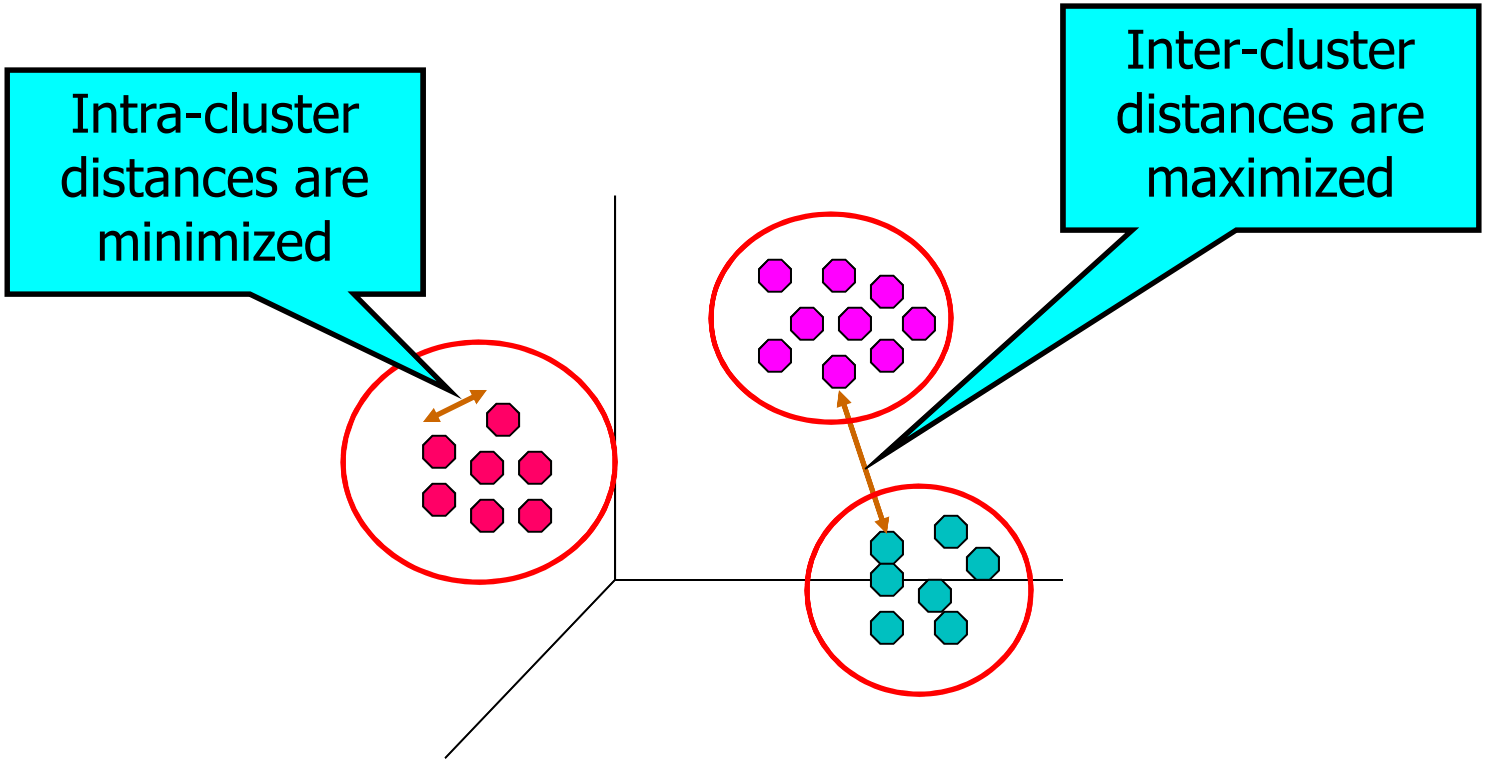
# Clustering:
# k-Means, Agglomerative, DBSCAN


Tan, Steinbach, Kumar

(With Modification by Yufei Tao)

# What is Cluster Analysis?

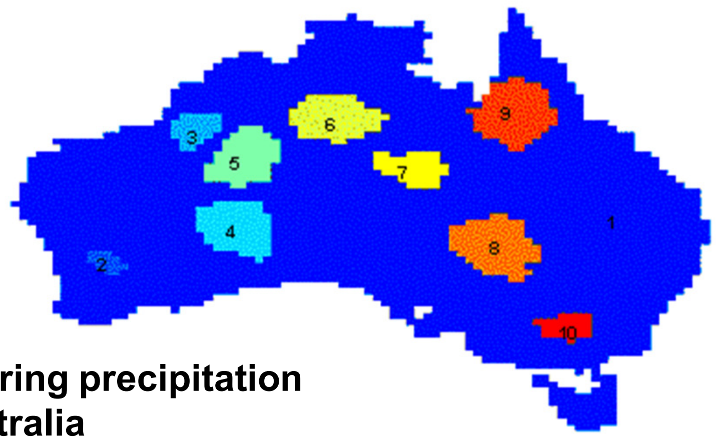
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



# Applications of Cluster Analysis

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

- **Data Utilization**
  - **Summarization**
  - **Compression**



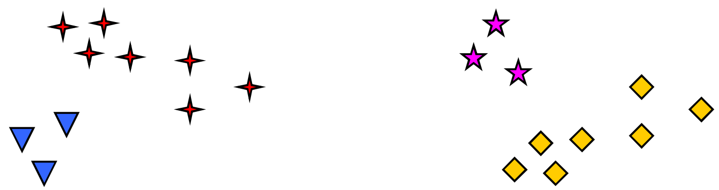

**Clustering precipitation in Australia**

# Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

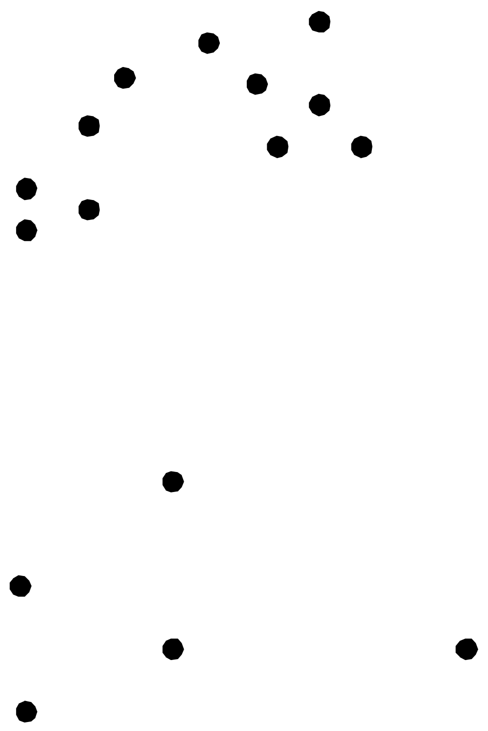


Four Clusters

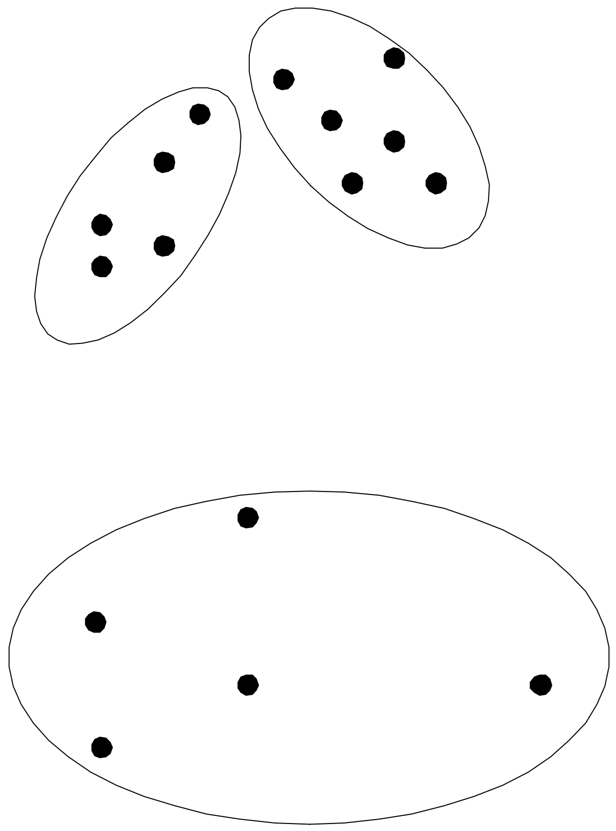
# Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping 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

