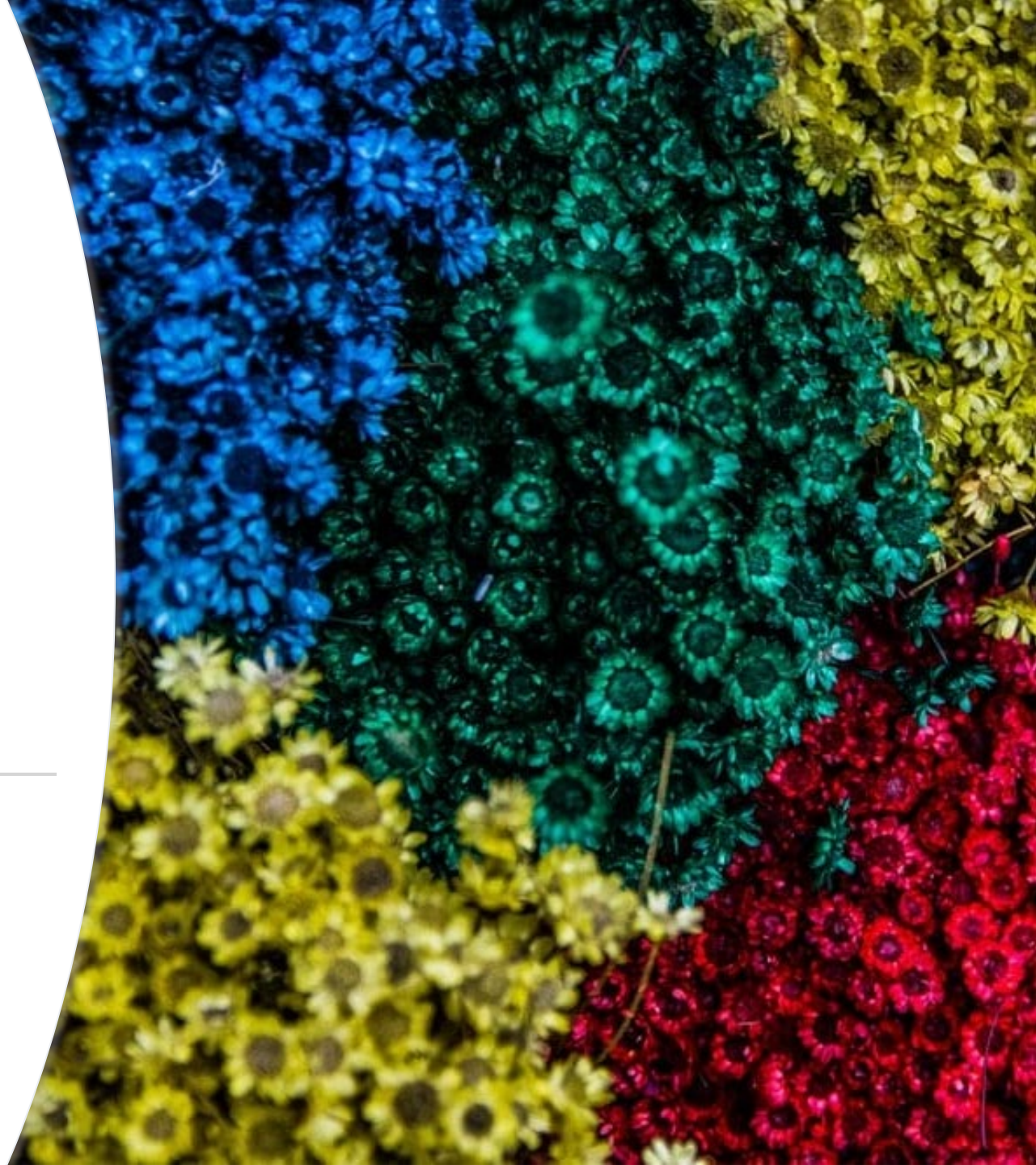# Introduction to Data Mining

Chapter 7
Cluster Analysis

by Michael Hahsler

Based in Slides by by Tan, Steinbach, Karpatne, Kumar

R Code Examples

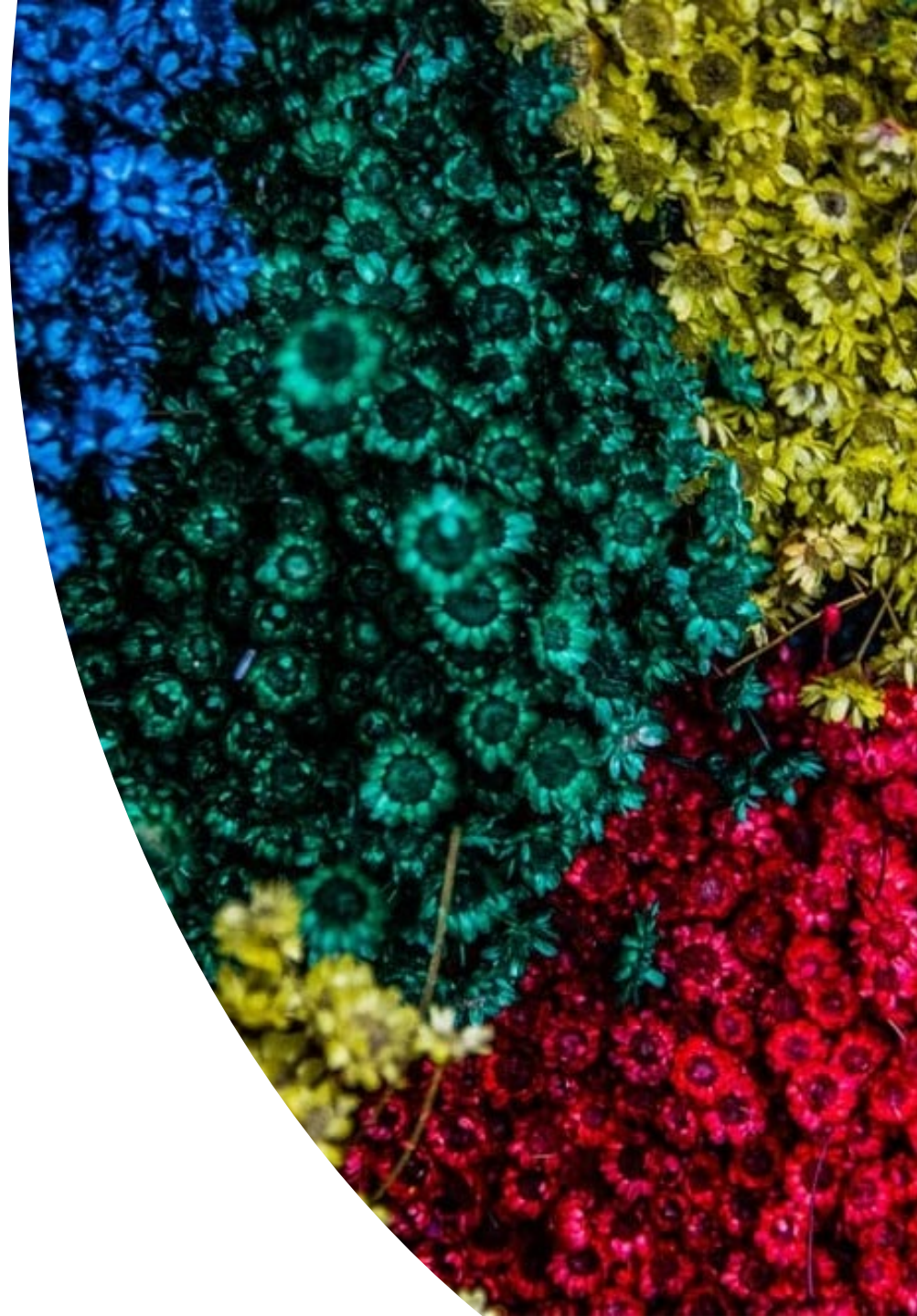- Available R Code examples are indicated on slides by the R logo

- The Examples are available at
  https://mhahsler.github.io/Introduction_to_Data_Mining_R_Examples/

Topics

- **Introduction**
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Validation

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.

Intra-cluster distances are minimized

Inter-cluster distances are maximized
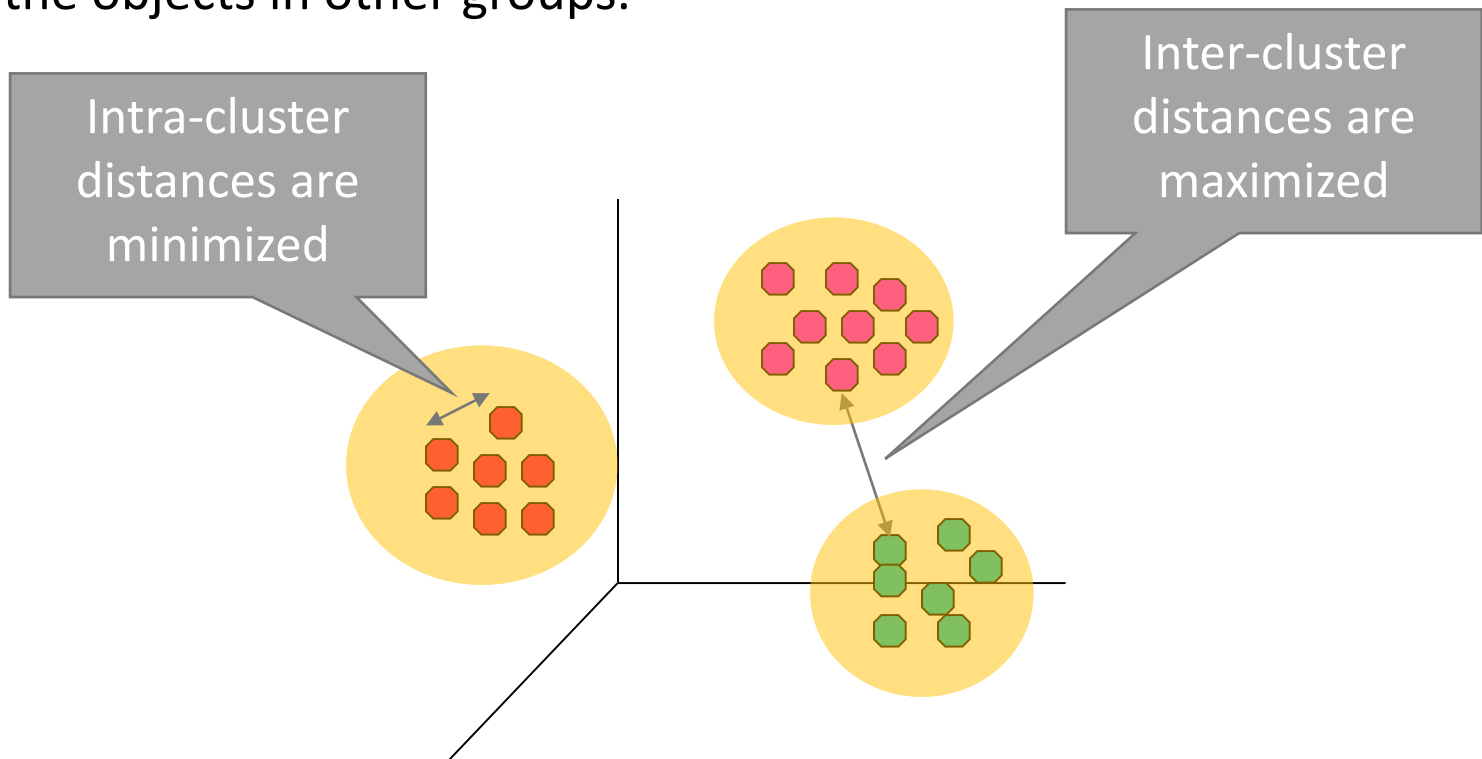
- A clustering is a set of clusters and each cluster contains a set of points.

Applications of Cluster Analysis

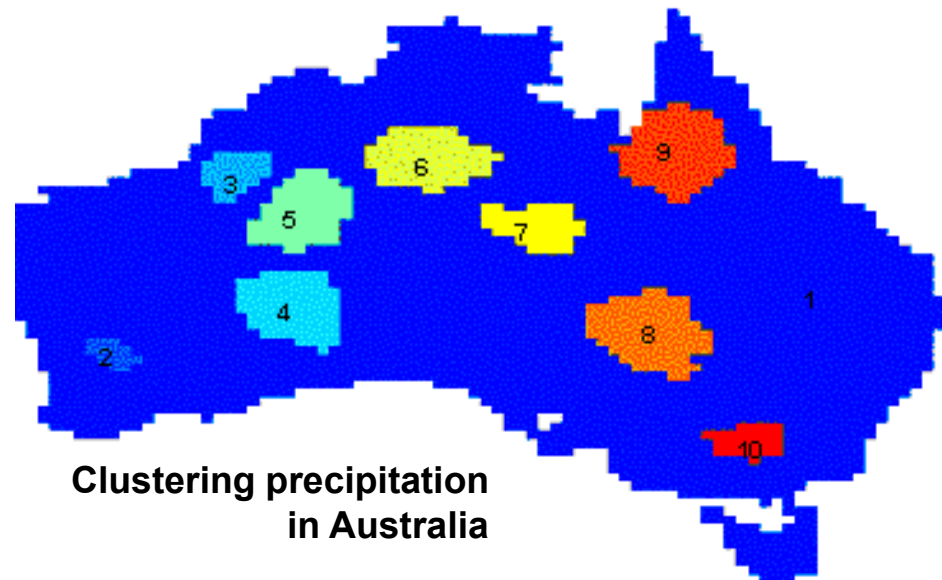|	Discovered Clusters	Industry Group
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- 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

Clustering precipitation in Australia

What is not Cluster Analysis?

- Supervised classification
  - Uses class label information

- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name

- Results of a query
  - Groupings are a result of an external specification

→ **Clustering uses only the data**

Similarity

- How do we measure
  similarity/proximity/dissimilarity/distance?

- Examples
  - Minkovsky distance: Manhattan distance, Euclidean Distance, etc.
  - Jaccard index for binary data
  - Gower's distance for mixed data (ratio/interval and nominal)
  - Correlation coefficient as similarity between variables
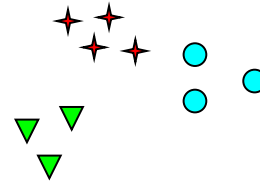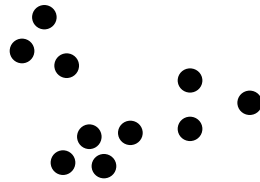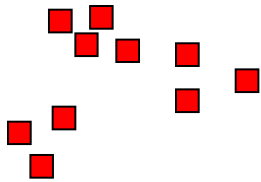
Notion of a Cluster can be Ambiguous
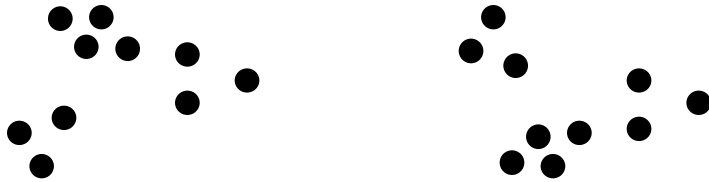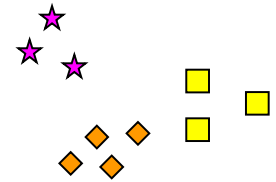
How many clusters?

