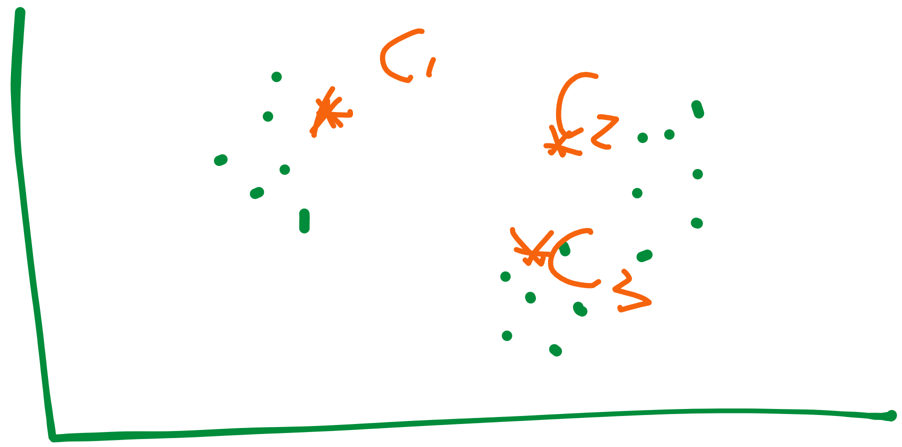
CLUSTERING

CLUSTERING ALGORITHMS

- K-means and its variants
- Hierarchical clustering
- Density-based clustering



partition

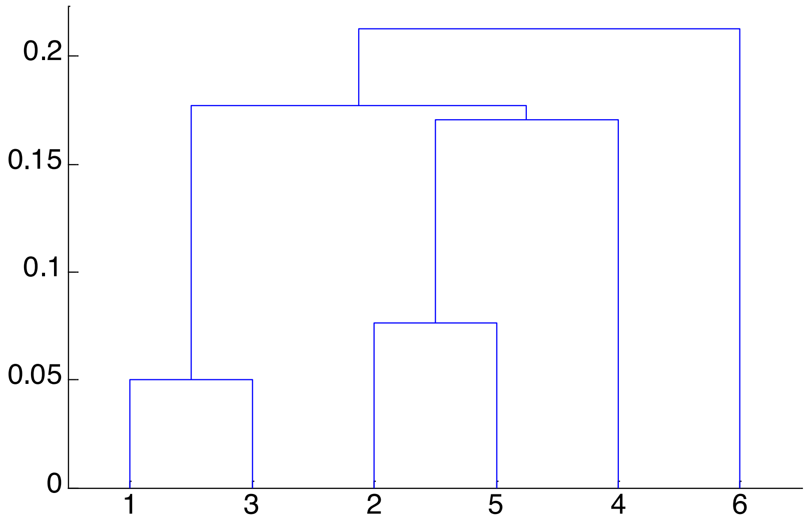
k-mean++

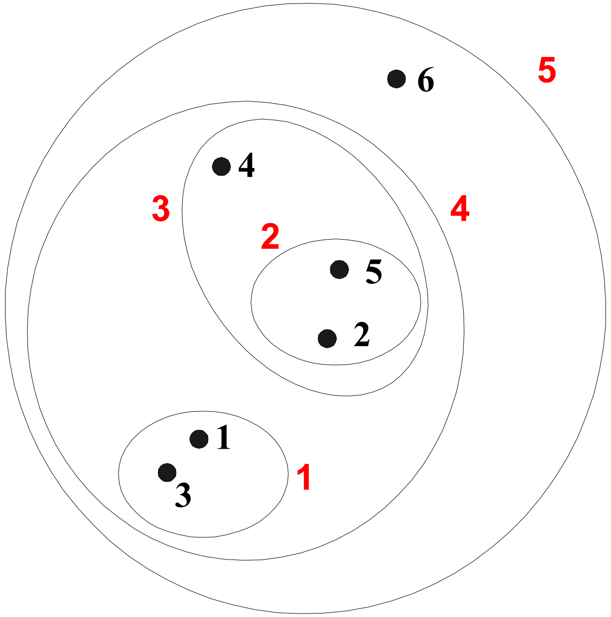
k=3

loop k-1 =2

HIERARCHICAL CLUSTERING

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

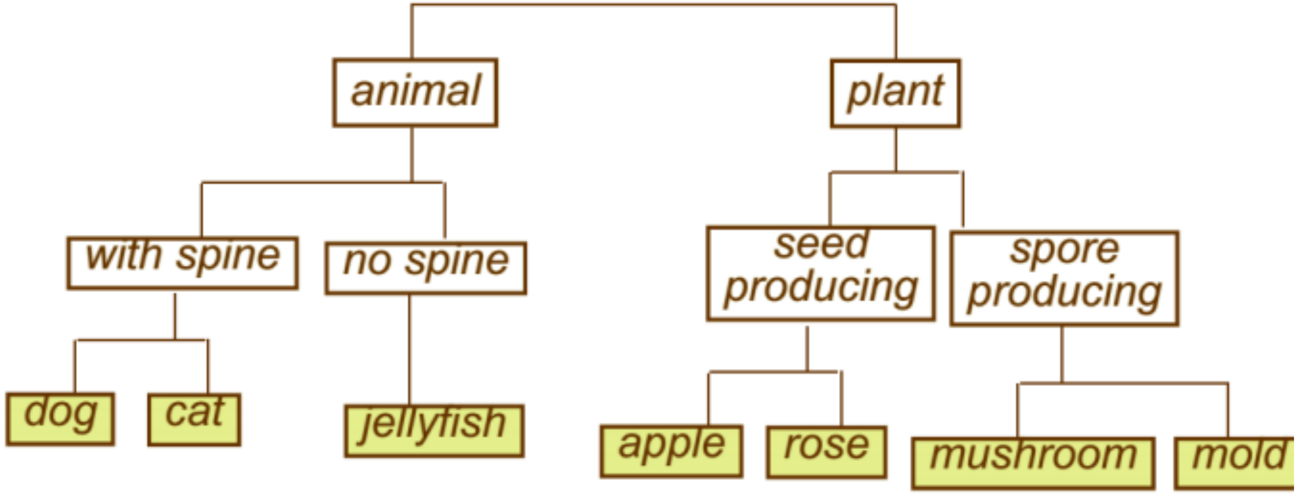




STRENGTHS OF HIERARCHICAL CLUSTERING

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

- They may correspond to meaningful taxonomies
  - Example: biological science

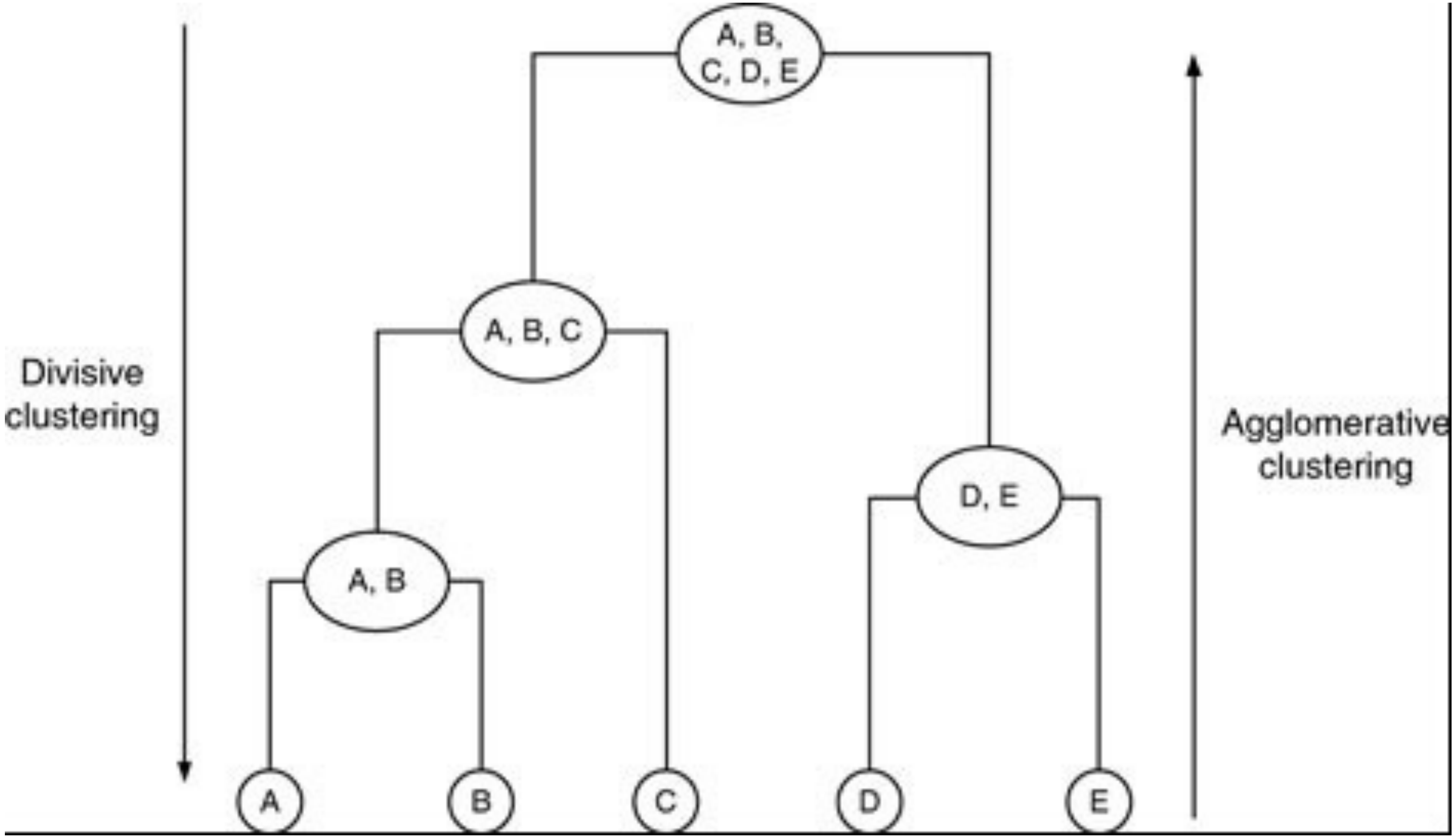


HIERARCHICAL CLUSTERING

