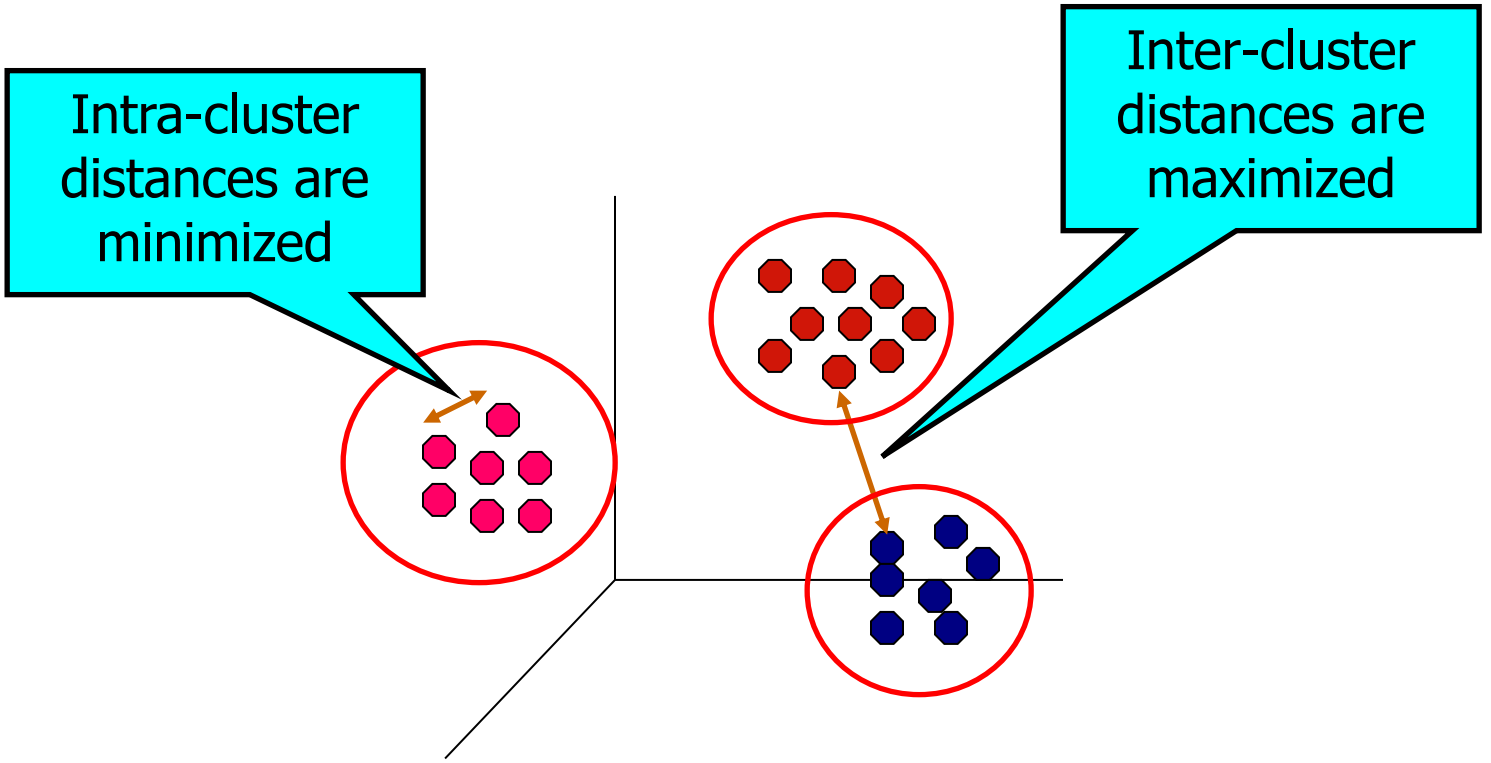
Data Science – Semester 6 – Spring 2022/2023

INTRODUCTION TO DATA MINING & Machine Learning

Lecture 7:
Clustering Techniques

What is Cluster Analysis?

- Finding **groups of objects** such that:
  - Members of a group are close/similar to each other
  - Members of different clusters are dissimilar



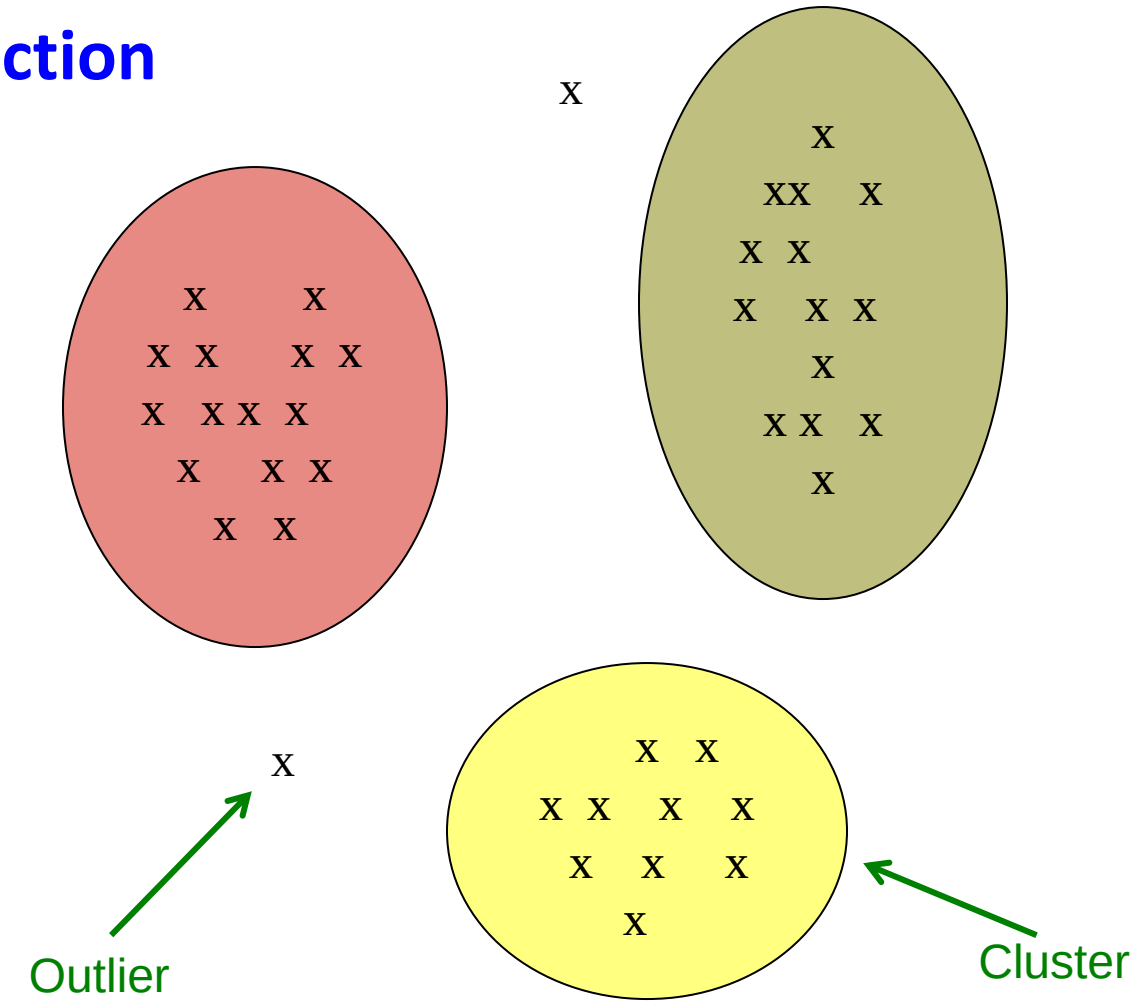
What is Cluster Analysis?

- **Usually:**
  - Points are in a high-dimensional space
  - Similarity is defined using a distance measure
    - » Euclidean, Cosine, Jaccard, edit distance, …

- **Unsupervised learning:** no predefined classes (i.e., learning by observations vs. learning by examples: supervised)

Applications of Cluster Analysis

- **Outlier detection**



Applications of Cluster Analysis

- **Clustering Galaxies**
- A catalog of 2 billion “sky objects” represents objects by their radiation in 7 dimensions (frequency bands)
- Cluster into similar objects, e.g., galaxies, nearby stars, quasars, etc.

Applications of Cluster Analysis

- **Music CDs: Music divides into categories, and customers prefer a few categories**
  - But what are categories really?

- Represent a CD by a set of customers who bought it

- **Similar CDs have similar sets of customers, and vice-versa**

Applications of Cluster Analysis

- **Finding topics of documents:**
- Represent a document by a vector $(x_1, x_2, \ldots, x_k)$, where $x_i = 1$ iff the i^{th} word (in some order) appears in the document

- **Documents with similar sets of words may be about the same topic**

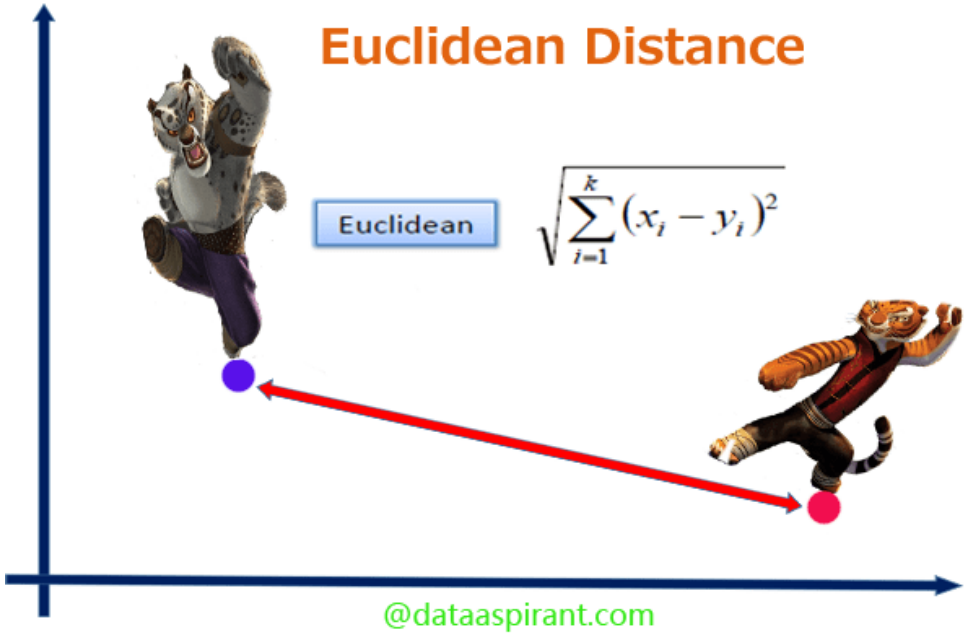
Common Similarity measures

- **Sets as vectors:**
  - Measure similarity by the **cosine distance**
- **Sets as sets:**
  - Measure similarity by the **Jaccard distance**
- **Sets as points:**
  - Measure similarity by **Euclidean distance**

