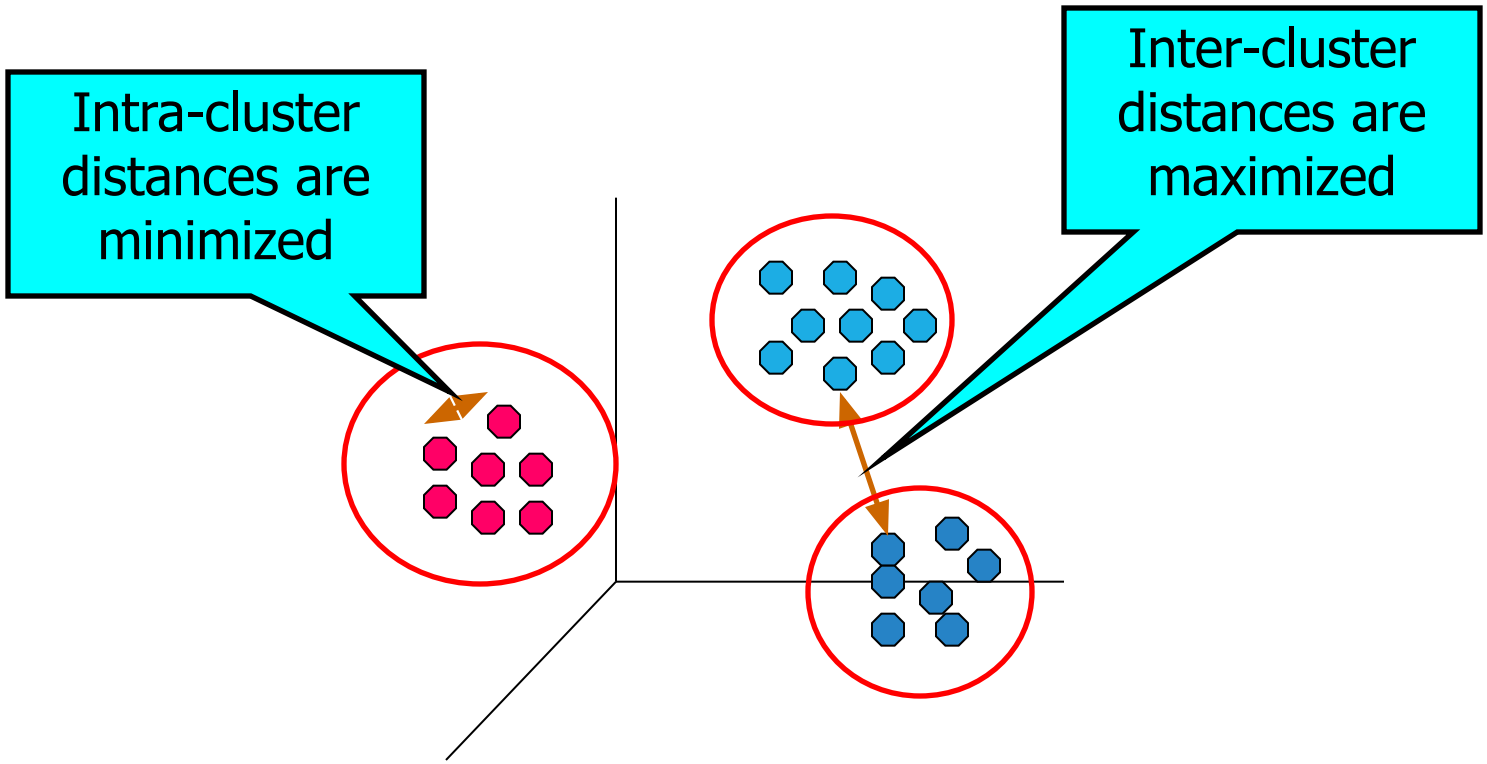
DATA MINING
CLUSTER ANALYSIS: BASIC CONCEPTS AND ALGORITHMS

Lecture Notes for Chapter 7

Introduction to Data Mining

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

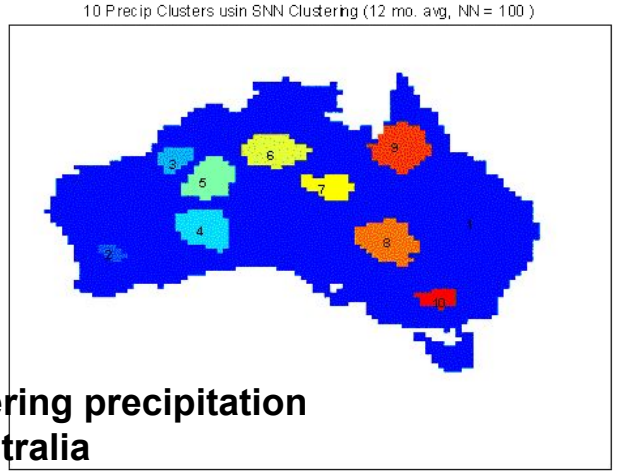
Understanding

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

	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

Summarization

- Reduce the size of large data sets



Clustering precipitation in Australia

NOTION OF A CLUSTER CAN BE AMBIGUOUS

How many clusters?

