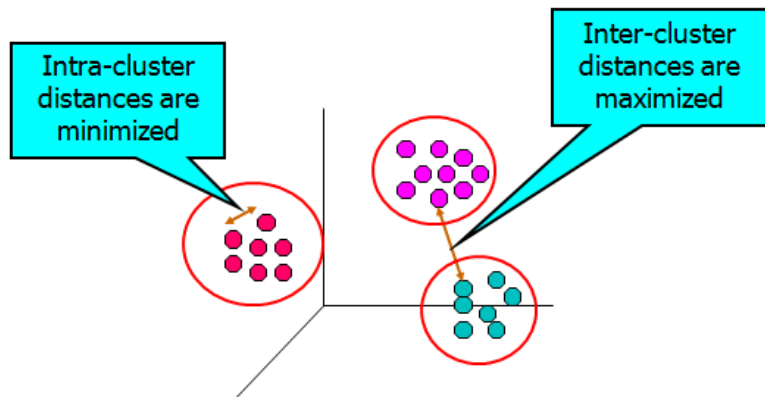


UNIT – 5

Cluster Analysis

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



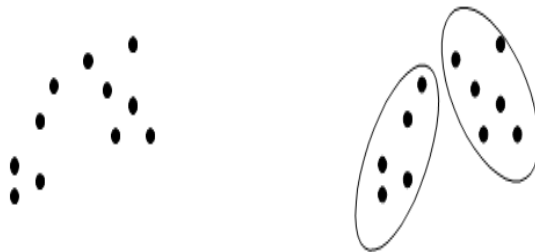
Types of Clusterings

An entire collection of clusters is commonly referred to as a clustering Types:

- Hierarchical versus Partitional
- Exclusive versus Overlapping versus Fuzzy
- Complete versus Partial

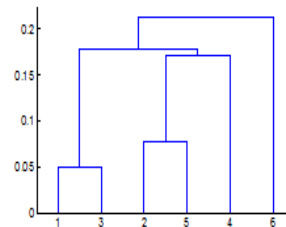
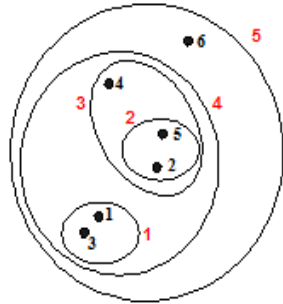
Partitional Clustering

It is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset



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
- Each node (cluster) in the tree (except for the leaf nodes) is the union of its children(subclusters), and the root of the tree is the cluster containing all the objects.
- Often, but not always, the leaves of the tree are singleton clusters of individual data objects.



- **Exclusive versus overlapping (or non-exclusive)**
 - In Exclusive clustering, assign each object to a single cluster
 - In non-exclusive clusterings, an object can simultaneously belong to more than one group (class)
 - A person at a university can be both an enrolled student and an employee of the university or 'border' points
- **Fuzzy clustering**
 - In a fuzzy clustering, every object belongs to every cluster with a membership weight that is between 0 (absolutely doesn't belong) and 1 (absolutely belongs).
 - In other words, clusters are treated as fuzzy sets.
 - In fuzzy clustering, we often impose the additional constraint that the sum of the weights for each object must equal 1
- **Complete versus Partial**
 - A complete clustering assigns every object to a cluster
 - In Partial clustering, some objects in a data set may not belong to well-defined groups
 - Many times objects in the data set may represent noise, outliers, or uninteresting background

Types of Clusters

- Well-separated clusters
- Prototype-Based clusters
- Graph-Based clusters
- Density-based clusters
- Shared-Property (Conceptual Clusters)

Well-Separated Clusters:

A cluster is a set of objects in which each object is closer (or more similar) to every other object in the cluster than to any object not in the cluster or

The distance between any two points in different groups is larger than the distance between any two points within a group.

Well-separated clusters do not need to be globular, but can have any shape



(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

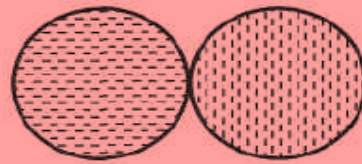
Prototype-Based:

Cluster is a set of objects in which each object is closer (more similar) to the prototype that defines the cluster than to the prototype of any other cluster.

For data with continuous attributes, the prototype of a cluster is often a centroid, i.e., the average (mean) of all the points in the cluster.

When the data has categorical attributes, the prototype is often a medoid, i.e., the most representative point of a cluster

PB clusters are commonly referred as center-based clusters, such clusters tend to be globular



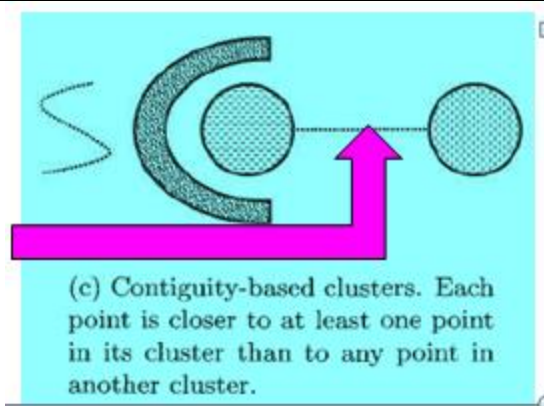
(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

Graph-Based: If the data is represented as a graph, where the nodes are objects and the links represent connections among objects then a cluster can be defined as a **connected component**.

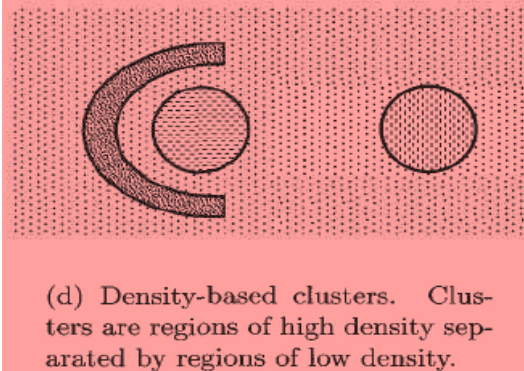
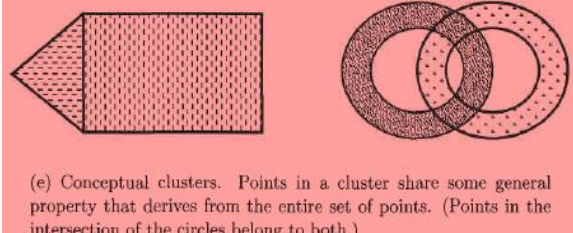
i.e., a group of objects that are connected to one another, but that have no connection to objects outside the group.