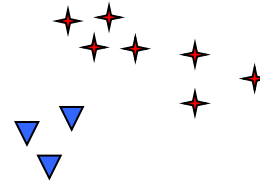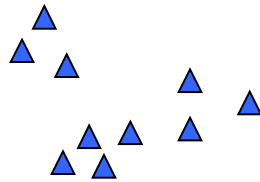# Notion of a Cluster can be Ambiguous
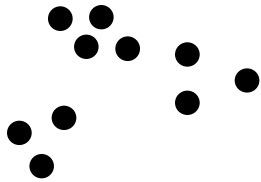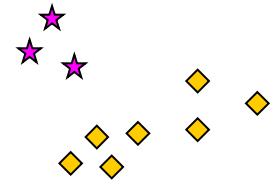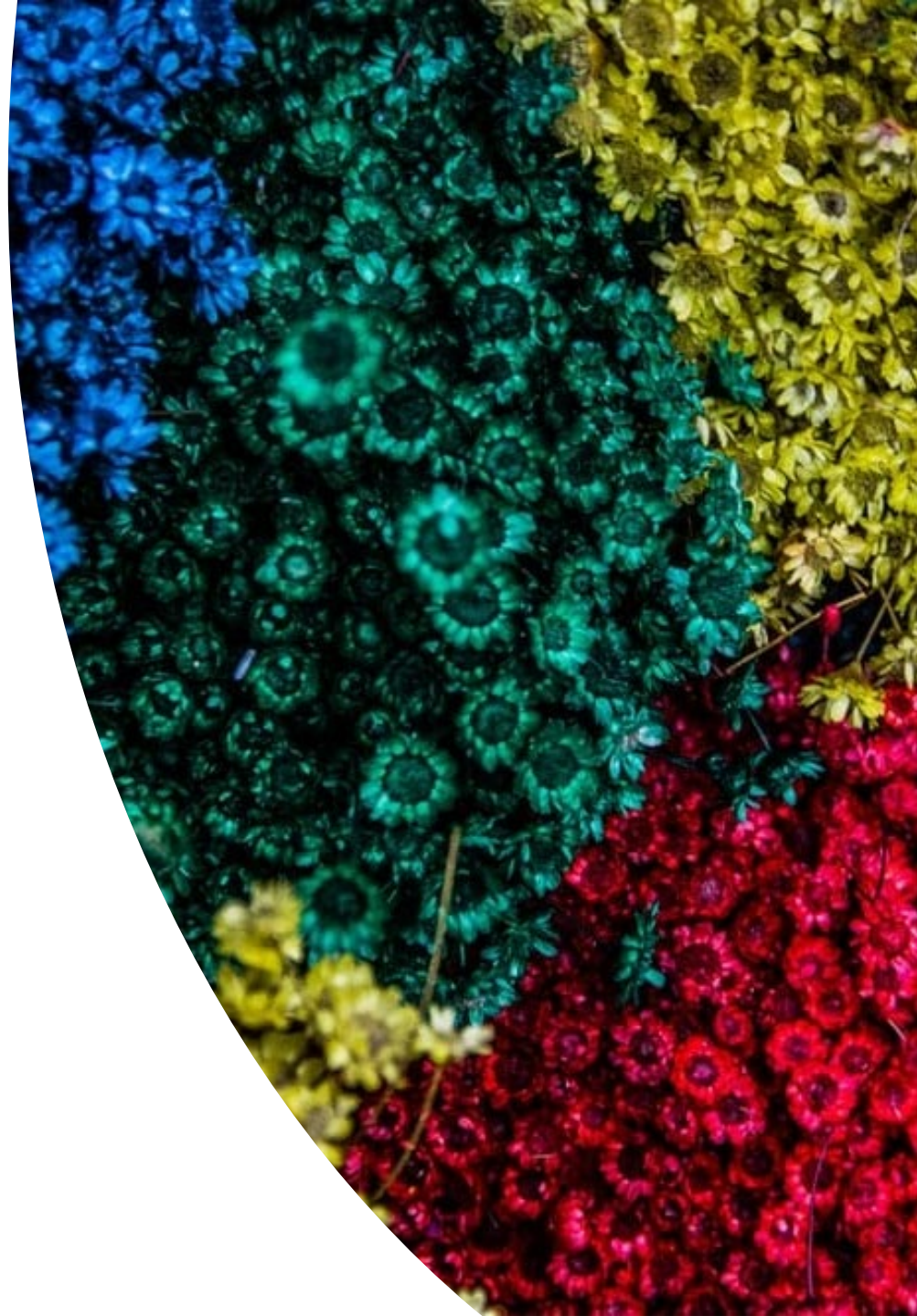

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Topics

- Introduction
- **Types of Clustering**
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
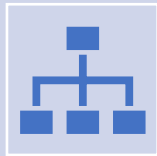 - Density-based Clustering
- Cluster Validation

Types of Clusterings

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
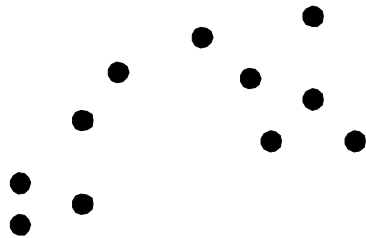
Original Points
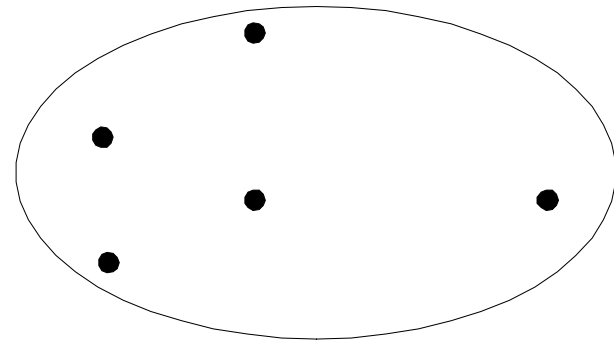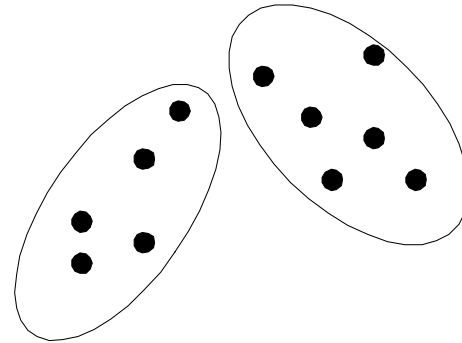
A Partitional Clustering

Hierarchical Clustering
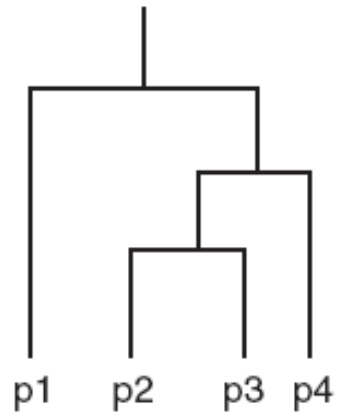
(a) Dendrogram.
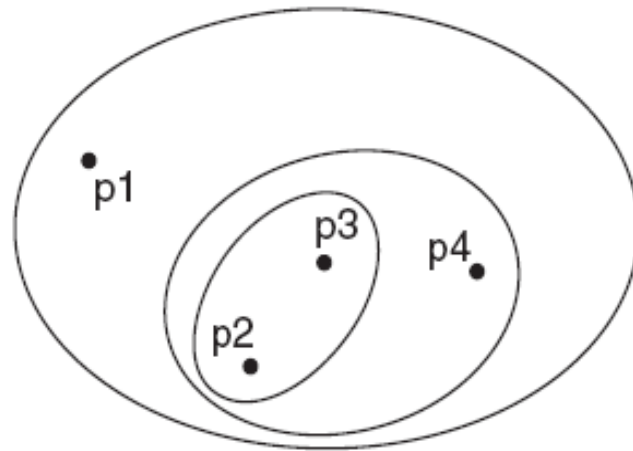
(b) Nested cluster diagram.

Figure 8.13. A hierarchical clustering of four points shown as a dendrogram and as nested clusters.

Other Distinctions Between Sets of Clusters

Exclusive versus non-exclusive

In non-exclusive clusterings, points may belong to multiple clusters.

Fuzzy versus non-fuzzy

In fuzzy clustering, a point belongs to every cluster with some membership weight between 0 and 1

Membership 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

Heterogeneous versus homogeneous

Cluster of widely different sizes, shapes, and densities

Topics

- Introduction
- Types of Clustering
- **Types of Clusters**
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Validation

Types of Clusters

Center-based Clusters

Well separated

Not well separated
(overlapping)
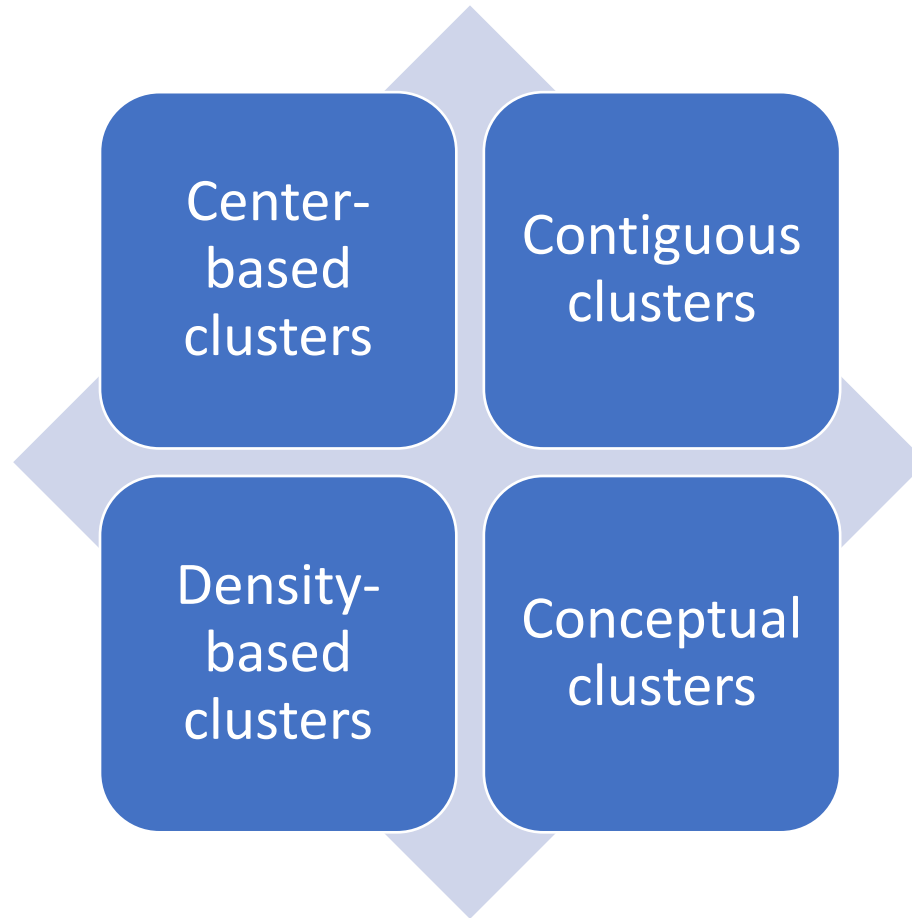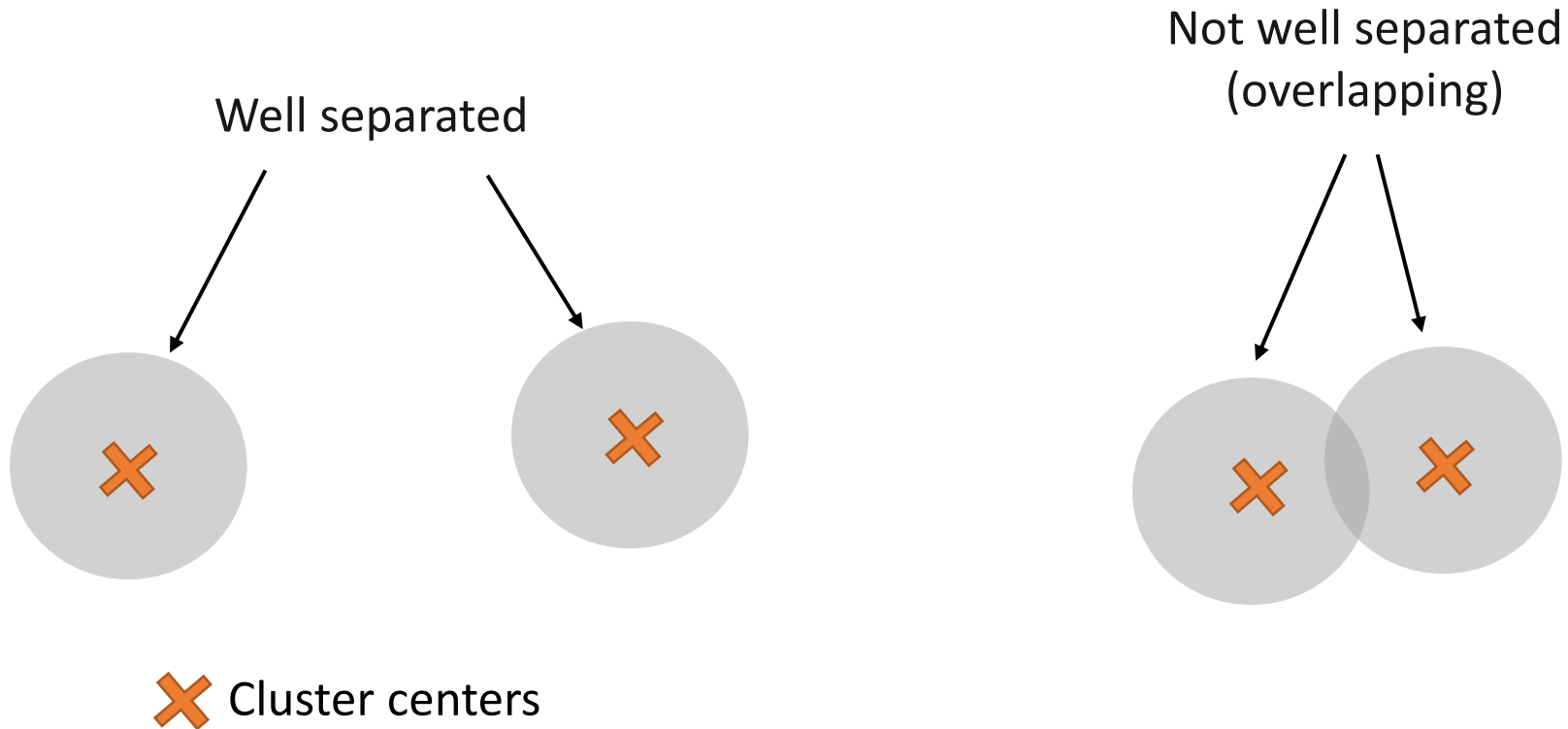
✖ Cluster centers

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

Contiguous and Density-based Clusters
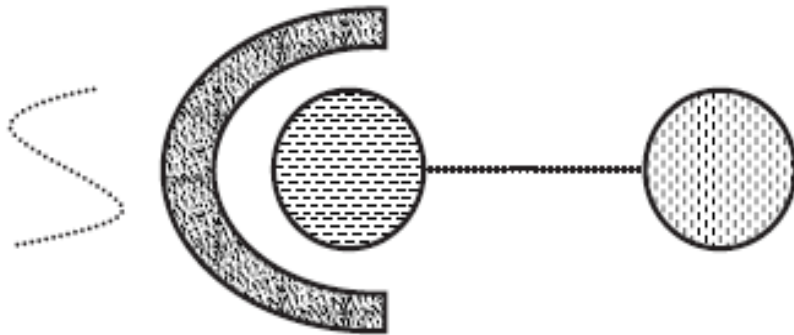
High density

(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.
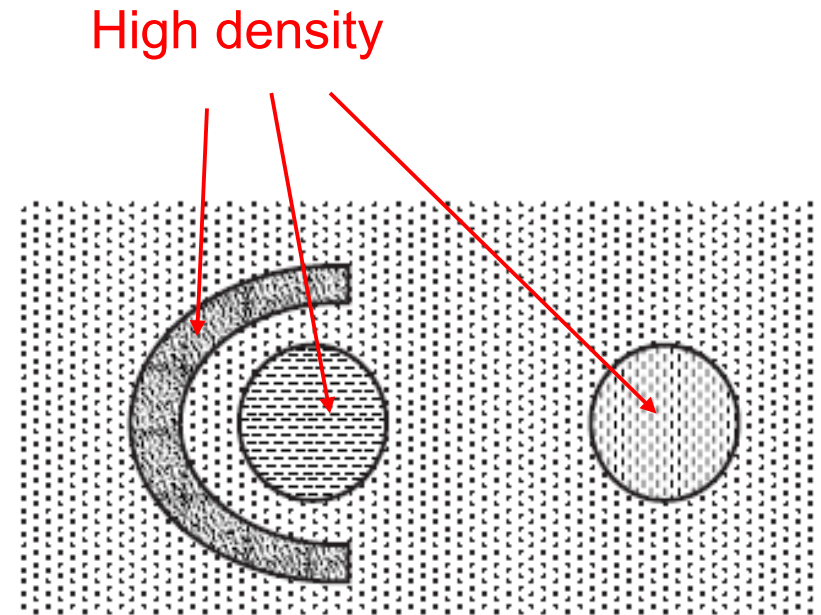
(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

Conceptual Clusters
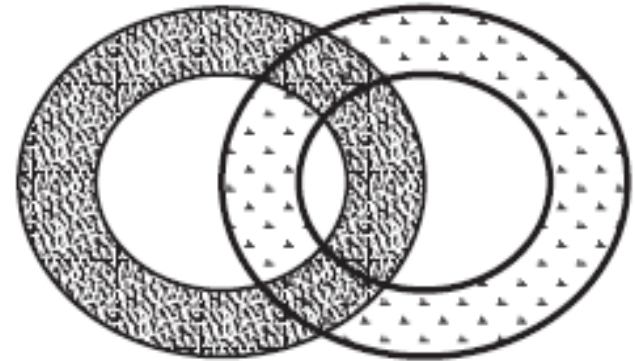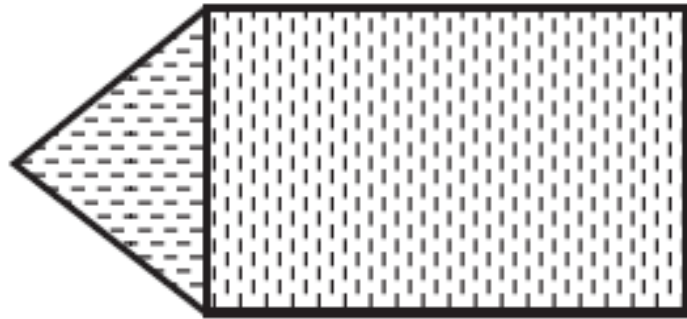
(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

Conceptual clusters are hard to detect since they are often not:
- Center-based
- Contiguity-based
- Density-based

Objective Functions

- The best clustering minimizes or maximizes an objective function.
- **Example**: Minimize the Sum of Squared Errors

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - m_i\|^2$$

x is a data point in cluster C_i, m_i is the center for cluster C_i as the mean of all points in the cluster and $\|\cdot\|$ is the L2 norm (= Euclidean distance).

- **Problem**: 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)

Objective Functions

Global objective function

- Typically used in **partitional clustering.** k-means uses SSE.
- **Mixture Models** assume that the data is a 'mixture' of a number of parametric statistical distributions (e.g., a mixture of Gaussians). Maximize log-likelihood of the model.

Local objective function

- **Hierarchical clustering** algorithms typically have local objectives.
- **Density-based clustering** is based on local density estimates.
- **Graph based approaches**. Graph partitioning (e.g., min-cut) and shared nearest neighbors.

We will talk about the objective functions when we talk about individual clustering algorithms.

