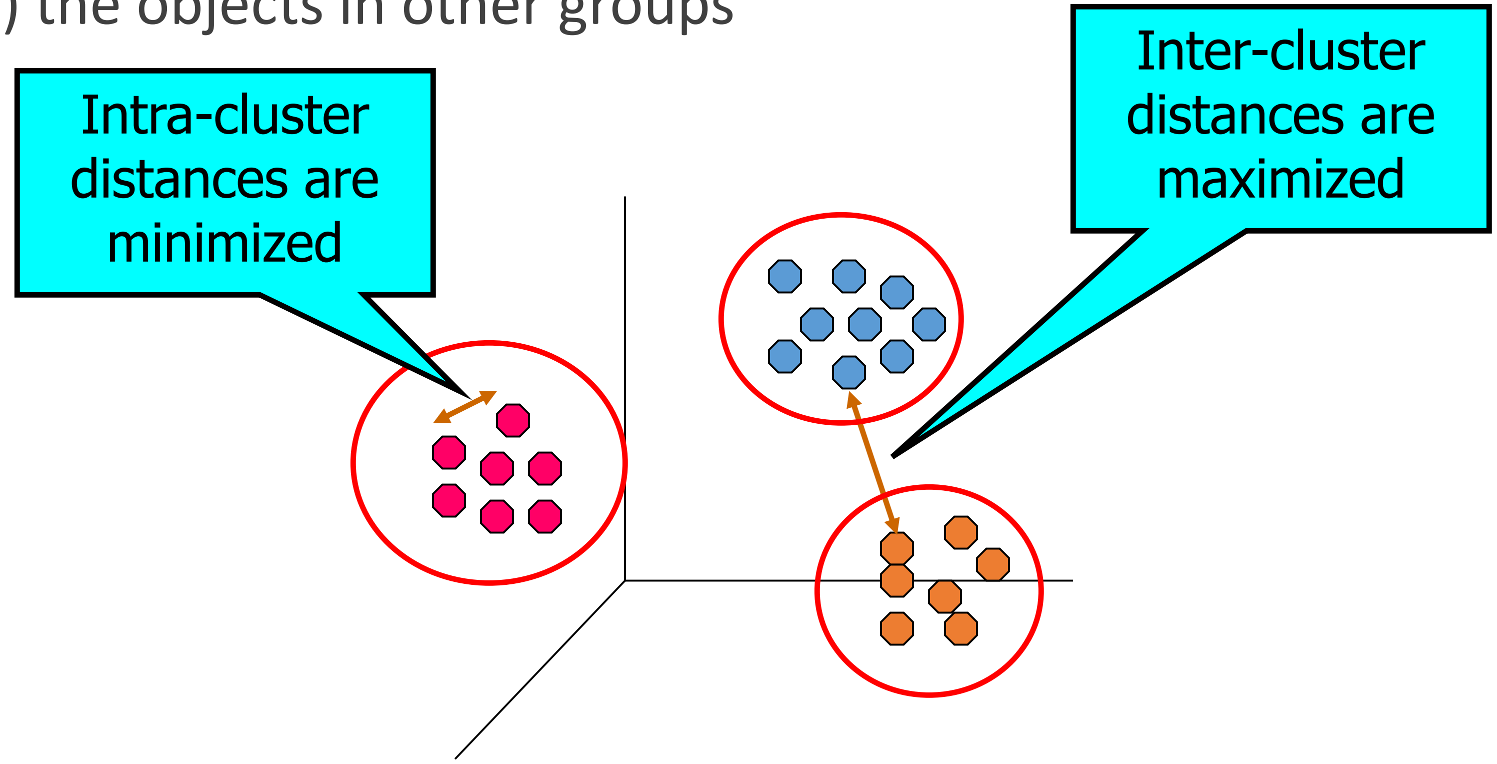
What is Unsupervised Learning?

- Unsupervised learning is where you only have unlabeled input data (X) and allow the algorithm to work on its own to discover the interesting structure or pattern in the data.
- These are called unsupervised learning because unlike supervised learning there is no correct answers and there is no teacher (i.e., learning from the labeled training data).
- Unsupervised learning problems can be further grouped into **clustering** and **association** rule mining.

What are the differences between supervised and Unsupervised Learning?

What is Cluster Analysis / Clustering?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Clustering (cont.)

- **Cluster:** a collection of data objects in which objects are similar to one another within the same cluster and dissimilar to the objects in other clusters.
- **Good Clustering:** produces high quality clusters in which intra-cluster similarity is high and inter-cluster similarity is low.

Factors that affect quality of clustering

- Similarity/distance measure and its implementation
- Definition and representation of cluster chosen
- Clustering algorithm

Applications of Clustering

- **Customer Segmentation:** This strategy is across functions, including banking, telecom, e-commerce, sports, advertising, sales, etc.
- **Document Clustering:** Cluster similar documents together
- **Image Clustering:** You can group similar images together.
- **Image Segmentation:** You can apply clustering to create clusters having similar pixels in the image together.
- **Recommendation Engines:** You can look at the songs liked by a person and then use clustering to find similar songs and finally recommend the most similar songs to him.

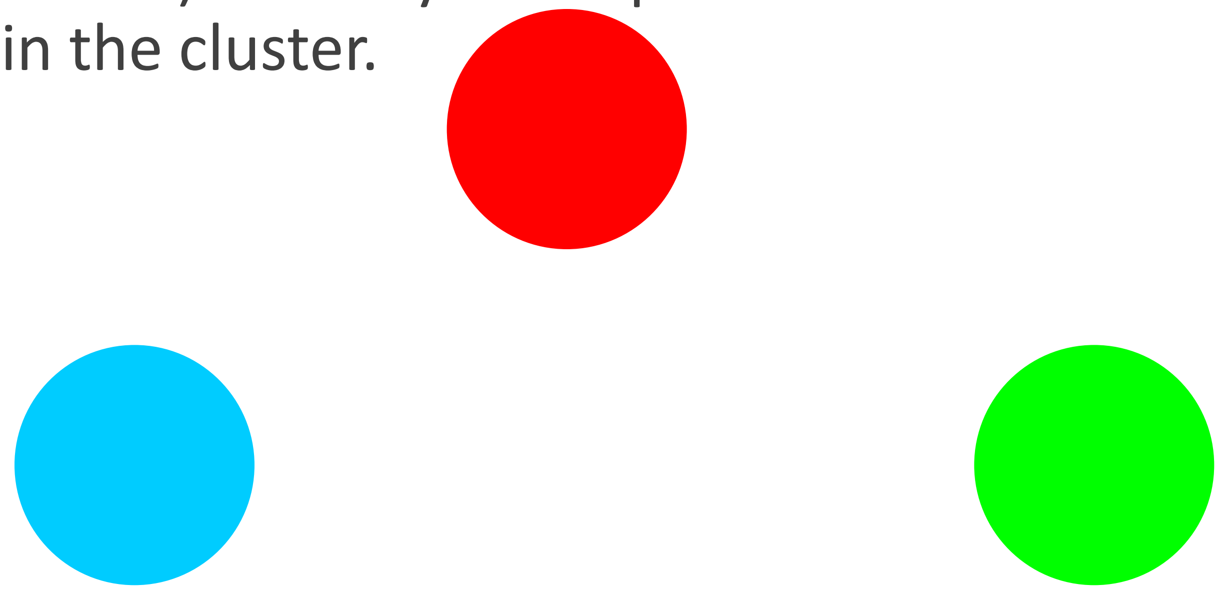
Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters

Types of Clusters: Well-Separated

Well-Separated Clusters:

- A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.