An example of graph-based clusters are contiguity-based clusters, where two objects are connected only if they are within a specified distance of each other.

Clusters are irregular, but can have trouble when noise is present.



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

<p>Density-based</p> <p>A cluster is a dense region of objects, which is separated by low-density regions, from other regions of high density.</p> <p>Used when the clusters are irregular and when noise and outliers are present.</p>	 <p>(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.</p>
<p>Shared-Property (Conceptual Clusters)</p> <p>Finds clusters that share some common property or represent a particular concept. A triangular area (cluster) is adjacent to a rectangular one, and there are two intertwined (connect or link closely) circles (clusters)</p>	 <p>(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.)</p>

Data Objects and Attribute Types

Data Objects

- Data sets are made up of data objects.
- A data object represents an entity.
- Examples:
 - sales database: customers, store items, sales
 - medical database: patients, treatments
 - university database: students, professors, courses
- Also called samples , examples, instances, data points, objects, tuples.
- Data objects are described by attributes.

What Is an Attribute?

- An attribute is a data field, representing a characteristic or feature of a data object.
 - The nouns attribute, dimension, feature, and variable are often used interchangeably in the literature.
 - The term dimension is commonly used in data warehousing.
 - Machine learning literature tends to use the term feature, while
 - Statisticians prefer the term variable.
 - Data mining and database professionals commonly use the term attribute
- Ex: : Attributes describing a customer object can include, customer ID, name, and address.

Attribute Types

- The type of an attribute is determined by the set of possible values
 - Nominal Attributes
 - Binary Attributes
 - Ordinal Attributes
 - Numeric: quantitative
 - Interval-scaled
 - Ratio-scaled

1. Nominal attributes

- The values of a nominal attribute are symbols or names of things.
- Each value represents some kind of category, code, or state etc.
 - Hair_color = {black, brown, grey, red, white}
 - marital status= {single, married, divorced, and widowed}
- Nominal attributes are also referred to as categorical.
- The values do not have any meaningful order.

2. Ordinal Attributes

An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them.

Size = {small, medium, large},

Grades={A+, A, A-, B+, and so on}

Professional ranks can be enumerated in a sequential order:

Eg.: assistant, associate, and full for professors

In survey, Customer satisfaction had the following ordinal categories:

0: very dissatisfied, 1: somewhat dissatisfied,

2: neutral, 3: satisfied, and 4: very satisfied.

3. Binary Attributes

- Nominal attribute with only 2 states (0 and 1)
- Symmetric binary: both outcomes equally important
 - ◆ e.g., gender
- Asymmetric binary: outcomes not equally important.
 - ◆ e.g., medical test (positive vs. negative)
 - ◆ Convention: assign 1 to most important outcome (e.g., HIV positive)

Note that nominal, binary, and ordinal attributes are qualitative. That is, they describe a feature of an object without giving an actual size or quantity

4. Numeric Attribute

Quantity (integer or real-valued)

Interval

- ◆ Measured on a scale of **equal-sized units**
- ◆ Values have order
 - E.g., *temperature in C° or F°, calendar dates*
- ◆ No true zero-point

Ratio

- ◆ Inherent **zero-point**

- ◆ We can speak of a value as being a multiple (or ratio) of another value.
- ◆ (10 K° is twice as high as 5 K°).
 - e.g., *temperature in Kelvin, length, counts, monetary quantities*

5. Discrete vs. Continuous Attributes

Discrete Attribute

- Has only a finite or countably infinite set of values
 - ◆ E.g., zip codes, profession, or the set of words in a collection of documents
- Sometimes, represented as integer variables
- **Continuous Attribute**
- Has real numbers as attribute values
 - ◆ E.g., temperature, height, or weight
- Continuous attributes are typically represented as floating-point variables

Proximity measures

Measuring Data Similarity and Dissimilarity

A cluster is a collection of data objects such that the objects within a cluster are similar to one another and dissimilar to the objects in other clusters.

- The higher the similarity value, the greater the similarity between objects
 - The higher the dissimilarity value, the more dissimilar the two objects are.
 - Distances are normally used to measure the similarity or dissimilarity between two data objects
 - Proximity refers to a similarity or dissimilarity
- **Data matrix (or object-by-attribute structure):**
 - This structure stores the n data objects in the form of a relational table, or n -by- p matrix (n objects \times p attributes):
 - **Dissimilarity matrix (or object-by-object structure):**
 - This structure stores a collection of proximities that are available for all pairs of n objects.
 - It is often represented by an n -by- n table:
 - where $\mathbf{d(i, j)}$ is the measured dissimilarity or “difference” between objects i and j .
 - $\text{Sim}(i, j) = 1 - d(i, j)$
 - Measures of similarity can often be expressed as a function of measures of dissimilarity

$$\begin{bmatrix} 0 \\ d(2,1) & 0 \\ d(3,1) & d(3,2) & 0 \\ \vdots & \vdots & \vdots \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Proximity Measure for Nominal Attributes:

- A nominal attribute can take on two or more states

Ex.: Map color is a nominal attribute that may have, say, five states: red, yellow, green, pink, and blue.

- Simple matching method:

The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p}$$

- where m is the number of matches (i.e., the number of attributes for which i and j are in the same state), and
- p is the total number of attributes describing the objects.