Euclidean distance

- Let $x=(x_1,x_2,..x_k)$ and $y=(y_1,y_2,\ldots,y_k)$ be two vectors
- Euclideaan distance corresponds to the **L2 norm** $\|\cdot\|_2$ **of the difference x-y** between the two vectors

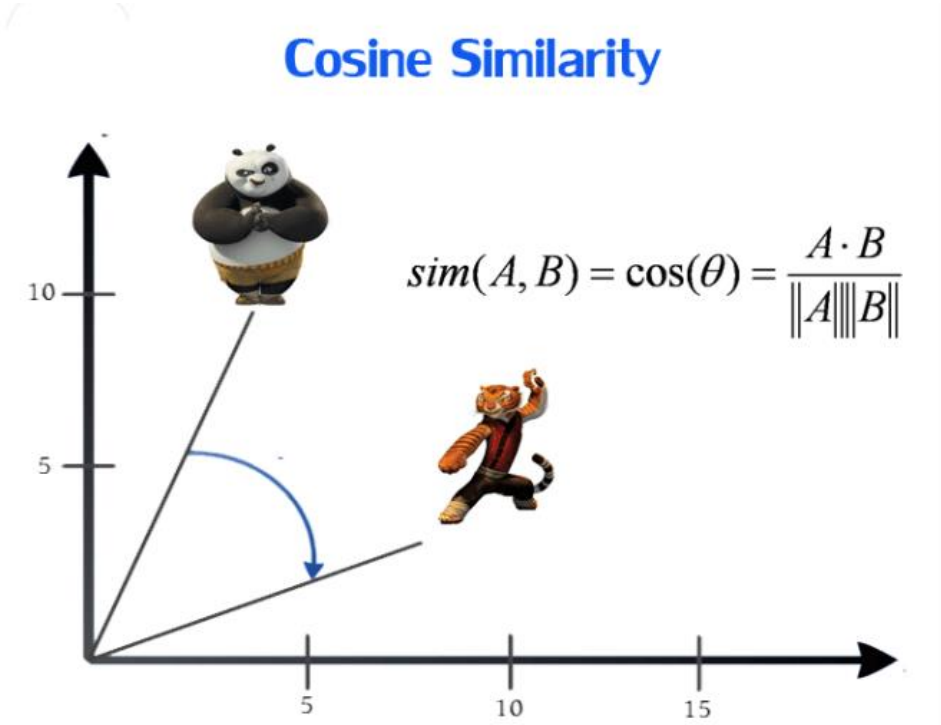
$$||x-y||_2 = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots+(x_n-y_n)^2}$$



Cosine similarity

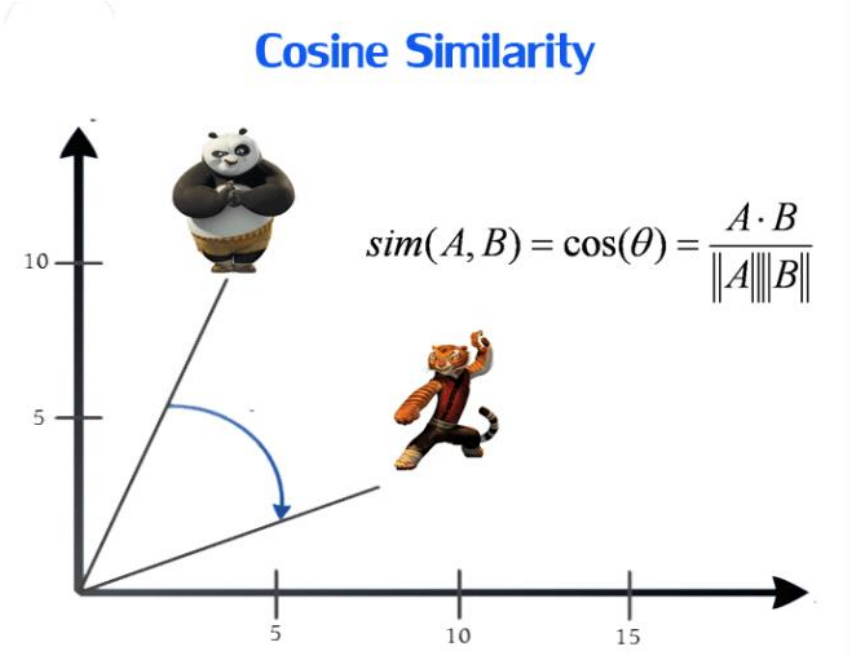
- Determines the **angle between two objects**
- Finds the **normalized dot product** (dot product divided by the product of their magnitudes).

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$



Cosine similarity

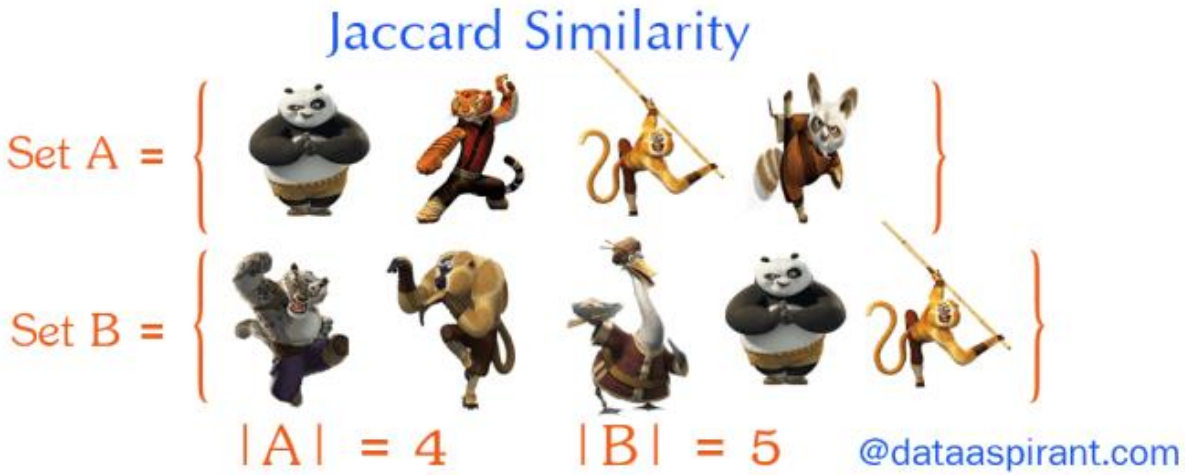
- Cosine similarity is a judgment of **orientation and not magnitude**
- Cosine sim = 1 → two vectors with the same orientation
- Cosine sim = 0 → two vectors oriented at 90° relative to each other



Jaccard similarity

- To compare two objects, jaccrad similarity looks at the **elements they have in common** (the intersection) and **divides it by the number of elements** the two objects have in total (the union).
- range is 0 to 1

$$\text{jaccardSim} = \frac{A \cap B}{A \cup B}$$



Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters?

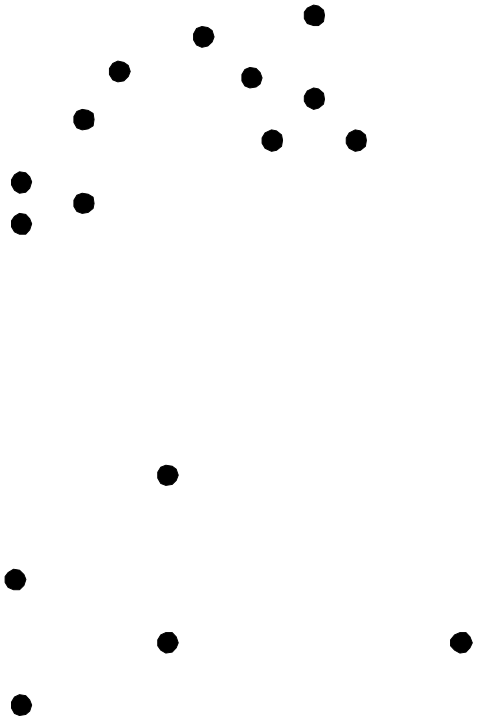
Two Clusters?

Four Clusters?

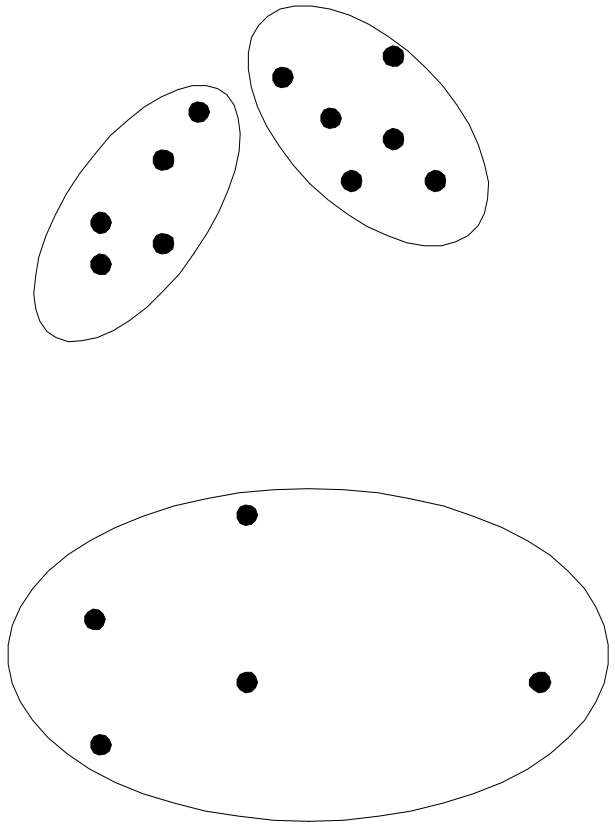
Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- **Partitional Clustering**
  - A division of 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

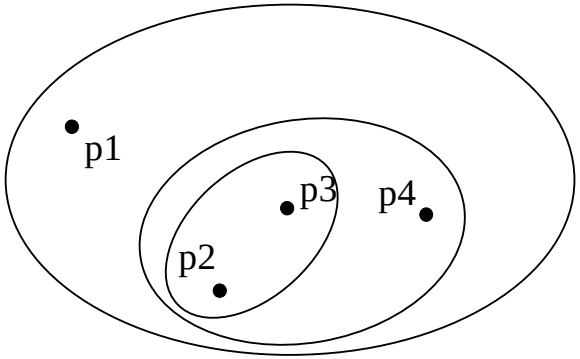


Original Points

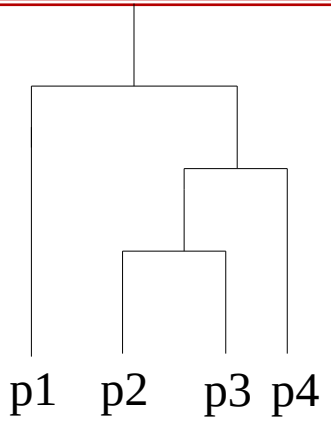


A Partitional Clustering

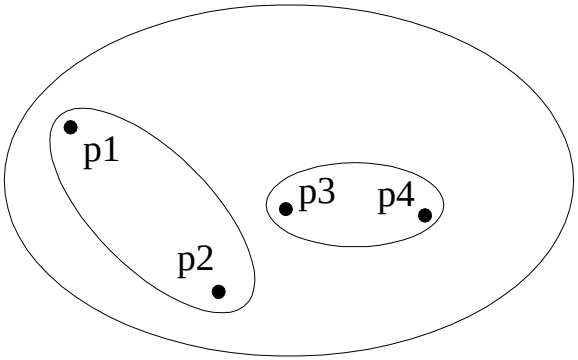
Hierarchical Clustering



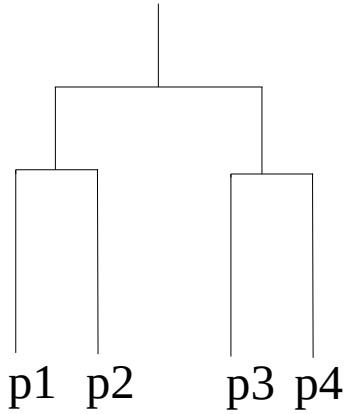
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

