

Data Mining
Cluster Analysis: Basic Concepts
and Algorithms

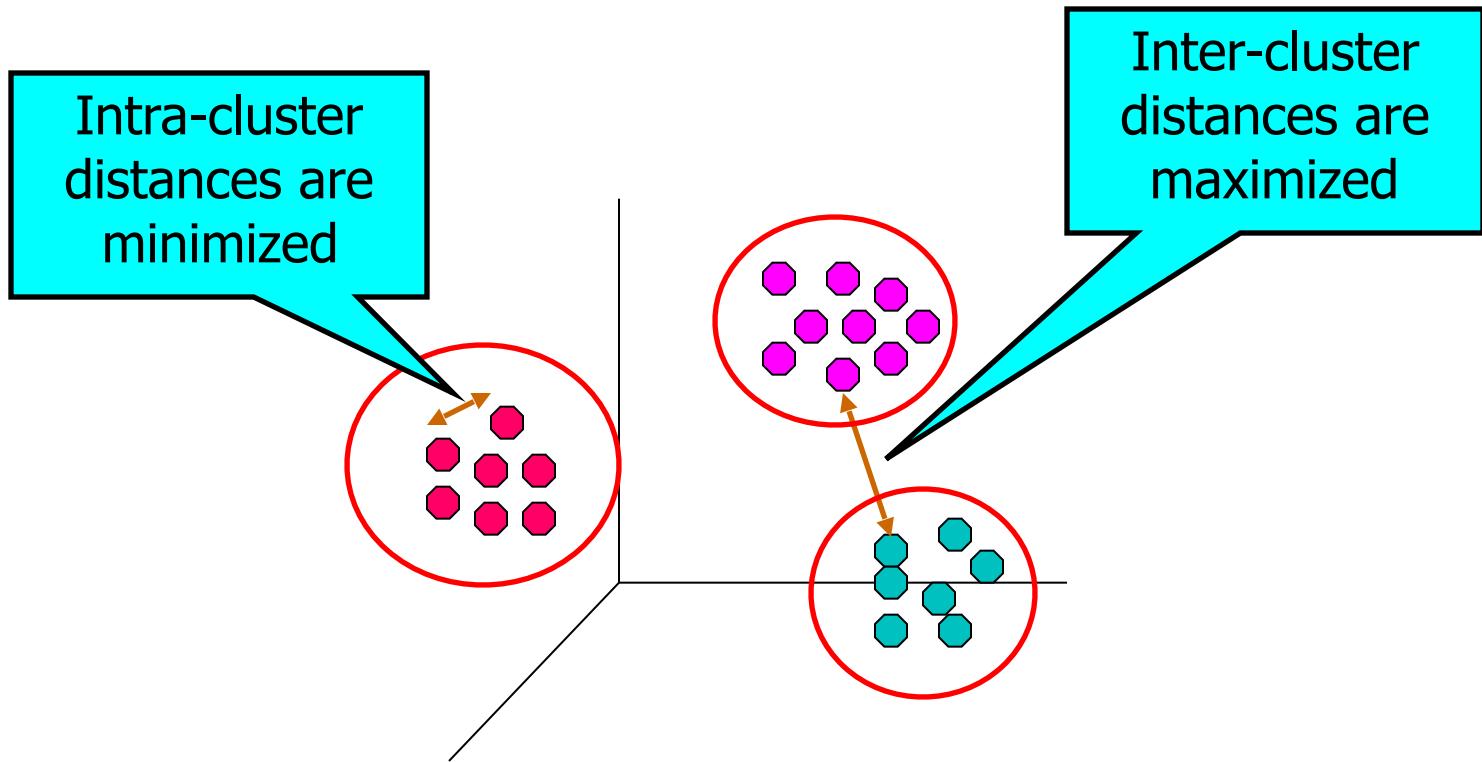
Lecture Notes for Chapter 8

Introduction to Data Mining
by
Tan, Steinbach, Kumar

dino.ienco@irstea.fr

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



Applications of Cluster Analysis

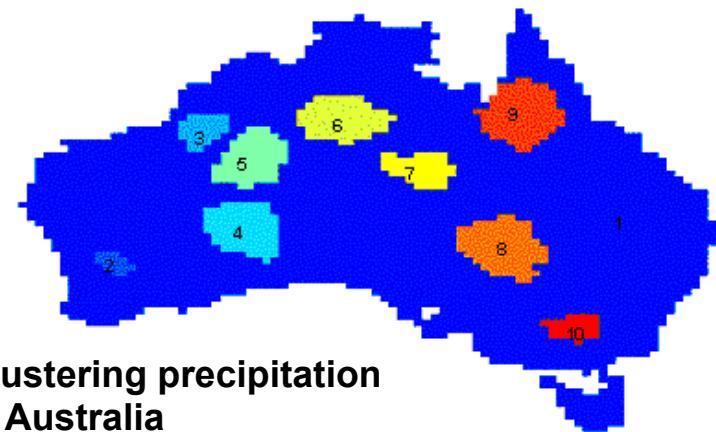
● Understanding

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

	<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

● Summarization

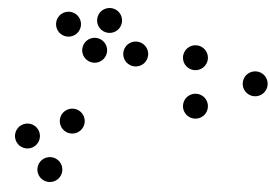
- Reduce the size of large data sets



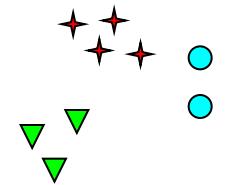
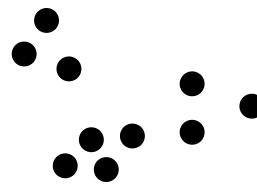
What is not Cluster Analysis?

- Supervised classification
 - Have class label information
- Simple segmentation
 - Dividing students into different registration groups alphabetically, by last name
- Results of a query
 - Groupings are a result of an external specification
- Graph partitioning
 - Some mutual relevance and synergy, but areas are not identical

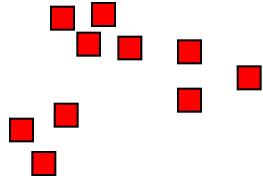
Notion of a Cluster can be Ambiguous



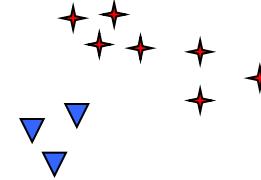
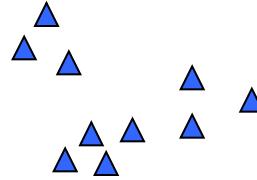
How many clusters?



Six Clusters



Two Clusters

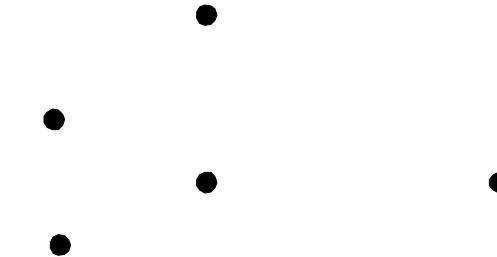
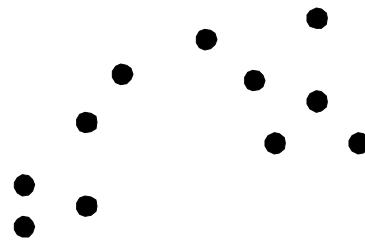


Four Clusters

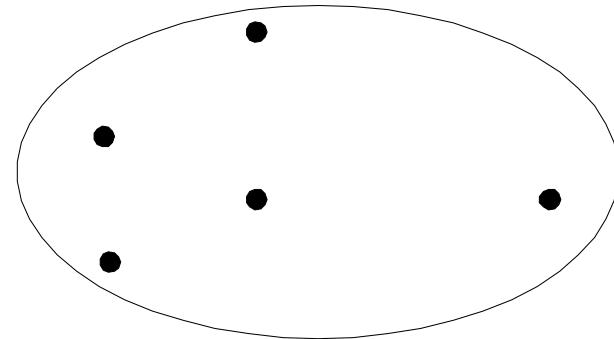
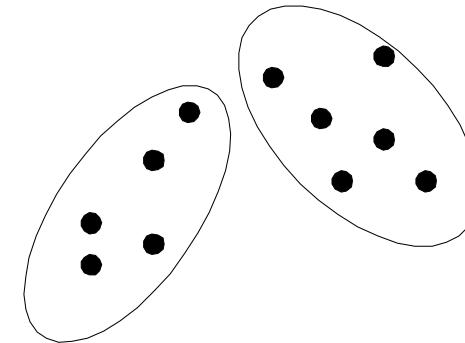
Types of Clusterings

- A **clustering** is a set of clusters
- Important distinction between **hierarchical** and **partitional** sets of clusters
- **Partitional Clustering**
 - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- **Hierarchical clustering**
 - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

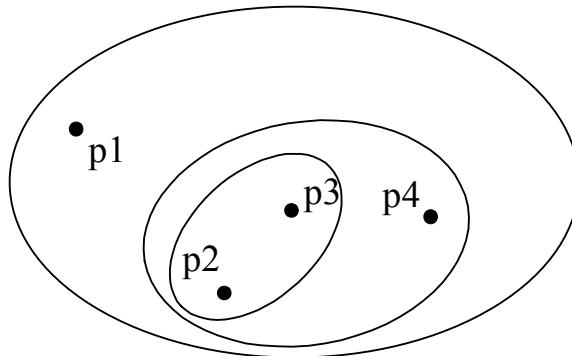


Original Points

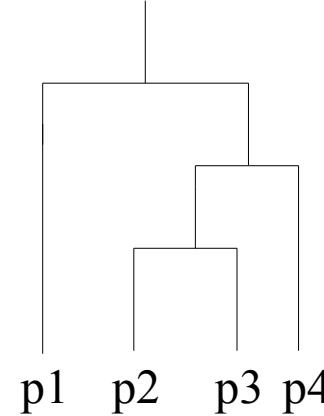


A Partitional Clustering

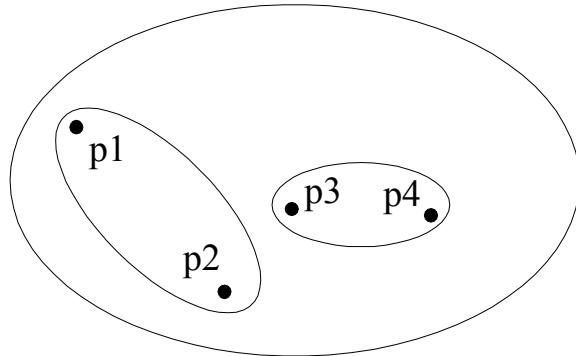
Hierarchical Clustering



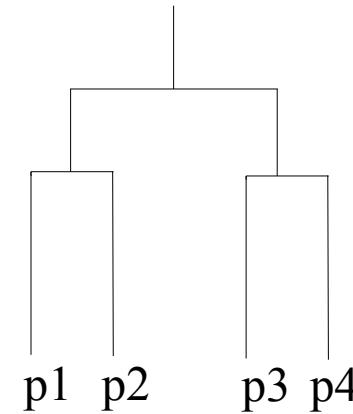
Traditional Hierarchical Clustering



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 represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
 - 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
- Heterogeneous versus homogeneous
 - Cluster of widely different sizes, shapes, and densities

Characteristics of the Input Data Are Important

- Type of proximity or density measure
 - This is a derived measure, but central to clustering
- Sparseness
 - Dictates type of similarity
 - Adds to efficiency
- Attribute type
 - Dictates type of similarity, distance for categorical, numerical and ordered attributes
- Dimensionality
 - Document high