Example:

Table	A Sample Data Table											
	Object Identifier	test-1 (nominal)										
	1	code A										
	2	code B										
	3	code C										
	4	code A										

Proximity Measures for Binary Attributes:

A contingency table for binary data		Contingency Table for Binary Attributes			
		Object j			
		1	0	sum	
Object i	1	q	r	$q + r$	
	0	s	t	$s + t$	
	sum	$q + s$	$r + t$	p	

Distance measure for symmetric binary variables:	$d(i, j) = \frac{r + s}{q + r + s + t}$
Distance measure for asymmetric binary variables:	$d(i, j) = \frac{r + s}{q + r + s}$
Jaccard coefficient (similarity measure for asymmetric binary variables):	$sim(i, j) = \frac{q}{q + r + s} = 1 - d(i, j).$

Example: Relational Table Where Patients Are Described by Binary Attributes

Gender is a symmetric attribute

The remaining attributes are asymmetric binary

Let the values Y and P be 1, and the value N be 0

Of the three patients, Jack and Mary are the most likely to have a similar disease

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Dissimilarity of Numeric Data: Minkowski Distance

Minkowski distance: A popular distance measure where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p -dimensional data objects, and h is the order	$d(i, j) = \sqrt[h]{ x_{i1} - x_{j1} ^h + x_{i2} - x_{j2} ^h + \dots + x_{ip} - x_{jp} ^h}$
<i>Special case:</i> $h = 1$: Manhattan (city block, L_1 norm) distance	$d(i, j) = x_{i1} - x_{j1} + x_{i2} - x_{j2} + \dots + x_{ip} - x_{jp} $
<i>Special case:</i> $h = 2$: (L_2 norm) Euclidean distance	$d(i, j) = \sqrt{(x_{i1} - x_{j1} ^2 + x_{i2} - x_{j2} ^2 + \dots + x_{ip} - x_{jp} ^2)}$
Cosine Similarity	$\cos(d_1, d_2) = (d_1 \bullet d_2) / d_1 d_2 $ <p>where \bullet indicates vector dot product,</p>

<p><i>Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. Let x and y be two vectors for comparison.</i></p>	<p>d: the length of vector d</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------

Clustering Algorithms

Partitioning clustering

- K-means and its variants
- K-medoid

Hierarchical clustering

- Agglomerative (Bottom-Up)
- Divisive (Top-Down)

Density-based clustering

- DBSCAN

Important techniques for cluster analysis:


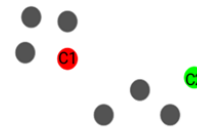

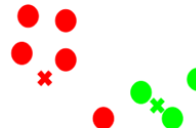

1. **K-means:** This is a prototype-based, partitioning clustering technique that attempts to find a user-specified number of clusters (K), which are represented by their centroids.
2. **Agglomerative Hierarchical Clustering:** It refers to a collection of closely related clustering techniques that produce a hierarchical clustering by starting with each point as a singleton cluster and then repeatedly merging the two closest clusters until a single.
 - Some of these techniques have a natural interpretation in terms of graph-based clustering, while others are prototype-based
3. **DBSCAN:** This is a density-based clustering algorithm that produces a partitioning clustering, in which the number of clusters is automatically determined by the algorithm.
 - Points in low-density regions are classified as noise and omitted;
 - Thus, DBSCAN does not produce a complete clustering

K-means

- Prototype-based clustering techniques create a one-level partitioning of the data objects
- Most prominent techniques are K-means and K-medoid
- K-means defines a prototype in terms of a centroid, which is usually the mean of a group of points
- K-medoid defines a prototype in terms of a medoid, which is the most representative point for a group of points
- It requires only a proximity measure for a pair of objects
- A centroid almost never corresponds to an actual data point,
- A medoid, by its definition, must be an actual data point
- K-means is one of the oldest and most widely used clustering algorithms

The Basic K-means Algorithm

- We first choose K initial centroids, where K is a user specified parameter, namely, the number of clusters desired.
- Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster.
- The centroid of each cluster is then updated based on the points assigned to the cluster.
- We repeat the assignment and update steps until no point changes clusters, or equivalently, until the centroids remain the same.

Step 1: Choose the number of clusters k	Step 2: Select k random points from the data as <u>centroids</u>	Step 3: Assign all the points to the closest cluster <u>centroid</u>	Step 4: <u>Recompute the centroids</u> of newly formed clusters	Step 5: Repeat steps 3 and 4
 Given Data				

Algorithm Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

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.

Complexity is $O(n * K * I * d)$

n = number of points, K = number of clusters,

I = number of iterations, d = number of attributes

Evaluating K-means Clusters

Most common measure is Sum of Squared Error (SSE)-Objective Function

- For each point, the error is the distance to the nearest cluster
- To get SSE, we square these errors and sum them.

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

- x is a data point in cluster C_i and c_i is the representative point for cluster C_i
- \blacklozenge can show that c_i corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase K , ie. the number of clusters

Choosing Initial Centroids

When random initialization of centroids is used, different runs of K-means typically produce different total SSEs



Figure 8.4. Three optimal and non-optimal clusters.

Importance of Choosing Initial Centroids

Randomly selected initial centroids may be poor. In the below Figure, even though the initial centroids seem to be better distributed, we obtain a suboptimal clustering, with higher squared error.

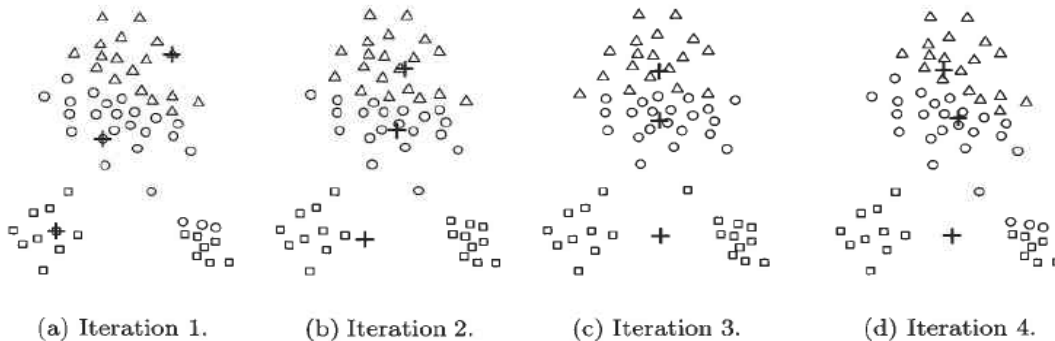


Figure 8.5. Poor starting centroids for K-means.