- **Exclusive versus non-exclusive**
  - In non-exclusive clustering, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- **Fuzzy versus non-fuzzy**
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- **Partial versus complete**
  - In some cases, we only want to cluster some of the data

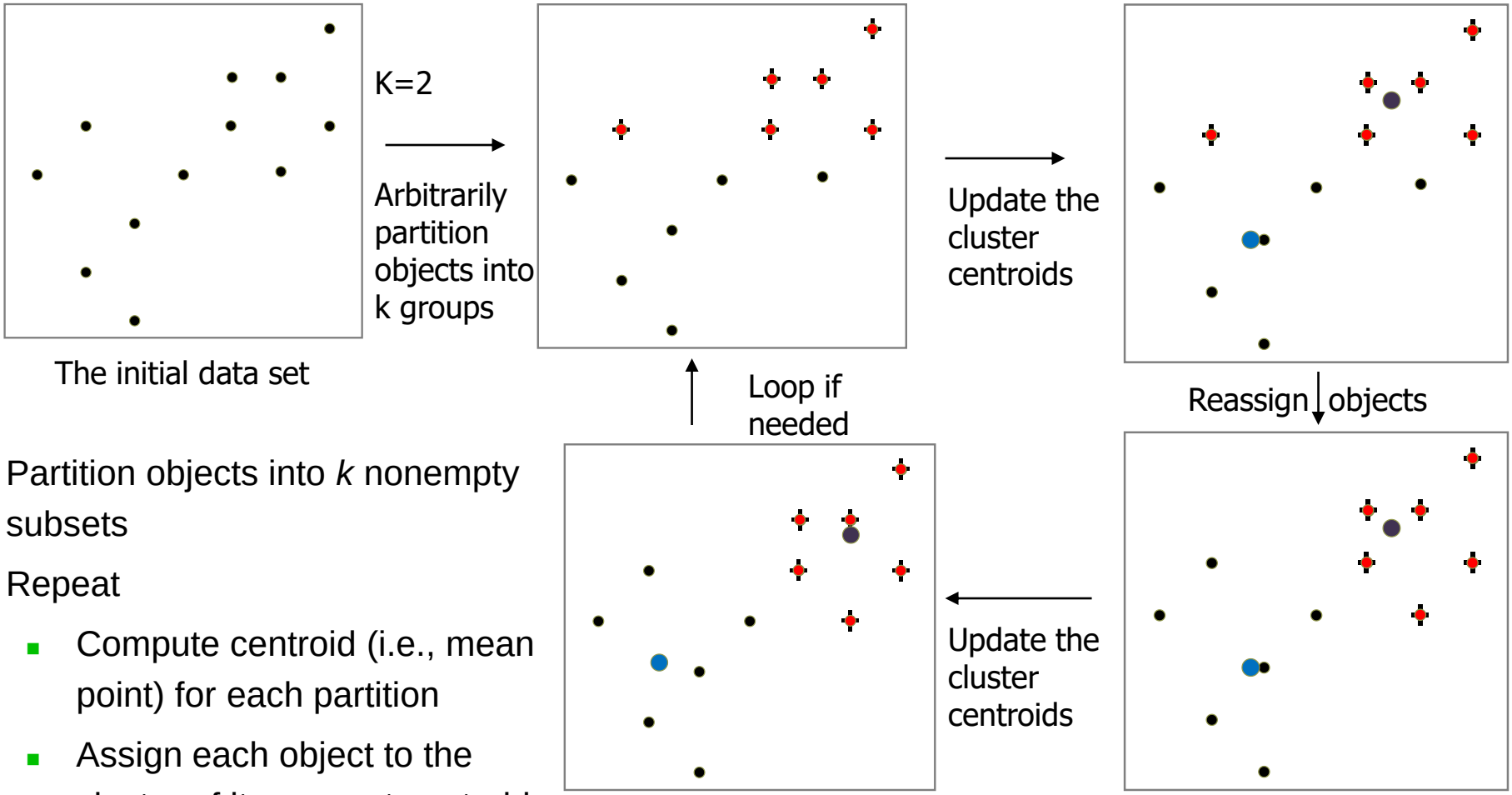
Clustering Algorithms

- **K-means** and its variants
- **Hierarchical clustering**
- Density-based clustering

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  3. Assign each object to the cluster with the nearest centroid point (according to a distance measure)
  4. Go back to Step 2, stop when the assignment does not change

The *K-Means* Clustering Method



1. Partition objects into *k* nonempty subsets
2. Repeat
   - Compute centroid (i.e., mean point) for each partition
   - Assign each object to the cluster of its nearest centroid
- Until no change

Example

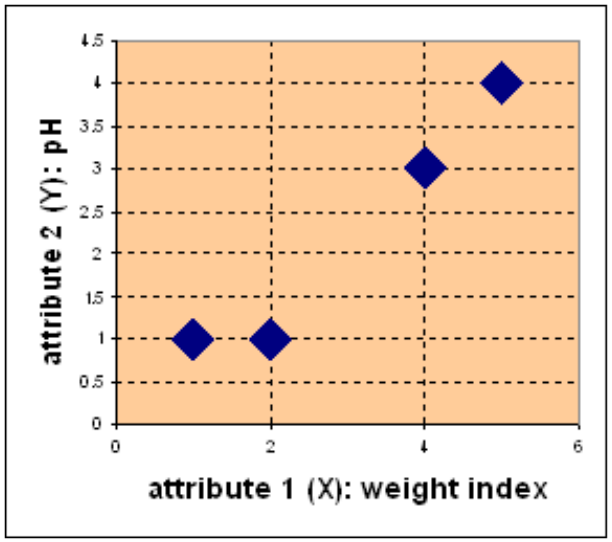
- Suppose we have 4 types of medicines. Each medicine has two types of features

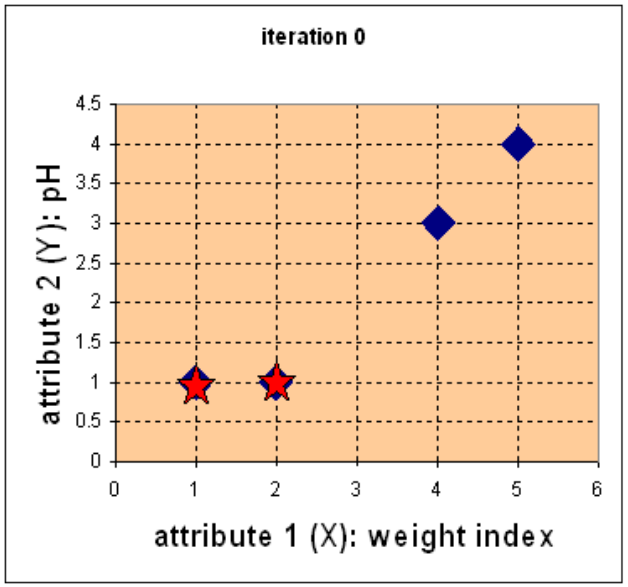
Medicine	weight index	pH
A	1	1
B	2	1
C	4	3
D	5	4

- Goal: group medicines based on their features (2 groups, **K=2**)

Example

- Iteration 0: Choose 2 random centroids
  - We choose c1=(1,1) → A
  - and c2=(2,1) → B





Example

- Iteration 1: compute distance to centroids (e.g. using Euclidean)

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{matrix}$$

- Each column of the matrix represents one object (one point, one row)
- First row → distance of each object to the first centroid (c1)
- Second row → distance of each object to the second centroid (c2)

Example

- Iteration 1: compute distance to centroids (e.g. using Euclidean)

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{matrix}$$

Example: distance from medicine C=(4,3) to c1=(1,1) is

$$\sqrt{(4-1)^2 + (3-1)^2} = 3.61$$

Example

- Iteration 2: objects clustering. We assign each object to a cluster based on the minimum distance

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{matrix}$$

$$\mathbf{G}^0 = \begin{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \\ A \quad B \quad C \quad D \end{matrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

Cluster 1 contains only A

Cluster 2 contains B, C, D

Example

- Iteration 3: repeat. Compute new centroids of each group

Cluster 1 contains only A = (1,1)

Cluster 2 contains B = (2,1), C=(4,3), D=(5,4)

Cluster 1 only one point → c1 = (1,1)

Cluster 2: centroid is the average coordinate

$$\mathbf{c}_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$$

Example

- Iteration4: repeat. Compute distance to new centroids
  - C1 = (1,1), c2 = (11/3, 8/3)

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

- Iteration 5: objects clustering based on minimum distance
  - cluster 1: A,B
  - Cluster 2: C,D

$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$
$$A \quad B \quad C \quad D$$

Example

- Iteration6: repeat. Compute new centroids
  - c1=(3/2, 1), c2=(9/2, 7/2)

- Iteration 7: compute distance to new centroids

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1\frac{1}{2}, 1) & group-1 \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

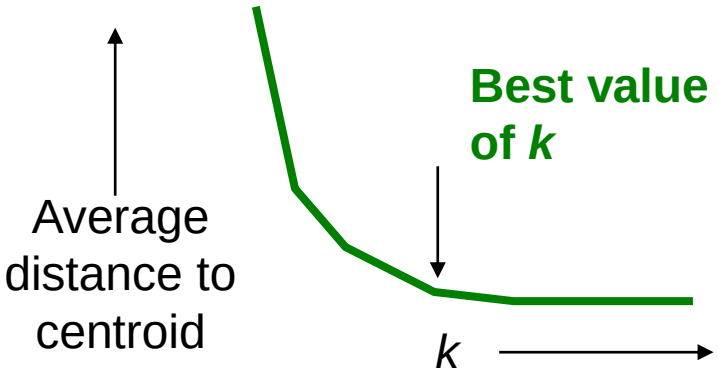
- Iteration 8: create clusters
  - Cluster 1: A,B
  - Cluster 2: C,D
  - **No changes. Stop**

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$
$$A \quad B \quad C \quad D$$