Six Clusters

Two Clusters

Four Clusters

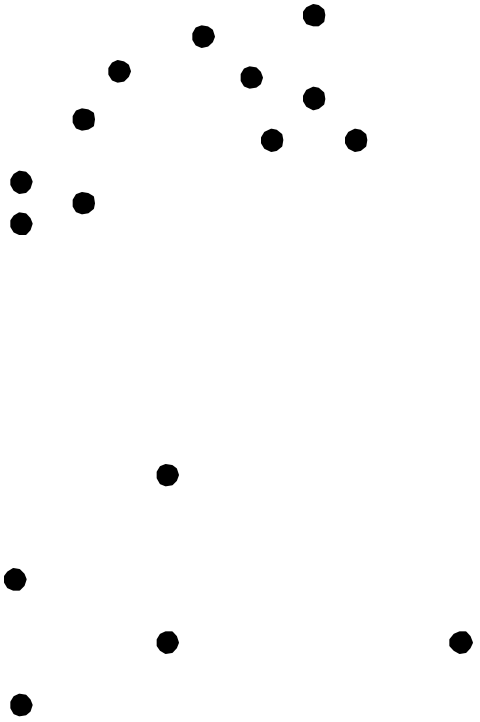
TYPES OF CLUSTERINGS

A clustering is a set of clusters

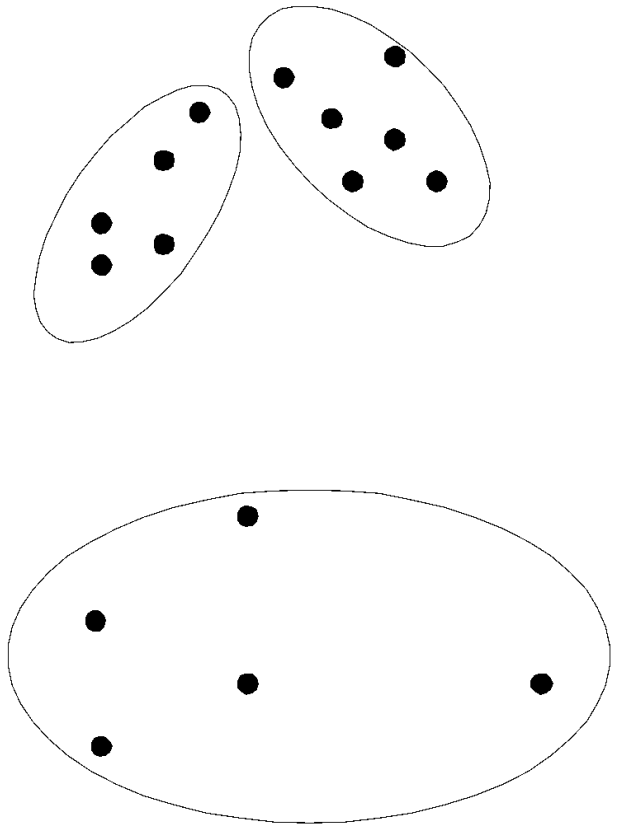
Important distinction between hierarchical and partitional sets of clusters

- 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

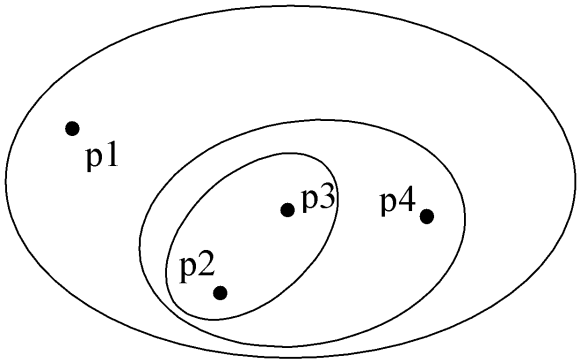


Original Points

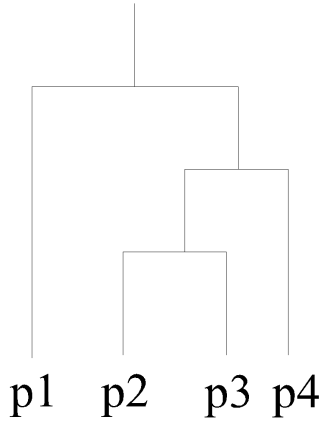


A Partitional Clustering

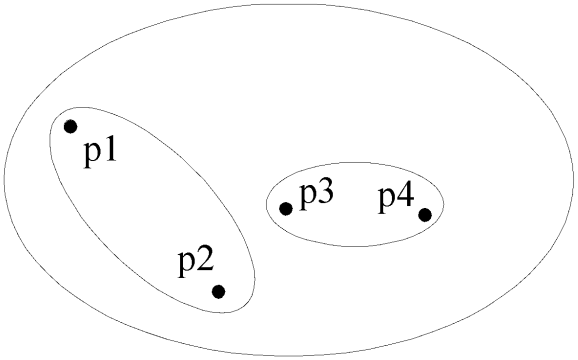
HIERARCHICAL CLUSTERING



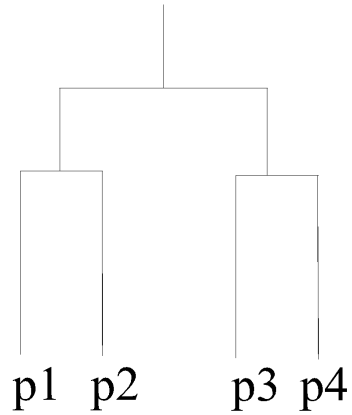
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

OTHER DISTINCTIONS BETWEEN SETS OF CLUSTERS

Exclusive versus non-exclusive

- In non-exclusive clusterings, points may belong to multiple clusters.
  - Can belong to multiple classes or could be 'border' points
  - Fuzzy clustering (one type of non-exclusive)
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics

Partial versus complete

- In some cases, we only want to cluster some of the data

CLUSTERING ALGORITHMS

K-means and its variants

Hierarchical clustering

Density-based clustering

K-MEANS CLUSTERING

Number of clusters, K, must be specified

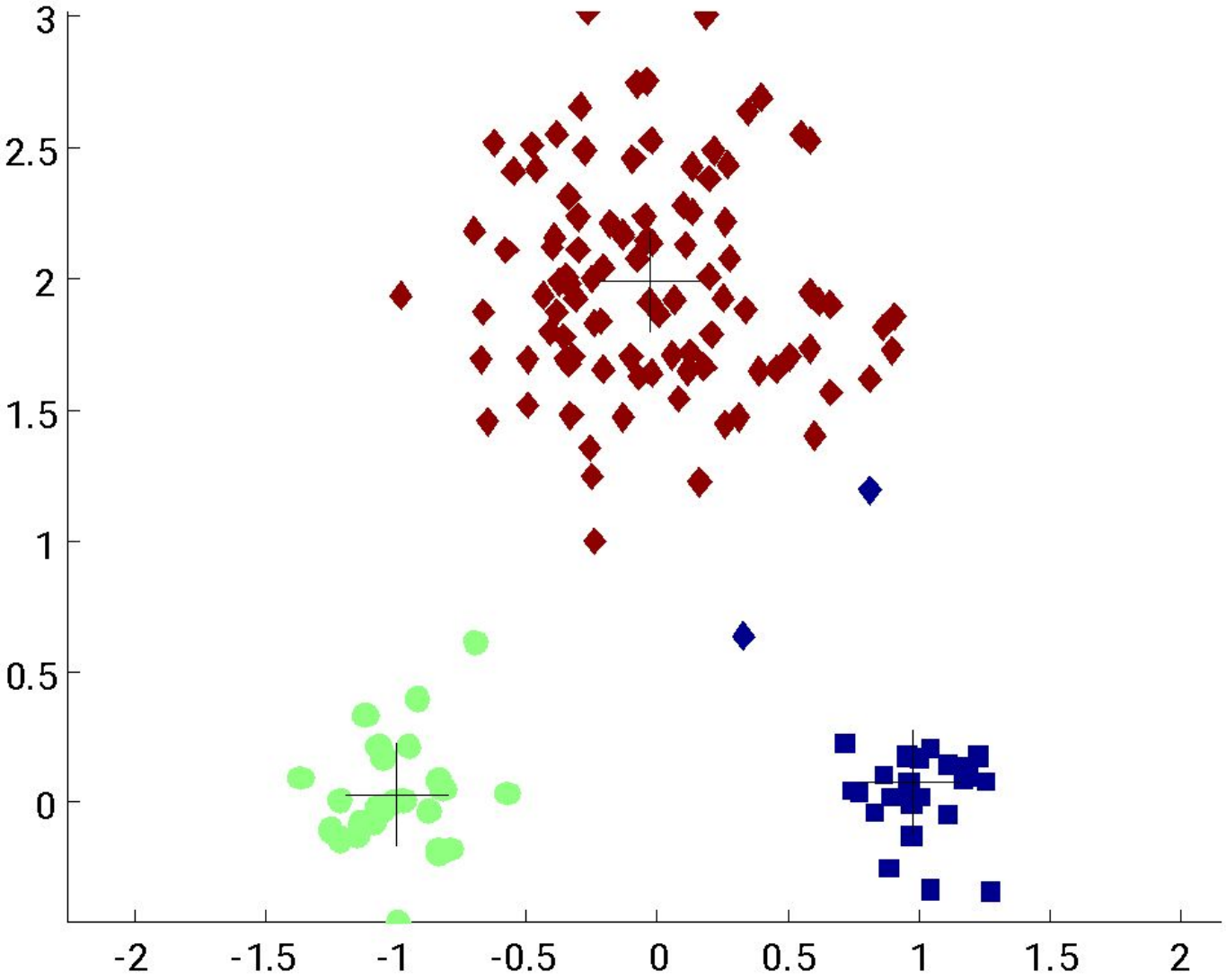
Each cluster is associated with a centroid (center point)

Each point is assigned to the cluster with the closest centroid

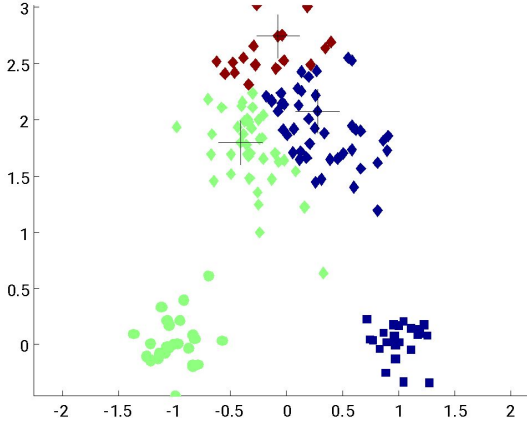
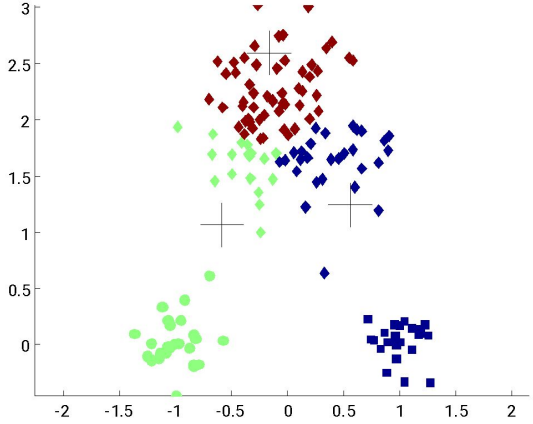
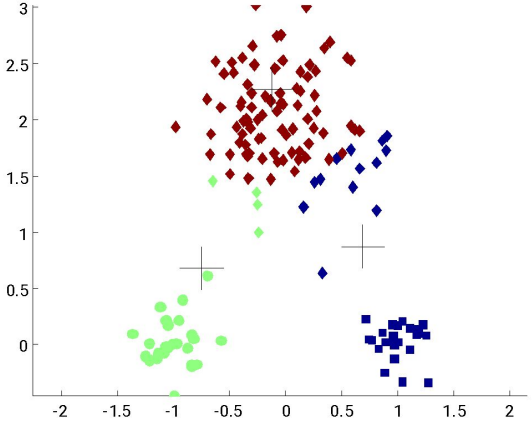
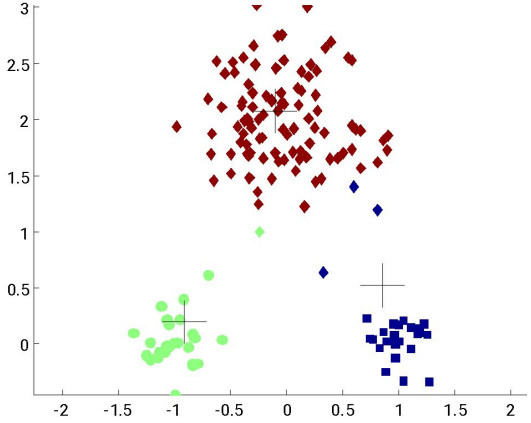
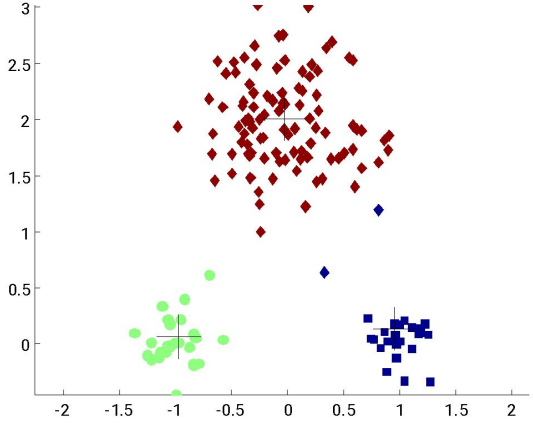
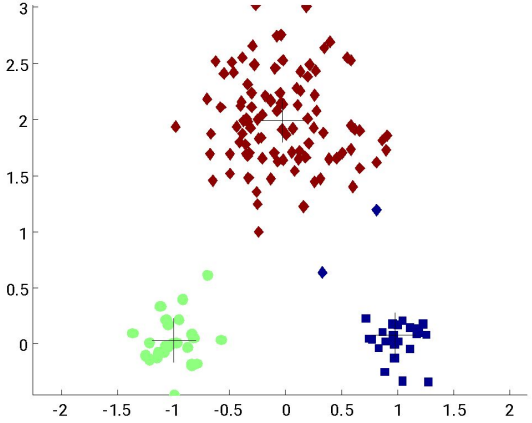
The basic algorithm is very simple