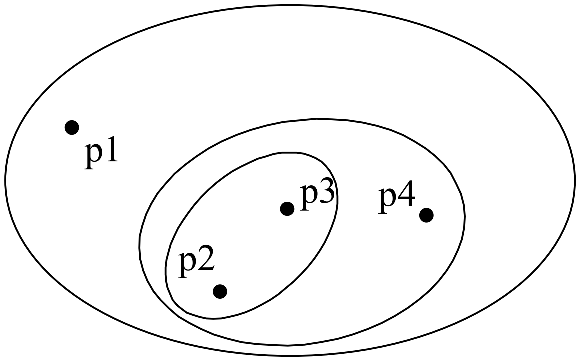


Original Points

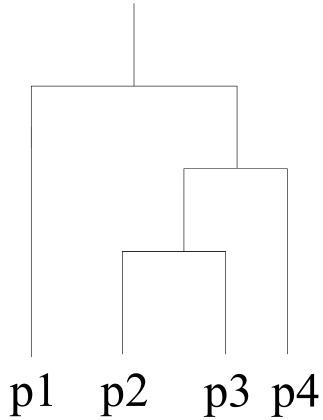


A Partitional  Clustering

# Hierarchical Clustering



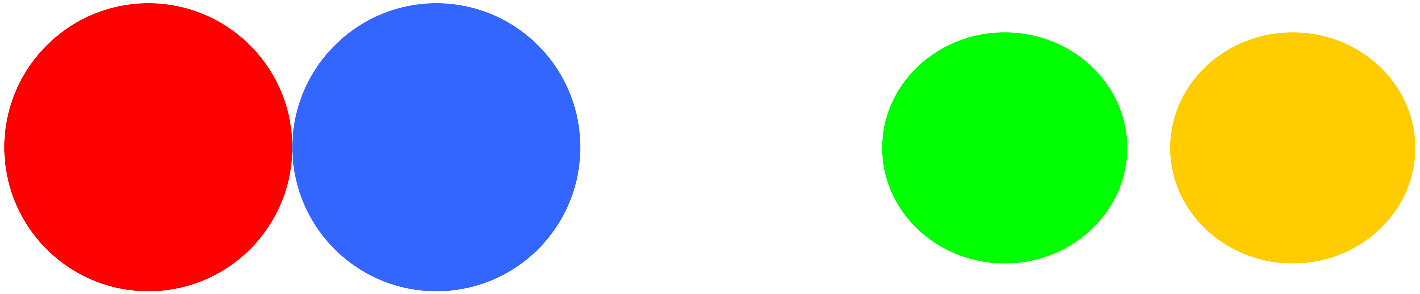

Hierarchical Clustering




Dendrogram

# Types of Clusters: Center-Based

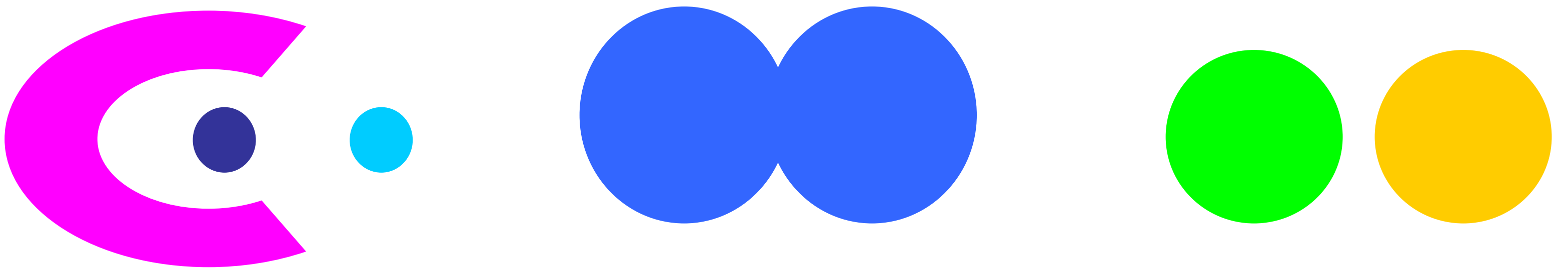
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative"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



**4 center-based clusters**

# Types of Clusters: 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**

# Clustering Algorithms

- K-center (the previous lecture).
  – Think: how?
- K-means
- Hierarchical clustering
- Density-based clustering

# K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid.
  - The centroid of a point set S is the point p whose x- (y-) coordinate is the mean of the x- (y-) coordinates of the points in S.
- Number of clusters, K, is an input parameter.

```
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
```

# K-means Clustering – Details

- Initial centroids are important, as discussed later.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- In practice, the stopping condition may be changed to ‘Until relatively few points change clusters’

# Evaluating K-means Clusters

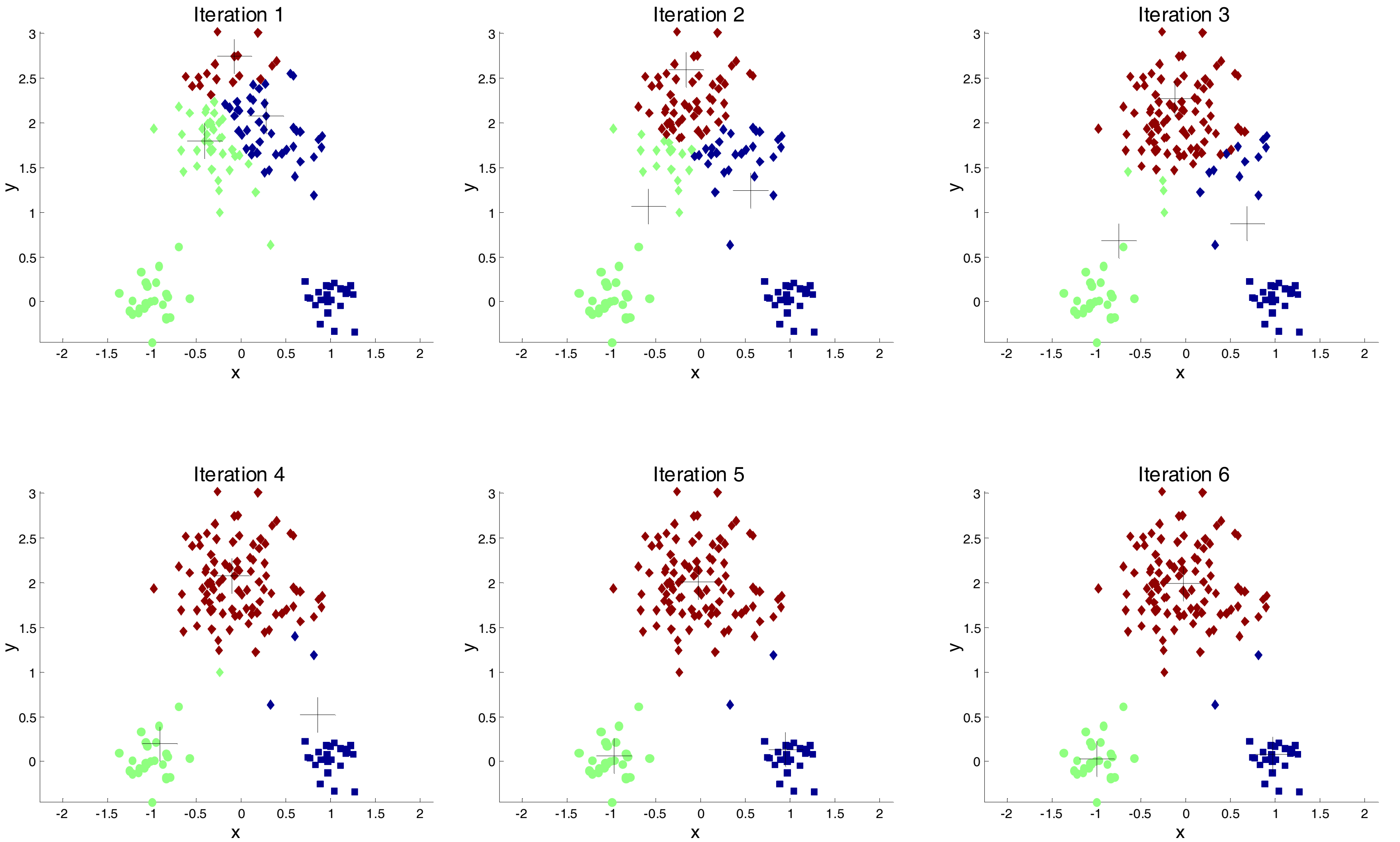
- Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

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

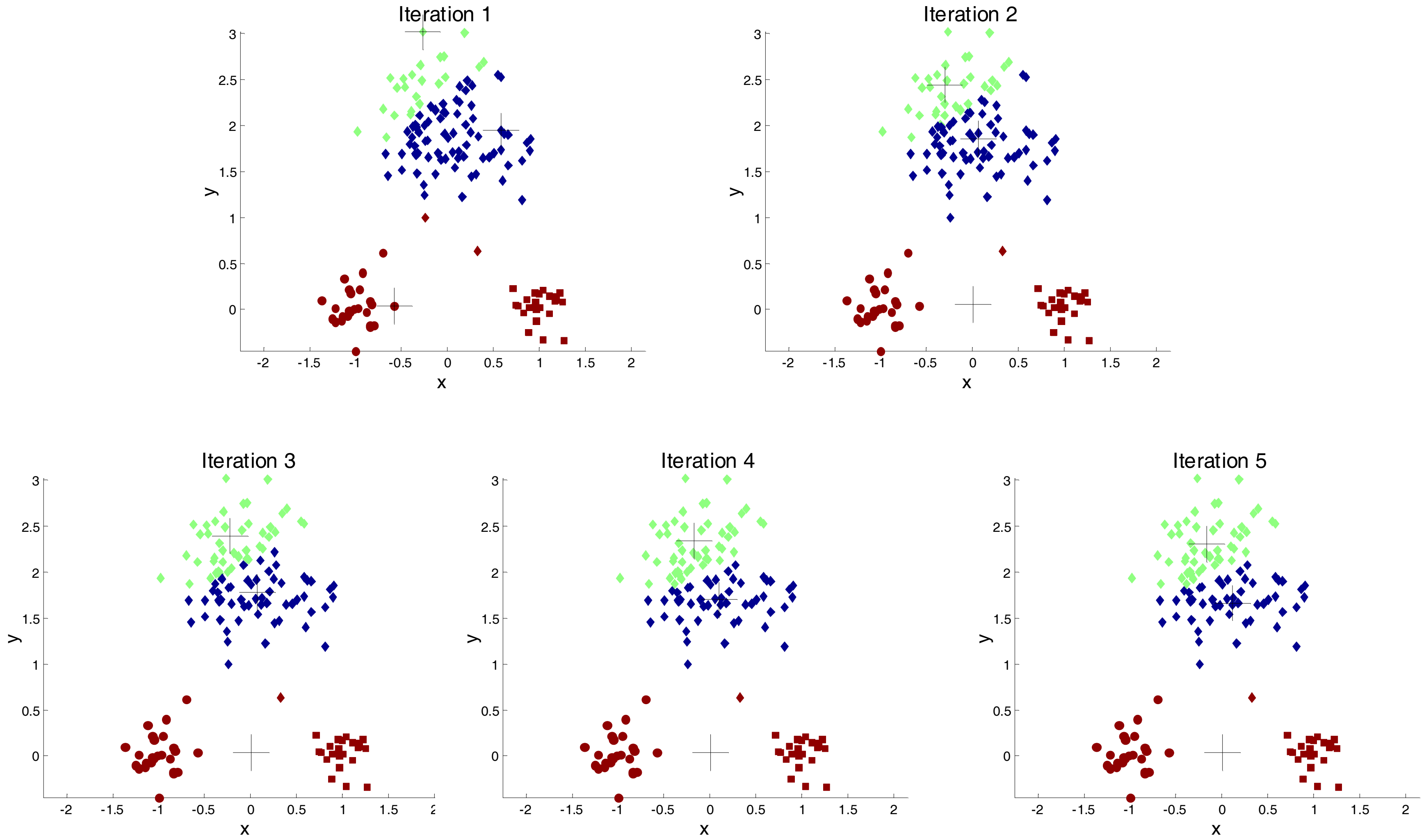
  - $x$ is a data point in cluster $C_i$ and $m_i$ is the centroid of $C_i$

