



CST8506 – Advanced Machine Learning

Week 6: Clustering

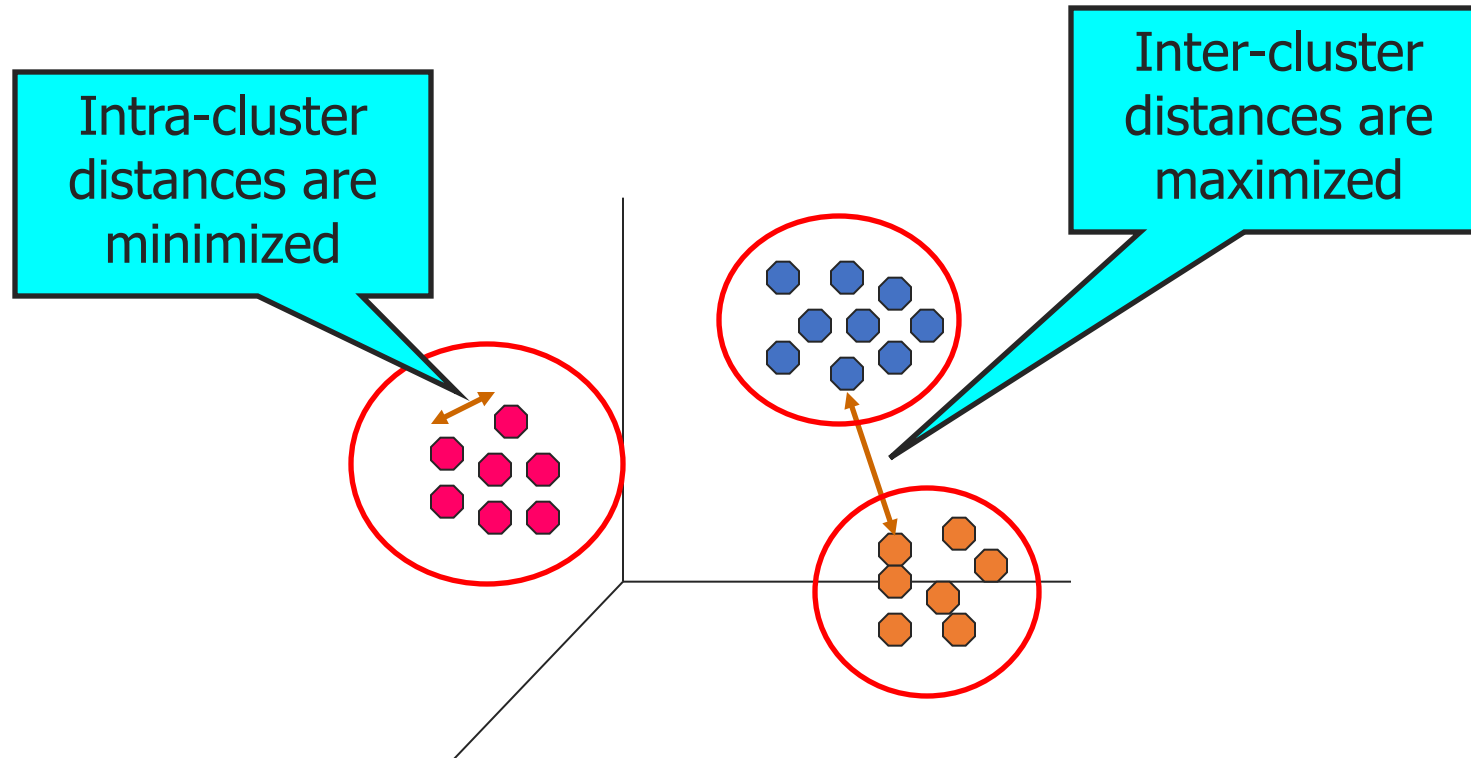
Dr. Abbas Akkasi
Winter 2026

- **Clustering**
- **Types of Clustering**
- **Types of clusters**
- **K-Means**
- **Hierarchical Clustering**
- **DBSCAN**
- **EM**
- **Cluster Validity**



What is Cluster Analysis?

Given a **set of objects**, place them in **groups** such that the objects in a group are similar (or related) to one another and different from (or unrelated to) the objects in other groups



Applications of Cluster Analysis

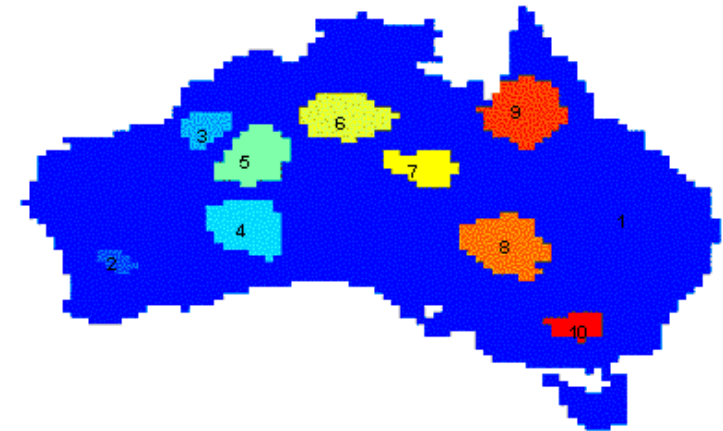
➤ Understanding

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

➤ Summarization

- Reduce the size of large data sets

	<i>Discovered Clusters</i>	<i>Industry Group</i>
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

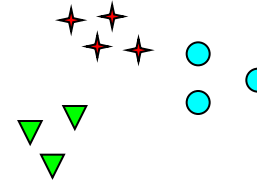


Clustering precipitation
in Australia

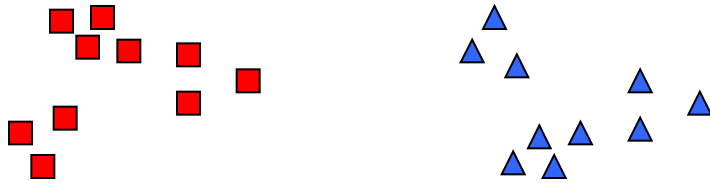
Notion of a Cluster can be Ambiguous



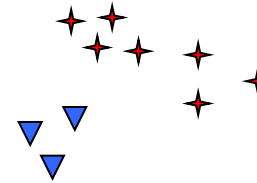
How many clusters?



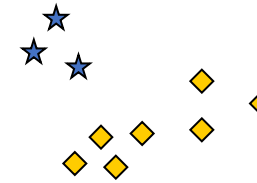
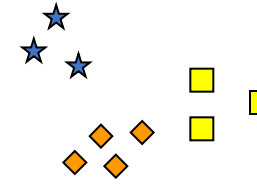
Six Clusters



Two Clusters



Four Clusters

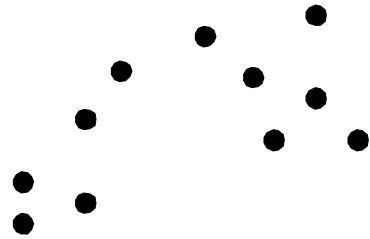


Types of Clusterings

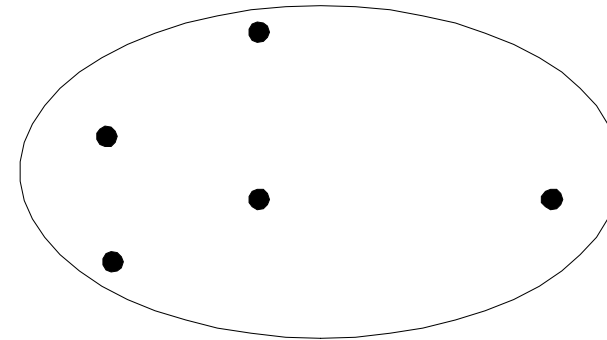
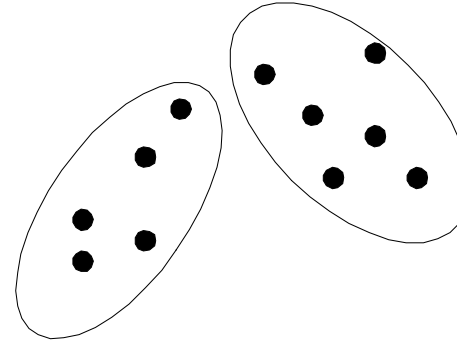
- A **clustering** is a set of clusters
- Important distinction between **hierarchical** and **partitional** sets of clusters
 - ❑ Partitional Clustering
 - A division of data objects into non-overlapping subsets (clusters)
 - ❑ Hierarchical clustering
 - A set of nested clusters organized as a hierarchical tree