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

- 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 proximity 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

EXAMPLE

A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2)

The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as-

$$P(a, b) = |x2 - x1| + |y2 - y1|$$

ITERATION 1

- We calculate the distance of each point from each of the center of the three clusters.
- The distance is calculated by using the given distance function.

P(A1, C1)

= |x2 – x1| + |y2 – y1|

= |2 – 2| + |10 – 10|

= 0

ITERATION 1 - CONTINUED

P(A1, C2)

= |x2 – x1| + |y2 – y1|

= |5 – 2| + |8 – 10|

= 3 + 2

= 5

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

CENTROID UPDATE

First cluster contains points-

- A1(2, 10)

Second cluster contains points-

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)
- A8(4, 9)

CENTROID UPDATE

Third cluster contains points

- A2(2, 5)
- A7(1, 2)
- We re-compute the new cluster clusters.
- The new cluster center is computed by taking mean of all the points contained in that cluster.

CENTROID UPDATE

For Cluster-01:

We have only one point A1(2, 10) in Cluster-01.

So, the cluster center remains the same.

For Cluster-02

= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)

= (6, 6)

For Cluster-03

= ((2 + 1)/2, (5 + 2)/2)

= (1.5, 3.5)

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1

CENTROID UPDATE

Cluster 01 points

- A1(2, 10)
- A8(4, 9)

Cluster 02 points

- A3(8, 4)
- A4(5, 8)
- A5(7, 5)
- A6(6, 4)

CENTROID UPDATE

Third cluster contains points-

- A2(2, 5)
- A7(1, 2)

Center of Cluster-01

= ((2 + 4)/2, (10 + 9)/2)

= (3, 9.5)

CENTROID UPDATE

Center of Cluster-02

= ((8 + 5 + 7 + 6)/4, (4 + 8 + 5 + 4)/4)

= (6.5, 5.25)

Center of Cluster-02

= ((8 + 5 + 7 + 6)/4, (4 + 8 + 5 + 4)/4)

= (6.5, 5.25)

FINAL CENTERS

After second iteration, the center of the three clusters are-

- C1(3, 9.5)
- C2(6.5, 5.25)
- C3(1.5, 3.5)

Sample No.	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

AGGLOMERATIVE CLUSTERING EXAMPLE: SINGLE LINK

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.13	0		
P5	0.34	0.14	0.28	0.23	0	
P6	0.24	0.24	0.10	0.22	0.39	0

FIRST MERGE

D(P1, P2) = SQRT((X-A)^2 + (Y-B)^2)

Merge P6 and P3

RECALCULATE DISTANCE

min ((P3,P6), P1) = min ((P3,P1), (P6,P1)) = min (0.22,0.24) = 0.22

min ((P3,P6), P2) = min ((P3,P2), (P6,P2)) = min (0.14,0.24) = 0.14

min ((P3,P6), P4) = min ((P3,P4), (P6,P4)) = min (0.13,0.22) = 0.13

min ((P3,P6), P5) = min ((P3,P5), (P6,P5)) = min (0.28,0.39) = 0.28

	$P1$	$P2$	$P3, P6$	$P4$	$P5$
$P1$	0				
$P2$	0.23	0			
$P3, P6$	0.22	0.14	0		
$P4$	0.37	0.19	0.13	0	
$P5$	0.34	0.14	0.28	0.23	0

UPDATED DISTANCE MATRIX

Merge P4 and P3,P6

REPEAT

min (((P3,P6) P4), P1) = min (((P3,P6), P1), (P4,P1))
= min (0.22,0.37) = 0.22

min (((P3,P6), P4), P2) = min (((P3,P6), P2), (P4,P2))
= min (0.14,0.19) = 0.14

min (((P3,P6), P4), P5) = min (((P3,P6), P5), (P4,P5))
= min (0.28,0.23) = 0.23

	$P1$	$P2$	$P3, P6, P4$	$P5$
$P1$	0			
$P2$	0.23	0		
$P3, P6, P4$	0.22	0.14	0	
$P5$	0.34	0.14	0.23	0

UPDATE DISTANCE MATRIX

Merge P2 and P5

RECALCULATE DISTANCE

min ((P2,P5), P1) = min ((P2,P1), (P5,P1)) = min (0.23, 0.34) = 0.23

min ((P2,P5), (P3,P6,P4)) = min ((P3,P6,P4), (P3,P6,P4)) = min (0.14. 0.23) = 0.14

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

UPDATE DISTANCE MATRIX

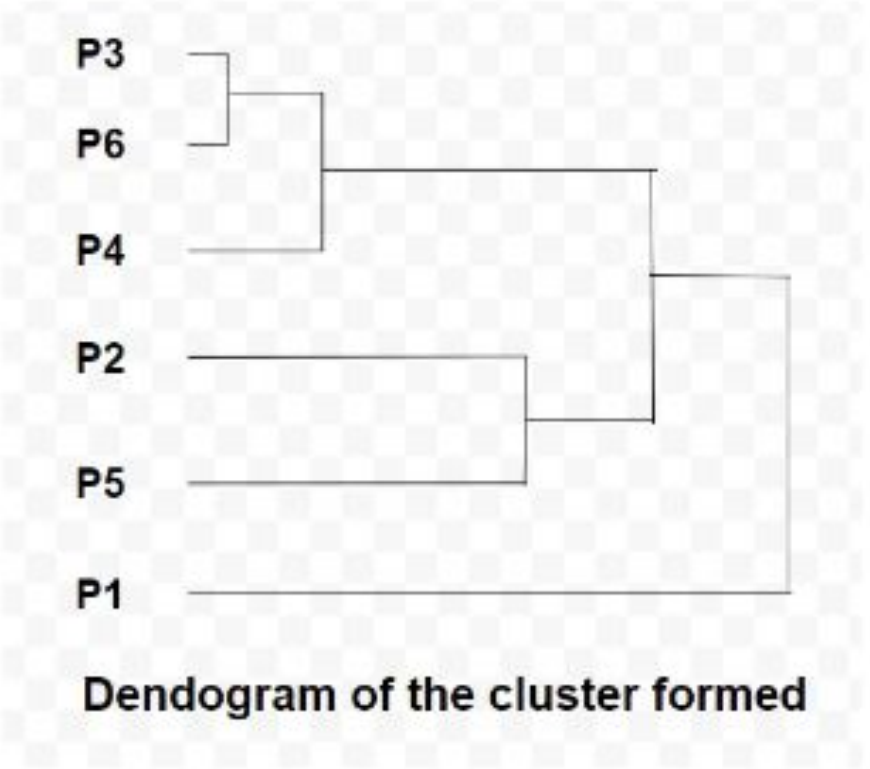
Merge P3,P6,P4 and P2, P5

RECALCULATE DISTANCE

min ((P2,P5,P3,P6,P4), P1) = min ((P2,P5), P1), ((P3,P6,P4), P1)) = min (0.23, 0.22) = 0.22

$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

FINAL MATRIX



Dendogram of the cluster formed

AGGLOMERATIVE CLUSTERING EXAMPLE: COMPLETE LINK

Sample No	X	Y
P1	1	1
P2	1.5	1.5
P3	5	5
P4	3	4
P5	4	4
P6	3	3.5

	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$
$P1$	0					
$P2$	0.71	0				
$P3$	5.66	4.95	0			
$P4$	3.6	2.92	2.24	0		
$P5$	4.24	3.53	1.41	1.0	0	
$P6$	3.20	2.5	2.5	0.5	1.12	0

DISTANCE MATRIX

UPDATE DISTANCE MATRIX

max (d(P4,P6), P1) = max (d(P4,P1), d(P6,P1))
= **max** (3.6, 3.2) = 3.6

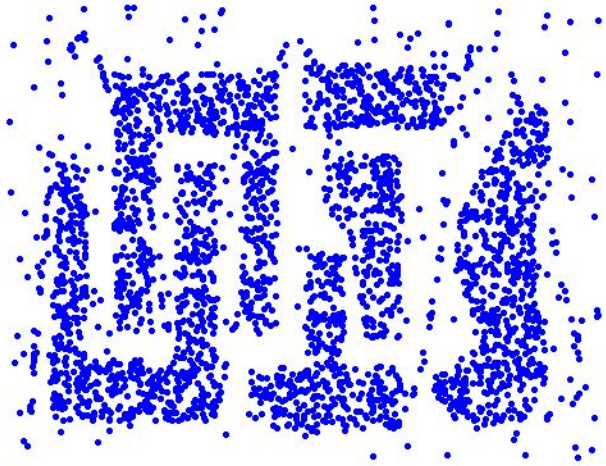
max (d(P4,P6), P2) = max (d(P4,P2), d(P6,P2)) =
max (2.92, 2.5) = 2.92

max (d(P4,P6), P3) = max (d(P4,P3), d(P6,P3)) =
max (2.24, 2.5) = 2.5

max (d(P4,P6), P5) = max (d(P4,P5), d(P6,P5)) =
max (1.0, 1.12) = 1.12

DENSITY BASED CLUSTERING

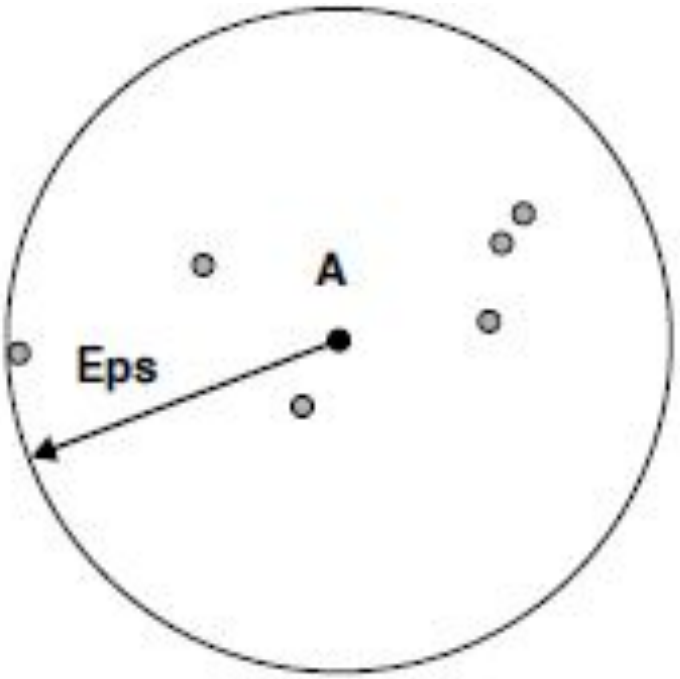
Clusters are regions of high density that are separated from one another by regions on low density.



DBSCAN

DBSCAN is a density-based algorithm.

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

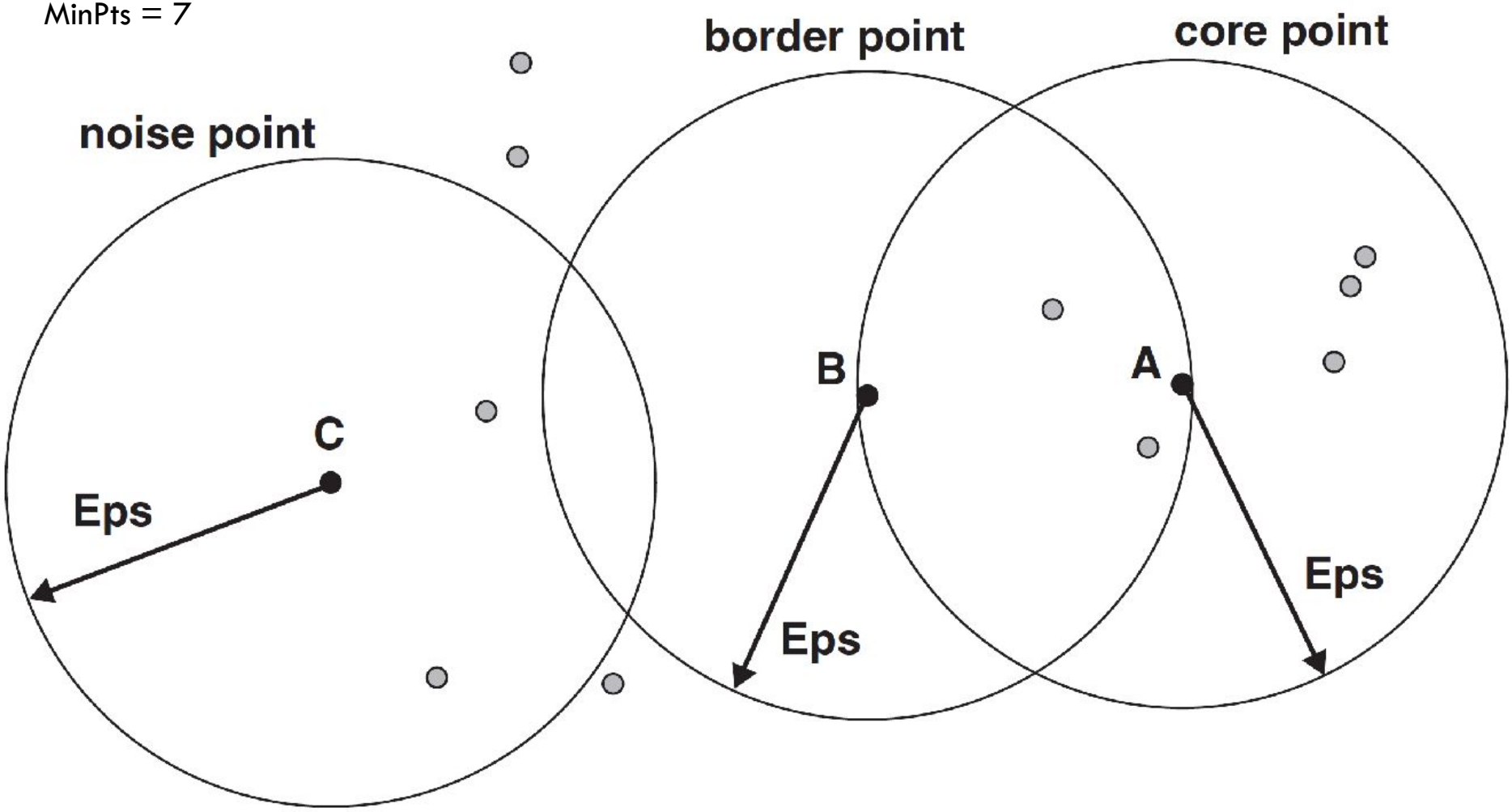


DBSCAN

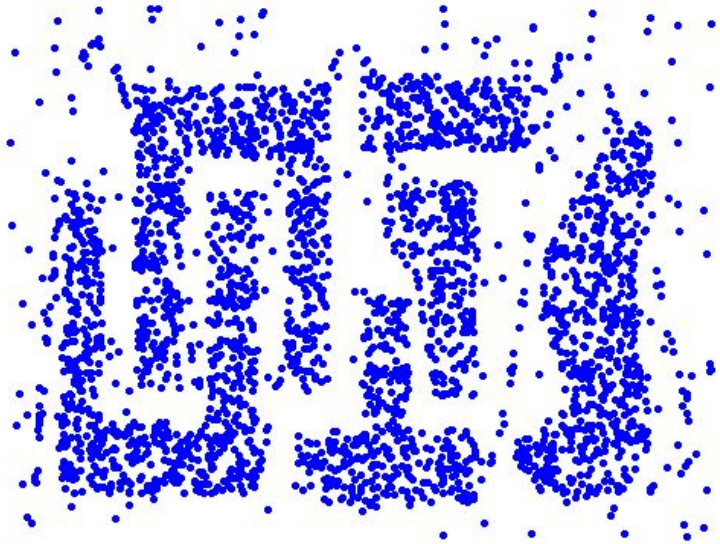
- A point is a core point if it has at least a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
  - Counts the point itself
- A border point is not a core point, but is in the neighborhood of a core point
- A noise point is any point that is not a core point or a border point

DBSCAN: CORE, BORDER, AND NOISE POINTS

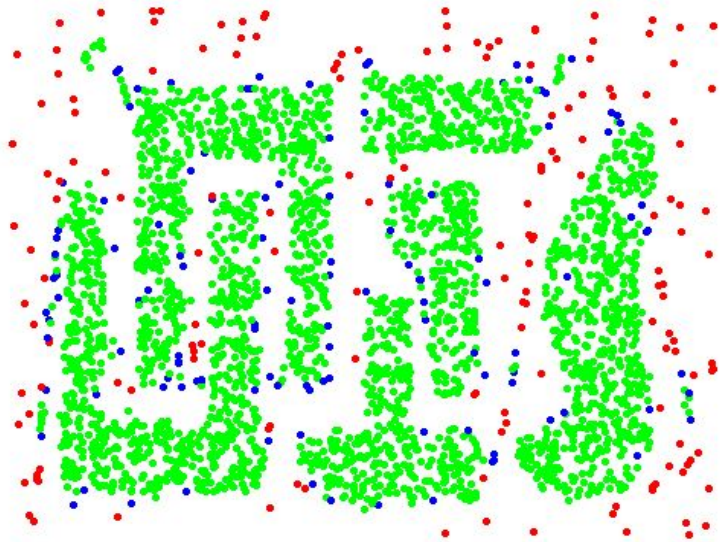
MinPts = 7



DBSCAN: CORE, BORDER AND NOISE POINTS



Original Points



Point types: core, border and noise

Eps = 10, MinPts = 4

DBSCAN

Given the previous definitions of core points, border points, and noise points, the DBSCAN algorithm can be informally described as follows. Any two core points that are close enough—within a distance, *Eps* of one another—are put in the same cluster. Likewise, any border point that is close enough to a core point is put in the same cluster as the core point. (Ties may need to be resolved if a border point is close to core points from different clusters.) Noise points are discarded.

