

---

# CS 584

# Data Mining

Clustering 2

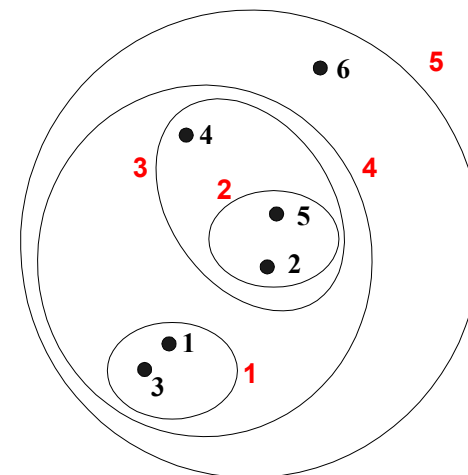
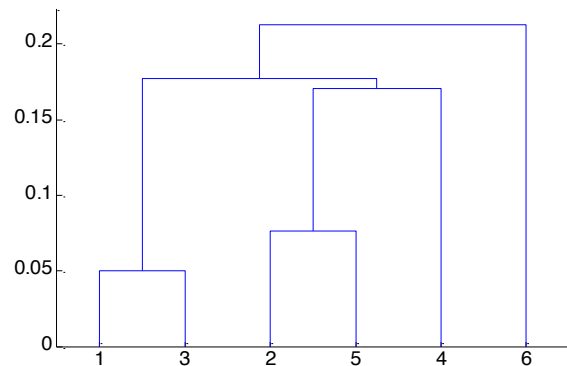
---

# Roadmap for Today

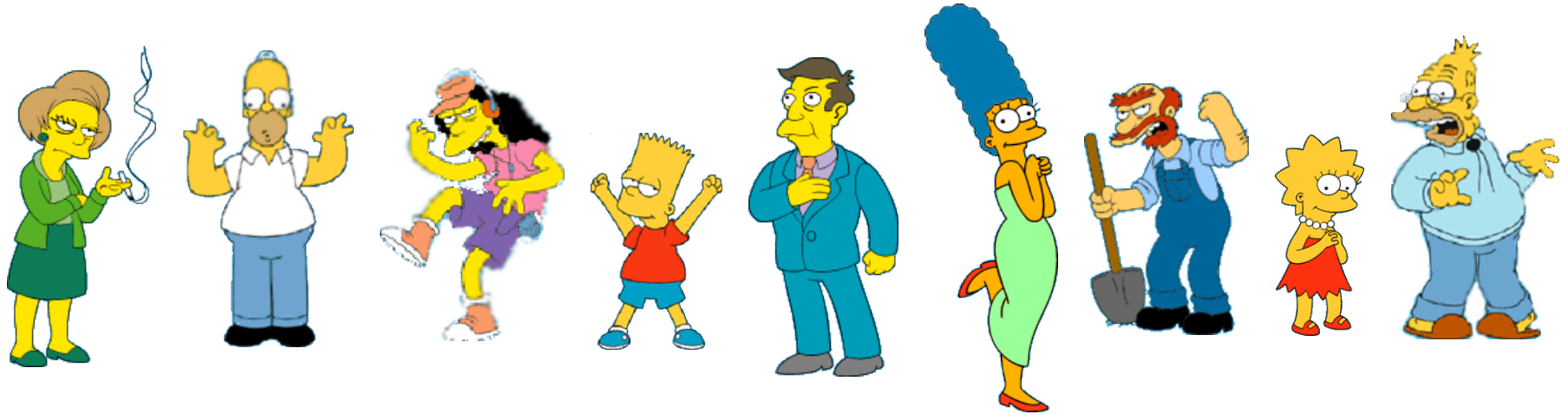
- Last time we talked about Partitional clustering (k-means and its variants)
- Today we will talk about two more types of clustering algorithms
  - Hierarchical clustering
  - Density-based clustering (DBScan)
- Cluster validity

# 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

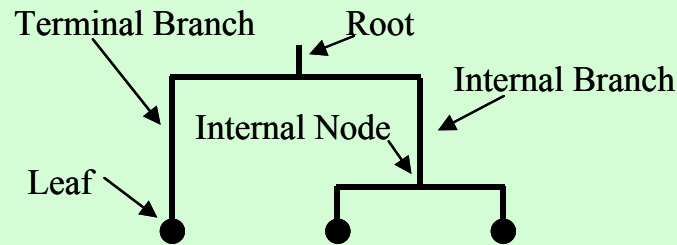


What is a natural grouping among these objects?