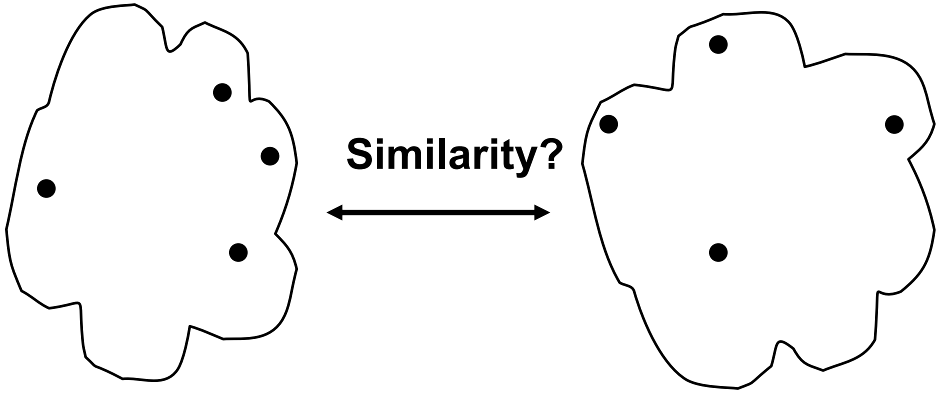
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster 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

HIERARCHICAL CLUSTERING



HOW TO DEFINE INTER-CLUSTER DISTANCE

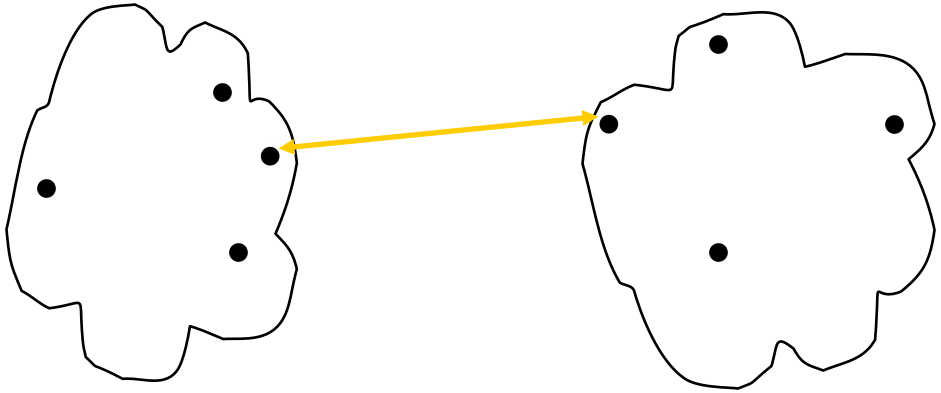


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

HOW TO DEFINE INTER-CLUSTER SIMILARITY

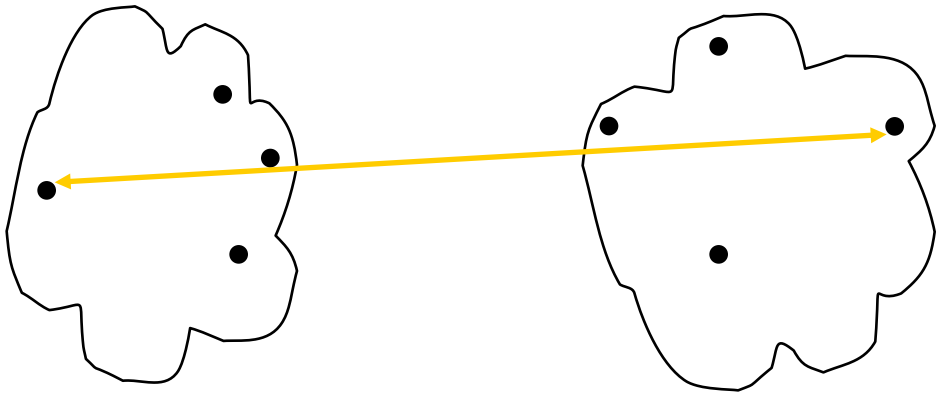


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

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

Proximity 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

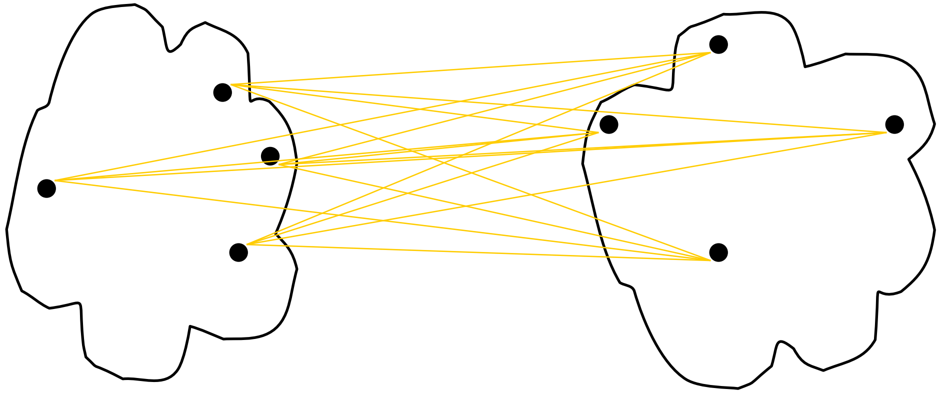
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

.