Getting the *k* right

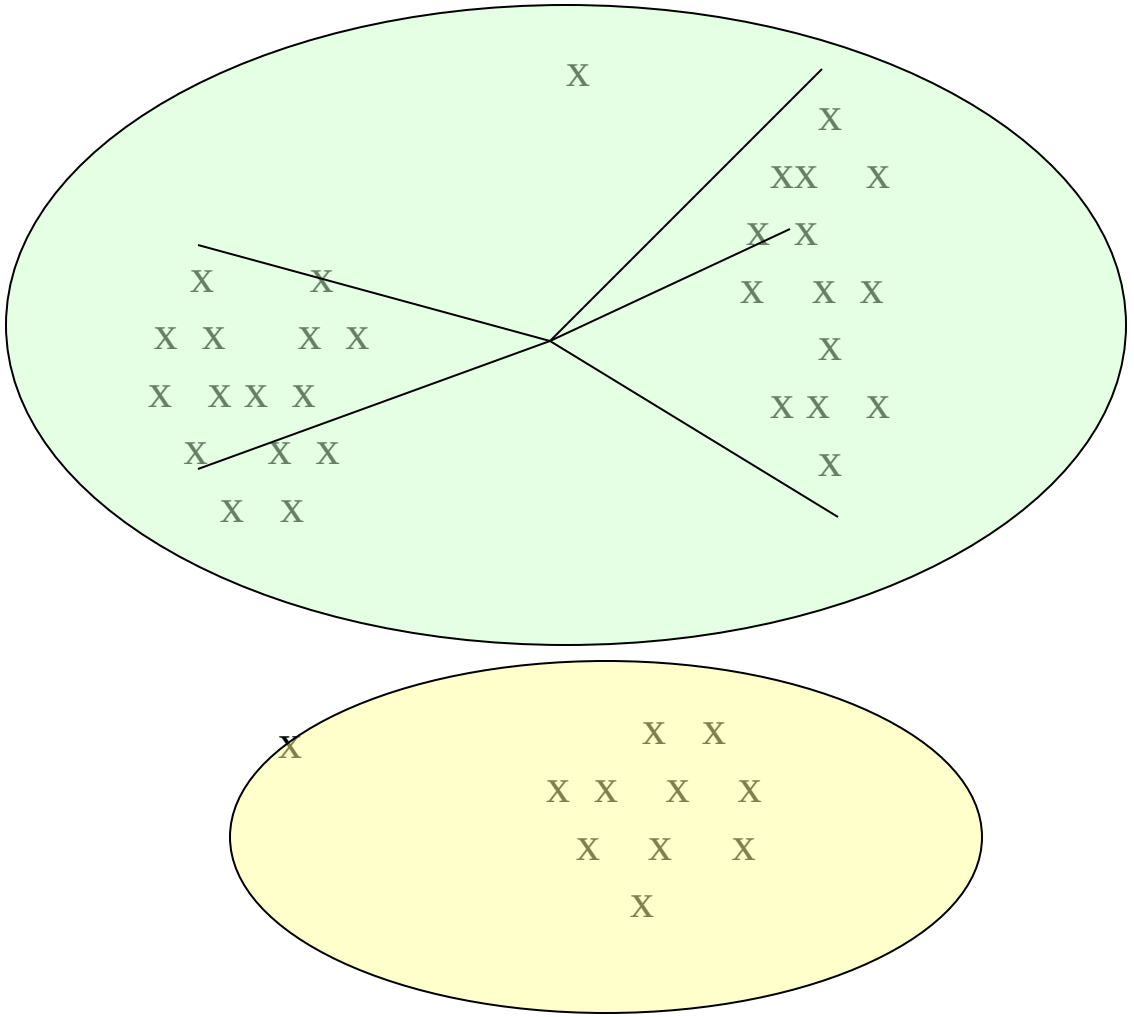
How to select *k*?

- Try different **k**, looking at the change in the average distance to centroid as **k** increases
- Average falls rapidly until right **k**, then changes little



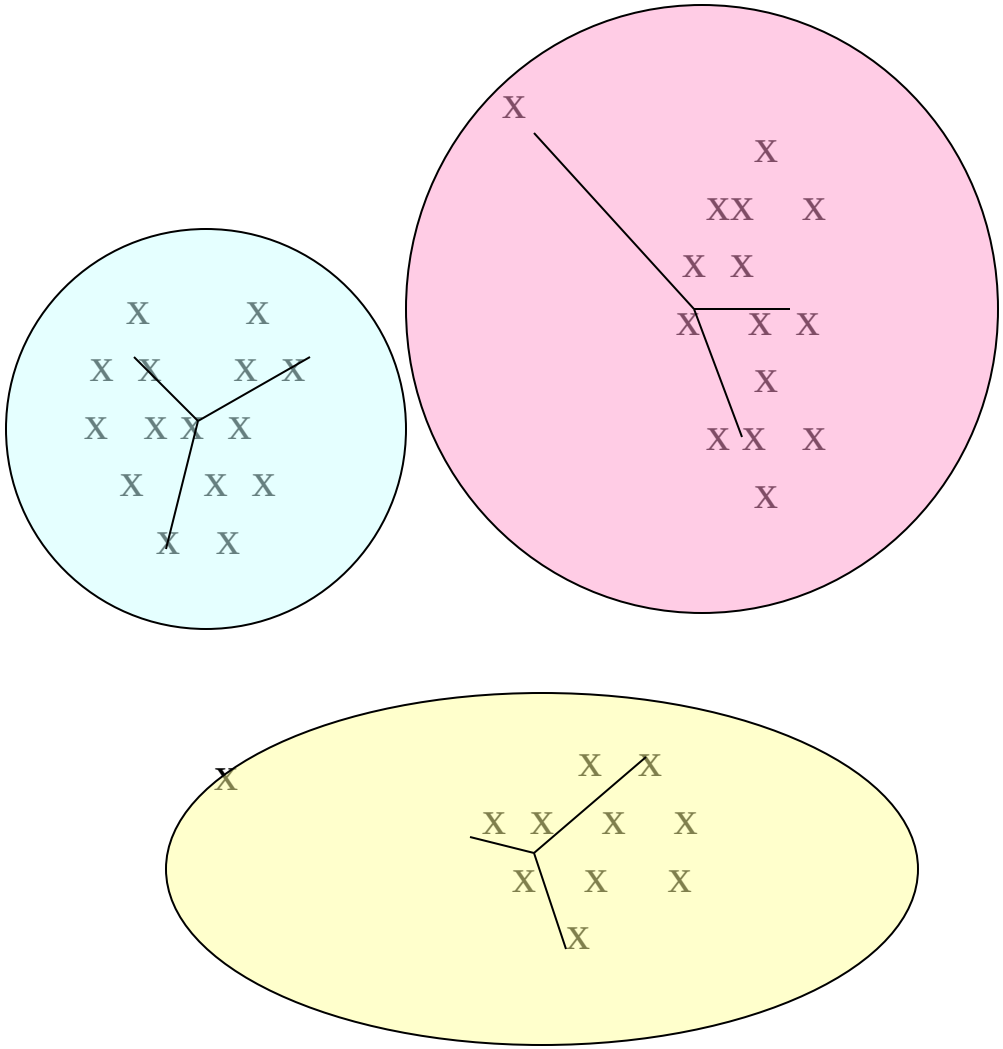
Example: Picking *k*

Too few;
many long
distances
to centroid.



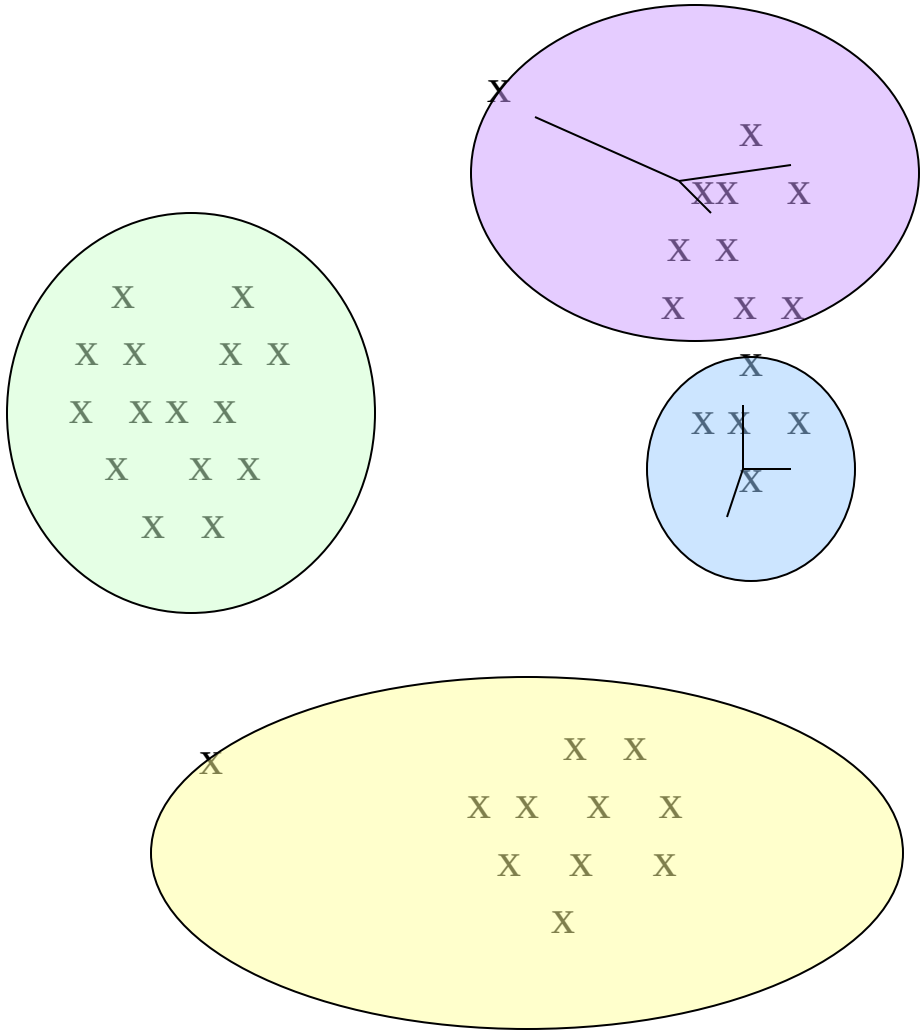
Example: Picking *k*

Just right;
distances
rather short.



Example: Picking *k*

Too many;
little improvement
in average
distance.



