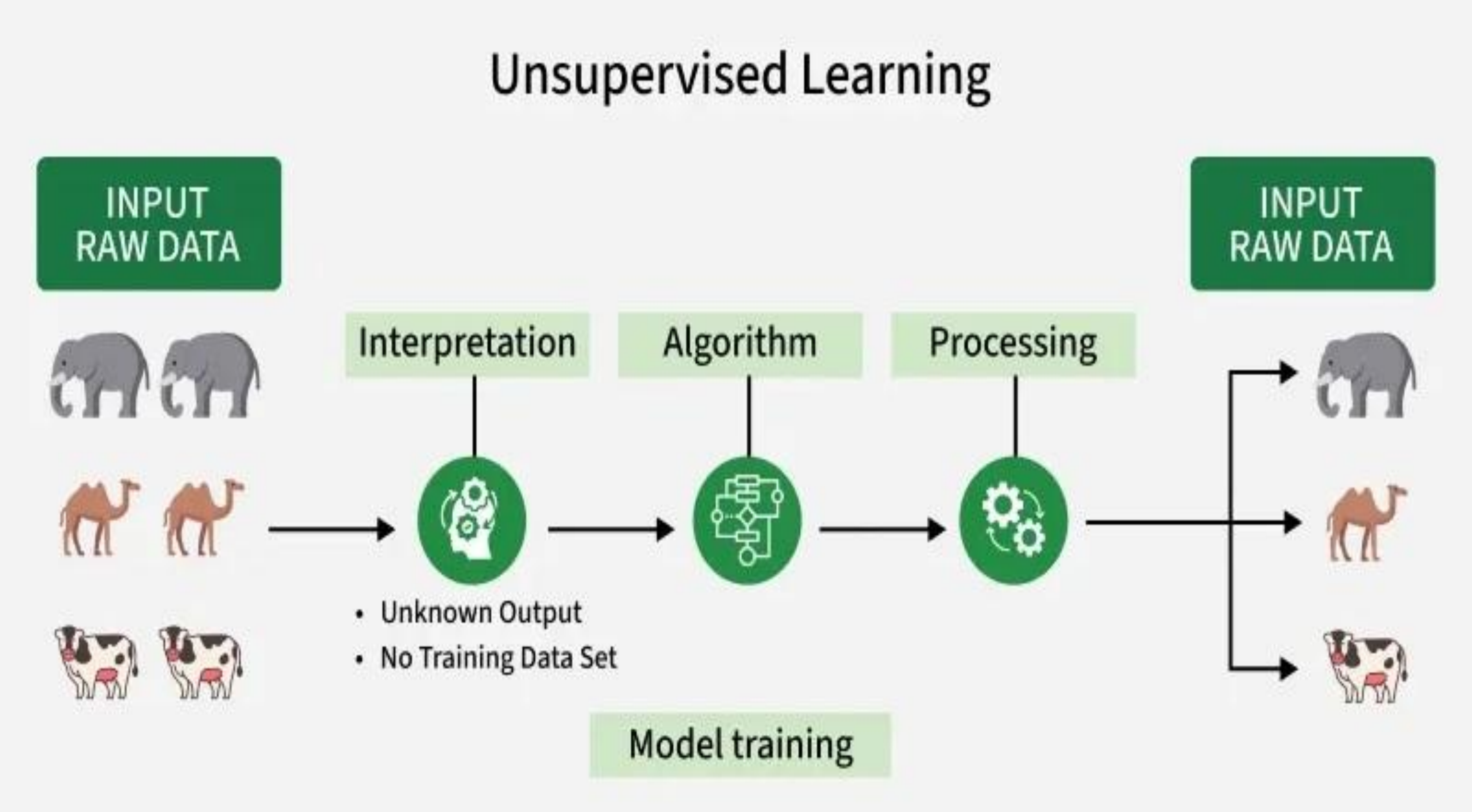
Un-Supervised Learning

Sachin Tripathi

IIT(ISM), Dhanbad

Unsupervised Learning



Working of Unsupervised Learning

❑ **Collect Unlabeled Data**

Gather a dataset without predefined labels or categories.

❑ **Select an Algorithm**

Choose a suitable unsupervised algorithm such as clustering like K-Means, association rule learning like Apriori or dimensionality reduction like PCA based on the goal.

❑ **Train the Model on Raw Data**

Feed the entire unlabeled dataset to the algorithm.
The algorithm looks for similarities, relationships or hidden structures within the data.

❑ Group or Transform Data

The algorithm organizes data into groups (clusters), rules or lower-dimensional forms without human input.

❑ Interpret and Use Results

Analyze the discovered groups, rules or features to gain insights or use them for further tasks like visualization, anomaly detection or as input for other models.

Unsupervised Learning Algorithms

- ❑ Clustering Algorithms
- ❑ Association Rule Learning
- ❑ Dimensionality Reduction

Clustering

- ❑ Clustering is an unsupervised machine learning technique that groups unlabeled data into clusters based on similarity. Its goal is to discover patterns or relationships within the data without any prior knowledge of categories or labels.
- ❑ Groups data points that share similar features or characteristics.
- ❑ Helps find natural groupings in raw, unclassified data.
- ❑ Commonly used for customer segmentation, anomaly detection and data organization.
- ❑ Works purely from the input data without any output labels.

Common clustering algorithms:

- ❑ **K-mean clustering:** Groups data into K clusters based on how close the points are to each other.
- ❑ **Hierarchical Clustering:** Creates clusters by building a tree step-by-step, either merging or splitting groups.
- ❑ **Density Based Clustering (DBSCAN):**Finds clusters in dense areas and treats scattered points as noise.
- ❑ **Mean Shift Clustering:** Discovers clusters by moving points toward the most crowded areas.
- ❑ **Spectral Clustering:** Groups data by analyzing connections between points using graphs.

Association Rule Learning

- Association rule learning is a rule-based unsupervised learning technique used to discover interesting relationships between variables in large datasets.
- It identifies patterns in the form of “if-then” rules, showing how the presence of some items in the data implies the presence of others.

- Finds frequent item combinations and the rules connecting them.
- Commonly used in market basket analysis to understand product purchase relationships.
- Helps retailers design promotions and cross-selling strategies.

Association Rule Learning algorithms:

- ❑ Apriori Algorithms: Finds patterns by exploring frequent item combinations step-by-step.
- ❑ FP-Growth Algorithm: An Efficient Alternative to Apriori. It quickly identifies frequent patterns without generating candidate sets.
- ❑ E-Clat Algorithms: Uses intersections of itemsets to efficiently find frequent patterns.
- ❑ Efficient Tree Based Algorithm : Scales to handle large datasets by organizing data in tree structures.

Dimensionality Reduction

- Dimensionality Reduction is the process of decreasing the number of features or variables in a dataset while retaining as much of the original information as possible.
- This technique helps simplify complex data making it easier to analyze and visualize. It also improves the efficiency and performance of machine learning algorithms by reducing noise and computational cost.
- It reduces the dataset’s feature space from many dimensions to fewer, more meaningful ones.
- Helps focus on the most important traits or patterns in the data.
- Commonly used to improve model speed and reduce overfitting.

Common Algorithm

- ❑ Principal Component Analysis (PCA): Reduces dimensions by transforming data into uncorrelated principal components.
- ❑ Linear Discriminant Analysis (LDA): Reduces dimensions while maximizing class separability for classification tasks.
- ❑ Non-negative Matrix Factorization (NMF): Breaks data into non-negative parts to simplify representation.
- ❑ Locally Linear Embedding (LLE): Reduces dimensions while preserving the relationships between nearby points.
- ❑ Isomap: Captures global data structure by preserving distances along a manifold.

Applications of Unsupervised learning

- ❑ Customer Segmentation: Algorithms cluster customers based on purchasing behavior or demographics, enabling targeted marketing strategies.
- ❑ Anomaly Detection: Identifies unusual patterns in data, aiding fraud detection, cybersecurity and equipment failure prevention.
- ❑ Recommendation Systems: Suggests products, movies or music by analyzing user behavior and preferences.
- ❑ Image and Text Clustering: Groups similar images or documents for tasks like organization, classification or content recommendation.
- ❑ Social Network Analysis: Detects communities or trends in user interactions on social media platforms.

Advantages

- No need for labeled data: Works with raw, unlabeled data hence saving time and effort on data annotation.
- Discovers hidden patterns: Finds natural groupings and structures that might be missed by humans.
- Handles complex and large datasets: Effective for high-dimensional or vast amounts of data.
- Useful for anomaly detection: Can identify outliers and unusual data points without prior examples.

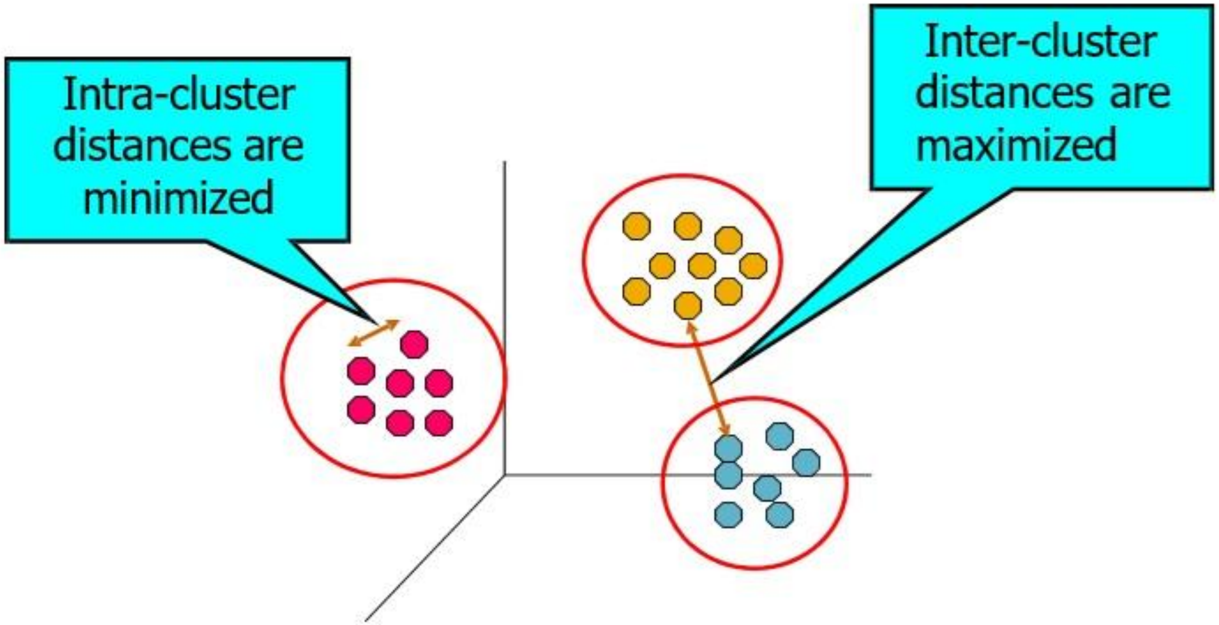
Challenges

- ❑ Noisy Data: Outliers and noise can distort patterns and reduce the effectiveness of algorithms.
- ❑ Assumption Dependence: Algorithms often rely on assumptions (e.g., cluster shapes) which may not match the actual data structure.
- ❑ Overfitting Risk: Overfitting can occur when models capture noise instead of meaningful patterns in the data.
- ❑ Limited Guidance: The absence of labels restricts the ability to guide the algorithm toward specific outcomes.

- Cluster Interpretability: Results such as clusters may lack clear meaning or alignment with real-world categories.
- Sensitivity to Parameters: Many algorithms require careful tuning of hyperparameters such as the number of clusters in k-means.
- Lack of Ground Truth: Unsupervised learning lacks labeled data making it difficult to evaluate the accuracy of results.

Clustering Algorithms

- ❑ In general a grouping of objects such that the objects in a group (cluster) are similar (or related) to one another and different from (or unrelated to) the objects in other groups


Intra-cluster distances are minimized
Inter-cluster distances are maximized

❑ Clustering is an unsupervised learning problem.

❑ Key objective is to identify distinct groups (called clusters) based on some notion of similarity within a given dataset.

❑ The most popularly used clustering techniques are k-means (divisive) and hierarchical (agglomerative).

Challenges in clustering

- Scalability - We need highly scalable clustering algorithms to deal with large databases.

- Ability to deal with different kind of attributes -Algorithms should be capable to be applied on any kind of data such as interval based (numerical) data,categorical, binary data.