- Ideally, we want to find the K clusters to minimize SSE.

# Example with k = 3



# Importance of Choosing Initial Centroids



# Choosing the Initial Centroids

- A strategy that works for any distance definition:
  - Randomly pick k points.

- A better strategy when the distance definition satisfies triangle inequality:
  - Solution of the k-center problem.

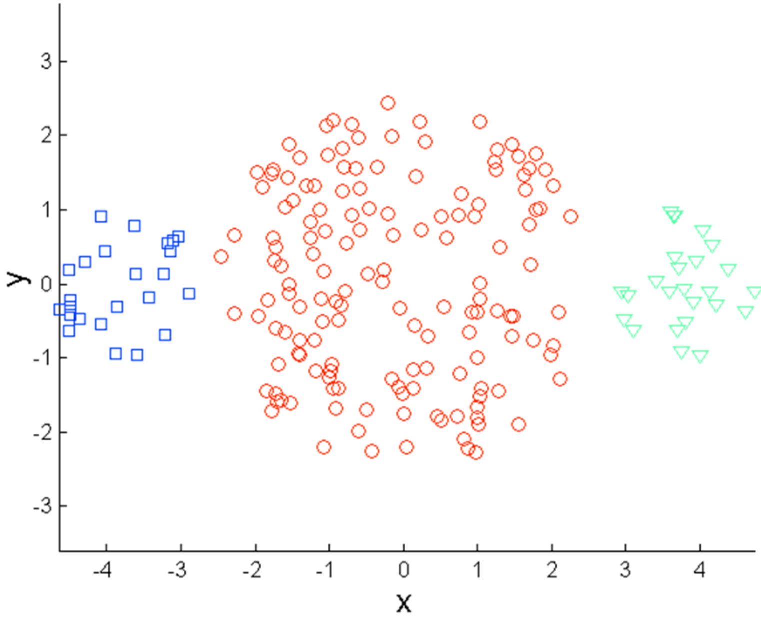
- An even better strategy for Euclidean distance:
  - See next.

## Initial Centriod Selection for Euclidean Distance

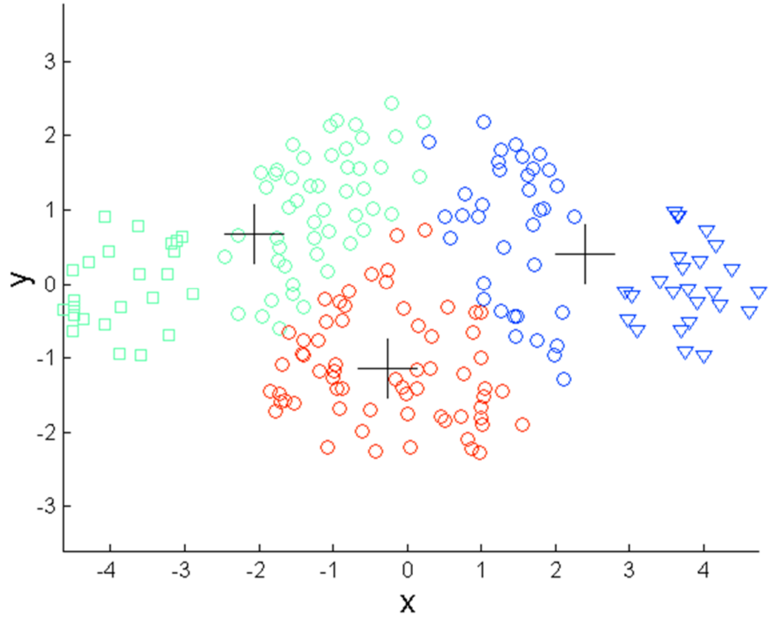
- P = the input point set
- S = an empty centroid set
- add a point to S uniformly at random
- for i = 2 to k
  - for each point p in P, calculate D(p) as the minimum distance from p to the points already in S
  - sample a point in P by ensuring that each point p in P is sampled with a probability proportional to $(D(p))^2$
  - add the sampled point to S

- The above algorithm allows k-means to achieve an approximation ratio of O(lg k). Namely, if the optimal k clusters has SSE s, then k-means guarantees returning clusters with SSE at most O(s lg k).