Type of proximity or distance measures

GIVEN VECTORS $X = (X_1, \dots, X_N)$, $Y = (Y_1, \dots, Y_N)$

- EUCLIDEAN DISTANCE:

$$d_E(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- COSINE DISTANCE:

$$d_C(x,y) = 1 - \frac{\sum x_i^* y_i}{\sqrt{\sum (x_i)^2} * \sqrt{\sum (y_i)^2}}$$

- JACCARD DISTANCE:

$$d_J(x,y) = 1 - \frac{|x \cap y|}{|x \cup y|}$$

Clustering Algorithms

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

K-means (Partitional Clustering)

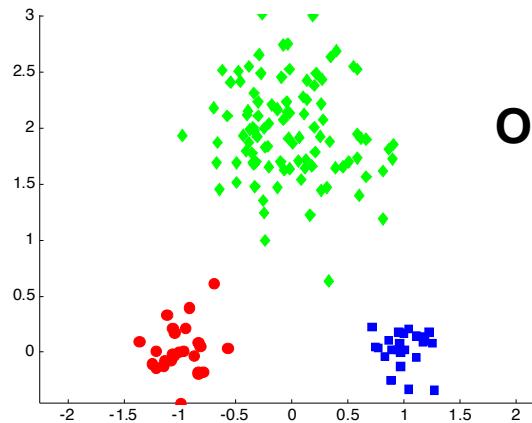
- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

-
- 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
-

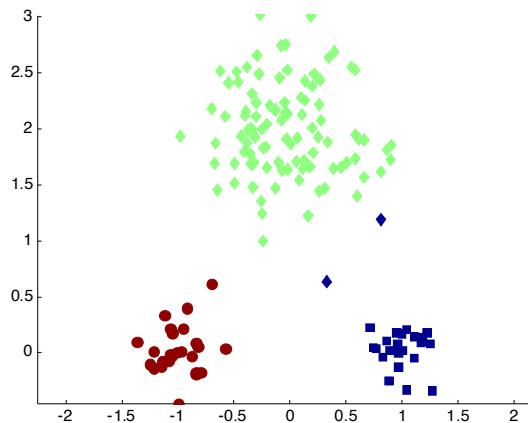
K-means Clustering – Details

- 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.
 - 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

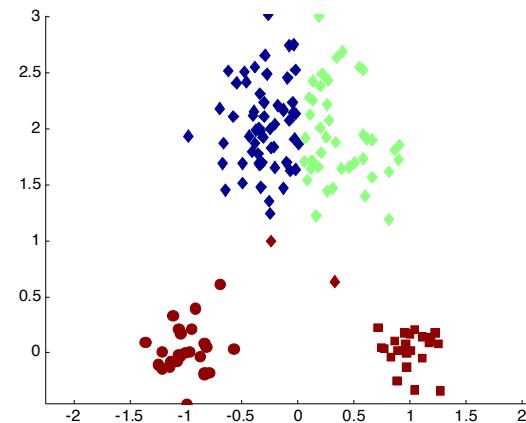
Two different K-means Clusterings



Original Points

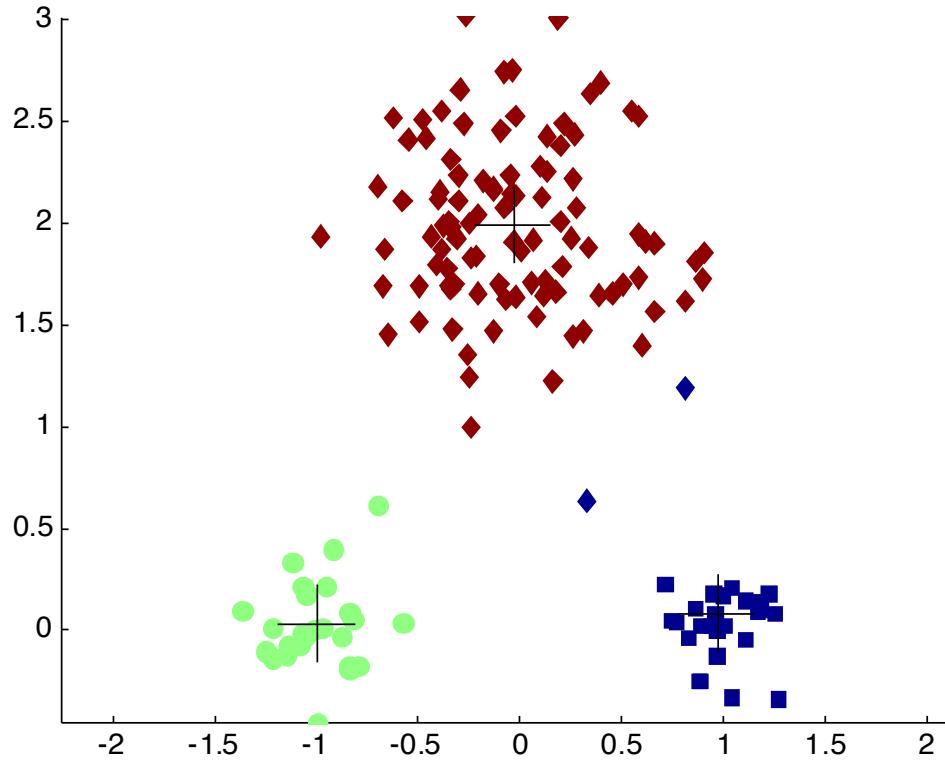


Optimal Clustering

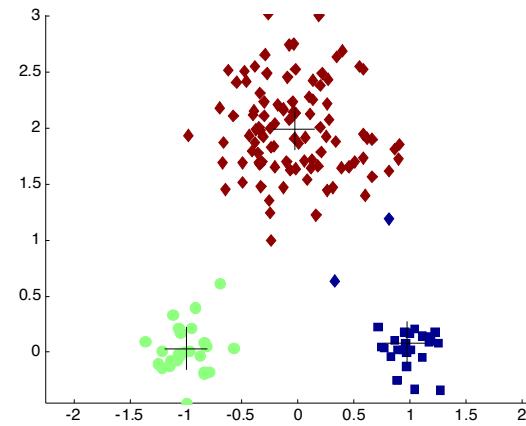
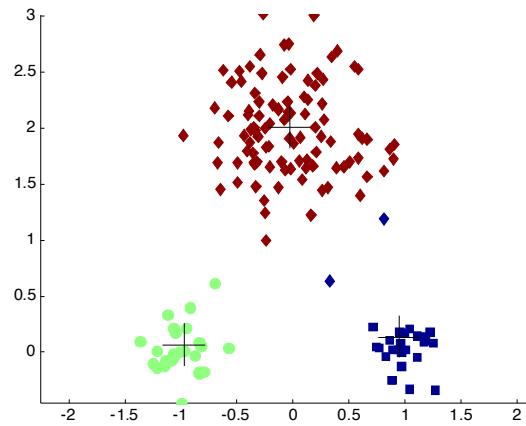
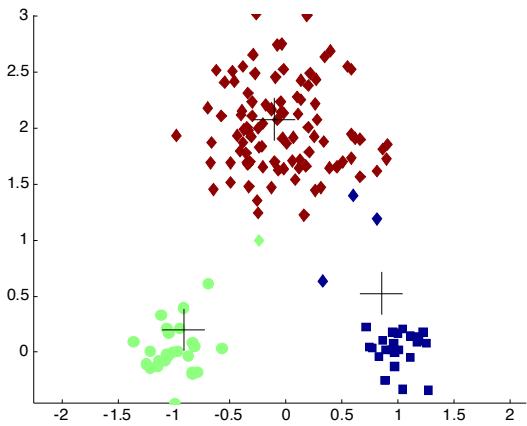
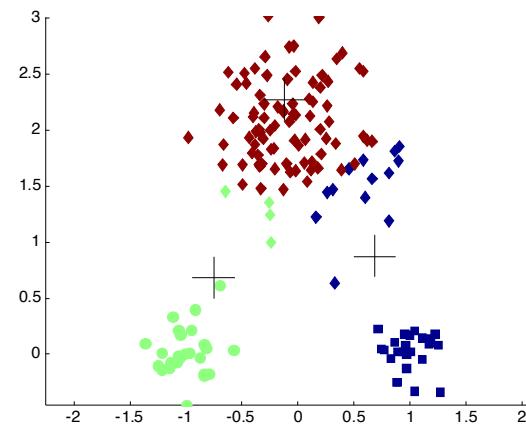
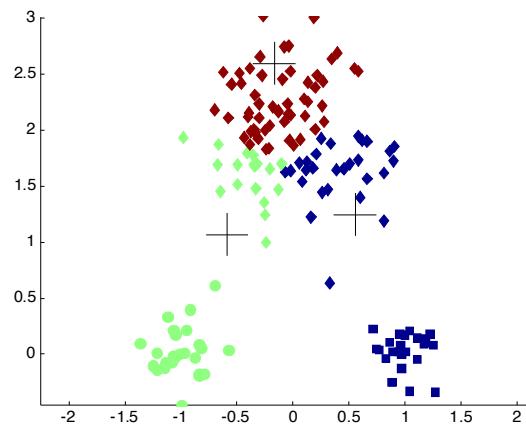
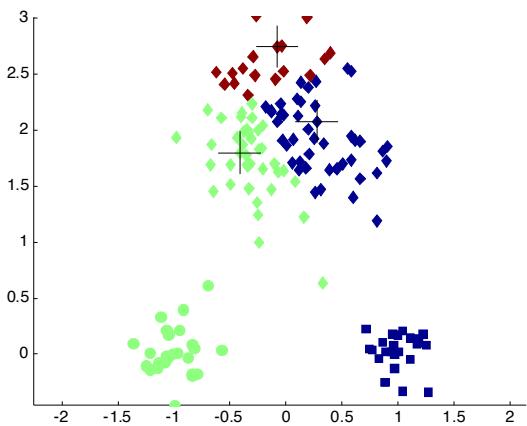


Sub-optimal Clustering

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
 - For each point, the error is the distance to the nearest cluster
 - 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 representative point for cluster C_i
 - ◆ can show that m_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 , the number of clusters
 - ◆ A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

K-means Clusters (Example)

The process goes on until the convergence is reached:

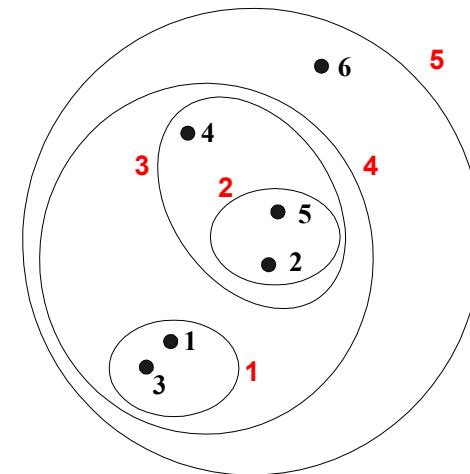
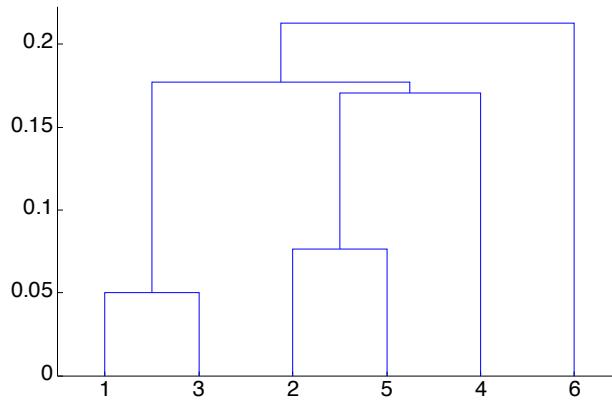
- Predefined number of iteration
- Objective function changement is lesser than a specified threshold