Partitional Clustering

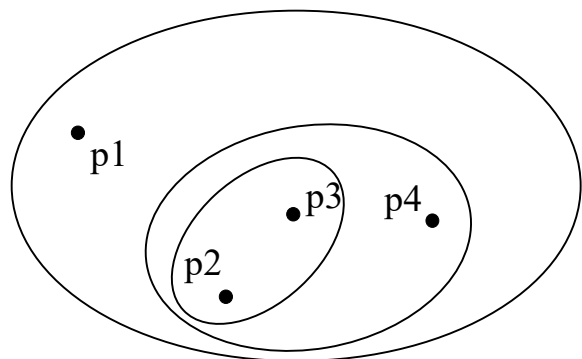


Original Points

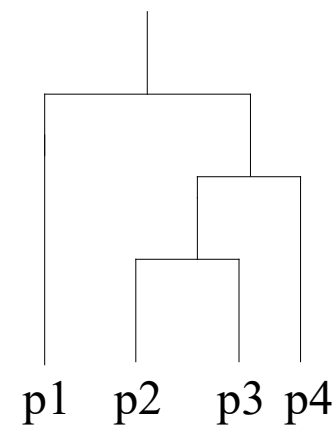


A Partitional Clustering

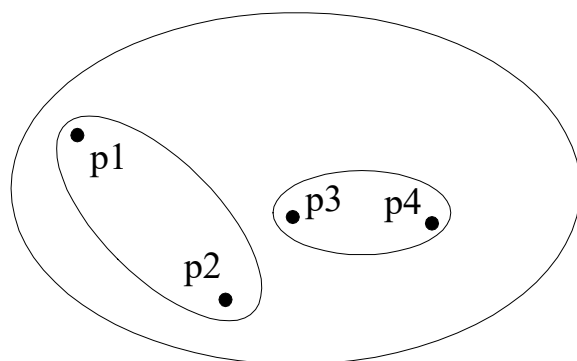
Hierarchical Clustering



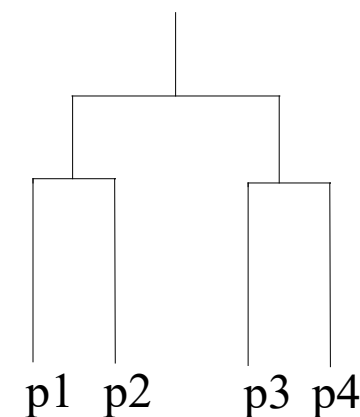
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



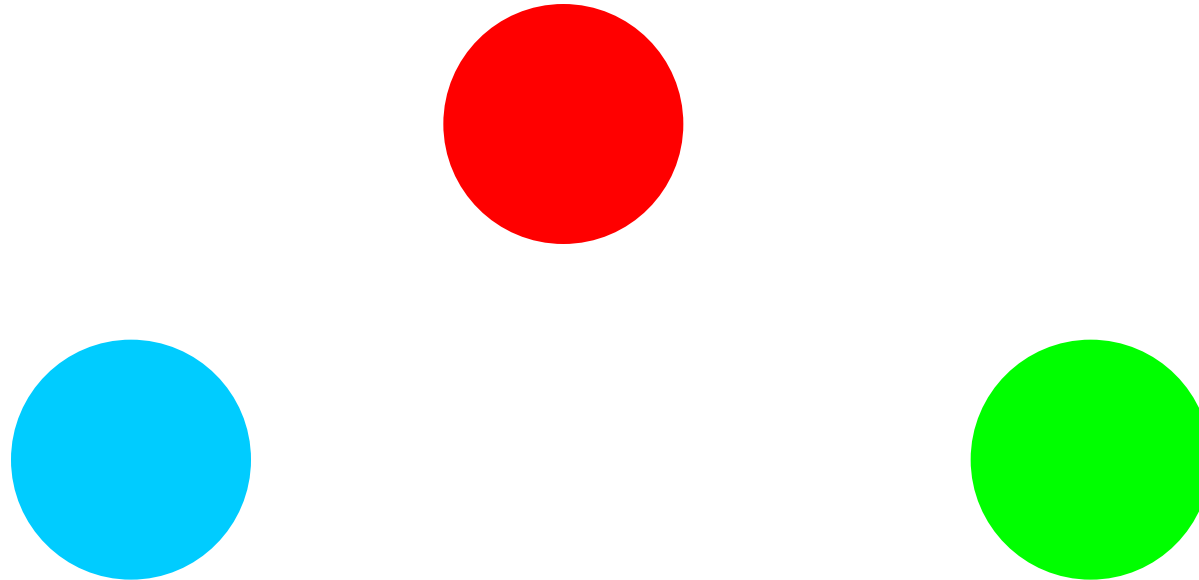
Non-traditional Dendrogram

Types of Clusters

- Well-separated clusters
- Prototype-based clusters
- Contiguity-based clusters
- Density-based clusters

Types of Clusters: Well-Separated

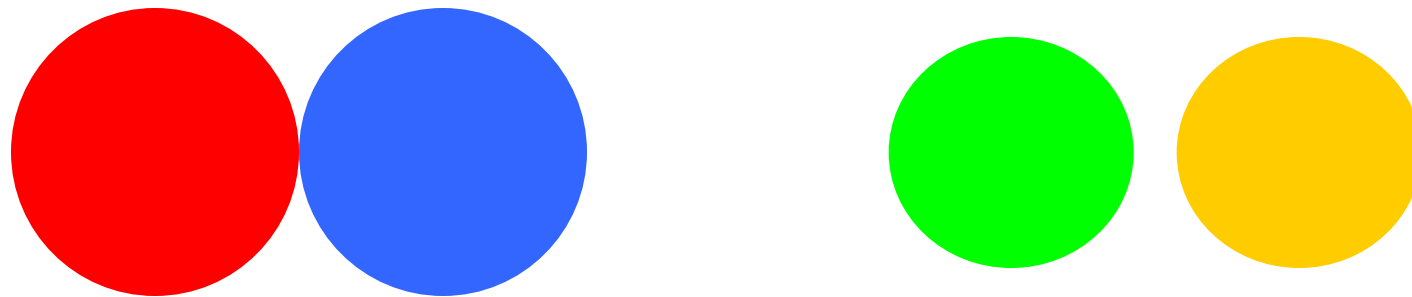
- **Well-Separated Clusters:**
 - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Prototype-Based

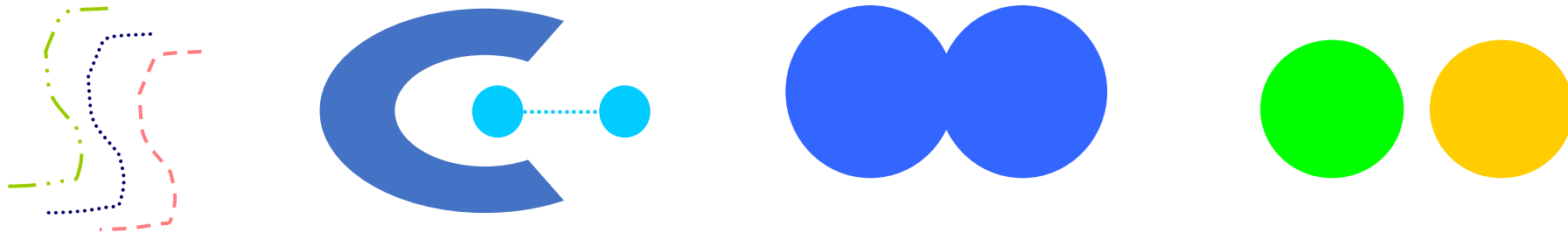
- Prototype-based
 - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the prototype or “center” of a cluster, than to the center of any other cluster
 - The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**,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



4 center-based clusters

Types of Clusters: Contiguity-Based

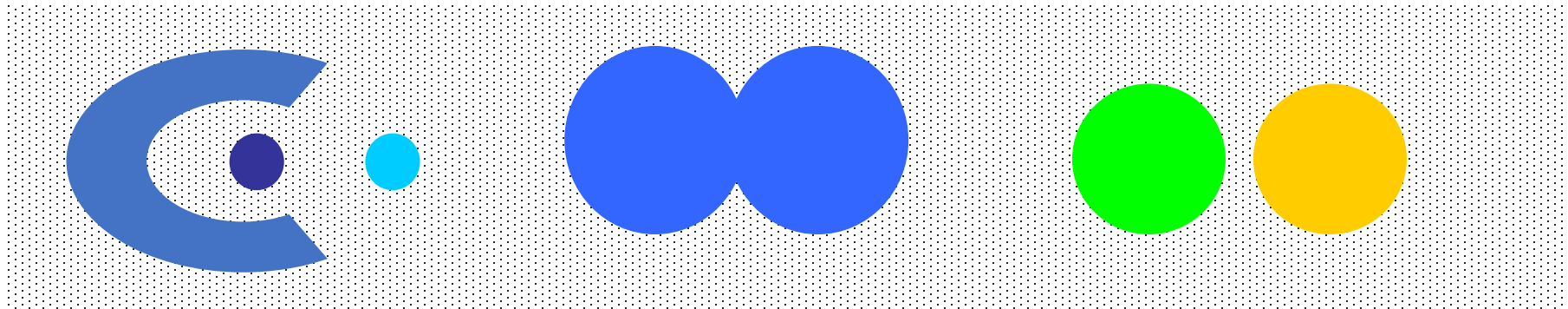
- Contiguous Cluster (Nearest neighbor or Transitive)
 - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point **not** in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

- Density-based
 - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
 - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



6 density-based clusters

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering
- Distribution based Method - EM

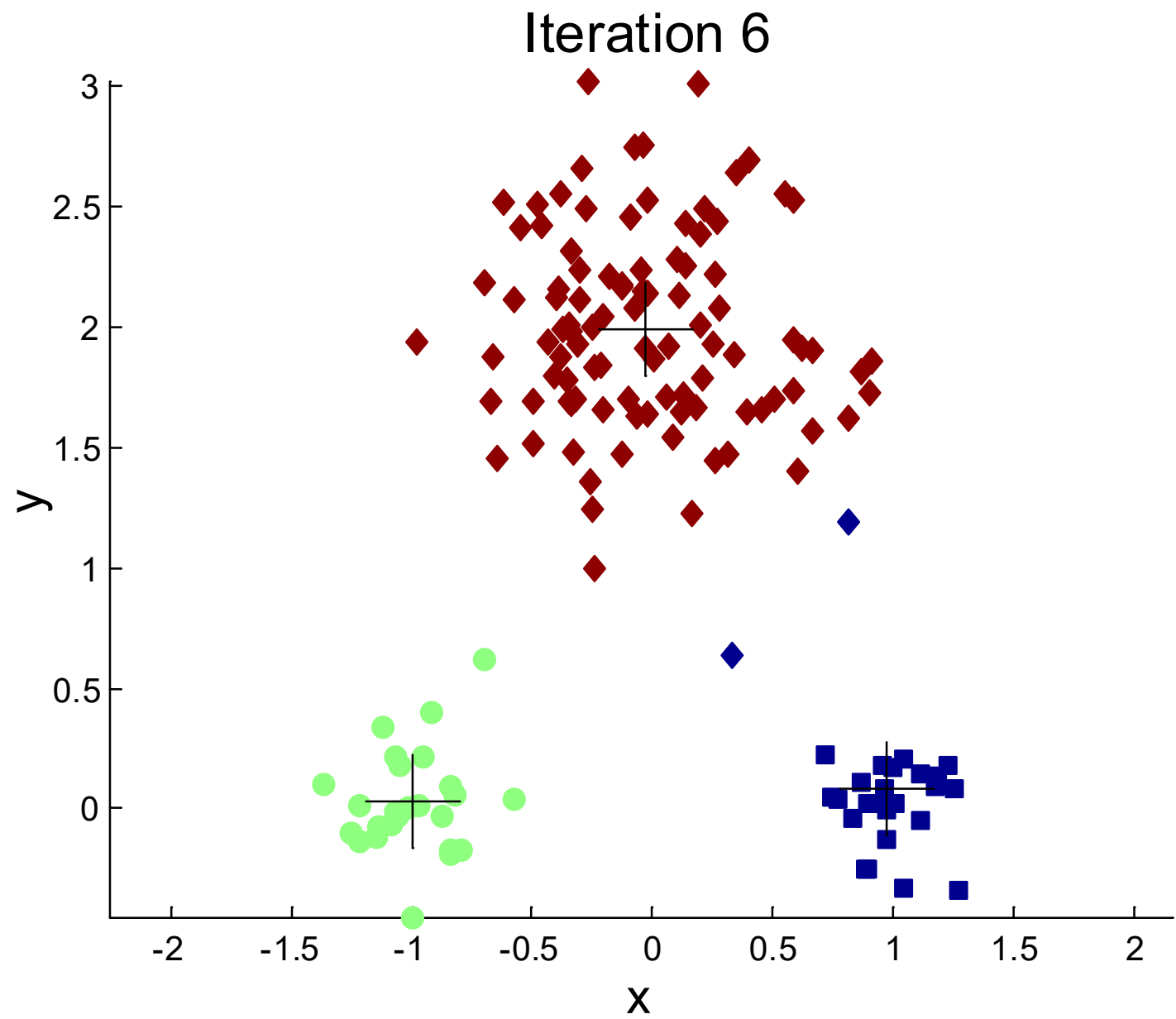


K-means Clustering

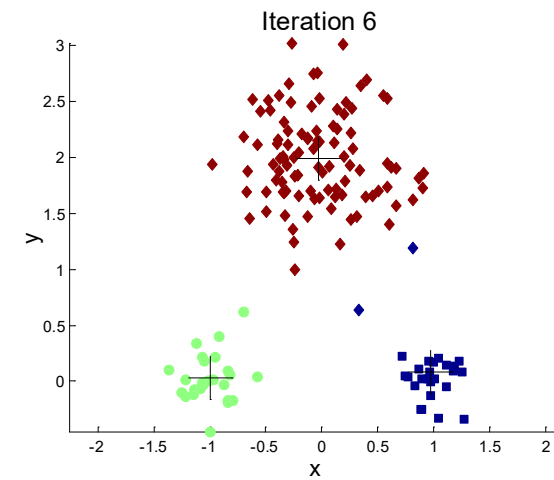
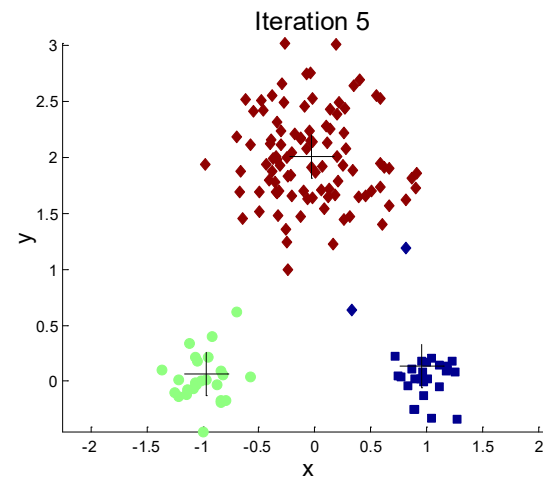
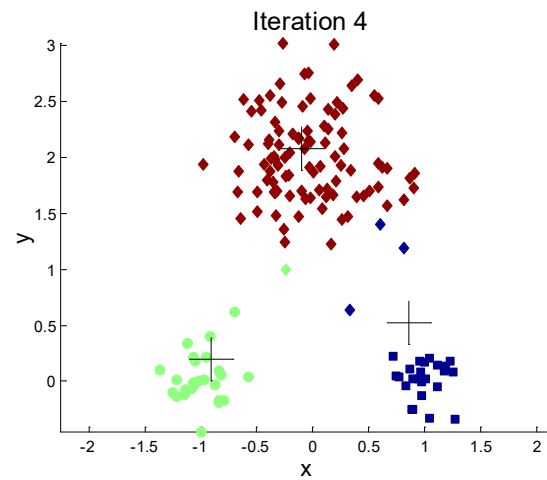
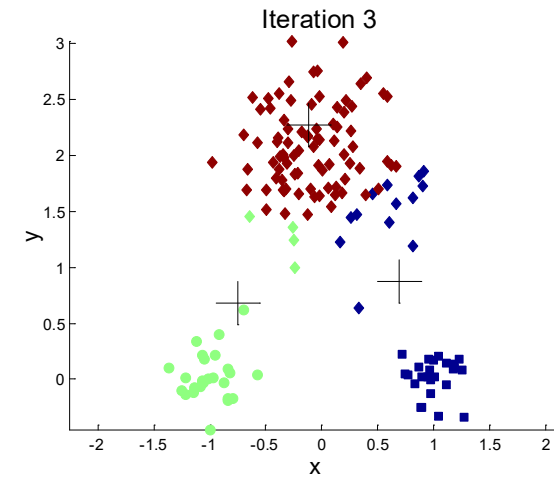
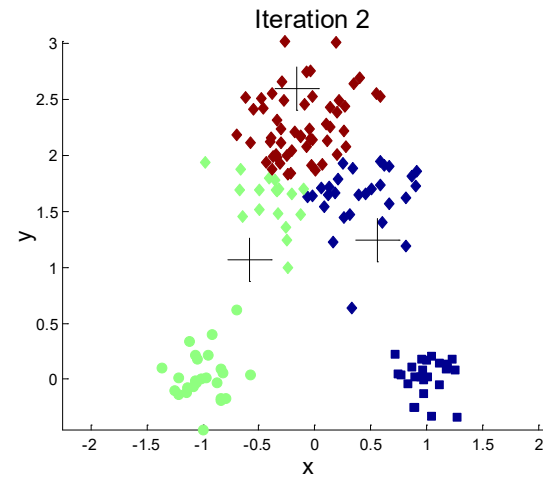
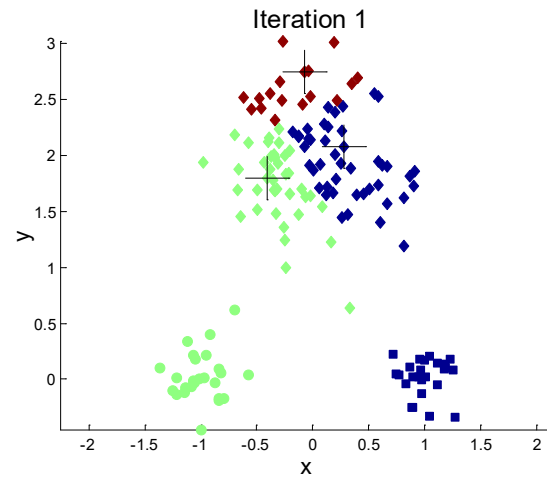
- Partitional clustering approach
- Number of clusters, K , must be specified
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

-
- 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
-

Example of K-means Clustering



Example of K-means Clustering



K-means Clustering – Details

➤ Simple iterative algorithm.

- Choose initial centroids;
- repeat {assign each point to a nearest centroid; re-compute cluster centroids}
- until centroids stop changing.

➤ Initial centroids are often chosen randomly.

- Clusters produced can vary from one run to another

➤ The centroid is (typically) the mean of the points in the cluster, but other definitions are possible.

➤ K-means will converge for common distance measures with appropriately defined centroid

➤ 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

- A common objective function (used with Euclidean distance measure) is Sum of Squared Error (SSE)
 - For each point, the error is the distance to the nearest cluster center
 - To get SSE, we square these errors and sum them.

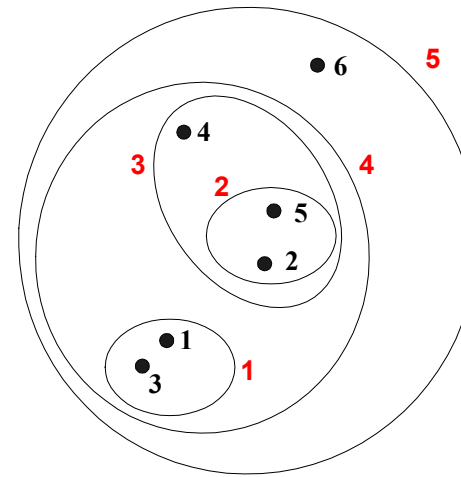
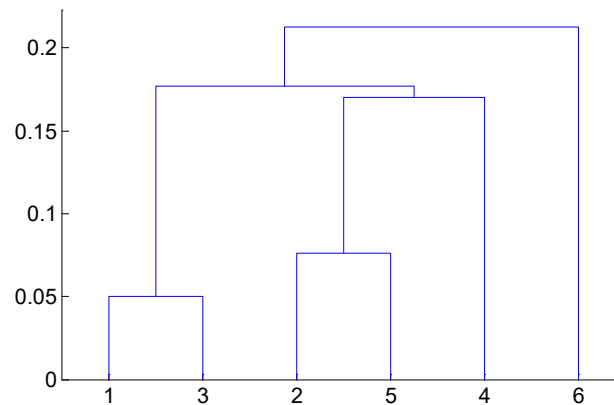
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the centroid (mean) for cluster C_i
- SSE improves in each iteration of K-means until it reaches a local or global minima.



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
- 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 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



Key Idea: Successively merge closest clusters

Basic algorithm

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

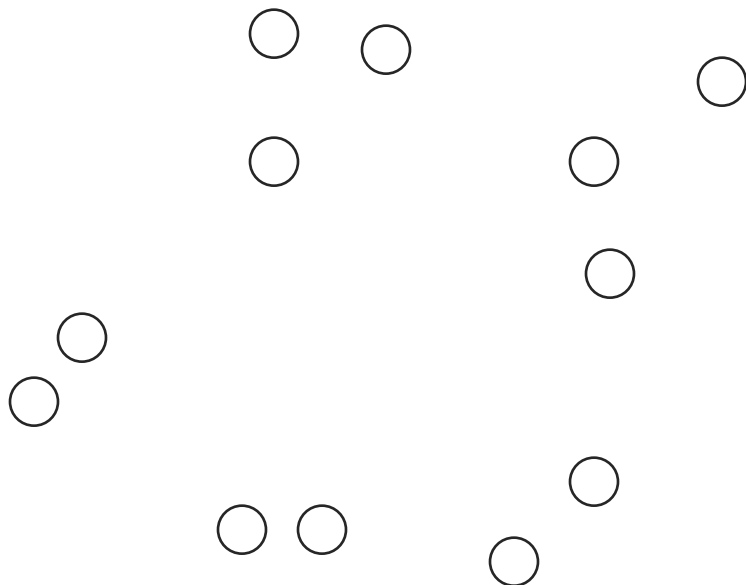
Key operation is the computation of the distance of two clusters

- Different approaches to defining the distance between clusters distinguish the different algorithms



Steps 1 and 2

- Start with clusters of individual points and a proximity matrix



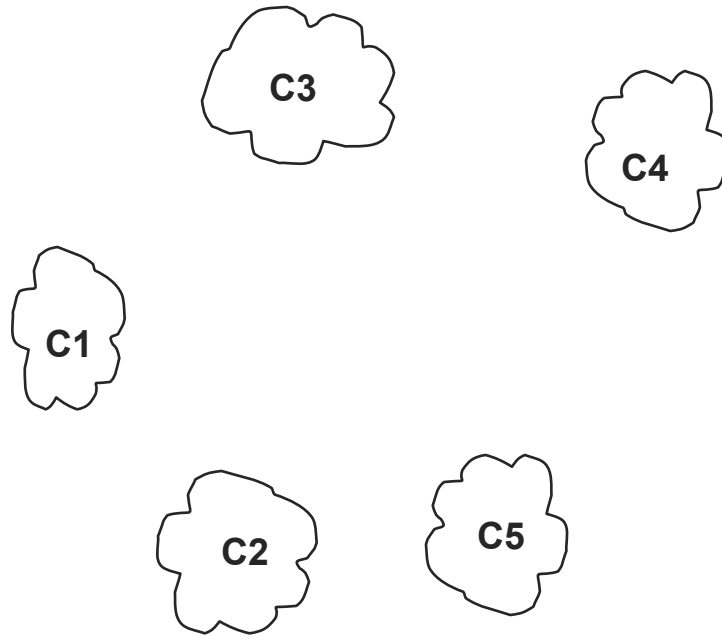
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix



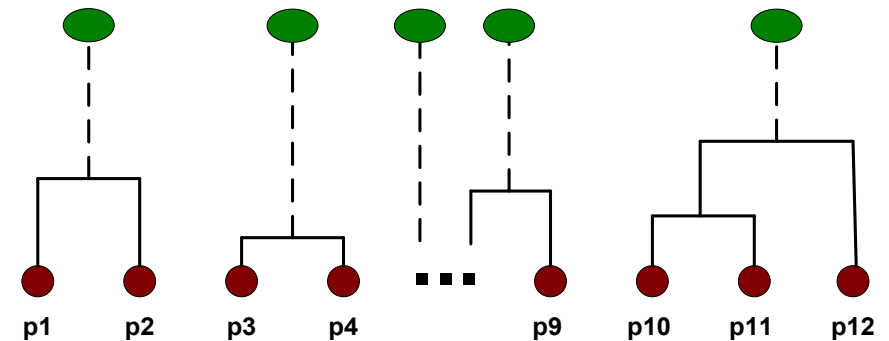
Intermediate Situation

- After some merging steps, we have some clusters



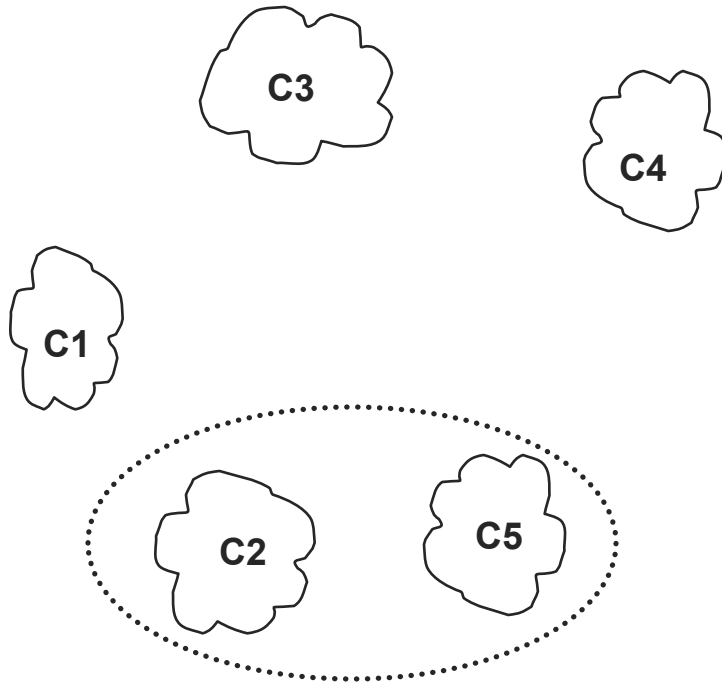
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix



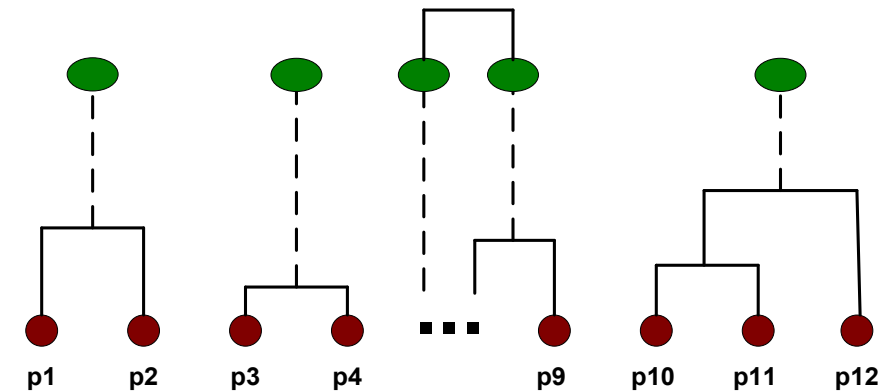
Step 4

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



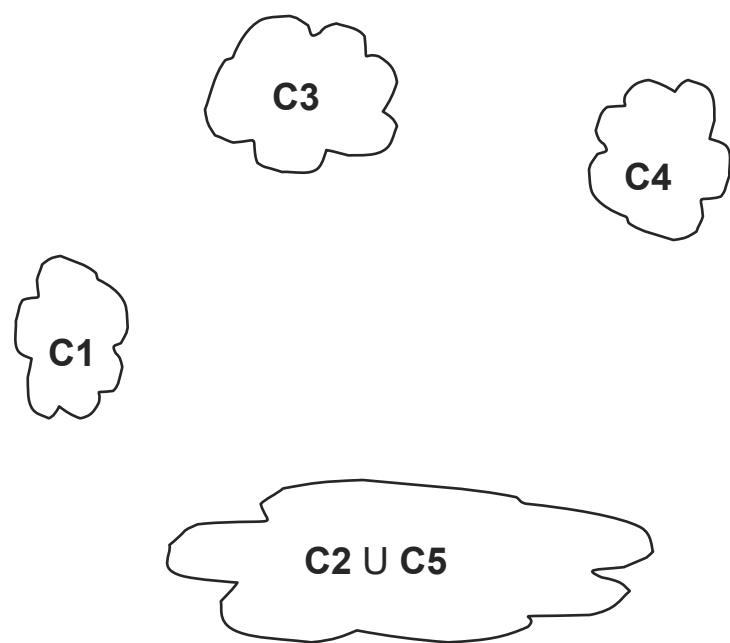
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix



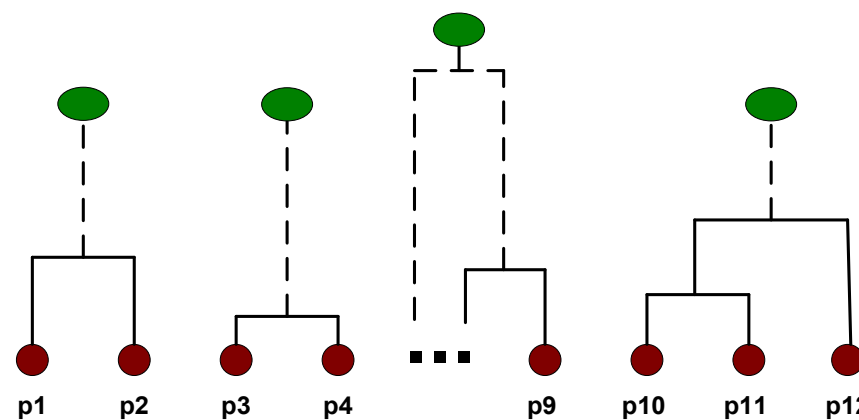
Step 5

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

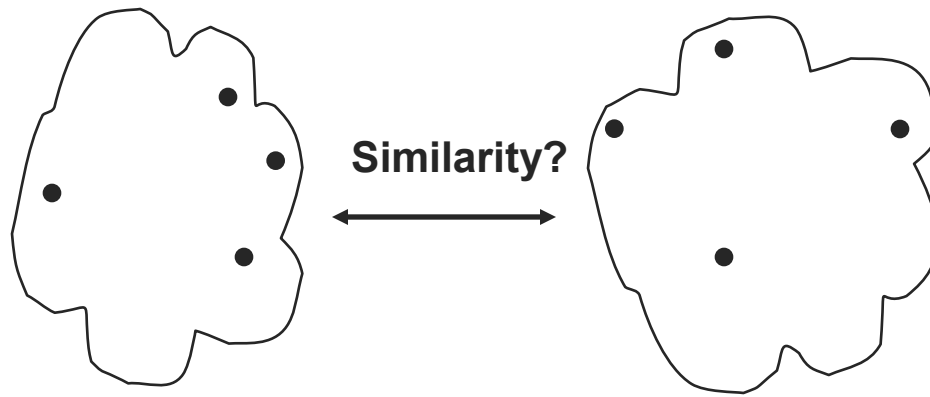


		C2 U C5		
	C1	C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Distance Matrix



How to Define Inter-Cluster Distance

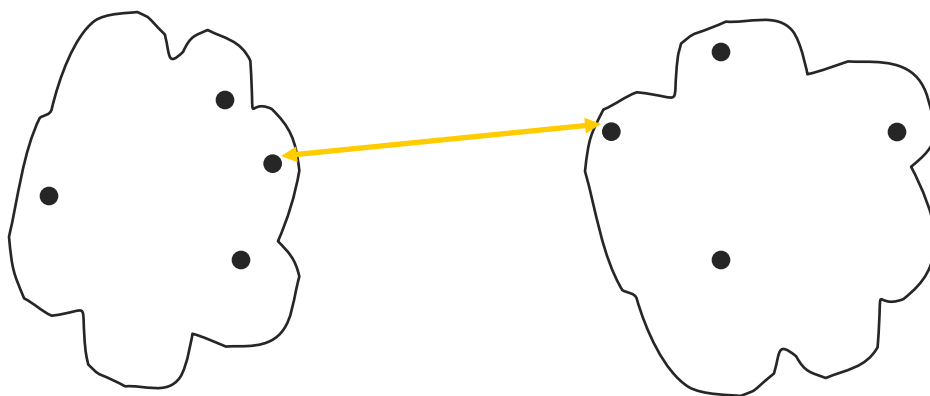


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Distance Matrix

How to Define Inter-Cluster Similarity

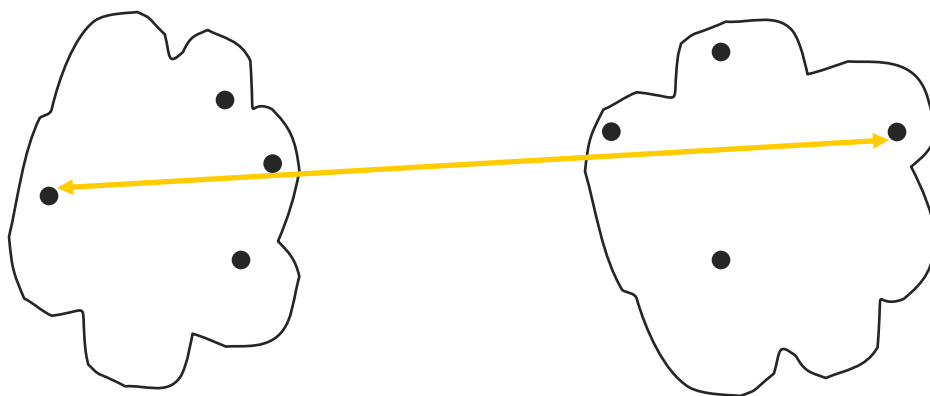


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Desnity Matrix

How to Define Inter-Cluster Similarity

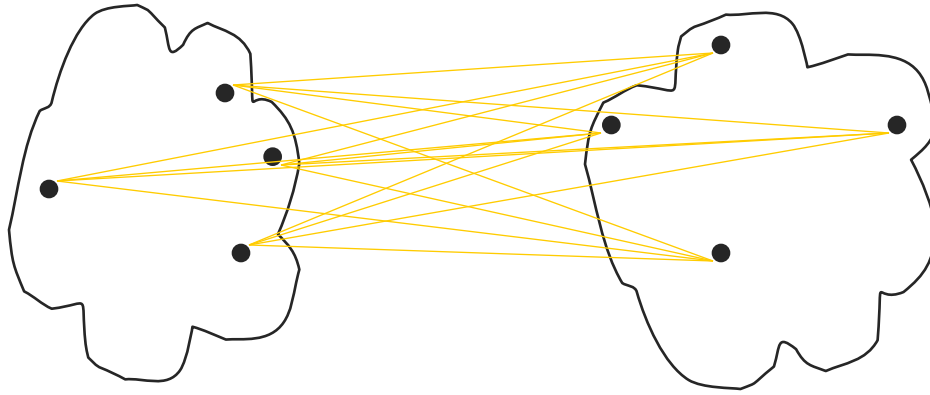


- MIN
- **MAX**
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Density Matrix

How to Define Inter-Cluster Similarity

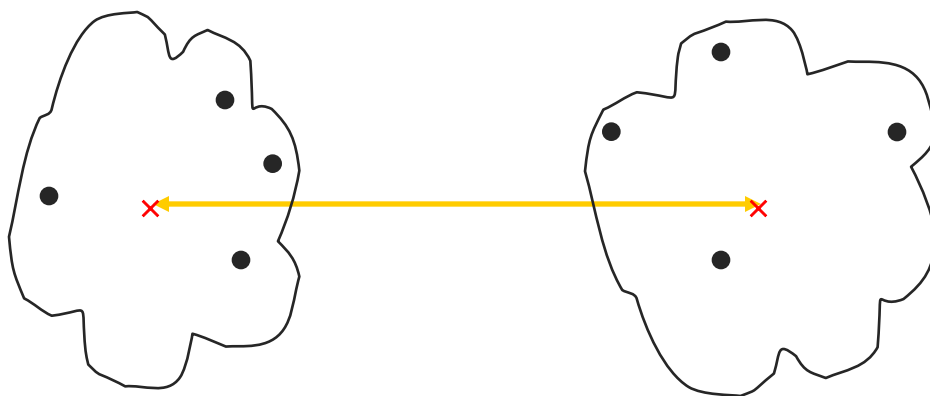


- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Density Matrix

How to Define Inter-Cluster Similarity

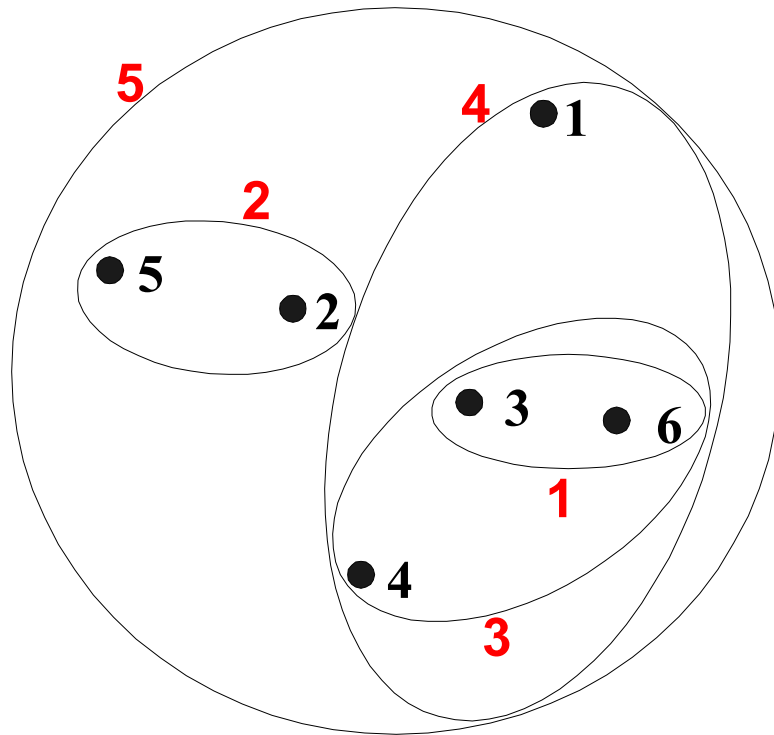


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

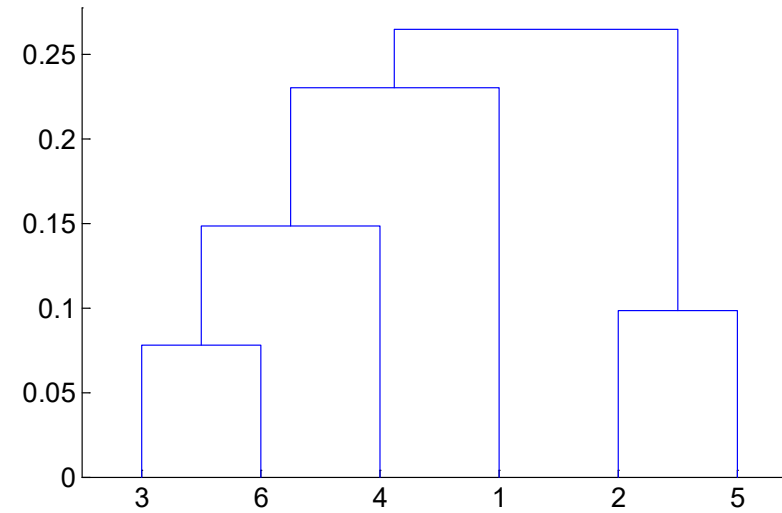
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						

· Density Matrix

Hierarchical Clustering: Group Average

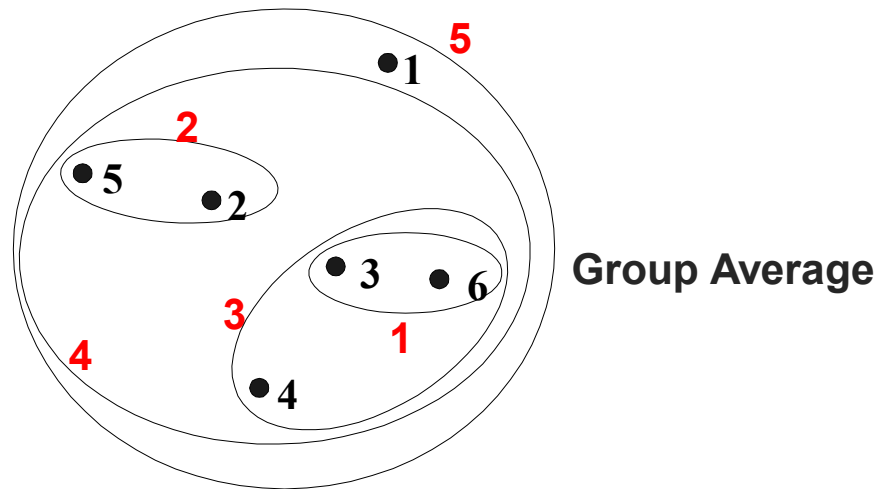
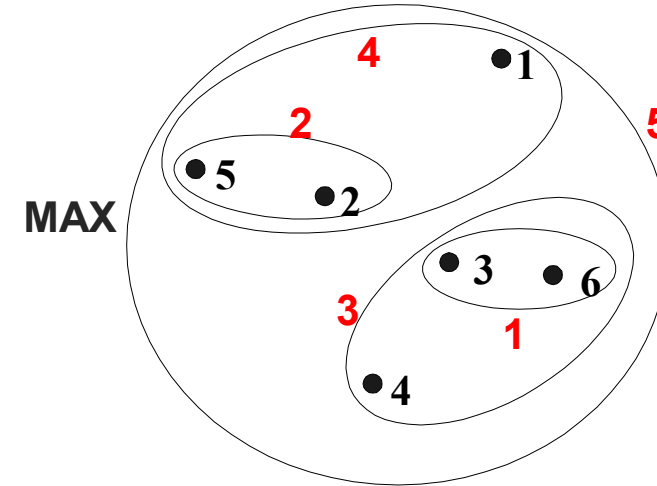
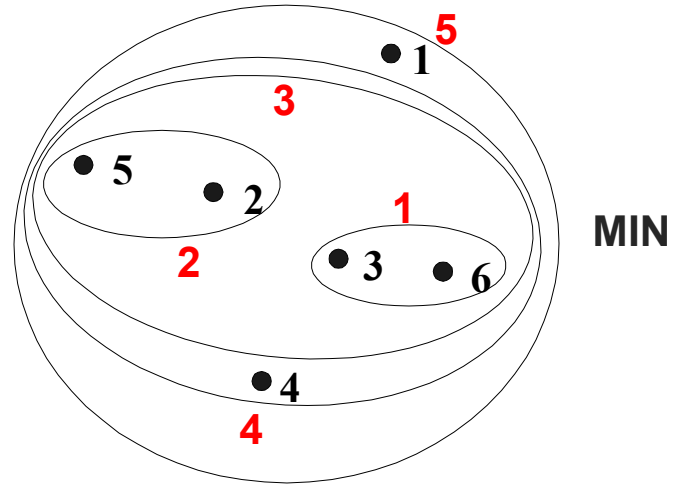


Nested Clusters



Dendrogram

Hierarchical Clustering: Comparison



Density Based Clustering

Clusters are **regions of high density** that are separated from one another by **regions of low density**.