Evaluating K-means Clusters

- Most common measure is **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 for cluster C_i
- Given two sets of clusters, we prefer the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Comments on the *K-Means* Method

- Strength: Efficient (polynomial time)

- Weakness
  - Applicable only to objects in a continuous n-dimensional space
  - Need to specify k, the number of clusters, in advance (there are ways to automatically determine the best k
  - Sensitive to noisy data and outliers

Variations of the *K-Means* Method

- Most of the variants of the *k-means* differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- **Handling categorical data: *k-modes***
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a **frequency-based** method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

Pre-processing and Post-processing

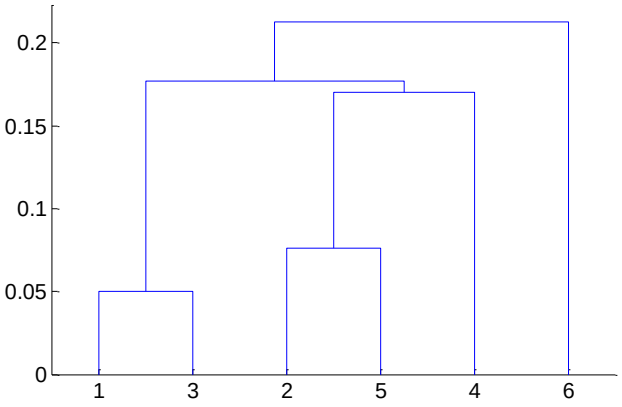
- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

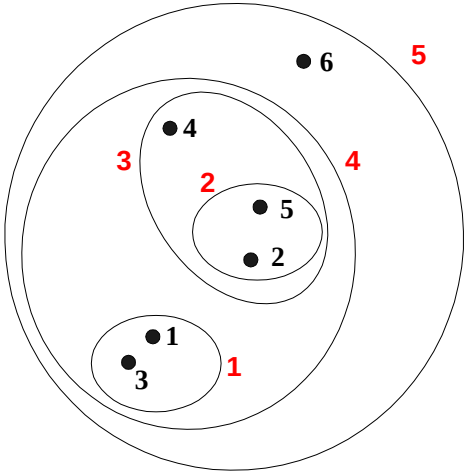
Clustering Algorithms

- K-means and its variants
- **Hierarchical clustering**
- Density-based clustering

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 dendrogram** at the proper level (termination condition)

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

Hierarchical Clustering

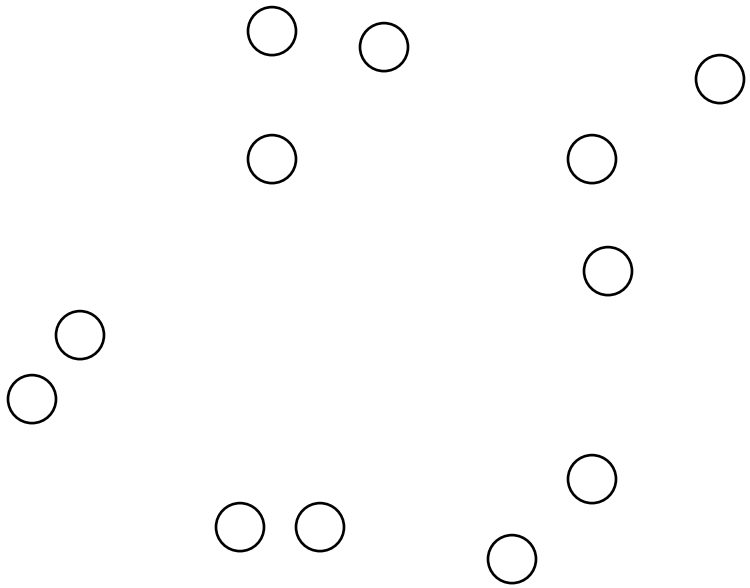
- Two main types of hierarchical clustering
  - **Agglomerative (bottom-up approach)**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - **Divisive (top-down approach)**
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains an individual 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

- Most 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

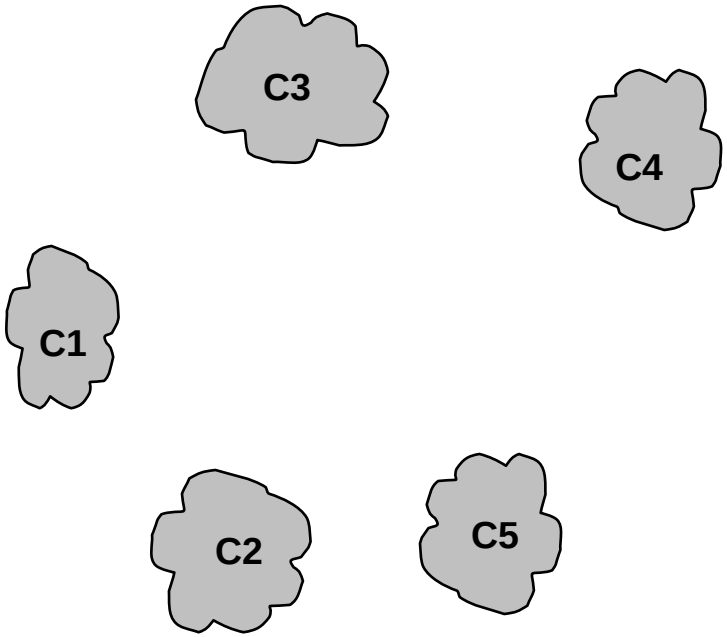


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

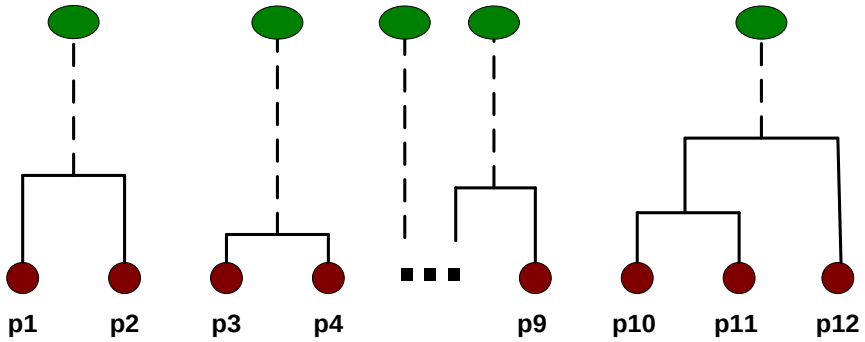
Intermediate Situation

- After some merging steps, we have some clusters



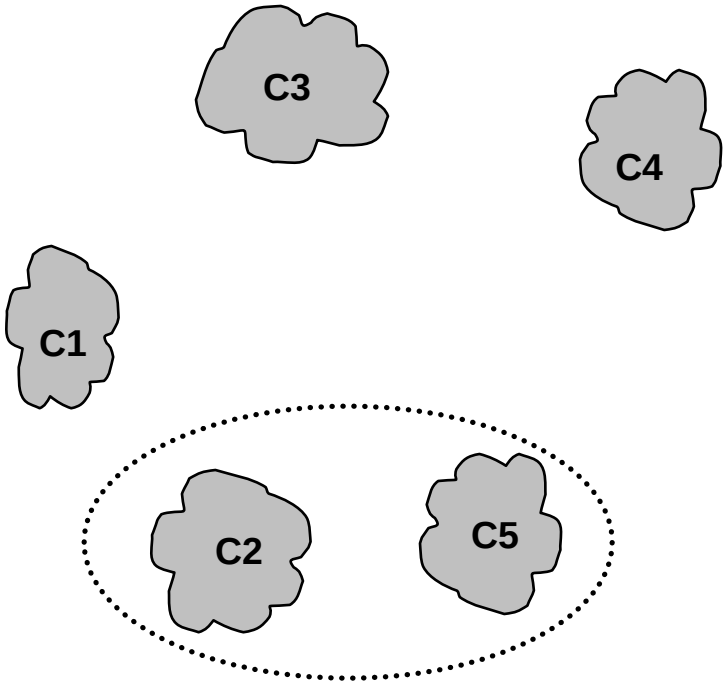
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



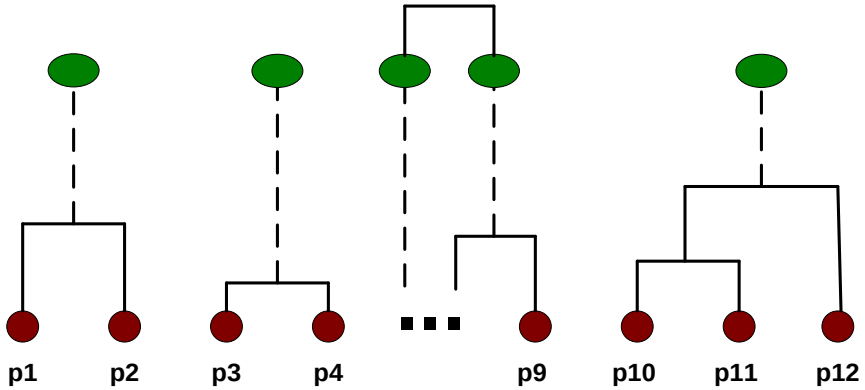
Intermediate Situation

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



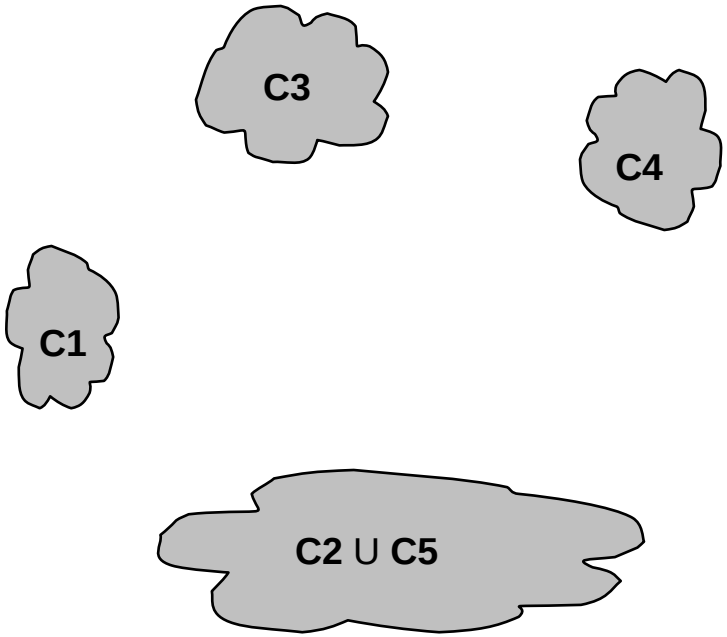
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