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.

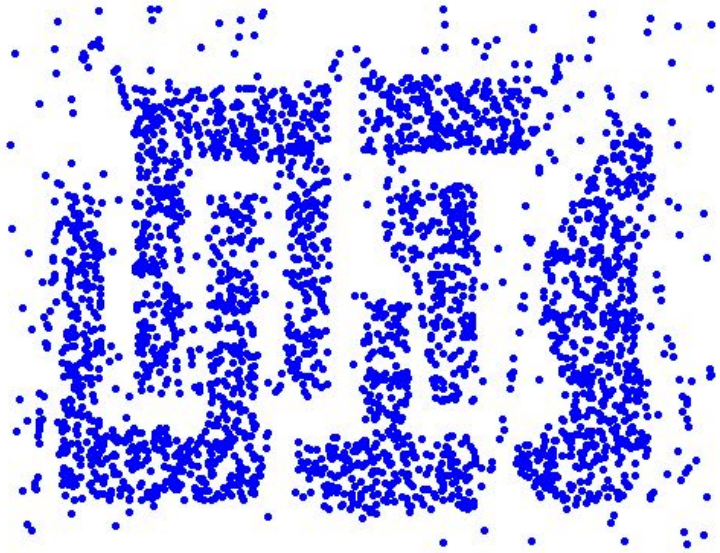
2: Eliminate noise points.

3: Put an edge between all core points within a distance *Eps* of each other.

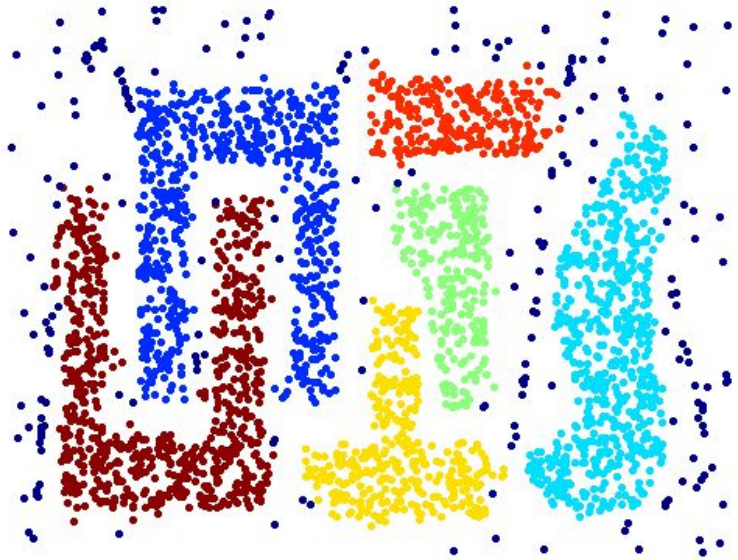
4: Make each group of connected core points into a separate cluster.

5: Assign each border point to one of the clusters of its associated core points

WHEN DBSCAN WORKS WELL



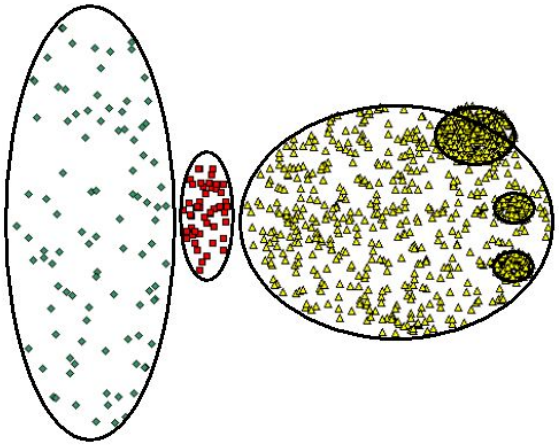
Original Points



Clusters (dark blue points indicate noise)

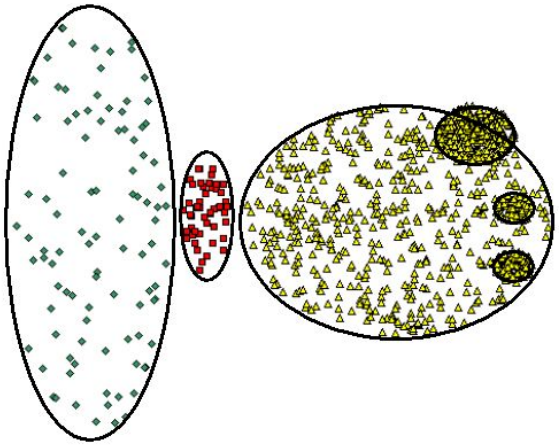
- **Can handle clusters of different shapes and sizes**
- **Resistant to noise**