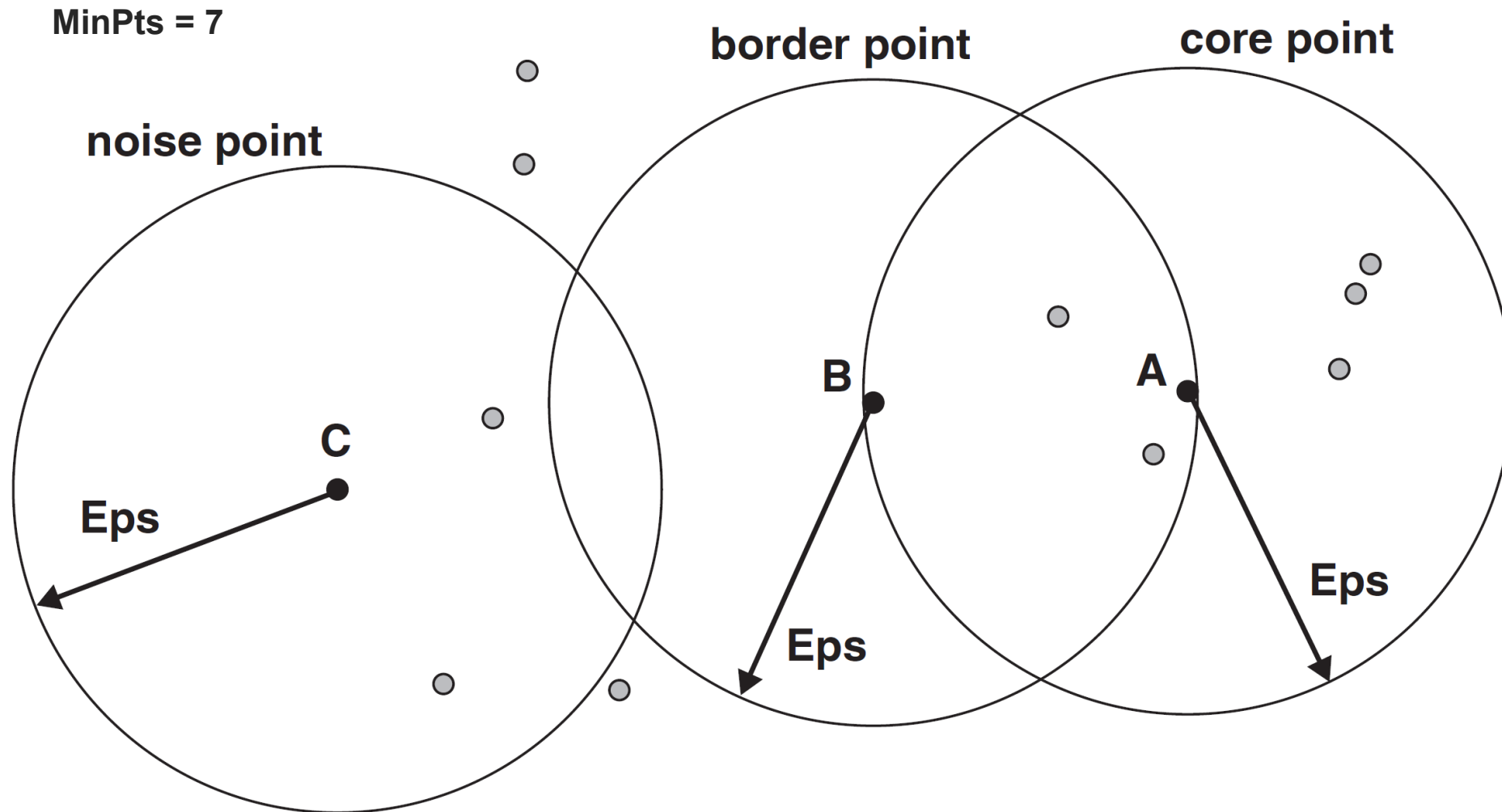


DBSCAN - Density-based spatial clustering of applications with noise

- DBSCAN is a density-based algorithm.
 - Density = number of points within a specified radius (ϵ)
 - A point is a **core point** if it has **at least** a specified number of points (MinPts) within ϵ
 - These are points that are at the interior of a cluster
 - Counts the point itself
 - A **border point** is not a core point, but is in the neighborhood of a core point
 - A **noise point** is any point that is not a core point or a border point



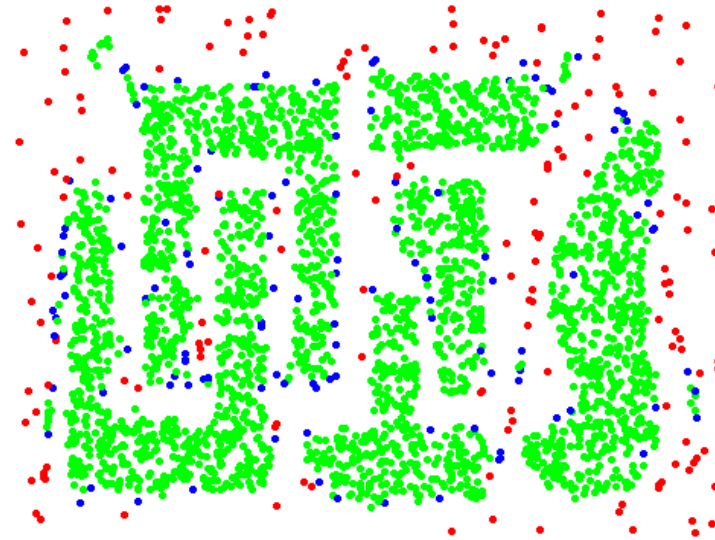
DBSCAN: Core, Border, and Noise Points



DBSCAN: Core, Border and Noise Points



Original Points



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

Eps = 10, MinPts = 4

DBSCAN Algorithm

Form clusters using core points, and assign border points to one of its neighboring clusters

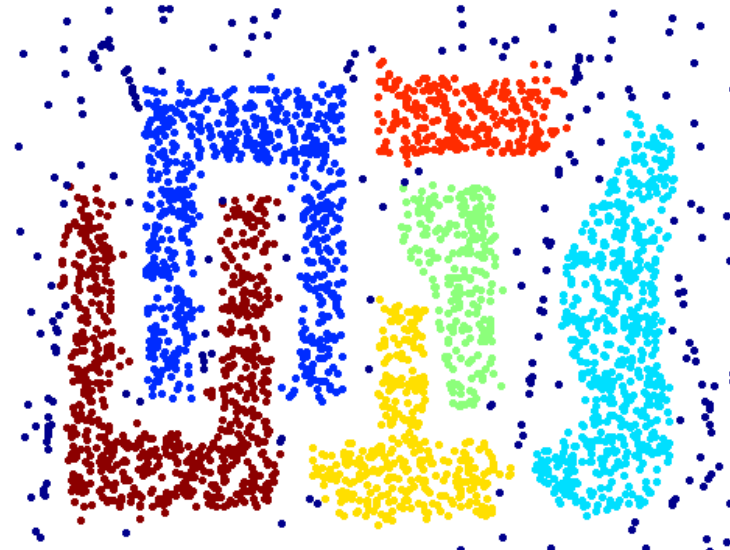
- 1: Label all points as core, border, or noise points.
- 2: Eliminate noise points.
- 3: Put an edge between all core points within a distance ϵ of each other.
- 4: Make each group of connected core points into a separate cluster.
- 5: Assign each border point to one of the clusters of its associated core points



When DBSCAN Works Well



Original Points



Clusters (dark blue points indicate noise)

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

Distribution-based Clustering

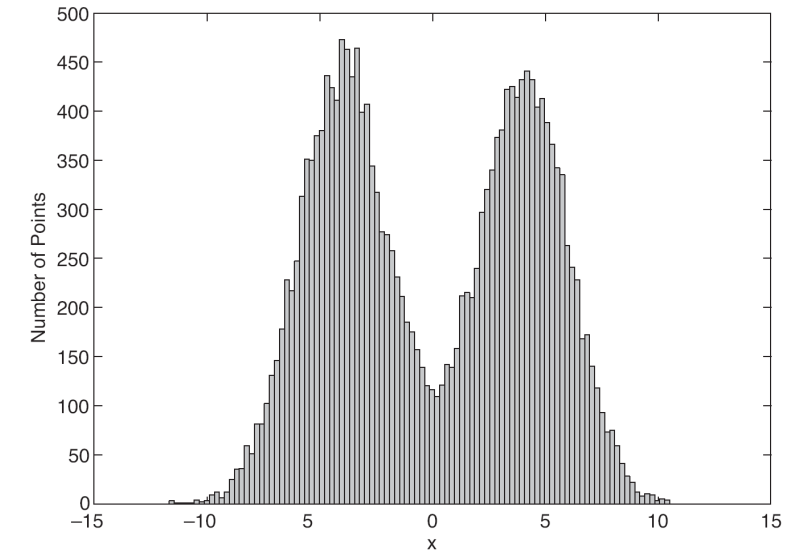
- Idea is to model the set of data points as arising from a mixture of distributions
 - Typically, normal (Gaussian) distribution is used
 - But other distributions have been very profitably used
- Clusters are found by estimating the parameters of the statistical distributions using the Expectation-Maximization (EM) algorithm



Distribution-based Clustering: Example

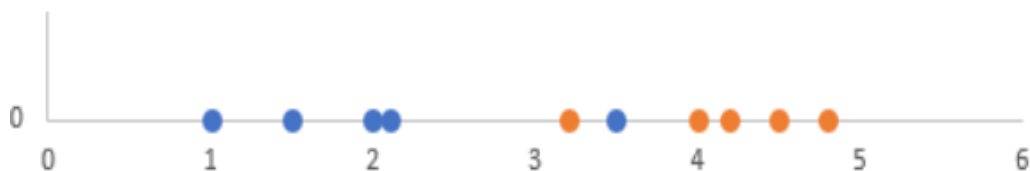
Informal example: consider modeling the points that generate the following histogram.