Solutions to Initial Centroids Problem

Perform multiple runs, each with a different set of randomly chosen initial centroids, and then select the set of clusters with the minimum SSE

- Eg. Next two Slide
- While simple, this strategy may not work very well, depending on the data set and the number of clusters sought

Another effective approach is to take a sample of points and cluster them using a hierarchical clustering technique.

- K clusters are extracted from the hierarchical clustering, and the centroids of those clusters are used as the initial centroids.
- This approach often works well, but is practical only if
 - (1) the sample is relatively small, and
 - (2) K is relatively small compared to the sample size.

Select the first point at random or take the centroid of all points.

- Then, for each successive initial centroid, select the point that is farthest from any of the initial centroids already selected
- centroids that is guaranteed to be not only randomly selected but also well separated
- Unfortunately, such an approach can select outliers, rather than points in dense regions (clusters)

Postprocessing

- We can change the total SSE by performing various operations on the clusters, such as splitting or merging clusters.

Bisecting K-means

- Has less trouble with initialization because it performs several trial bisections and takes the one with the lowest SSE, and because there are only two centroids at each step

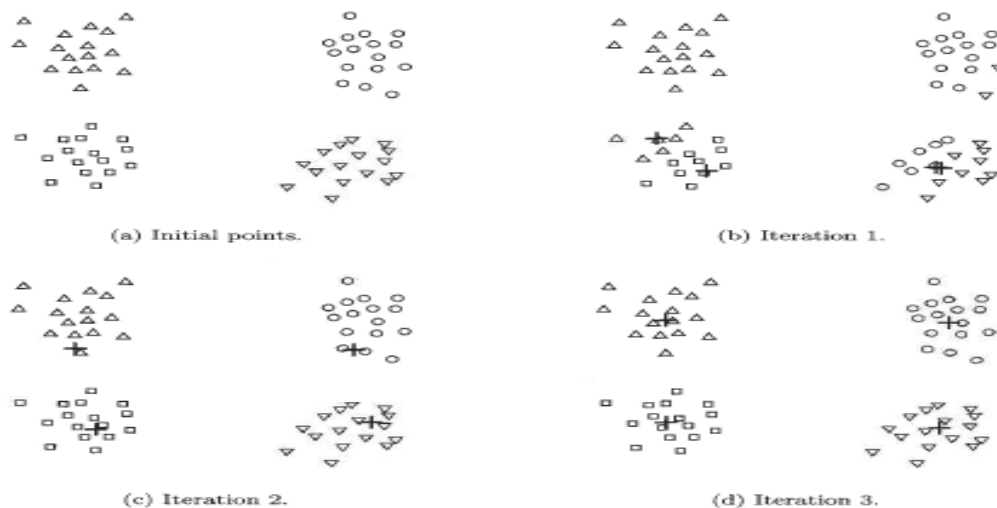


Figure 8.6. Two pairs of clusters with a pair of initial centroids within each pair of clusters.

The data consists of two pairs of clusters, where the clusters in each (top-bottom) pair are closer to each other than to the clusters in the other pair.

Figure 8.6 (b- d) shows that if we start with two initial centroids per pair of clusters, then even when both centroids are in a single cluster, the centroids will redistribute themselves so that the "true" clusters are found.

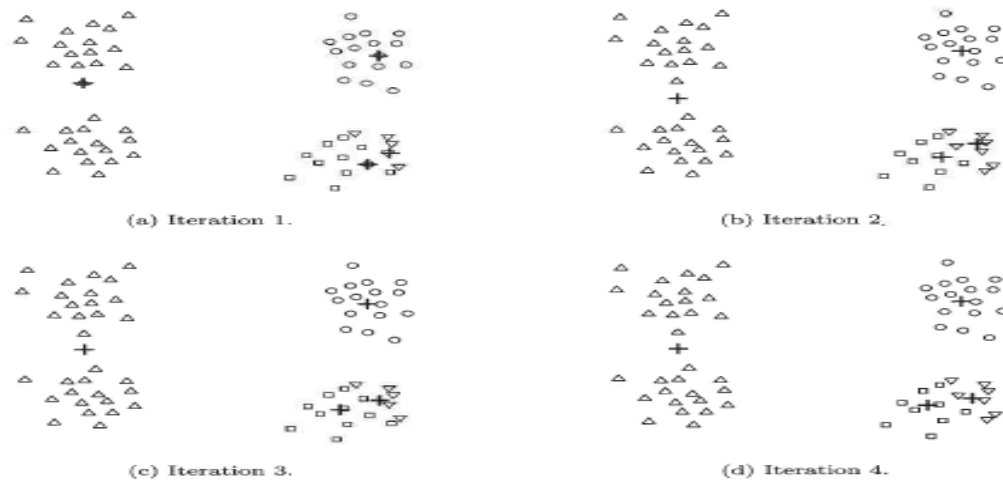


Figure 8.7. Two pairs of clusters with more or fewer than two initial centroids within a pair of clusters.

Figure 8.7 shows that if a pair of clusters has only one initial centroid and the other pair has three, then two of the true clusters will be combined and one true cluster will be split

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.

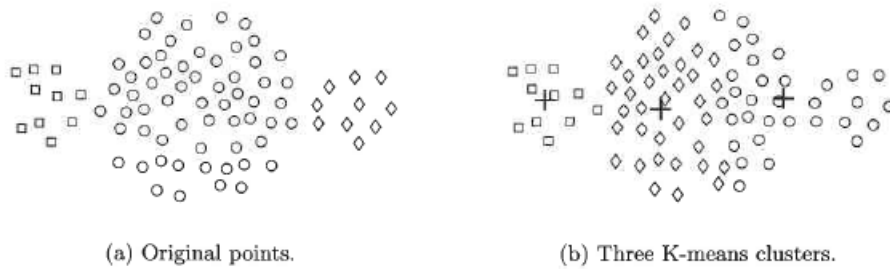


Figure K-means with clusters of different size.