# Limitations of K-means: Differing Sizes

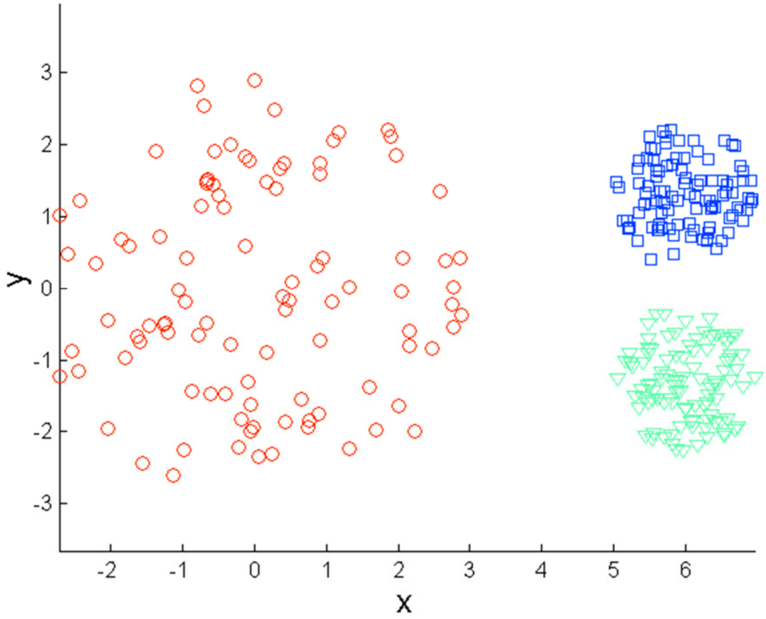



**Original Points**

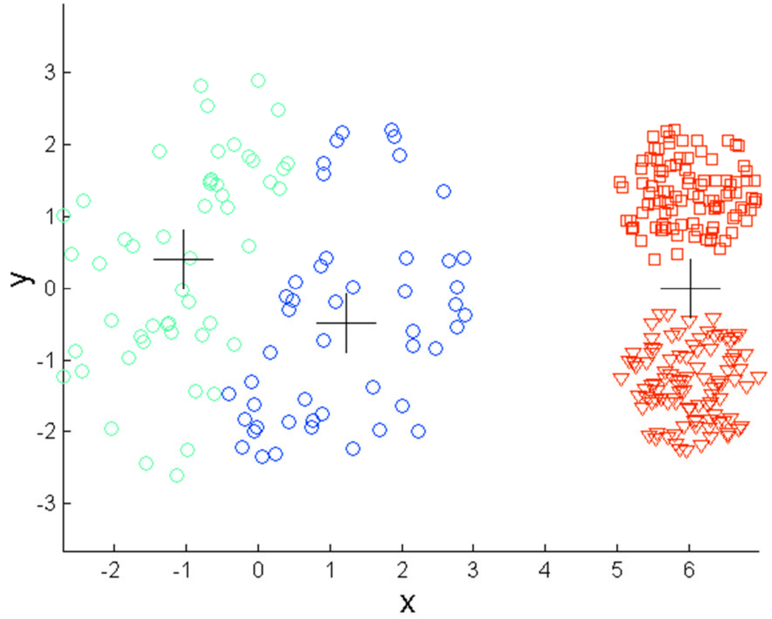



**K-means (3 Clusters)**

# Limitations of K-means: Differing Density

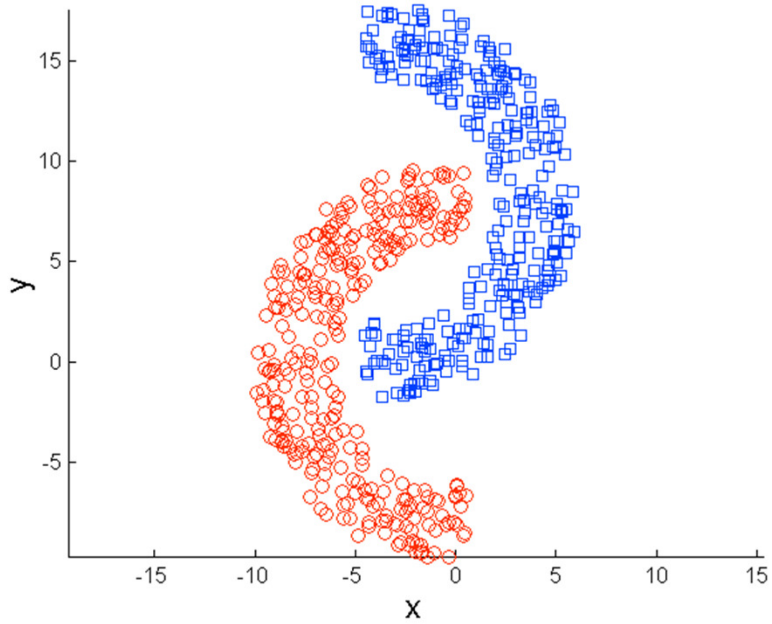



**Original Points**

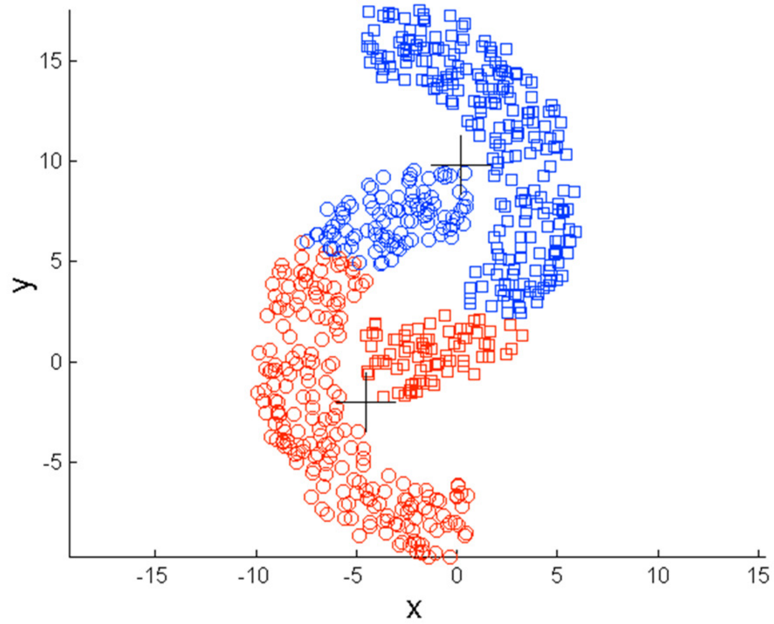



**K-means (3 Clusters)**

# Limitations of K-means: Non-globular Shapes



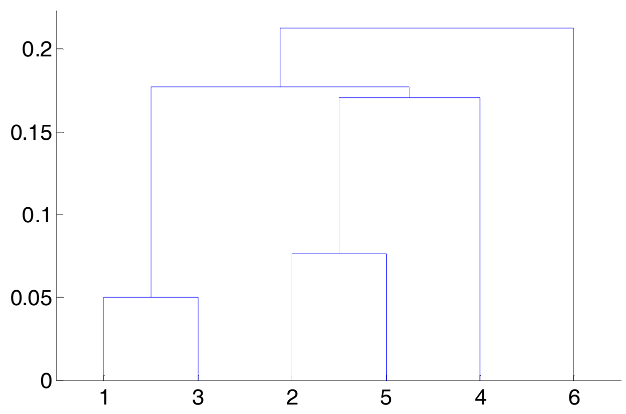

**Original Points**

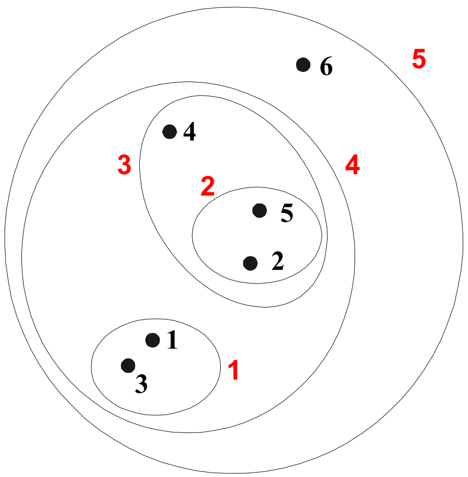



**K-means (2 Clusters)**

# 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, …)

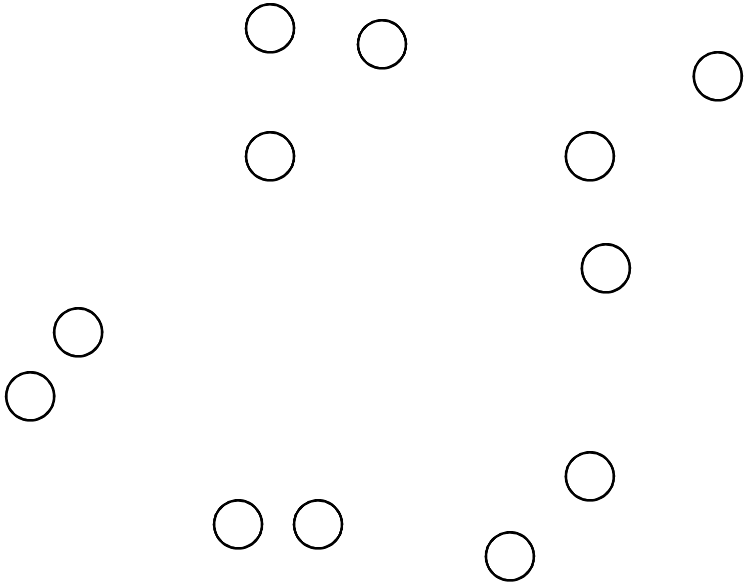
# Agglomerative Clustering Algorithm

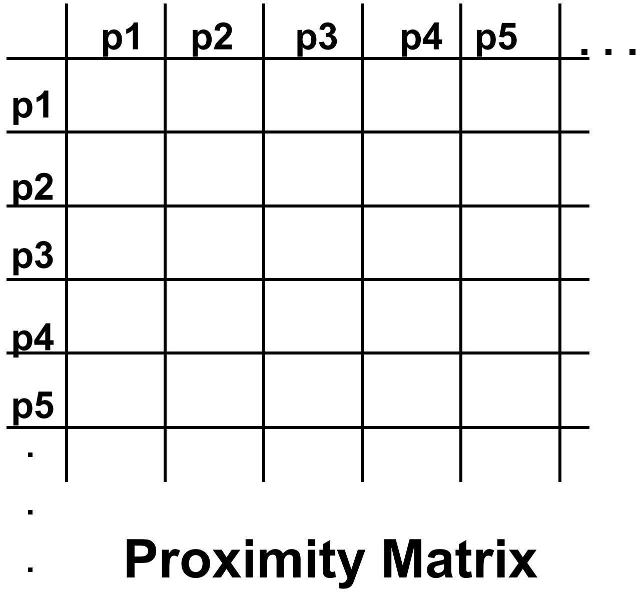
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