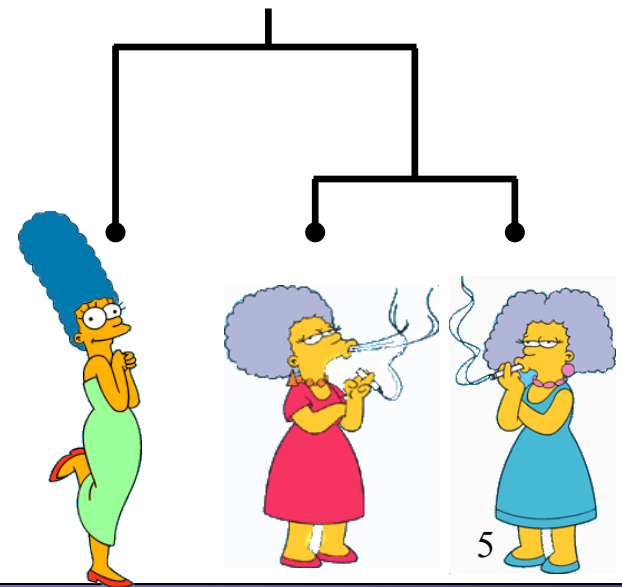
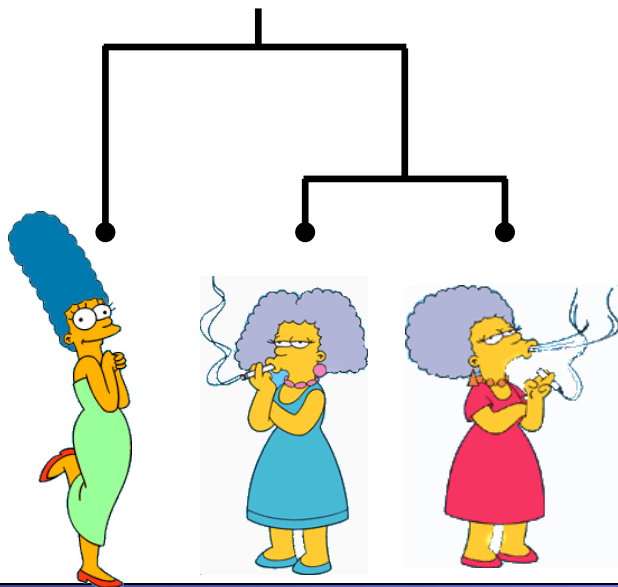


# A Useful Tool for Summarizing Similarity Measurements

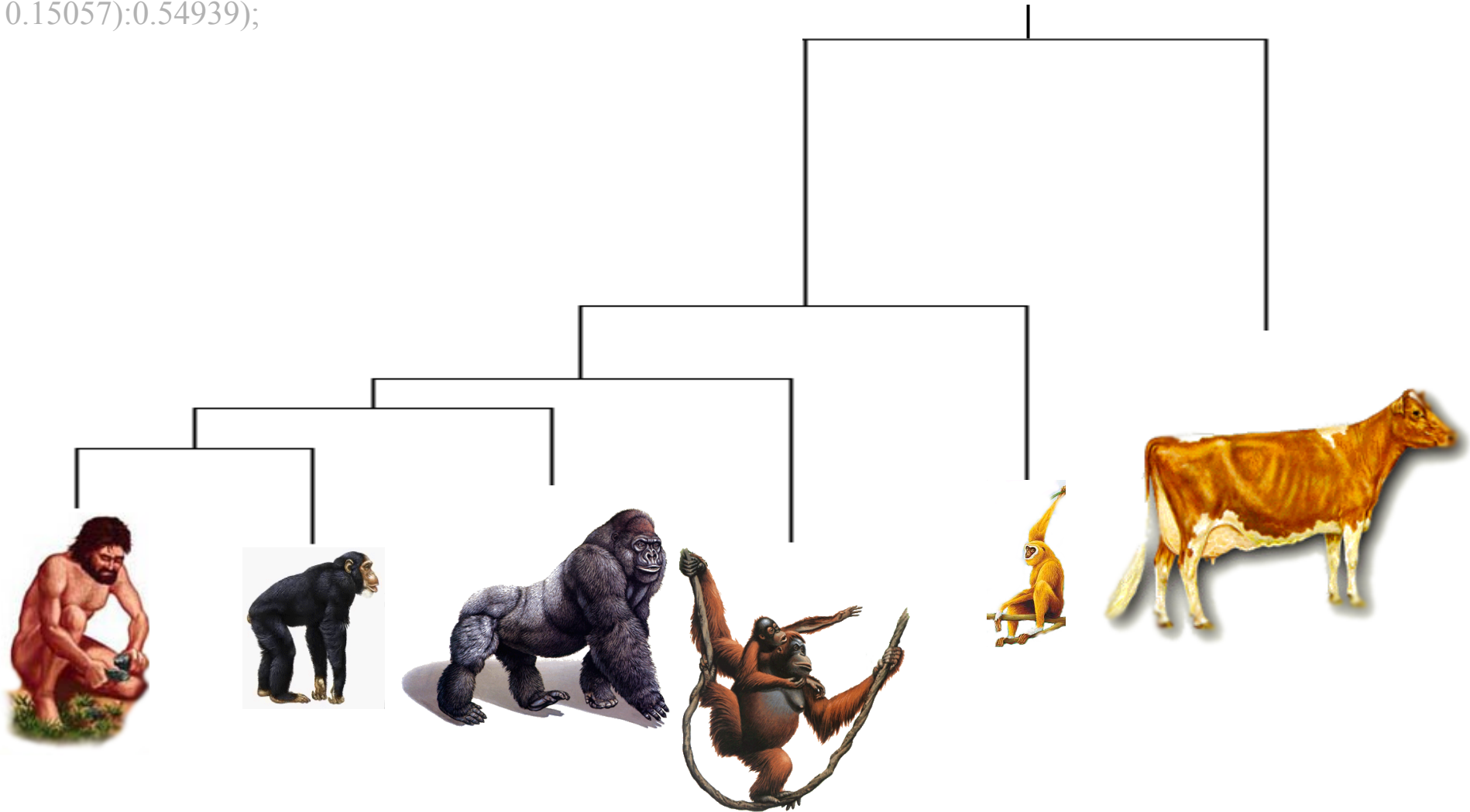
Dendrogram:



The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.



(Bovine:0.69395,(Gibbon:0.36079,(Orangutan:  
0.33636,(Gorilla:0.17147,(Chimp:  
0.19268,Human:0.11927):0.08386):0.06124):  
0.15057):0.54939);



Note that hierarchies are commonly used to organize information, for example in a web portal.

Yahoo's hierarchy is manually created, we will focus on automatic creation of hierarchies in data mining.

Web Site Directory - Sites organized by subject

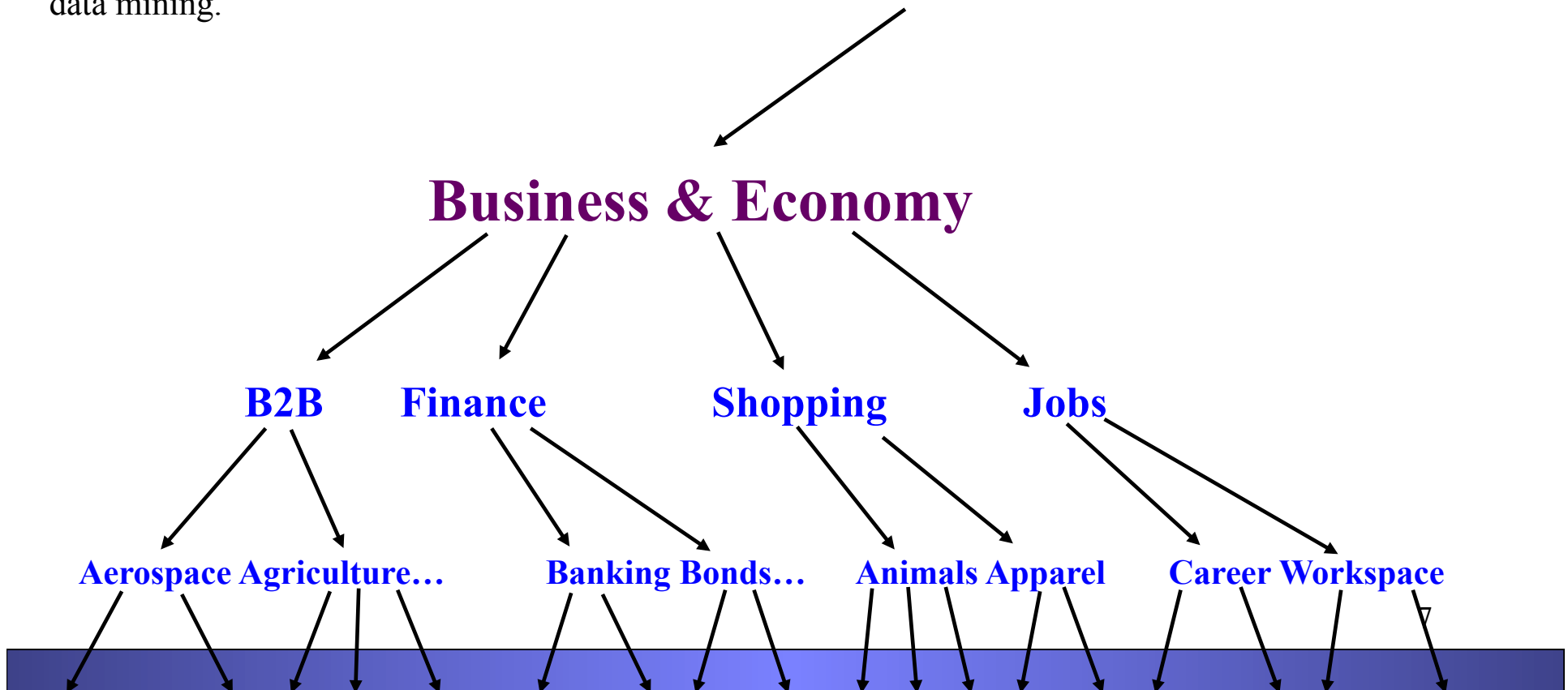
[Suggest your site](#)

**Business & Economy**  
[B2B](#), [Finance](#), [Shopping](#), [Jobs](#)...

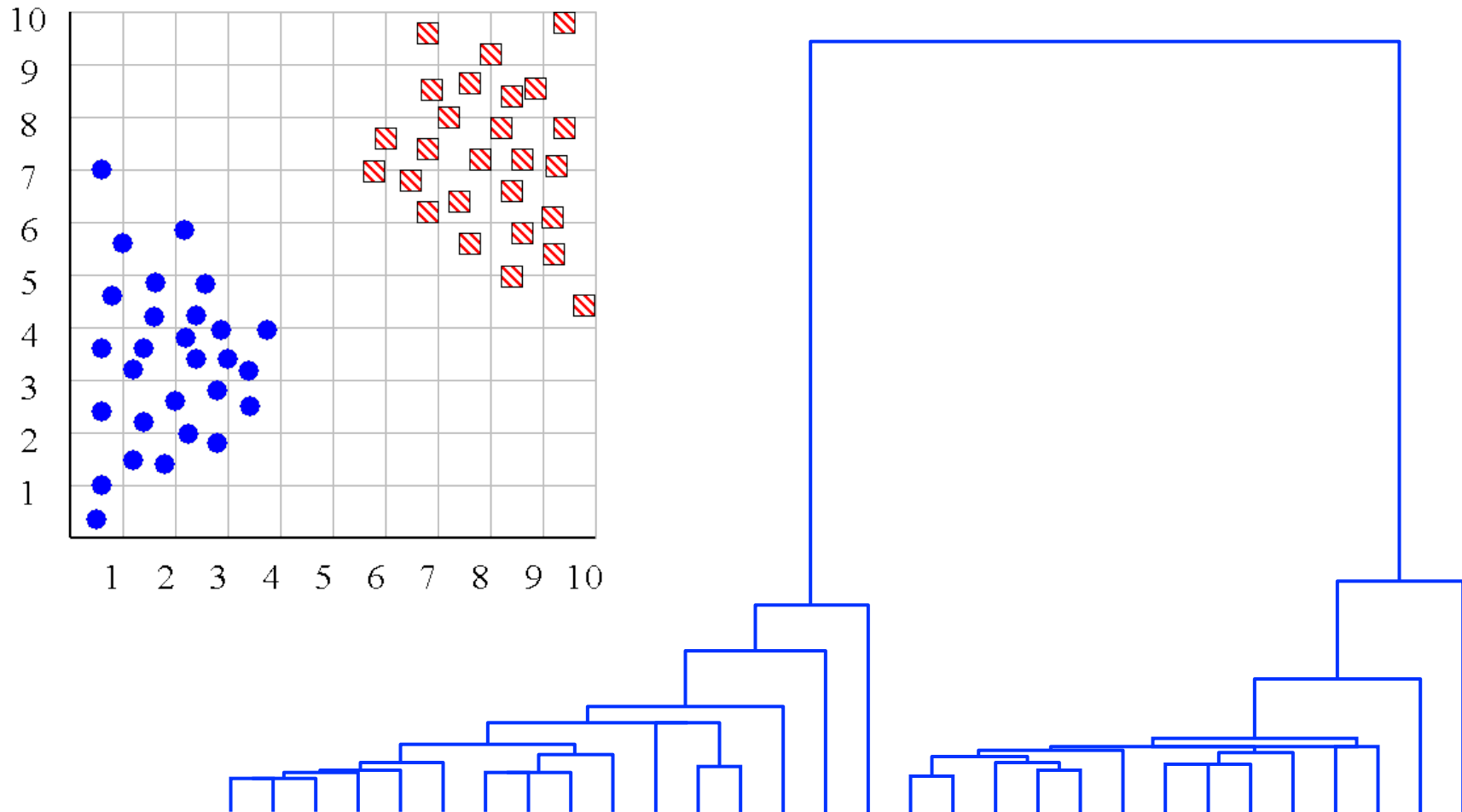
**Regional**  
[Countries](#), [Regions](#), [US States](#)...

**Computers & Internet**  
[Internet](#), [WWW](#), [Software](#), [Games](#)...

**Society & Culture**  
[People](#), [Environment](#), [Religion](#)...



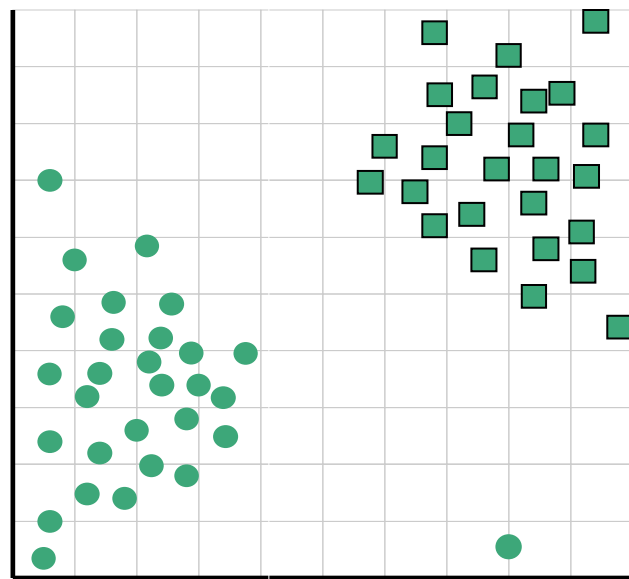
We can look at the dendrogram to determine the “correct” number of clusters. In this case, the two highly separated subtrees are highly suggestive of two clusters. (Things are rarely this clear cut, unfortunately)



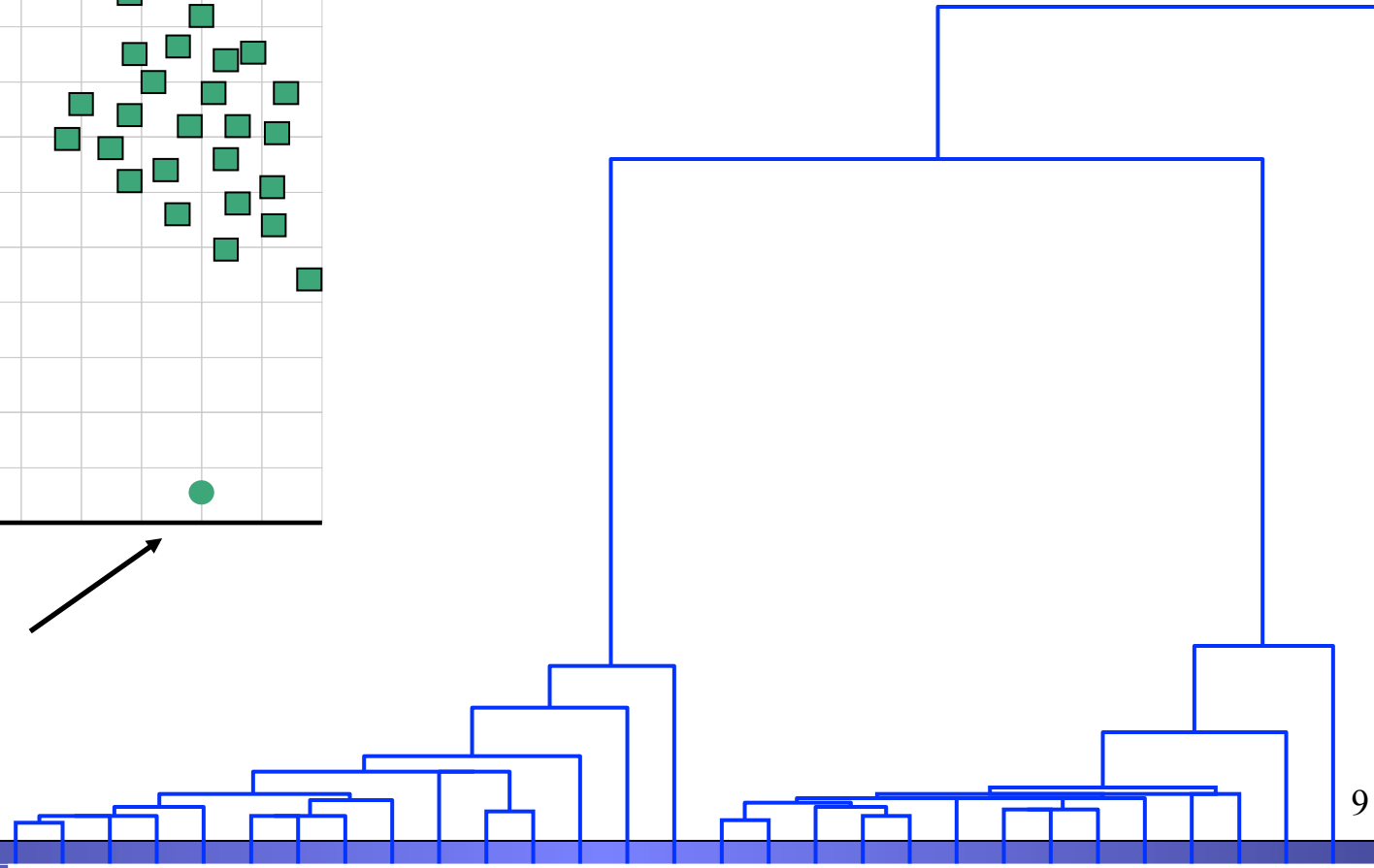
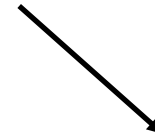
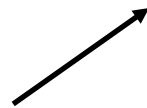


# One potential use of a dendrogram is to detect outliers

The single isolated branch is suggestive of a data point that is very different to all others



Outlier



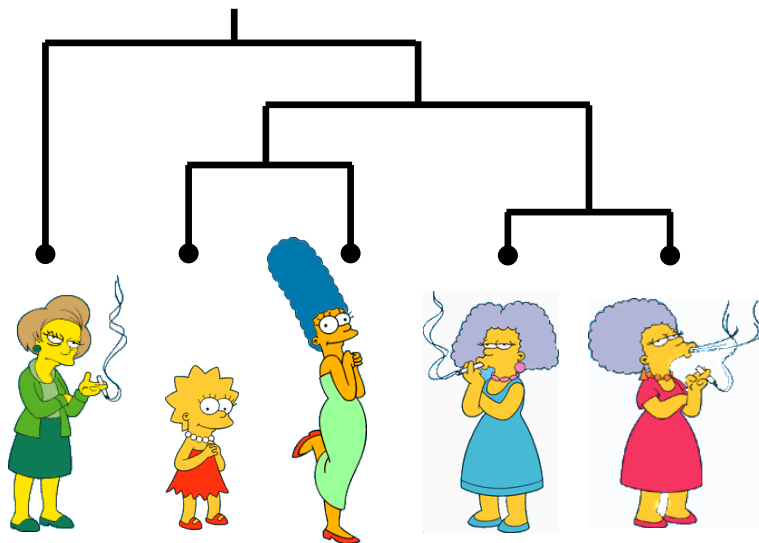
# Strengths of Hierarchical Clustering

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

# Hierarchical Clustering

The number of dendrograms with  $n$  leaves =  $(2n - 3)! / [(2^{(n-2)}) (n - 2)!]$

Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	...
10	34,459,425



Since we cannot test all possible trees we will have to heuristic search of all possible trees. We could do this..

**Bottom-Up (agglomerative):** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

**Top-Down (divisive):** Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides.











# Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  - Compute the proximity matrix
  - Let each data point be a cluster
  - Repeat
    - Merge the two closest clusters
    - Update the proximity matrix
  - 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

We begin with a distance matrix which contains the distances between every pair of objects in our database.

$$D(\text{Mrs. Krabappel}, \text{Lisa Simpson}) = 8$$

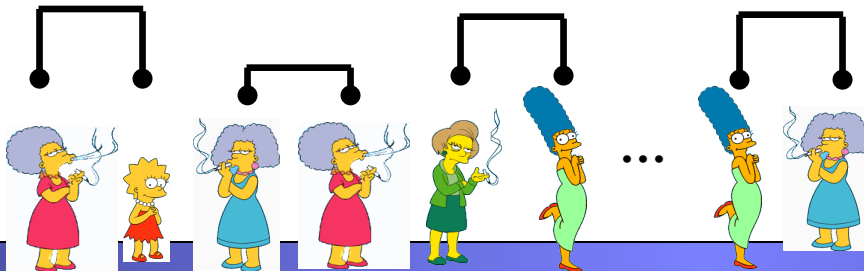
$$D(\text{Marge Simpson}, \text{Bart Simpson}) = 1$$

				
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0
				13

## Bottom-Up (agglomerative):

Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

Consider all possible merges...



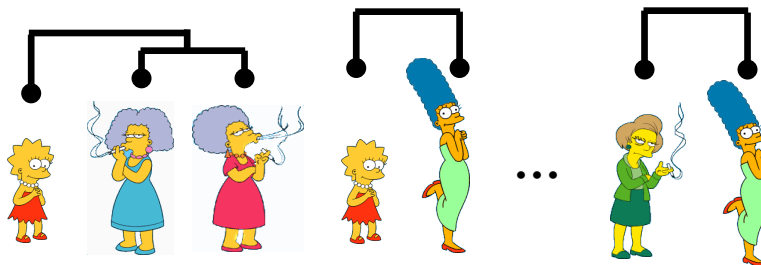
Choose the best



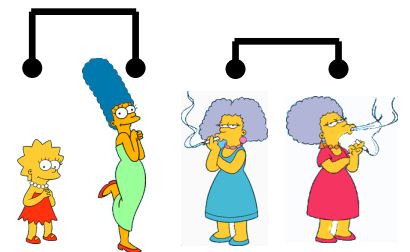
## Bottom-Up (agglomerative):

Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

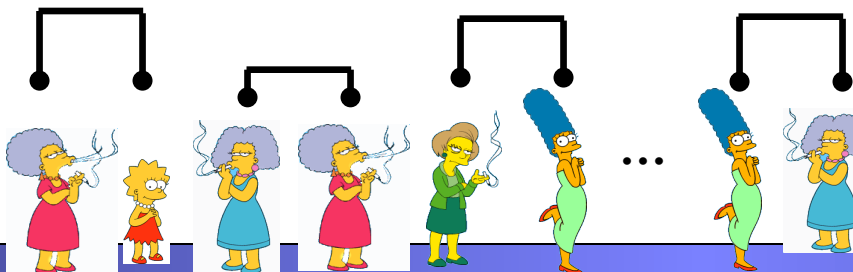
Consider all possible merges...



Choose the best



Consider all possible merges...



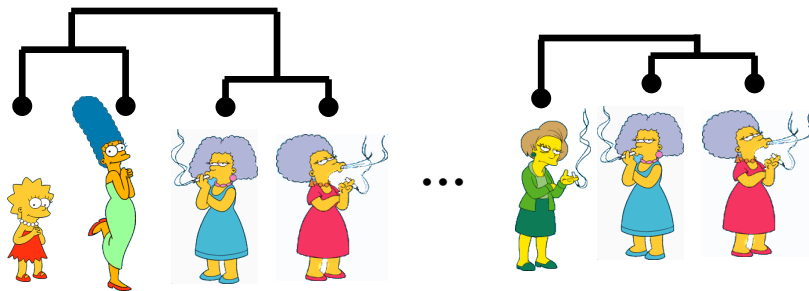
Choose the best



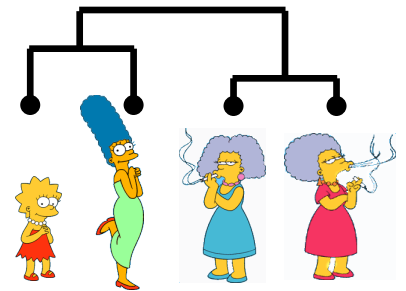
## Bottom-Up (agglomerative):

Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

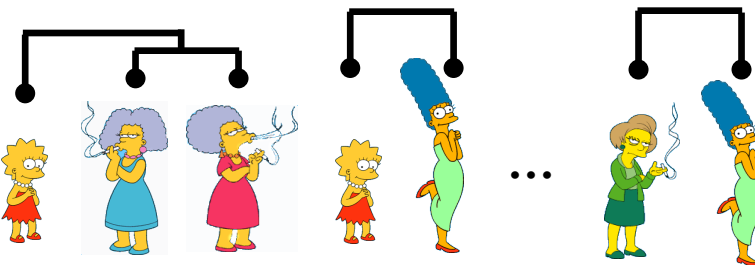
Consider all possible merges...