- Discovery of clusters with attribute shape – The clustering algorithm should be capable of detect cluster of arbitrary shape.

- It should not be bounded to only distance measures that tend to find spherical cluster of small size.

- ❑ High dimensionality - The clustering algorithm should not only be able to handle low- dimensional data but also the high dimensional space.

- ❑ Ability to deal with noisy data - Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.

- ❑ Interpretability - The clustering results should be interpretable, comprehensible and usable.

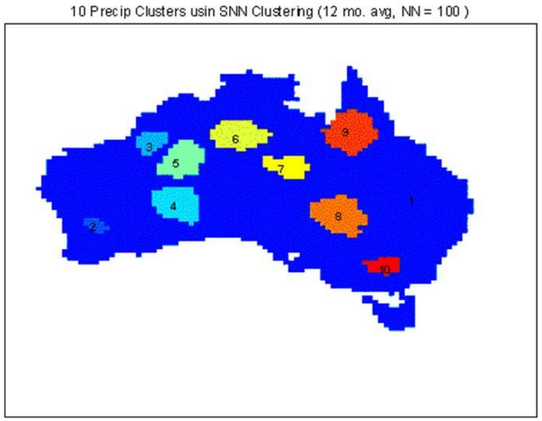
Applications

- ❑ Pattern Recognition
- ❑ Spatial Data Analysis
- ❑ Image Processing
- ❑ Economic Science (especially market research)
- ❑ Crime analysis
- ❑ Bio informatics
- ❑ Medical Imaging
- ❑ Robotics
- ❑ Climatology

Applications of Cluster Analysis

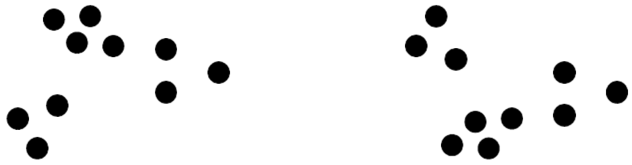
- Understanding
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

- Summarization
  - Reduce the size of large data sets

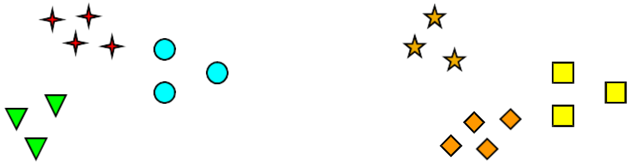

10 Precip Clusters usin SNN Clustering (12 mo. avg, NN = 100)
1
2
3
4
5
6
7
8
9
10

Clustering precipitation in Australia

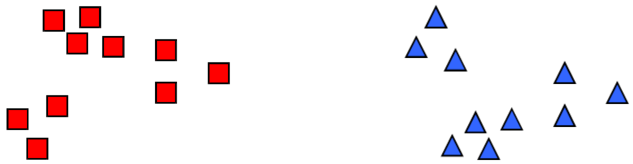
Notion of Cluster Can be Ambiguous



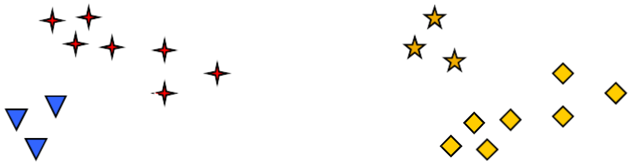
How many clusters?



Six Clusters



Two Clusters

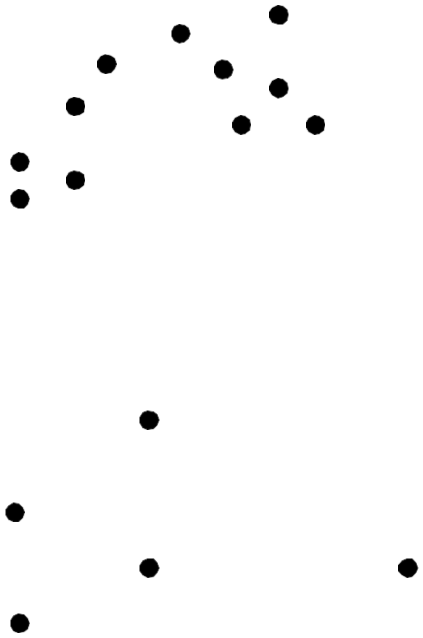


Four Clusters

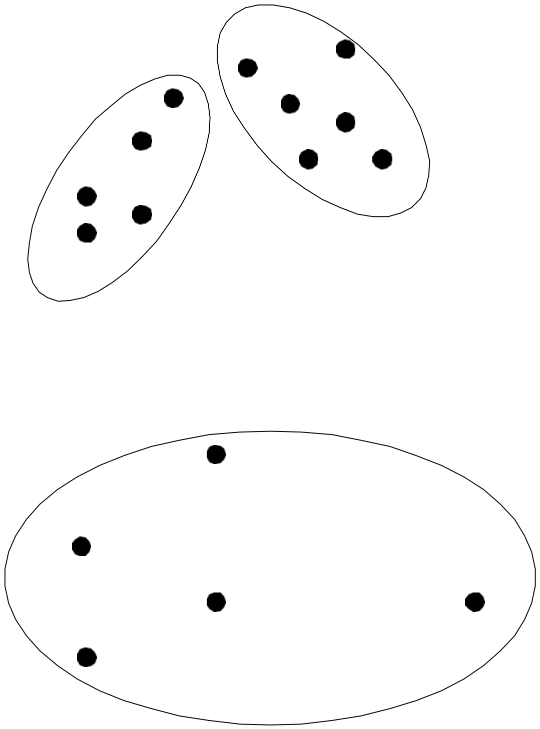
Types of Clustering

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

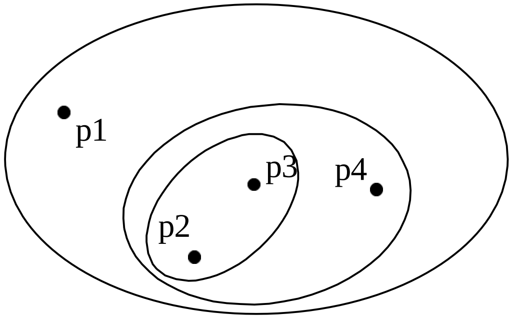


Original Points

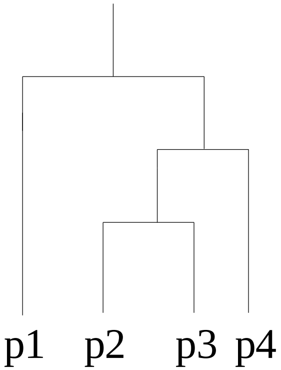


A Partitional Clustering

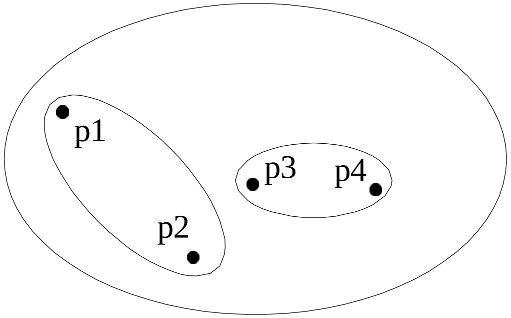
Hierarchical Clustering



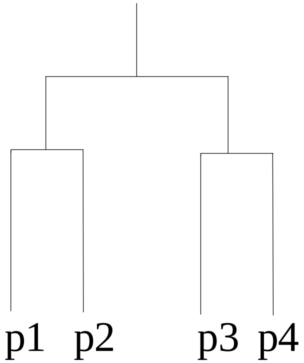
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



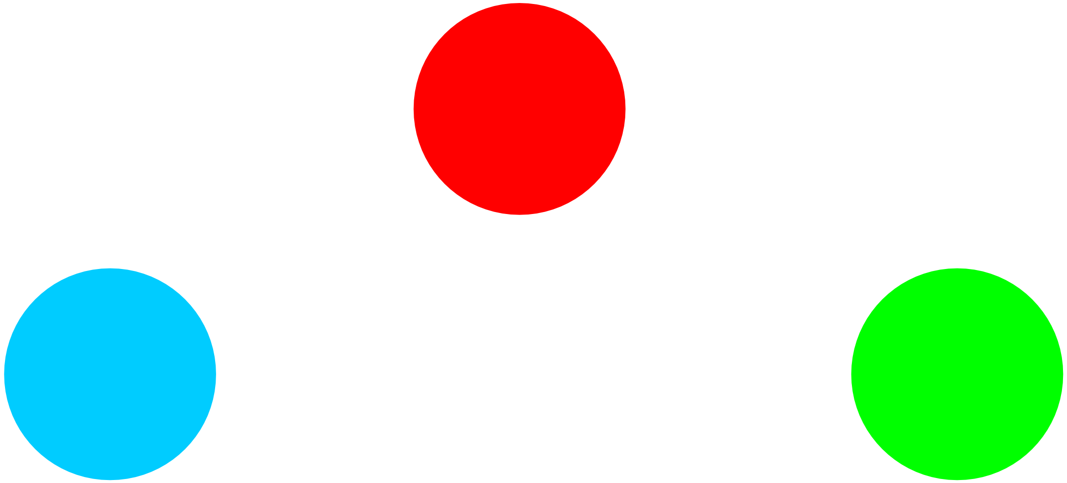
Non-traditional Dendrogram

Other Types of Clustering

- Exclusive (or non-overlapping) versus non- exclusive (or overlapping)
  - In non-exclusive clusterings, points may belong to multiple clusters.
    - Points that belong to multiple classes, or ‘border’ points
- Fuzzy (or soft) versus non-fuzzy (or hard)
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
    - Weights usually must sum to 1 (often interpreted as probabilities)
- Partial versus complete
  - In some cases, we only want to cluster some of the data

Types of   Cluster : Well-Separated

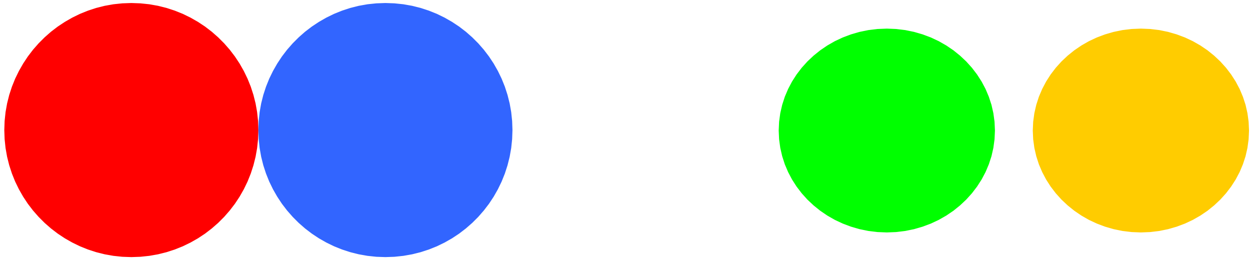
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Cluster : Center Based

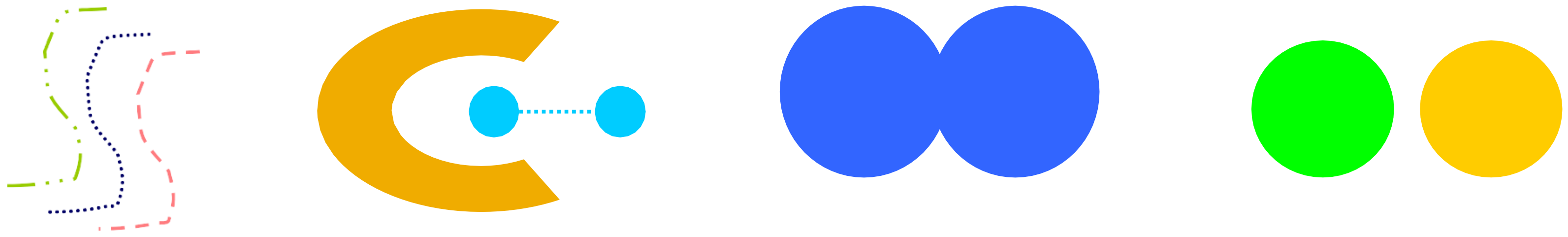
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the minimizer of distances from all the points in the cluster, or a medoid,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



4 center-based clusters

Types of Cluster : Contiguous Cluster

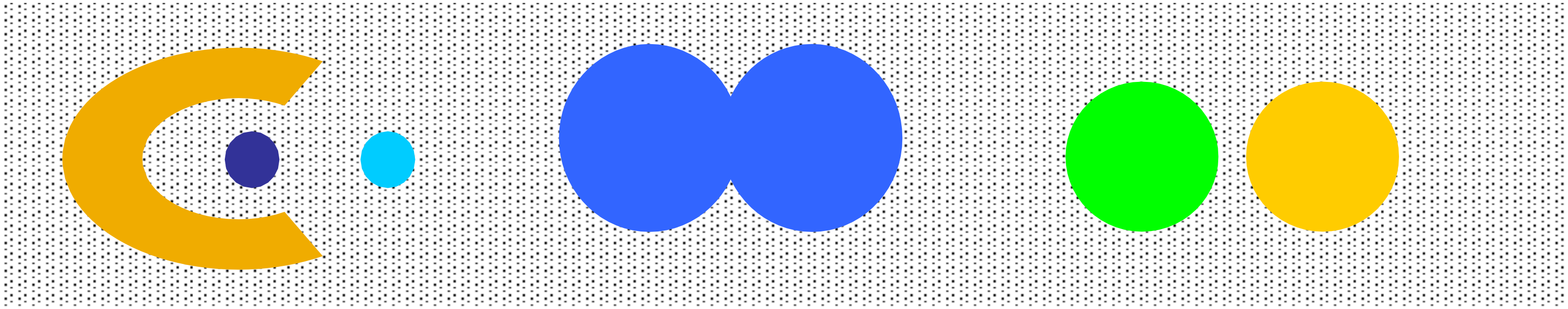
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Cluster : Density Based

- Density-based
  - 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.