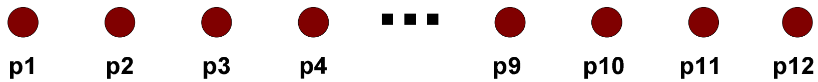
# Starting Situation

- Start with clusters of individual points and a proximity matrix



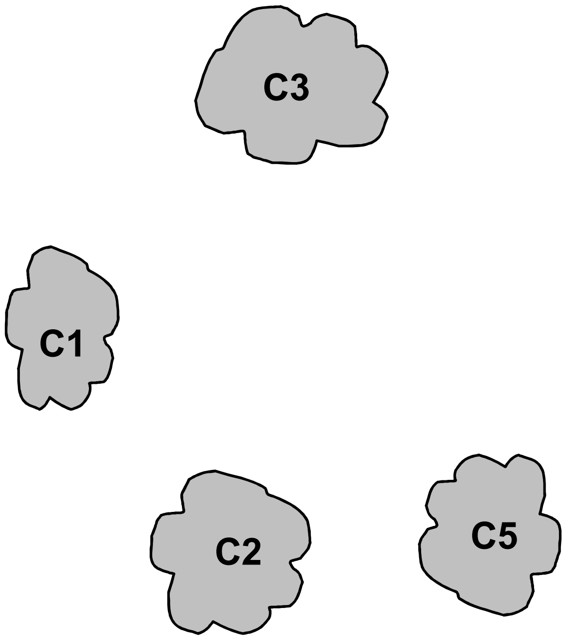



Proximity Matrix



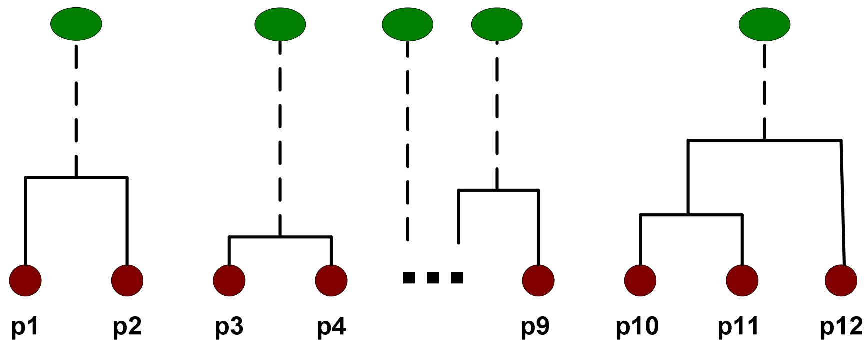
# Intermediate Situation

- After some merging steps, we have some clusters



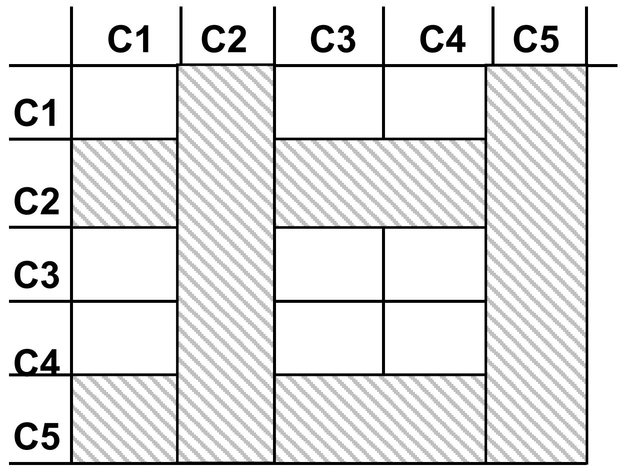

| | C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|---|
| C1 | | | | | |
| C2 | | | | | |
| C3 | | | | | |
| C4 | | | | | |
| C5 | | | | | |

**Proximity Matrix**

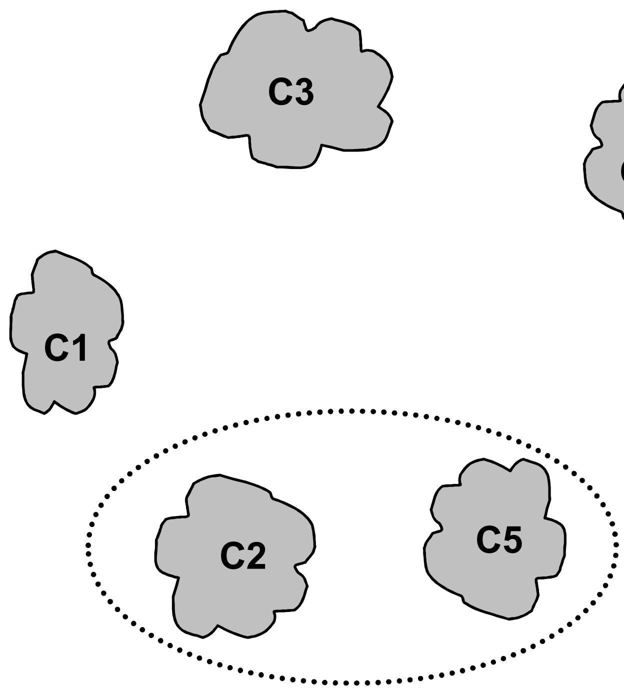


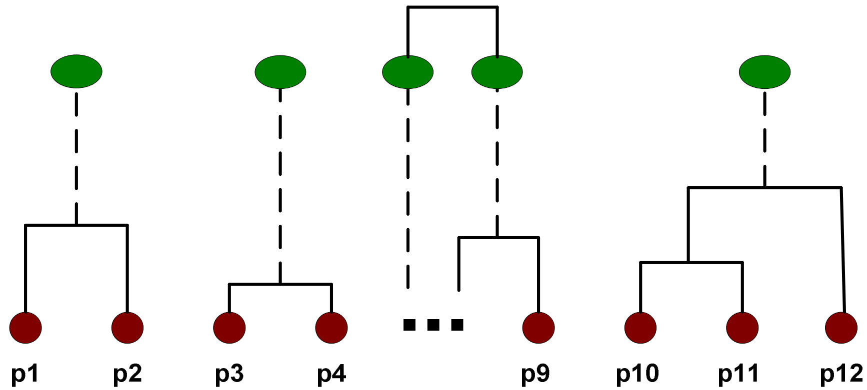
# Intermediate Situation

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



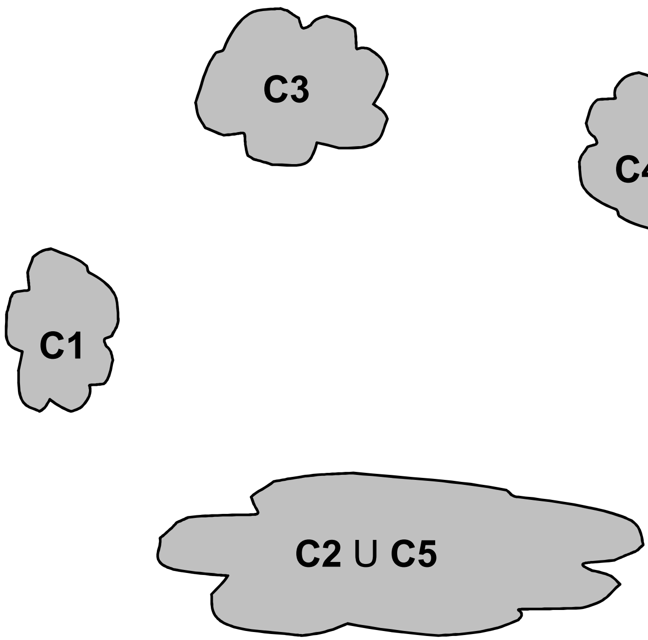

**Proximity Matrix**



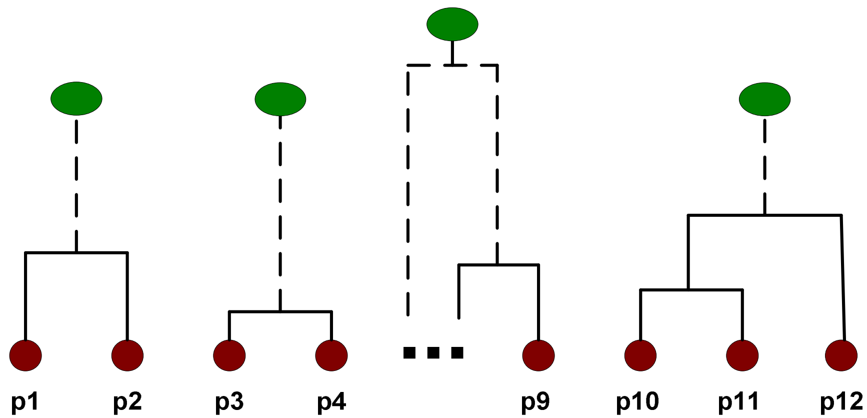
# After Merging

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

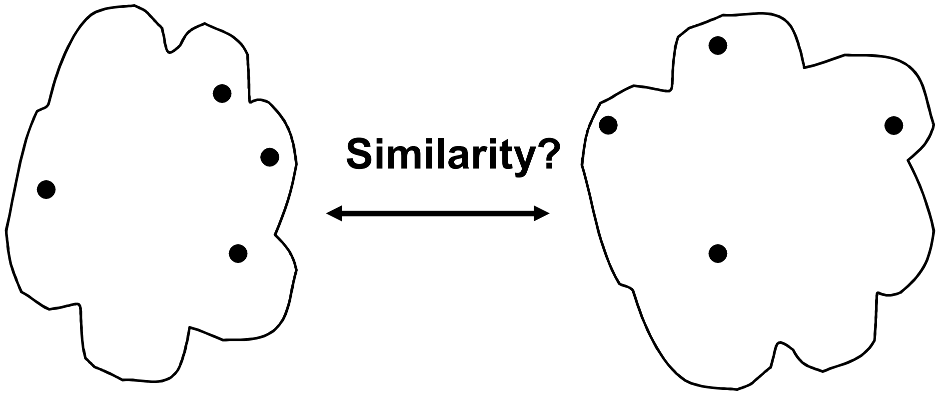



| | C1 | C2 U C5 | C3 | C4 |
|---|---|---|---|---|
| C1 | | ? | | |
| C2 U C5 | ? | ? | ? | ? |
| C3 | | ? | | |
| C4 | | ? | | |