Topics

- Introduction
- Types of Clustering
- Types of Clusters
- **Clustering Algorithms**
    - **K-Means Clustering**
    - Hierarchical Clustering
    - Density-based Clustering
- Cluster Validation

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

Lloyd's algorithm (Voronoi iteration):

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

K-means Clustering – Details

- **Initial centroids** are often chosen randomly.
    - —Clusters produced vary from one run to another.
- The centroid is the mean of the points in the cluster.
- 'Closeness' is measured by **Euclidean distance**
- K-means will converge (points stop changing assignment) typically in the first few iterations (<10).
    - —Sometimes 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

K-Means Example
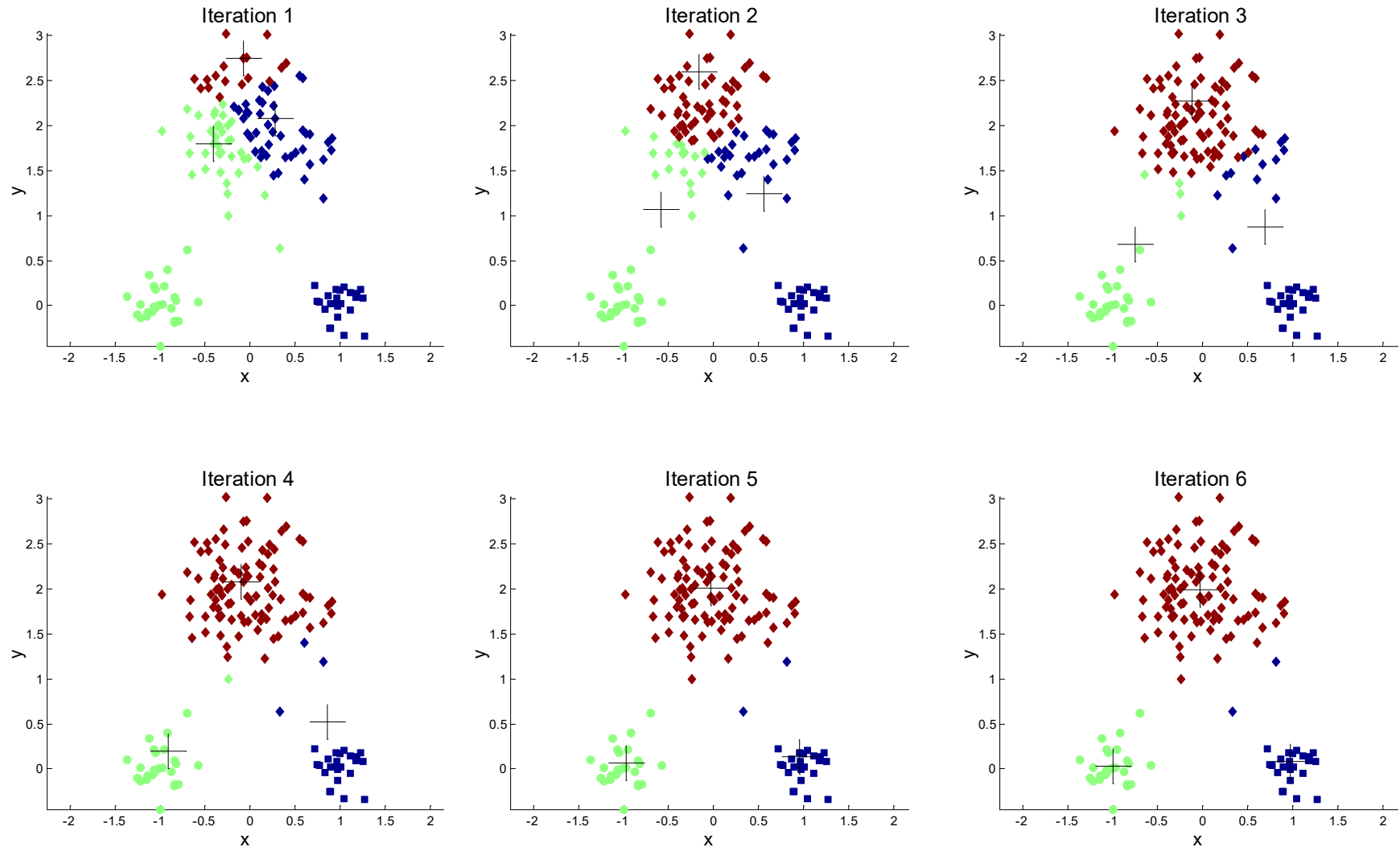
See visualization on course web site

Importance of Choosing Initial Centroids ...
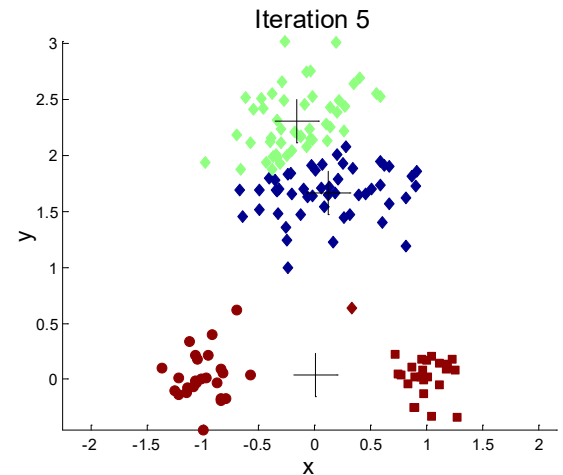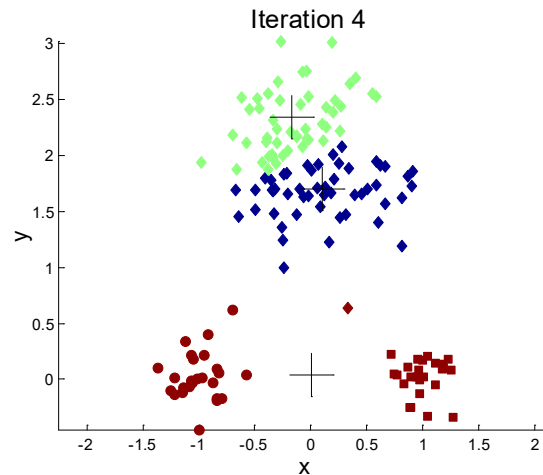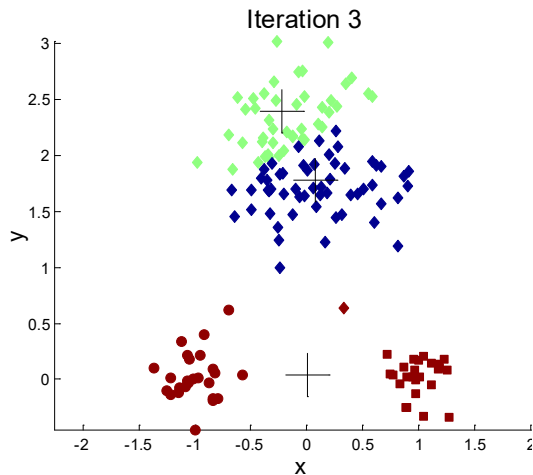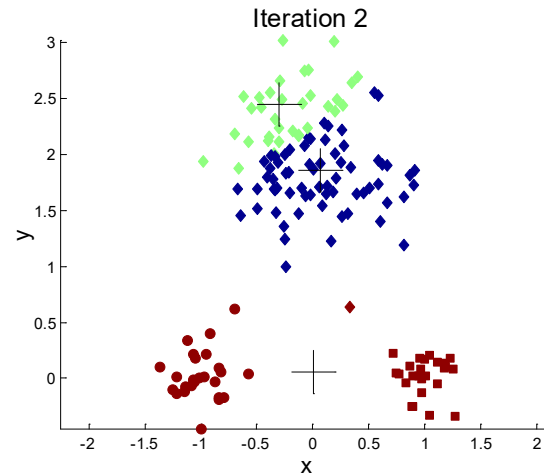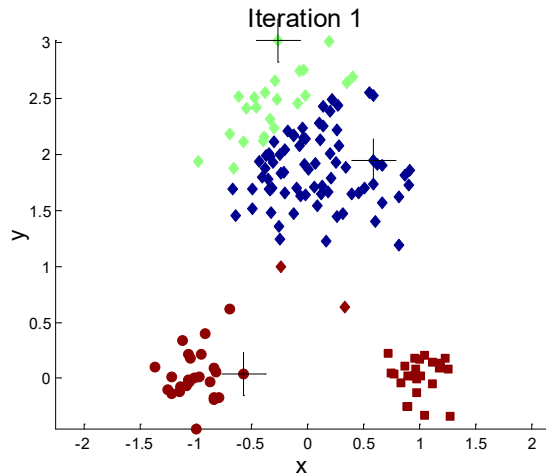
Solutions to Initial Centroids Problem

- Multiple runs. This is standard in most tools and typically helps.

- Sample and use hierarchical clustering to determine initial centroids.

- Select more than k initial centroids and then select among these initial centroids the ones that are far away from each other.

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 center

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - m_i\|^2$$

  - x is a data point in cluster C_i, m_i is the center for cluster C_i as the mean of all points in the cluster and $\|\cdot\|$ is the L2 norm (= Euclidean distance).
  - Given two clusterings, we can choose the one with the smallest error
  - Only compare clusterings with the same K! One easy way to reduce SSE is to increase K, the number of clusters

- **Note**: K-Means is a heuristic to minimize SSE.

Pre-processing and Post-processing

- Pre-processing
  - —**Normalize** the data (e.g., scale to unit standard deviation)
  - —**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

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.

Limitations of K-means: Differing Sizes
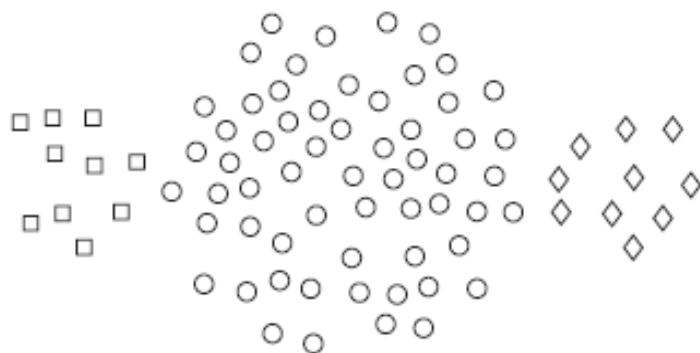
(a) Original points.
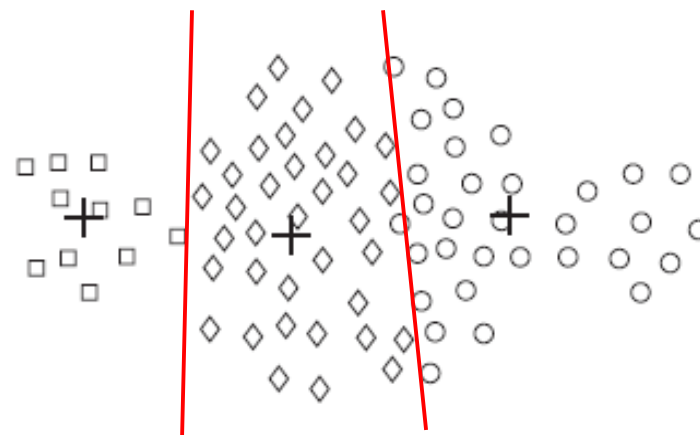
(b) Three K-means clusters.

Limitations of K-means: Differing Density

(a) Original points.
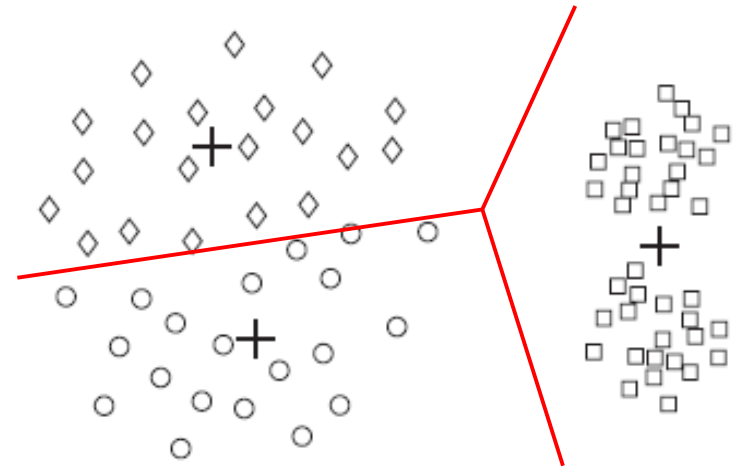
(b) Three K-means clusters.

Limitations of K-means: Non-globular Shapes
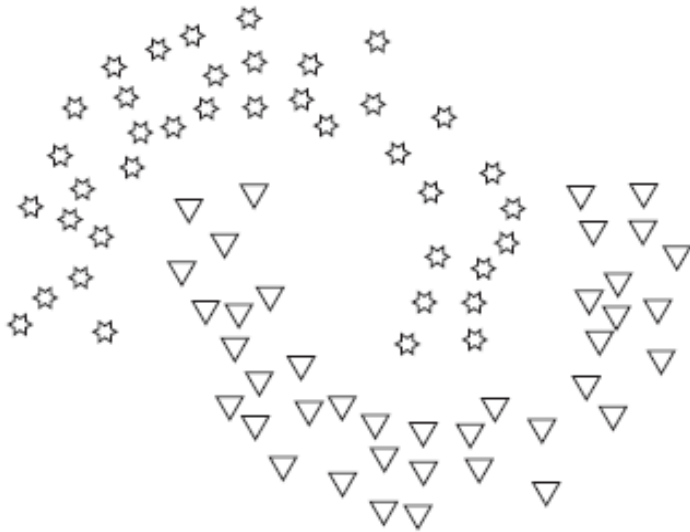
(a) Original points.
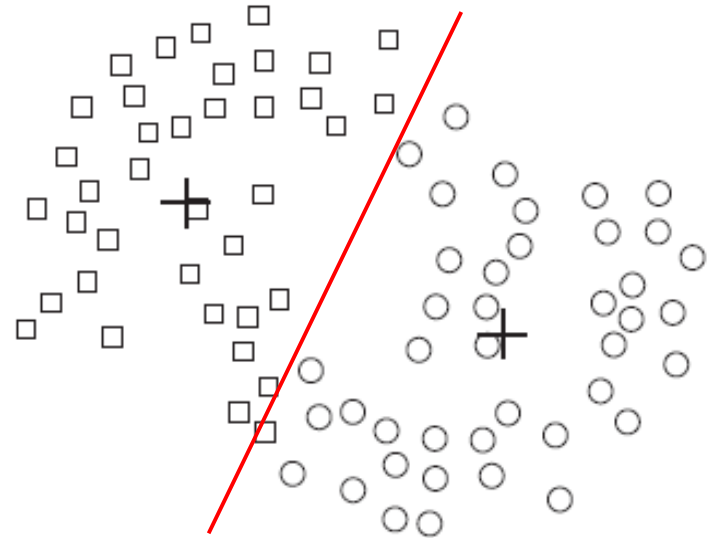
(b) Two K-means clusters.

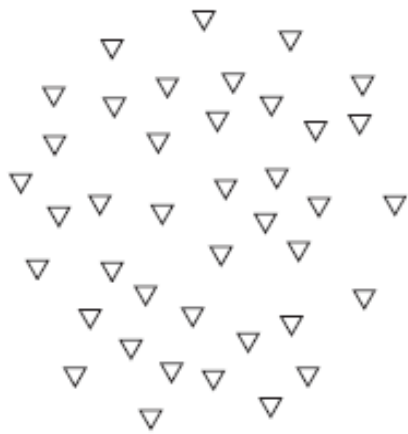# Overcoming K-means Limitations
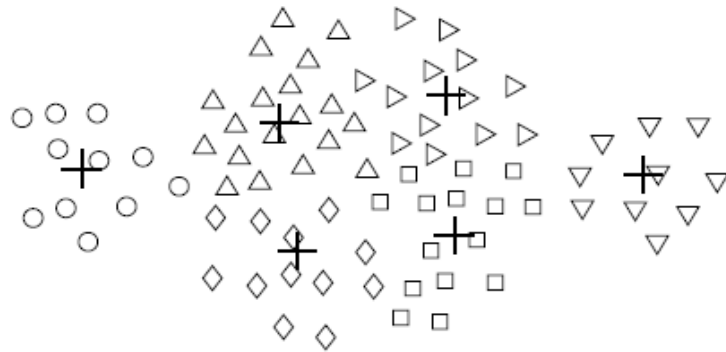
(a) Unequal sizes.

(b) Unequal densities.
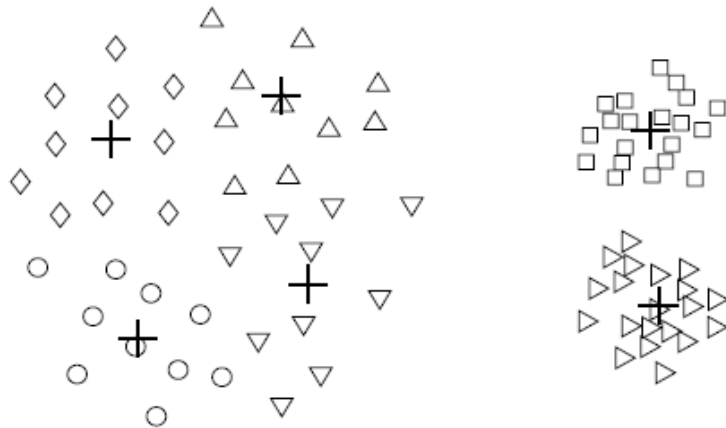
Use a larger number of clusters

Several clusters represent a true cluster

Overcoming K-means Limitations
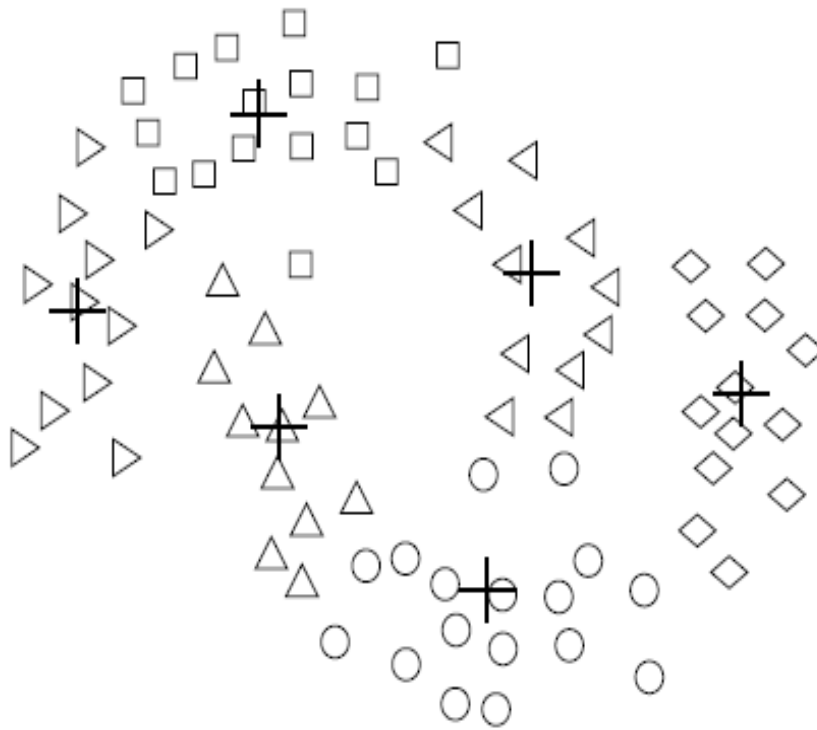
(c) Non-spherical shapes.