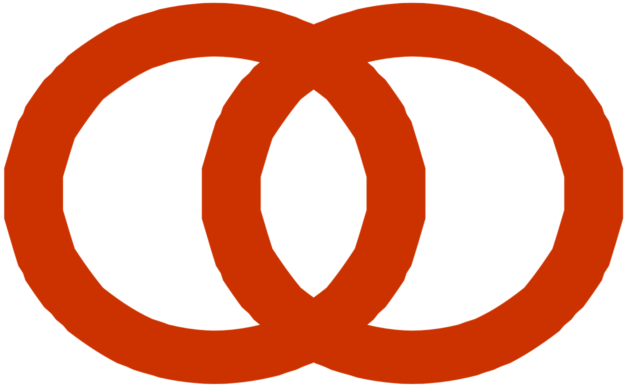


6 density-based clusters

Types of Cluster : Conceptual Cluster

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

.



2 Overlapping Circles

Objective Function

- Clustering as an optimization problem
  - Finds clusters that minimize or maximize an objective function.
  - Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering algorithms typically have local objectives
    - Partitional algorithms typically have global objectives

- ❑ A variation of the global objective function approach is to fit the data to a parameterized model.
  - The parameters for the model are determined from the data, and they determine the clustering
  - E.g., Mixture models assume that the data is a ‘mixture' of a number of statistical distributions.

K-Means Clustering

- The key objective of a k-means algorithm is to organize data into clusters such that there is high intra-cluster similarity and low inter-cluster similarity.
- An item will only belong to one cluster, not several, that is, it generates a specific number of disjoint, nonhierarchical clusters.
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid

- Number of clusters, K, must be specified

- The objective is to minimize the sum of distances of the points to their respective centroid

- Problem: Given a set X of n points in a d- dimensional space and an integer K group the points into K clusters C= {C_1, C_2,…,C_k} such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, c)$$

is minimized, where c_i is the centroid of the points in cluster C_i

- Most common definition is with euclidean distance, minimizing the Sum of Squared Error (SSE) function
  - Sometimes K-means is defined like that

- Problem: Given a set X of n points in a d-dimensional space and an integer K group the points into K clusters C= {C_1, C_2,…,C_k} such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

is minimized, where c_i is the mean of the points in cluster C_i

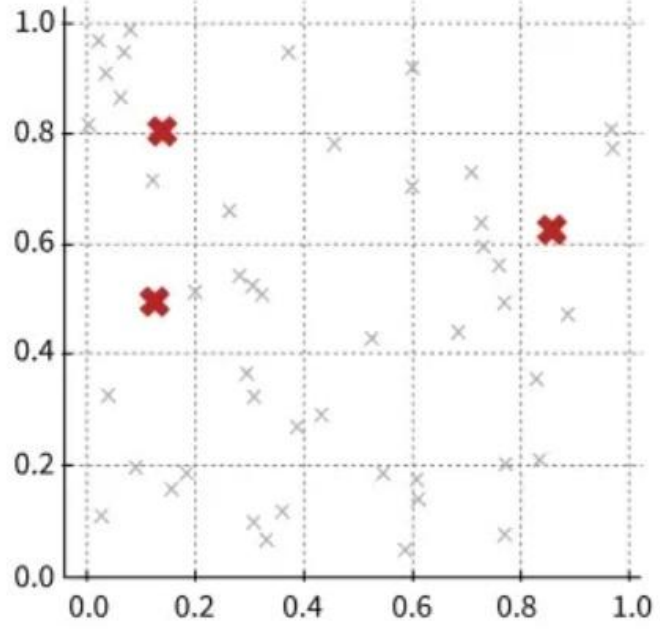
Sum of Squared Error (SSE)

K-Means Algorithm

- The algorithm will categorize the items into "k" groups or clusters of similarity.
- To calculate that similarity Euclidian Distance as a measurement. The algorithm works as follows:
- **Initialization:** We begin by randomly selecting k cluster centroids.
- **Assignment Step:** Each data point is assigned to the nearest centroid, forming clusters.
- **Update Step:** After the assignment, we recalculate the centroid of each cluster by averaging the points within it.
- **Repeat:** This process repeats until the centroids no longer change or the maximum number of iterations is reached.

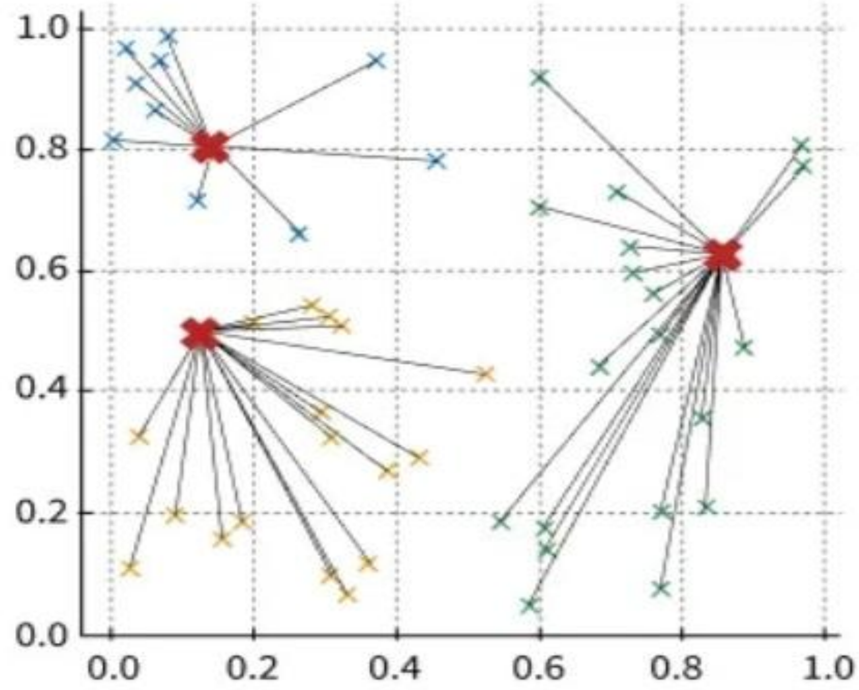
Choose Initial Centroids

- Centroids are randomly chosen from the data points. These represents the initial cluster centre



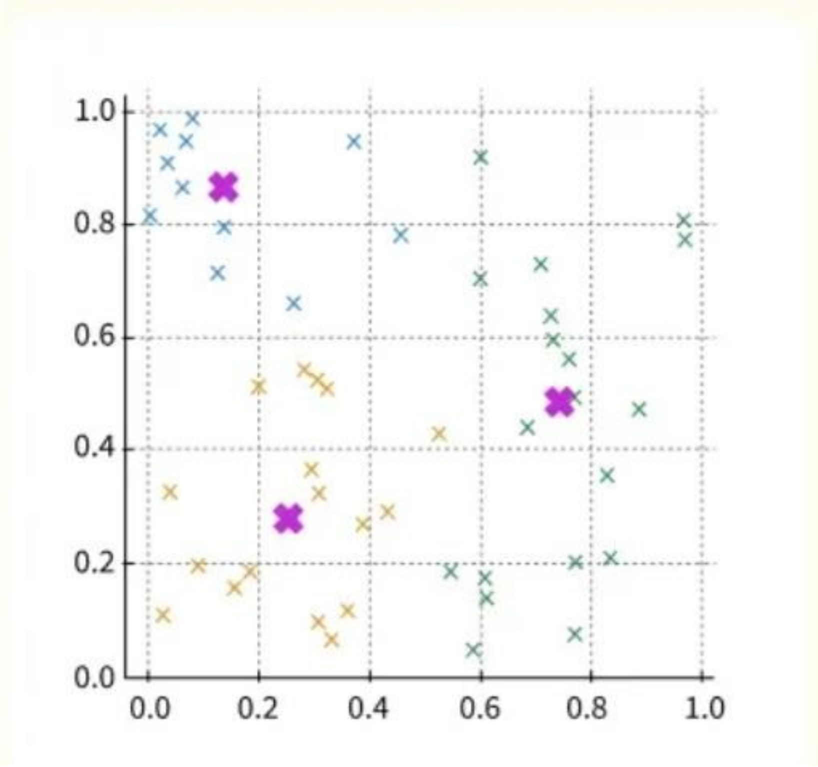
Assign Points to Nearest Centroid

- Each point is assigned to the nearest centroid, forming cluster



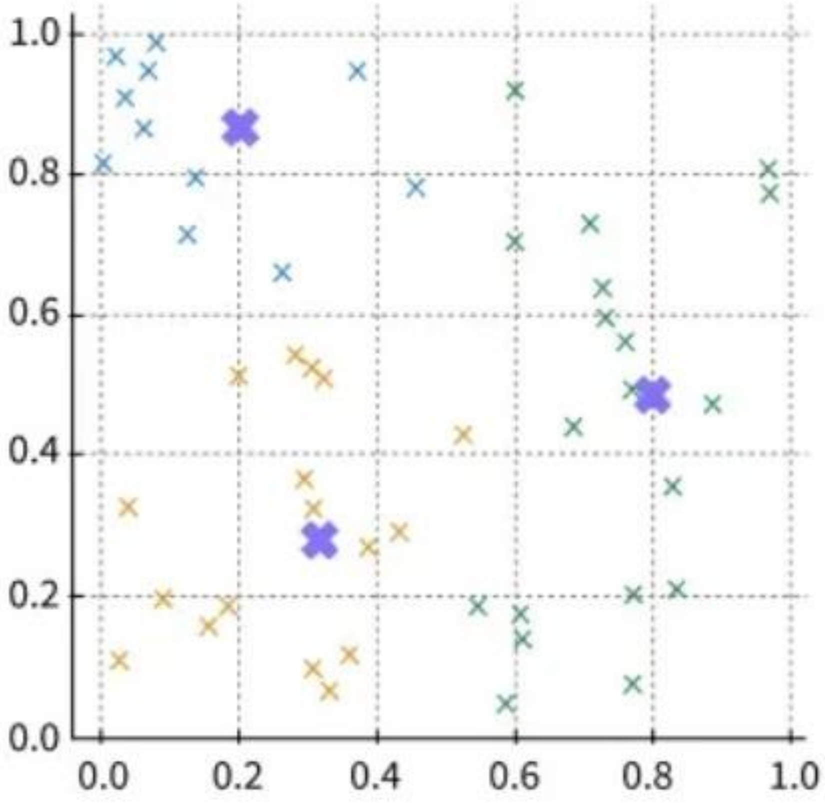
Update the Centroids

- Centroids are recalculated as the mean of the points in each cluster



Repeat Until Convergence

- This process repeats until the centroids stabilize and do not move further



Why Use K-Means Clustering?

❑ Data Segmentation: One of the most common uses of K-Means is segmenting data into distinct groups. For example, businesses use K-Means to group customers based on behavior, such as purchasing patterns or website interaction.

❑ Image Compression: K-Means can be used to reduce the complexity of images by grouping similar pixels into clusters, effectively compressing the image. This is useful for image storage and processing.

❑ **Document Clustering:** In natural language processing (NLP), K-Means is used to group similar documents or articles together. It’s often used in applications like recommendation systems or news categorization.