**Proximity Matrix**



# How to Define Inter-Cluster Similarity

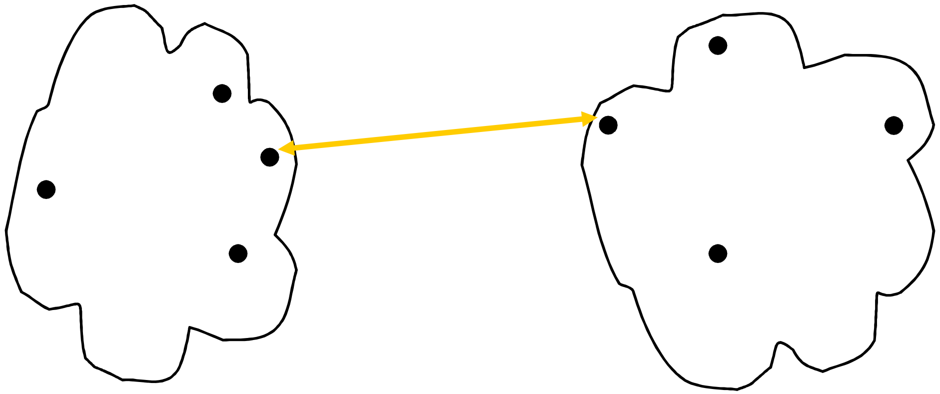



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average

# How to Define Inter-Cluster Similarity

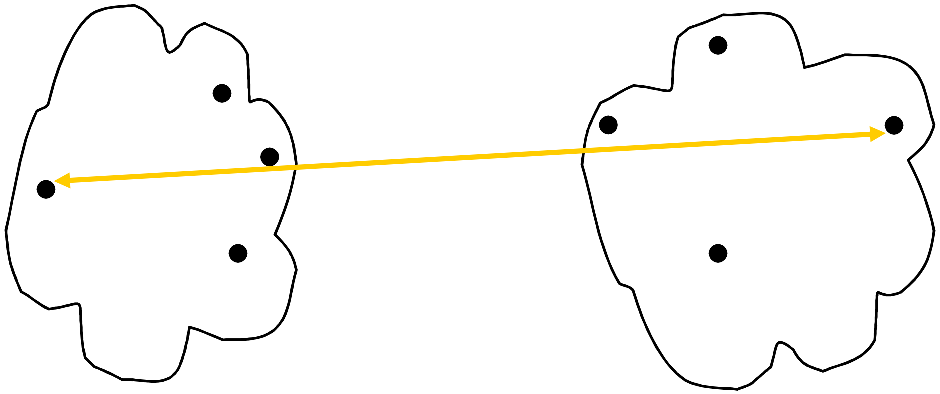


| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average

# How to Define Inter-Cluster Similarity

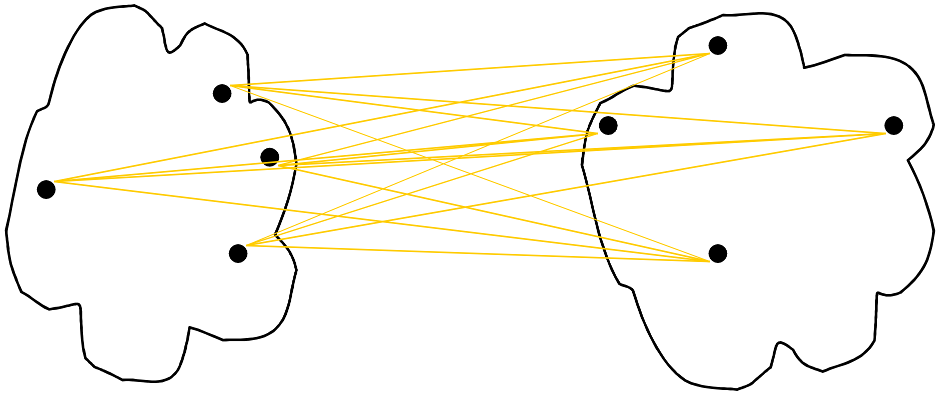


- MIN
- MAX
- Group Average

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . . . | | | | | | |

**Proximity Matrix**

# How to Define Inter-Cluster Similarity

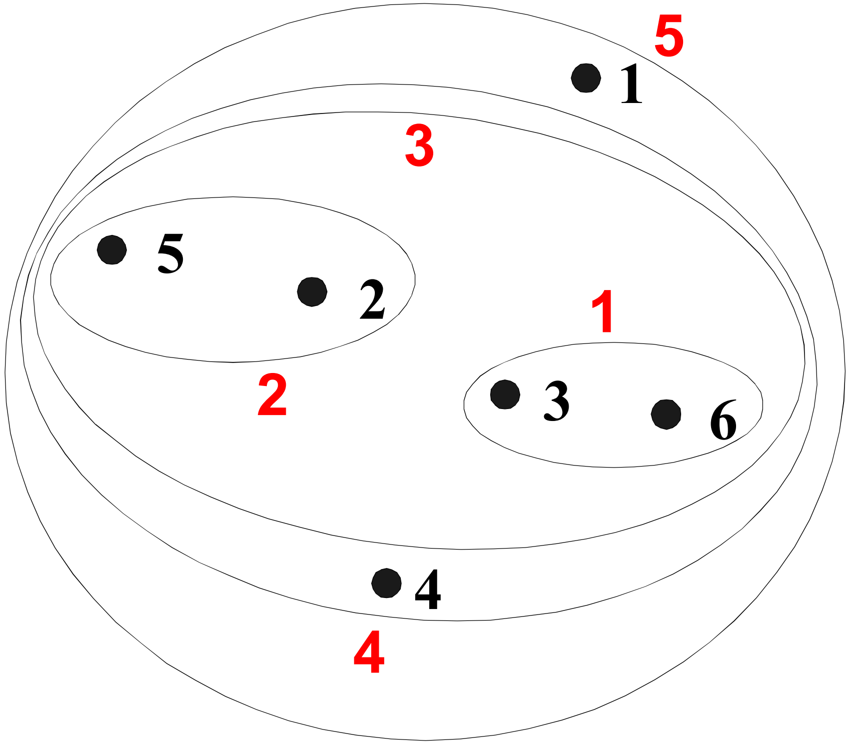


- MIN
- MAX
- Group Average

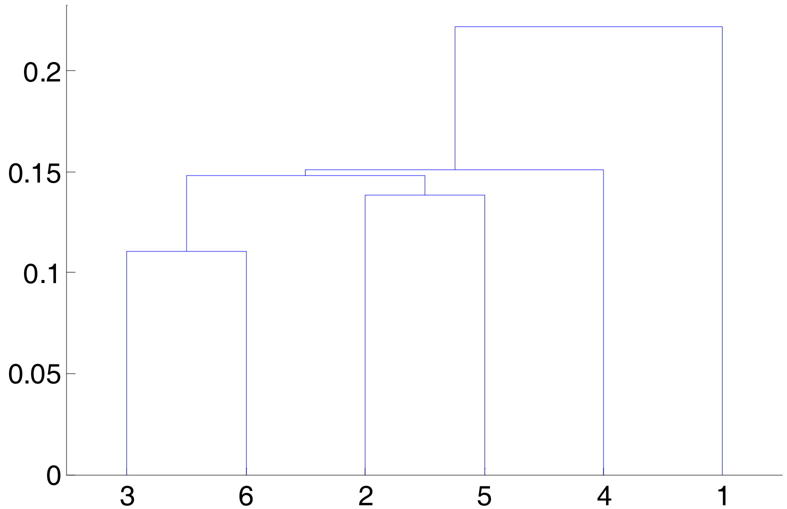
| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