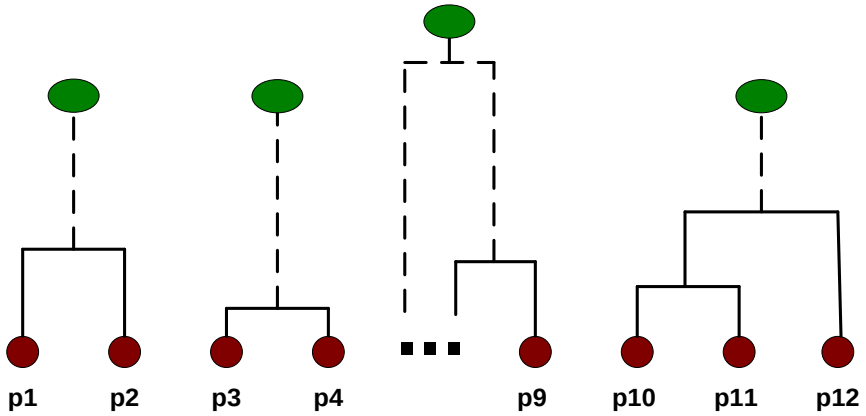
After Merging

- The question is **"How do we update the proximity matrix?"**

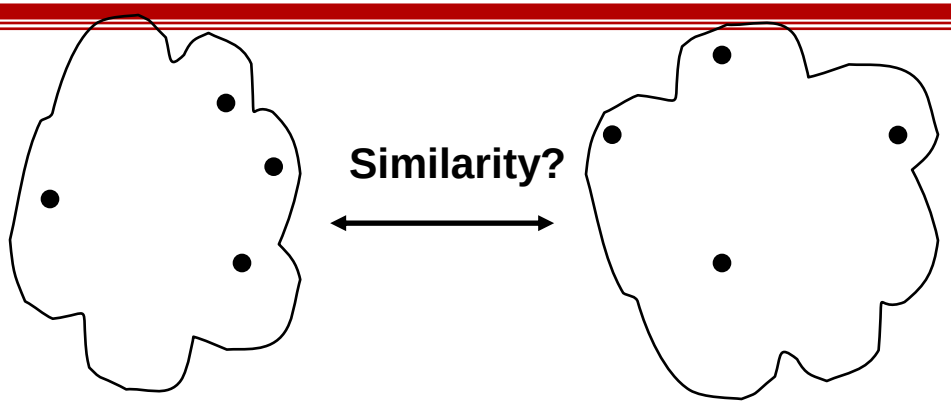


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Distance

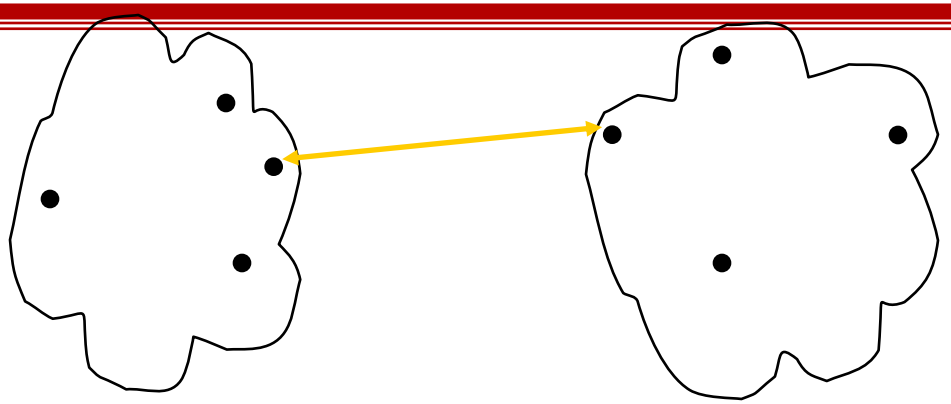


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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 Distance

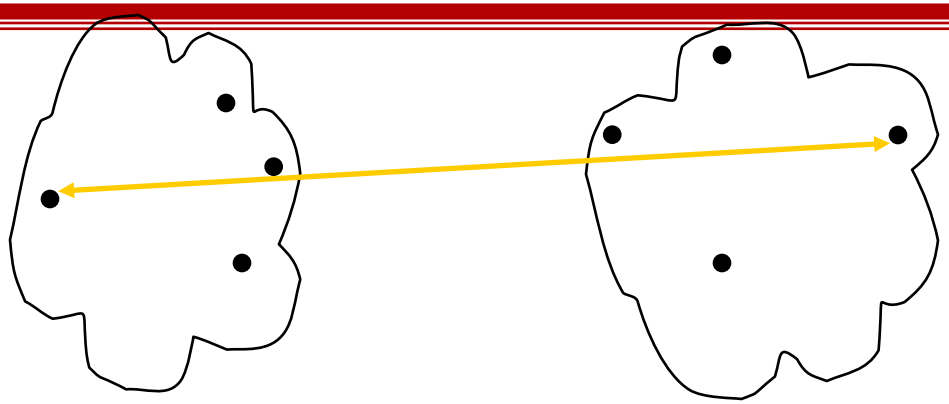


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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 Distance

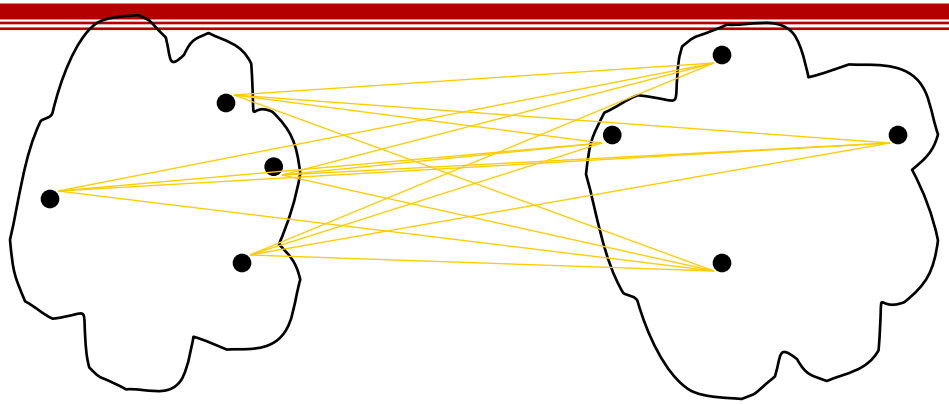


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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 Distance

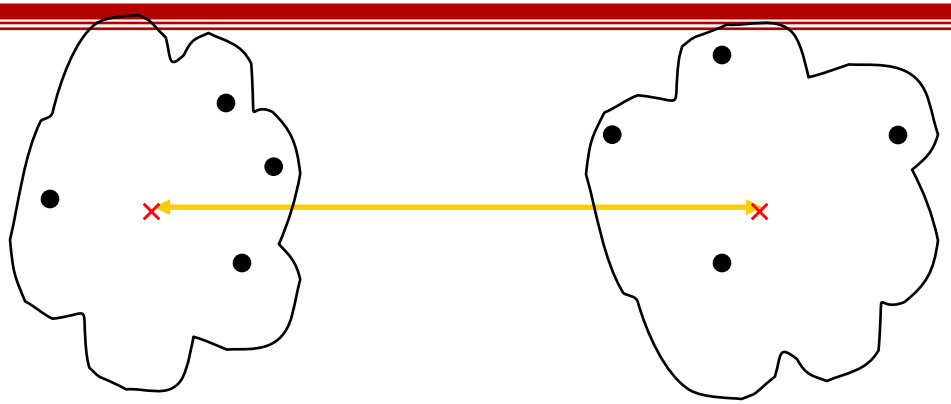


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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 Distance



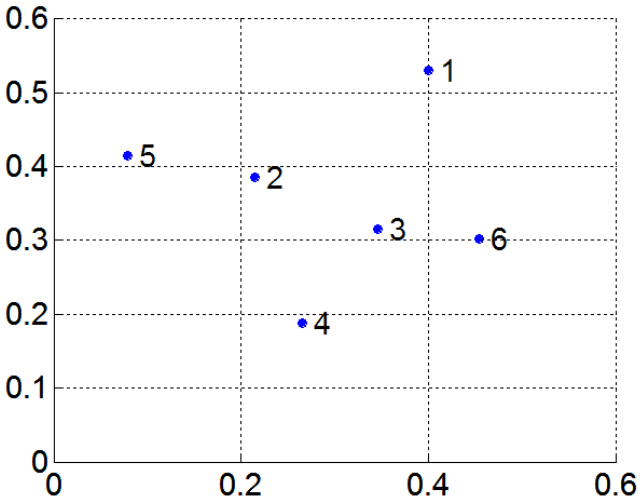
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

MIN or Single Link

- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph
- Example:

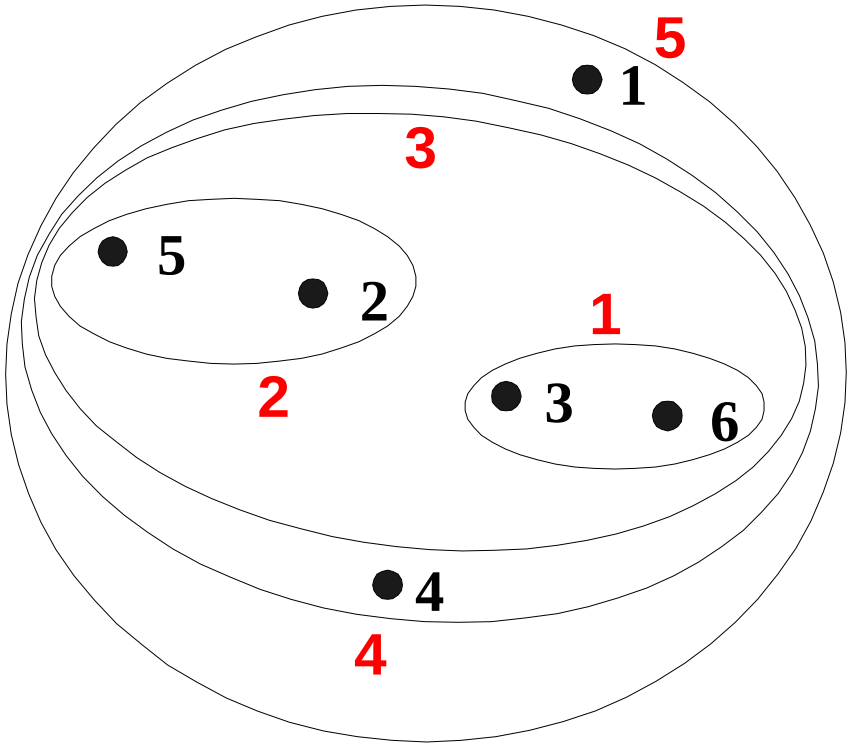


Distance Matrix:

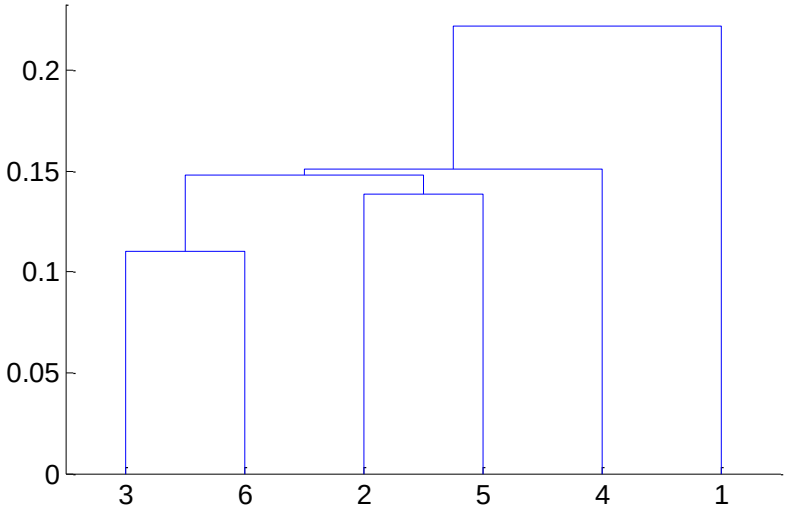
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MIN