WHEN DBSCAN DOES NOT WORK WELL

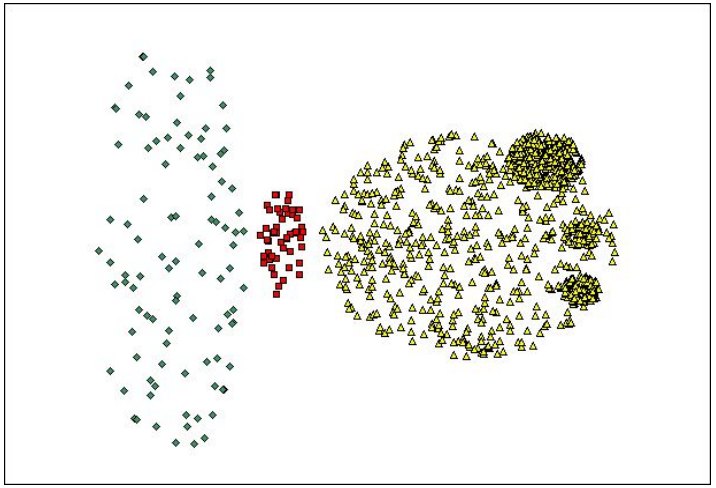


Original Points

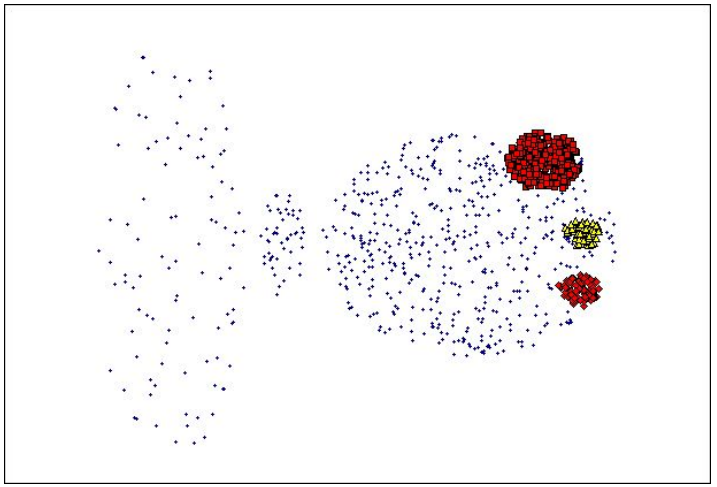
WHEN DBSCAN DOES NOT WORK WELL



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

- **Varying densities**
- **High-dimensional data**

CLUSTER VALIDITY

For supervised classification we have a variety of measures to evaluate how good our model is

- Accuracy, precision, recall

For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?

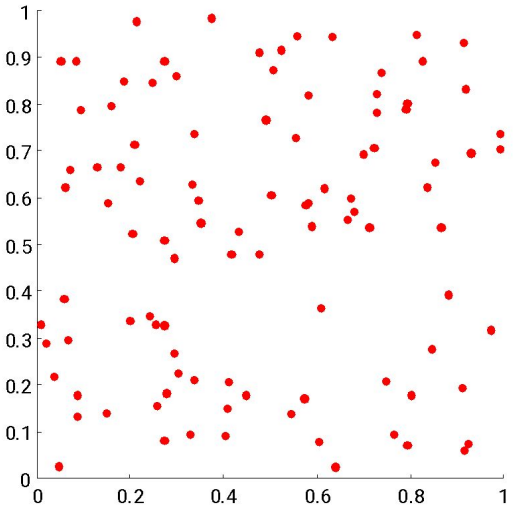
But "clusters are in the eye of the beholder"!

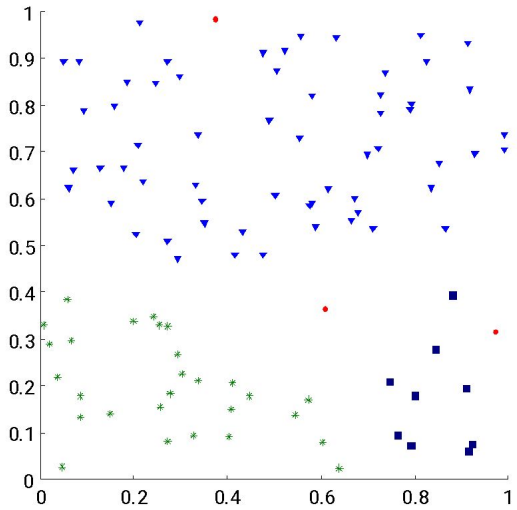
- In practice the clusters we find are defined by the clustering algorithm

Then why do we want to evaluate them?

- To avoid finding patterns in noise
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

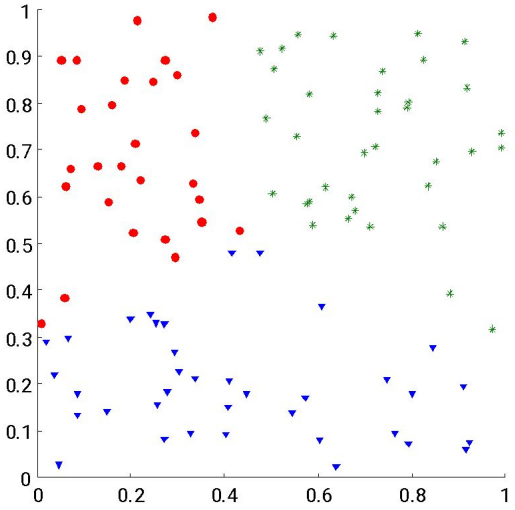
Random Points

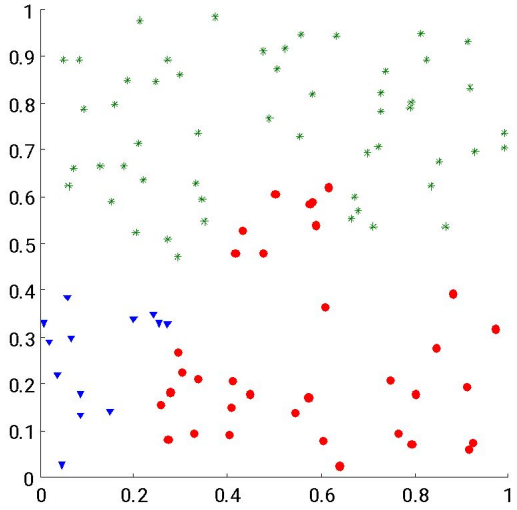




DBSCAN

K-means





Complete Link

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.
  - Entropy
  - 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 clusters or clusterings

UNSUPERVISED MEASURES: COHESION AND SEPARATION

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

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

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)

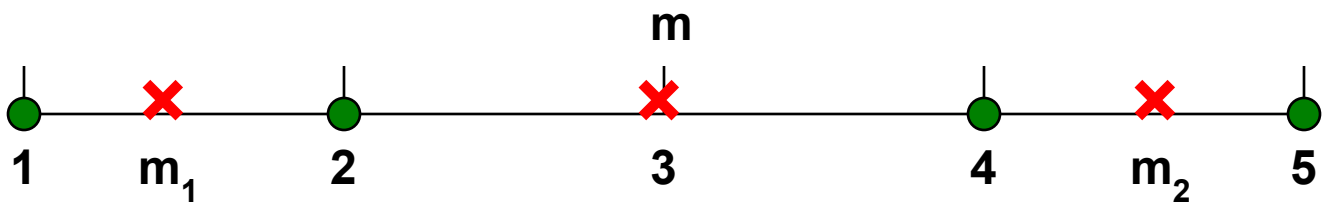
Separation is measured by the between cluster sum of squares

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

UNSUPERVISED MEASURES: COHESION AND SEPARATION

Example: SSE

- SSB + SSE = 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:

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

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

$$Total = 1 + 9 = 10$$