# Hierarchical Clustering: MIN

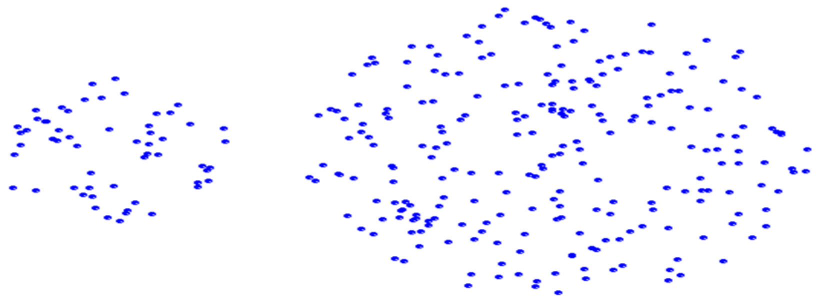



Nested Clusters

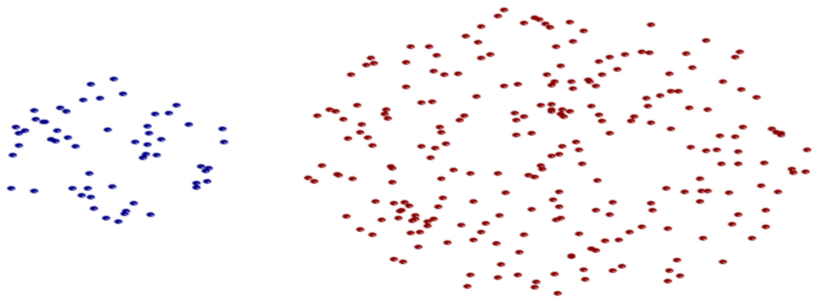



Dendrogram

# Strength of MIN



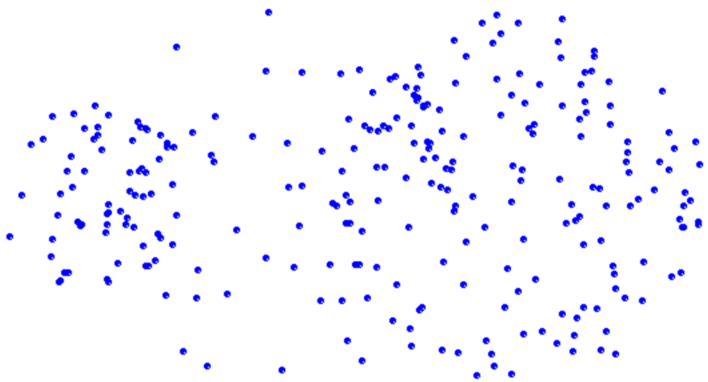
Original Points



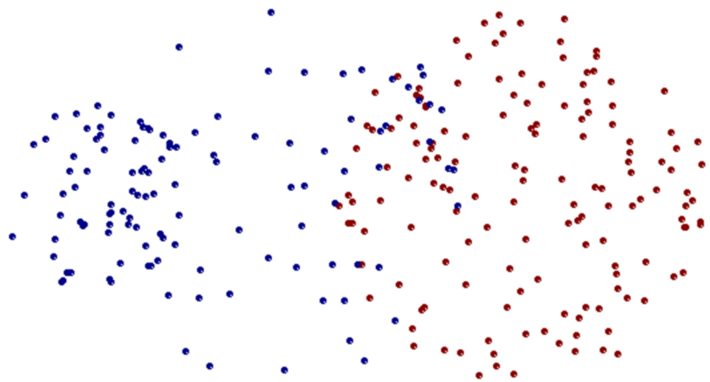
Two Clusters

- Can handle non-elliptical shapes

# Limitations of MIN



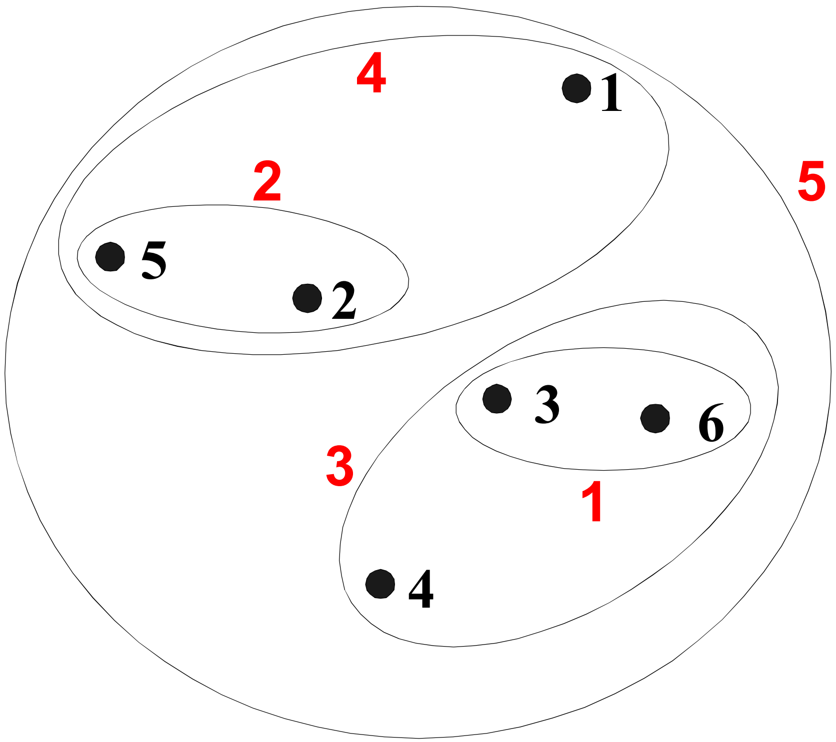
**Original Points**



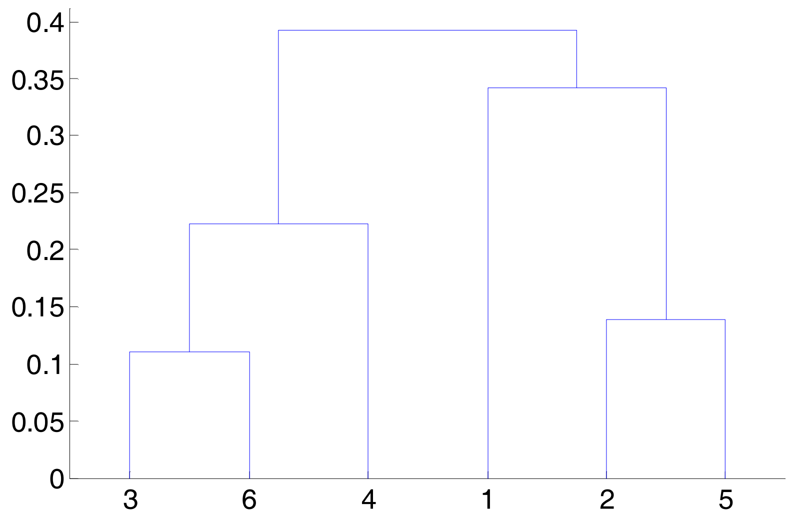
**Two Clusters**

- **Sensitive to noise and outliers**

# Hierarchical Clustering: MAX

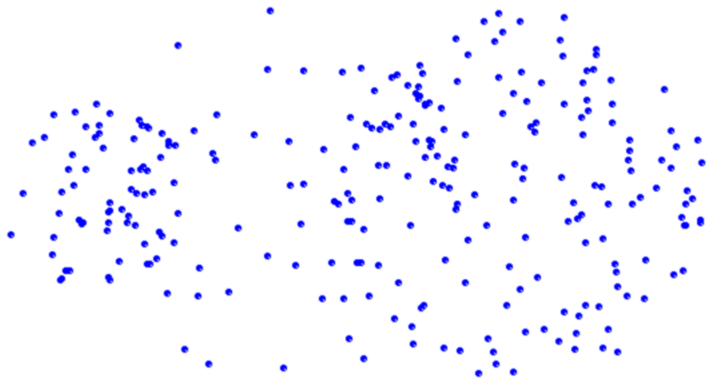



Nested Clusters

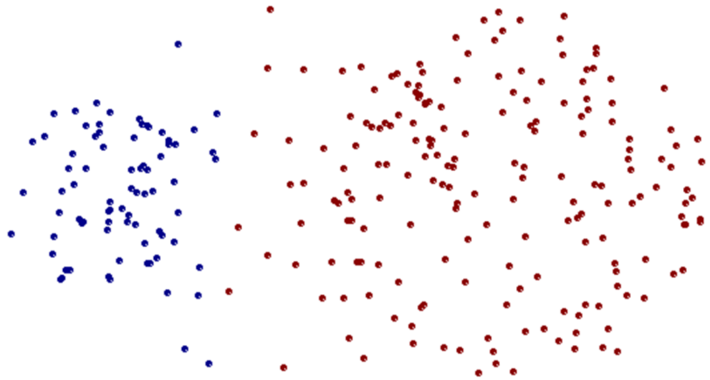



Dendrogram

# Strength of MAX



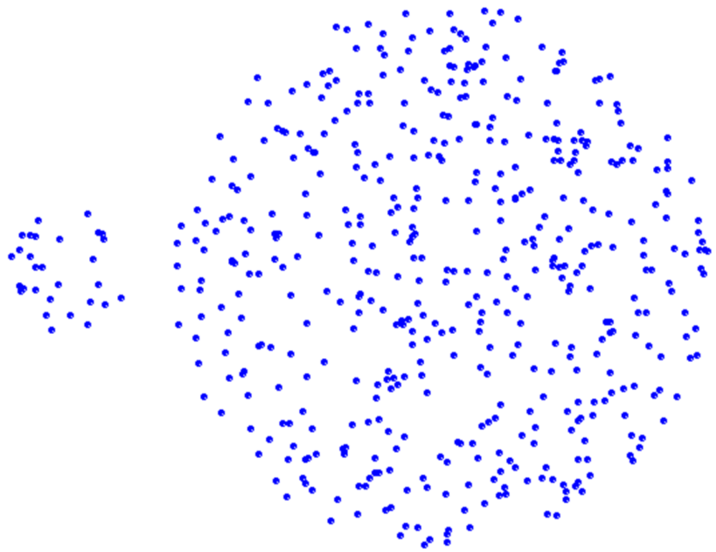
Original Points



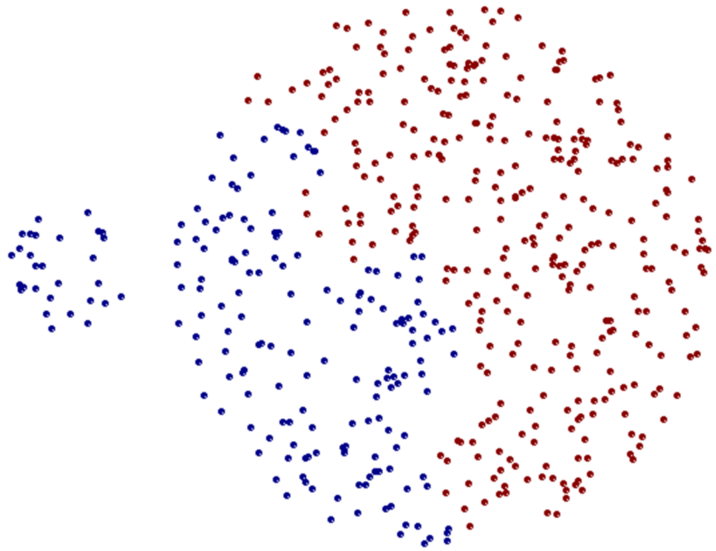
Two Clusters

• Less susceptible to noise and outliers

# Limitations of MAX



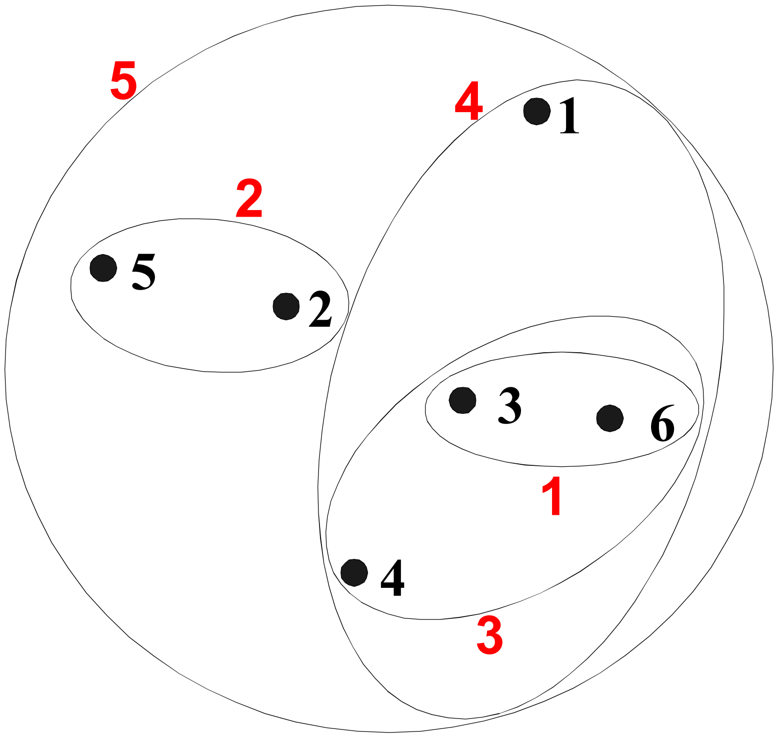
Original Points

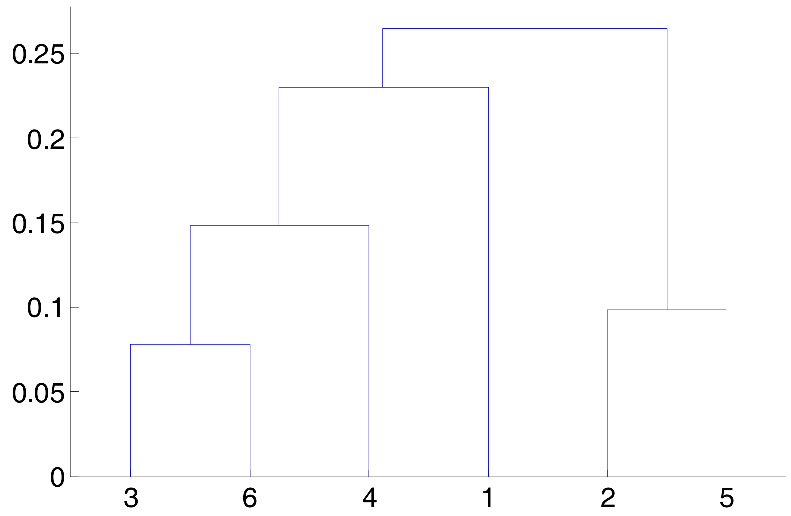


Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