Use a larger number of clusters

Several clusters represent a true cluster

Topics

- Introduction
- Types of Clustering
- Types of Clusters
- **Clustering Algorithms**
  - K-Means Clustering
  - **Hierarchical Clustering**
  - Density-based Clustering
- Cluster Validation
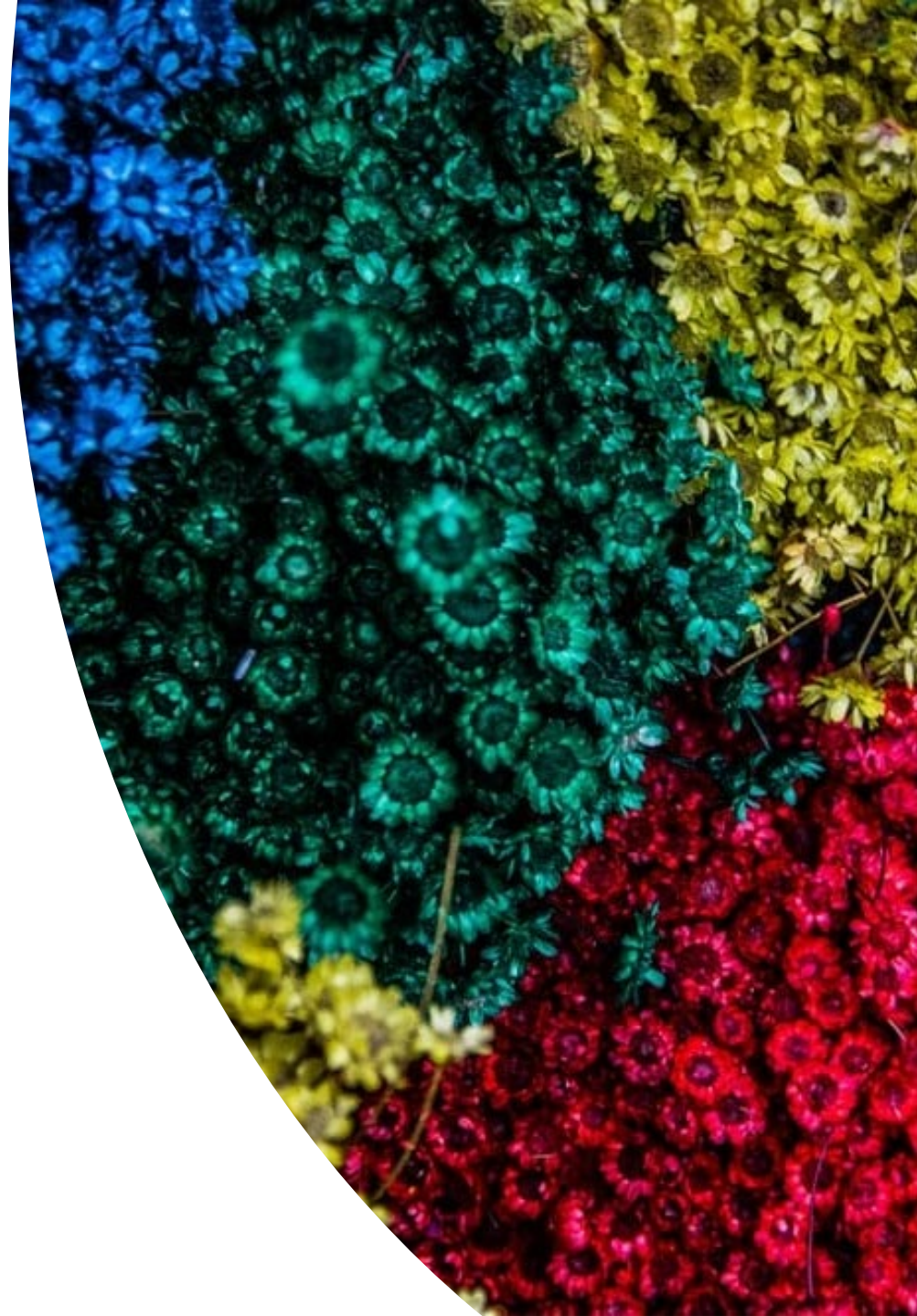
Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree called a **dendrogram**. The dendrogram shows at what distance points join into a cluster.
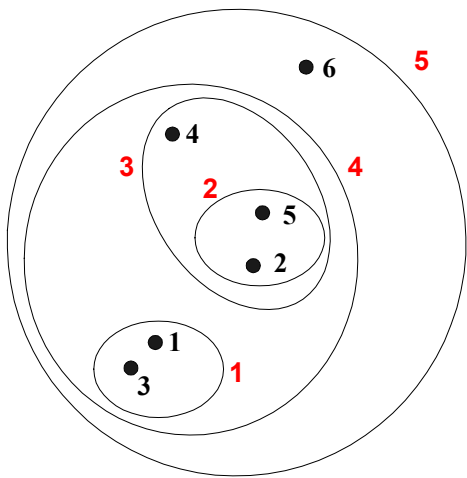
Dendrogram
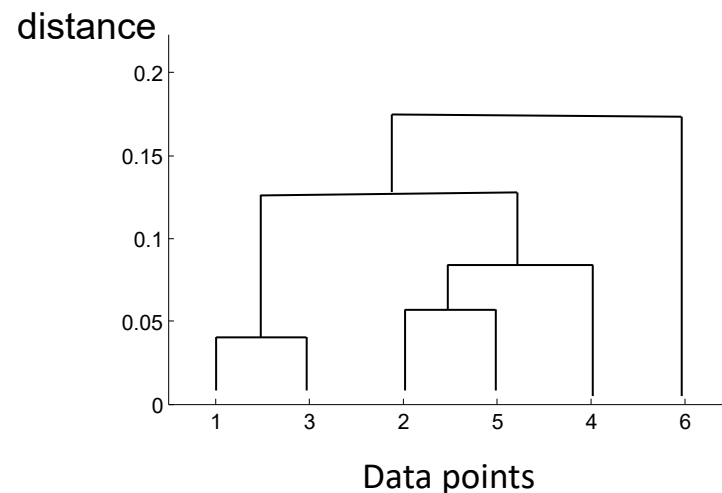
Strengths of Hierarchical Clustering

- You 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, …)

TREE OF LIFE

PLANTAE
ANGIOSPERM

ANIMALIA

FISH

REPTILE

BIRD

MAMMAL

LYCOPSID

FILICOPSID

AMPHIBIAN

GYMNOSPERM

CHORDATA

BRACHIOPOD

BRYOZOA

SPHENOPSID

ECHINODERM

FUNGI

BRYOPHYTA

GREEN ALGAE

ARTHROPOD

MOLLUSCA

CNIDARIA

ANNELID

BROWN-GREEN ALGAE

PORIFERA

EUBACTERIA

ARCHAE-
BACTERIA

PROTISTA
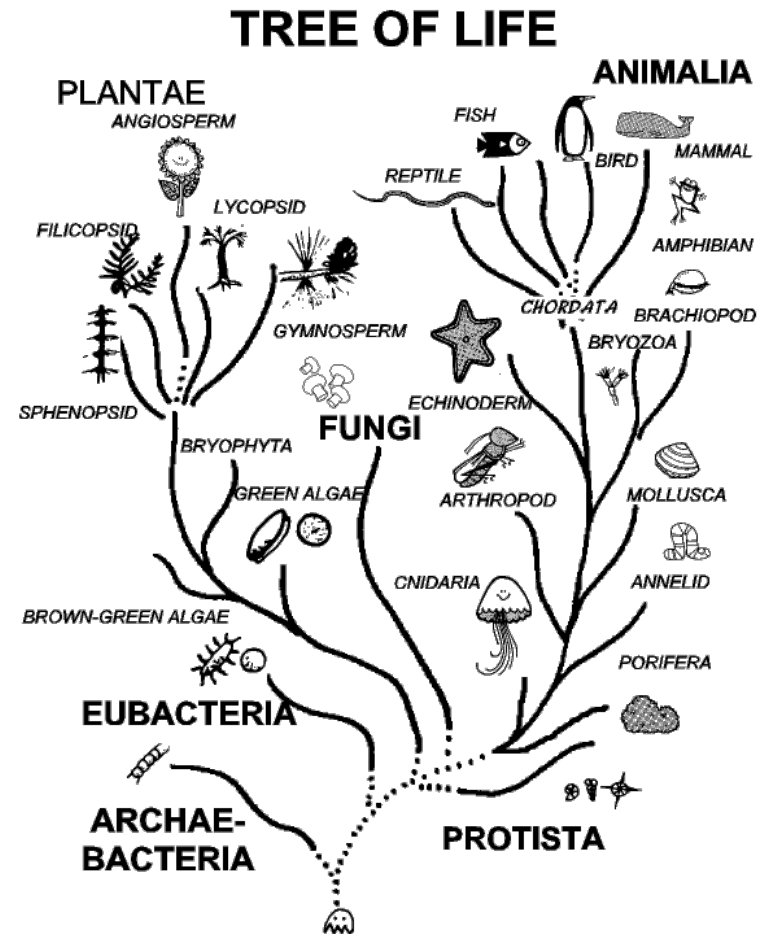
Hierarchical Clustering

- 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

- Agglomerative approach is more popular.
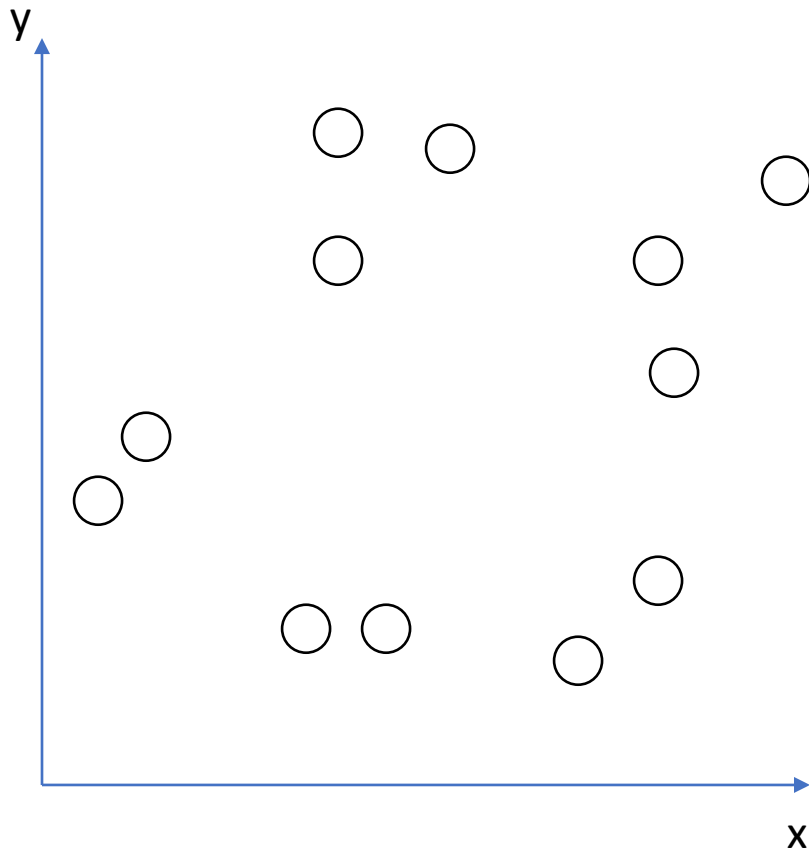- Basic algorithm is straightforward

  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

- A key operation is to compute the proximity between two clusters.

Starting Situation

- Start with clusters of individual points and a proximity matrix

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

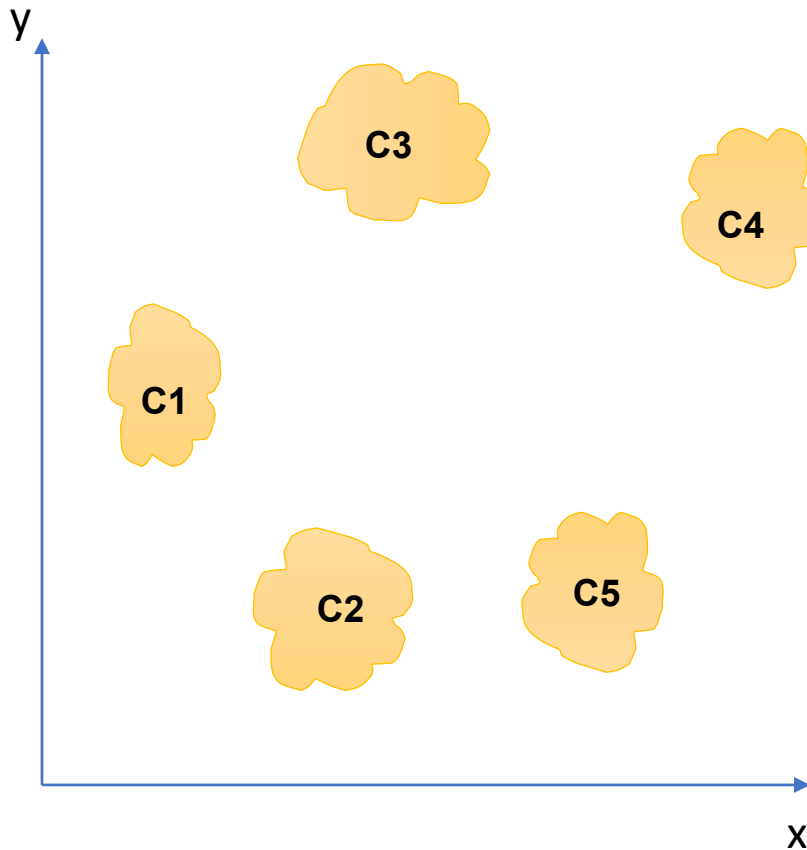**Proximity Matrix**

Dendrogram

Intermediate Situation

- After some merging steps, we have some clusters

Proximity Matrix

Dendrogram
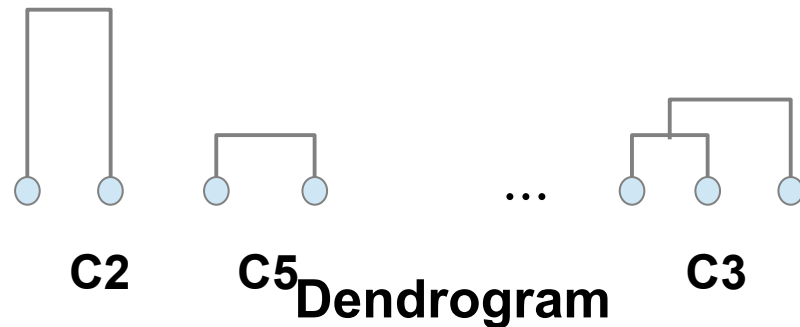
Intermediate Situation

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

Proximity Matrix

Dendrogram
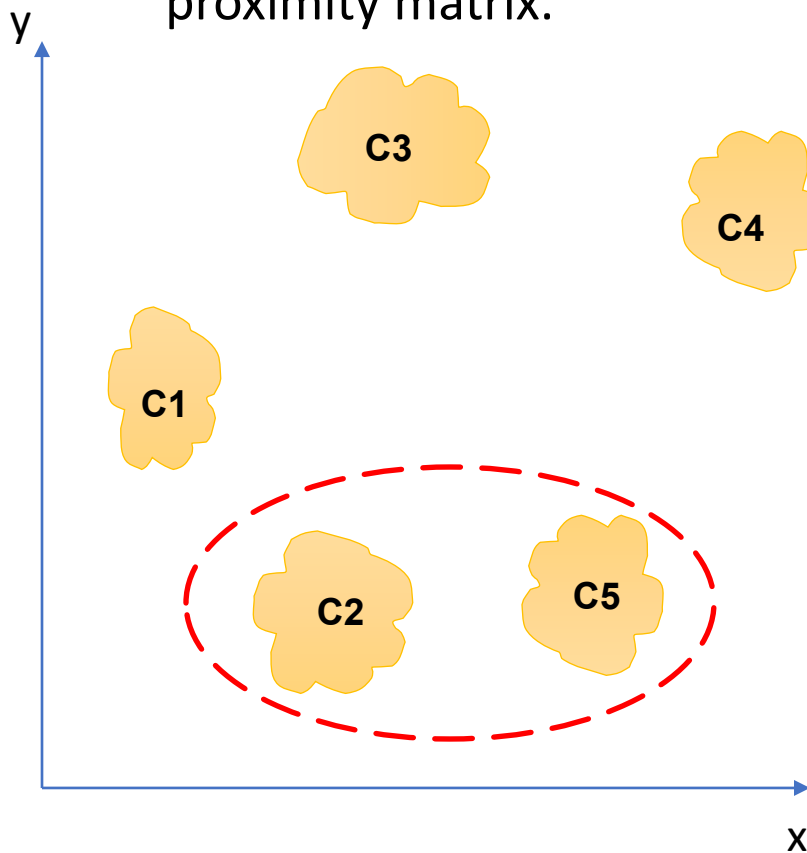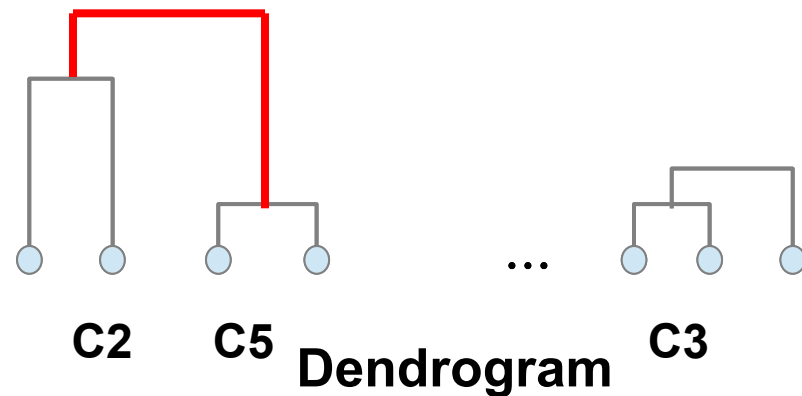
After Merging

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

Proximity Matrix

	C1	C2 ∪ C5	C3	C4
C1		?		
C2 ∪ C5	?	?	?	?
C3		?		
C4		?		