❑ **Organizing Large Datasets:** When dealing with large datasets, K-Means can help in organizing the data into smaller, more manageable chunks based on similarities, improving the efficiency of data analysis.

❑ Anomaly Detection: K-Means can be applied to detect anomalies or outliers by identifying data points that do not belong to any of the clusters.

K-means Algorithm- Initialization

- Initial centroids are often chosen randomly.
- Clusters produced vary from one run to another.

Importance of Choosing Initial Centroid

- The **choice of initial centroids in K-Means** is **very important**, because it can affect:
  - The **final cluster results**
  - The **speed of convergence**
  - Whether the algorithm finds the **global or a local optimum**.

Example Illustration

- ❑ Suppose given 1D data points:2, 4, 6, 8, 10

  Assume K = 2 clusters.

Case 1 — Good centroid initialization
Initial centroids: **2** and **10**
•Cluster 1 → near 2 → {2, 4, 6}
•Cluster 2 → near 10 → {8, 10}
Final centroids → (2+4+6)/3 = 4, (8+10)/2 = 9

➢ **Clear separation** of two clusters.

Case 2 — Poor centroid initialization

Initial centroids: **4** and **6**

Iteration 1:

•Points {2, 4} near 4

•Points {6, 8, 10} near 6

New centroids → (2+4)/2 = 3, (6+8+10)/3 = 8

After updates → Clusters {2,4,6} and {8,10}, but it took **extra iterations**, and sometimes results may even **differ**.

What this shows

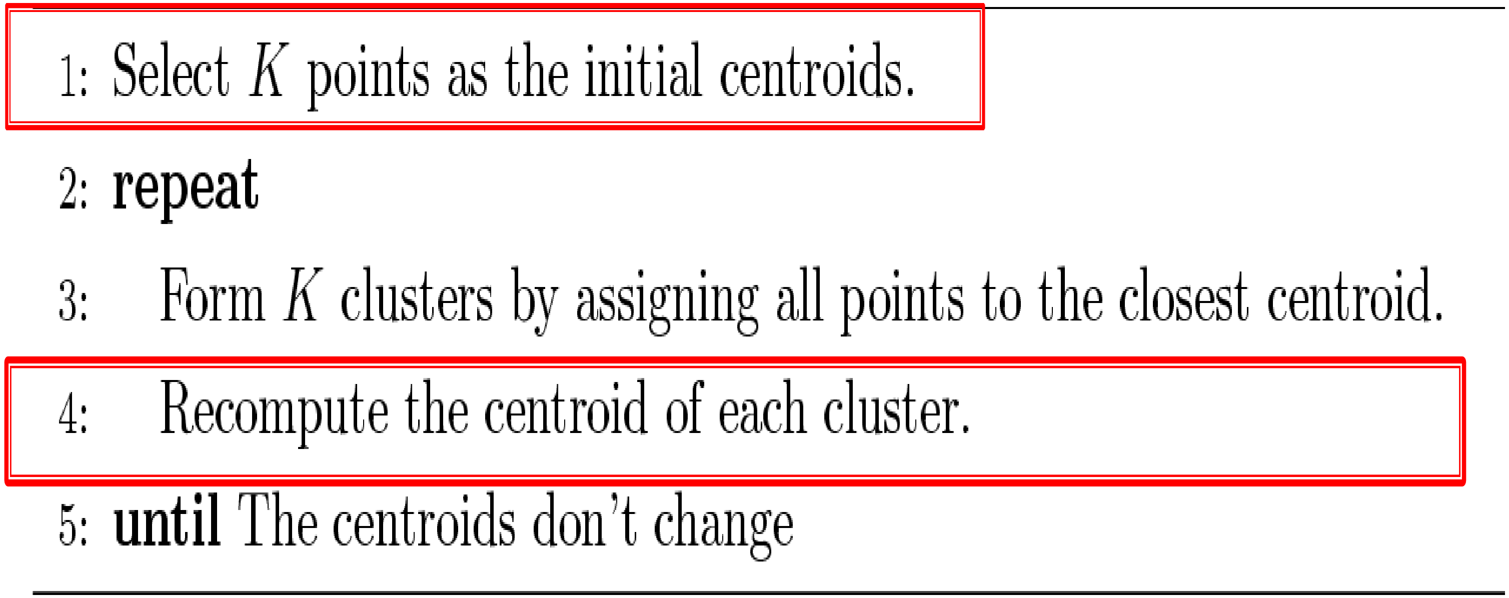
- Different initial centroids → **different clustering results**
- Poor choice may cause:
  - **Slow convergence**
  - **Wrong cluster boundaries**
  - **Local minima**

Complexity of the K-means Problem

- NP-hard if the dimensionality of the data is at least 2 (d>=2)
  - Finding the best solution in polynomial time is infeasible
- For d=1 the problem is solvable in polynomial time (how?)
- A simple iterative algorithm works quite well in practice

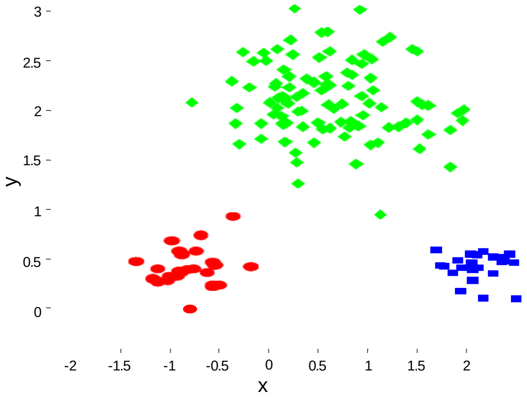
K-means Algorithm

- Also known as Lloyd’s algorithm.
- K-means is sometimes synonymous with this algorithm

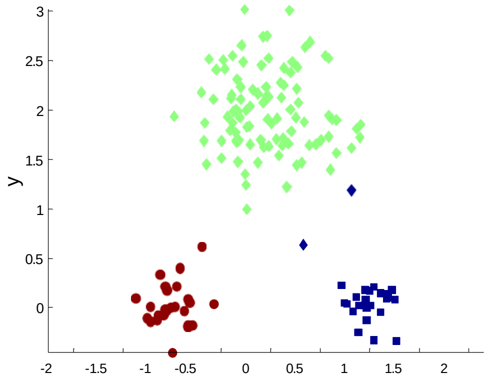


```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don’t change
```