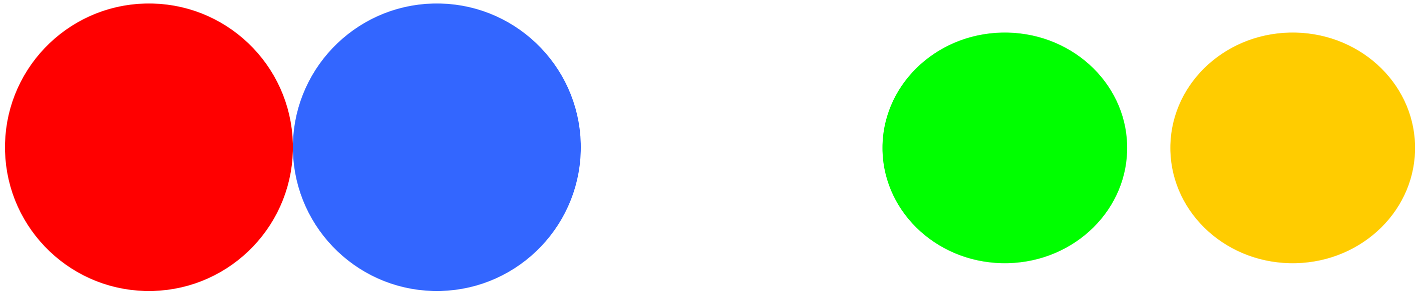


3 well-separated clusters

Types of Clusters: Center-Based or Partitioned

Center-based or Partitioned Clusters:

- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
- The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster

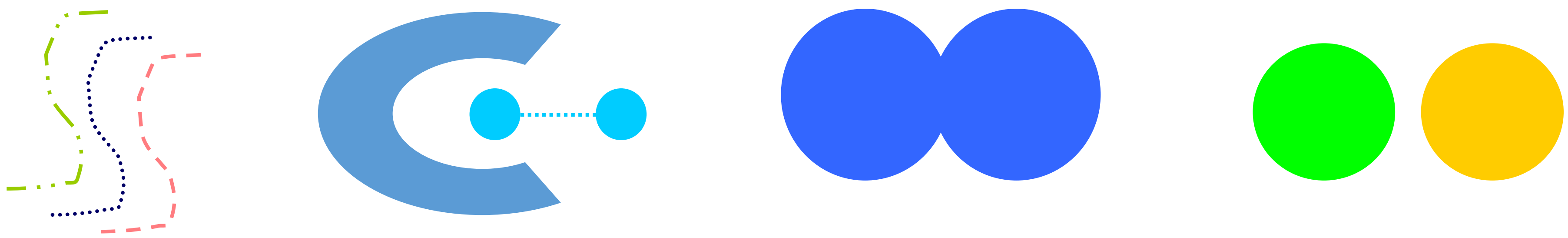


4 center-based clusters

Types of Clusters: Contiguity-Based

Contiguous Clusters (Nearest neighbor or Transitive):

- A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

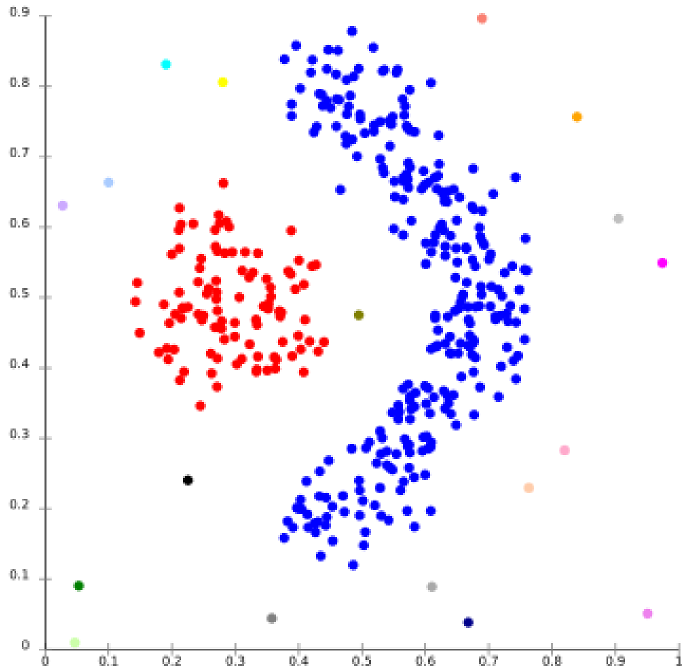


8 contiguous clusters

Types of Clusters: Density-Based

Density-based clusters:

- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Used when the clusters are irregular or intertwined, and when noise and outliers are present.



Popular Clustering Algorithms

- k-Means Clustering
- Hierarchical Clustering
- DBSCAN Clustering

k-Means Clustering

- This is a center-based, partitioned clustering technique that attempts to find a user-specified number of clusters (k), which are represented by their centroids.
- The algorithm is simple

k-Means Algorithm

Given k, the k-means algorithm:

Step 1: Select Initial centroids

select k initial centroids randomly

Step 2: Relocation

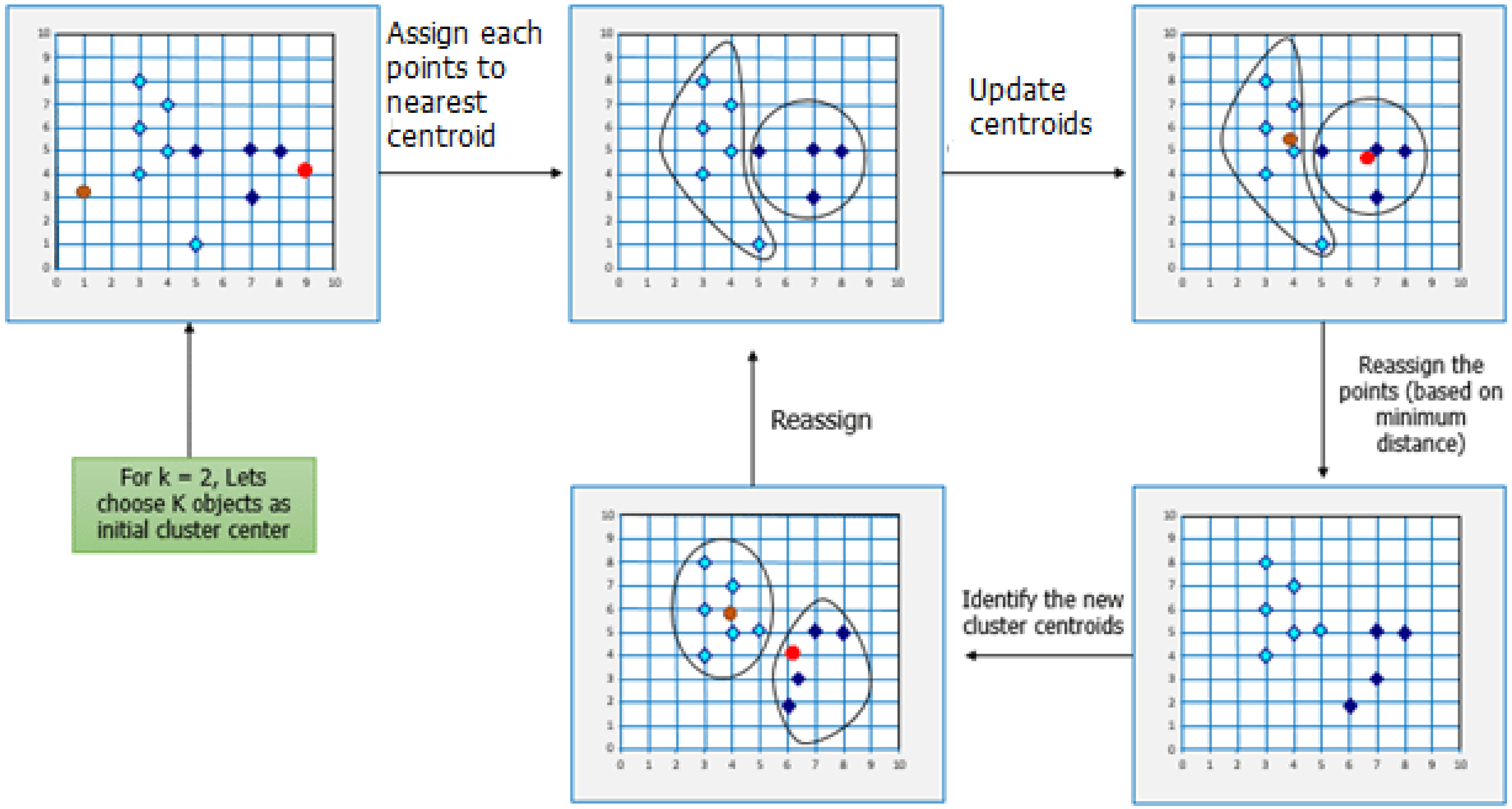
assign each object to the nearest centroid

Step 3: Update Centroids

compute mean as the centroids of the clusters of the current partition

Step 4: Go back to Step 2, stop when no more new relocation

Example: k-Means Clustering



Animated Example: k-Means Clustering

[Source: Andrey A. Shabalin]

Some issues with k-Means Clustering

- **How do we choose the number of clusters k?**

  Try different values of k, evaluate them and choose the best k value.

- **How do we choose the initial centroids?**

  Randomly choose k examples (data points) from the dataset as the initial centroids or

  Randomly choose k points in the dataset space as the initial centroids

- **How to assign data points to centroids?**

  Assign each data point to the closest centroid. 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.

- **How do we recompute/update the centroids?**

  Centroid of each cluster is (typically) the mean of the data points in the cluster. If there are *n* points on a 2D space, the mean calculation looks like: $\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}$

Evaluating k-Means Clustering

- The basic idea of k-means is to minimize variance or sum of squared errors (SSE) within clusters
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

where x is a data point in cluster C_i and m_i is the representative point (centroid) for cluster C_i

Stopping Criteria for k-Means Algorithm

- Centroids of newly formed clusters do not change
- No more new relocation i.e., no data points change their cluster location
- Maximum number of iterations are reached

Strength and weakness of k-Means Clustering

Strength

- Relatively efficient: $O(tkn)$, where n is # of objects, k is # of clusters, and t is # of iterations. Normally, k, $t \ll n$.
- Often terminates at a *local optimum*.

Weakness

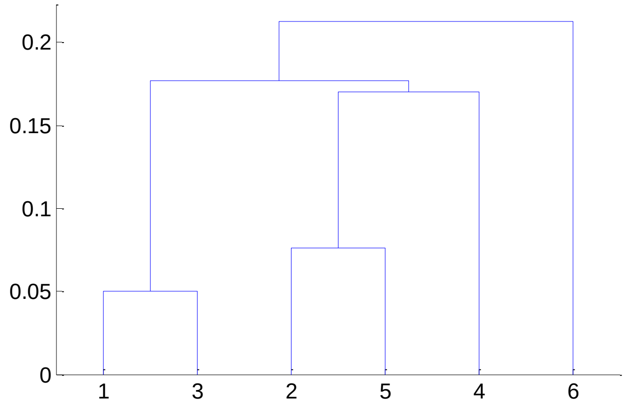
- Applicable only when *mean* is defined
- Need to specify *k,* the *number* of clusters, in advance.
- Unable to handle noisy data and *outliers.*
- Cannot handle clusters of different sizes & densities
- Not suitable to discover clusters with *non-convex shapes.*

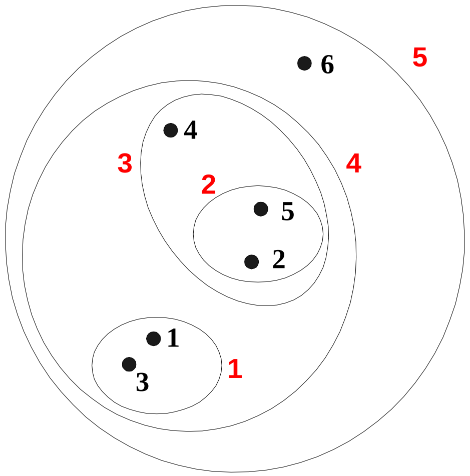
HW: k-Means Clustering Example & Implementation

- K-Means Clustering with Scikit-Learn
- Introduction to K-Means Clustering in Python with scikit-learn
- K-Means Clustering in Python with scikit-learn

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  – Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- They may correspond to meaningful taxonomies
  – Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

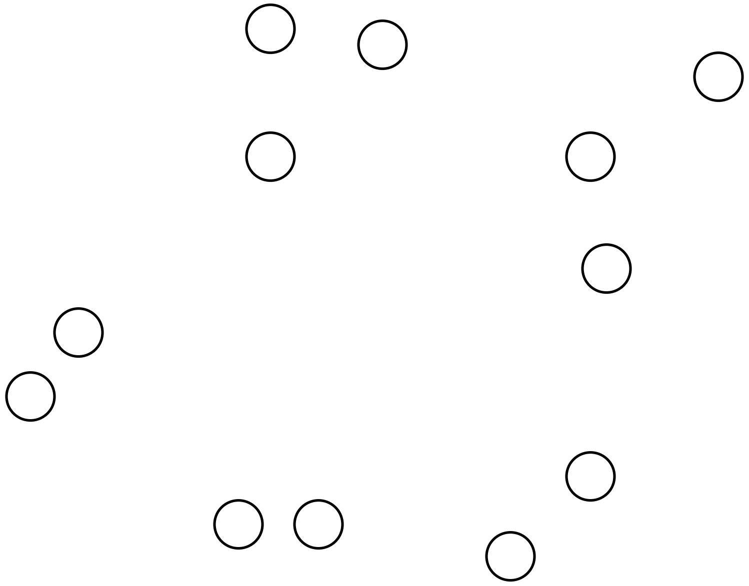
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

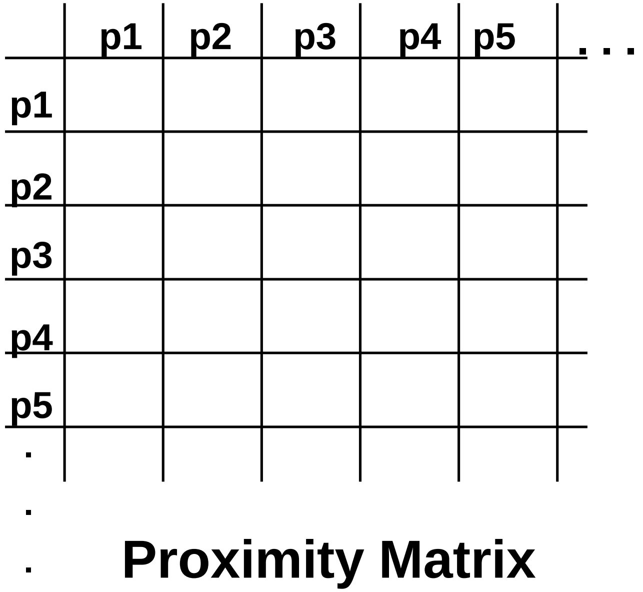
Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix to reflect the proximity between the new cluster & other clusters
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  – Different approaches to defining the distance between clusters distinguish the different algorithms

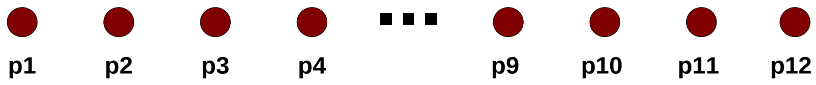
Starting Situation

- Start with clusters of individual points and a proximity matrix



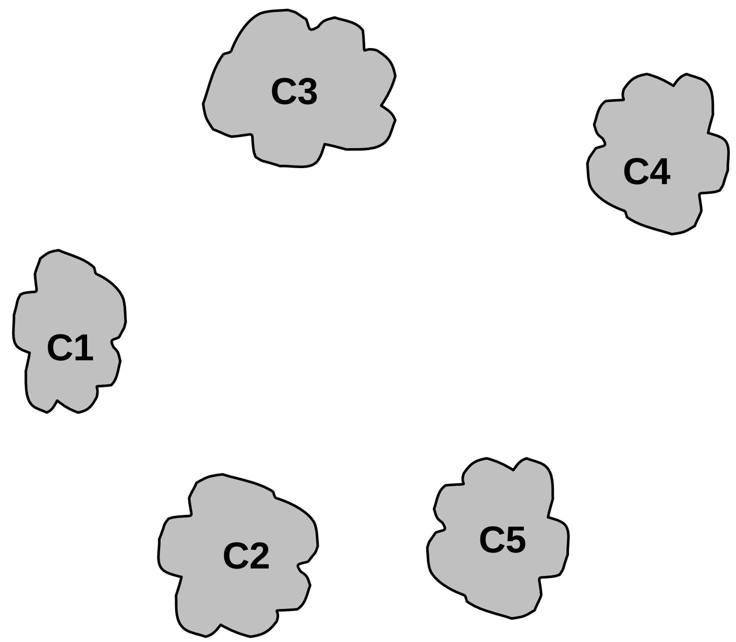


Proximity Matrix



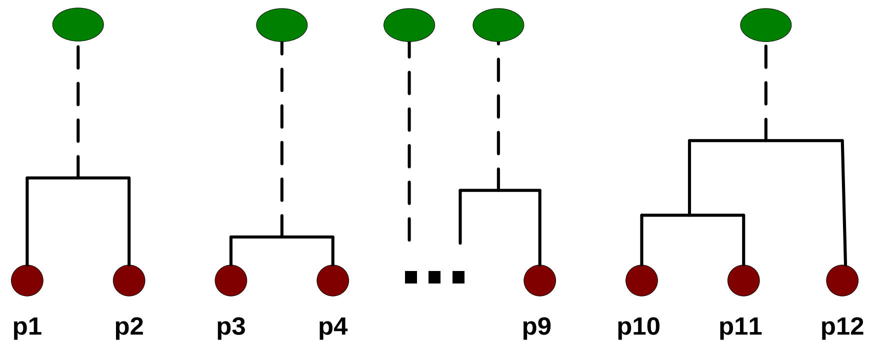
Intermediate Situation

- After some merging steps, we have some clusters



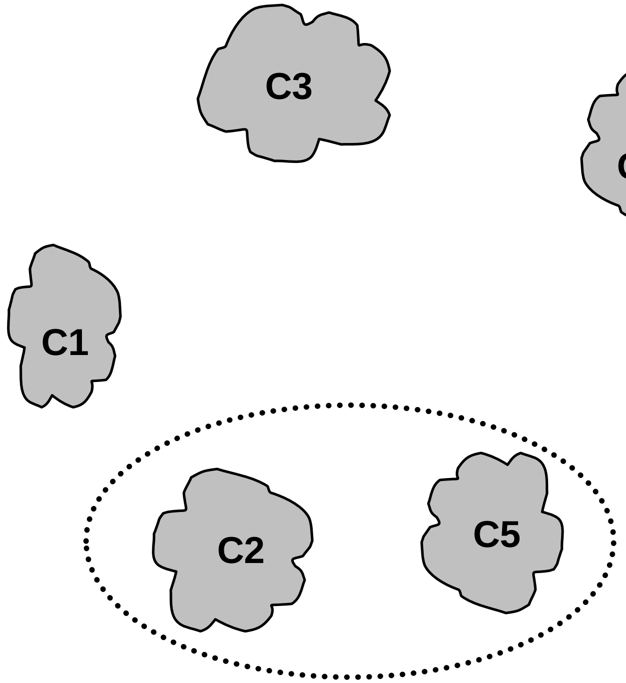
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

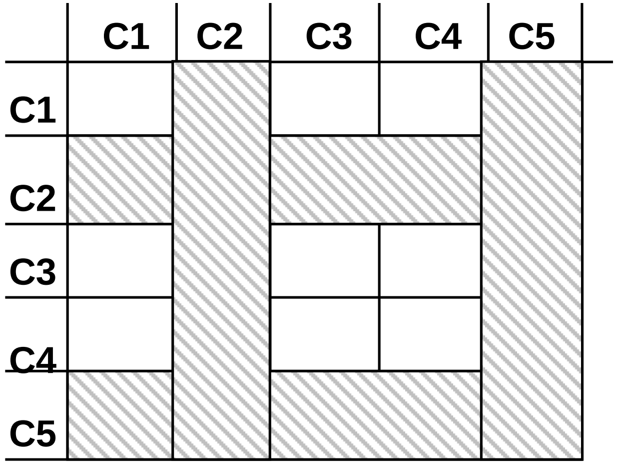
Proximity Matrix



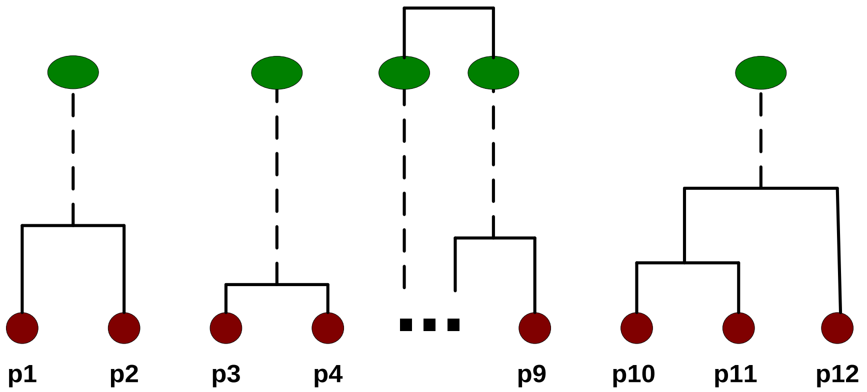
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