- Looks like a combination of two normal (Gaussian) distributions
- Suppose we can estimate the mean and standard deviation of each normal distribution.
 - This completely describes the two clusters
 - We can compute the probabilities with which each point belongs to each cluster
 - Can assign each point to the cluster (distribution) for which it is most probable.



$$prob(x_i|\Theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

EM ...



If we know the source of data, it is easy to estimate parameters.

If we know the parameters, we could easily assign each point to the closest distribution.

If we do not know the source and the parameters, then ???

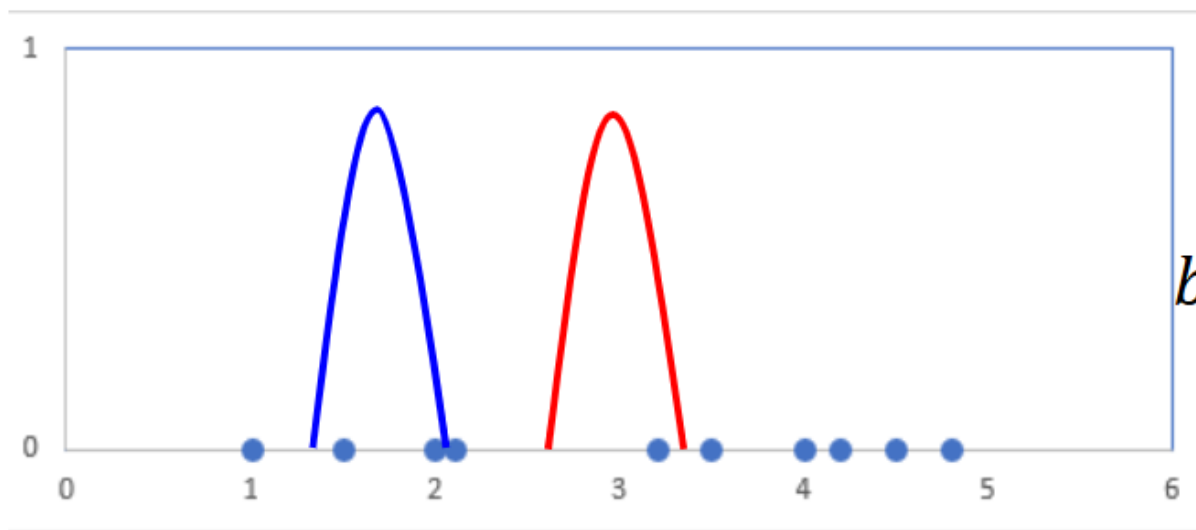
EM algorithm

- Starts with k randomly placed Gaussians (in this example, $k = 2$) $(\mu_a, \sigma_a^2), (\mu_b, \sigma_b^2)$
- For each point x_i , $P(b|x_i) \leftarrow$ does it look like it belongs to blue group (E-step)
 - Computes the probability to be in blue cluster or orange cluster
 - So, soft clustering
- Calculate new $(\mu_a, \sigma_a^2), (\mu_b, \sigma_b^2)$ to fit points assigned to them (M-step)



More Detailed EM Algorithm

Assumption: $k=2$ (2 Gaussians)



$$P(x_i|b) = \frac{1}{\sqrt{2\pi} \sigma_b} e^{-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}}$$

$$b_i = P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b) + P(x_i|a)P(a)}$$

$$\mu_b = \frac{b_1 x_1 + b_2 x_2 + \dots + b_n x_n}{b_1 + b_2 + \dots + b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1 - \mu_b)^2 + b_2(x_2 - \mu_b)^2 + \dots + b_n(x_n - \mu_b)^2}{b_1 + b_2 + \dots + b_n}$$

Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following two types.
 - **Supervised:** Used to measure the extent to which cluster labels match externally supplied class labels.
 - Often called **external indices** because they use information external to the data
 - **Unsupervised:** Used to measure the goodness of a clustering structure *without* respect to external information.
 - Sum of Squared Error (SSE)
 - Often called **internal indices** because they only use information in the data
- You can use supervised or unsupervised measures to compare clusterings methods



Unsupervised 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)

- Separation is measured by the between cluster sum of squares

Where $|C_i|$ is the size of cluster i , and m is the

global (grand) mean of *all data points*,

before clustering.

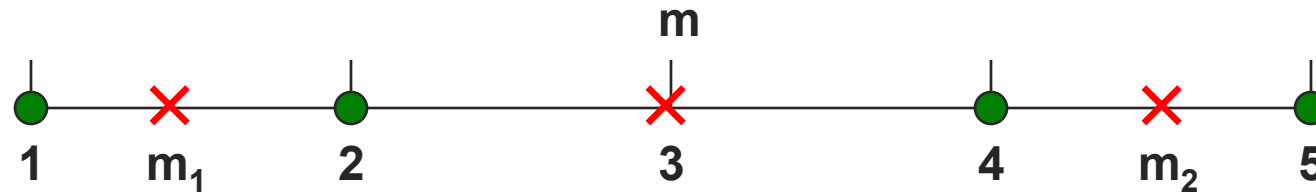
$$SSE = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

$$SSB = \sum_i |C_i| (m - m_i)^2$$

Unsupervised Measures: Cohesion and Separation

Example: SSE

$SSB + SSE = \text{constant}$



K=1 cluster:

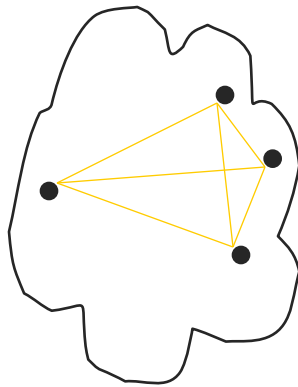
$$SSE = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$$
$$SSB = 4 \times (3 - 3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K=2 clusters:

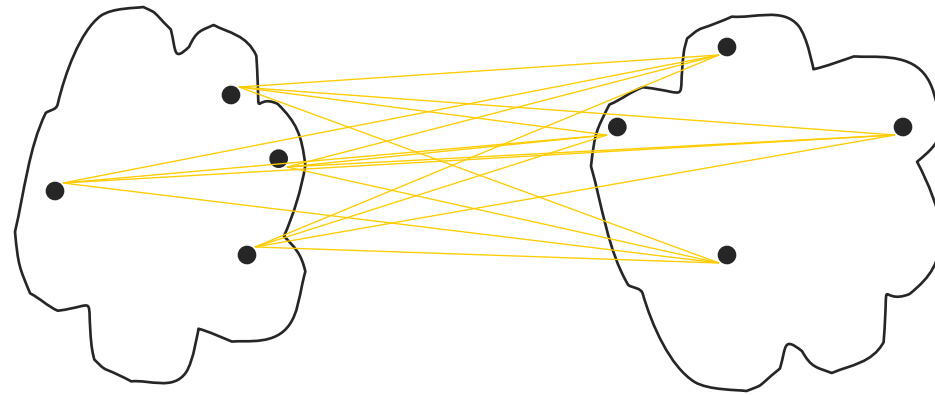
$$SSE = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1$$
$$SSB = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9$$
$$Total = 1 + 9 = 10$$

Unsupervised Measures: Cohesion and Separation

- A distance 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



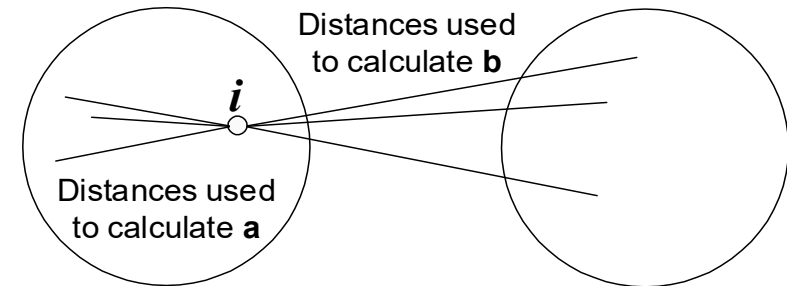
separation

Unsupervised Measures: Silhouette Coefficient

- **Silhouette coefficient** combines ideas of both cohesion and separation, but for individual points, as well as clusters
- 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

$$s = (b - a) / \max(a, b)$$

- Value can vary between -1 and 1
- Typically ranges between 0 and 1.
- The closer to 1 the better.



The **silhouette coefficient** tells you **how well each point fits in its cluster**.