Hierarchical Clustering I

- Two main types of hierarchical clustering
 - Agglomerative:
 - ◆ Start with the points as individual clusters
 - ◆ At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
 - Divisive:
 - ◆ Start with one, all-inclusive cluster
 - ◆ At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
 - Merge or split one cluster at a time

Hierarchical Clustering II

- 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

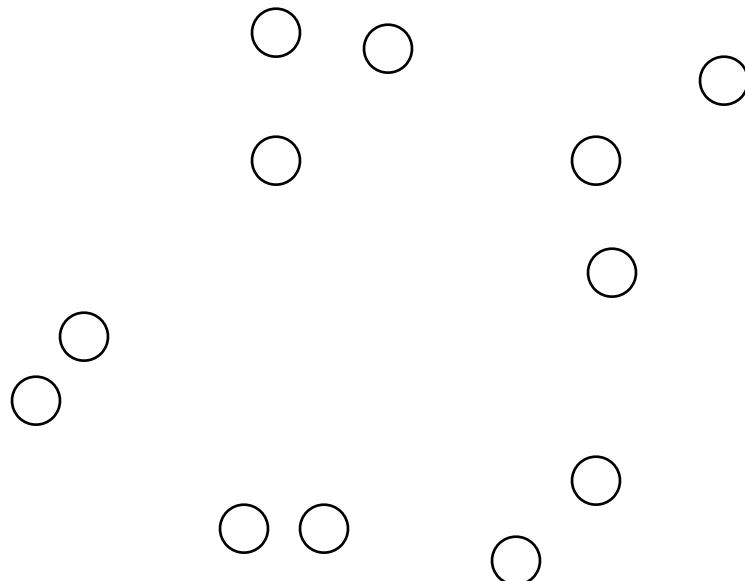
- Do not have to assume any particular number of clusters
 - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
- They may correspond to meaningful taxonomies
 - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
 1. Compute the proximity matrix
 2. Let each data point be a cluster
 3. **Repeat**
 4. Merge the two closest clusters
 5. Update the proximity matrix
 6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
 - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

- Start with clusters of individual points and a proximity matrix

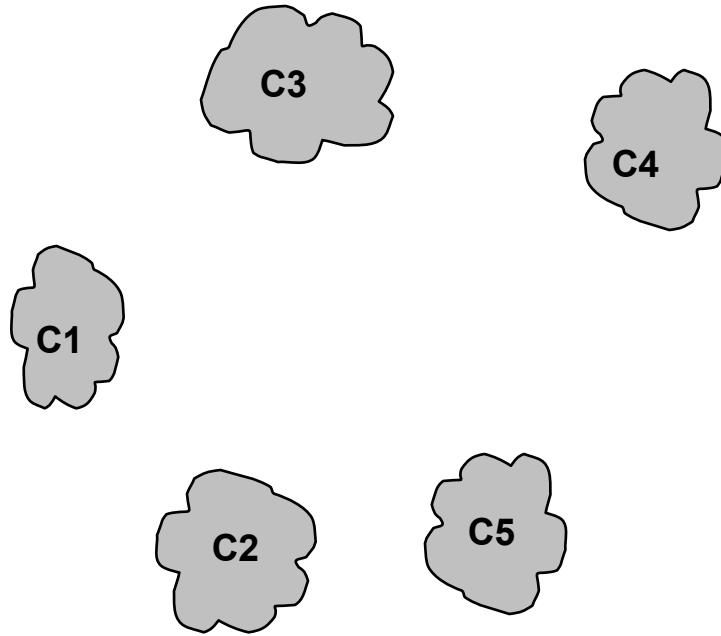


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
Proximity Matrix						

p1 p2 p3 p4 ... p9 p10 p11 p12

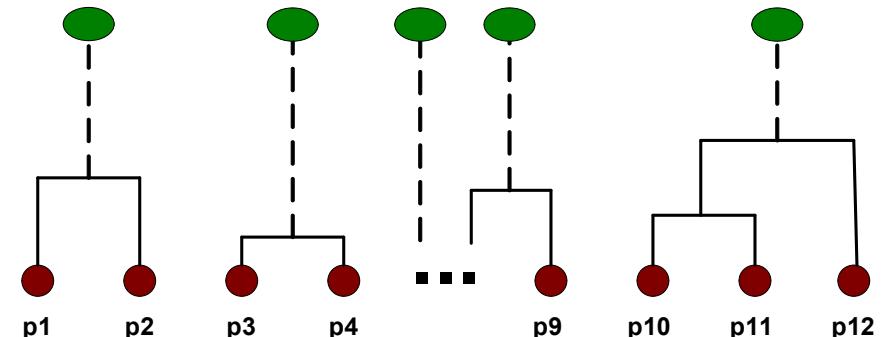
Intermediate Situation

- After some merging steps, we have some clusters



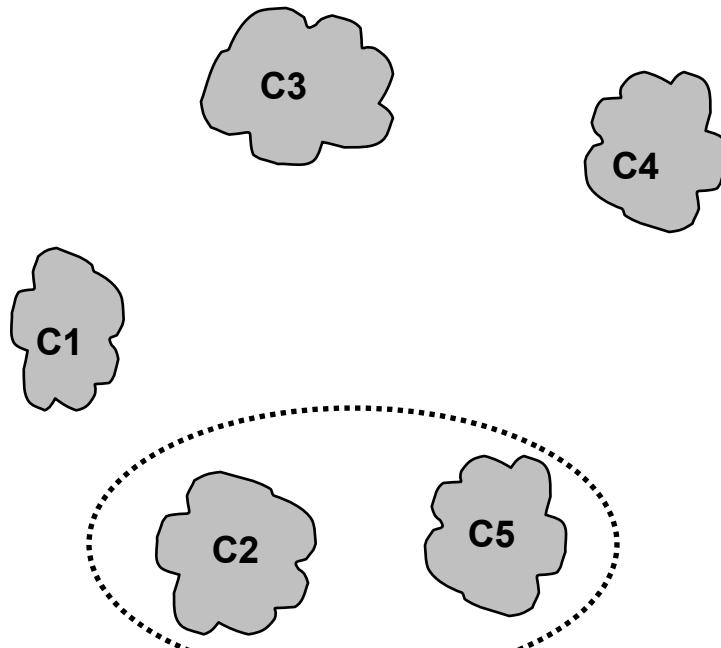
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



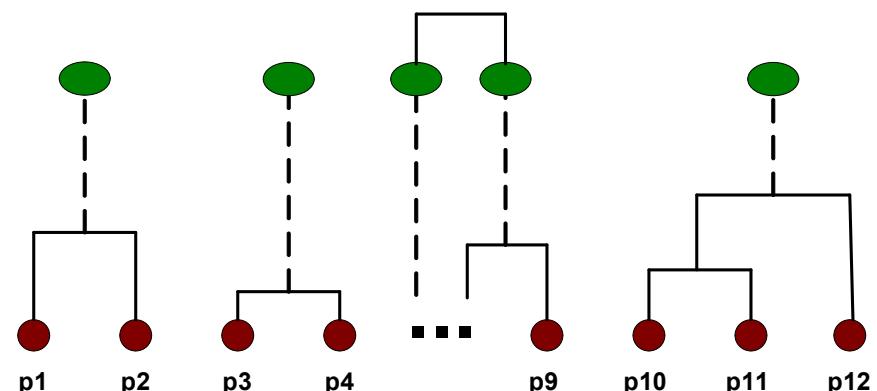
Intermediate Situation

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



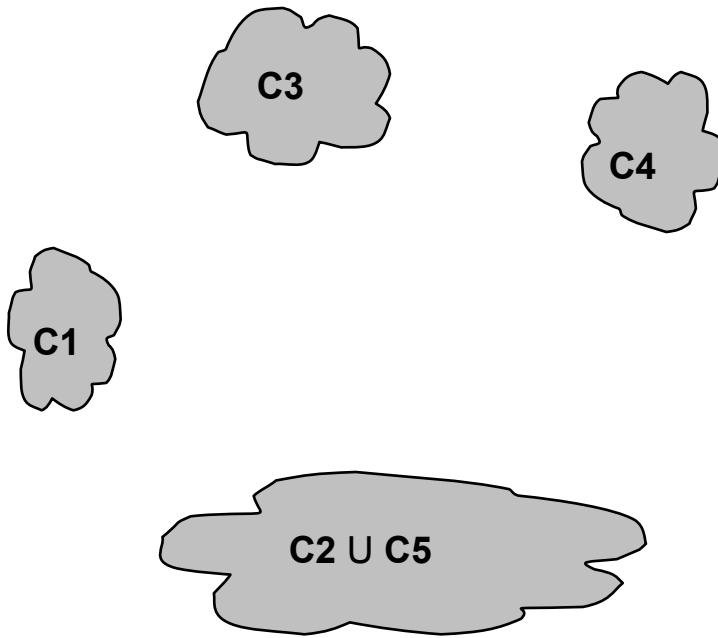
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



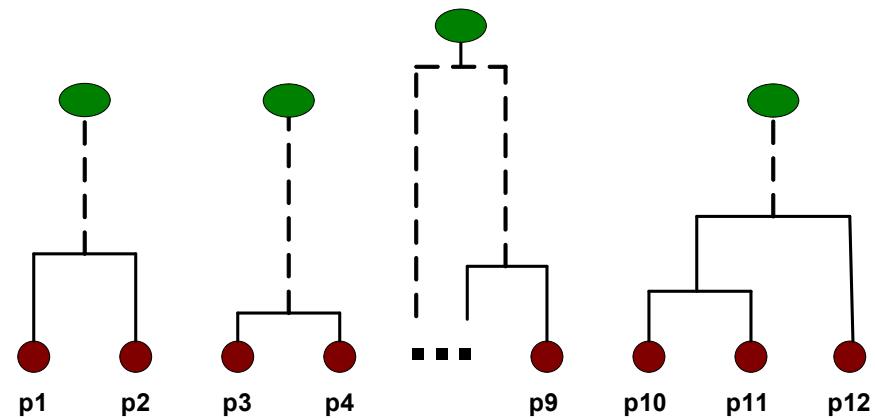
After Merging

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

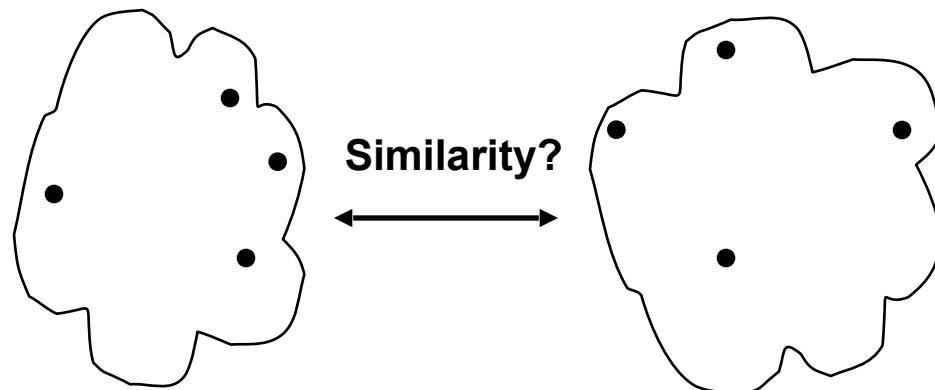


		C1		C3	C4
		C1	?		
C2 U C5		?	?	?	?
		C3	?		
		C4	?		