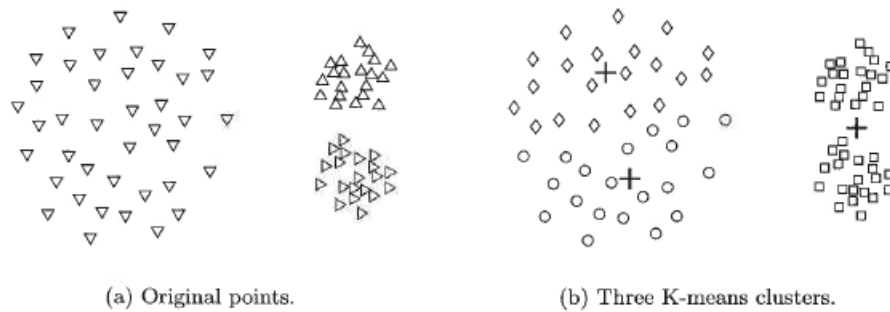


Figure K-means with clusters of different density.

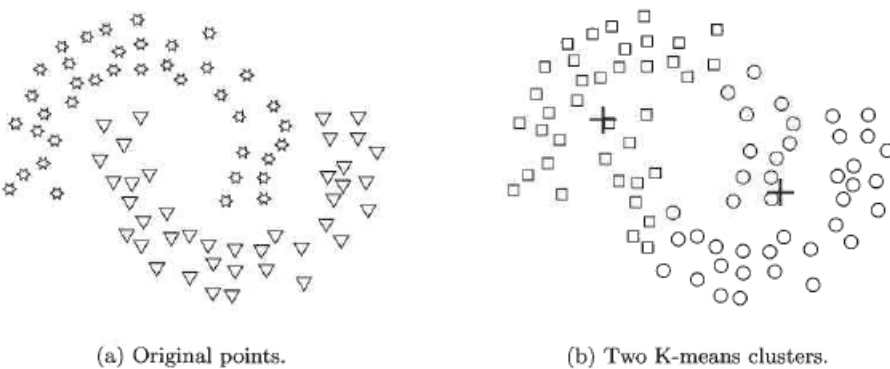


Figure K-means with non-globular clusters.

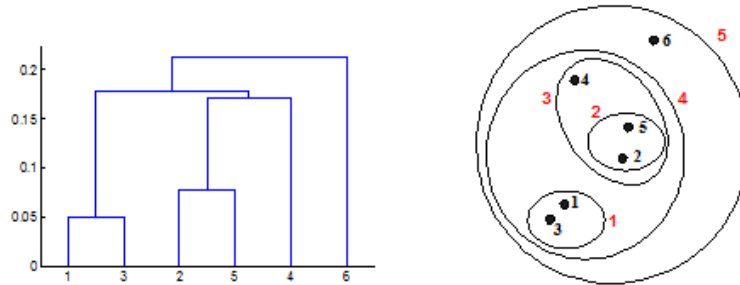
Strengths and Weaknesses of K-means

- K-means is simple and can be used for a wide variety of data types.
- It is also quite efficient, even though multiple runs are often performed.
- Some variants, including bisecting K-means, are even more efficient, and are less susceptible to initialization problems.
- It cannot handle non-globular clusters or clusters of different sizes and densities, although it can typically find pure subclusters if a large enough number of clusters is specified.
- K-means also has trouble clustering data that contains outliers. Outlier detection and removal can help significantly in such situations.

- Finally, K-means is restricted to data for which there is a notion of a center (centroid).

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
 - Displays both the cluster-subcluster relationships



- Two main types of hierarchical clustering
 - Agglomerative:
 - ◆ Starting with individual points as clusters
 - ◆ Successively merge the two closest clusters until only one cluster remains
 - Divisive:
 - ◆ Start with one, all-inclusive cluster
 - ◆ At each step, split a cluster until only singleton clusters of individual points remain
- Traditional hierarchical algorithms use a similarity or distance matrix
 - Merge or split one cluster at a time

Basic Agglomerative Hierarchical Clustering Algorithm

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

- 1: Compute the proximity matrix, if necessary.
 - 2: **repeat**
 - 3: Merge the closest two clusters.
 - 4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
 - 5: **until** Only one 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

How to Define Inter-Cluster Similarity

- MIN or Single Link
- MAX or Complete Linkage or CLIQUE

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

Eg.

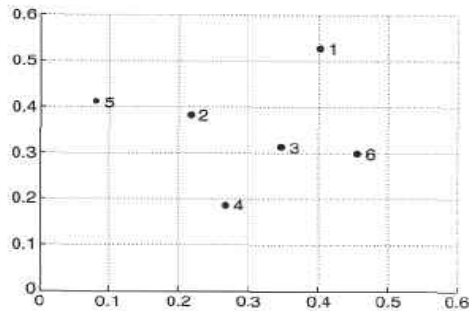


Figure 8.15. Set of 6 two-dimensional points.

Point	<i>x</i> Coordinate	<i>y</i> Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

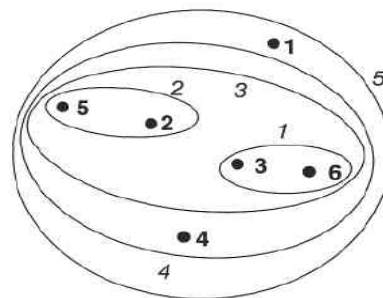
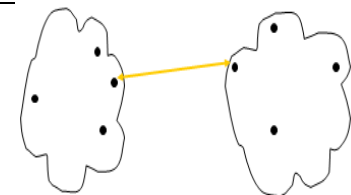
Table 8.3. *xy* coordinates of 6 points.

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

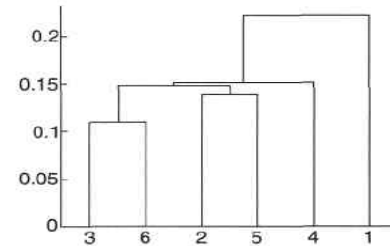
Table 8.4. Euclidean distance matrix for 6 points.

1. Cluster Similarity: MIN or Single Link

-- The proximity of two clusters is defined as the minimum of the distance (maximum of the similarity) between any two points in the two different clusters.
 -- Determined by one pair of points, i.e., by one link in the proximity graph



(a) Single link clustering.



(b) Single link dendrogram.

Figure 8.16. Single link clustering of the six points shown in Figure 8.15.

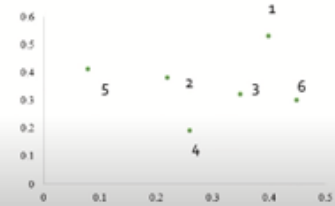
$$\begin{aligned}
 \text{dist}(\{3, 6\}, \{2, 5\}) &= \min(\text{dist}(3, 2), \text{dist}(6, 2), \text{dist}(3, 5), \text{dist}(6, 5)) \\
 &= \min(0.15, 0.25, 0.28, 0.39) \\
 &= 0.15.
 \end{aligned}$$

Hierarchical Clustering: MIN or Single Link – Explanation with example

- Find the clusters using single link technique. Use Euclidean distance, and draw the dendrogram.

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30



- Calculate Euclidean distance, create the distance matrix.

$$\text{Distance} [(x,y), (a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$$

$$\text{Distance (P1,P2)} = \sqrt{(0.40 - 0.22)^2 + (0.53 - 0.38)^2}$$

$$(0.40, 0.53), (0.22, 0.38) = \sqrt{(0.18)^2 + (0.15)^2}$$

$$= \sqrt{0.0324 + 0.0225}$$

$$= \sqrt{0.0549}$$

$$= 0.23$$

- The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

The proximity of two clusters is defined as the **minimum of the distance** (maximum of the similarity) between any two points in the two different clusters. Determined by one pair of points, i.e., by one link in the proximity graph.

- The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



- The distance matrix for cluster P3,P6

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

The updated distance matrix for cluster P3, P6

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

The distance matrix is

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

To update the distance matrix $\text{MIN}[\text{dist}(\text{P3,P6}), \text{P1}]$

$\text{MIN}(\text{dist}(\text{P3,P1}), (\text{P6,P1}))$

$$= \min[(0.22, 0.23)]$$

$$= 0.22$$

To update the distance matrix $\text{MIN}[\text{dist}(\text{P3,P6}), \text{P2}]$

$\text{MIN}(\text{dist}(\text{P3,P2}), (\text{P6,P2}))$

$$= \min[(0.15, 0.25)]$$

$$= 0.15$$

To update the distance matrix $\text{MIN}[\text{dist}(\text{P3,P6}), \text{P4}]$

$\text{MIN}(\text{dist}(\text{P3,P4}), (\text{P6,P4}))$

$$= \min[(0.15, 0.22)]$$

$$= 0.15$$

To update the distance matrix $\text{MIN}[\text{dist}(\text{P3,P6}), \text{P5}]$

$\text{MIN}(\text{dist}(\text{P3,P5}), (\text{P6,P5}))$

$$= \min[(0.28, 0.39)]$$

$$= 0.28$$



The distance matrix fro cluster P2, P5

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

The updated distance matrix for cluster P2,P5

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

The distance matrix is

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

To update the distance matrix $\text{MIN}[\text{dist}(P2,P5),P1]$

$\text{MIN}[\text{dist}(P2,P1), (P5,P1)]$

$= \min[(0.23,0.34)]$

$= 0.23$

To update the distance matrix $\text{MIN}[\text{dist}(P2,P5),(P3,P6)]$

$\text{MIN}[\text{dist}(P2,(P3,P6)), (P5,(P3,P6))]$

$= \min[(0.15,0.28)]$

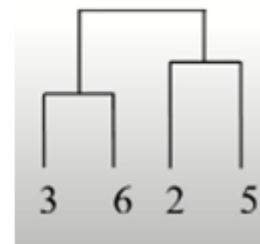
$= 0.15$

To update the distance matrix $\text{MIN}[\text{dist}(P2,P5),P4]$

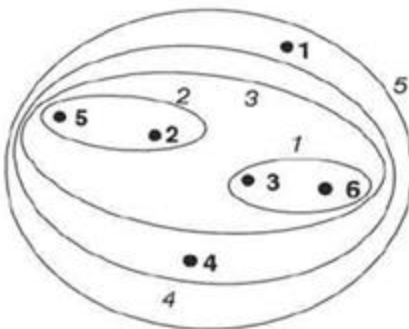
$\text{MIN}[\text{dist}(P2,P4), (P5,P4)]$

$= \min[(0.20,0.29)]$

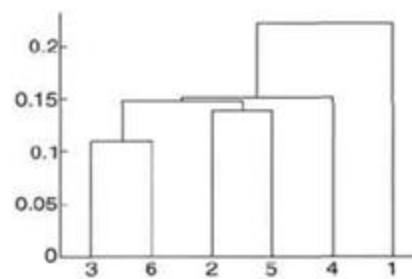
$= 0.20$



Final Output:



(a) Single link clustering.

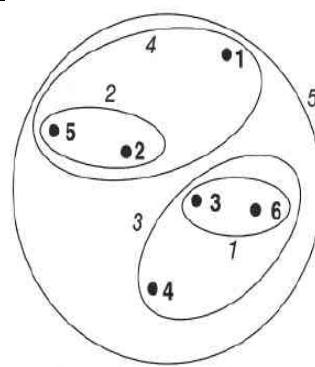


(b) Single link dendrogram.

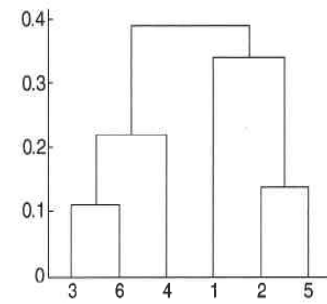
Figure Single link clustering of the six points

2. Cluster Similarity: MAX or Complete Linkage or CLIQUE

-- The proximity of two clusters is defined as the maximum of the distance (minimum of the similarity) between any two points in the two different clusters. (CLIQUE Clustering In QUEst)
 -- Determined by all pairs of points in the two clusters
 -- As with single link, points 3 and 6 are merged first.
 -- As with single link, points 3 and 6 are merged first. However, {3,6} is merged with {4}, instead of {2,5} or {1} because

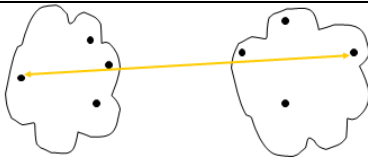


(a) Complete link clustering.