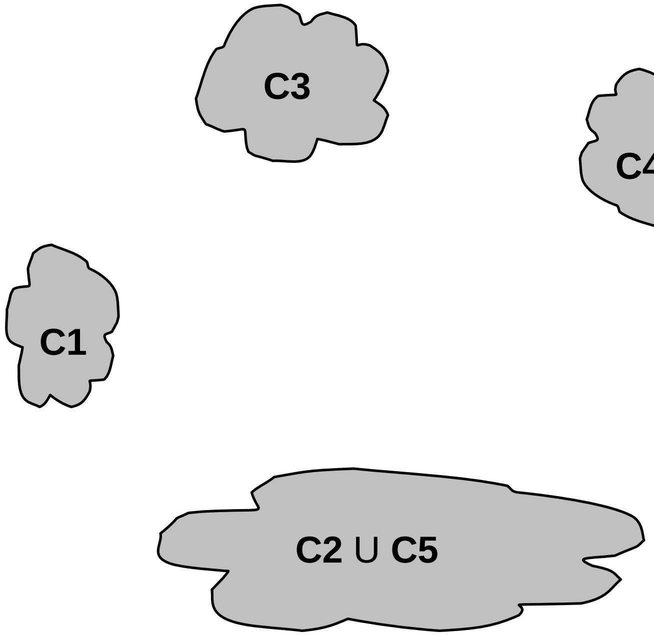


Proximity Matrix



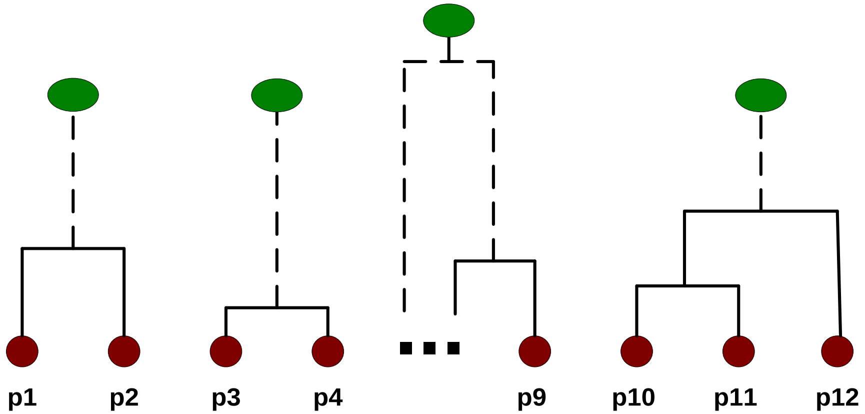
After Merging

- The question is “How do we update the proximity matrix?”