Proximity Matrix



How to Define Inter-Cluster Similarity

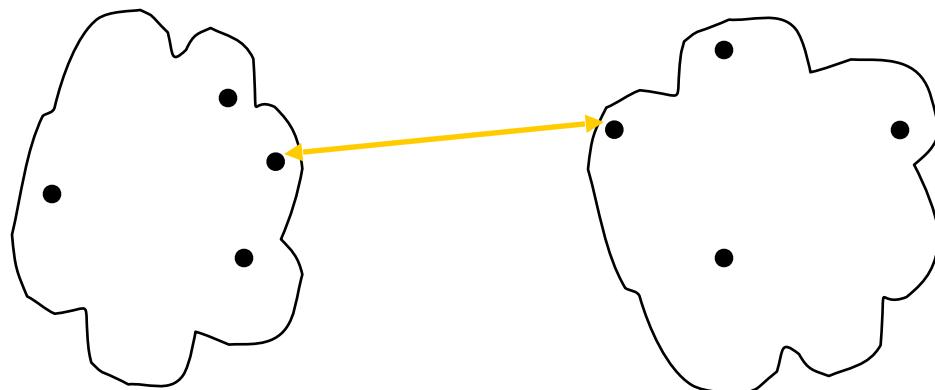


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

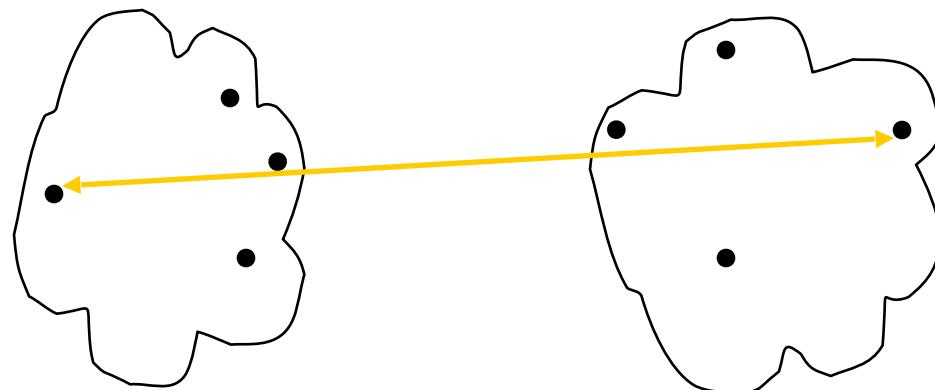


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

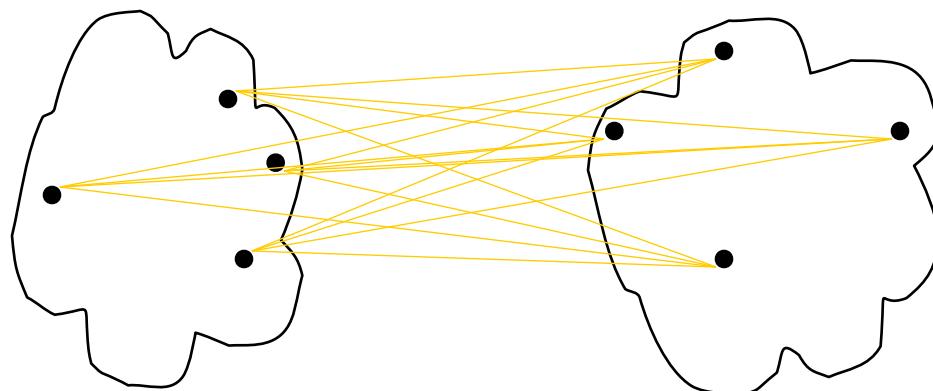


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

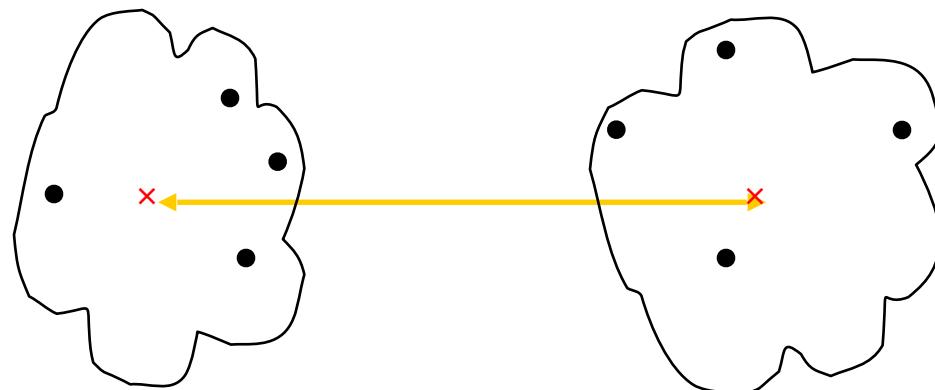


- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

	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

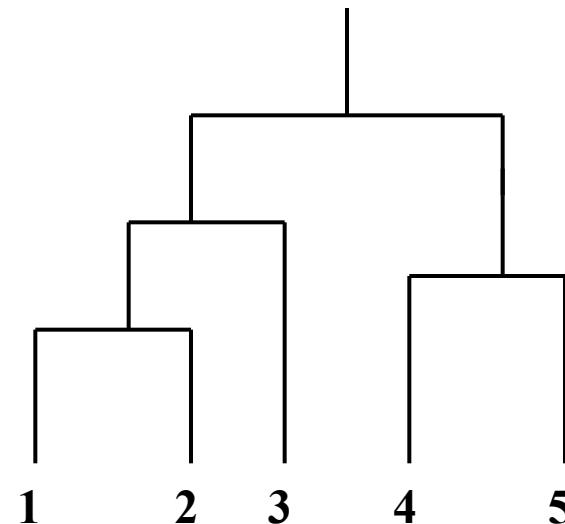
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

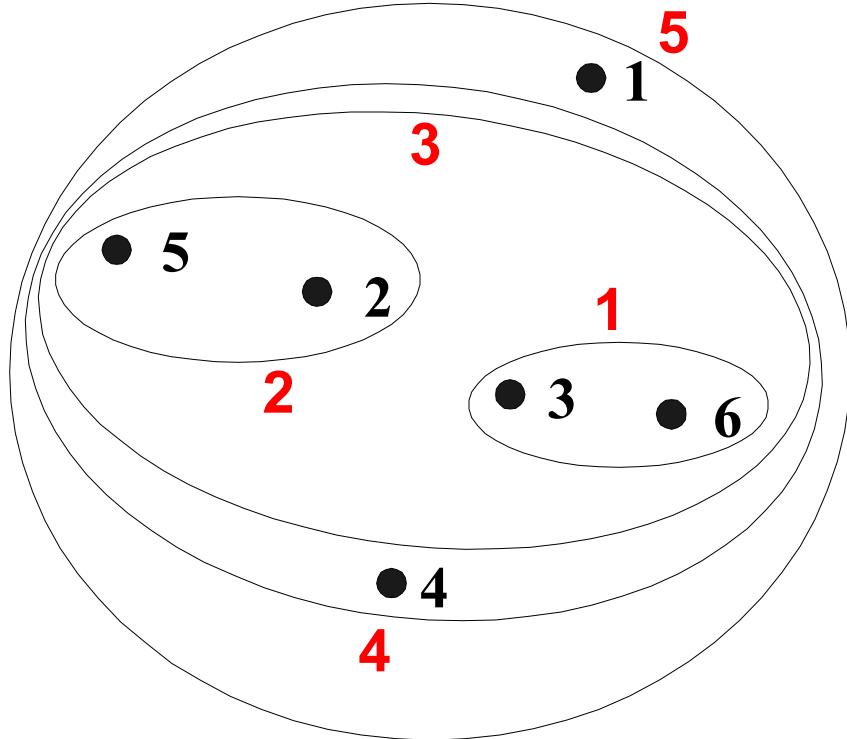
Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
 - Determined by one pair of points, i.e., by one link in the proximity graph.

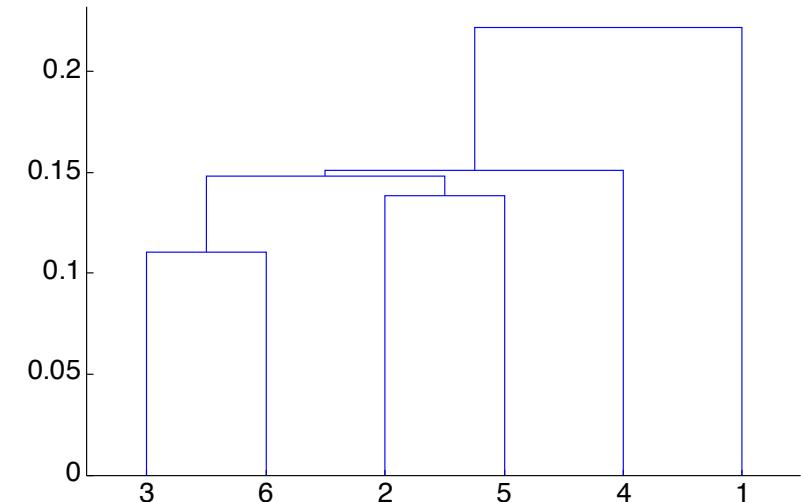
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

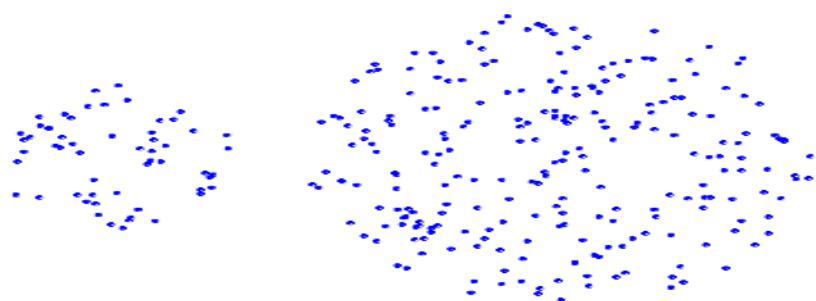


Nested Clusters

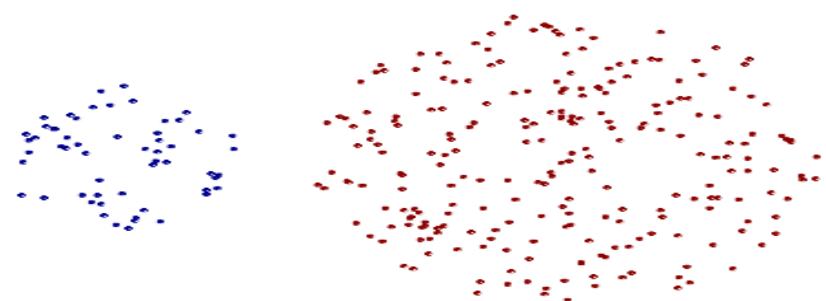


Dendrogram

Strength of MIN



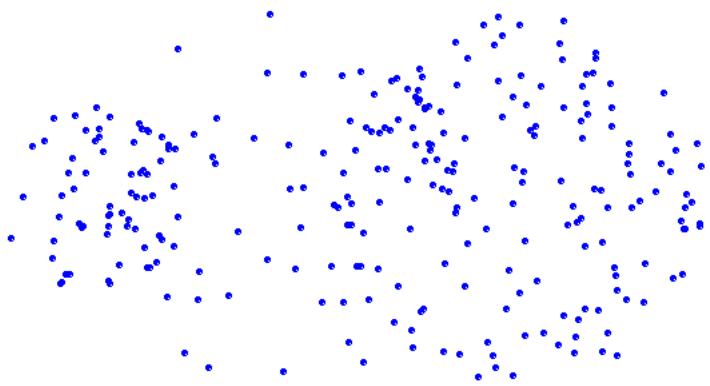
Original Points



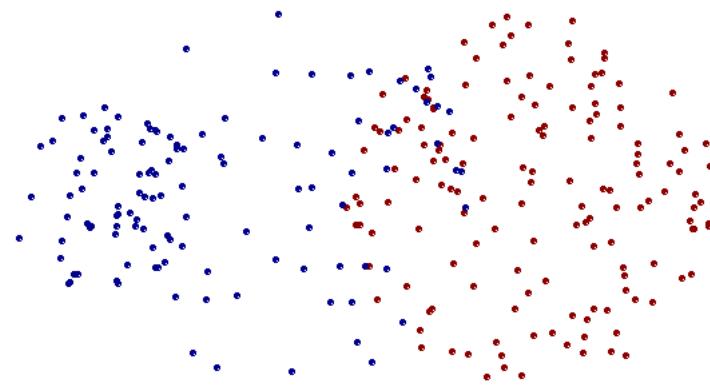
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