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

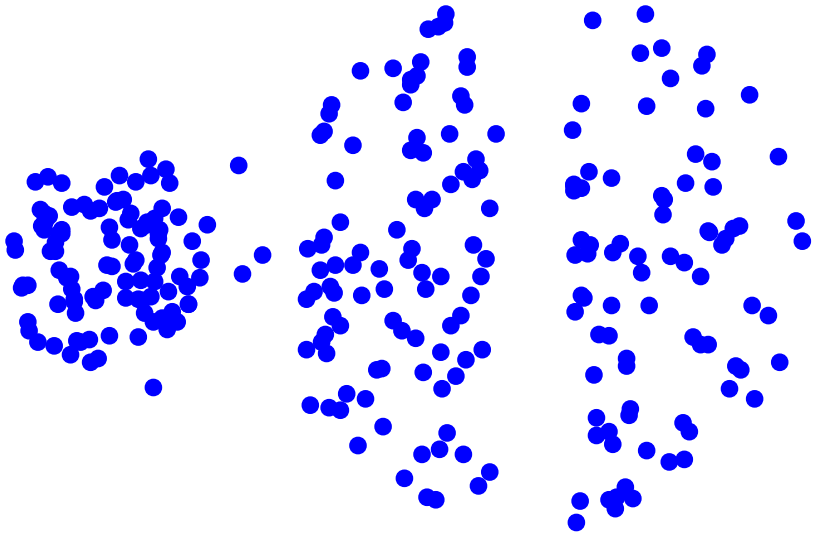


Nested Clusters

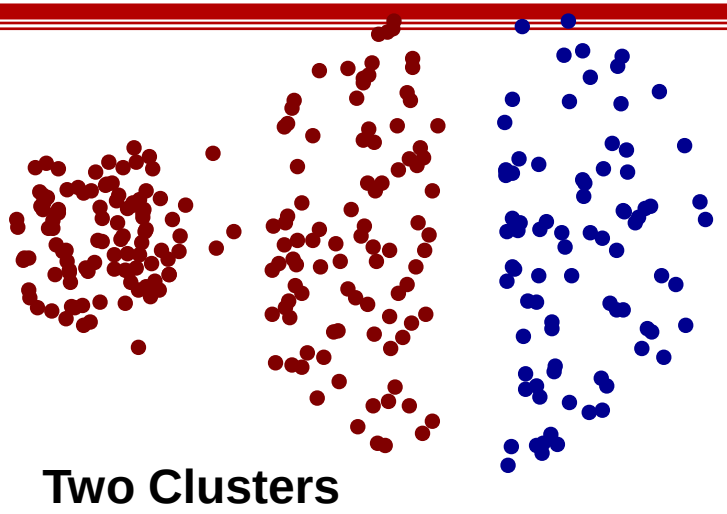


Dendrogram

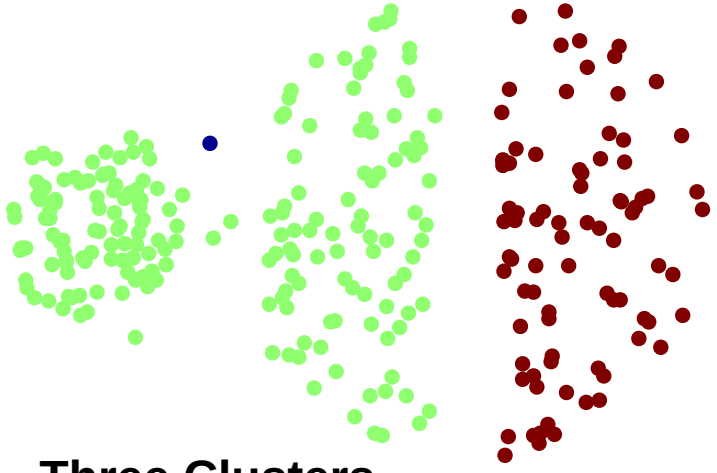
Limitations of MIN



Original Points



Two Clusters

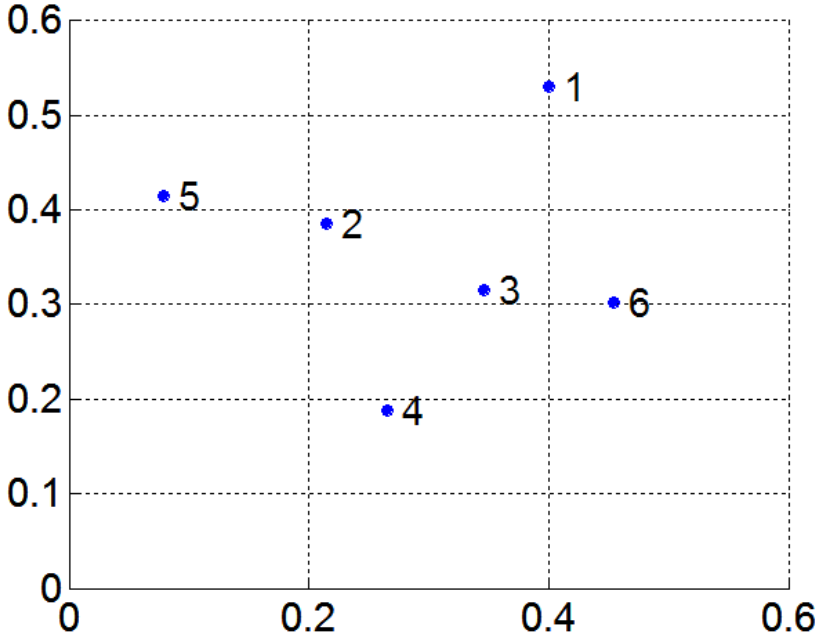


Three Clusters

- Sensitive to noise and outliers

MAX or Complete Linkage

- Proximity of two clusters is based on the two most distant points in the different clusters
  - Determined by all pairs of points in the two clusters

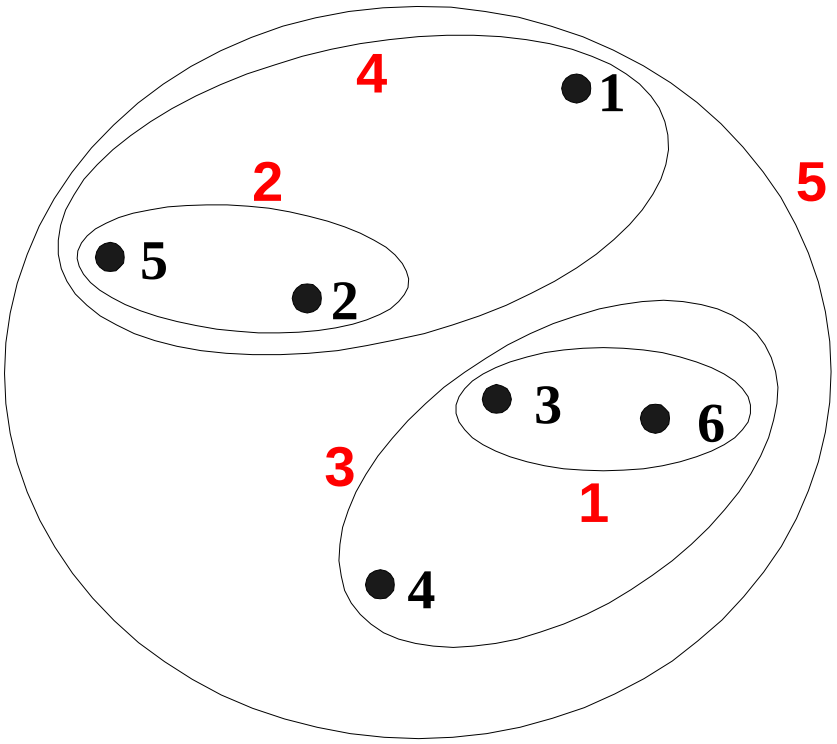


Distance Matrix:

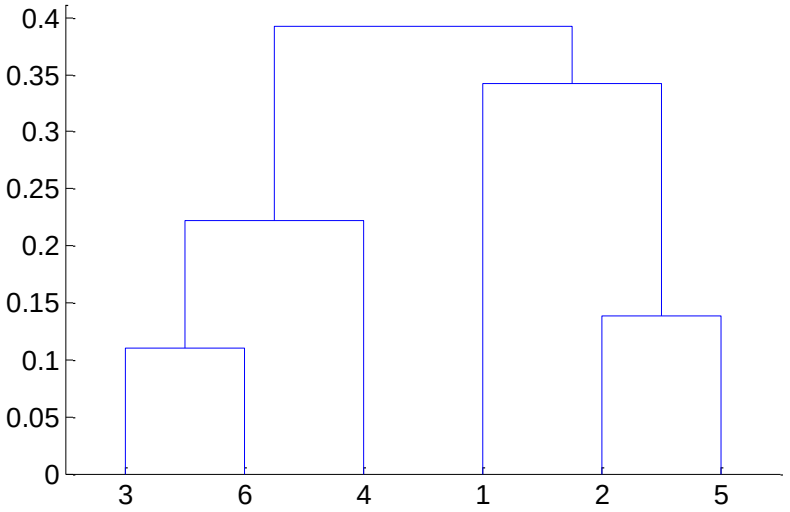
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MAX