C2   C5   **Dendrogram**   C3

How to Define Inter-Cluster Similarity
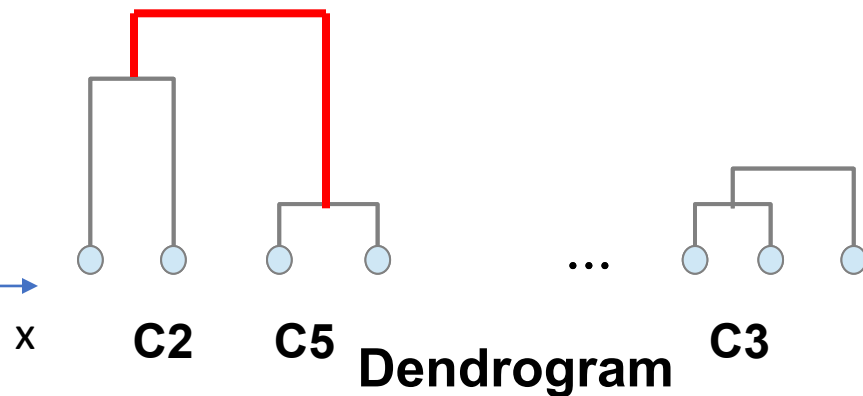
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity

- **MIN (Single Link)**
- MAX (Complete Link)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
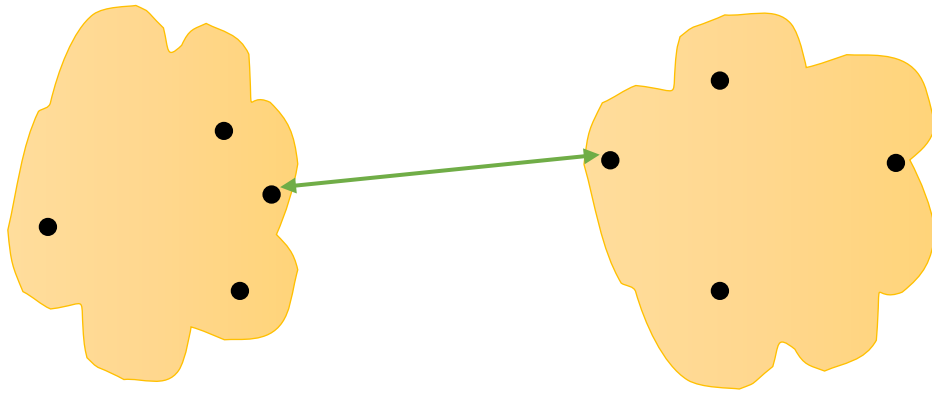
  Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

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

Proximity Matrix
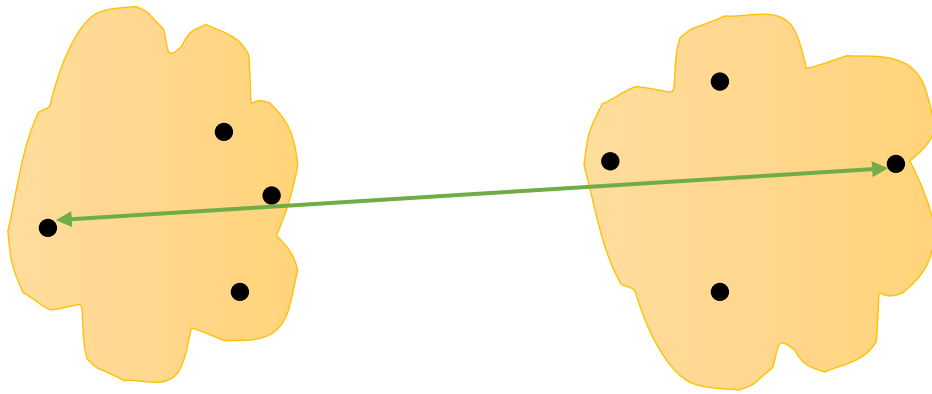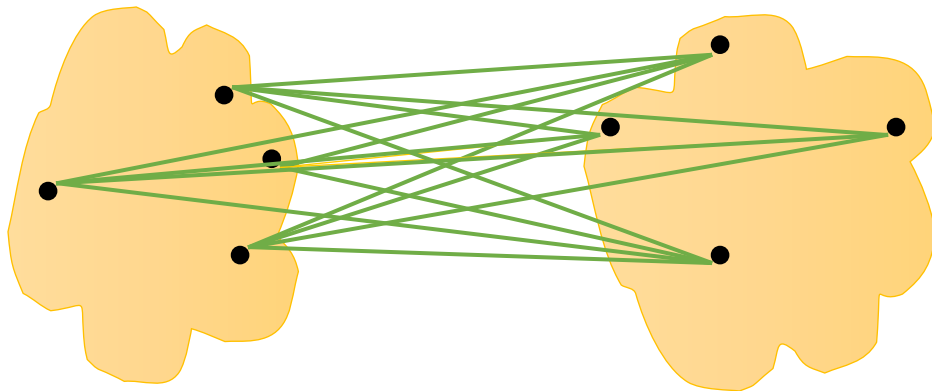
- MIN (Single Link)
- **MAX (Complete Link)**
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

  Ward's Method uses squared error

How to Define Inter-Cluster Similarity

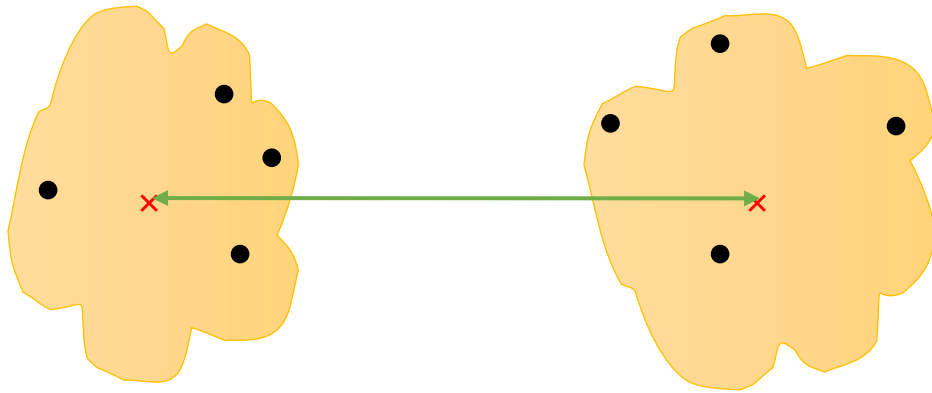|	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN (Single Link)
- MAX (Complete Link)
- **Group Average (Average Link)**
- Distance Between Centroids
- Other methods driven by an objective function

  Ward's Method uses squared error

How to Define Inter-Cluster Similarity

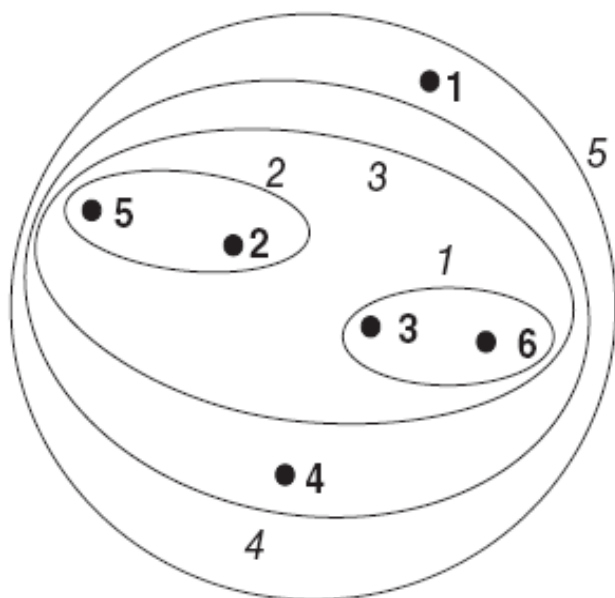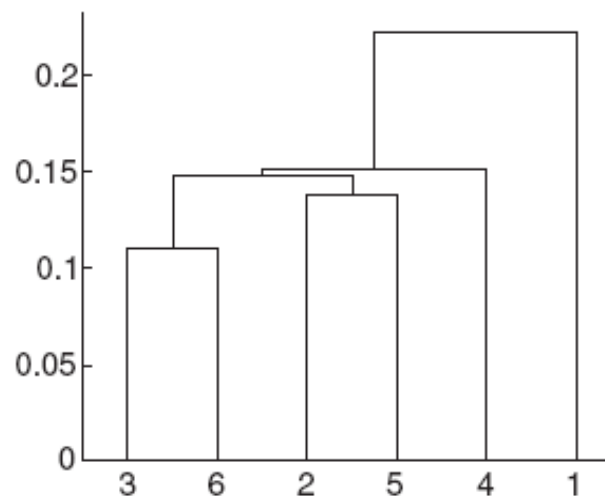|	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

- MIN (Single Link)
- MAX (Complete Link)
- Group Average
- **Distance Between Centroids**
- Other methods driven by an objective function

  Ward's Method uses squared error
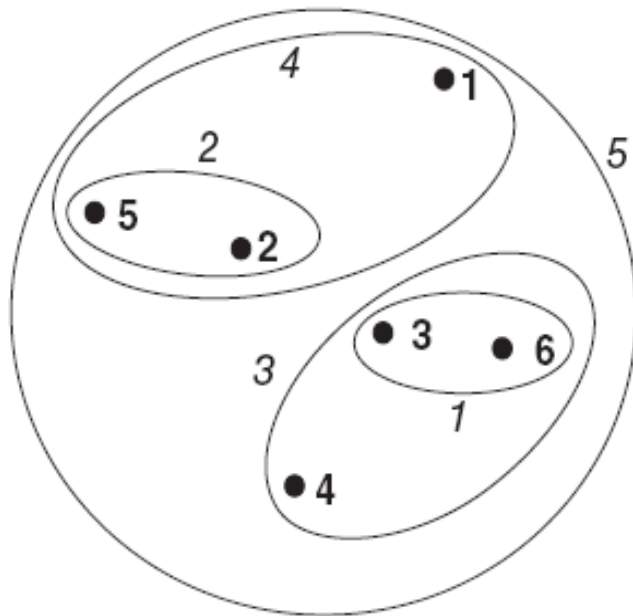
Single Link

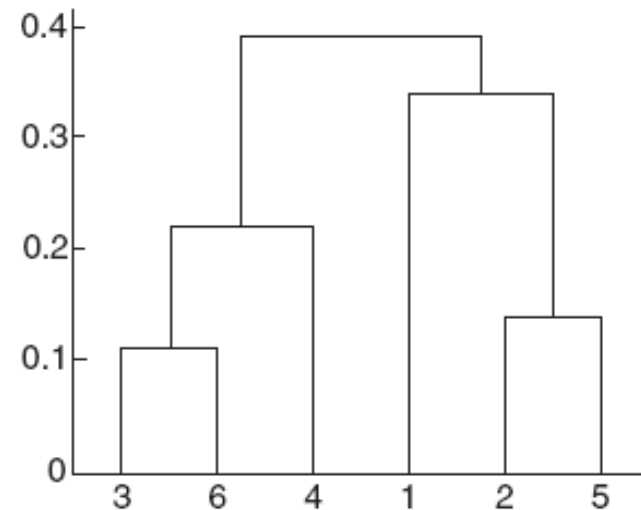(a) Single link clustering.

(b) Single link dendrogram.

Advantage: Non-spherical, non-convex clusters
Problem: Chaining

Complete Link

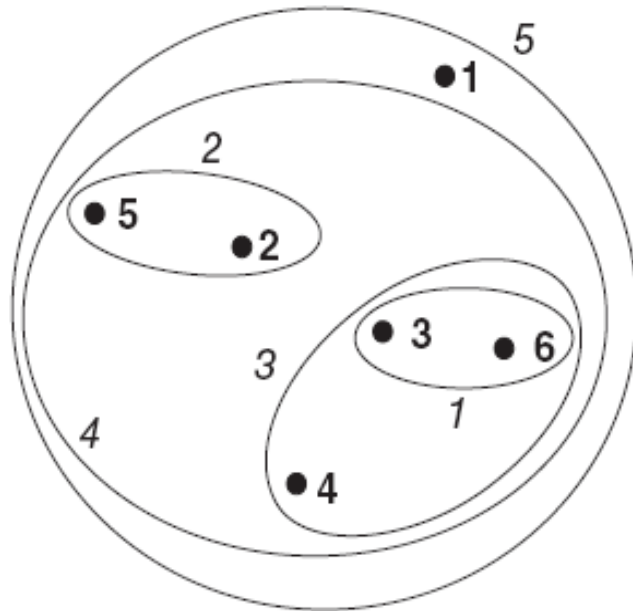(a) Complete link clustering.
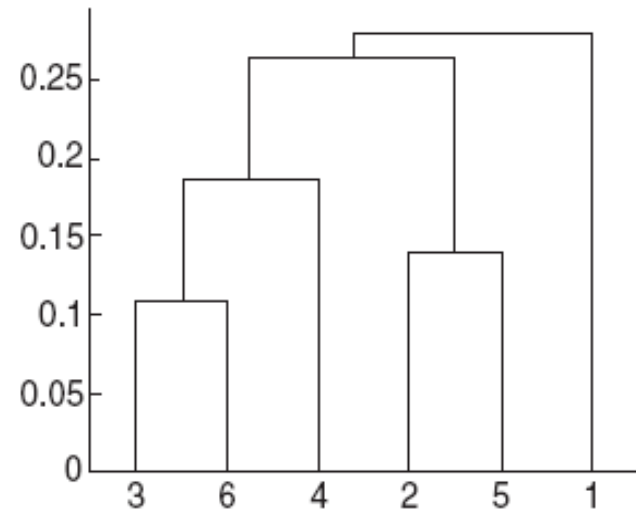
(b) Complete link dendrogram.

Advantage: more robust against noise (no chaining)

Problem: Tends to break large clusters,
Biased towards globular clusters

Average Link

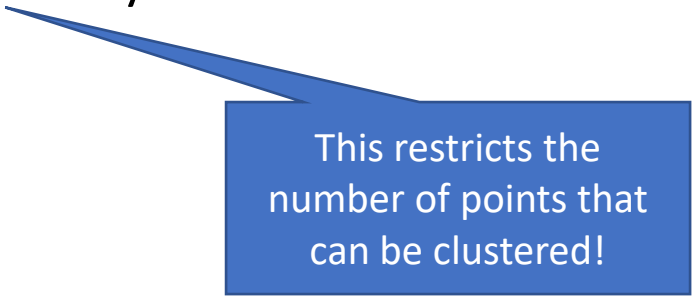(a) Group average clustering.

(b) Group average dendrogram.

Compromise between Single and Complete Link

Cluster Similarity: Ward's Method

- Similarity of two clusters is based on the increase in squared error when two clusters are merged

- Less susceptible to noise and outliers

- Biased towards globular clusters

- **Hierarchical analogue of K-means**

Hierarchical Clustering: Complexity

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

This restricts the number of points that can be clustered!

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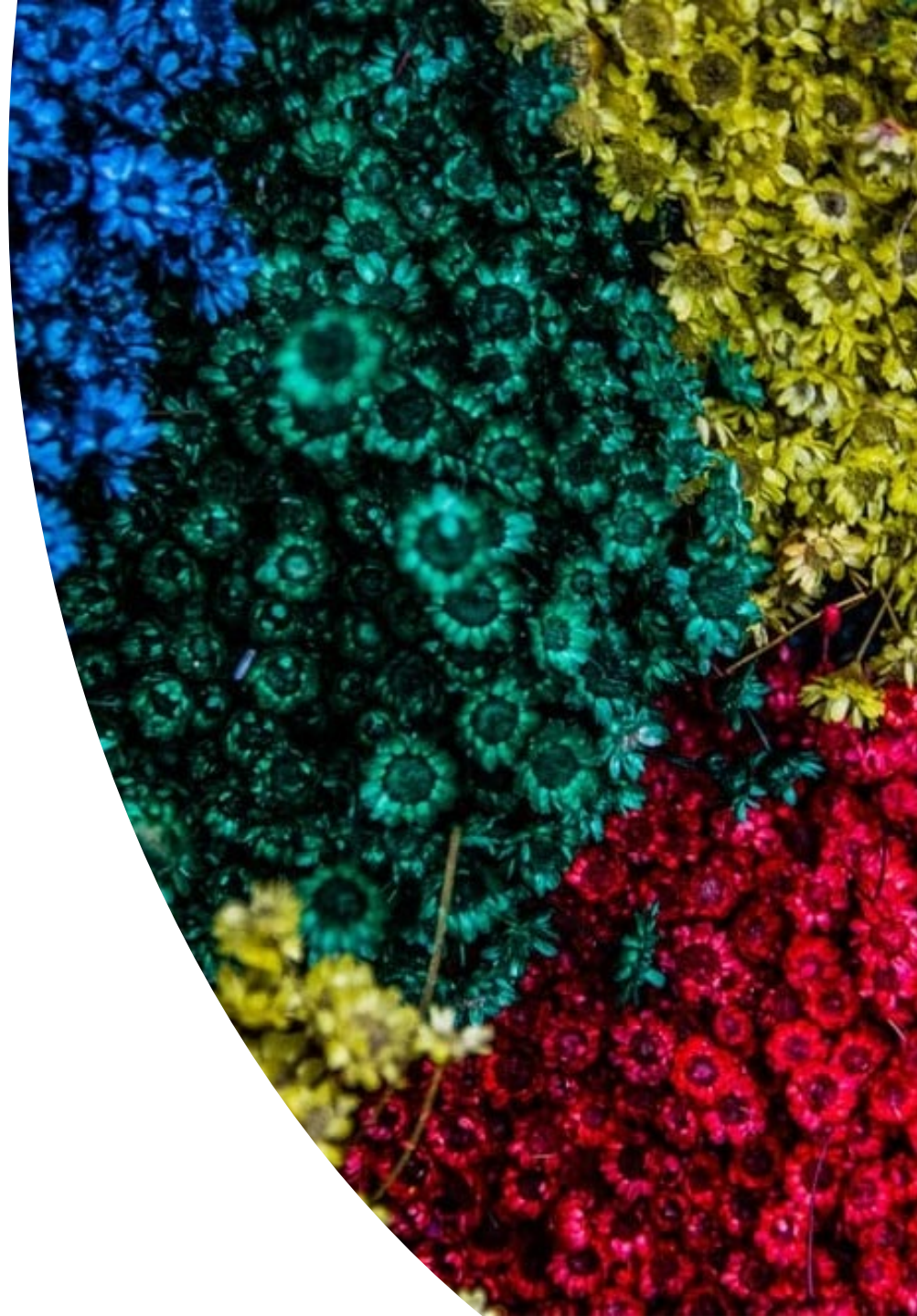 - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Hierarchical Clustering: Limitations

- **Greedy**: Once a decision is made to combine two clusters, it cannot be undone

- **No global 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
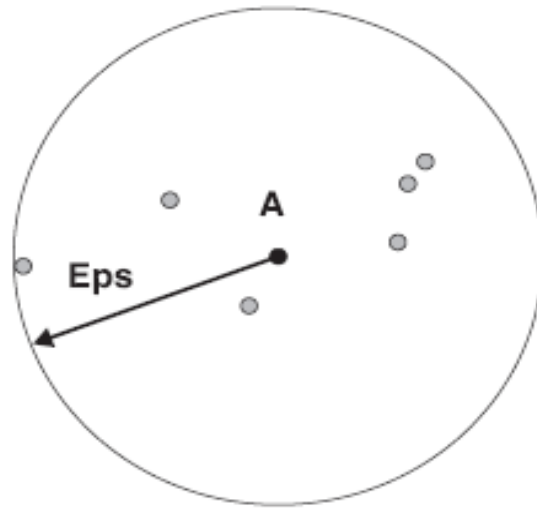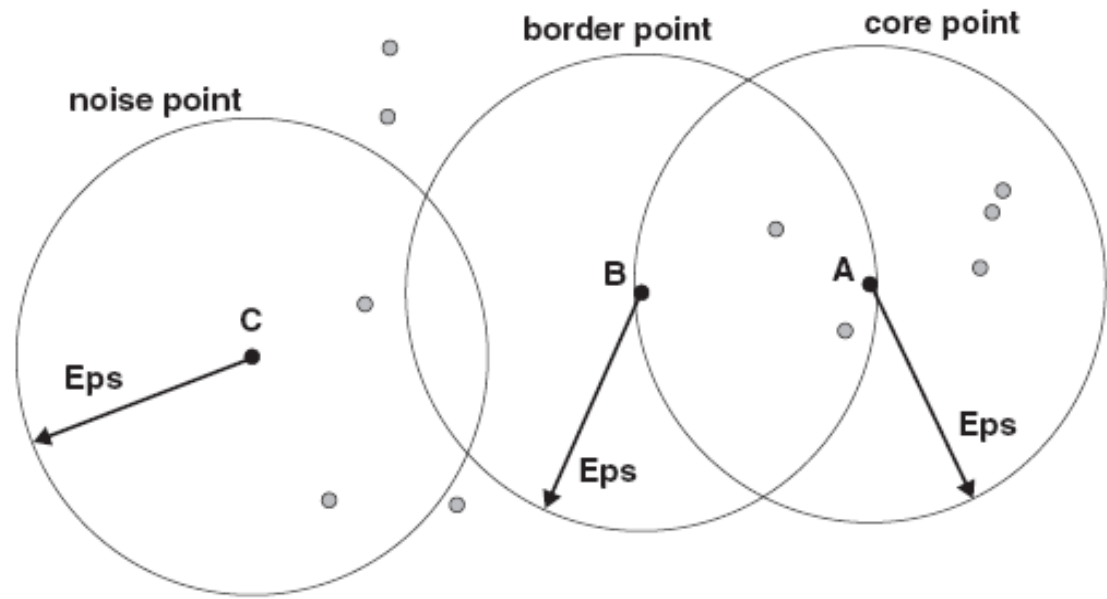 - Chaining, breaking large clusters

Topics

DBSCAN

Density = 7 points

- Density = number of points within a specified radius (Eps)

DBSCAN

MinPts = 5

- 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 Algorithm

```
DBSCAN(D, eps, MinPts)
  C = 0
  for each unvisited point P in dataset D
    mark P as visited
    NeighborPts = regionQuery(P, eps)
    if sizeof(NeighborPts) < MinPts
      mark P as NOISE
    else
      C = next cluster
      expandCluster(P, NeighborPts, C, eps, MinPts)

expandCluster(P, NeighborPts, C, eps, MinPts)
  add P to cluster C
  for each point P' in NeighborPts
    if P' is not visited
      mark P' as visited
      NeighborPts' = regionQuery(P', eps)
      if sizeof(NeighborPts') >= MinPts
        NeighborPts = NeighborPts joined with NeighborPts'
    if P' is not yet member of any cluster
      add P' to cluster C
```
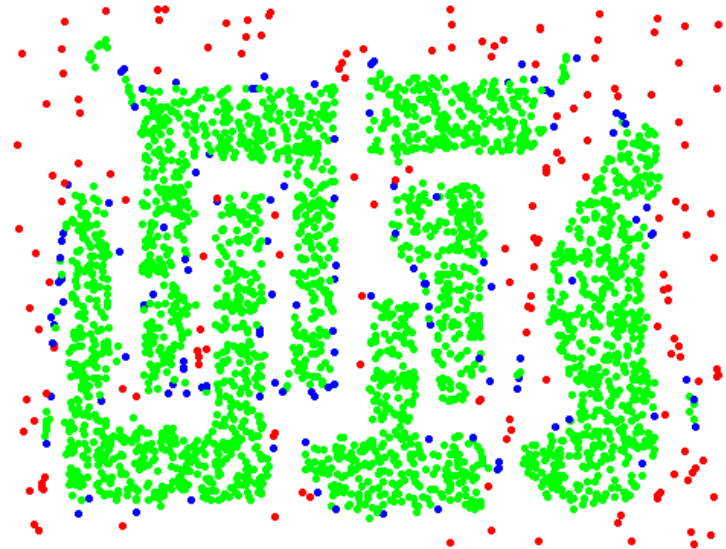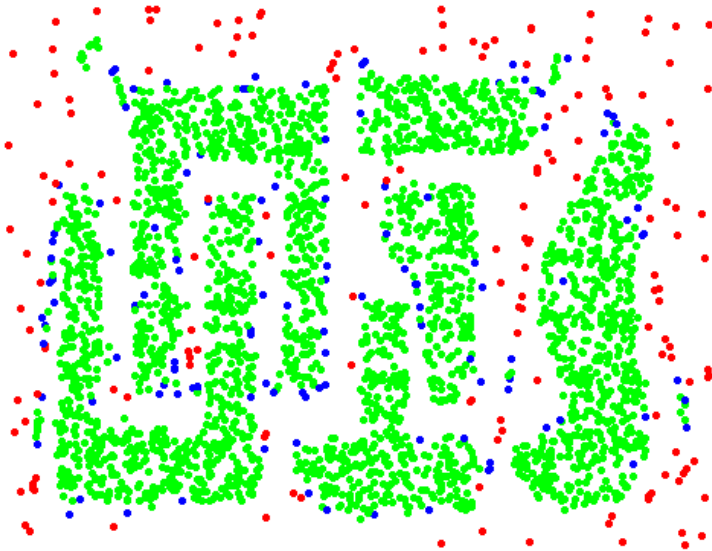
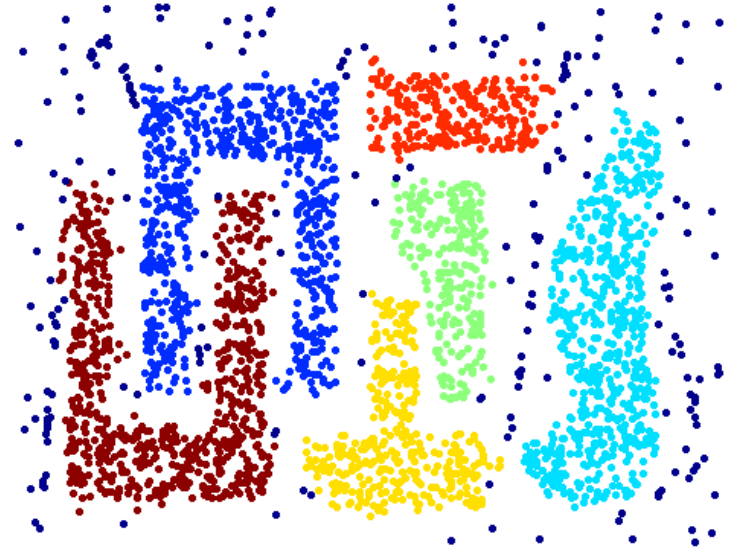# DBSCAN: Core, Border and Noise Points

Original Points

Point types: core, border and noise

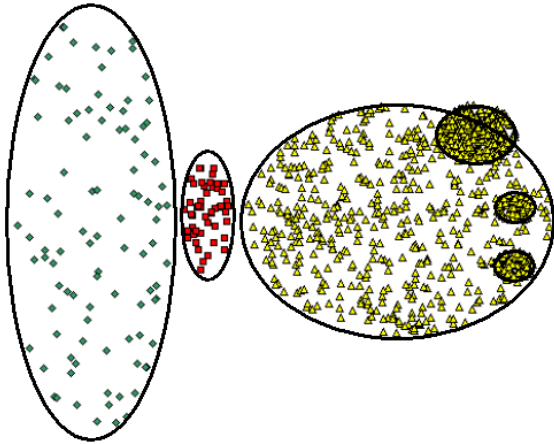**Eps = 10, MinPts = 4**

DBSCAN: Determine Clusters

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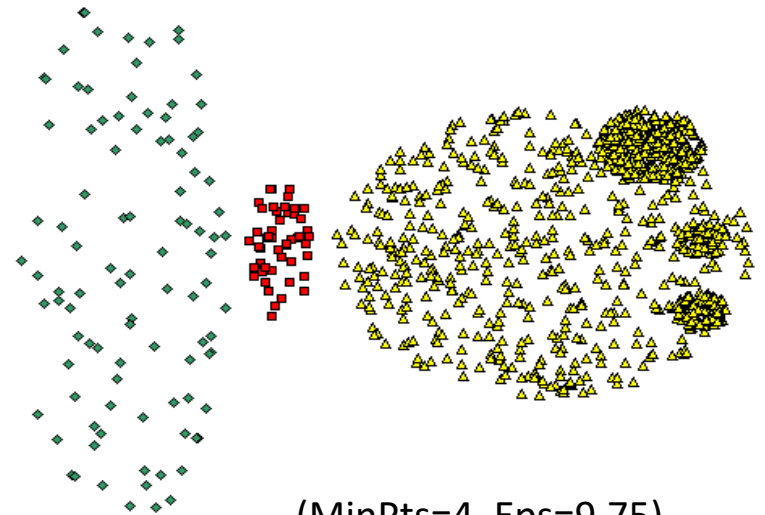border and noise**

Clusters

- Resistant to **Noise**

- Can handle clusters of different **shapes and sizes**

- **Eps and MinPts** depend on each other and can be hard to specify
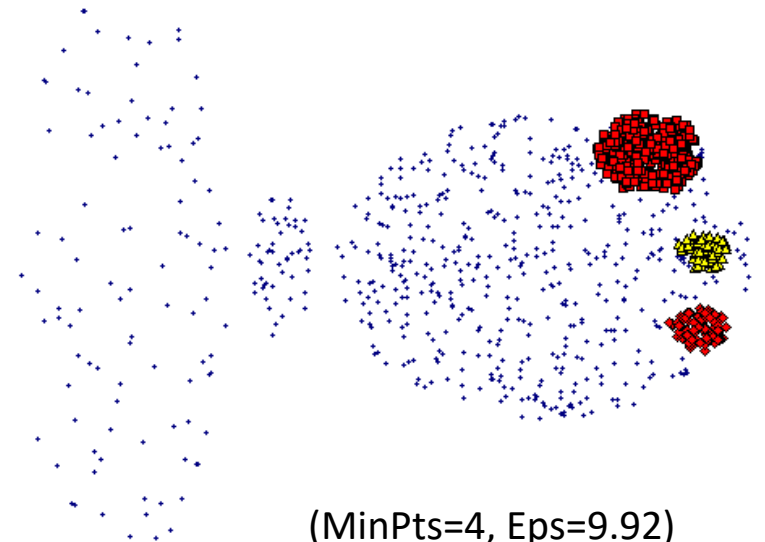
When DBSCAN Does NOT Work Well

Original Points
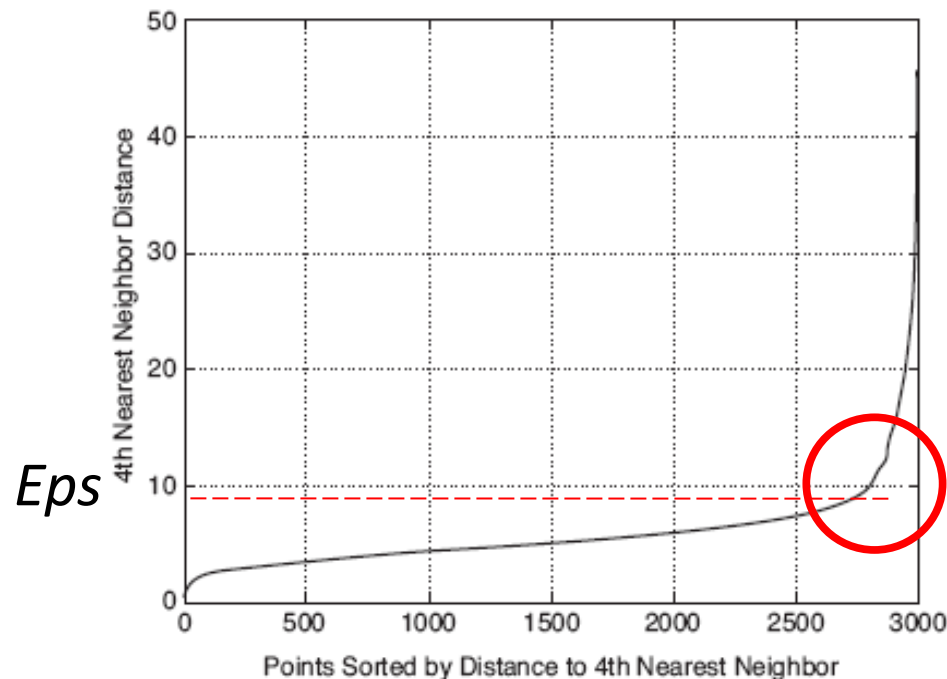
- Varying densities
- High-dimensional data

(MinPts=4, Eps=9.75).