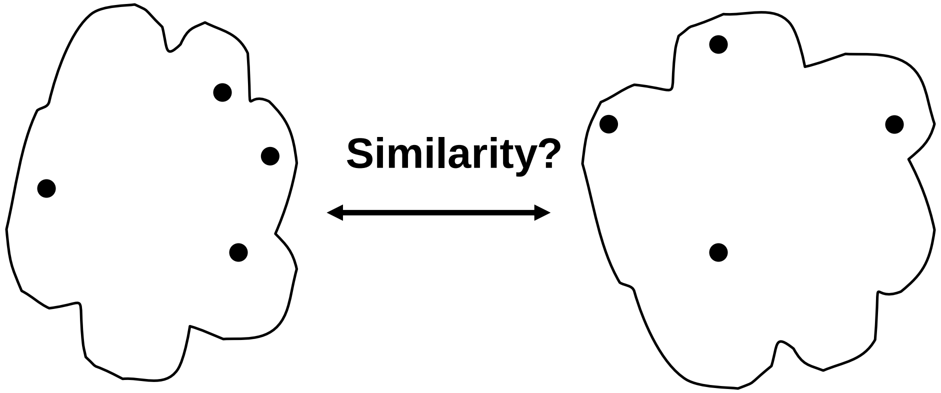


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

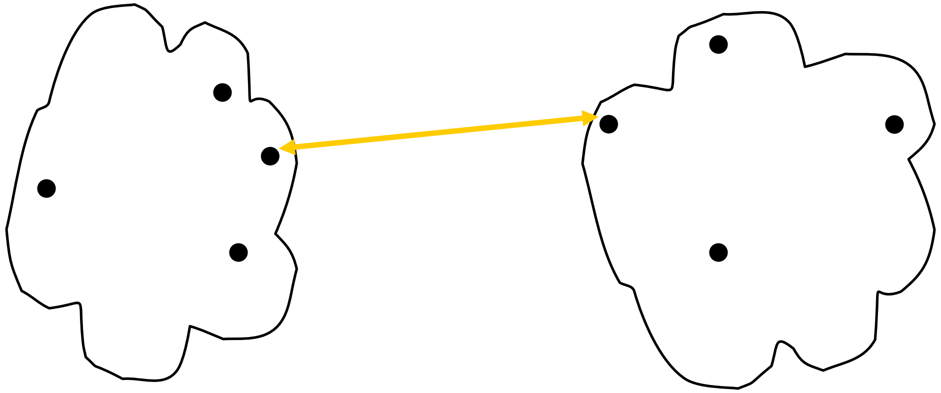


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

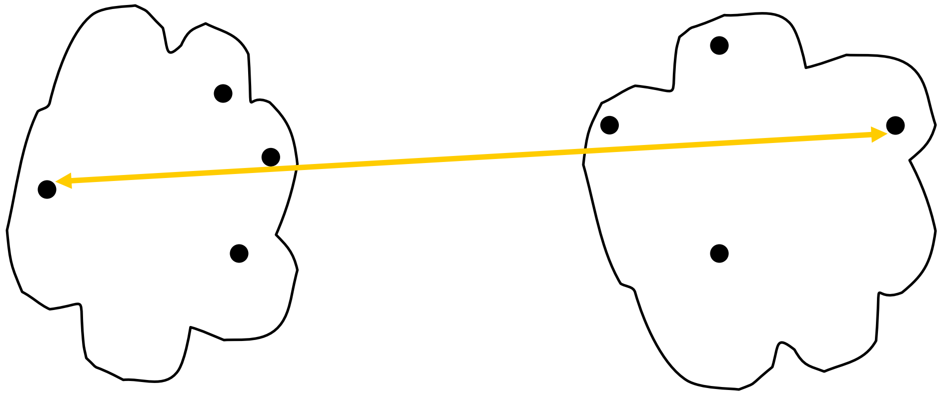


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

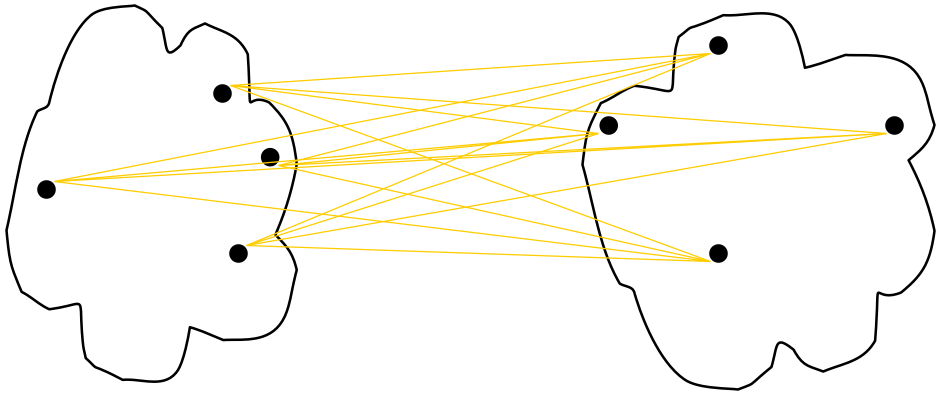


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

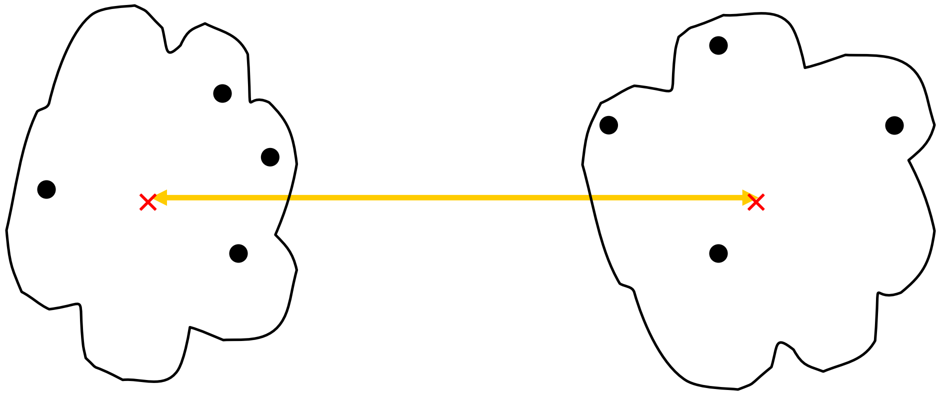


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

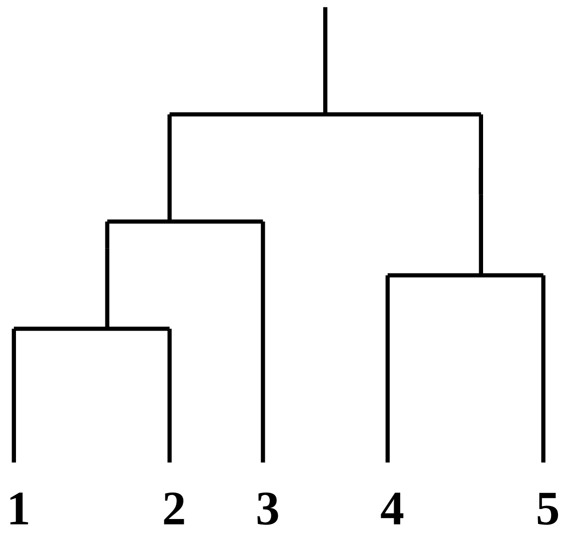
Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

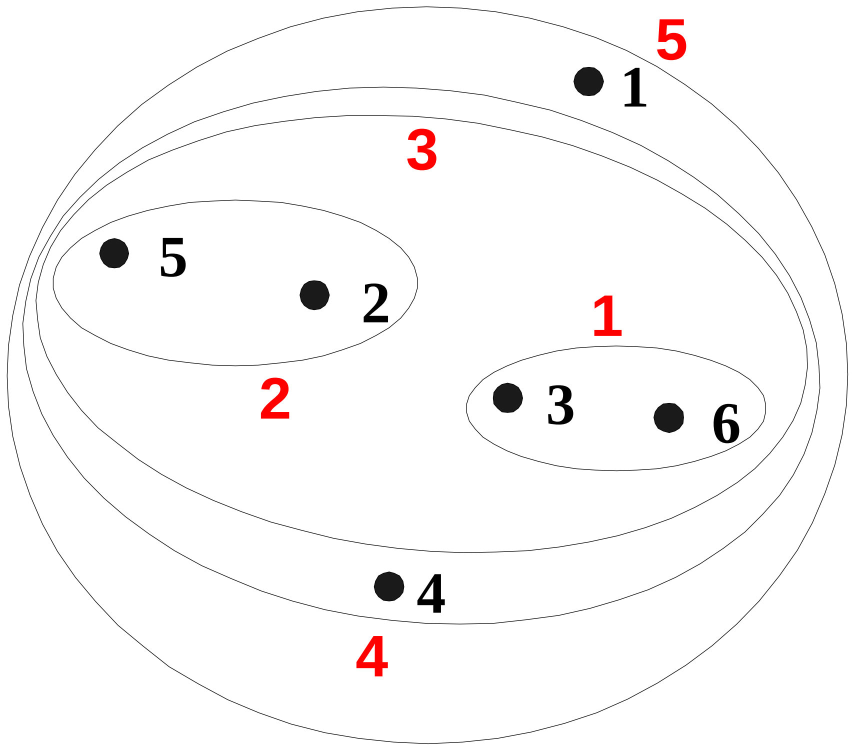
Cluster Similarity: MIN or Single Link

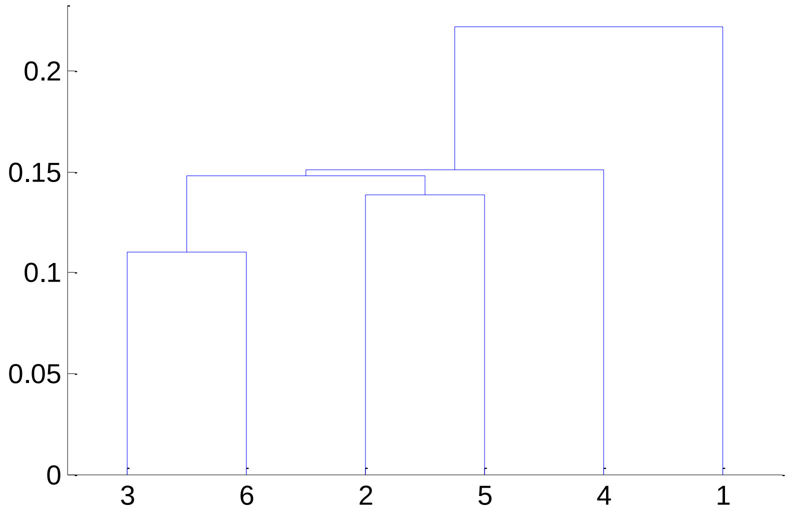
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