Original Points



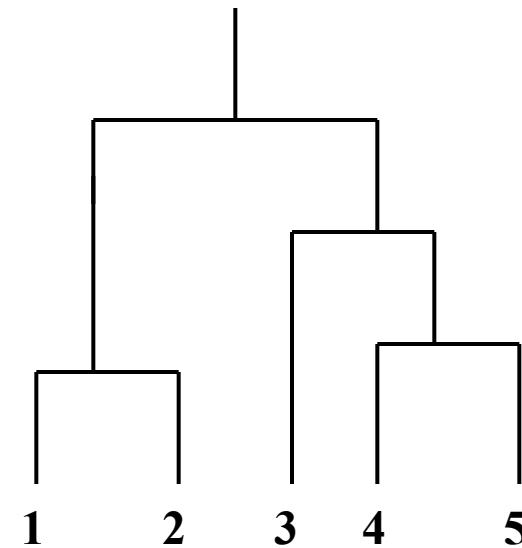
Two Clusters

- Sensitive to noise and outliers

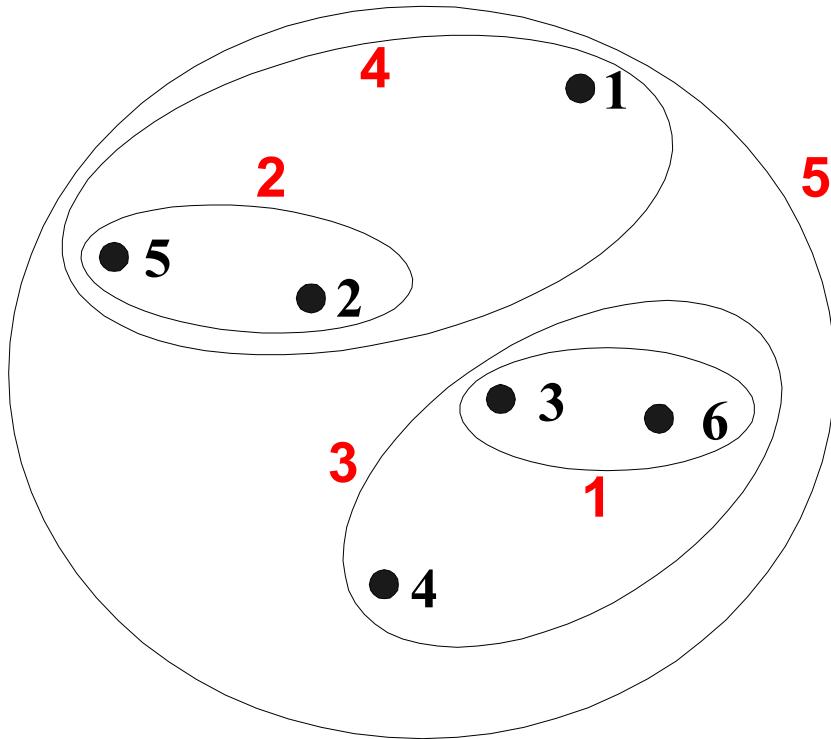
Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
 - Determined by all pairs of points in the two clusters

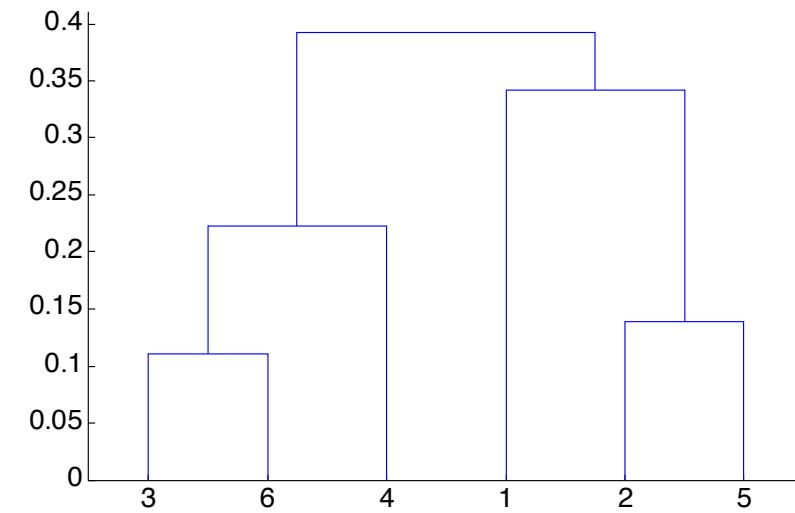
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

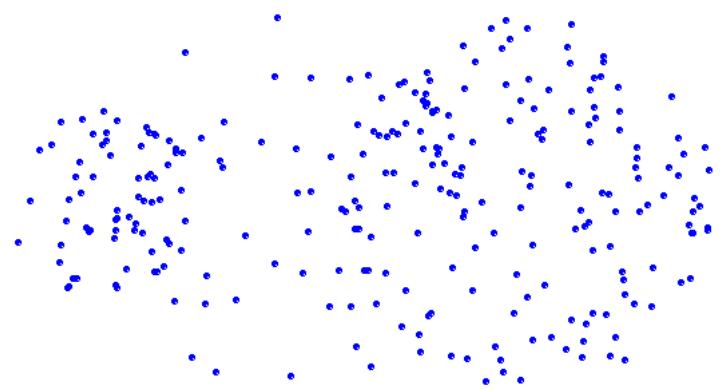


Nested Clusters

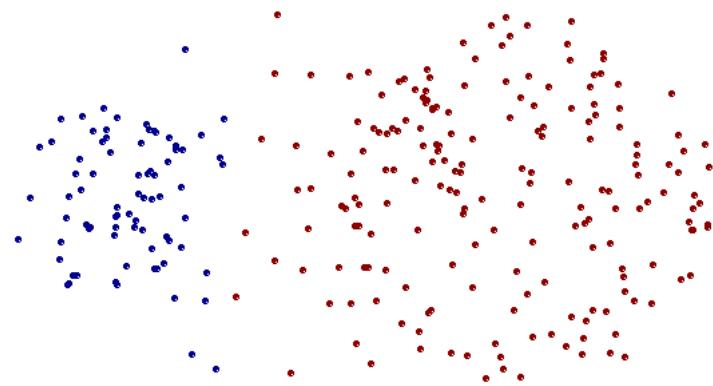


Dendrogram

Strength of MAX



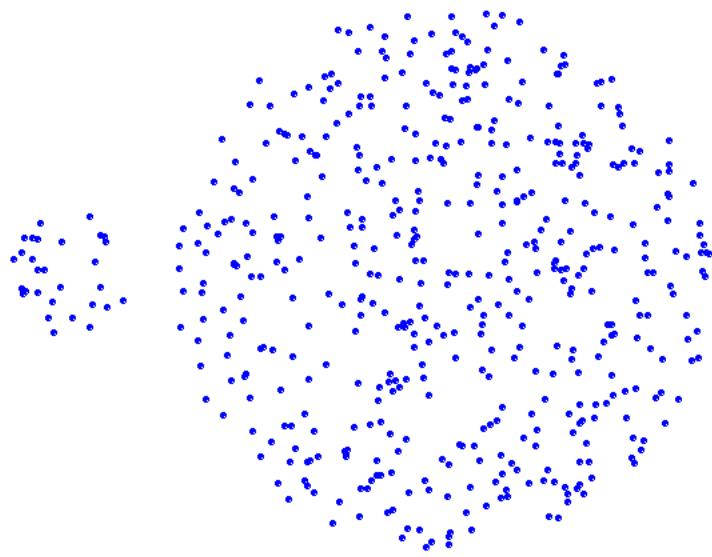
Original Points



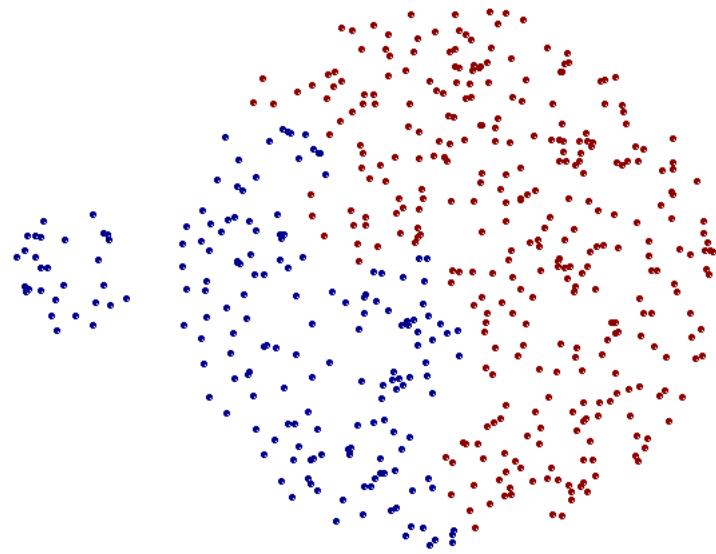
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

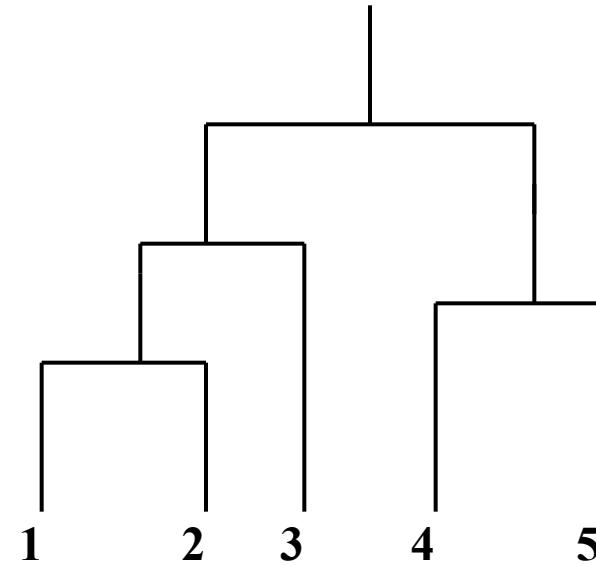
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