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

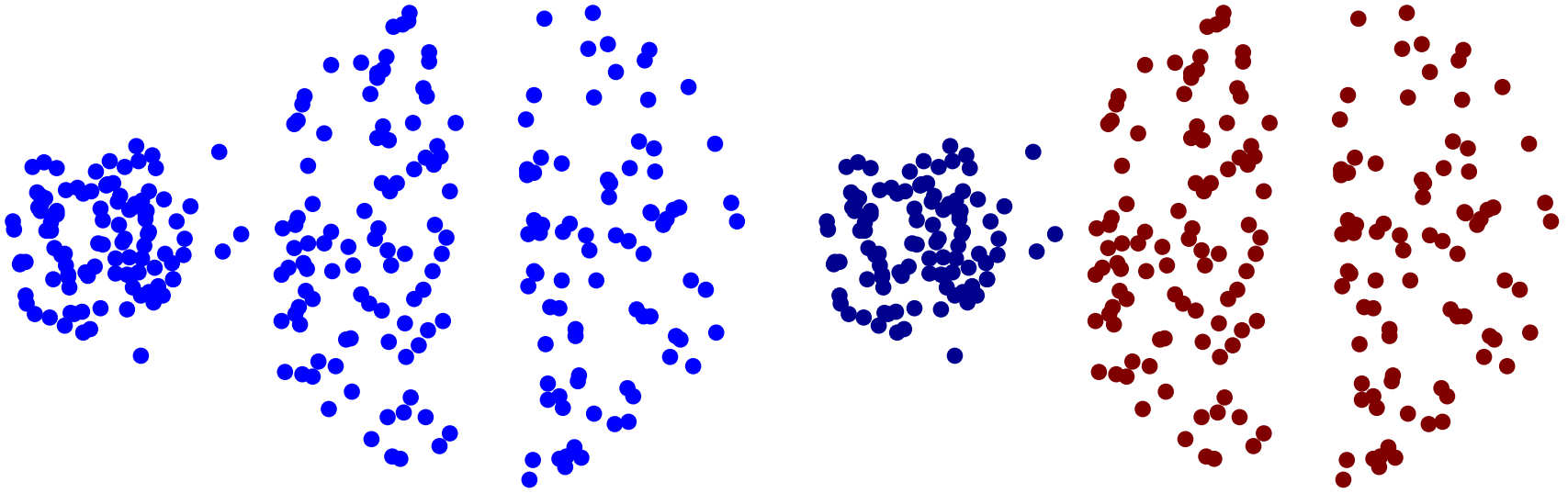


Nested Clusters



Dendrogram

Strength of MAX

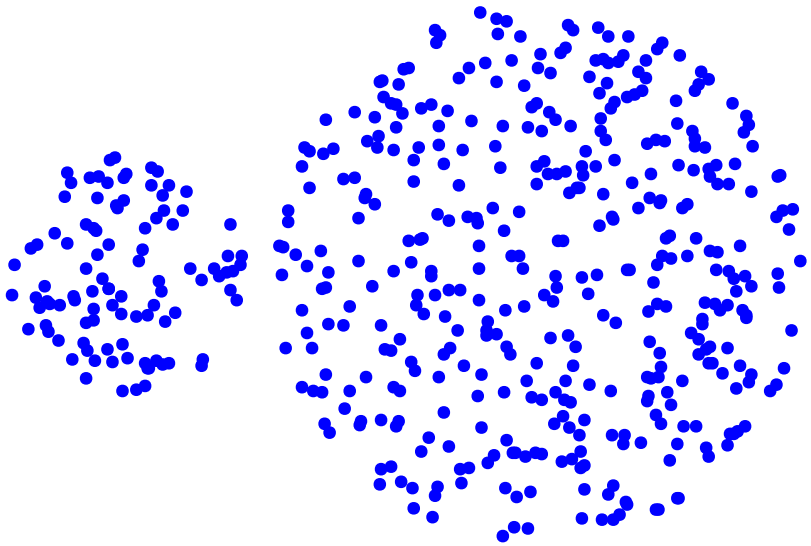


Original Points

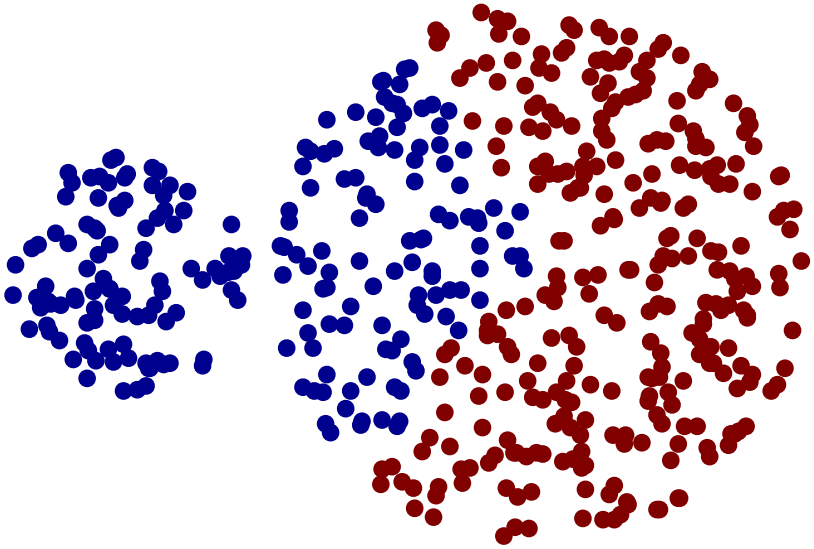
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

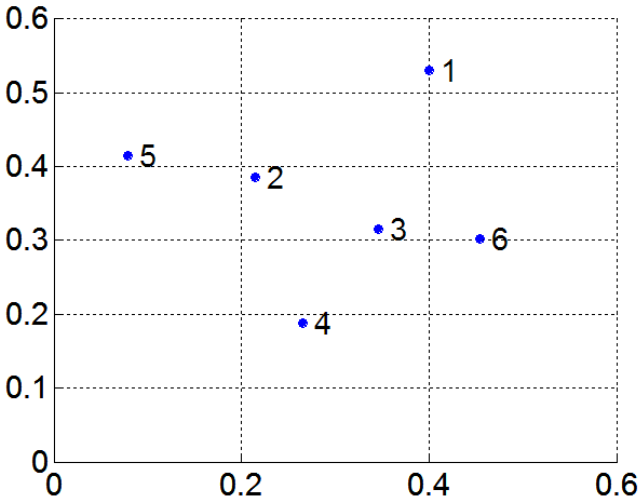
- Tends to break large clusters
- Biased towards globular clusters

Group Average

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

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| \times |\text{Cluster}_j|}$$

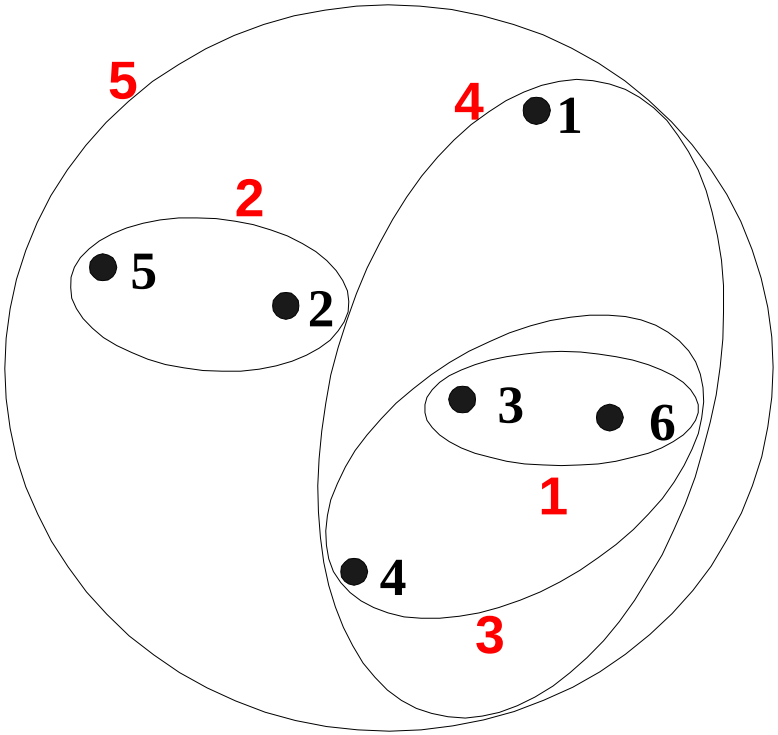
- Need to use average connectivity for scalability since total proximity favors large clusters



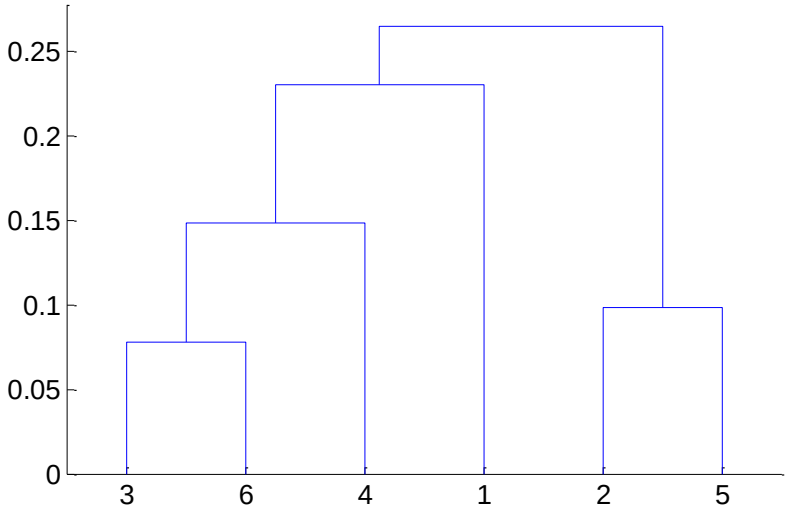
Distance Matrix:

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: Group Average



Nested Clusters

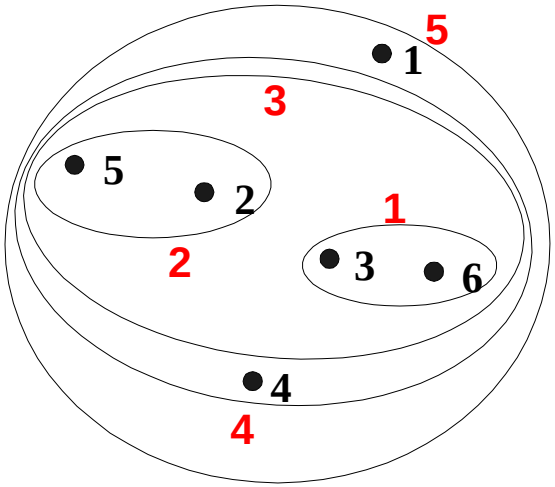


Dendrogram

Hierarchical Clustering: Group Average

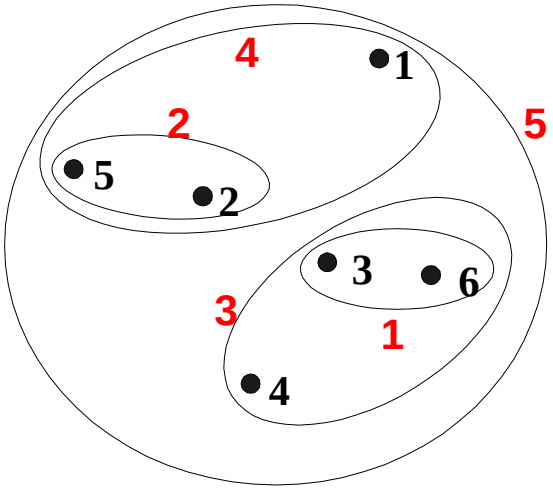
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

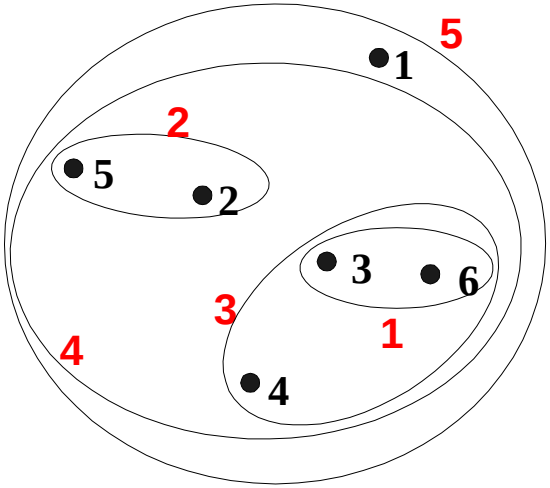
Hierarchical Clustering: Comparison



MIN

MAX





Group Average

Clustering evaluation methods

- **Supervised:** Compare the clustering result to a ground truth

- **Unsupervised:** No ground truth is available

Unsupervised Clustering Evaluation

- **Two metrics:**

- **Silhouette Coefficient:**

- a higher Silhouette Coefficient score relates to a model with better defined clusters. The Silhouette Coefficient is defined for each sample and is composed of two scores:
- **a**: The mean distance between a sample and all other points in the same class.
- **b**: The mean distance between a sample and all other points in the *next nearest cluster*.
- The Silhouette Coefficient *s* for a single sample is then given as: