Lecture 6
Introduction to Clustering

What is clustering?

In clustering or unsupervised learning no training data, with class labeling, are available. The goal becomes:

Group the data into a number of sensible clusters (groups).

This unravels similarities and differences among the available data.

⇨ Applications:

→Engineering

→Bioinformatics

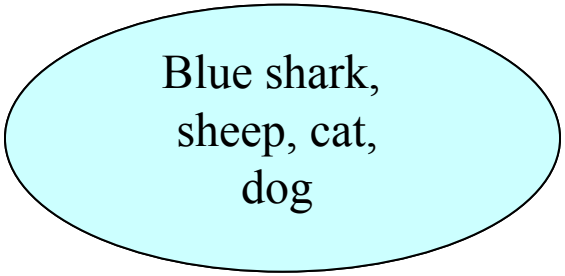
→Social Sciences

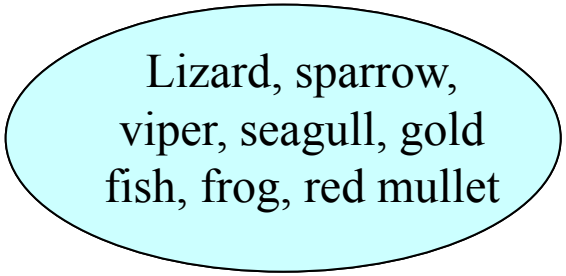
→Medicine

→Data and Web Mining

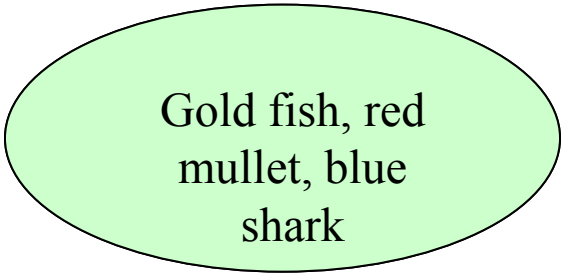
⇨ To perform clustering of a data set, a clustering criterion must first be adopted. Different clustering criteria lead, in general, to different clusters.

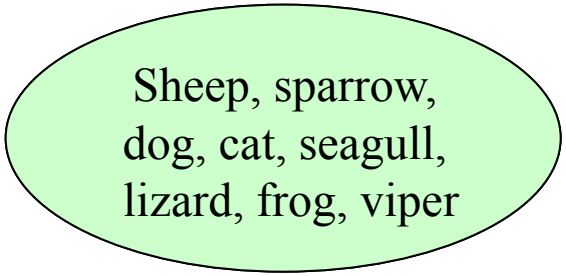
⇨ A simple example





1. Two clusters
2. Clustering criterion: *How mammals bear their progeny*





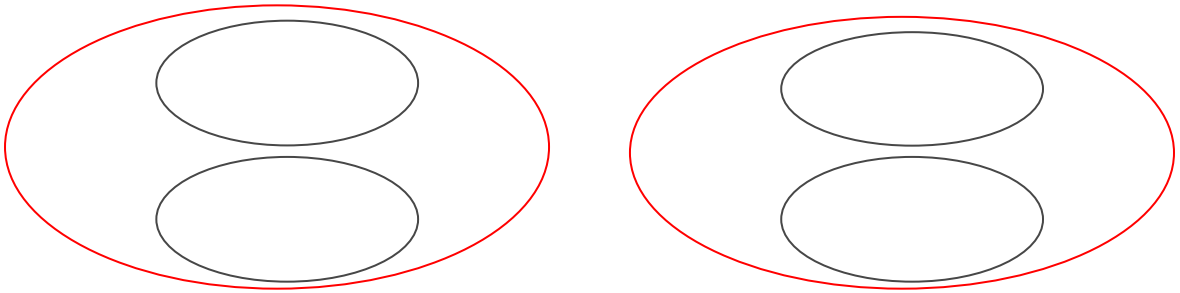
1. Two clusters
2. Clustering criterion: *Existence of lungs*

➔Clustering task stages

- ⇨ **Feature Selection**: Information rich features-Parsimony
- ⇨ **Proximity Measure**: This quantifies the term similar or dissimilar.
- ⇨ **Clustering Criterion**: This consists of a cost function or some type of rules.
- ⇨ **Clustering Algorithm:** This consists of the set of **steps** followed to reveal the structure, based on the similarity measure and the adopted criterion.
- ⇨ Validation of the results.
- ⇨ Interpretation of the results.

⇨ Depending on the similarity measure, the clustering criterion and the clustering algorithm different clusters may result. **Subjectivity** is a reality to live with from now on.

⇨ A simple example: How many clusters??



2 or 4 ??

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

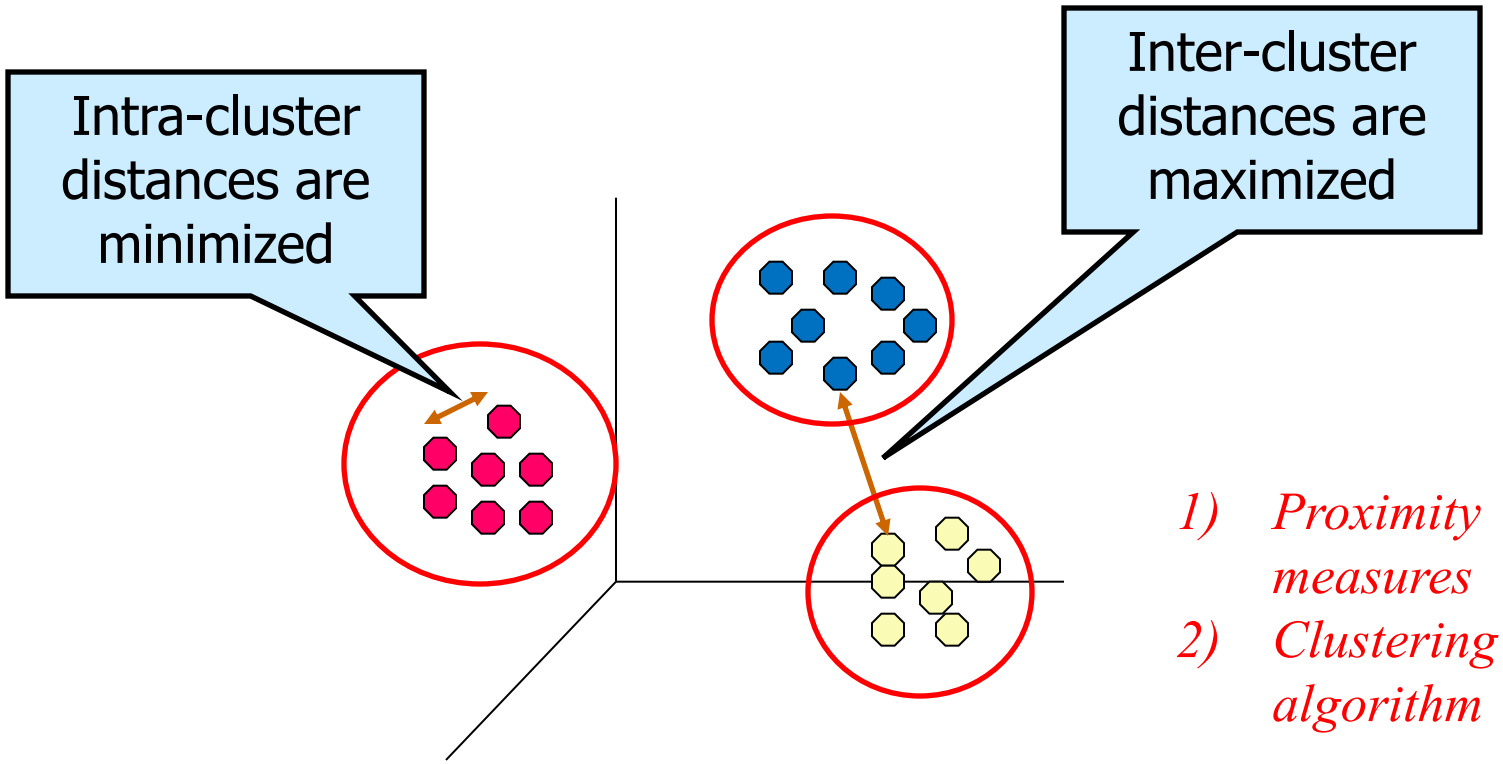
Four Clusters

➔ Basic application areas for clustering

- ⇨ Data reduction. All data vectors within a cluster are substituted (represented) by the corresponding cluster representative.
- ⇨ Hypothesis generation.
- ⇨ Hypothesis testing.
- ⇨ Prediction based on groups.

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**



1) Proximity measures
2) Clustering algorithm

Types of Clusterings

➔ **A clustering is a set of clusters**

➔ **Important distinction between hierarchical and partitional sets of clusters**

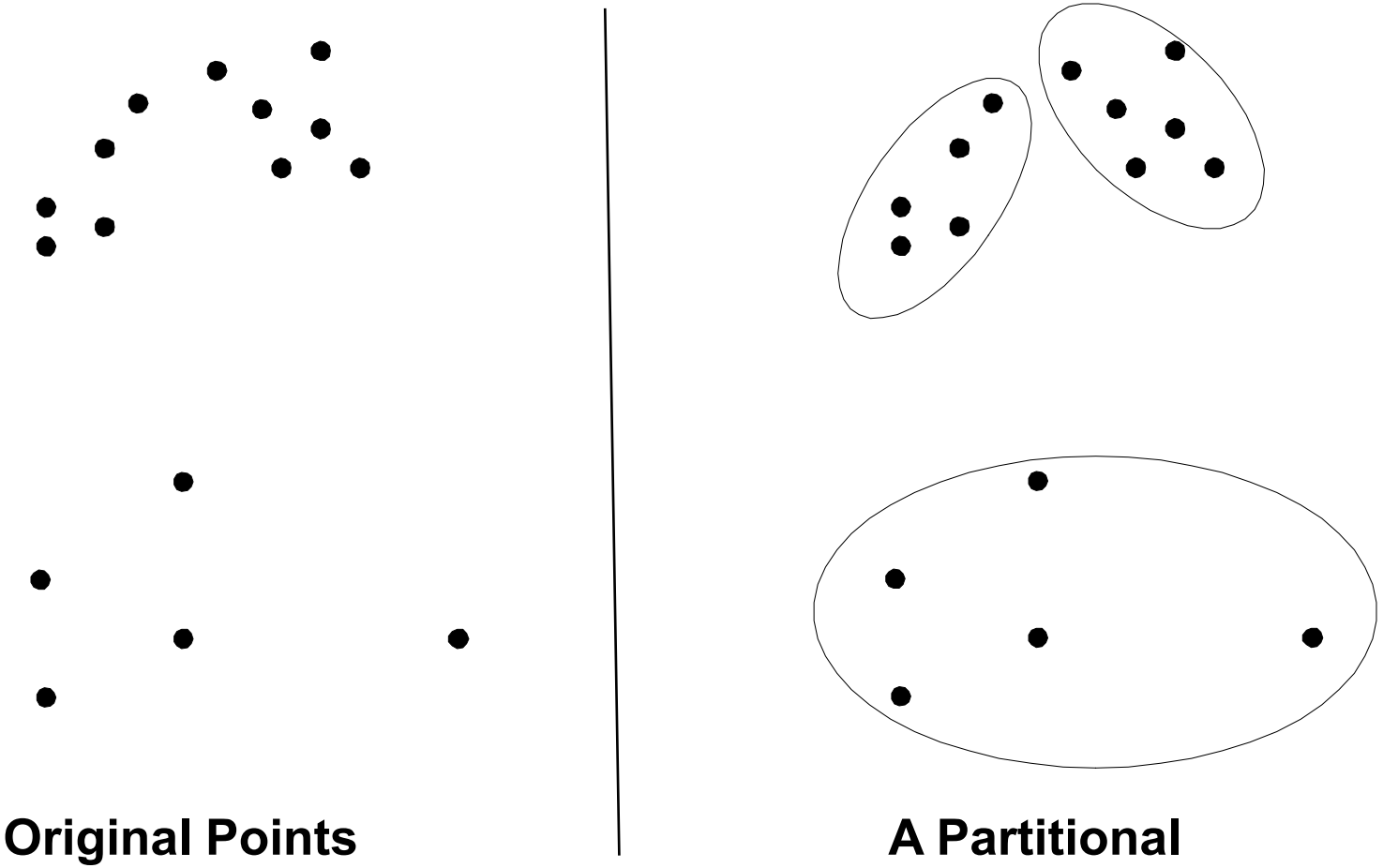
➔ **Partitional Clustering**

⇨ 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

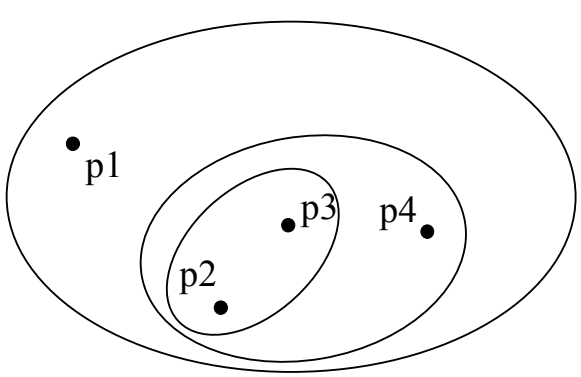
Partitional Clustering



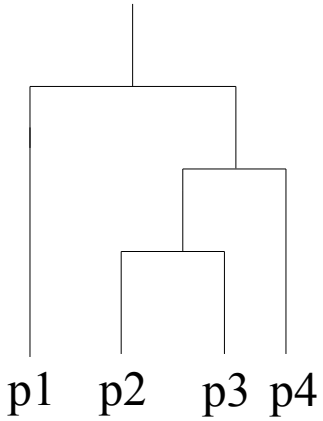
Original Points

A Partitional Clustering

Hierarchical Clustering

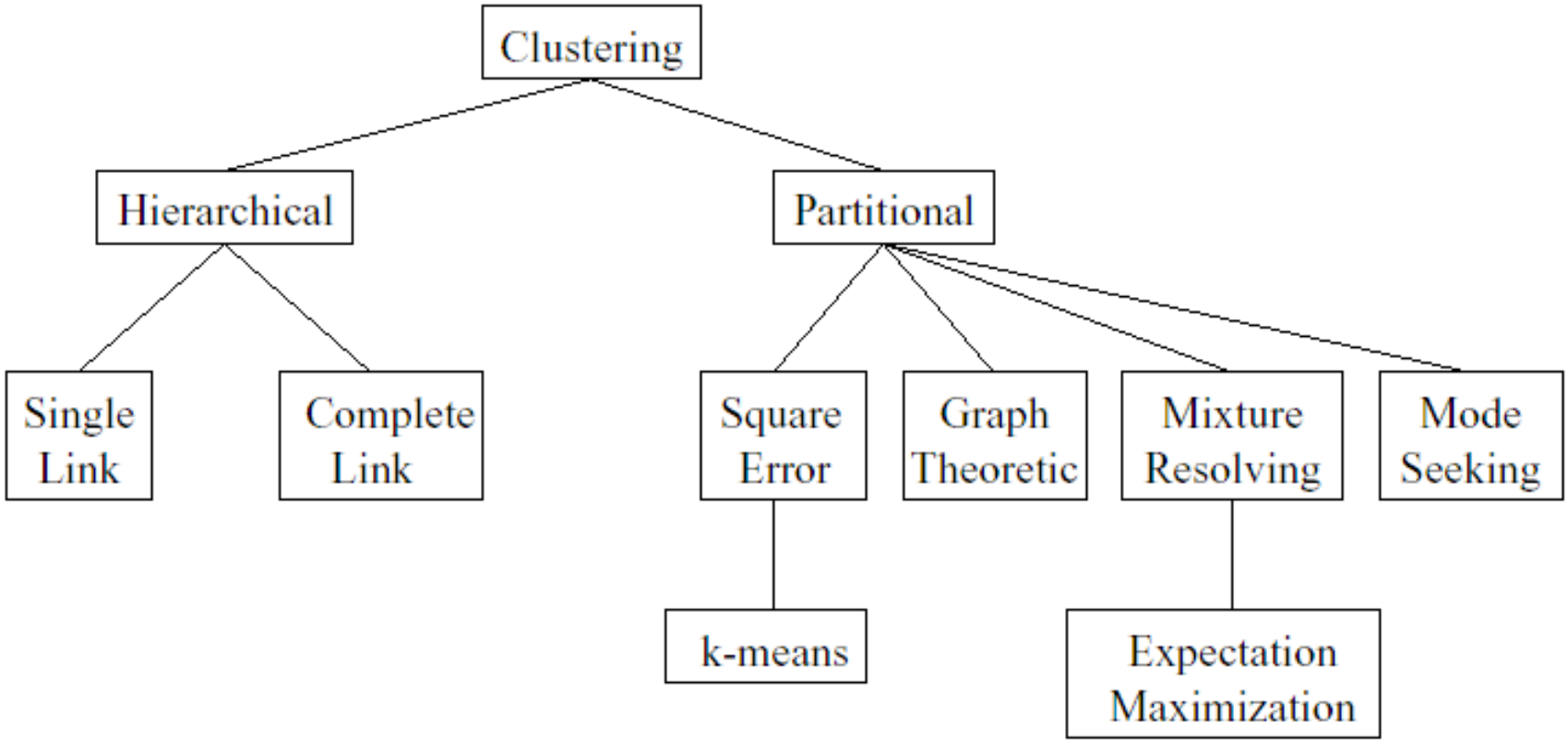


Traditional Hierarchical Clustering



Traditional Dendrogram

Taxonomy of Clustering Approaches



➔ Clustering Definitions

⇨ Hard Clustering: Each point belongs to a single cluster

➔Let $X = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$

➔An m-clustering R of X, is defined as the partition of X into m sets (clusters), $C_1, C_2, \ldots, C_m$, so that

» $C_i \neq \varnothing, i = 1,2,\ldots,m$

» $\bigcup_{i=1}^{m} C_i = X$

» $C_i \cap C_j = \varnothing, \ i \neq j, \ i, j = 1,2,\ldots,m$

In addition, data in C_i are more similar to each other and less similar to the data in the rest of the clusters. Quantifying the terms similar-dissimilar depends on the types of clusters that are **expected** to underlie the structure of X.

⇨ **Fuzzy clustering**: Each point belongs to all clusters up to some degree.

A fuzzy clustering of X into m clusters is characterized by m functions

→ $u_j : \underline{x} \rightarrow [0,1], \quad j = 1,2,\ldots,m$

→ $\sum_{j=1}^{m} u_j(\underline{x}_i) = 1, \quad i = 1,2,\ldots,N$

→ $0 < \sum_{i=1}^{N} u_j(\underline{x}_i) < N, \quad j = 1,2,\ldots,m$

These are known as membership functions. Thus, each $\underline{x}_i$ belongs to any cluster "up to some degree", depending on the value of

$$u_j(\underline{x}_i), \quad j = 1,2,\ldots,m$$

$u_j(\underline{x}_i)$ close to $1 \Rightarrow$ high grade of membership of $\underline{x}_i$ to cluster j.

$u_j(\underline{x}_i)$ close to $0 \Rightarrow$ low grade of membership.

Proximity Measures

Which Proximity?

➔ ***Between vectors***

⇨ Dissimilarity measure (between vectors of X) is a function

$$d: \; X \times X \longrightarrow \Re$$

with the following properties

→ $\exists d_0 \in \Re: \; -\infty < d_0 \le d(\underline{x}, \underline{y}) < +\infty, \;\; \forall \underline{x}, \underline{y} \in X$

→ $d(\underline{x}, \underline{x}) = d_0, \;\; \forall \underline{x} \in X$

→ $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x}), \;\; \forall \underline{x}, \underline{y} \in X$

If in addition

→ $d(\underline{x}, \underline{y}) = d_0 \;\; if \;\; and \;\; only \;\; if \;\; \underline{x} = \underline{y}$

→ $d(\underline{x}, \underline{z}) \leq d(\underline{x}, \underline{y}) + d(\underline{y}, \underline{z}), \;\; \forall \underline{x}, \underline{y}, \underline{z} \in X$

(triangular inequality)

d is called a metric dissimilarity measure.

⇨ Similarity measure (between vectors of X) is a function

$$s: \; X \times X \longrightarrow \Re$$

with the following properties

$$\rightarrow \exists s_0 \in \Re: \; -\infty < s(\underline{x}, \underline{y}) \leq s_0 < +\infty, \;\; \forall \underline{x}, \underline{y} \in X$$

$$\rightarrow s(\underline{x}, \underline{x}) = s_0, \;\; \forall \underline{x} \in X$$

$$\rightarrow s(\underline{x}, \underline{y}) = s(\underline{y}, \underline{x}), \;\; \forall \underline{x}, \underline{y} \in X$$

If in addition

$\rightarrow s(\underline{x},\underline{y}) = s_0 \ \ if \ \ and \ \ only \ \ if \ \ \underline{x} = \underline{y}$

$\rightarrow s(\underline{x},\underline{y})s(\underline{y},\underline{z}) \leq [s(\underline{x},\underline{y}) + s(\underline{y},\underline{z})]s(\underline{x},\underline{z}), \ \ \forall \underline{x},\underline{y},\underline{z} \in X$

s is called a metric similarity measure.

➔ *Between sets*

Let $D_i \subset X$, i=1,…,k and U=$\{D_1,\ldots,D_k\}$

A proximity measure $\wp$ on U is a function

$$\wp : U \times U \longrightarrow \Re$$

A dissimilarity measure has to satisfy the relations of dissimilarity measure between vectors, where D_i's are used in place of $\underline{x}$, $\underline{y}$ (similarly for similarity measures).

Proximity measures between vectors

➔ Real-valued vectors

⇨ Dissimilarity measures (DMs)

➔ *Weighted l_p metric DMs*

$$d_p(\underline{x},\underline{y}) = \left(\sum_{i=1}^{l} w_i \mid x_i - y_i \mid^p\right)^{1/p}$$

Interesting instances are obtained for

» p=1 (*weighted Manhattan* norm)

» p=2 (***weighted Euclidean* norm**)

» $p=\infty$ ($d_\infty(\underline{x},\underline{y})=\max_{1\le i\le l} w_i|x_i-y_i|$)

→*Other measures*

» $$d_G(\underline{x},\underline{y}) = -\log_{10}\left(1-\frac{1}{l}\sum_{j=1}^{l}\frac{|x_j-y_j|}{b_j-a_j}\right)$$

where b_j and a_j are the maximum and the minimum values of the j-th feature, among the vectors of X (dependence on the current data set)

» $$d_Q(\underline{x},\underline{y}) = \sqrt{\frac{1}{l}\sum_{j=1}^{l}\left(\frac{x_j-y_j}{x_j+y_j}\right)^2}$$

⇨ Similarity measures

→*Inner product*

$$s_{inner}(\underline{x},\underline{y}) = \underline{x}^T \underline{y} = \sum_{i=1}^{l} x_i y_i$$

→*Tanimoto measure*

$$s_T(\underline{x},\underline{y}) = \frac{\underline{x}^T \underline{y}}{\| \underline{x} \|^2 + \| \underline{y} \|^2 - \underline{x}^T \underline{y}}$$

$$s_T(\underline{x},\underline{y}) = 1 - \frac{d_2(\underline{x},\underline{y})}{\| \underline{x} \| + \| \underline{y} \|}$$

➔Discrete-valued vectors

⇨ Let $F=\{0,1,\ldots,k\text{-}1\}$ be a set of symbols and $X=\{\underline{x}_1,\ldots,\underline{x}_N\} \subset F^l$

⇨ Let $A(\underline{x},\underline{y})=[a_{ij}]$, $i, j=0,1,\ldots,k\text{-}1$, where a_{ij} is the number of places where $\underline{x}$ has the i-th symbol and $\underline{y}$ has the j-th symbol.

$$\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} a_{ij} = l$$

Several proximity measures can be expressed as combinations of the elements of $A(\underline{x},\underline{y})$.

⇨ Dissimilarity measures:

➔The Hamming distance (number of places where $\underline{x}$ and $\underline{y}$ differ)

$$d_H(\underline{x},\underline{y}) = \sum_{i=0}^{k-1}\sum_{\substack{j=0 \\ j\neq i}}^{k-1} a_{ij}$$

➔The l_1 distance $\quad d_1(\underline{x},\underline{y}) = \sum_{i=1}^{l} | x_i - y_i |$

⇨ Similarity measures:

→Tanimoto measure : $$s_T(\underline{x},\underline{y}) = \frac{\sum_{i=1}^{k-1} a_{ii}}{n_x + n_y - \sum_{i=1}^{k-1}\sum_{j=1}^{k-1} a_{ij}}$$

where $$n_x = \sum_{i=1}^{k-1}\sum_{j=0}^{k-1} a_{ij}, \quad n_y = \sum_{i=0}^{k-1}\sum_{j=1}^{k-1} a_{ij},$$

→Measures that exclude a_{00}: $$\sum_{i=1}^{k-1} a_{ii} / l \qquad \sum_{i=1}^{k-1} a_{ii} /(l - a_{00})$$

→Measures that include a_{00}: $$\sum_{i=0}^{k-1} a_{ii} / l$$

➔Mixed-valued vectors

Some of the coordinates of the vectors $\underline{x}$ are real and the rest are discrete.

Methods for measuring the proximity between two such $\underline{x}_i$ and $\underline{x}_j$:

⇨ Adopt a proximity measure (PM) suitable for real-valued vectors.

⇨ Convert the real-valued features to discrete ones and employ a discrete PM.

The more general case of mixed-valued vectors:

⇨ Here nominal, ordinal, interval-scaled, ratio-scaled features are treated separately.

Proximity between a vector and a set

➔ **Let** $X=\{\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_N\}$ **and** $C \subset X$, $\underline{x} \in X$

➔ ***All points of*** C ***contribute to the definition of*** $\wp(\underline{x}, C)$

⇨ Max proximity function

$$\wp^{ps}_{\max}(\underline{x}, C) = \max_{\underline{y}\in C} \wp(\underline{x}, \underline{y})$$

⇨ Min proximity function

$$\wp^{ps}_{\min}(\underline{x}, C) = \min_{\underline{y}\in C} \wp(\underline{x}, \underline{y})$$

⇨ Average proximity function

$$\wp^{ps}_{avg}(\underline{x}, C) = \frac{1}{n_C} \sum_{\underline{y}\in C} \wp(\underline{x}, \underline{y})$$ (n_C is the cardinality of C)

➔ ***A representative(s) of C, r_C, contributes to the definition of $\wp(\underline{x}, C)$***

In this case: $\wp(\underline{x}, C) = \wp(\underline{x}, \underline{r}_C)$,

Typical representatives are:

⇨ The mean vector:

$$\underline{m}_p = \left(1/n_C \right) \sum_{\underline{y} \in C} \underline{y}$$

where n_C is the cardinality of C

⇨ The mean center:

$$\underline{m}_C \in C: \ \sum_{\underline{y} \in C} d(\underline{m}_C, \underline{y}) \leq \sum_{\underline{y} \in C} d(\underline{z}, \underline{y}), \ \forall \underline{z} \in C$$

⇨ The median center:

$$\underline{m}_{med} \in C: \ med(d(\underline{m}_{med}, \underline{y}) \mid \underline{y} \in C) \leq med(d(\underline{z}, \underline{y}) \mid \underline{y} \in C), \ \forall \underline{z} \in C$$

Other representatives (e.g., hyperplanes, hyperspheres) are useful in certain applications (e.g., object identification using clustering techniques).

Proximity between sets

➔ **Let $X=\{\underline{x}_1,\ldots,\underline{x}_N\}$, D_i, $D_j \subset X$ and $n_i=|D_i|$, $n_j=|D_j|$**

➔ **All points of each set contribute to $\wp(D_i,D_j)$**

⇨ Max proximity

$$\wp^{ss}_{\max}(D_i,D_j)=\max_{\underline{x}\in D_i,\underline{y}\in D_j}\wp(\underline{x},\underline{y})$$

⇨ Min proximity

$$\wp^{ss}_{\min}(D_i,D_j)=\min_{\underline{x}\in D_i,\underline{y}\in D_j}\wp(\underline{x},\underline{y})$$

⇨ Average proximity

$$\wp^{ss}_{avg}(D_i,D_j)=\left(1/n_i n_j\right)\sum_{x\in D_i}\sum_{x\in D_j}\wp(\underline{x},\underline{y})$$

➢ Remarks:

- Different choices of proximity functions between sets may lead to totally different clustering results.
- Different proximity measures between vectors in the same proximity function between sets may lead to totally different clustering results.
- The only way to achieve a proper clustering is
  - by trial and error and,
  - taking into account the opinion of an expert in the field of application.

Clustering Algorithms

Clustering Algorithms

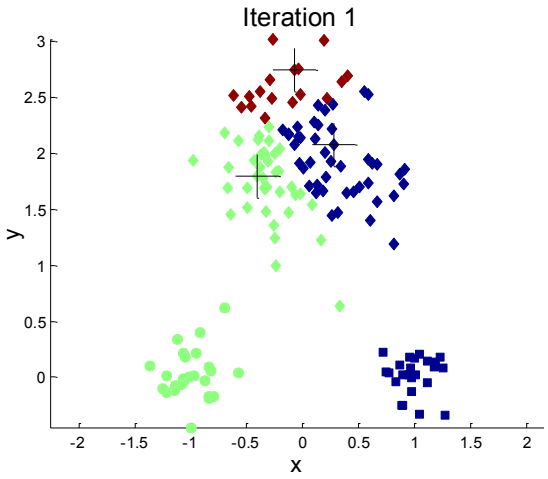
- K-means
- Hierarchical clustering
- Graph based clustering

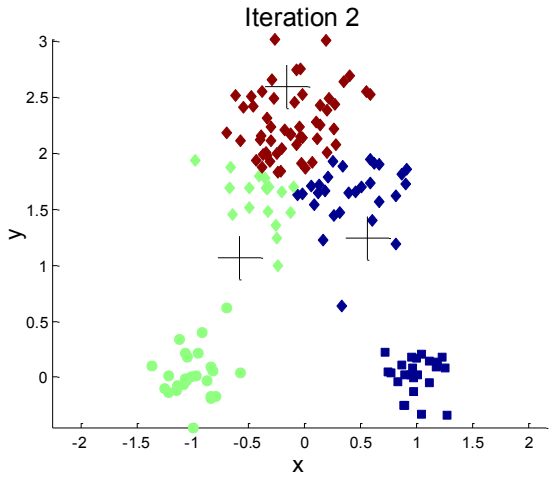
K-means Clustering

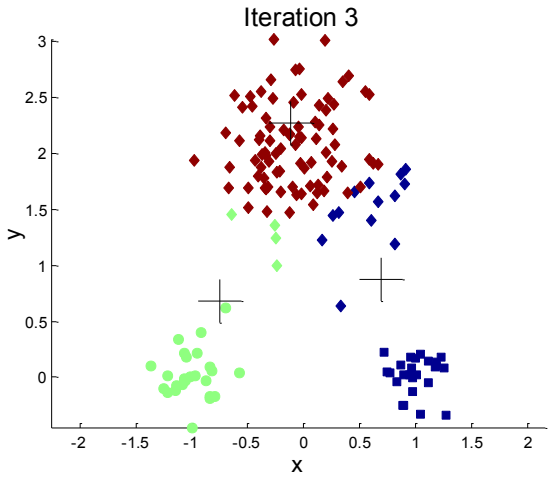
- **Partitional clustering approach**
- **Each cluster is associated with a centroid (center point)**
- **Each point is assigned to the cluster with the closest centroid**
- **Number of clusters, K, must be specified**
- **The basic algorithm is very simple**

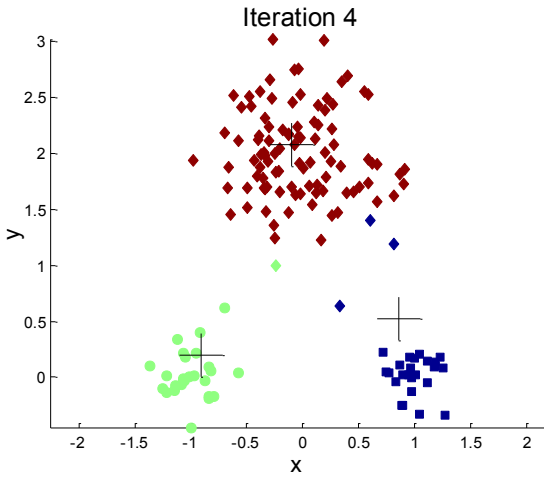
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

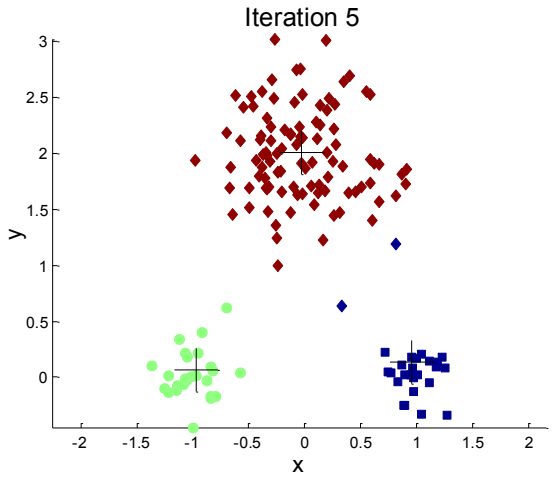
Illustration

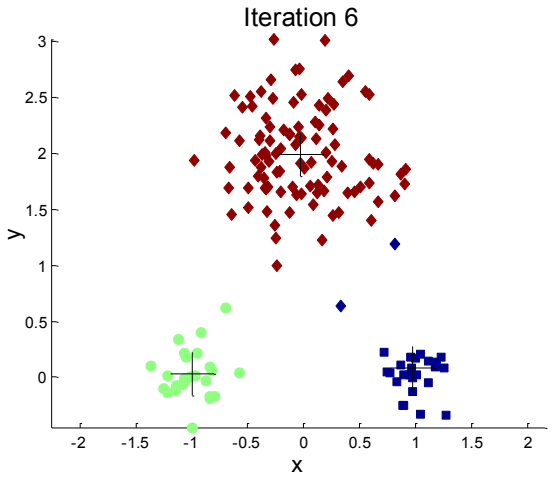








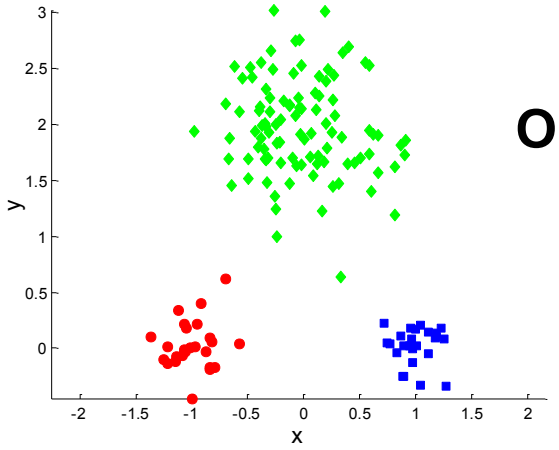




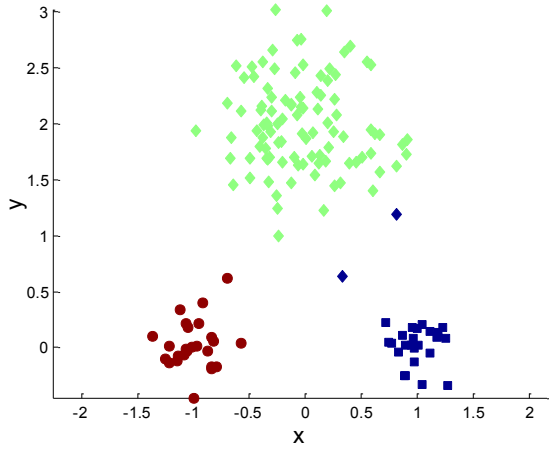
K-means Clustering – Details

- **Initial centroids are often chosen randomly.**
  - Clusters produced vary from one run to another.
- **The centroid is (typically) the mean of the points in the cluster.**
- **'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.**
- **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 = number of attributes

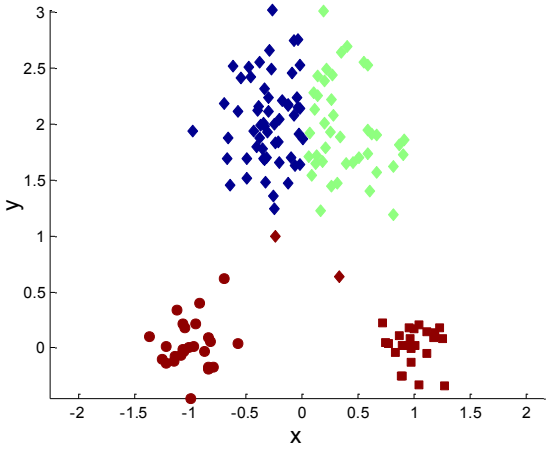
Two different K-means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

Problems with Selecting Initial Points

➔ **If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.**

- ⇨ Chance is relatively small when K is large
- ⇨ If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

- ⇨ For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
- ⇨ Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
- ⇨ Consider an example of five pairs of clusters

Solutions to Initial Centroids Problem

- **Multiple runs**
  - ⇨ Helps, but probability is not on your side
- **Sample and use hierarchical clustering to determine initial centroids**
- **Select more than k initial centroids and then select among these initial centroids**
  - ⇨ Select most widely separated
- **Bisecting K-means**
  - ⇨ Not as susceptible to initialization issues

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 representative point for cluster C_i
  - → can show that m_i corresponds to the center (mean) of the cluster
- ⇨ Given two clusters, we can choose the one with the smaller 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

Limitations of K-means

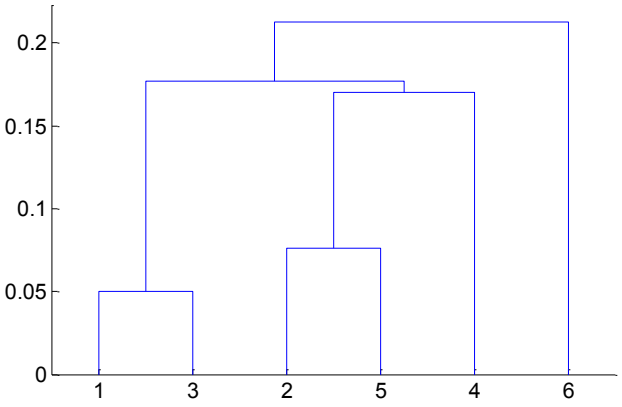
- **K-means has problems when clusters are of differing**
  - ⇨ Sizes
  - ⇨ Densities
  - ⇨ Non-globular shapes
- **K-means has problems when the data contains outliers.**
- **The number of clusters (K) is difficult to determine.**

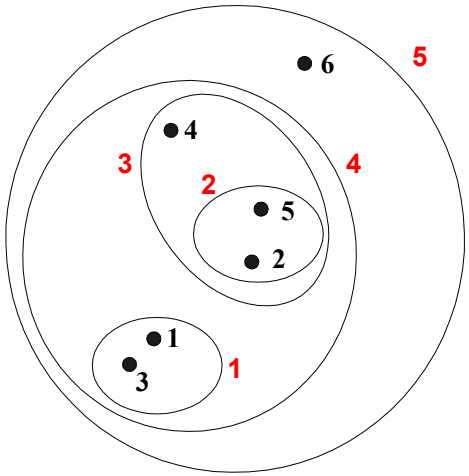
K Means in Python

```
from sklearn.cluster import KMeans
import numpy as np
X = np.array([[1, 2], [1, 4], [1, 0],
...              [4, 2], [4, 4], [4, 0]])
kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
kmeans.labels_
          array([0, 0, 0, 1, 1, 1], dtype=int32)
kmeans.predict([[0, 0], [4, 4]])
          array([0, 1], dtype=int32)
kmeans.cluster_centers_
          array([[ 1.,  2.], [ 4.,  2.]])
```

Hierarchical Clustering

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





Strengths of Hierarchical Clustering

- **Do not have to assume any particular number of clusters**
  - ⇨ Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level

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

Hierarchical Clustering

➔Two main types of hierarchical clustering

- ⇨Agglomerative:
  - → Start with the points as individual clusters
  - → At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- ⇨Divisive:
  - → Start with one, all-inclusive cluster
  - → At each step, split a cluster until each cluster contains a point (or there are k clusters)

➔Traditional hierarchical algorithms use a similarity or distance matrix

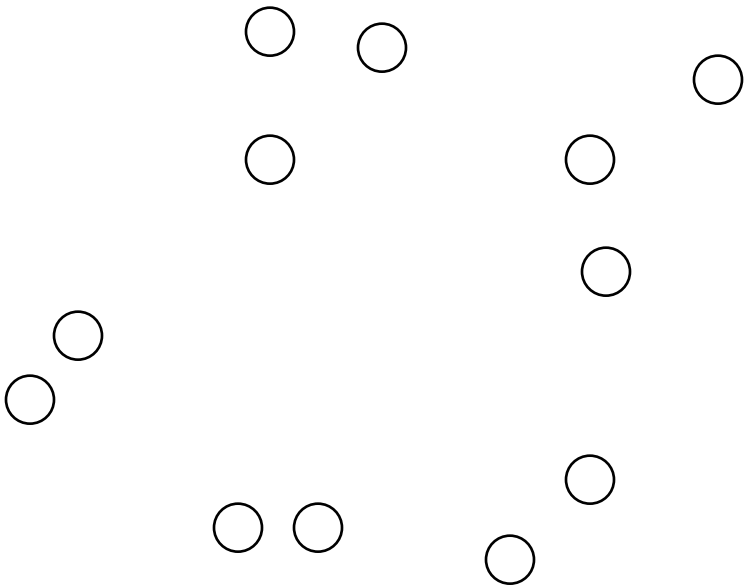
- ⇨ Merge or split one cluster at a time

Agglomerative Clustering Algorithm

- **More popular hierarchical clustering technique**
- **Basic algorithm is straightforward**
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  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

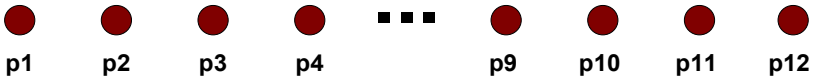
Starting Situation

➔ **Start with clusters of individual points and a proximity matrix**



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

Proximity Matrix

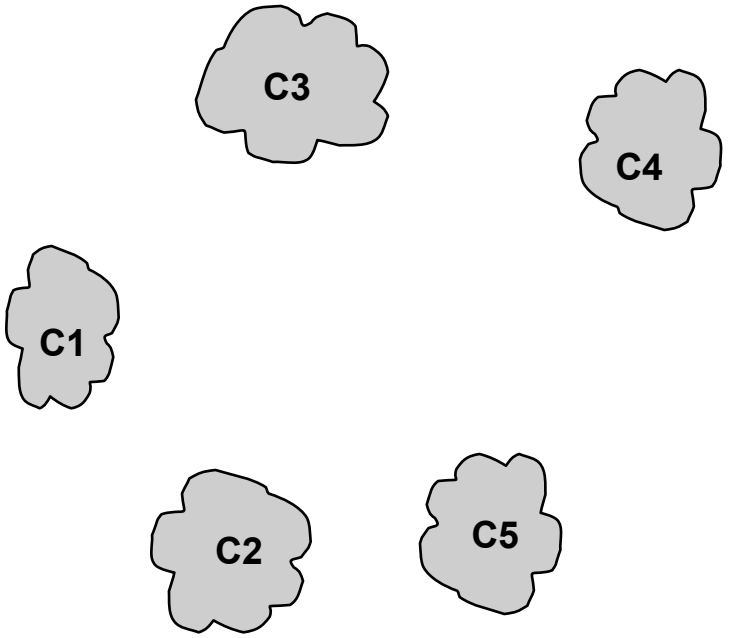


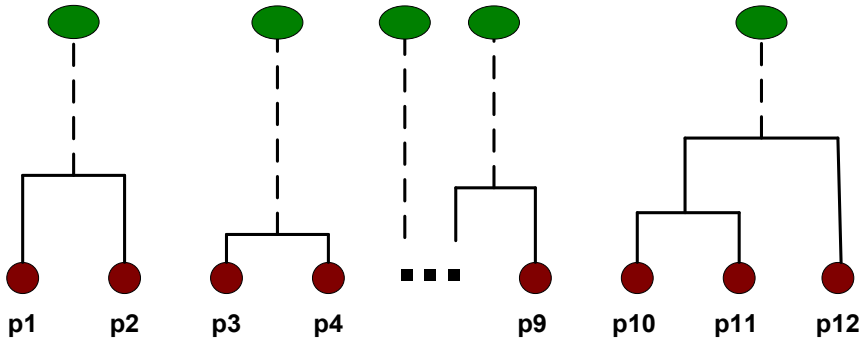
Intermediate Situation

➔ **After some merging steps, we have some clusters**

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix

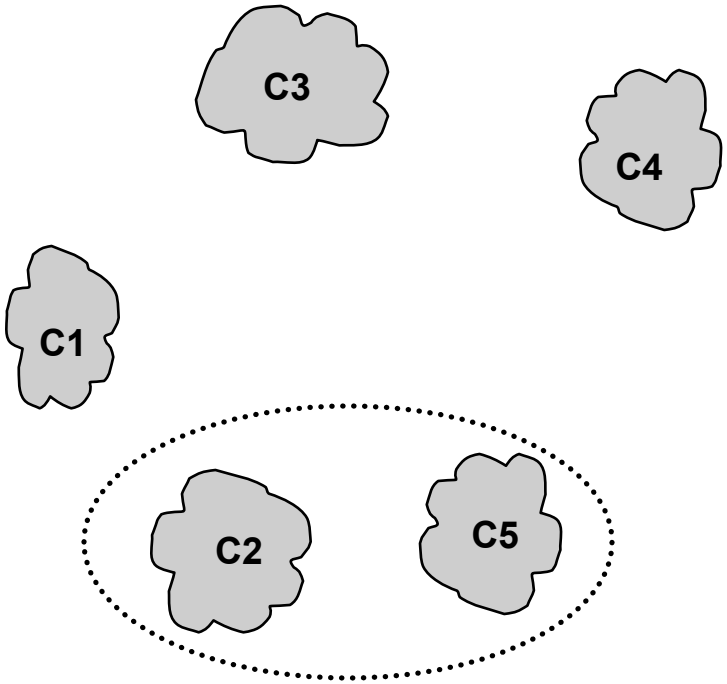




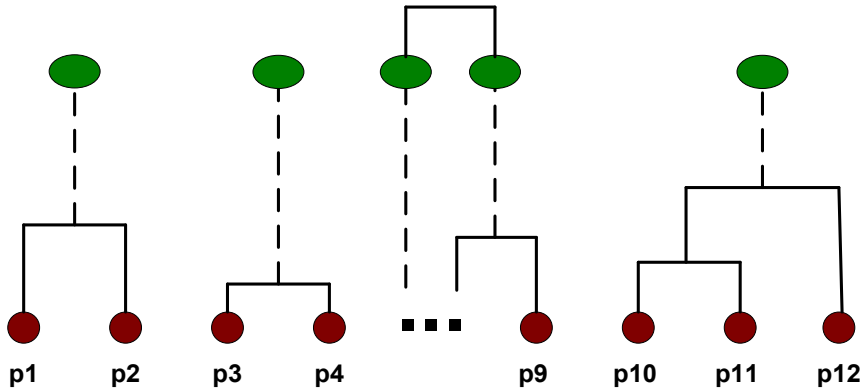
Intermediate Situation

➔ **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					

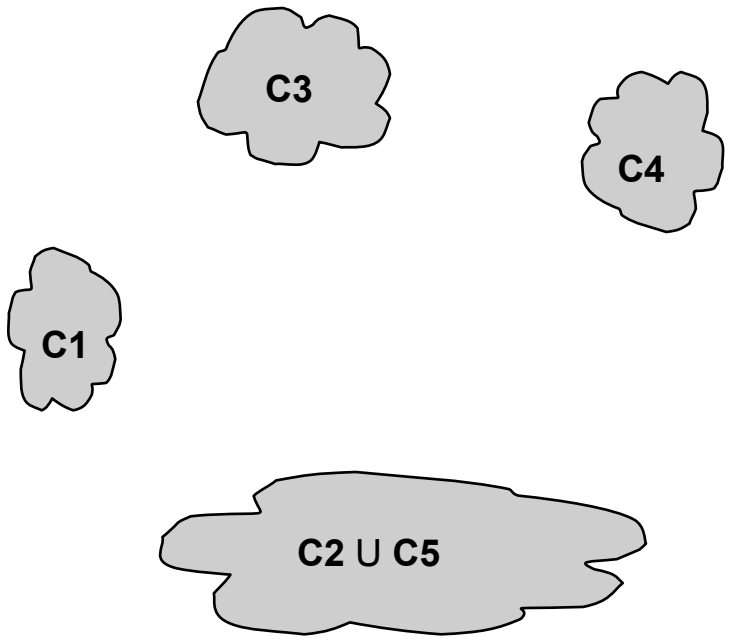


Proximity Matrix



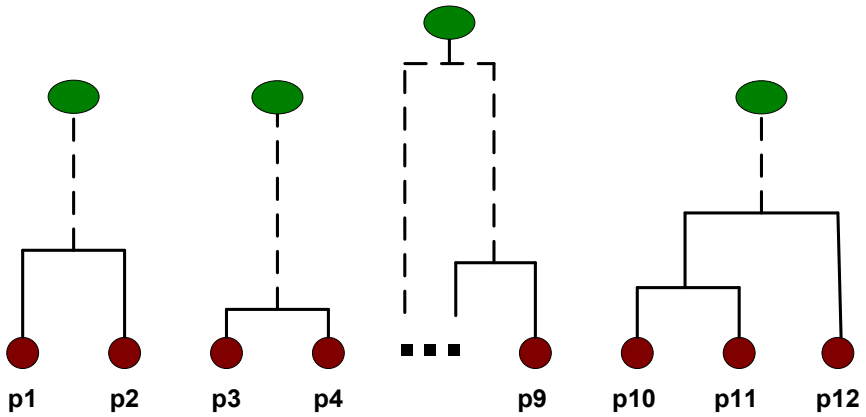
After Merging

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

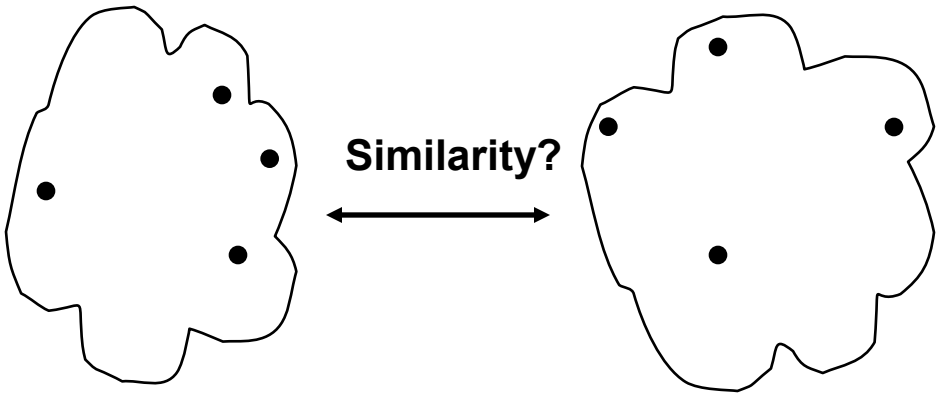


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity



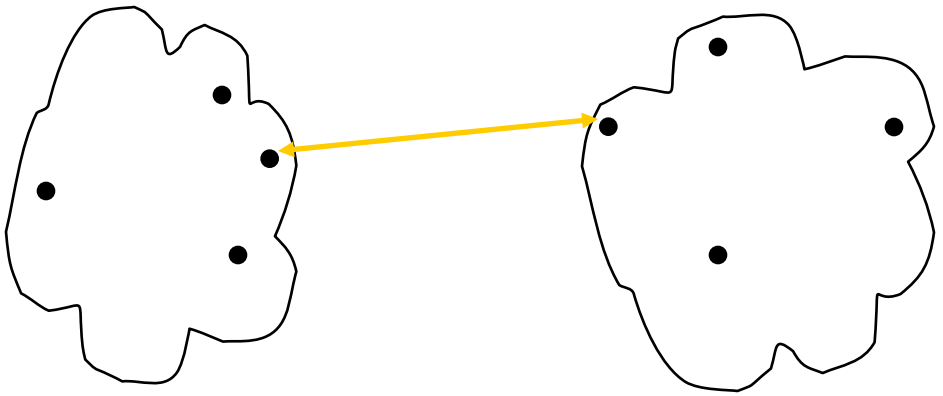
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

. .

. **Proximity Matrix**

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity

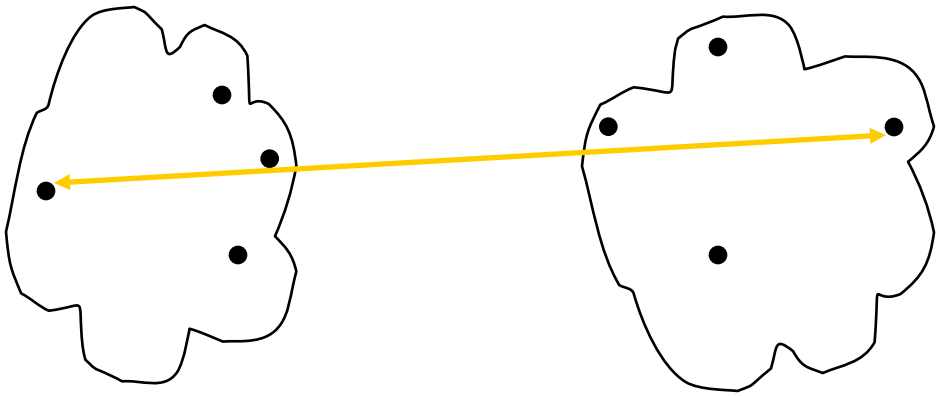


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity

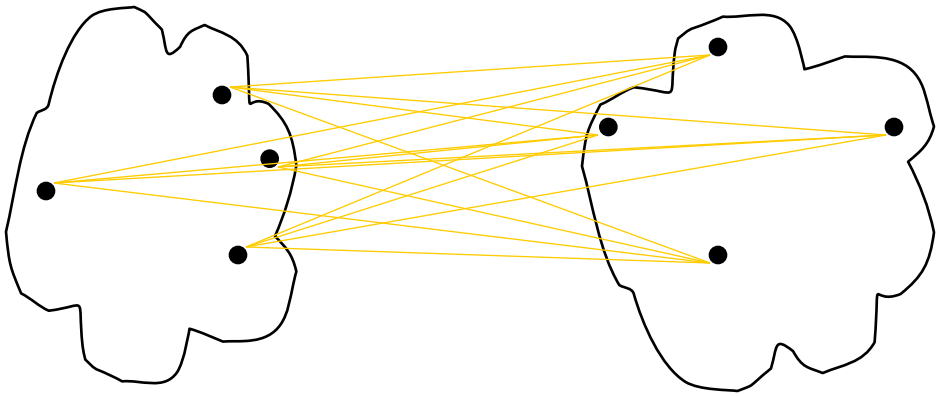


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity



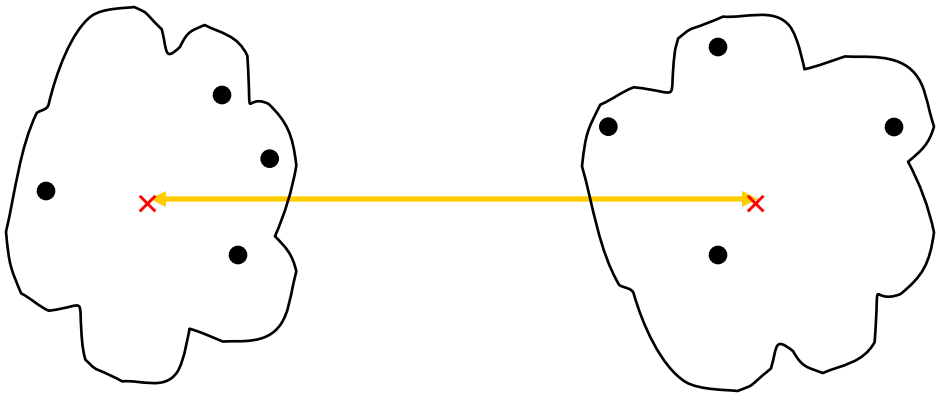
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

. .

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

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

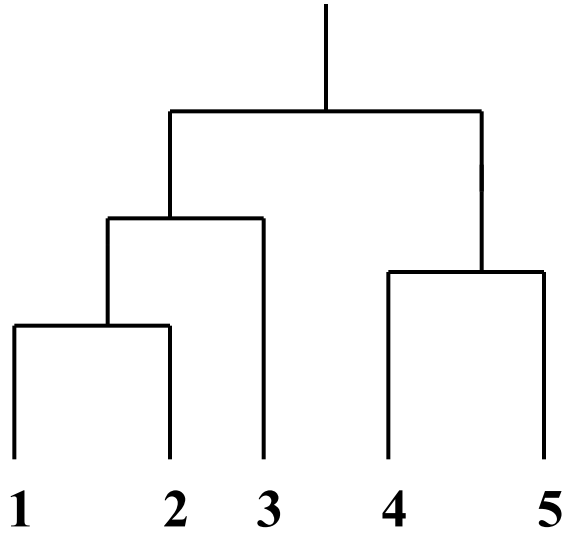
Proximity Matrix

Cluster Similarity: MIN (Single Link)

➔ **Similarity of two clusters is based on the two most similar (closest) points in the different clusters**

⇨ Determined by one pair of points, i.e., by one link in the proximity graph.

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

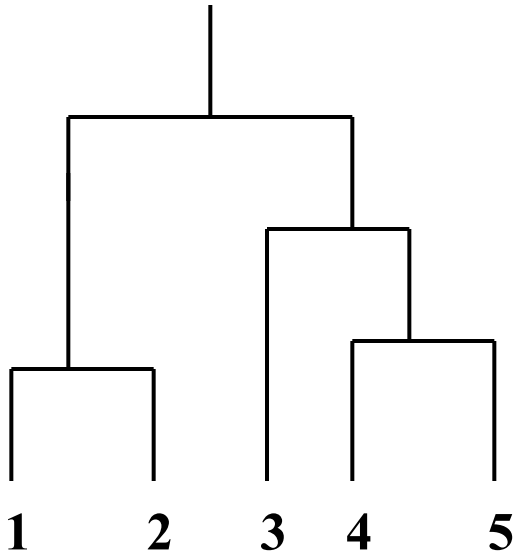


Cluster Similarity: MAX (Complete Linkage)

➔ **Similarity of two clusters is based on the two least similar (most distant) points in the different clusters**

⇨ Determined by all pairs of points in the two clusters

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



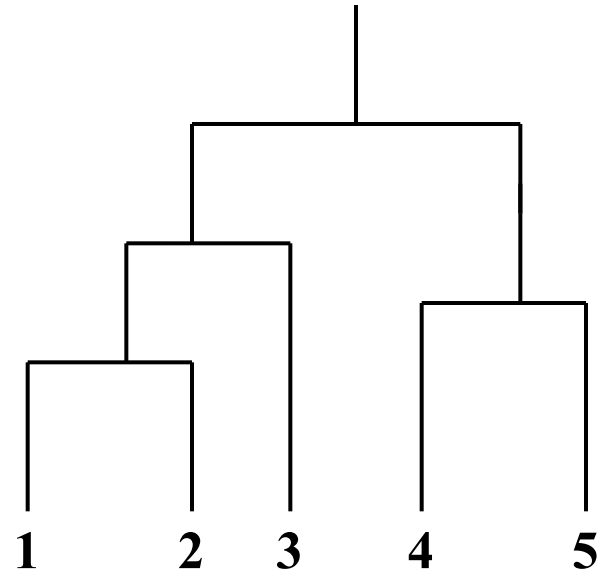
Cluster Similarity: Group Average

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

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

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

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



Hierarchical Clustering: Group Average

➔ **Compromise between Single and Complete Link**

➔ **Strengths**

⇨ Less susceptible to noise and outliers

➔ **Limitations**

⇨ Biased towards globular clusters

Hierarchical Clustering: 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

Hierarchical Clustering: Problems and Limitations

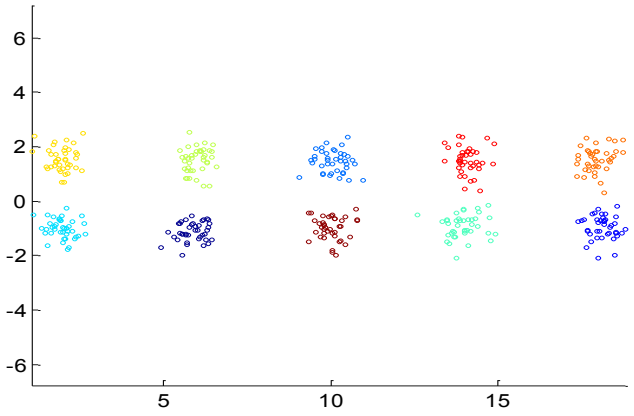
- **Once a decision is made to combine two clusters, it cannot be undone**
- **No objective function is directly minimized**
- **Different schemes have problems with one or more of the following:**
  - Sensitivity to noise and outliers (MIN)
  - Difficulty handling different sized clusters and non-convex shapes (Group average, MAX)
  - Breaking large clusters (MAX)

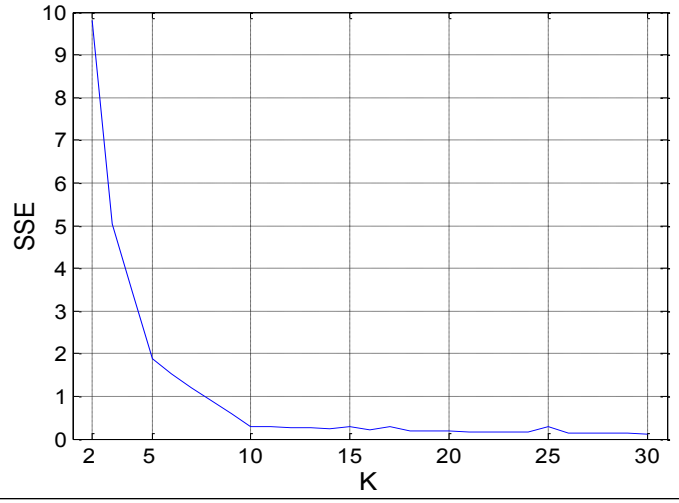
Measures of Cluster Validity

- **Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.**
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- **Sometimes these are referred to as criteria instead of indices**
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Internal Measures: SSE

- ➔ Clusters in complicated figures aren't well separated
- ➔ Internal Index: Used to measure the goodness of a clustering structure without respect to external information
- ➔ SSE is good for comparing two clusterings or two clusters (average SSE).
- ➔ Can also be used to estimate the number of clusters.





Internal Measures: Cohesion and Separation

- **Cluster Cohesion: Measures how closely related are objects in a cluster**
  - Example: SSE
- **Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters**
- **Example: Squared Error**
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

  - Separation is measured by the between cluster sum of squares

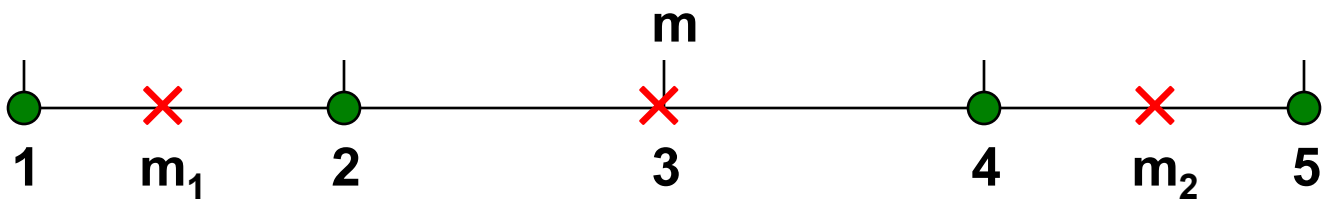
$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

➔ Example: SSE

⇨ BSS + WSS = constant



K=1 cluster:

$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

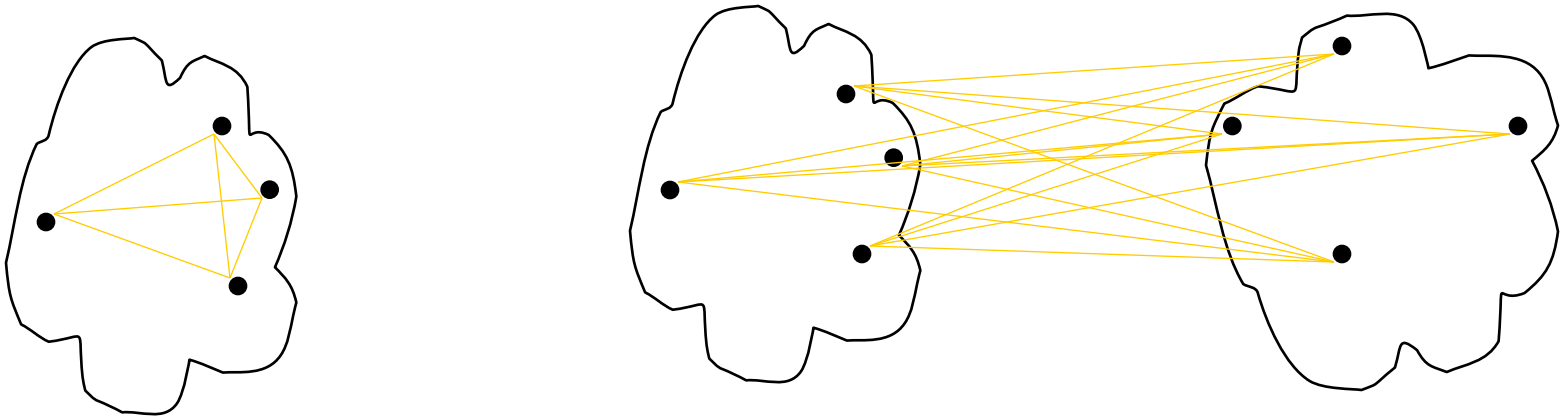
$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation

➔ **A proximity graph based approach can also be used for cohesion and separation.**

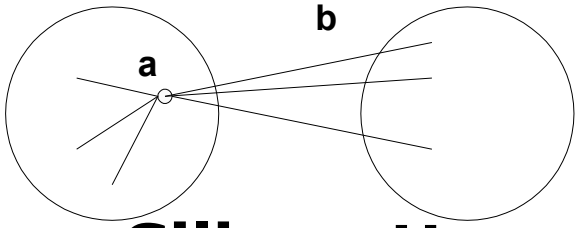
- ⇨ Cluster cohesion is the sum of the weight of all links within a cluster.
- ⇨ Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion separation

Internal Measures: Silhouette Coefficient

- **Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings**
- **For an individual point, *i***
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by
    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)
  - Typically between 0 and 1.
  - The closer to 1 the better.



- **Can calculate the Average Silhouette width for a cluster or a clustering**

External Measures of Cluster Validity: Entropy and Purity

Table K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity

"*The validation of clustering structures is the most difficult and frustrating part of cluster analysis.*

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

***Algorithms for Clustering Data*, Jain and Dubes**