# Hierarchical Clustering: Group Average




**Nested Clusters** **Dendrogram**

# Hierarchical Clustering: Group Average

- Compromise between MIN and MAX

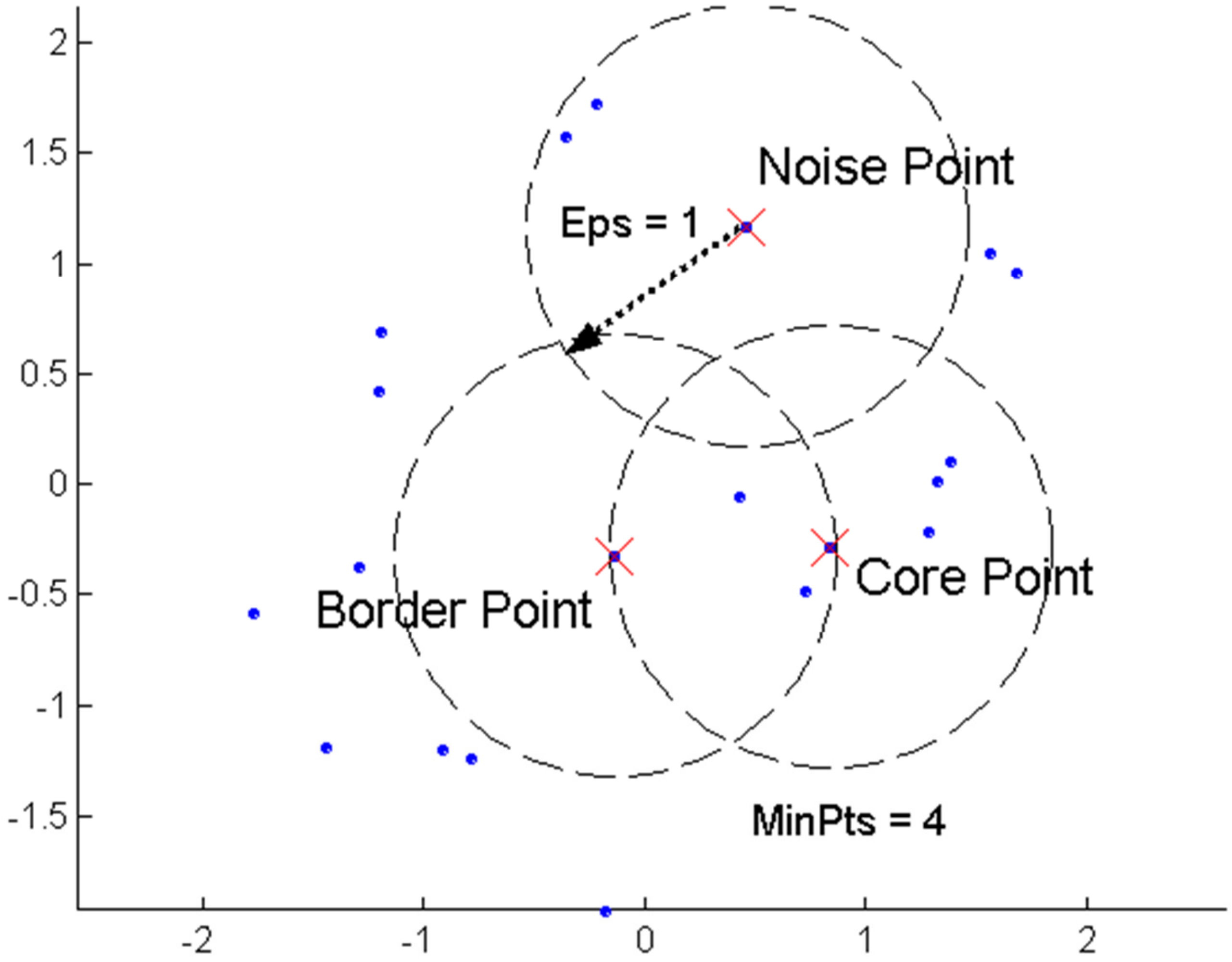
- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards ball-like clusters

# DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, 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: Core, Border, and Noise Points

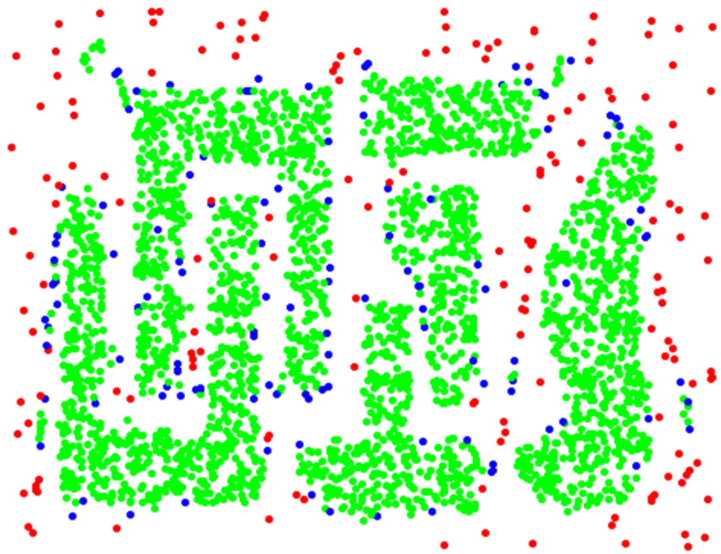


# DBSCAN Algorithm

- Eliminate noise points
- Put an edge between each pair of core points within distance Eps of each other
- Make each group of connected core points into a separate cluster
- Assign each border point arbitrarily to one of the clusters containing its associated core points

# DBSCAN: Core, Border and Noise Points

**Original Points**

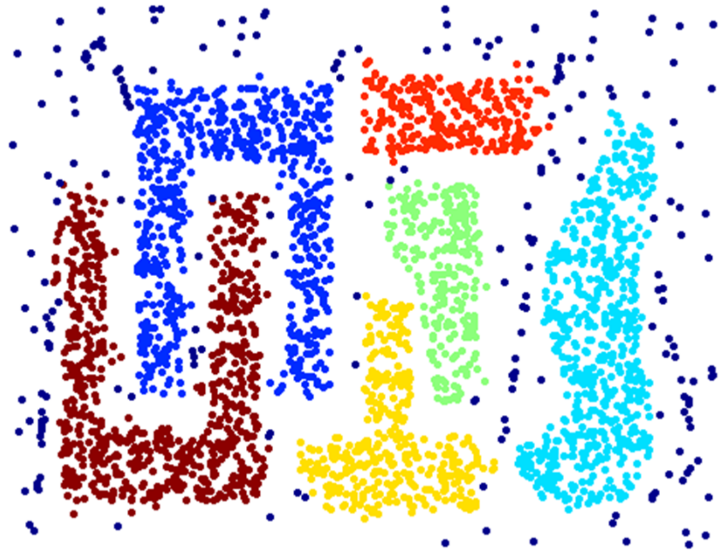


**Point types: core, border and noise**

**Eps = 10, MinPts = 4**

# When DBSCAN Works Well

Original Points



Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

# Drawback of DBSCAN

- Need to specify Eps and MinPts, which can be difficult in practice.