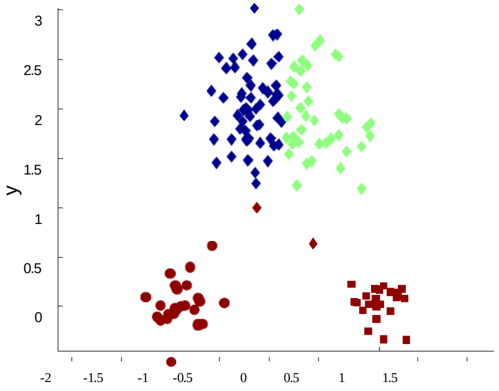
Two different K-means Clustering



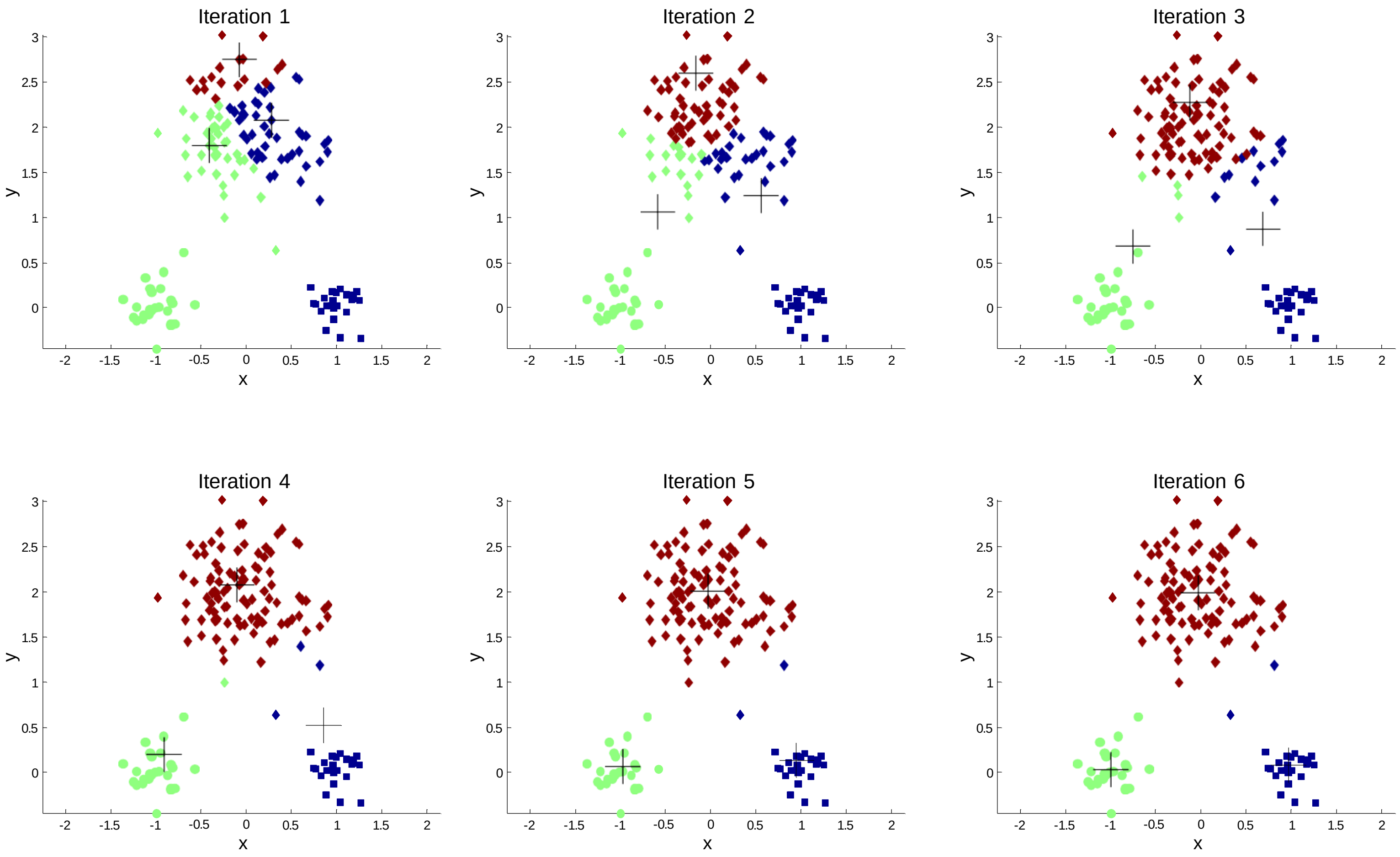
Original Points



Optimal Clustering



Sub-optimal Clustering



Iteration 1

Iteration 2

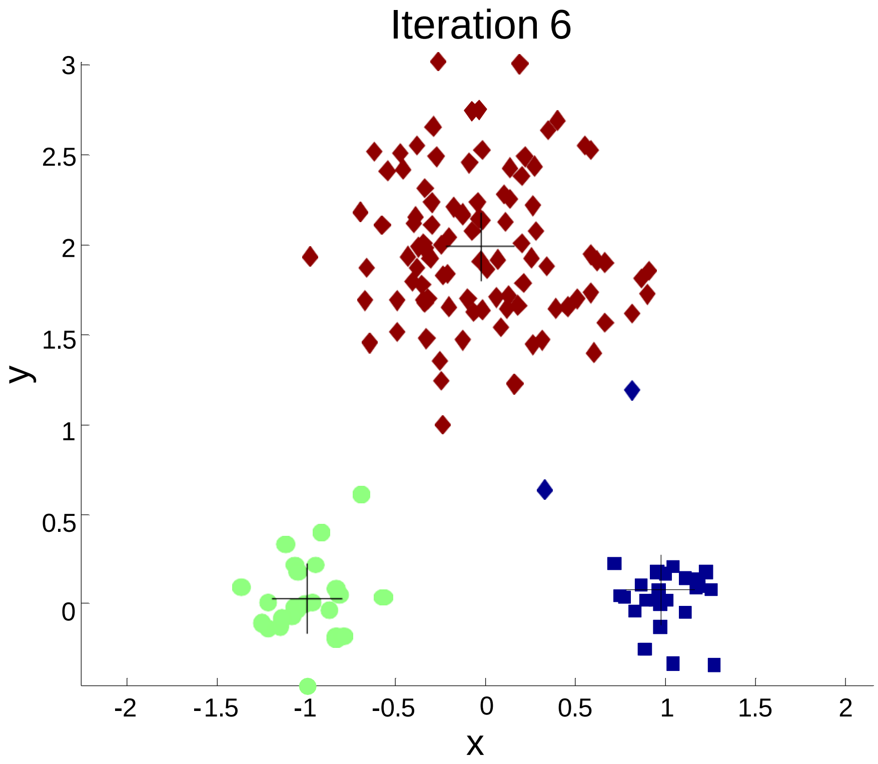
Iteration 3

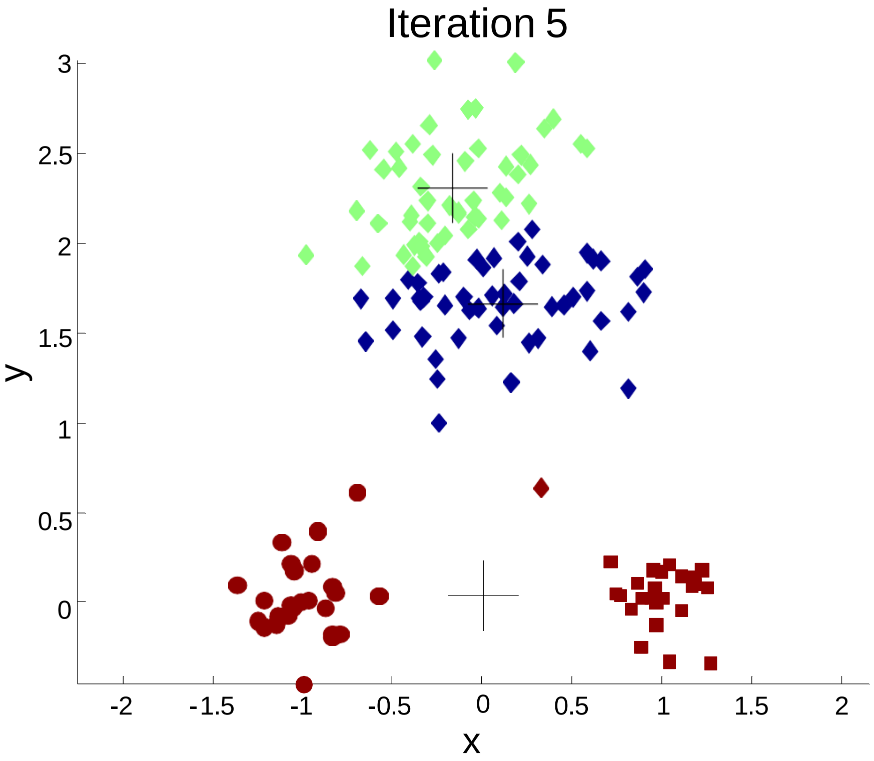
Iteration 4

Iteration 5

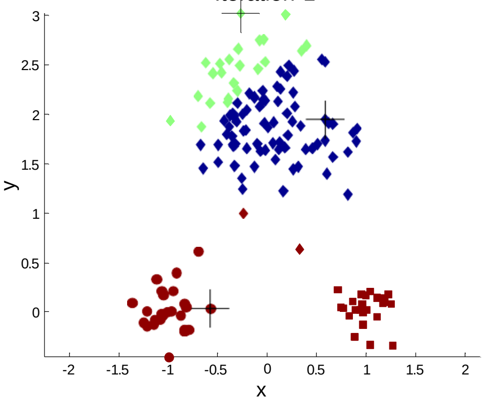
Iteration 6

Importance of Choosing Centroids

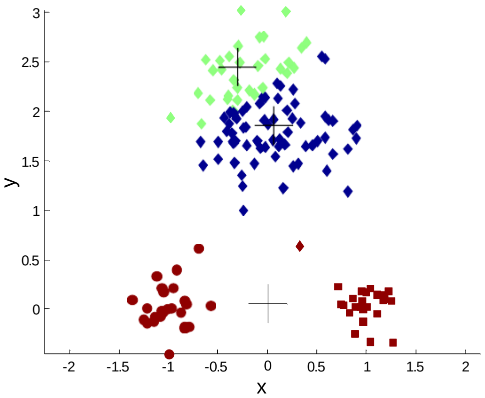



Iteration 5
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

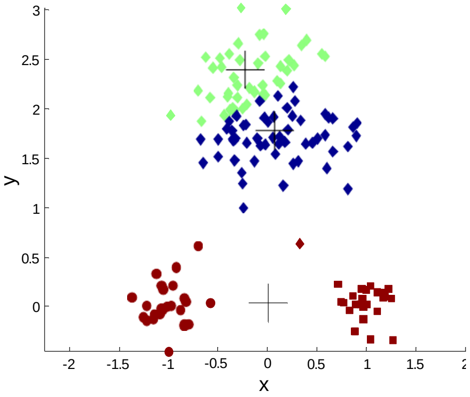
Iteration 1



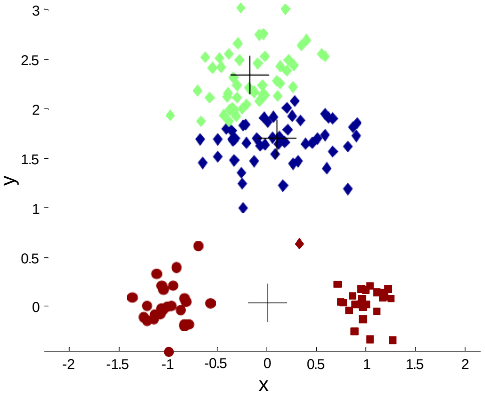
Iteration 2



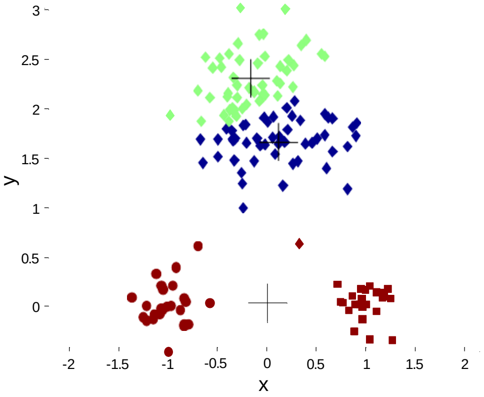
Iteration 3



Iteration 4



Iteration 5



Dealing with Initialization

- Do multiple runs and select the clustering with the smallest error

- Select original set of points by methods other than random . E.g.,pick the most distant (from each other) points as cluster centers (K-means++ algorithm)

K-means Algorithm- Centroids

- The centroid depends on the distance function. The minimizer for the distance function
- ‘Closeness’ is measured by Euclidean distance (SSE), cosine similarity, correlation, etc.
- Centroid:
  - The mean of the points in the cluster for SSE, and cosine similarity
  - The median for Manhattan distance.
- Finding the centroid is not always easy
  - It can be an NP-hard problem for some distance functions
    - E.g., median form multiple dimensions

K-means Algorithm- Convergence

- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations. Often the stopping condition is changed to ‘Until relatively few points change clusters’

- Complexity is O(n * K * I * d)

  n = number of points, K = number of clusters, I = number of iterations, d = dimensionality
- In general a fast and efficient algorithm

Limitations of K-means

- K-means has problems when clusters are of different Sizes, Densities, Non-globular shapes

- K-means has problems when the data contains outliers.

Suppose a cluster has points:[1,2,2,3,50]
The centroid (mean) = 1+2+2+3+505=11.6

- Notice how one outlier (50) pulled the centroid far away from the main cluster around 2–3.This shifts the boundary and causes incorrect clustering.

❑ Different Sizes

K-Means assumes all clusters are roughly equal in size.
If one cluster is much larger than another, the centroid of the large cluster dominates and pulls nearby points—even those belonging to the small cluster.

Example:

•One cluster has 100 points (large), another has 10 (small).
•K-Means centroid of the big cluster pulls the boundary closer, so small cluster points may be wrongly assigned.

Result: Small clusters get absorbed by large ones.

❑ Different Densities

K-Means assumes uniform density across clusters.
If one cluster is dense (points close together) and another is sparse (points far apart), the sparse cluster’s points may get split or mixed with others.

Example:

- Dense cluster near (1,1)
- Sparse cluster near (10,10)

- Because K-Means minimizes *average distance*, it may break the sparse cluster into parts.

Result: Uneven clustering — dense cluster identified correctly, sparse cluster fragmented.

❑ Non-globular (Non-spherical) Shapes

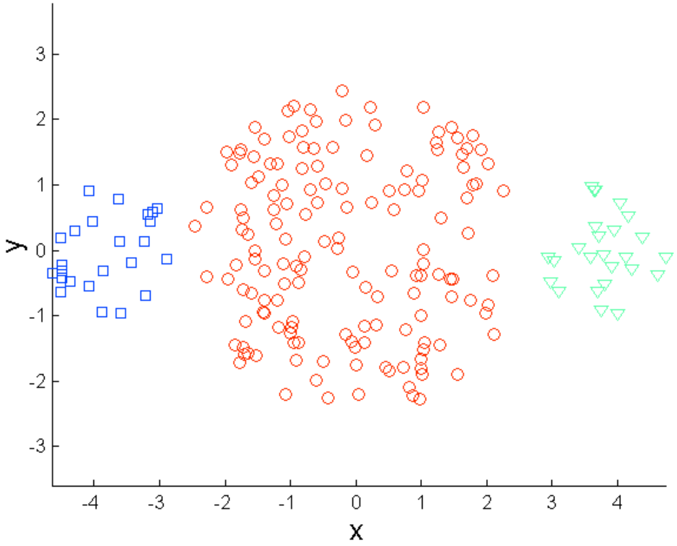
K-Means works best when clusters are spherical or circular (e.g., blobs).
It uses Euclidean distance, which cannot handle curved or elongated clusters.
Example:
Imagine two “crescent moon” or “ring” shaped clusters (like two half circles).
Even if they’re well-separated, K-Means draws straight-line boundaries, not curved ones, and will mix the two crescents.

Result: Incorrect cluster boundaries.

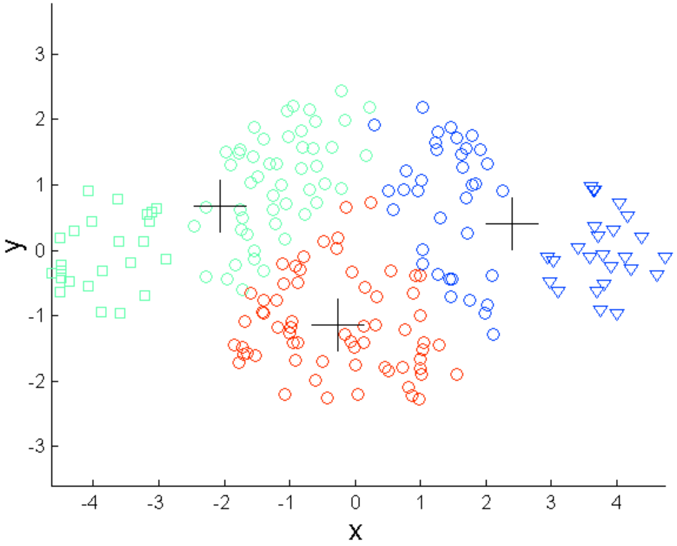
Summary Table

Problem Type	Why K-Means Fails	Result
Different Sizes	Larger clusters pull centroid boundaries	Small clusters merged
Different Densities	Sparse clusters split; dense ones dominate	Fragmented or merged clusters
Non-globular Shapes	Straight-line boundaries can’t fit complex shapes	Wrong cluster assignment