(b) Complete link dendrogram.

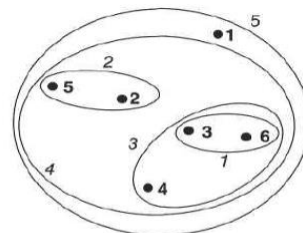
Figure 8.17. Complete link clustering of the six points shown in Figure 8.15.



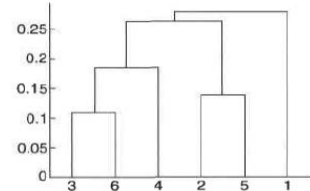
$$\begin{aligned} \text{dist}(\{3, 6\}, \{4\}) &= \max(\text{dist}(3, 4), \text{dist}(6, 4)) \\ &= \max(0.15, 0.22) \\ &= 0.22. \\ \text{dist}(\{3, 6\}, \{2, 5\}) &= \max(\text{dist}(3, 2), \text{dist}(6, 2), \text{dist}(3, 5), \text{dist}(6, 5)) \\ &= \max(0.15, 0.25, 0.28, 0.39) \\ &= 0.39. \\ \text{dist}(\{3, 6\}, \{1\}) &= \max(\text{dist}(3, 1), \text{dist}(6, 1)) \\ &= \max(0.22, 0.23) \\ &= 0.23. \end{aligned}$$

3. Cluster Similarity: Group Average

-- The proximity of two clusters is defined as the average pairwise proximity among all pairs of points in the different clusters.
 -- This is an intermediate approach between the single and complete link approaches.

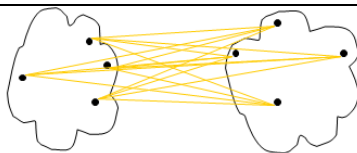


(a) Group average clustering.



(b) Group average dendrogram.

Figure 8.18. Group average clustering of the six points shown in Figure 8.15.

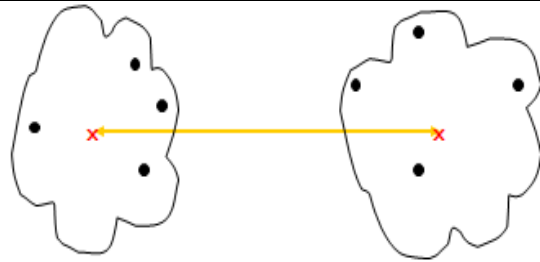


$\text{proximity}(C_i, C_j)$ of clusters C_i and C_j , which are of size m_i and m_j , respectively, is expressed by the following equation:

$$\text{proximity}(C_i, C_j) = \frac{\sum_{\substack{x \in C_i \\ y \in C_j}} \text{proximity}(x, y)}{m_i * m_j}$$

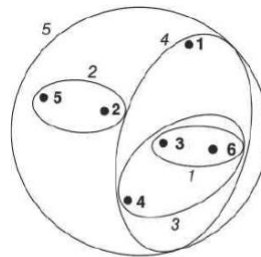
4. Cluster Similarity: Centroid methods

- **Centroid methods** calculate the proximity between two clusters by calculating the distance between the centroids of clusters.
- These techniques may seem similar to K-means

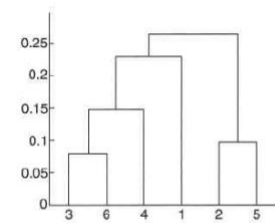


5. Cluster Similarity: Ward's Method

- The proximity between two clusters is defined as the increase in the squared error that results when two clusters are merged
- Similar to group average if distance between points is distance squared



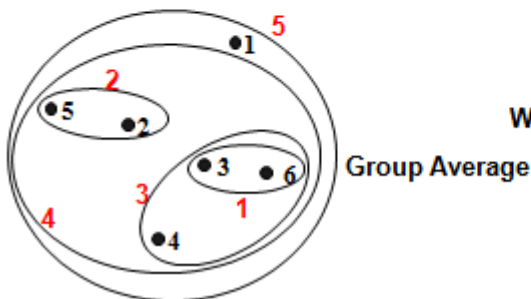
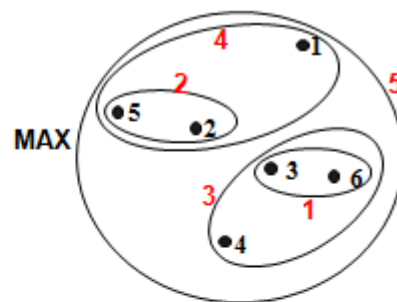
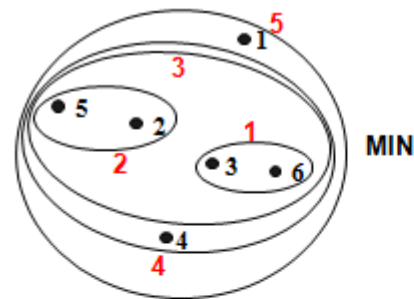
(a) Ward's clustering.



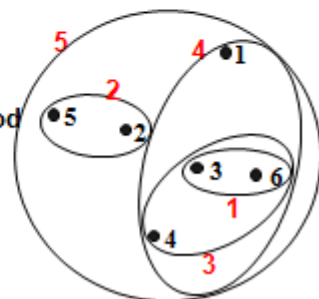
(b) Ward's dendrogram.

Figure 8.19. Ward's clustering of the six points shown in Figure 8.15.