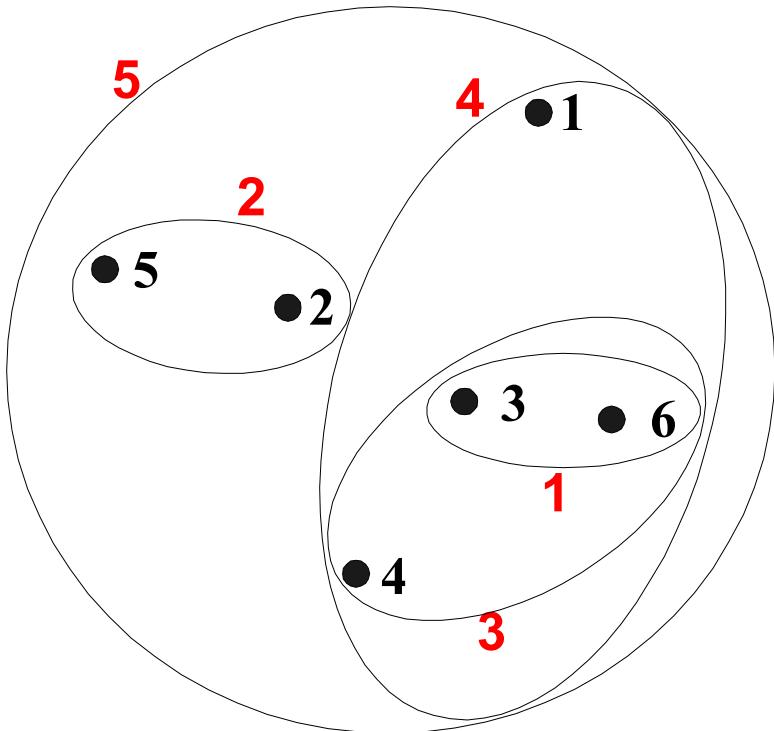
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

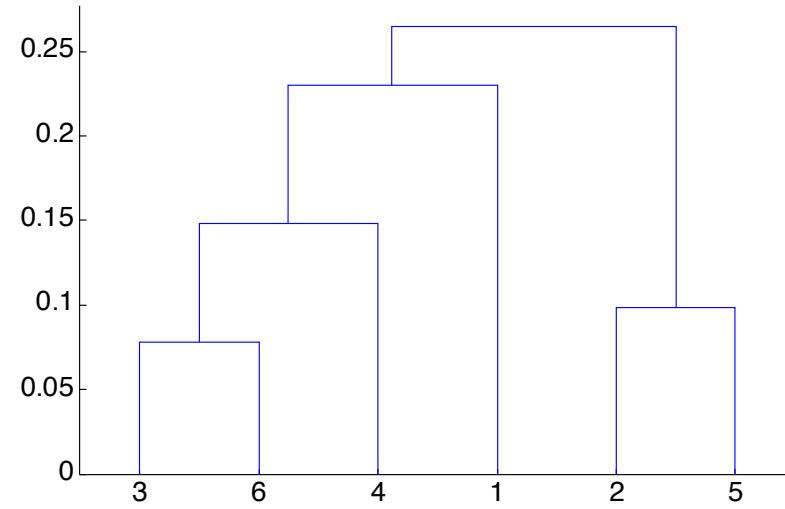
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



Nested Clusters

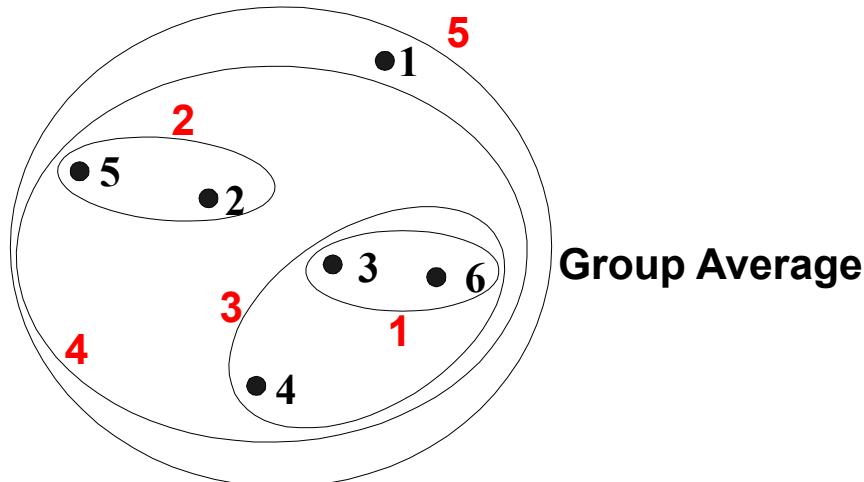
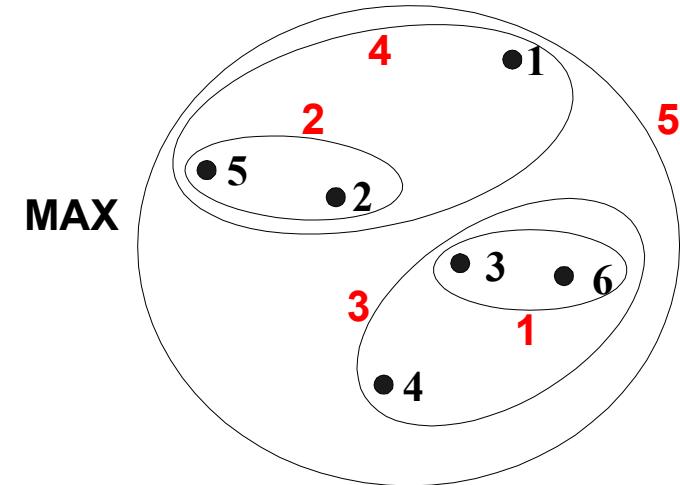
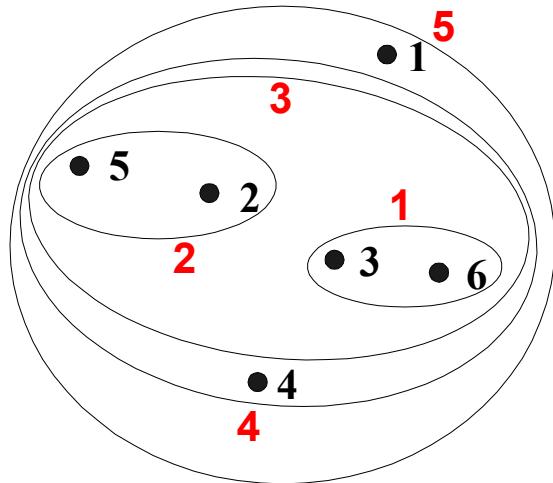


Dendrogram

Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link
- Strengths
 - Less susceptible to noise and outliers
- Limitations
 - Biased towards globular clusters

Hierarchical Clustering: Comparison



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 for some approaches

Hierarchical Clustering: Problems and Limitations

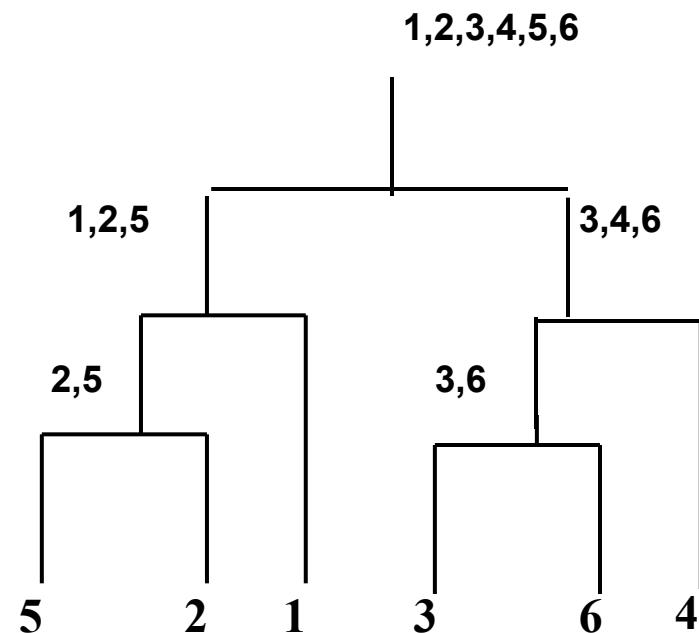
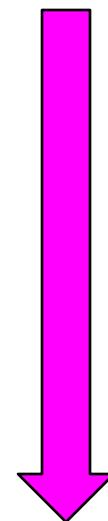
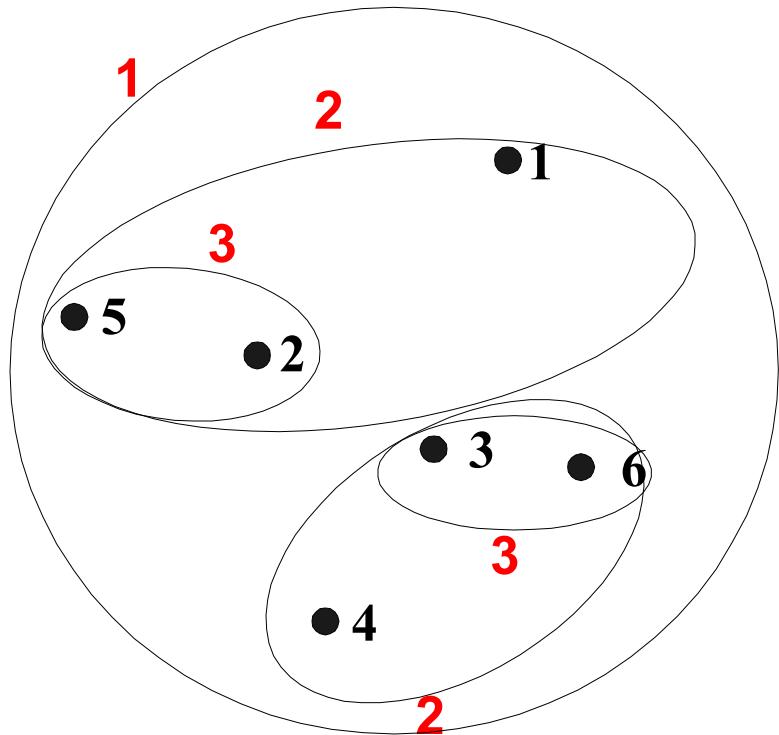
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
 - Sensitivity to noise and outliers
 - Difficulty handling different sized clusters and convex shapes
 - Breaking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

Bisecting K-means (Divisive Hierarchical Clust)

- Bisecting K-means algorithm
 - Variant of K-means that can produce a hierarchical clustering

```
1: Initialize the list of clusters to contain the cluster containing all points.  
2: repeat  
3:   Select a cluster from the list of clusters  
4:   for  $i = 1$  to number_of_iterations do  
5:     Bisect the selected cluster using basic K-means  
6:   end for  
7:   Add the two clusters from the bisection with the lowest SSE to the list of clusters.  
8: until Until the list of clusters contains  $K$  clusters
```

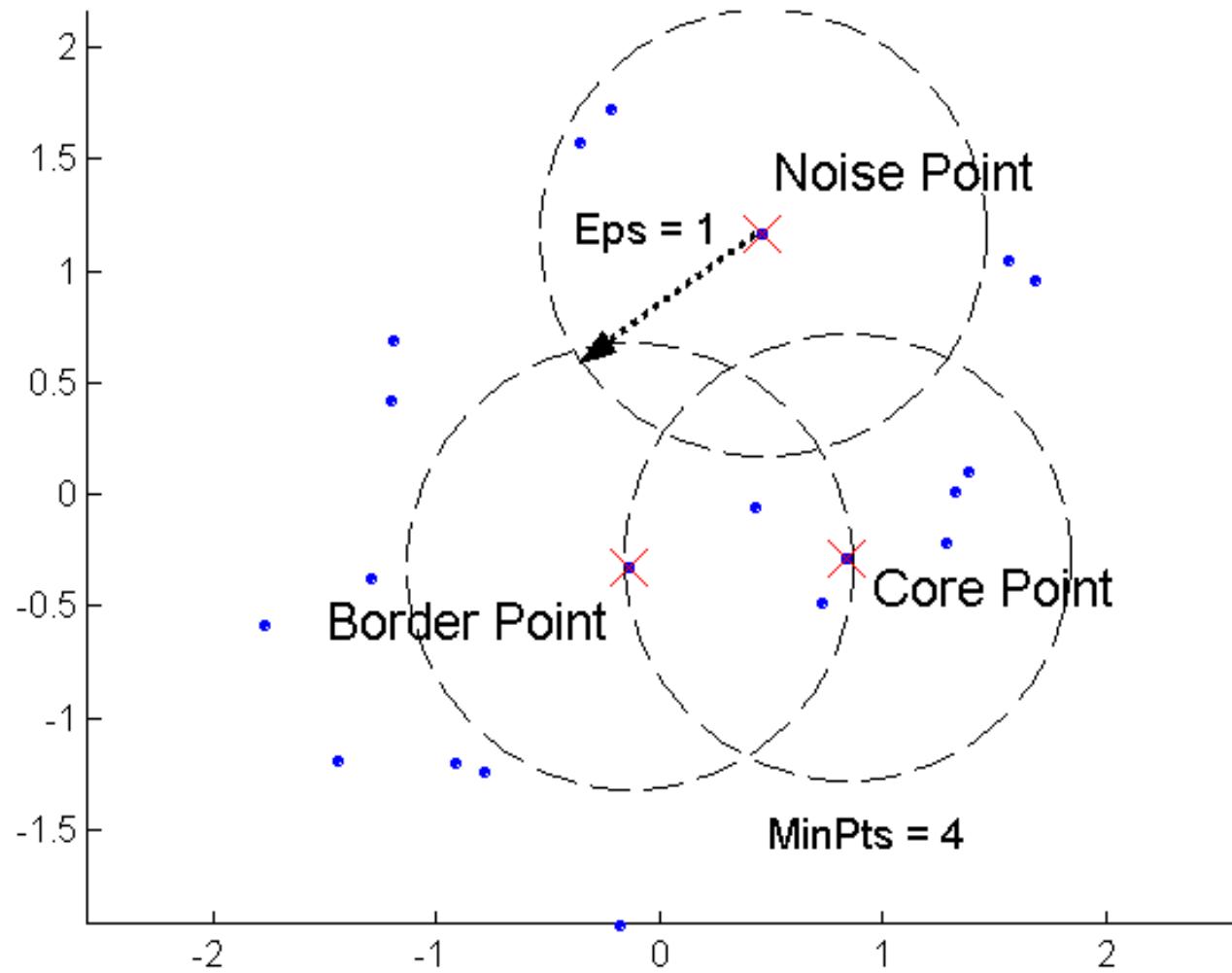
Bisecting K-means (Divisive Hierarchical Clust)



DBSCAN (Density Based Clustering)

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

DBSCAN: Core, Border, and Noise Points



DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

current_cluster_label $\leftarrow 1$

for all core points **do**

if the core point has no cluster label **then**

current_cluster_label $\leftarrow \text{current_cluster_label} + 1$

 Label the current core point with cluster label *current_cluster_label*

end if

for all points in the *Eps*-neighborhood, except *ith* the point itself **do**

if the point does not have a cluster label **then**

 Label the point with cluster label *current_cluster_label*

end if

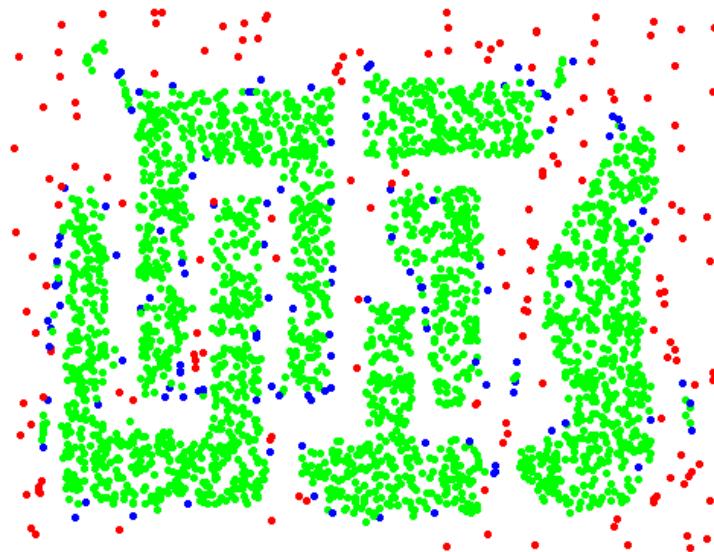
end for

end for

DBSCAN: Core, Border and Noise Points



Original Points



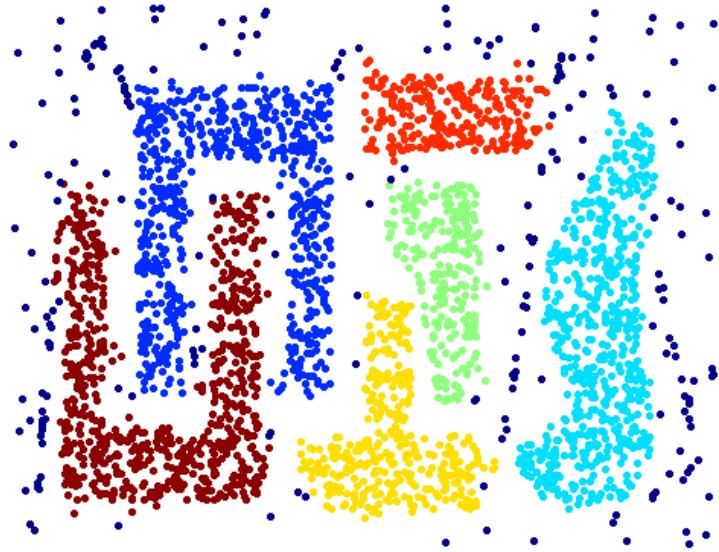
Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

When DBSCAN Works Well



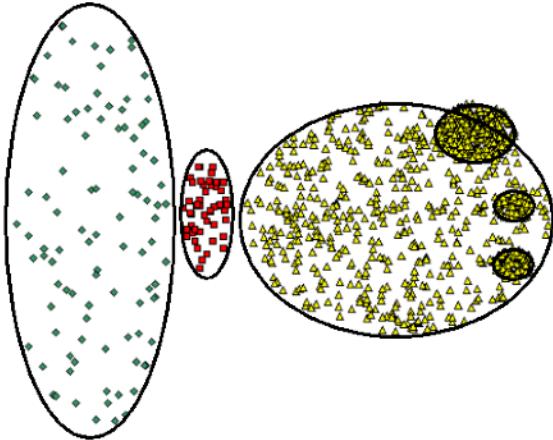
Original Points



Clusters

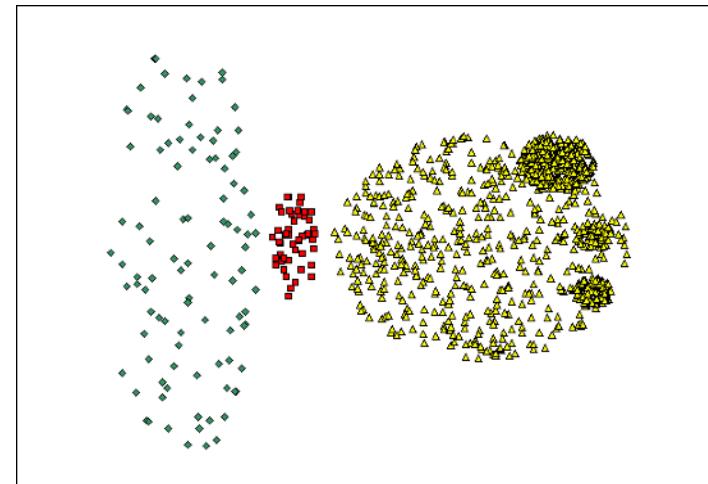
- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

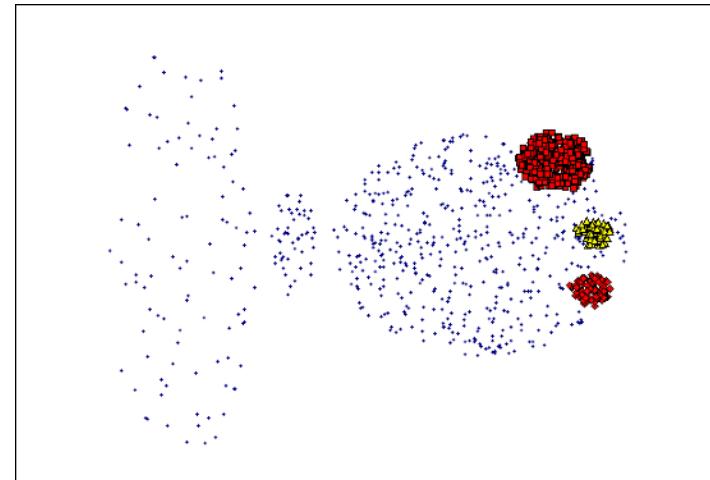


Original Points

- Varying densities
- High-dimensional data



($\text{MinPts}=4$, $\text{Eps}=9.75$).



($\text{MinPts}=4$, $\text{Eps}=9.92$)

Measures of Cluster Validity

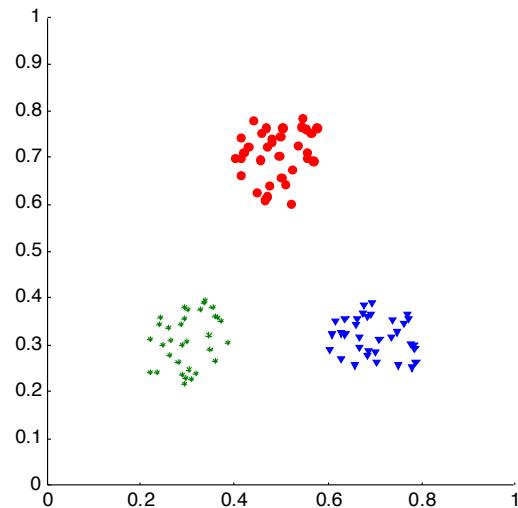
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
 - **External Index:** Used to measure the extent to which cluster labels match externally supplied class labels.
 - ◆ Entropy
 - **Internal Index:** Used to measure the goodness of a clustering structure *without* respect to external information.
 - ◆ Sum of Squared Error (SSE)
 - **Relative Index:** Used to compare two different clusterings or clusters.
 - ◆ Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as **criteria** instead of **indices**
 - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation

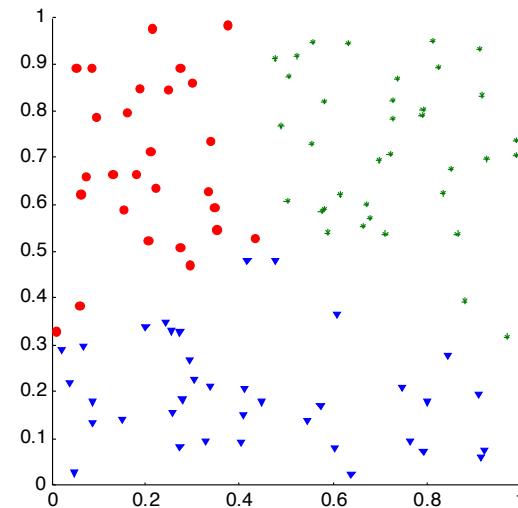
- Two matrices
 - Proximity Matrix (similarity matrix)
 - “Incidence” Matrix
 - ◆ One row and one column for each data point
 - ◆ An entry is 1 if the associated pair of points belong to the same cluster
 - ◆ An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
 - Since the matrices are symmetric, only the correlation between $n(n-1) / 2$ entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