(MinPts=4, Eps=9.92)

DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their kth nearest neighbors are at roughly the same distance
- Noise points have the kth nearest neighbor at farther distance
- So, plot sorted distance of every point to its kth nearest neighbor

MinPts = k

Eps

Some Other Clustering Algorithms

- **Center-based Clustering**
  - Fuzzy c-means
  - PAM (Partitioning Around Medoids)

- **Mixture Models**
  - Expectation-maximization (EM) algorithm

- **Hierarchical**
  - CURE (Clustering Using Representatives): shrinks points toward center
  - BIRCH (balanced iterative reducing and clustering using hierarchies)

- **Graph-based Clustering**
  - Graph partitioning on a sparsified proximity graph
  - Shared nearest-neighbor (SNN graph)

- **Spectral Clustering**
  - Reduce the dimensionality using the spectrum of the similarity, and cluster in this space.
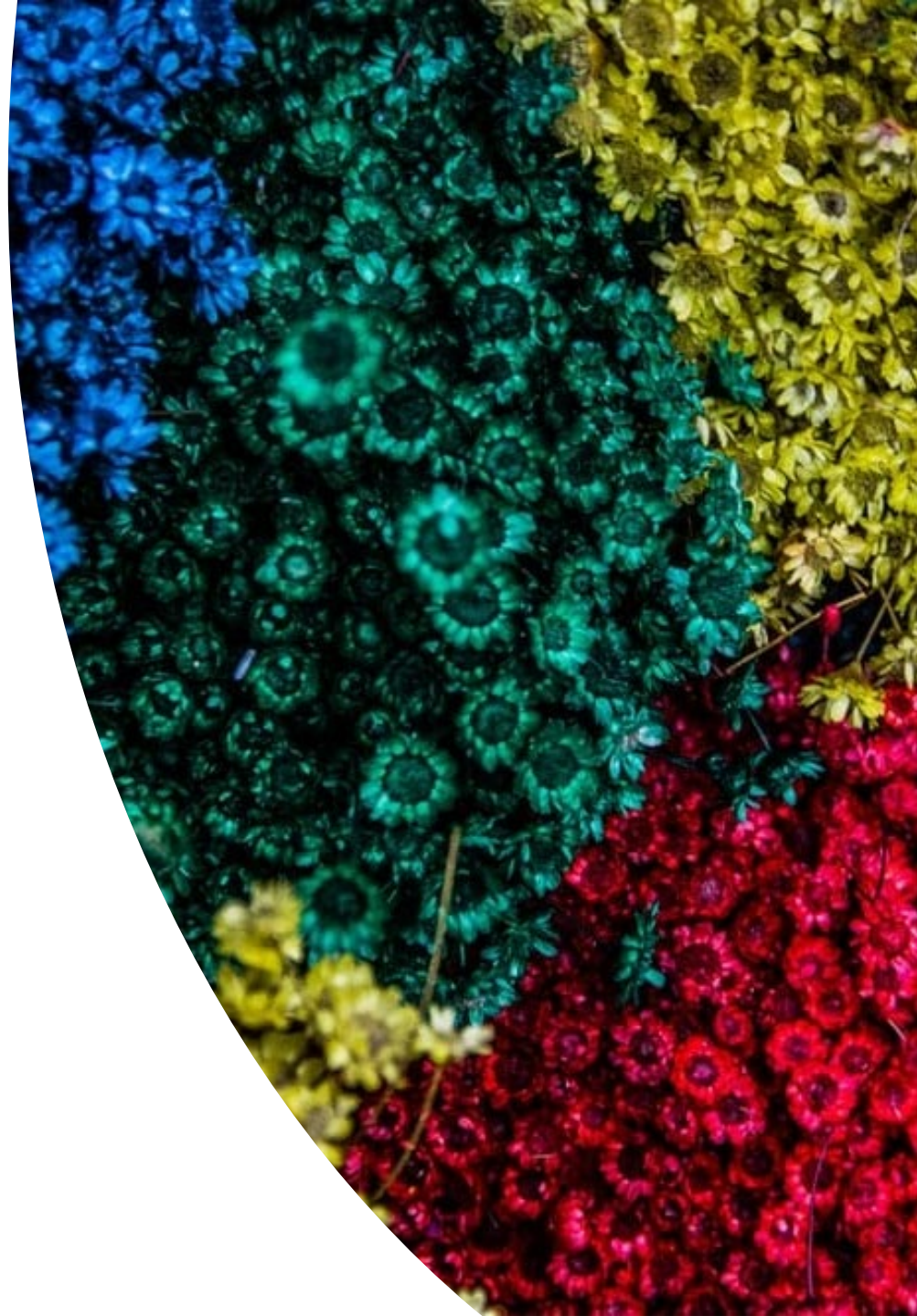
- **Subspace Clustering**
- **Data Stream Clustering**

Topics

- Introduction
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
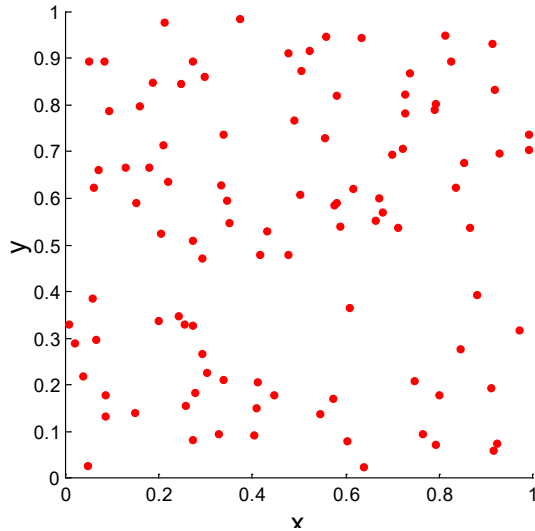- **Cluster Validation**

Cluster Validity

- For supervised classification (= we have a class label) we have a variety of measures to evaluate how good our model is: Accuracy, precision, recall

- For cluster analysis (=unsupervised learning), the analogous question is:
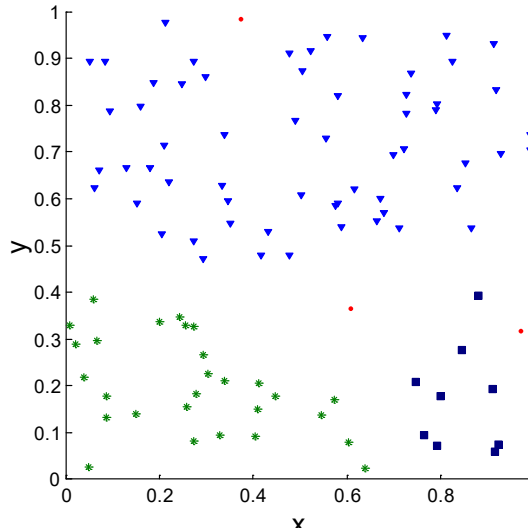
How to evaluate the "goodness" of the resulting clusters?
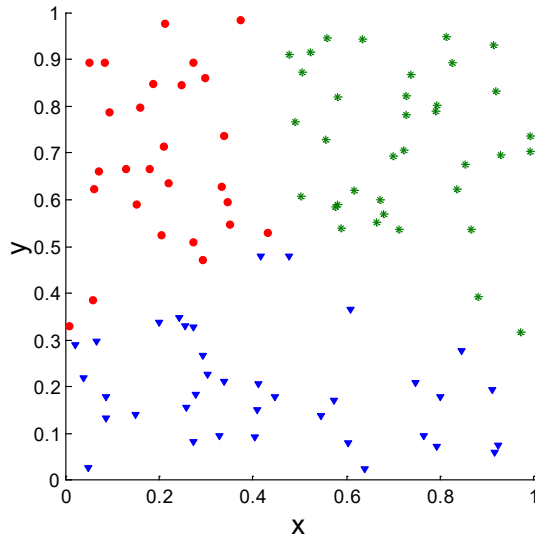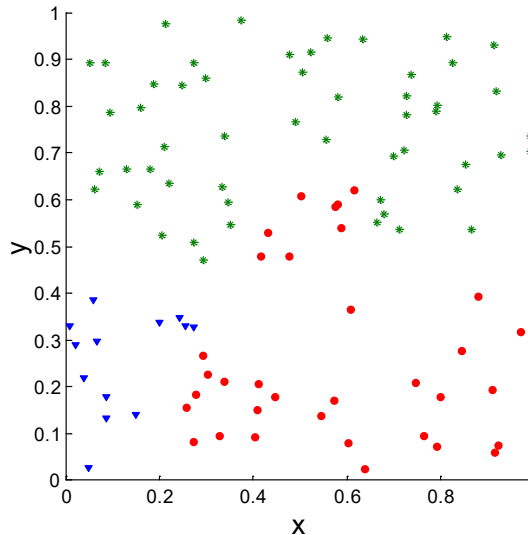
Clusters found in Random Data (Overfitting)

If you tell a clustering algorithm to find clusters then it will!

Different Aspects of Cluster Validation

1. Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data (e.g., to avoid overfitting).

2. **External Validation:** Compare the results of a cluster analysis to externally known class labels (ground truth).

3. **Internal Validation**: Evaluating how well the results of a cluster analysis fit the data without reference to external information.

4. **Compare clusterings** to determine which is better.

5. Determining the **'correct' number of clusters**.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.
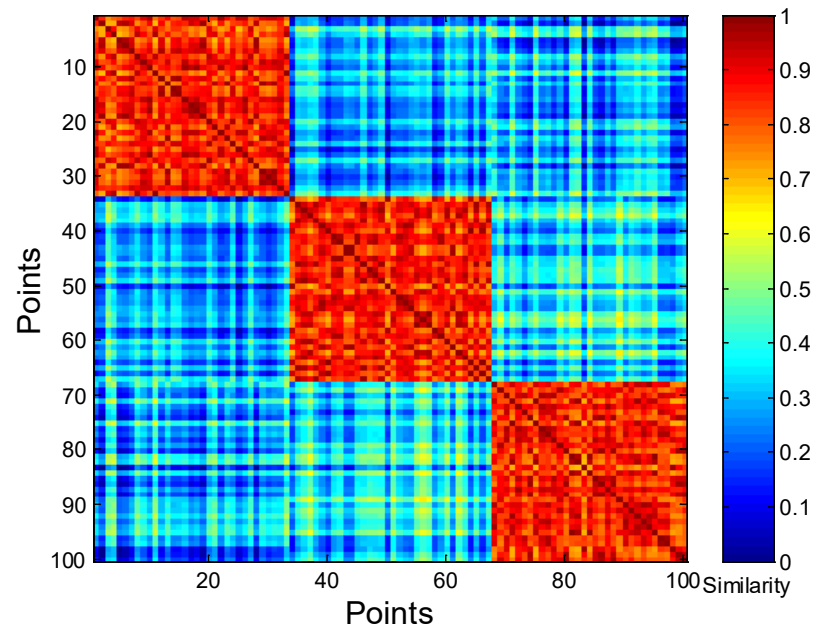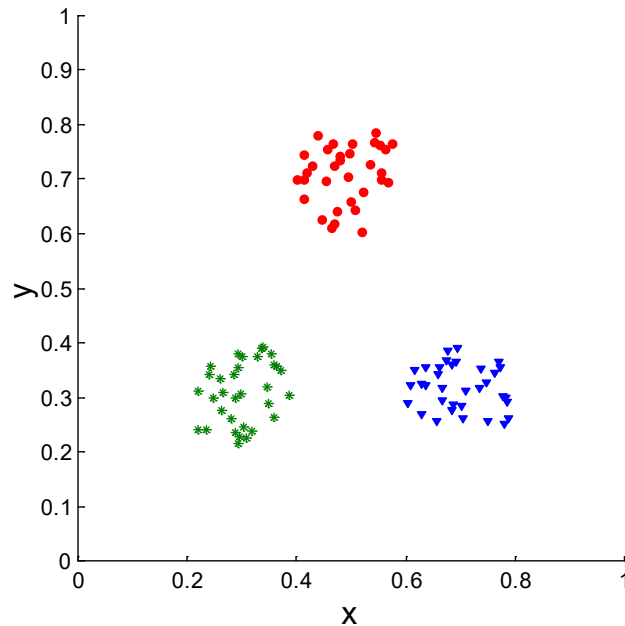
Measures of Cluster Validity

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, Purity, Rand index
- **Internal Index**: Used to measure the goodness of a clustering structure without respect to external information.
    - Sum of Squared Error (SSE), Silhouette coefficient
- **Relative Index**: Used to compare two different clusterings or clusters.
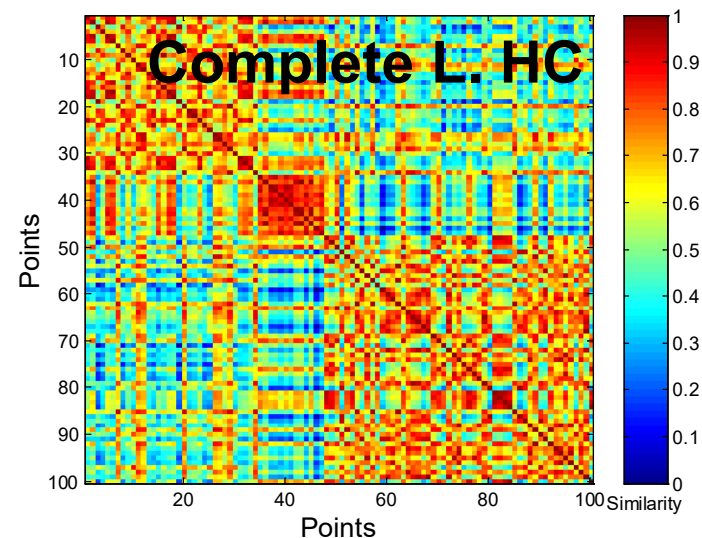   - Often an external or internal index is used for this function, e.g., SSE or entropy
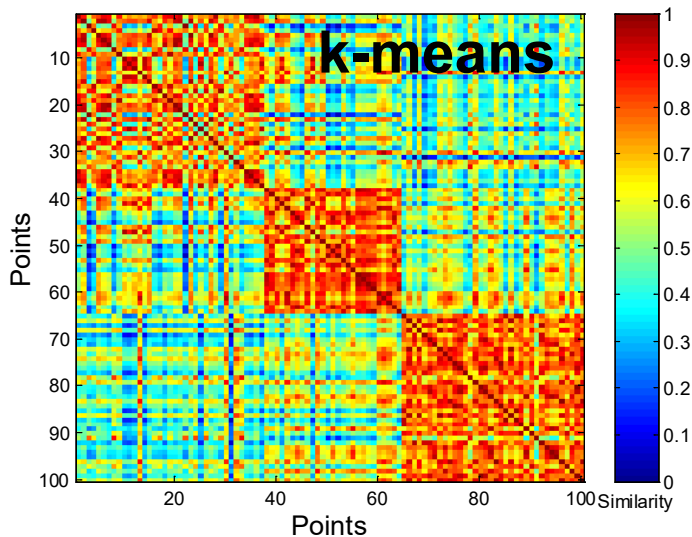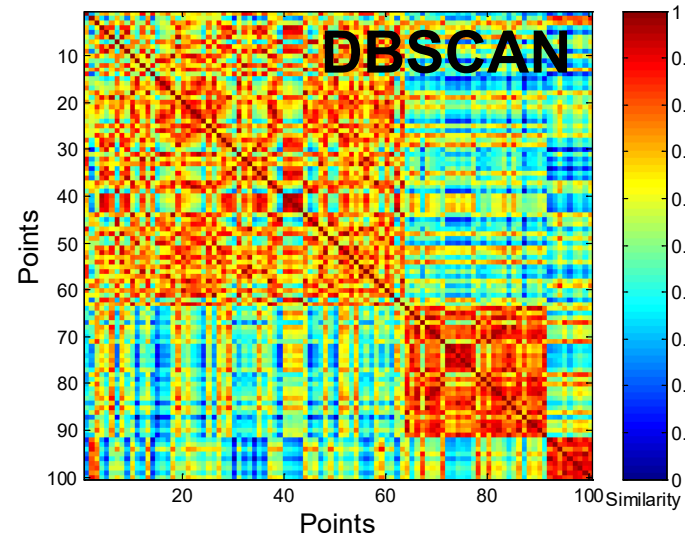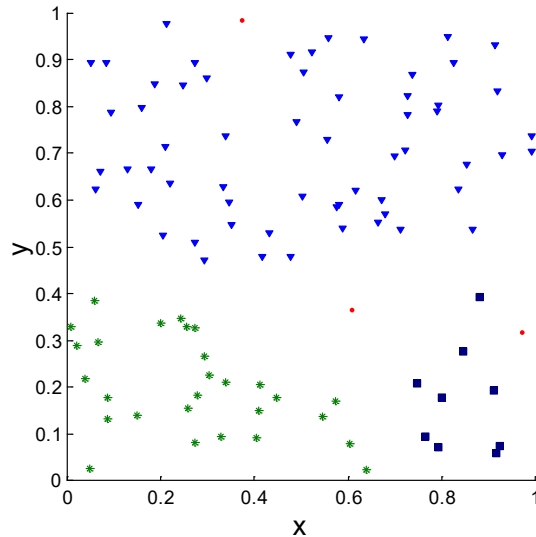
Similarity Matrix Visualization for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.

Similarity Matrix Visualization for Cluster Validation

- Clusters in random data are not as crisp

Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix representing the data
  - Incidence Matrix representing the clustering
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters

- Compute the correlation between the two matrices

- High correlation indicates that points that belong to the same cluster are close to each other.

- Not a good measure for some density or contiguity-based clusters (e.g., single link HC).