Differing Size

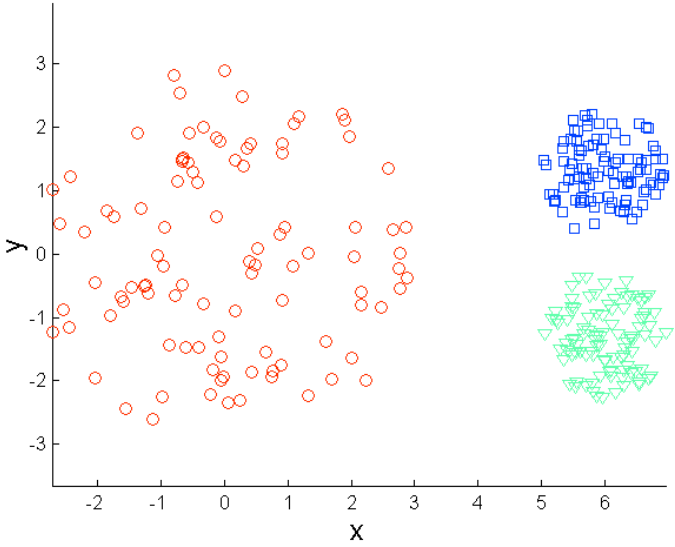


Original Points

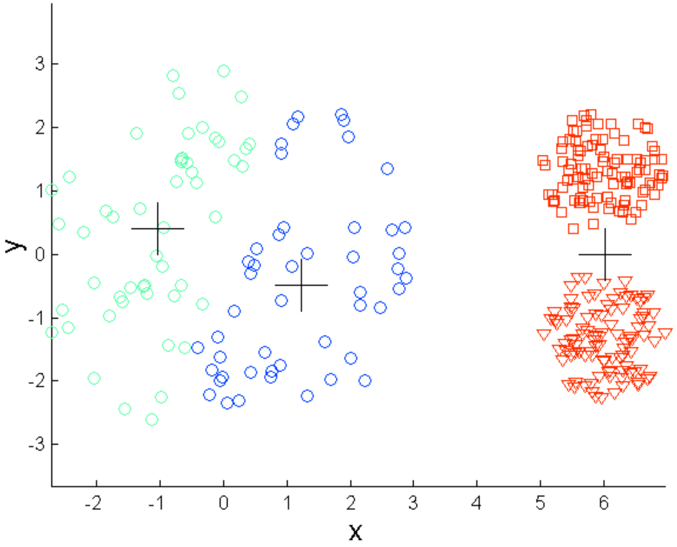


K-means (3 Clusters)

Differing Density

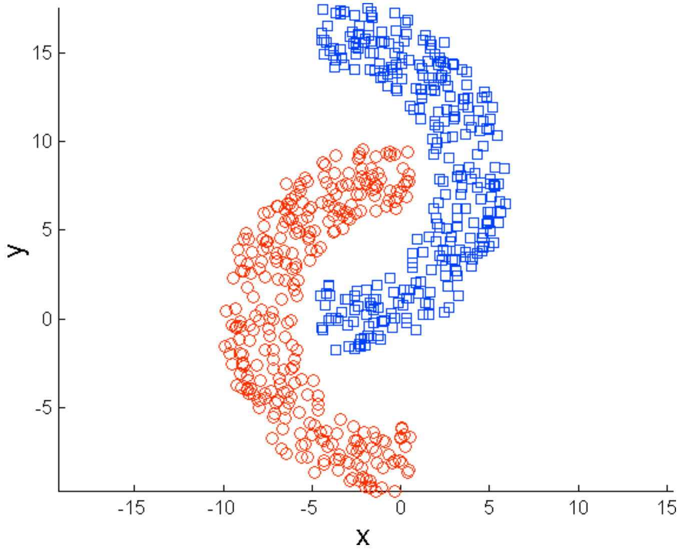


Original Points

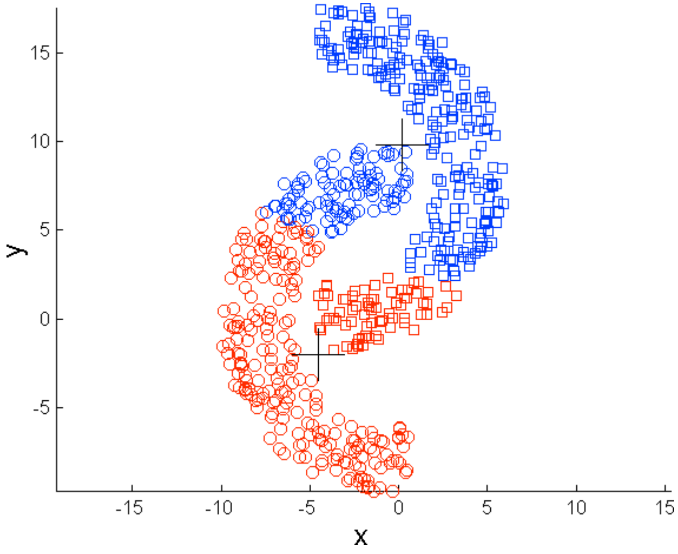


K-means (3 Clusters)

Non-globular Shapes



Original Points



K-means (2 Clusters)

Overcoming Limitations (Summary)

Problem	How to Overcome
Bad initialization	Use **K-Means++**
Outliers	Use **K-Medoids** or remove them
Unequal scale	**Normalize** data
Different shapes/densities	Use **DBSCAN / Mean-Shift**
Unknown K	Use **Elbow / Silhouette Method**
Overlapping clusters	Use **GMM**
High-dimensional data	Use **PCA**

Variations

- K-medoids: Similar problem definition as in K-means, but the centroid of the cluster is defined to be one of the points in the cluster (the medoid).

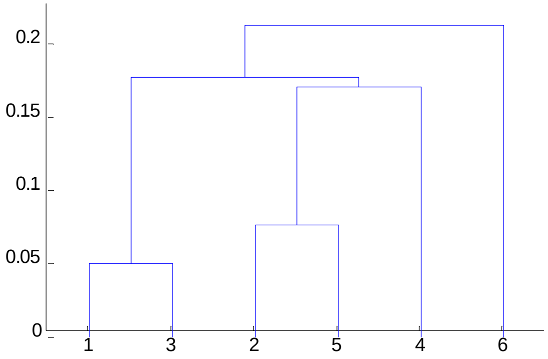
- K-centers: Similar problem definition as in K- means, but the goal now is to minimize the maximum diameter of the clusters (diameter of a cluster is maximum distance between any two points in the cluster).

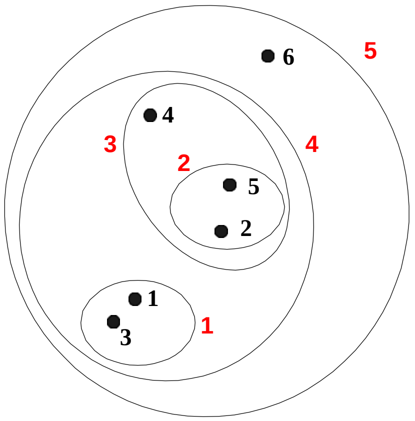
Hierarchical Clustering

- 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

- 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, …)

Agglomerative Clustering Algorithm

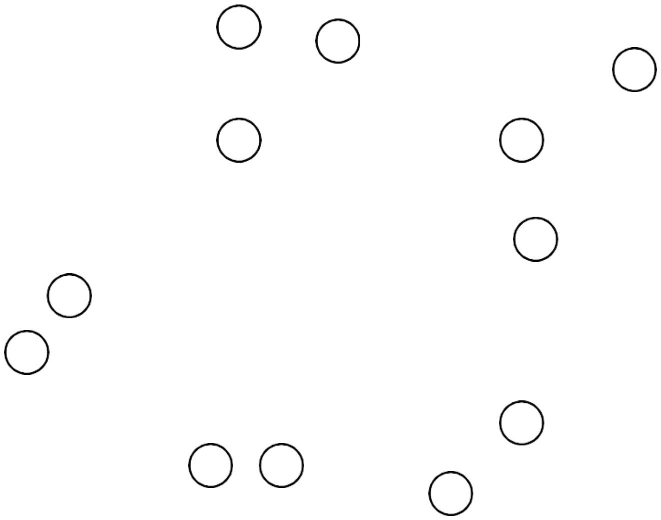
- More popular hierarchical clustering technique
- Basic algorithm is straightforward

```
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
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

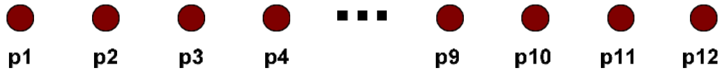
Starting Situation

- Start with clusters of individual points and a proximity matrix



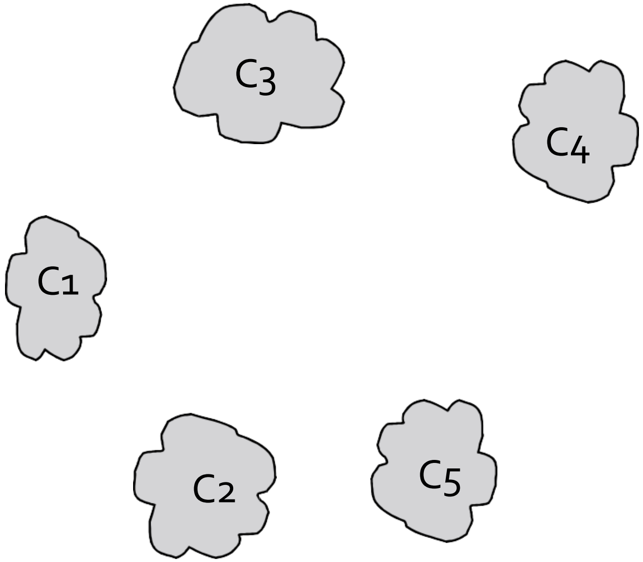
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix



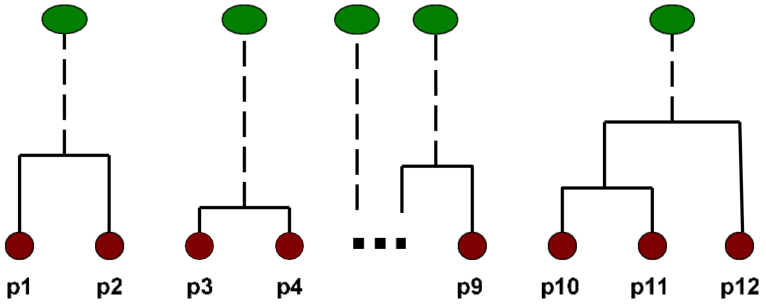
Intermediate Situation

- After some merging steps, we have some clusters

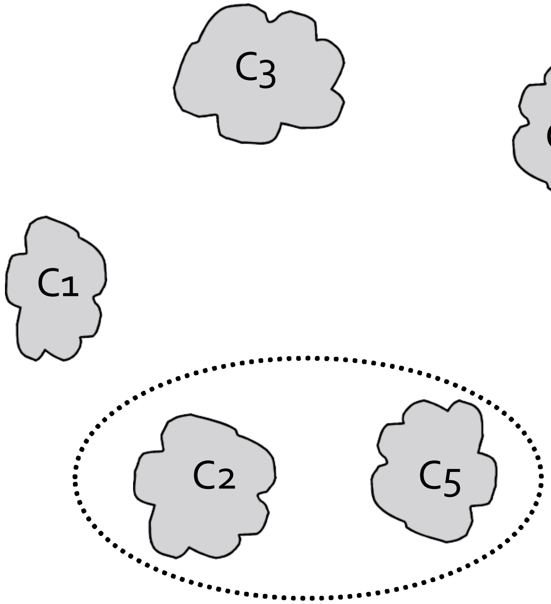


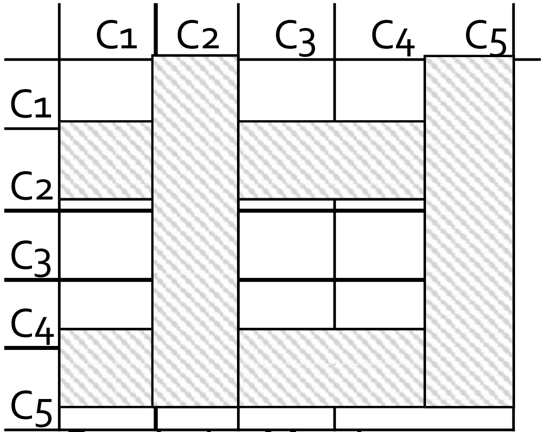
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix

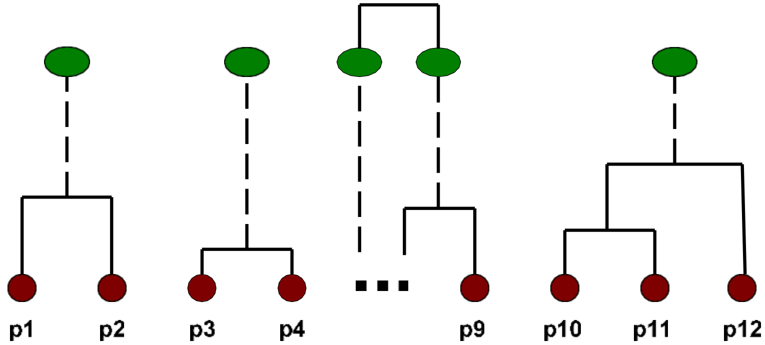


- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



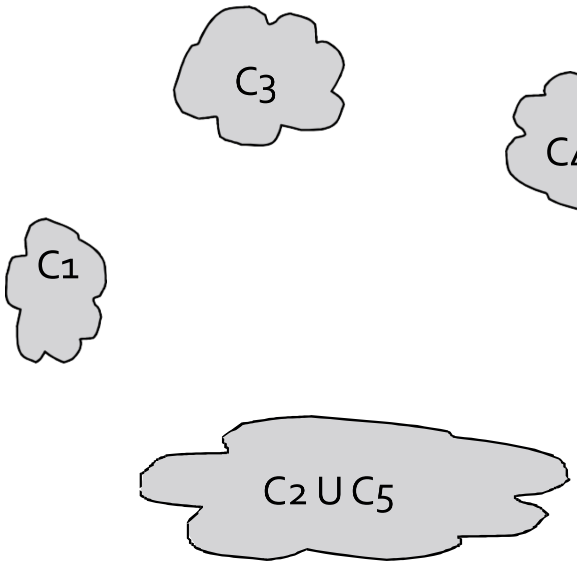


Proximity Matrix



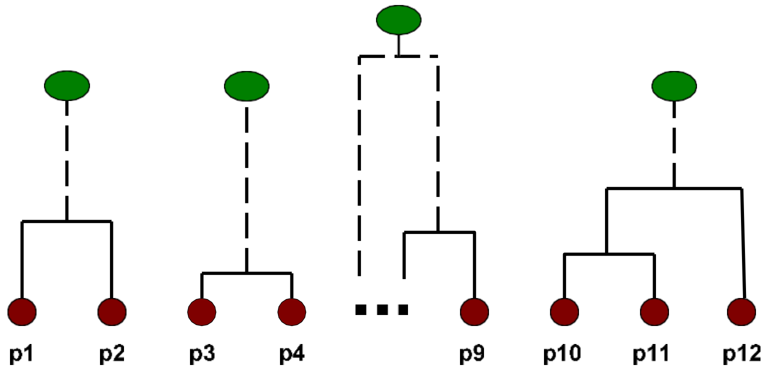
After Merging

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

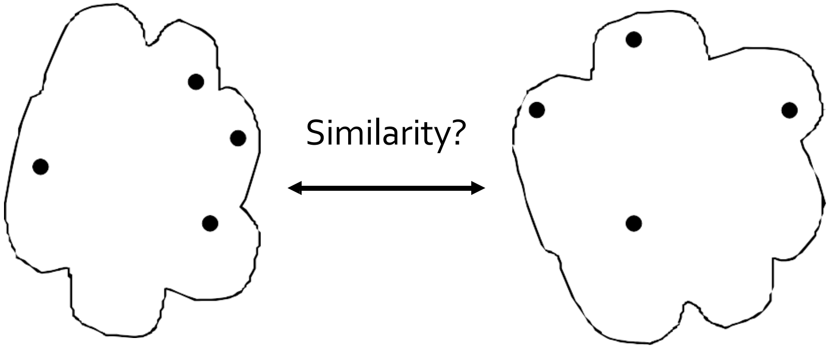


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

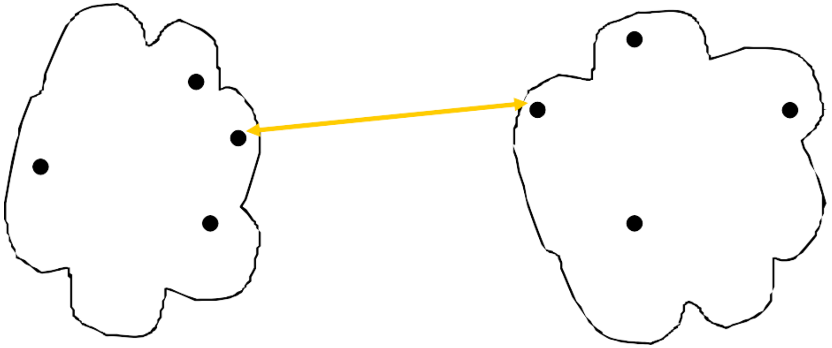


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

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

Proximity Matrix

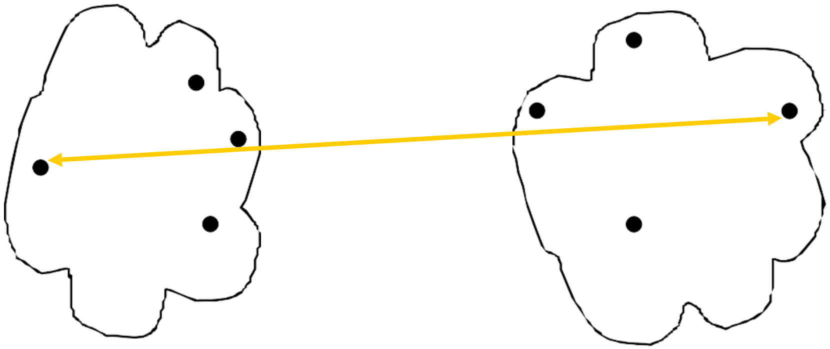
MIN



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

Proximity Matrix

MAX



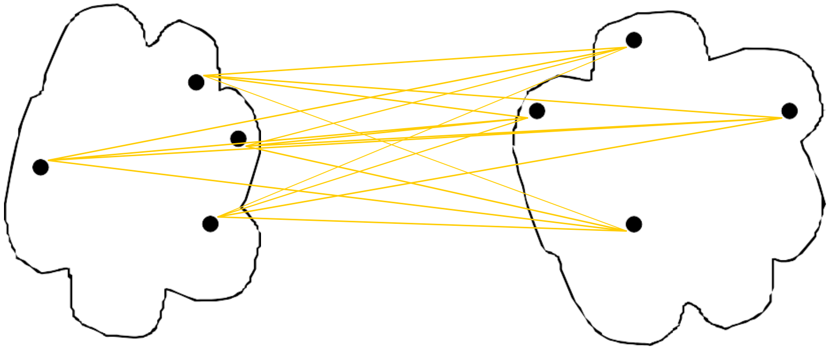
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

.

.

Proximity Matrix

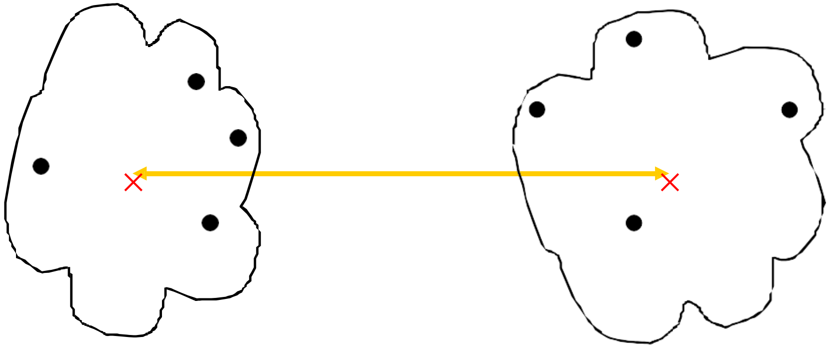
Group Average



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

Proximity Matrix

Distance Between Centroids



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

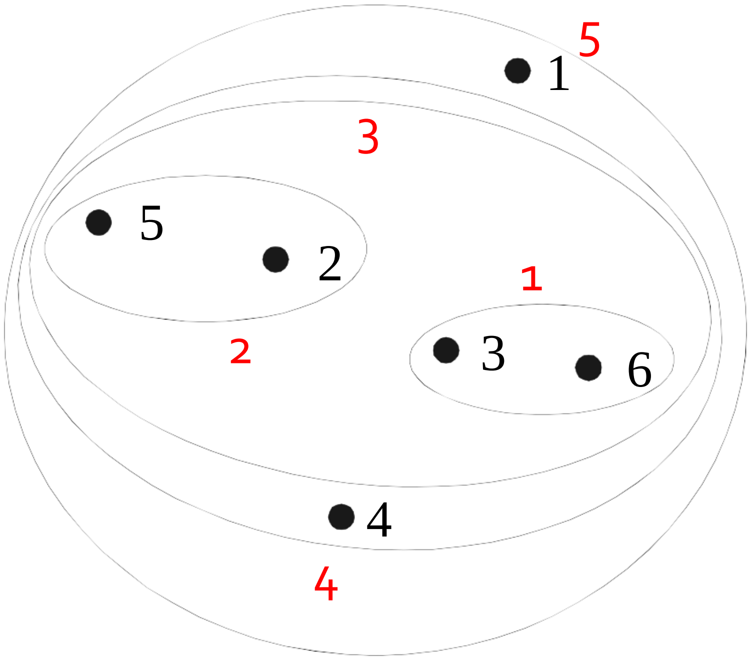
Proximity Matrix

Single Link- Complete Link

- Another way to view the processing of the hierarchical algorithm is that we create links between their elements in order of increasing distance
  - The MIN – Single Link, will merge two clusters when a single pair of elements is linked
  - The MAX – Complete Linkage will merge two clusters when all pairs of elements have been linked.