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



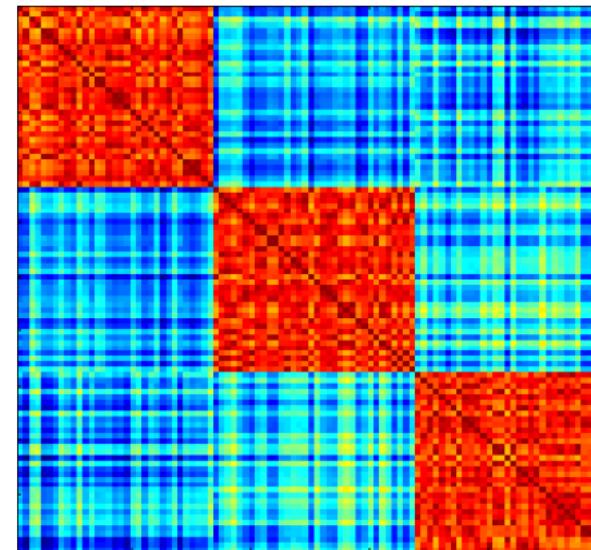
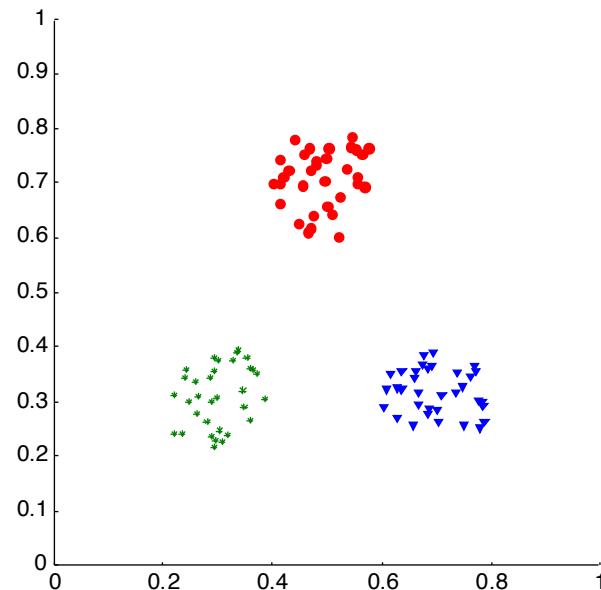
Corr = 0.9235



Corr = 0.5810

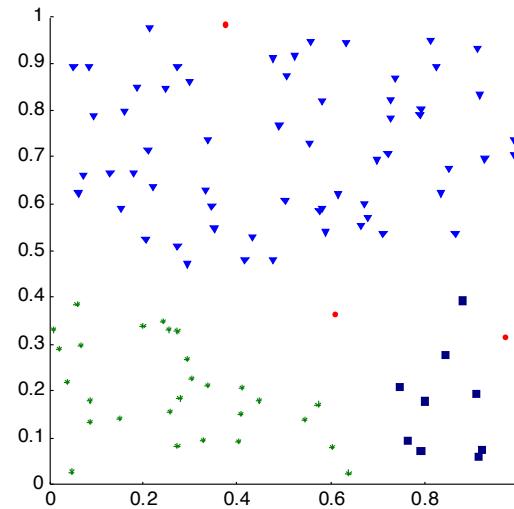
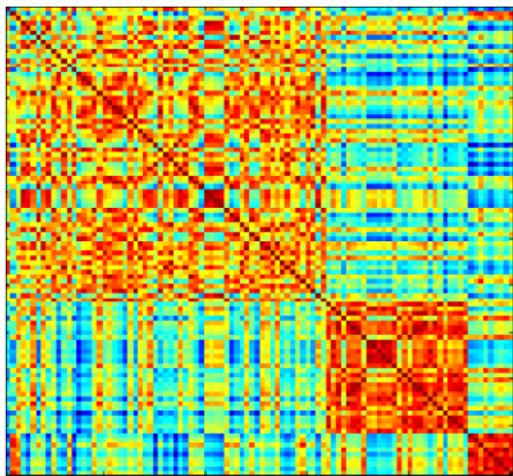
Internal Measures: Similarity Matrix

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



Using Similarity Matrix for Cluster Validation

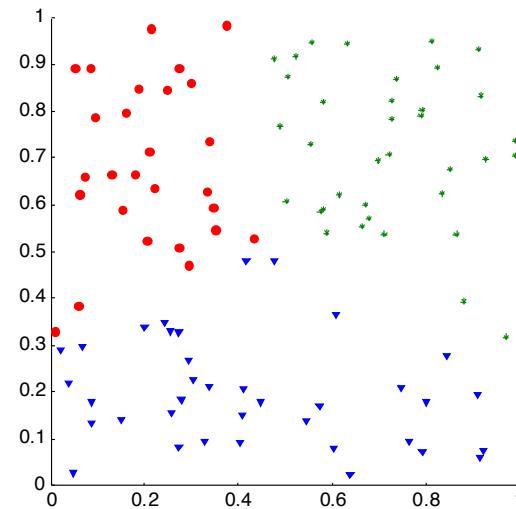
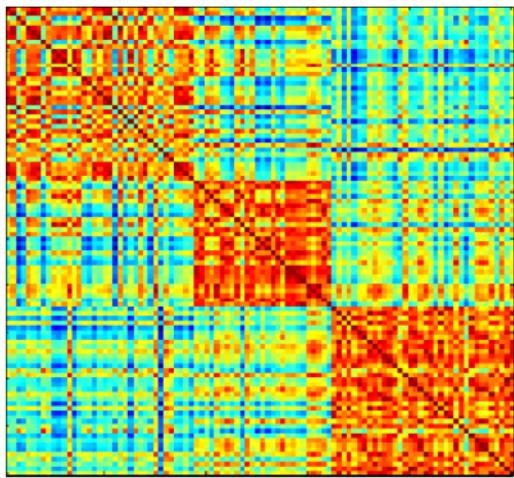
- Clusters in random data are not so crisp



DBSCAN

Using Similarity Matrix for Cluster Validation

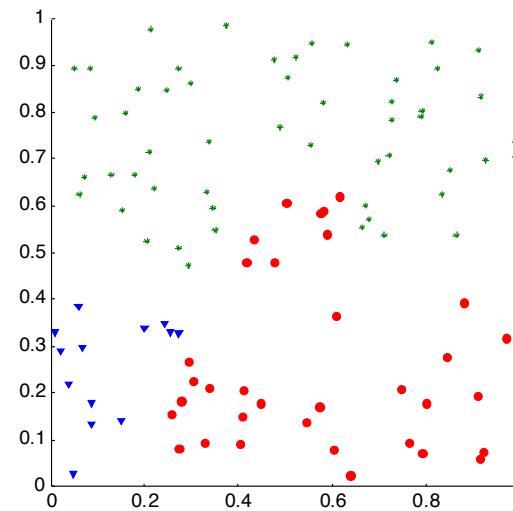
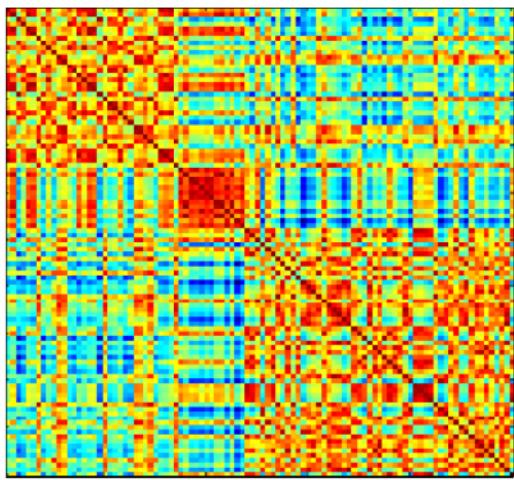
- Clusters in random data are not so crisp



K-means

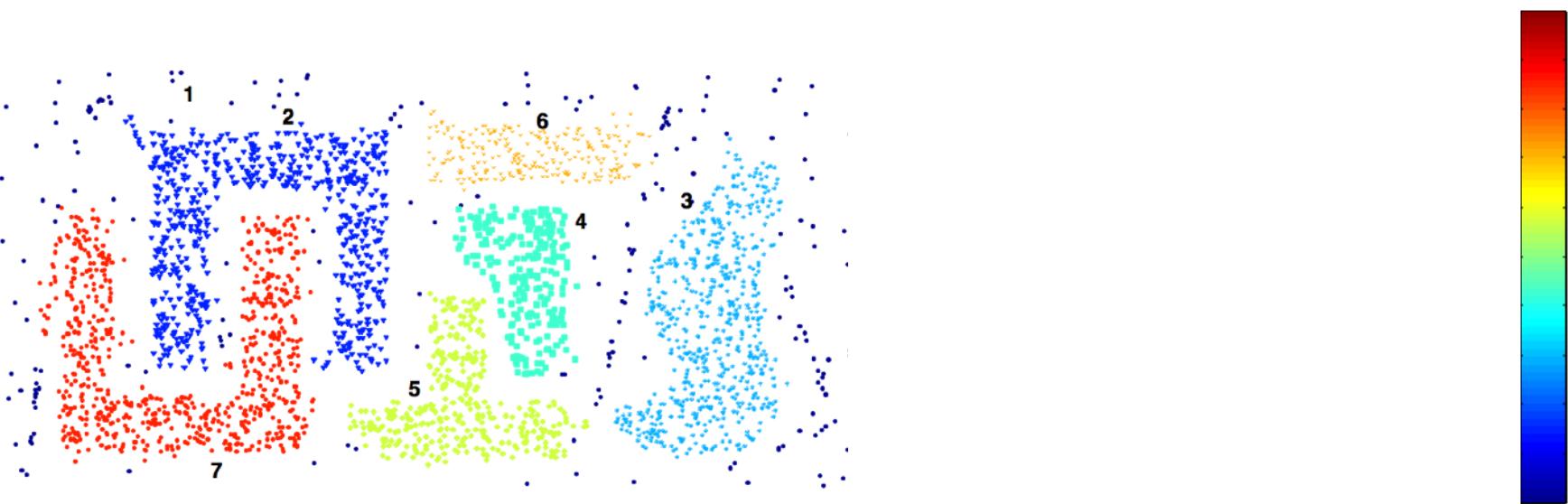
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp



Complete Link

Using Similarity Matrix for Cluster Validation



DBSCAN

Internal 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)

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

- Separation is measured by the between cluster sum of squares

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

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

External Measures: Accuracy

- Evaluate the clustering results w.r.t. an original information about the class
- We need to have the class value for each object, but the class value is not used in the clustering process
- Assign to each cluster a class value (mapping 1:1 between clusters and class values)
- Using the class value of each object compute the accuracy:

$$Accuracy = \frac{\text{number of correctly assigned instances}}{\text{total number of instances}}$$

External Measures: Normalized Mutual Information

- Evaluate the clustering results w.r.t. an original information about the class
- We need to have the class value for each object, but the class value is not used in the clustering process
- Based on Entropy and Information Theory
- Normalized Version of Mutual Information

X = Clustering over the object Y = Original Class Information

$$\text{NMI} = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

External Measures: Normalized Mutual Information

$$I(X;Y) = \sum_y p(y) \sum_x p(x|y) \log_2 \frac{p(x|y)}{p(x)}$$

$$H(X) = - \sum_{i=1}^n p(x_i) \log_b p(x_i)$$