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.
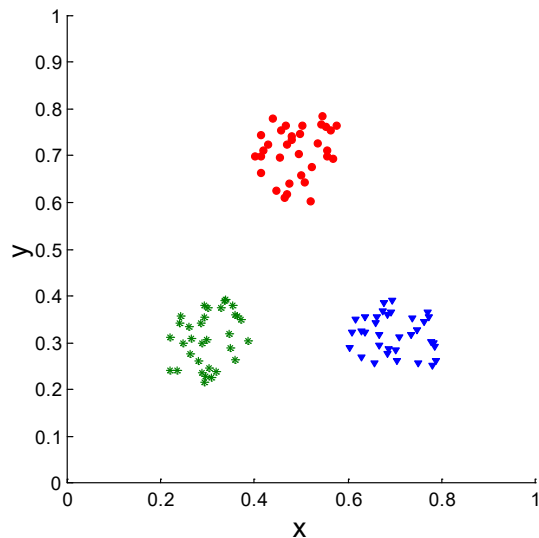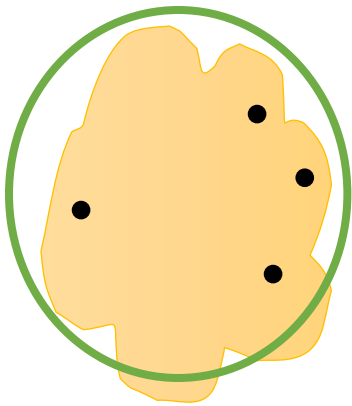
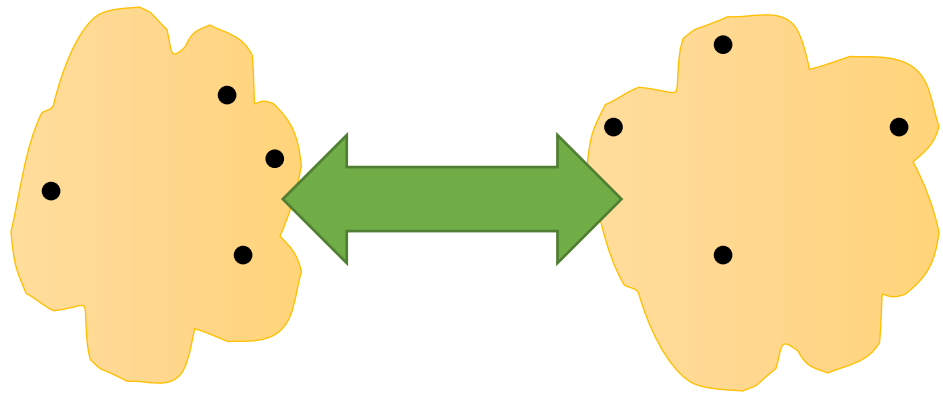

Corr = -0.9235
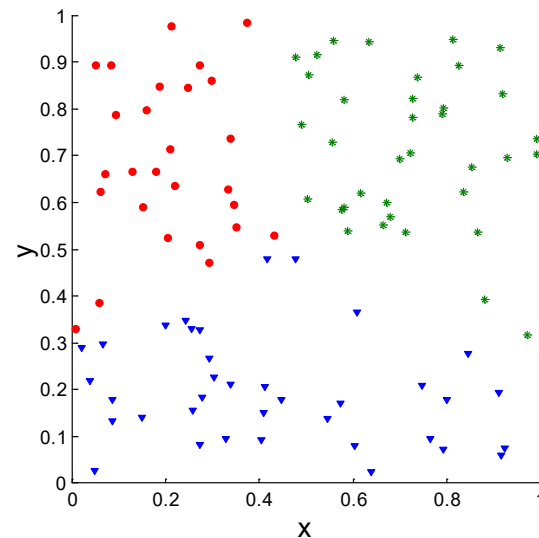
Corr = -0.5810

Note: Correlation is always negative between distance matrix and incidence matrix

Internal Measures: Cohesion and Separation

- **Cluster Cohesion**: Measures how closely related objects in a cluster are.

- **Cluster Separation:** Measure how distinct or well-separated a cluster is from other clusters.

cohesion

separation

Internal Measures: Sum of Squares

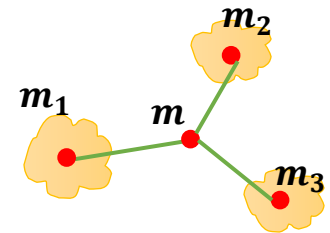- **Cluster Cohesion**: Within cluster sum of squares (WSS=SSE)

$$WSS = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - m_i\|^2$$
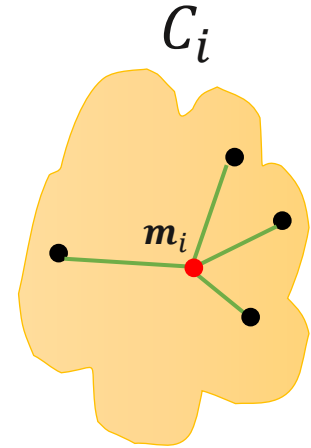
- **Cluster Separation:** Between cluster sum of squares (BSS)

$$BSS = \sum_{i=1}^{K} |C_i| \, \|m_i - m\|^2$$

Where $|C_i|$ is the size of cluster i and m is the centroid of the data space

- Total sum of squares: $TSS = \sum_x \|x - m\|^2$

$$TSS = WSS + BSS$$

C_i

Internal Measures: Sum of Squares

TSS = BSS + WSS = constant for a given data set
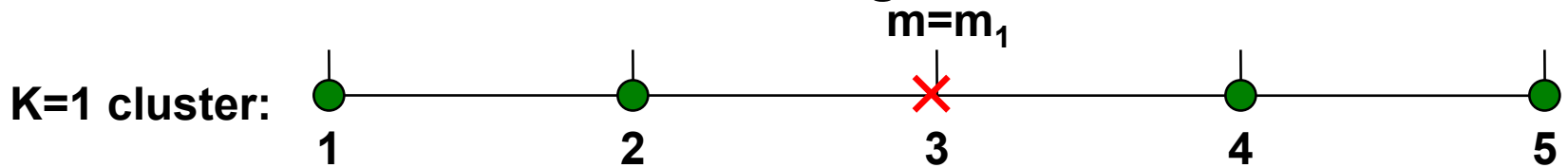
K=1 cluster:

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

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

$$Total = 10 + 0 = 10$$
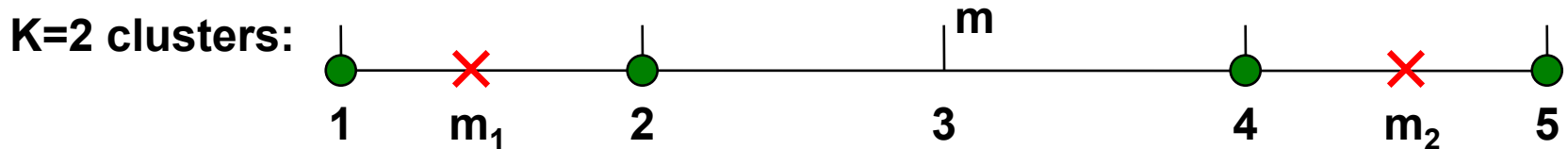
K=2 clusters:

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

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

$$Total = 1 + 9 = 10$$

Internal Measures: Choosing k with Sum of Squares

- SSE is good for comparing two clusterings or two clusters (average SSE).

- Can also be used to estimate the number of clusters

Data points forming 10 clusters

Look for the knee
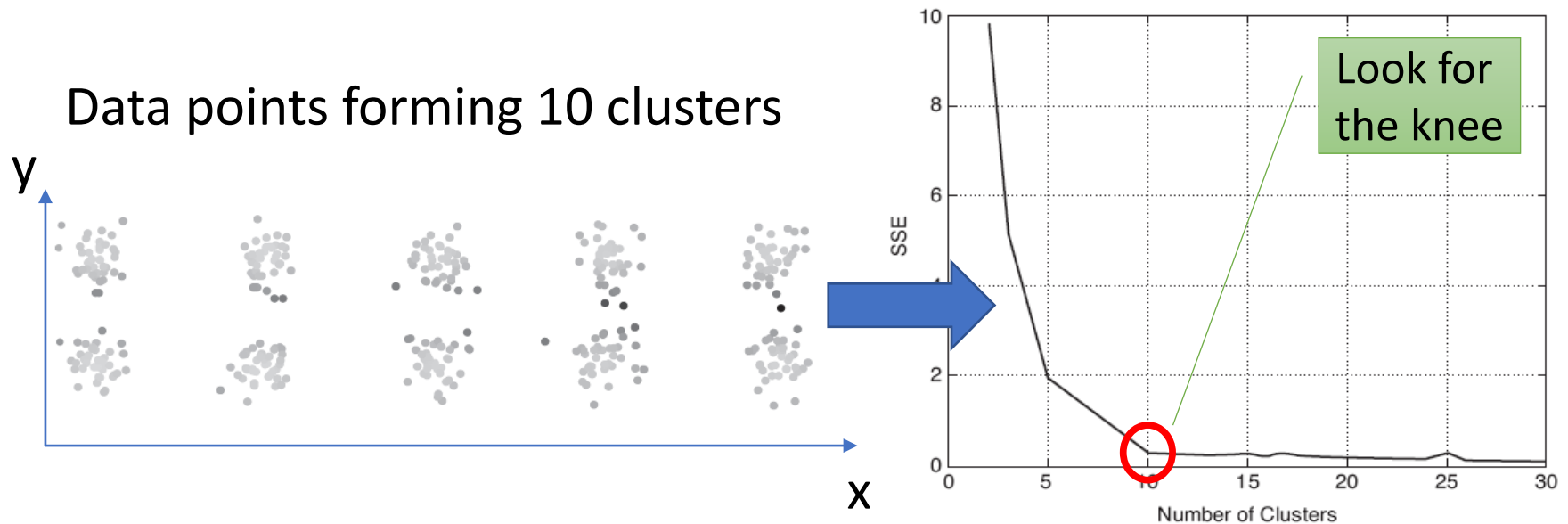
Internal Measures: Silhouette Coefficient

- 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.
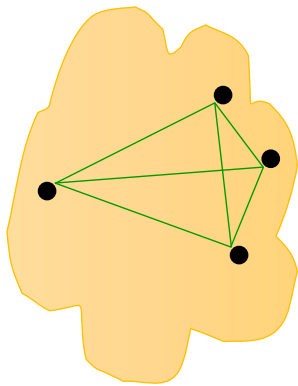
cohesion   separation

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points. For an individual point i:
  - Calculate a(i) = average dissimilarity of i to all other points in its cluster
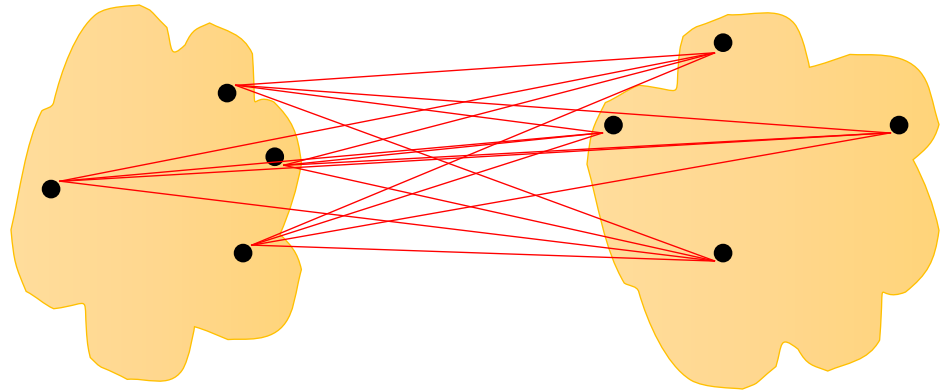 - Calculate b(i) = lowest average dissimilarity of i to any other)

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

$$-1 \leq s(i) \leq 1$$

Cohersion   Separation
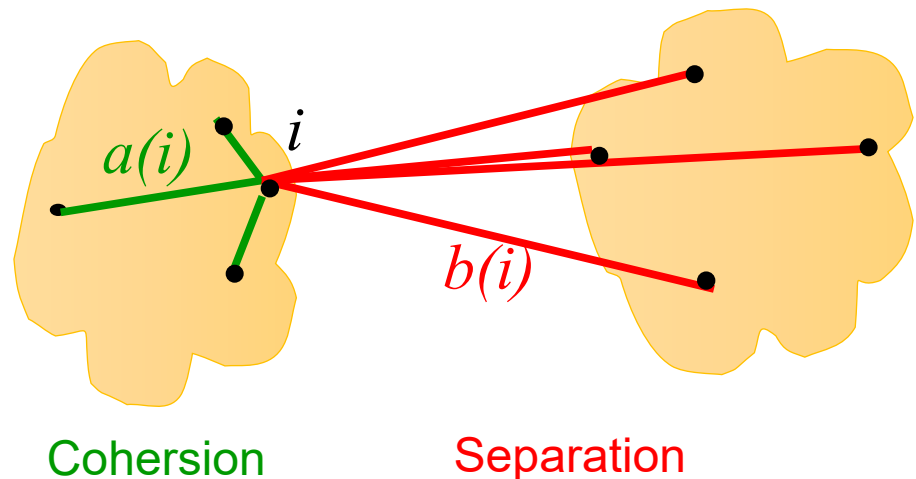
- The closer to 1 the better.
- Can calculate the Average Silhouette width for a cluster or a clustering

Internal Measures: Silhouette Plot

Silhouette plot of pam(x = dis.bc, k = 5)
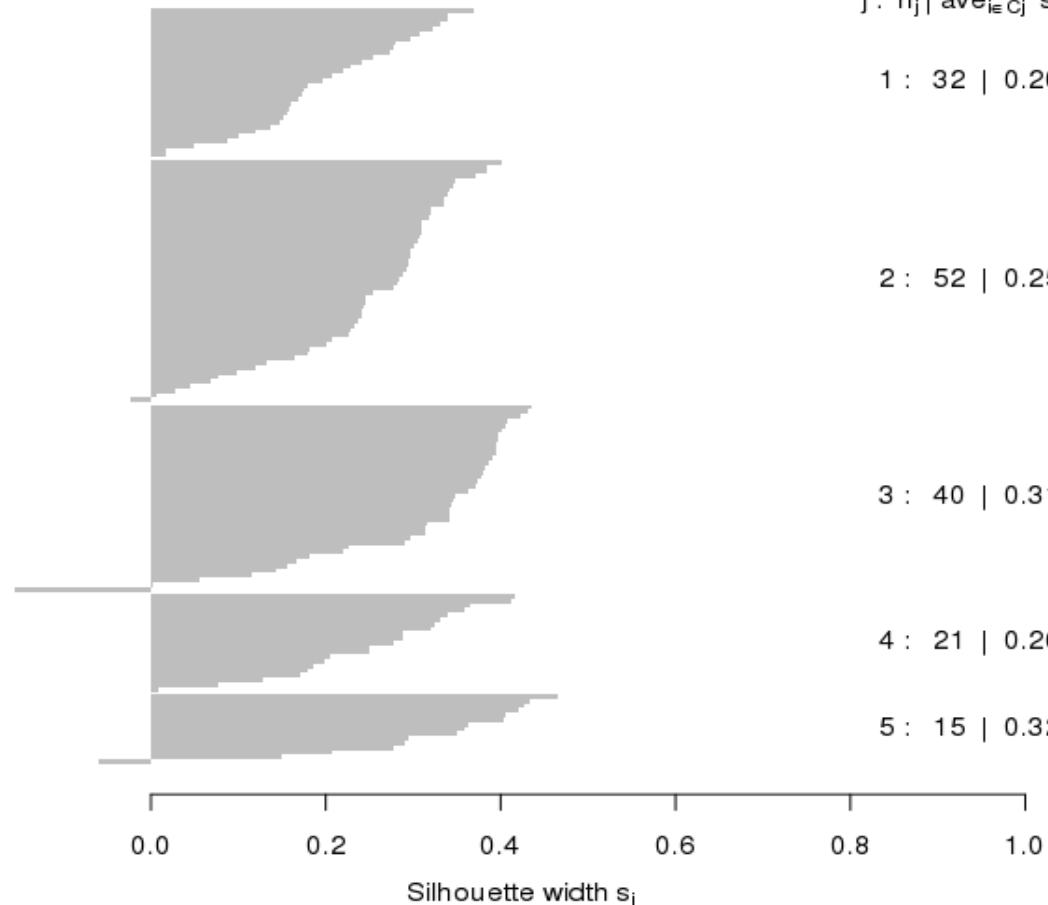
Internal Measures: Choosing k using the Average Silhouette Width
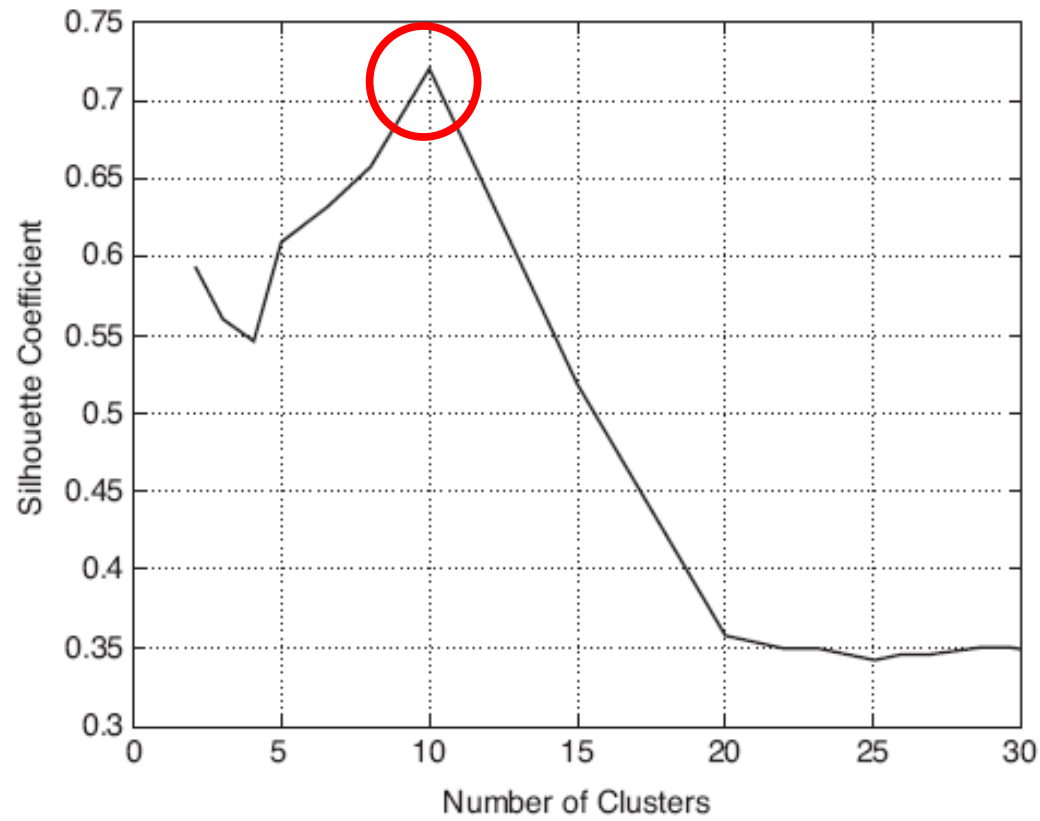
External Measures of Cluster Validity: Entropy and Purity

Table 5.9. 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 $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Other measures: Precision, Recall, F-measure, Rand, Adj. Rand

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