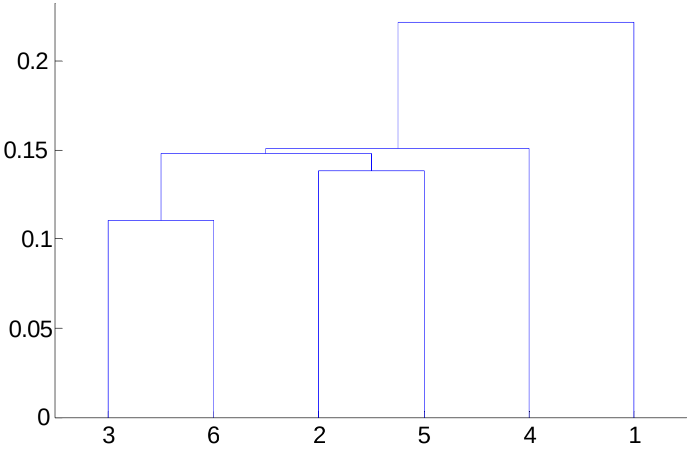
Hierarchical Clustering : MIN

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

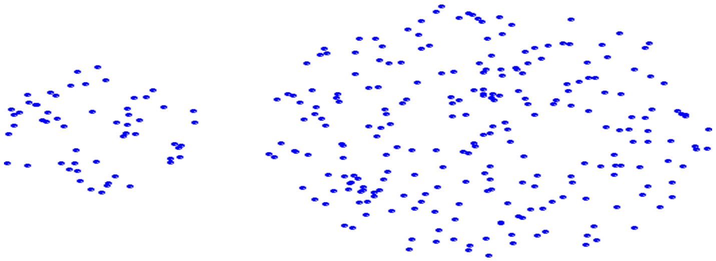


Nested Clusters

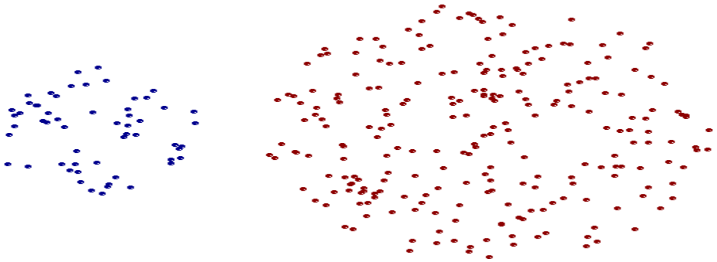
Dendrogram



Strength of MIN



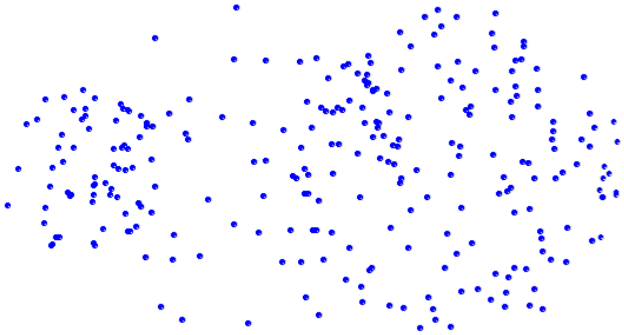
Original Points



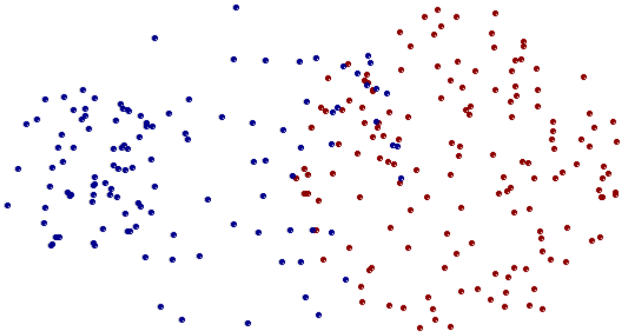
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



Original Points

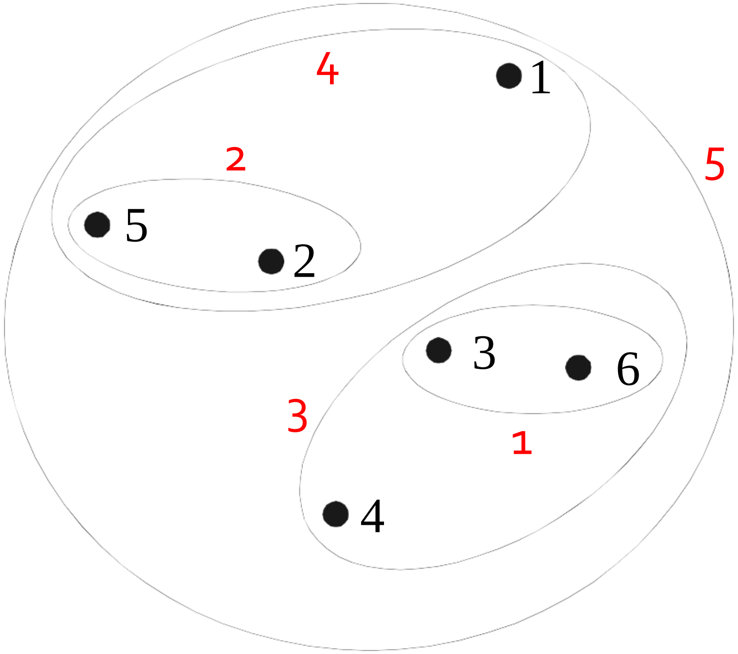


Two Clusters

- Sensitive to noise and outliers

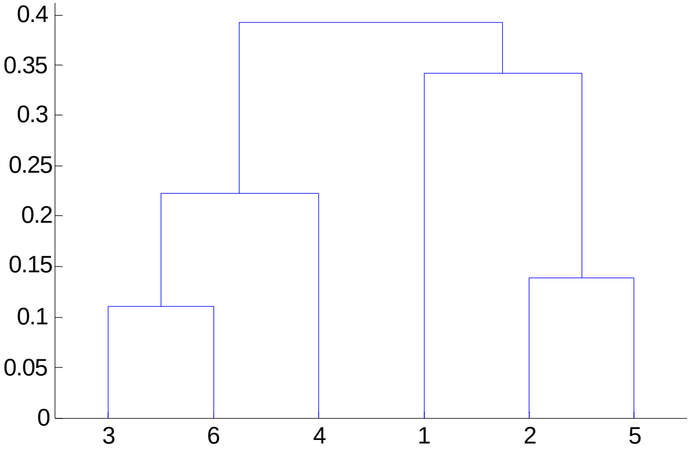
Hierarchical Clustering : MAX

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

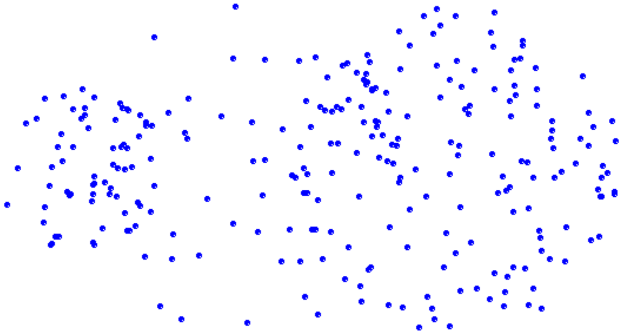


Nested Clusters

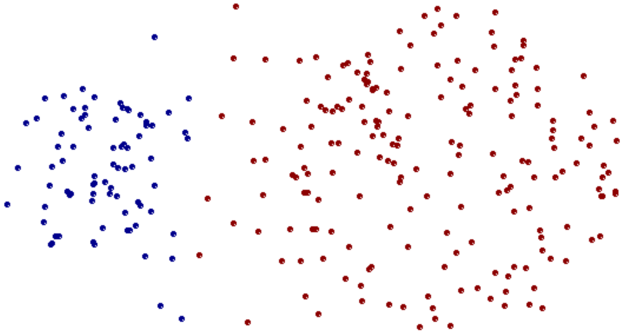
Dendrogram



Strength of MAX



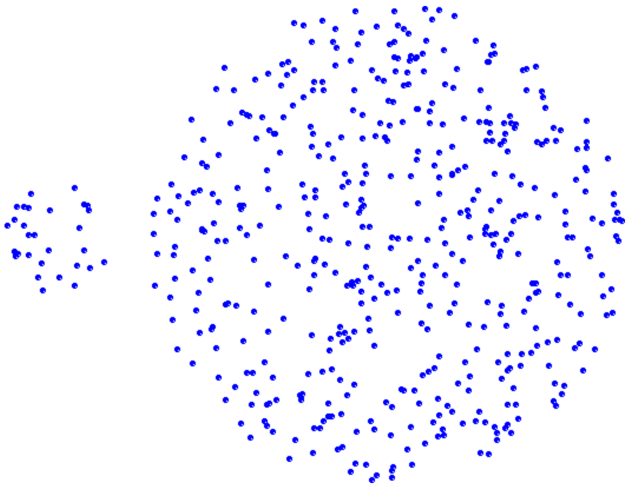
Original Points



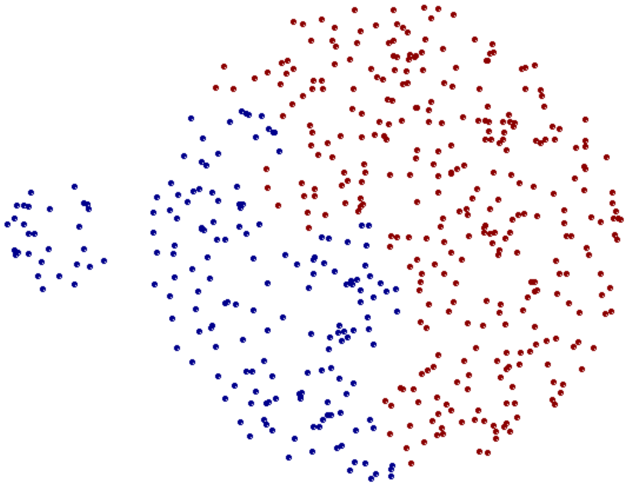
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

Cluster Similarity : Group Average

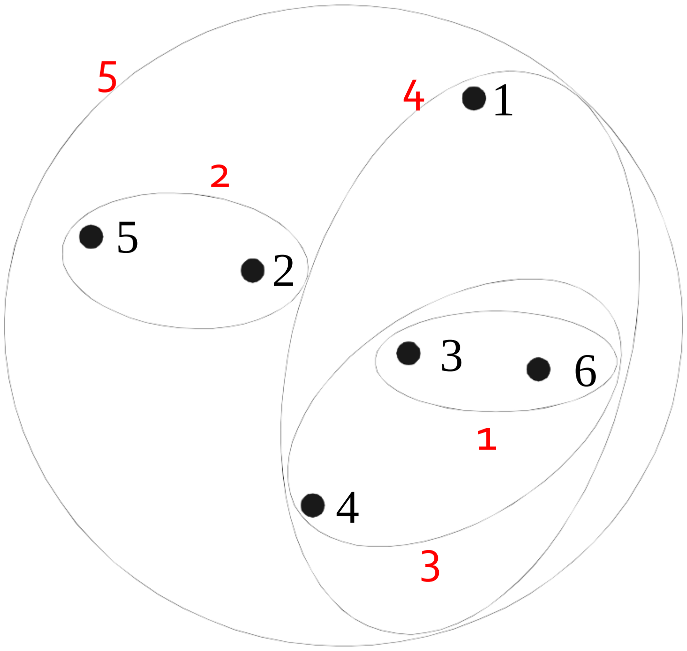
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

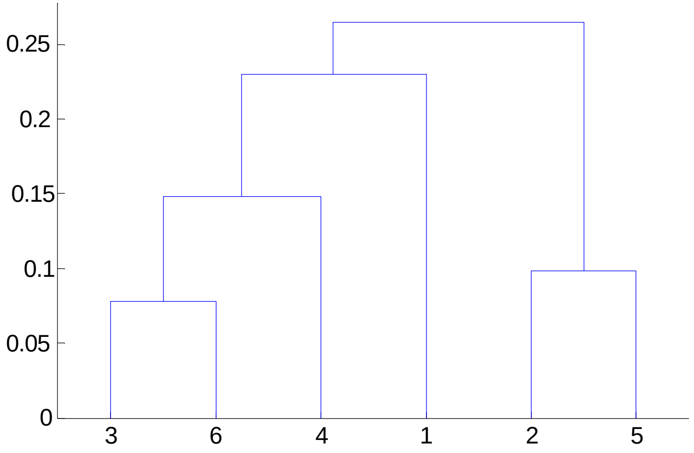
- Need to use average connectivity for scalability since total proximity favors large clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

Hierarchical Clustering : Group Average



	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0



Nested Clusters

Dendrogram

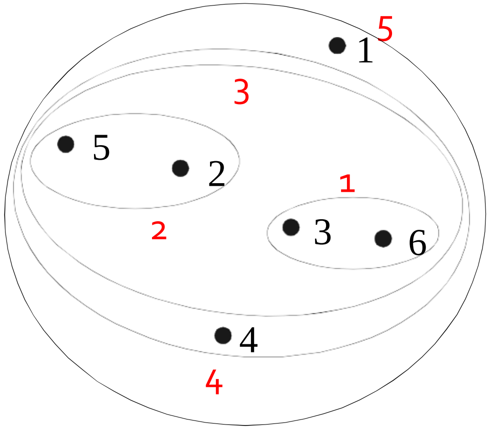
Group Average

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Cluster Similarity : Ward's Method

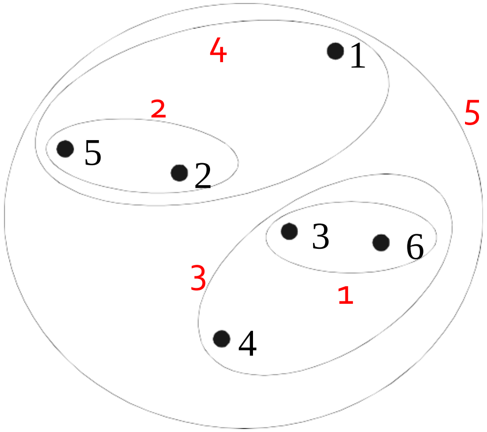
- Similarity of two clusters is based on the increase in squared error (SSE) when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

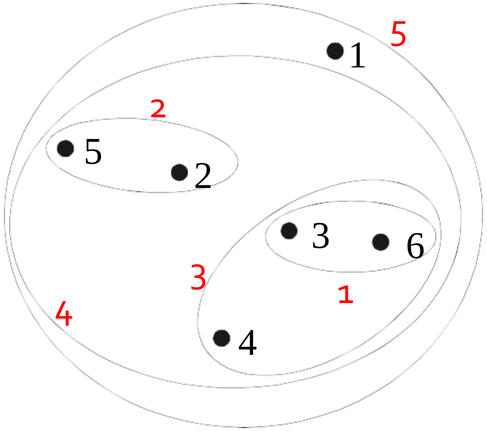
Hierarchical Clustering : Comparison



MIN

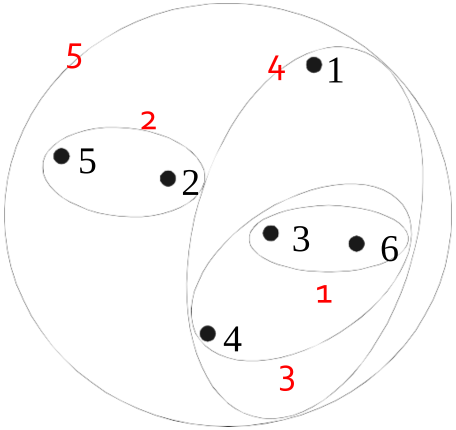
MAX





Group Average

Ward's Method



Time and Space Requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - N is the number of points.

- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Problems and Limitations

- Computational complexity in time and space
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

Thank You