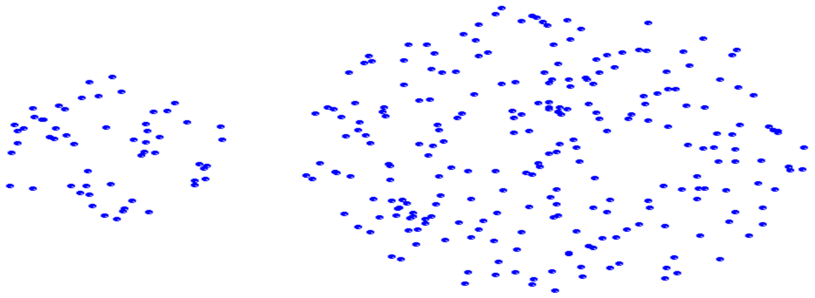




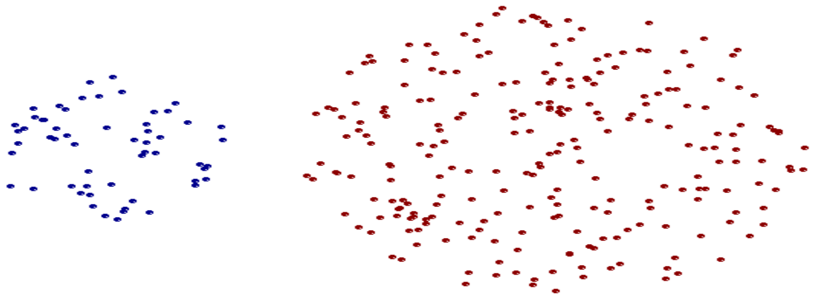
Nested Clusters

Dendrogram

Strength of MIN



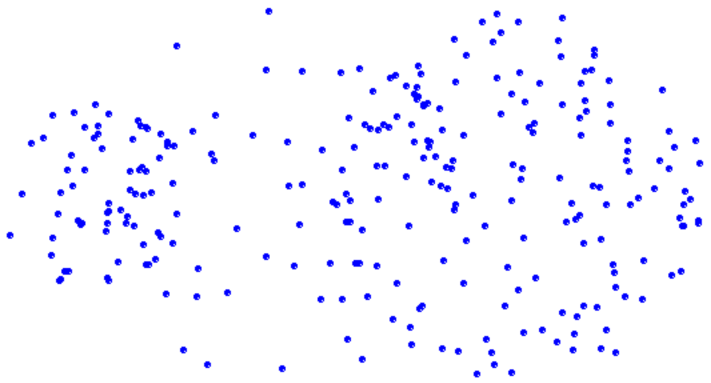
Original Points



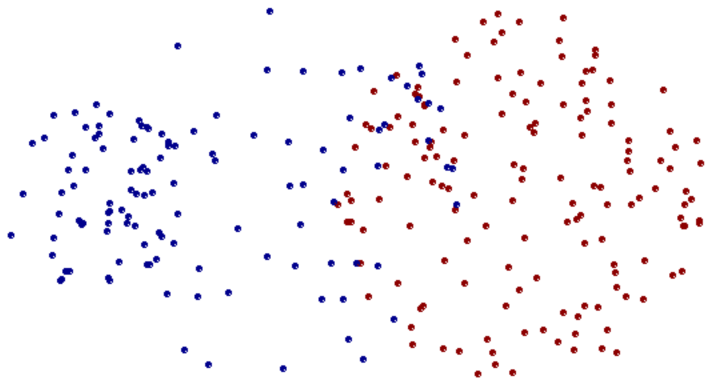
Two Clusters

- **Can handle non-elliptical shapes**

Limitations of MIN



Original Points



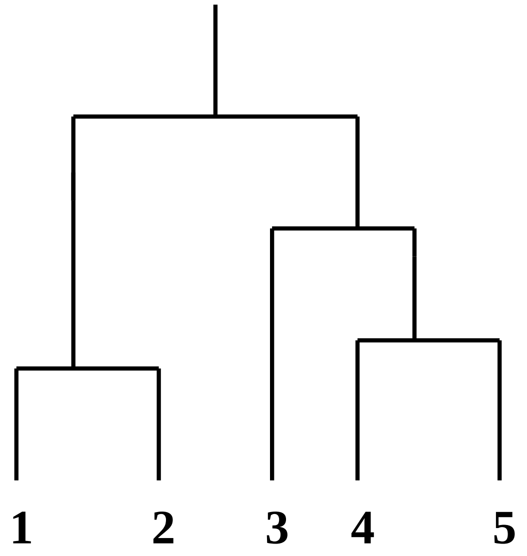
Two Clusters

• **Sensitive to noise and outliers**

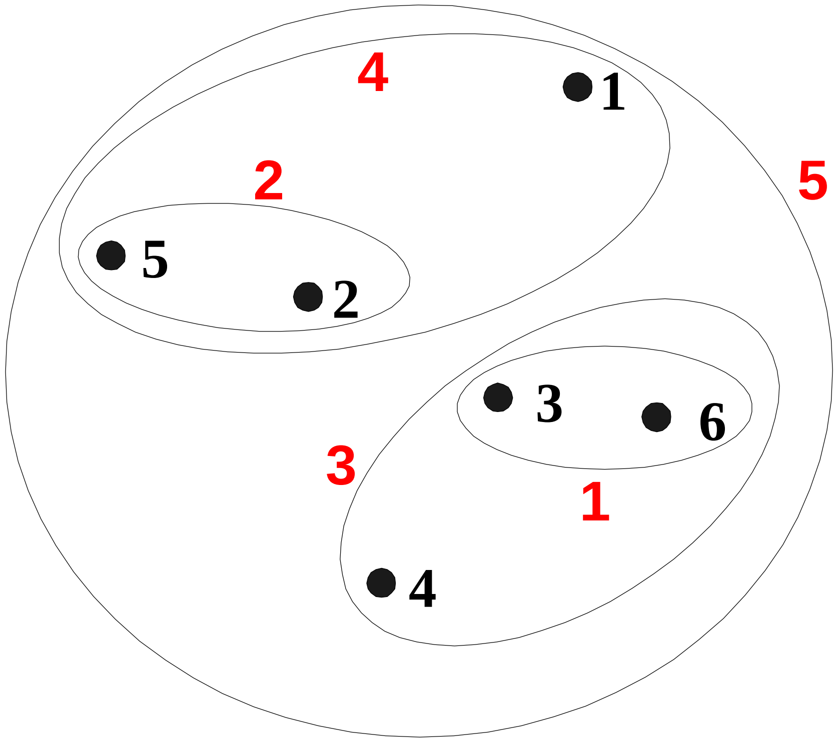
Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

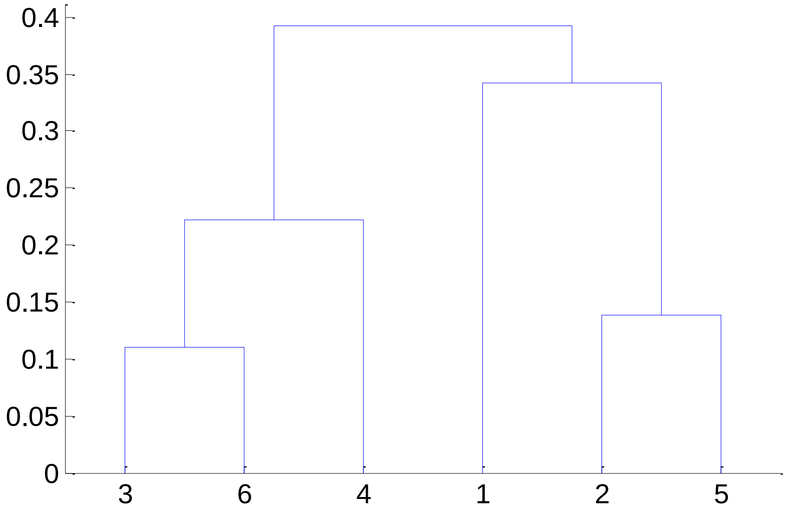
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