.

Proximity 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

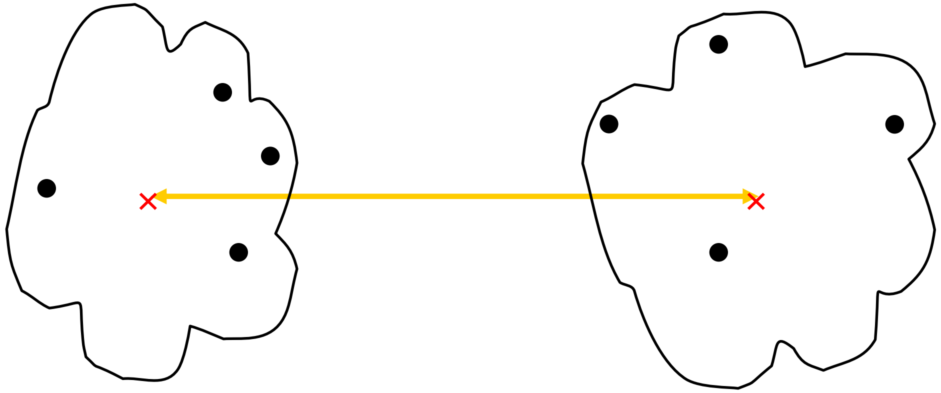
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

.

.

Proximity Matrix

HOW TO DEFINE INTER-CLUSTER SIMILARITY



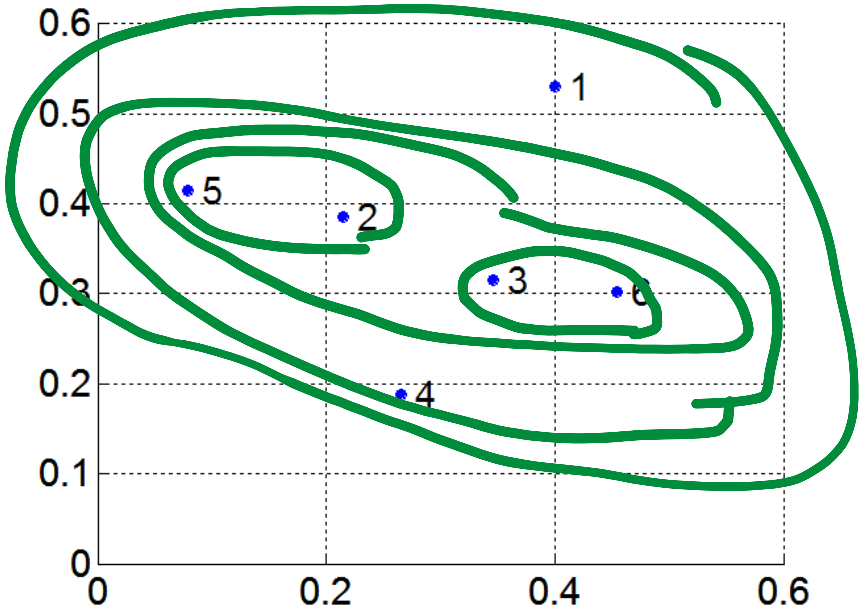
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids (centroid: average of all points in that cluster)
- Other methods driven by an objective function
  - Ward's Method uses squared error

MIN OR SINGLE LINK

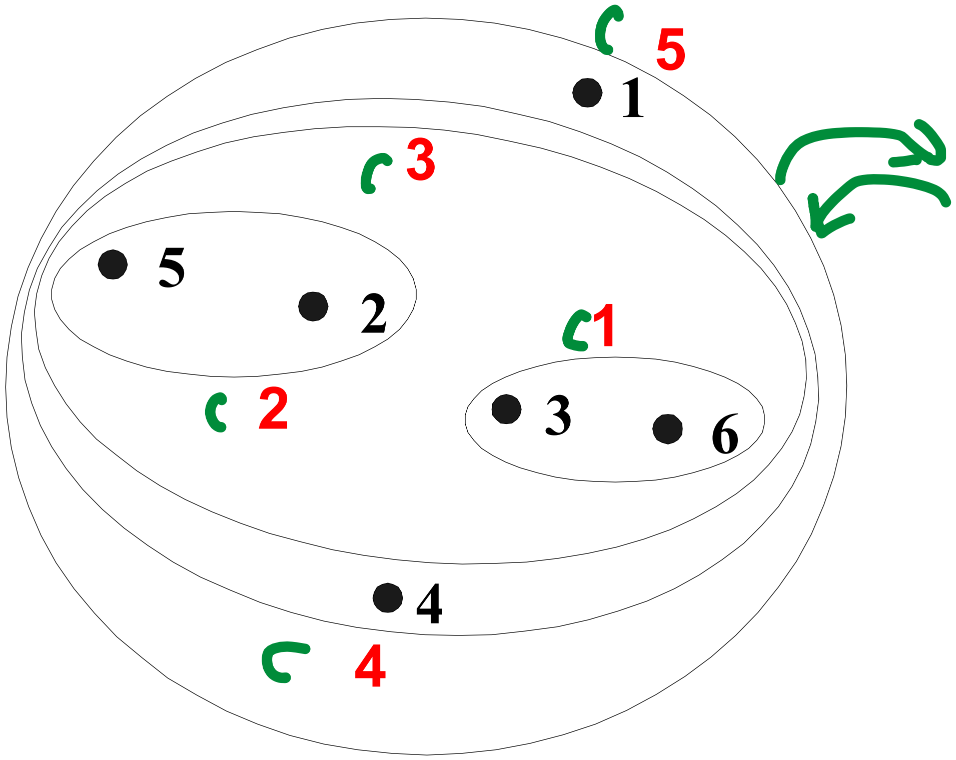
- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph
- Example:



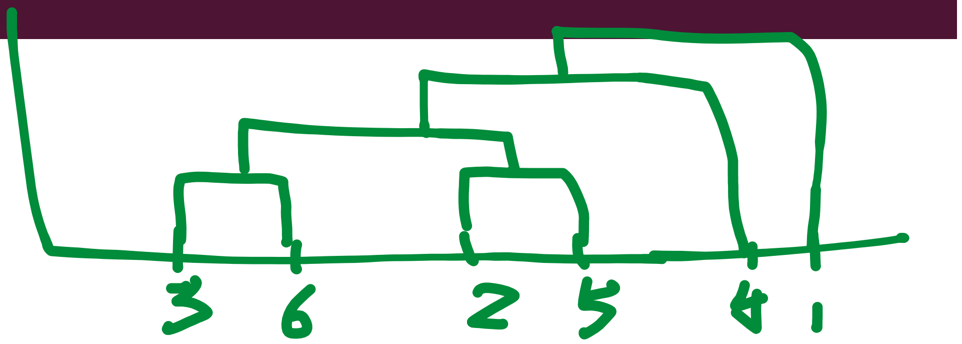
Distance Matrix:

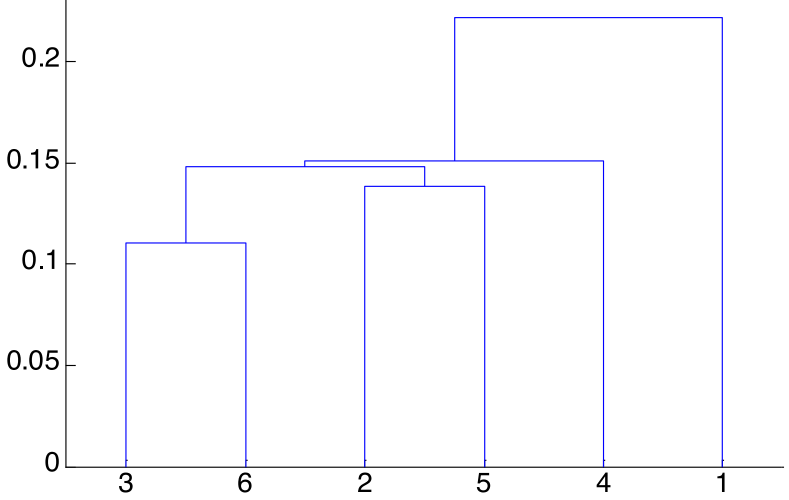
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

HIERARCHICAL CLUSTERING: MIN



Nested Clusters



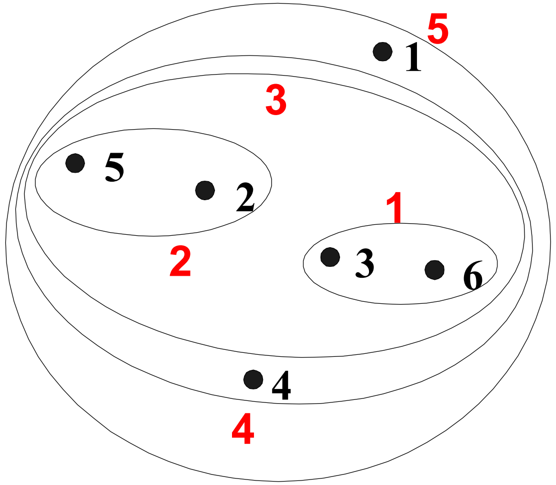


Dendrogram

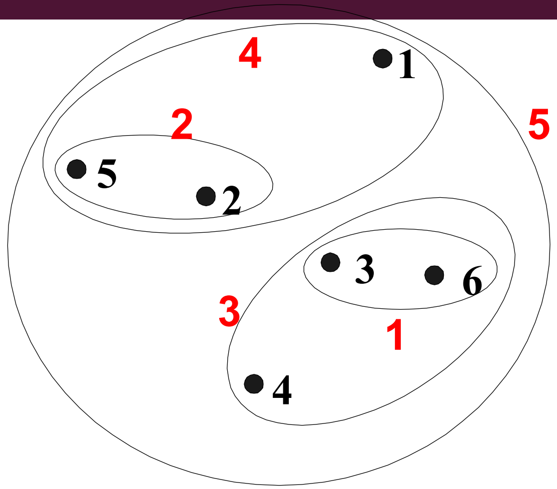
PROS AND CONS

	Min	Max	Group Average
Strength	Can handle non-elliptical shapes	Less susceptible to noise	Less susceptible to noise
Weakness	Sensitive to noise	Tends to break large clusters	Biased towards globular clusters
		Biased towards globular clusters	

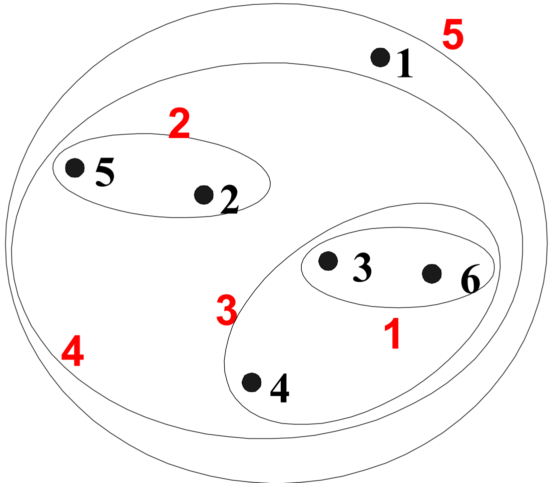
HIERARCHICAL CLUSTERING: COMPARISON



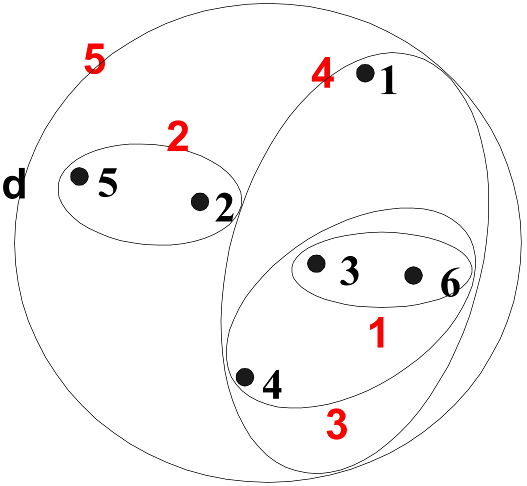
MIN



MAX



Group Average



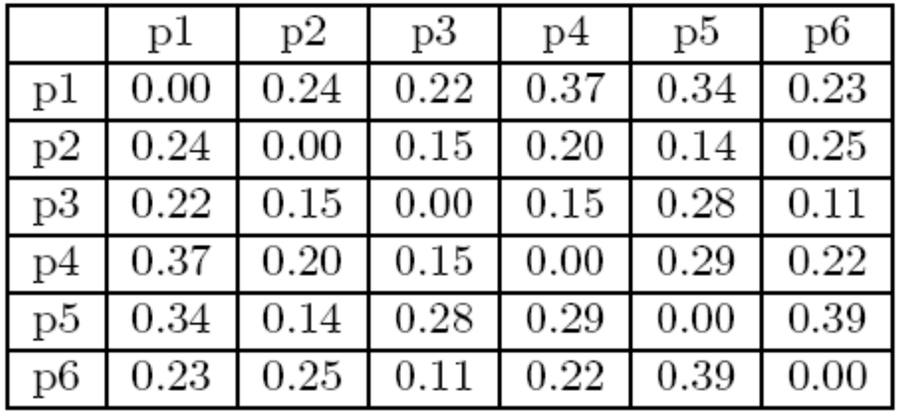
Ward's Method

HIERARCHICAL CLUSTERING: TIME AND SPACE REQUIREMENTS

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

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

Distance Matrix:



	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

HIERARCHICAL CLUSTERING: PROBLEMS AND LIMITATIONS

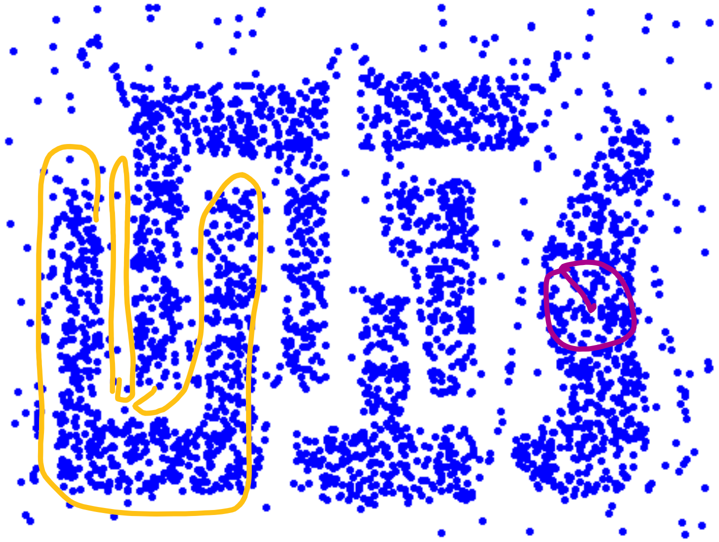
- Once a decision is made to combine two clusters, it cannot be undone
- No global objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise
  - Difficulty handling clusters of different sizes and non-globular shapes
  - Breaking large clusters => Max

CLUSTERING ALGORITHMS

- K-means and its variants
- Hierarchical clustering
- Density-based clustering

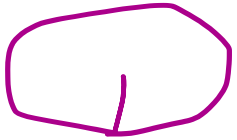
DENSITY BASED CLUSTERING

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



DENSITY-BASED ALGORITHM: DBSCAN

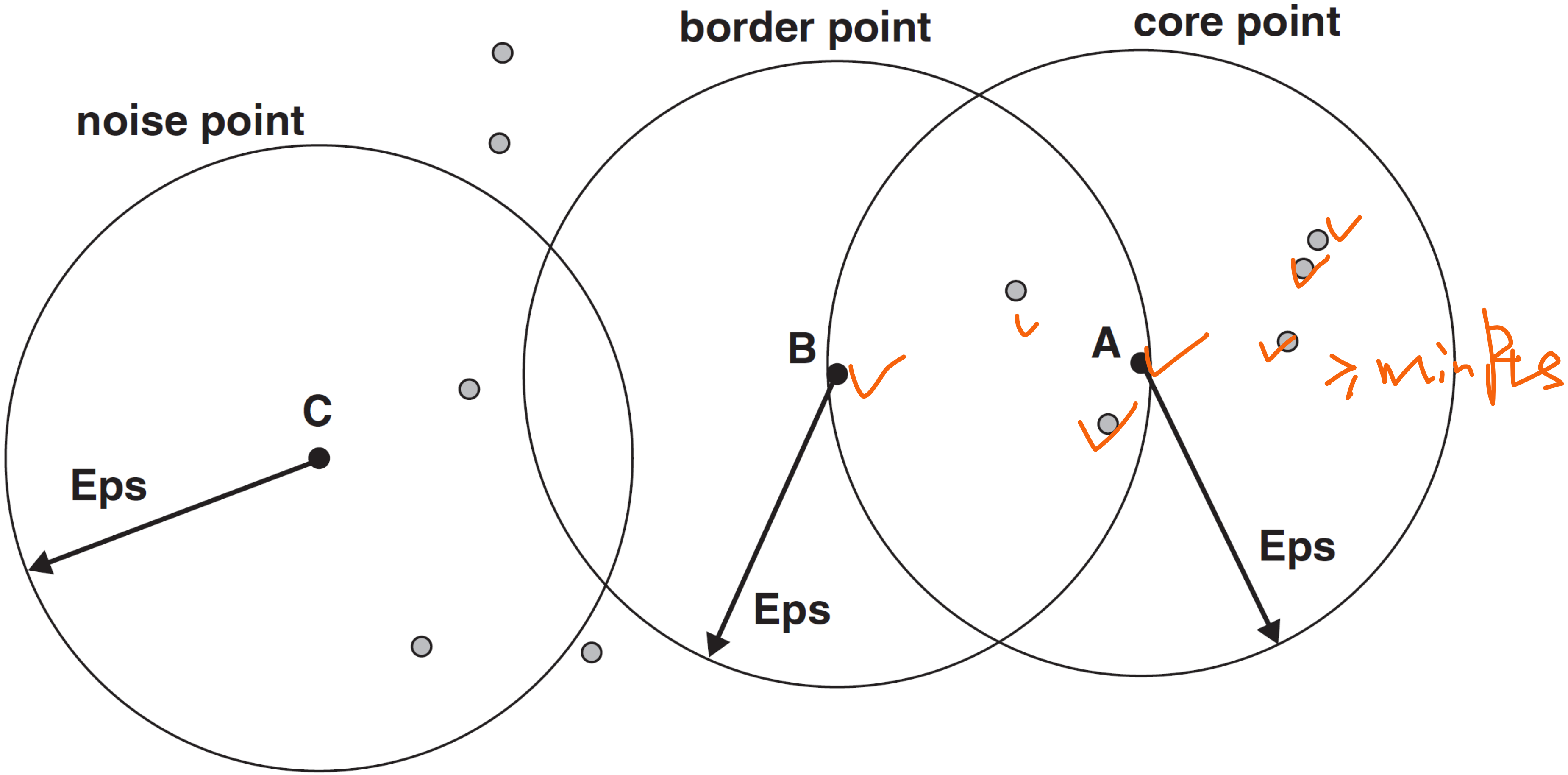
- DBSCAN: Density-based spatial clustering of applications with noise
- Density = number of points within a specified radius ε
- A point is a core point if it has at least a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
  - Counts the point itself



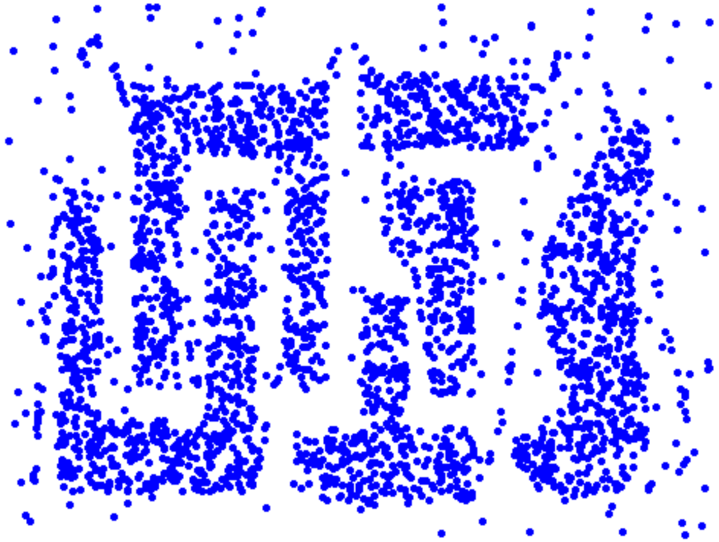
- A border point is not a core point, but is in the neighborhood of a core point
- A noise point is any point that is not a core point or a border point

DBSCAN: CORE, BORDER, AND NOISE POINTS

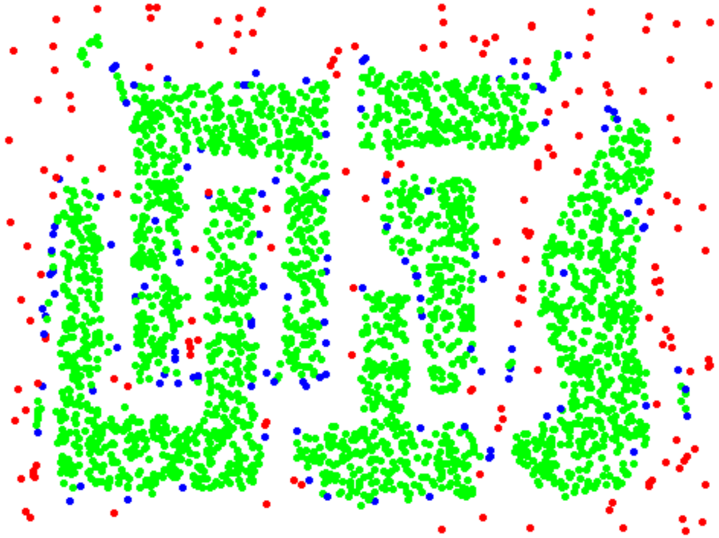
MinPts = 7



DBSCAN: CORE, BORDER AND NOISE POINTS



Original Points



Point types: core, border and noise

Eps = 10, MinPts = 4

DBSCAN 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.

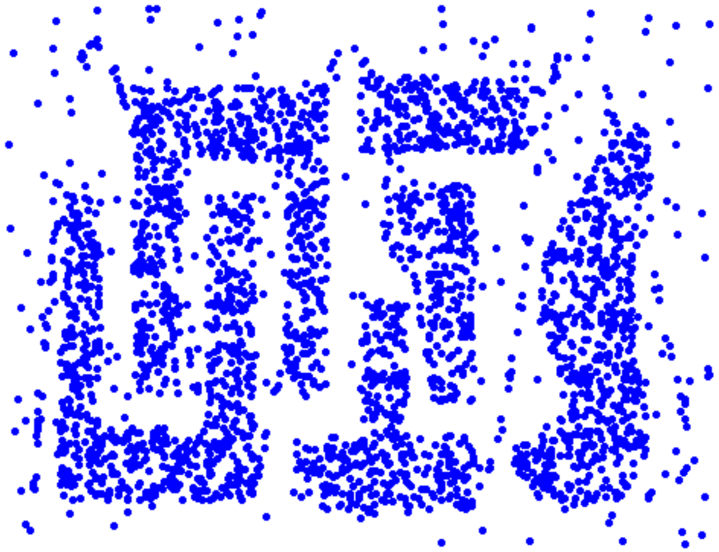
2: Eliminate noise points.

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

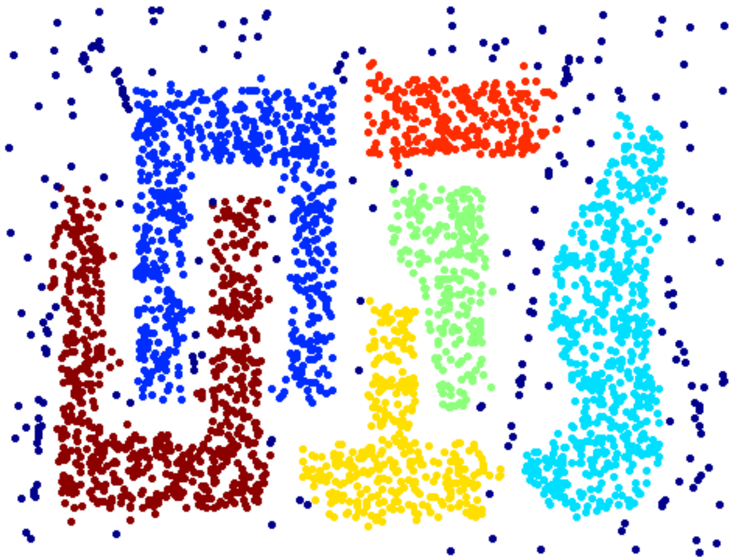
4: Make each group of connected core points into a separate cluster.

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

WHEN DBSCAN WORKS WELL



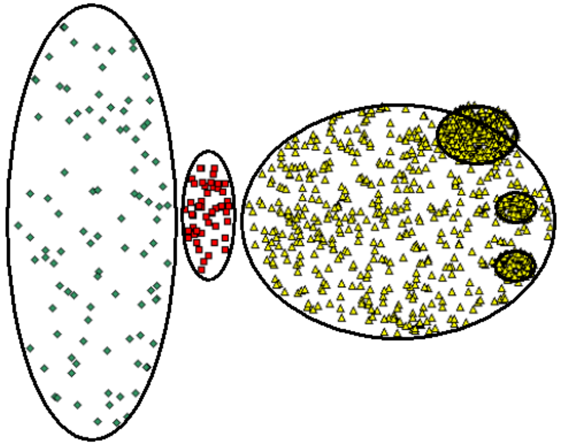
Original Points



Clusters (dark blue points indicate noise)

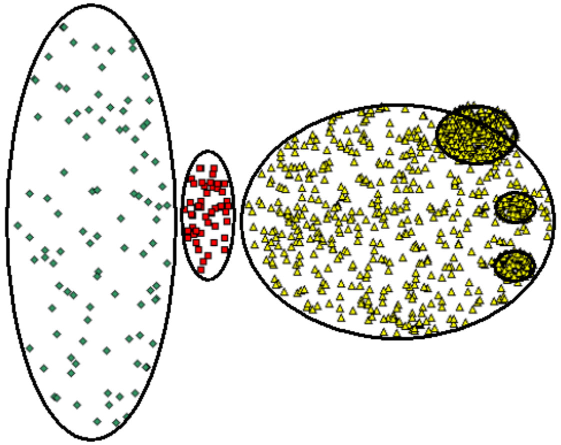
- **Can handle clusters of different shapes and sizes**
- **Resistant to noise**

WHEN DBSCAN DOES NOT WORK WELL



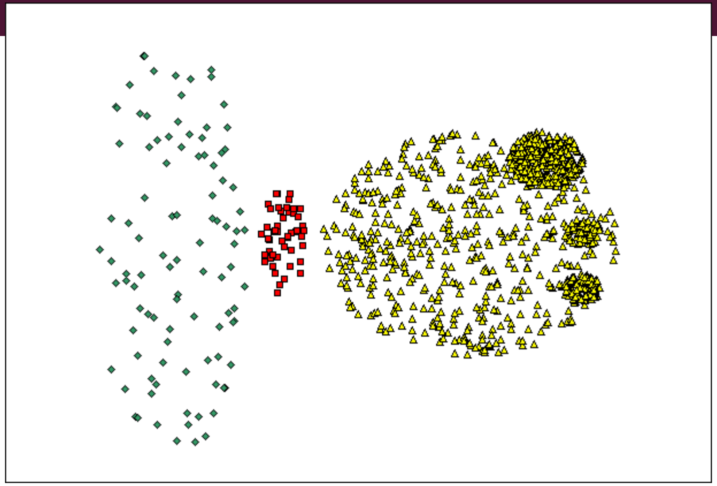
Original Points

WHEN DBSCAN DOES NOT WORK WELL

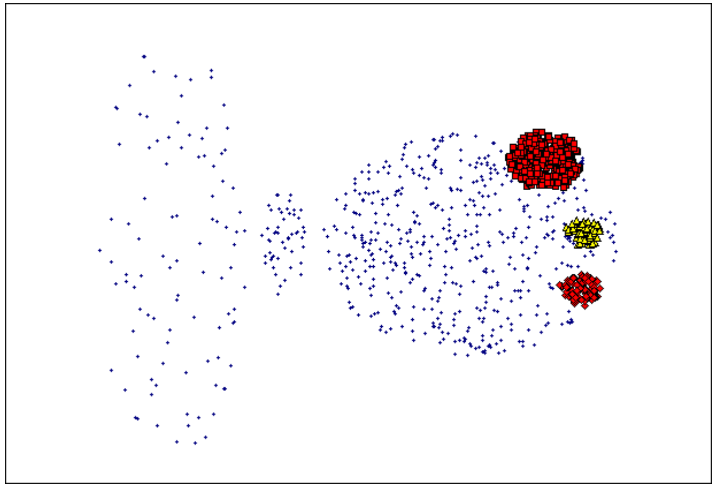


Original Points

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



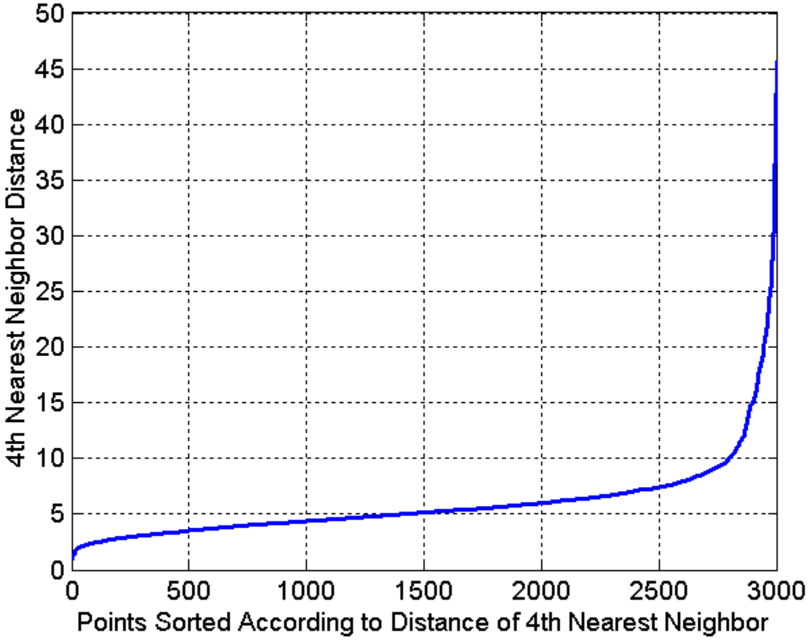
(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

DBSCAN: DETERMINING ε AND MINPTS

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



Would the graph be from the distance matrix?

1. Distance matrix – hierachiracal clustering
2. (proximity matrix)
3. The graph: density-based clustering

CLUSTER VALIDITY

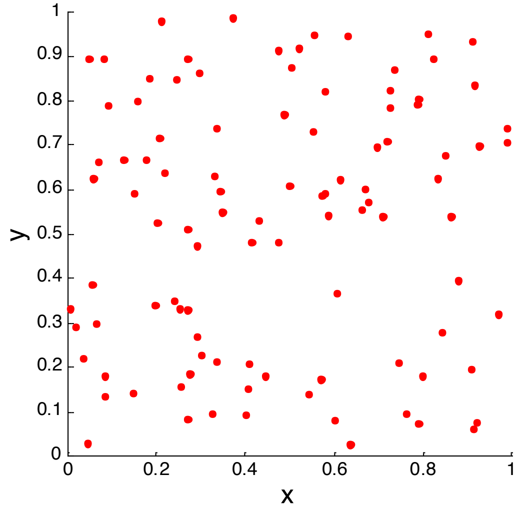
- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall (confusing matrix)

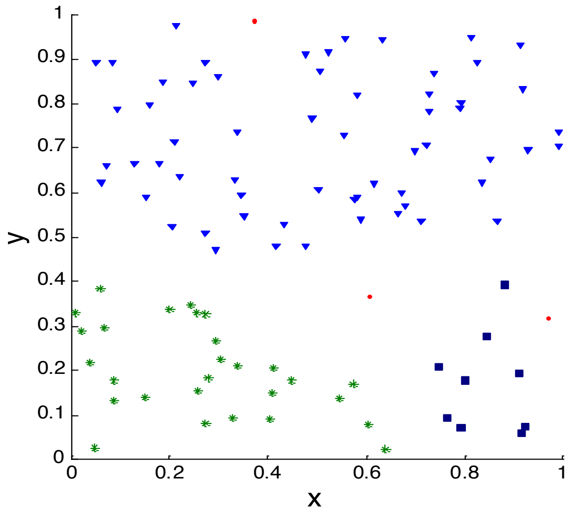
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?
- But "clusters are in the eye of the beholder"!
  - In practice the clusters we find are defined by the clustering algorithm

- 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

CLUSTERS FOUND IN RANDOM DATA

Random Points



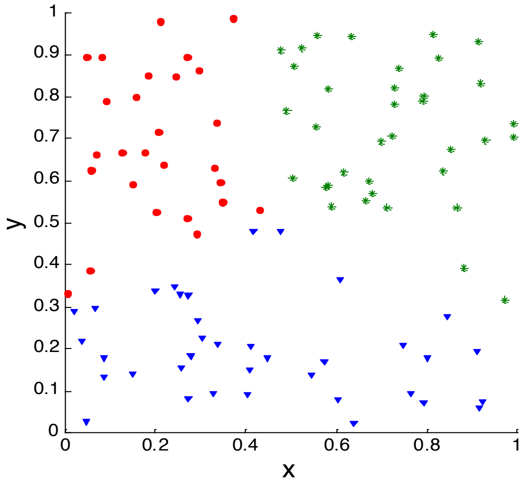


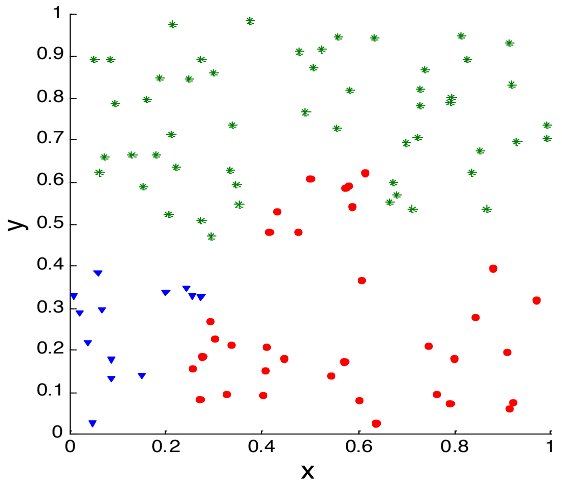
Density

DBSCAN

(density-based)

K-means





Max-D

Complete Link

Hie

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

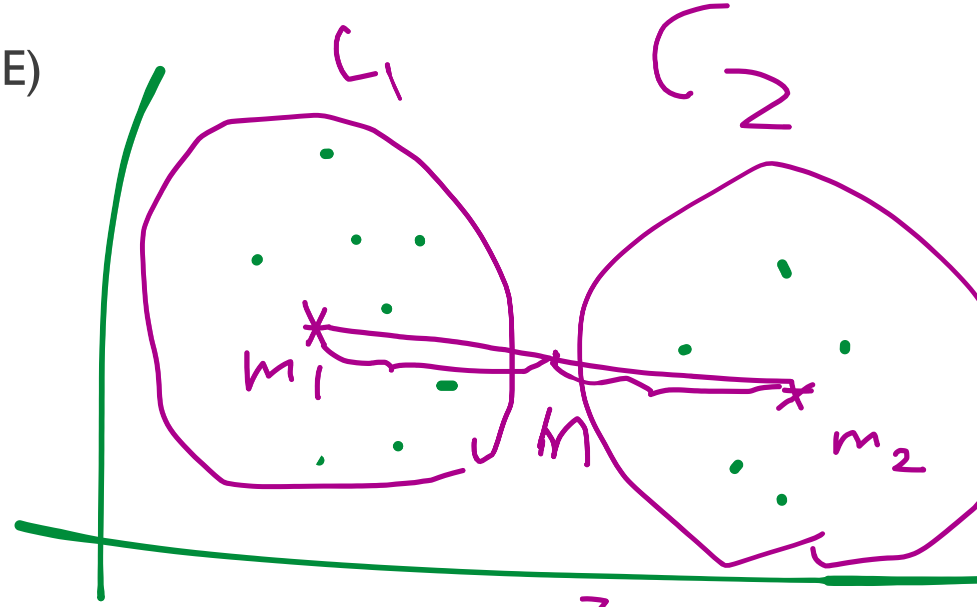
UNSUPERVISED MEASURES: COHESION AND SEPARATION

- 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

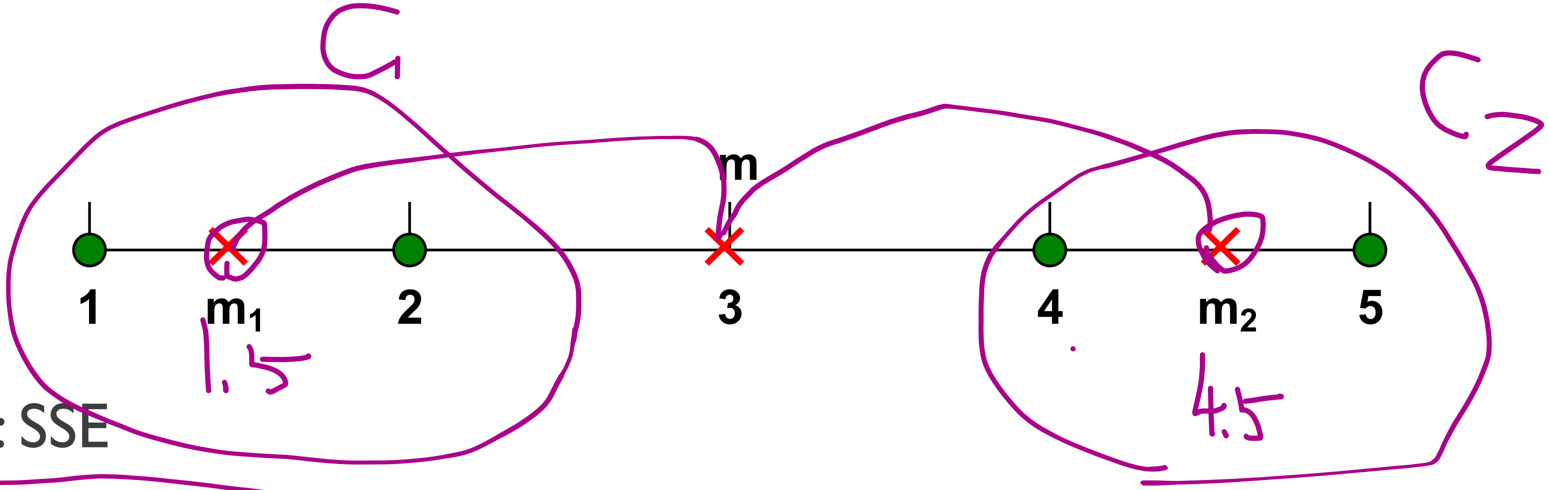
$$SSE = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

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



$8(m_1 - m)^2 + 5(m - m_2$

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:

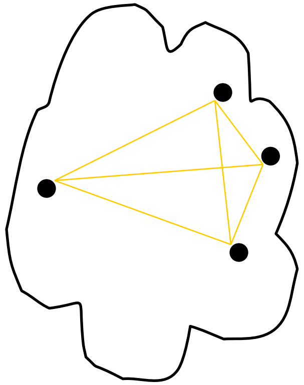
$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

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

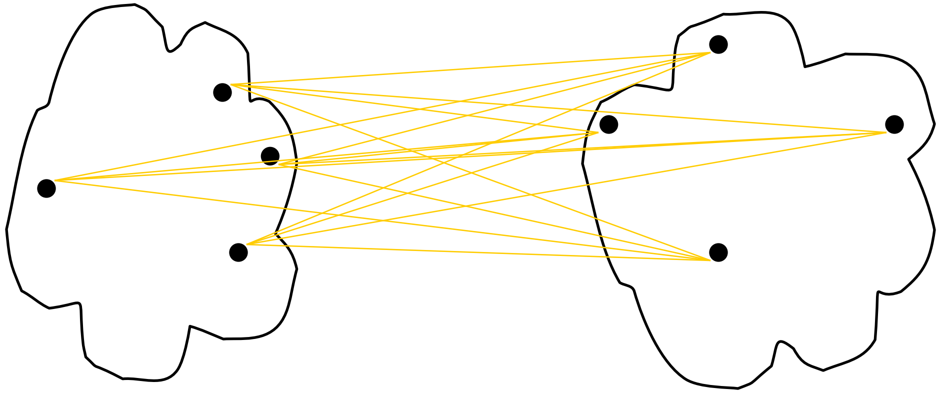
$$Total = 1 + 9 = 10$$

UNSUPERVISED MEASURES: COHESION AND SEPARATION

- A proximity graph-based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



separation

UNSUPERVISED MEASURES: SILHOUETTE COEFFICIENT

- Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*

  $i = 1$ C_1: $a = \frac{d(P_1 - P_4) + \cdots + d(P_3, P_4)}{3}$

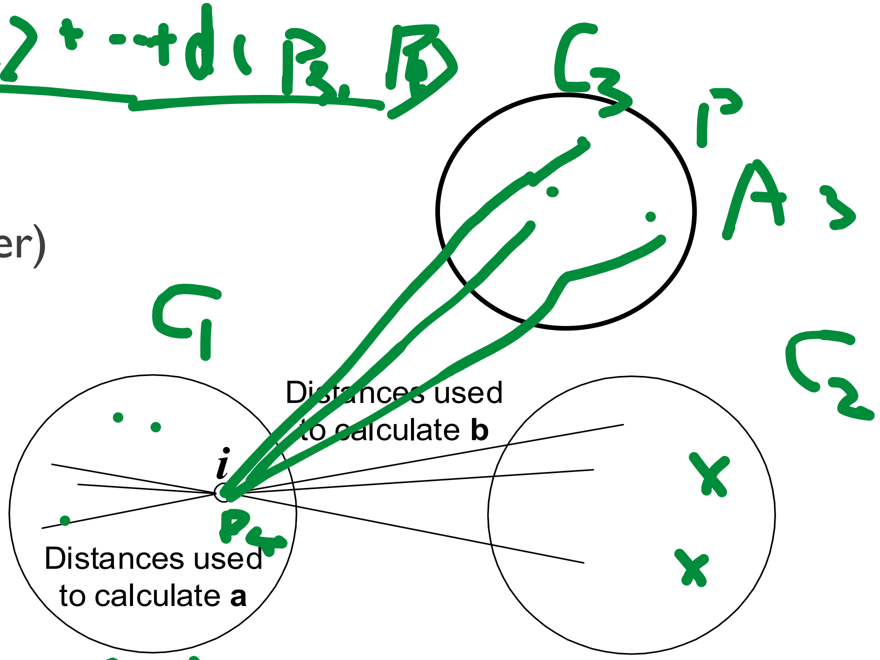
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

    s = (b – a) / max(a,b)

    $A_3 = \frac{d(P_4, P_5) + \cdots}{3}$

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

    $b = \text{Min}(A_3, A_2)$



C_3
P
A_3
C_1
C_2
i
P_4
Distances used to calculate **b**
Distances used to calculate **a**

$A_2 = \frac{d(P_4, P_7) + d}{2}$

- Can calculate the average silhouette coefficient for a cluster or a clustering