Hierarchical Clustering: Comparison

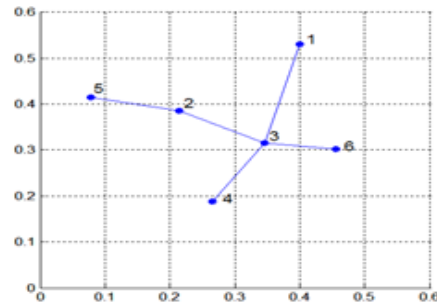
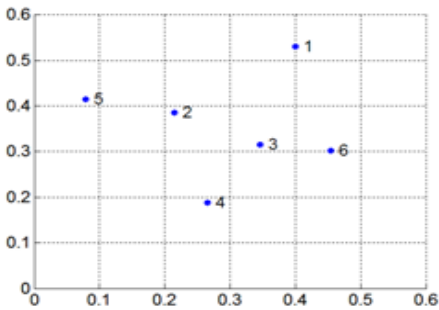


Ward's Method



Divisive Hierarchical Clustering(MST)

- Build MST (Minimum Spanning Tree)
 - Start with a tree that consists of any point
 - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
 - Add q to the tree and put an edge between p and q



Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

- 1: Compute a minimum spanning tree for the proximity graph.
 - 2: **repeat**
 - 3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
 - 4: **until** Only singleton clusters remain
-

Strengths and Weaknesses of Hierarchical Clustering

1. Strengths:

- 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
 - ◆ Animals can be classified into two main groups: **vertebrates** and **invertebrates**.

2. Weaknesses:

- No 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
 - Breaking large clusters

DBSCAN (Density-based spatial clustering of applications with noise)

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

Density-based clustering locates regions of high density that are separated from one another by regions of low density.

Traditional Density: Center-Based Approach

In the center-based approach, density is estimated for a particular point in the data set by counting the number of points within a specified radius, .Eps, of that point.

The number of points within a radius of Eps of point A is 7, including A itself

Classification of Points According to Center-Based Density

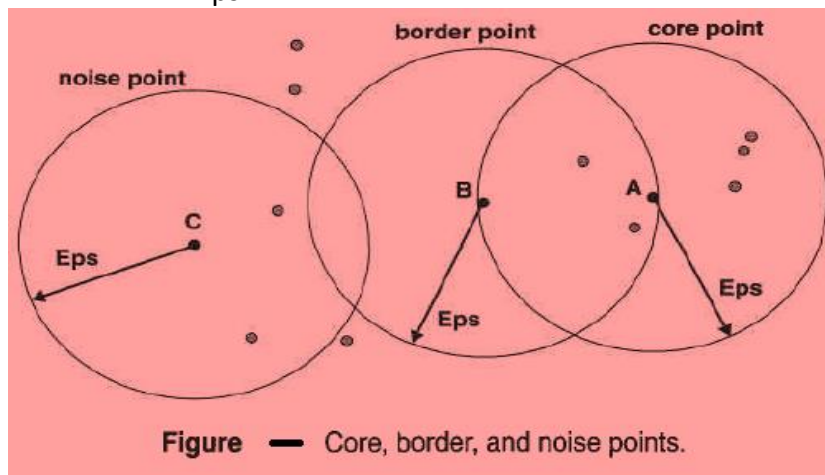
- (1) Core point
- (2) Border point
- (3) Noise point

Core points: 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
- In Figure, point A is a core point, for the indicated radius (Eps) if $\text{MinPts} \leq 7$.

Input parameters:

- MinPts
- Eps



Border points : A border point is not a core point, but falls within the neighborhood of a core point.

In Figure, point B is a border point.

Noise points: A noise point is any point that is neither a core point nor a border point. In Figure, point C is a noise point.

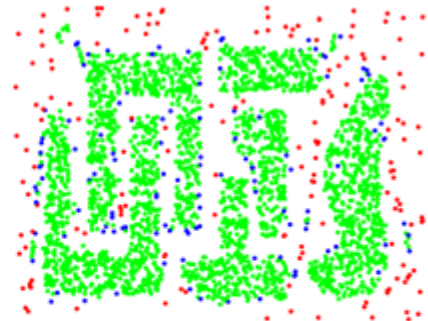
Algorithm 8.4 DBSCAN algorithm.

- 1: Label all points as core, border, or noise points.
- 2: Eliminate noise points.
- 3: Put an edge between all core points that are within Eps 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.



Original Points

$Eps = 10$, $MinPts = 4$

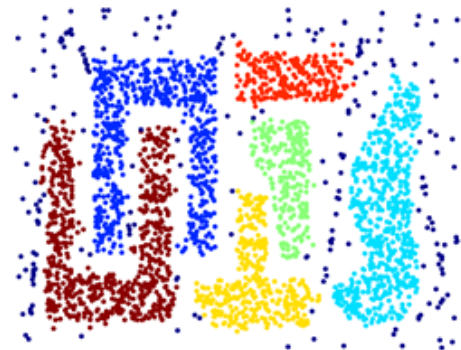


Point types: **core**,
border and **noise**

Strengths of DBSCAN



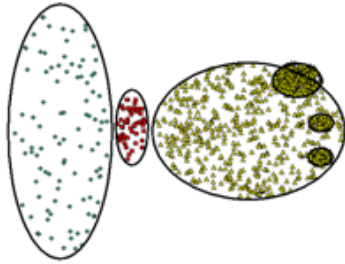
Original Points



Clusters

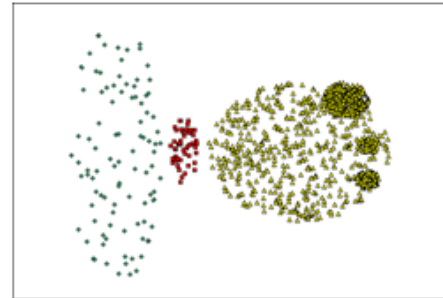
- Resistant to Noise
- Can handle clusters of different shapes and sizes

Weaknesses of DBSCAN

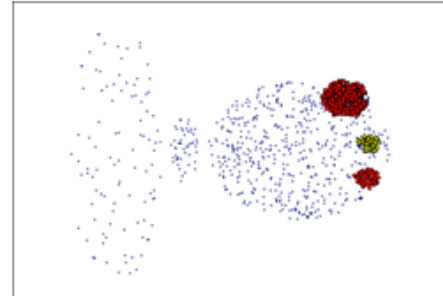


Original Points

- Clusters have widely varying densities
- High-dimensional data - density is more difficult to define
- DBSCAN can be expensive in case of high-dimensional data.



(MinPts=4, Eps=9.75)



(MinPts=4, Eps=9.92)