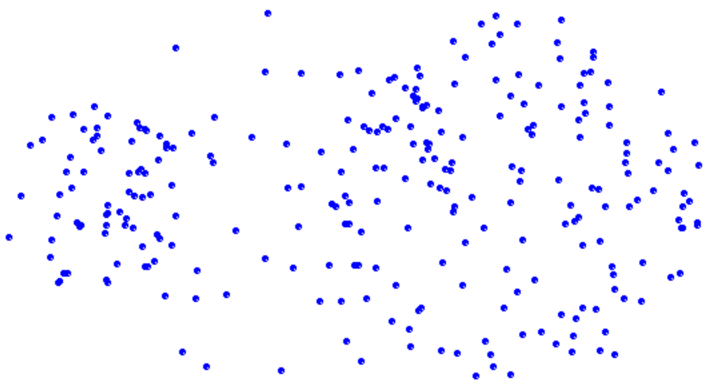


Nested Clusters

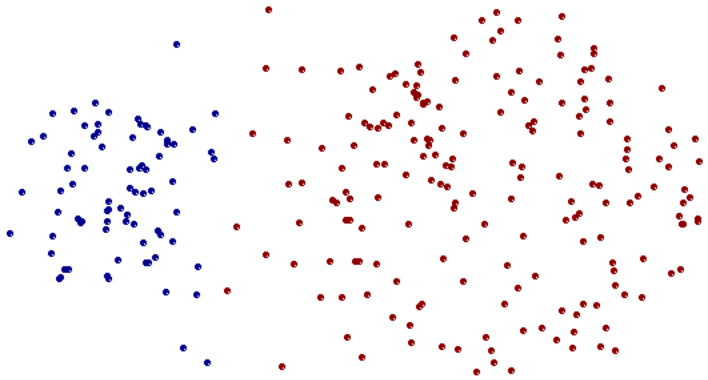


Dendrogram

Strength of MAX



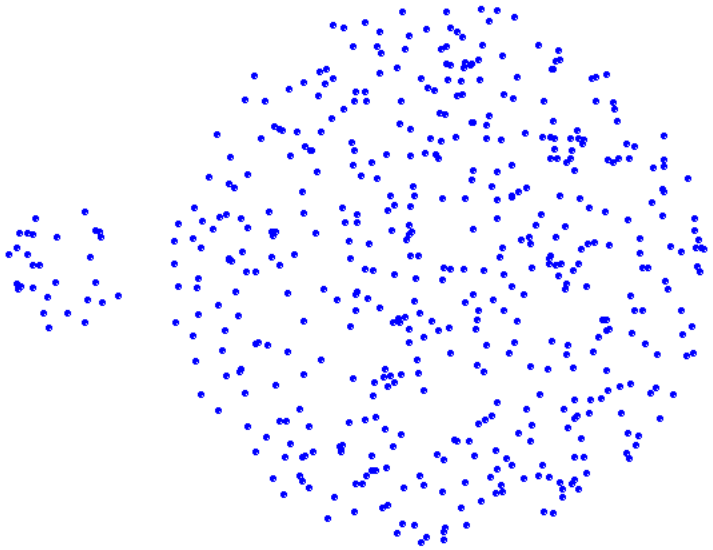
Original Points



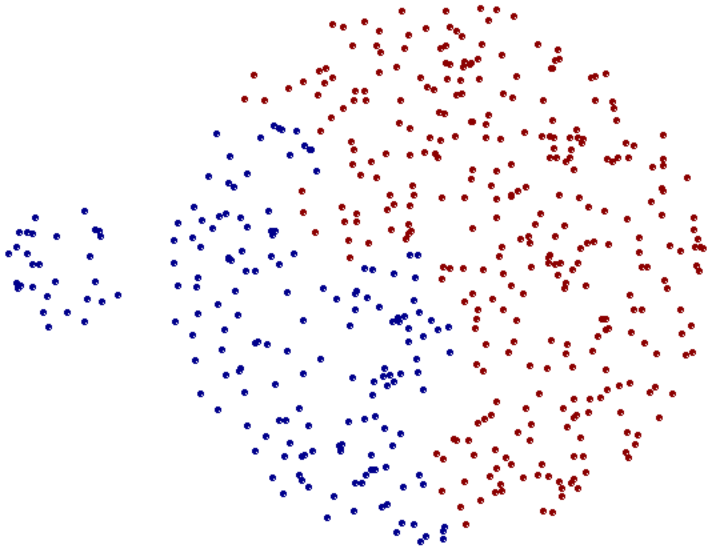
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

- **Tends to break large clusters**
- **Biased towards globular clusters**

Example of Single Link and Complete Link

Proximity matrix

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 2.3 & 3.4 & 1.2 & 3.7 \\ 2 & & 0 & 2.0 & 1.8 & 2.2 \\ 3 & & & 0 & 4.2 & 0.7 \\ 4 & & & & 0 & 4.4 \\ 5 & & & & & 0 \end{array}$$

Single-Link

$$\begin{array}{c|cccc} & 1 & 2 & 3,5 & 4 \\ \hline 1 & 0 & 2.3 & 3.4 & 1.2 \\ 2 & & 0 & 2.0 & 1.8 \\ 3,5 & & & 0 & 4.2 \\ 4 & & & & 0 \end{array}$$

$$\begin{array}{c|ccc} & 1,4 & 2 & 3,5 \\ \hline 1,4 & 0 & 1.8 & 3.4 \\ 2 & & 0 & 2.0 \\ 3,5 & & & 0 \end{array}$$

$$\begin{array}{c|cc} & 1,2,4 & 3,5 \\ \hline 1,2,4 & 0 & 2.0 \\ 3,5 & & 0 \end{array}$$

Complete-Link

$$\begin{array}{c|cccc} & 1 & 2 & 3,5 & 4 \\ \hline 1 & 0 & 2.3 & 3.7 & 1.2 \\ 2 & & 0 & 2.2 & 1.8 \\ 3,5 & & & 0 & 4.4 \\ 4 & & & & 0 \end{array}$$

$$\begin{array}{c|ccc} & 1,4 & 2 & 3,5 \\ \hline 1,4 & 0 & 2.3 & 4.4 \\ 2 & & 0 & 2,2 \\ 3,5 & & & 0 \end{array}$$

$$\begin{array}{c|cc} & 1,4 & 2,3,5 \\ \hline 1,4 & 0 & 4.4 \\ 2,3,5 & & 0 \end{array}$$

HW: Hierarchical Clustering Example & Implementation

- Hierarchical Clustering with Python and Scikit-Learn
- Scikit-Learn - Hierarchical Clustering
- A Beginner’s Guide to Hierarchical Clustering and how to Perform it in Python

Study Materials of Clustering

Lecture Notes for Chapter 7, Introduction to Data Mining by Tan, Steinbach, Kumar

An Introduction to Clustering and different methods of clustering

Getting your clustering right

JavaTpoint: Clustering in Machine Learning, K-Means Clustering Algorithm, Hierarchical Clustering