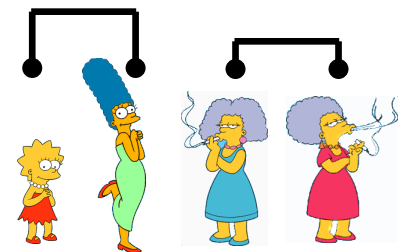
Choose the best



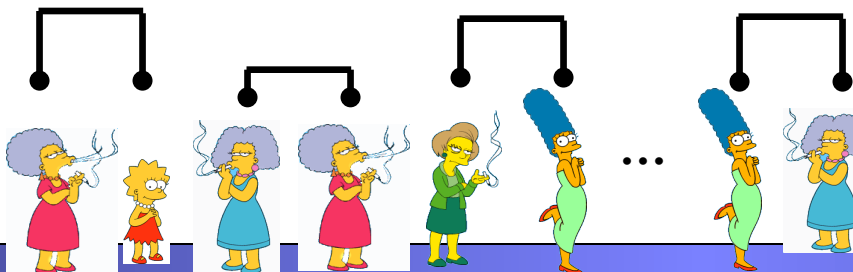
Consider all possible merges...



Choose the best



Consider all possible merges...



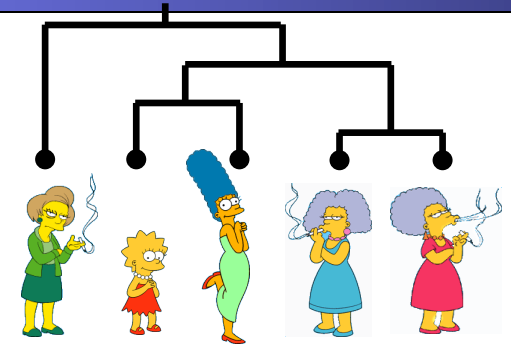
Choose the best



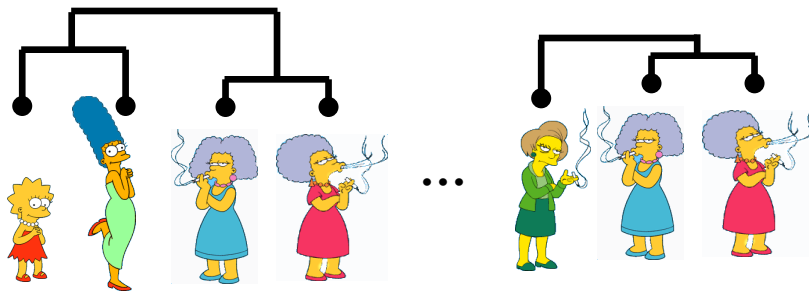


# Bottom-Up (agglomerative):

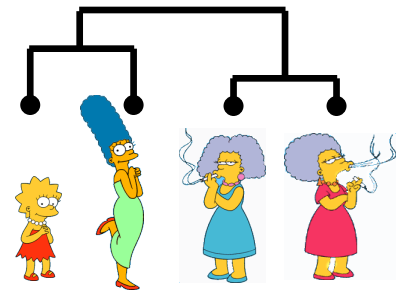
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



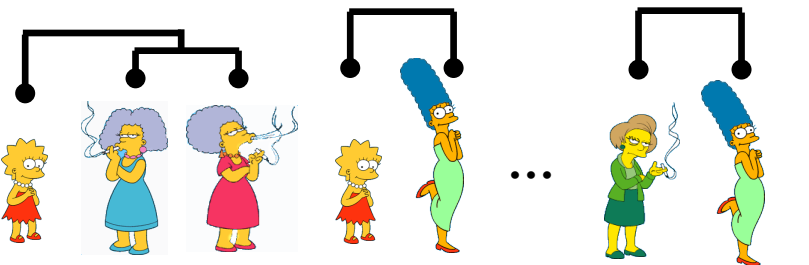
Consider all possible merges...



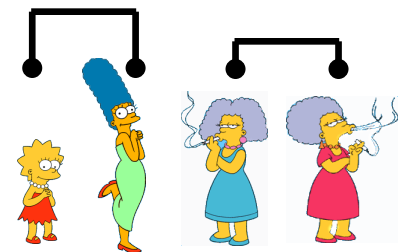
Choose the best



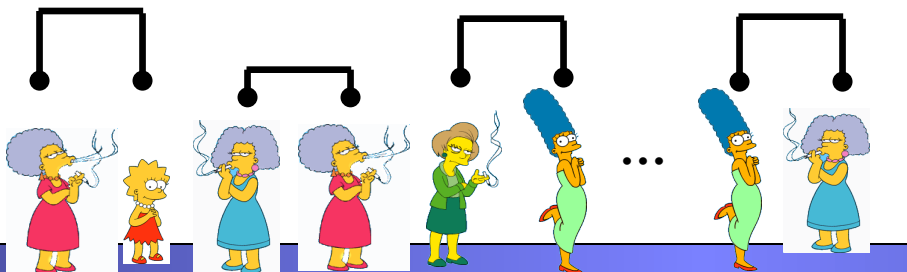
Consider all possible merges...



Choose the best



Consider all possible merges...

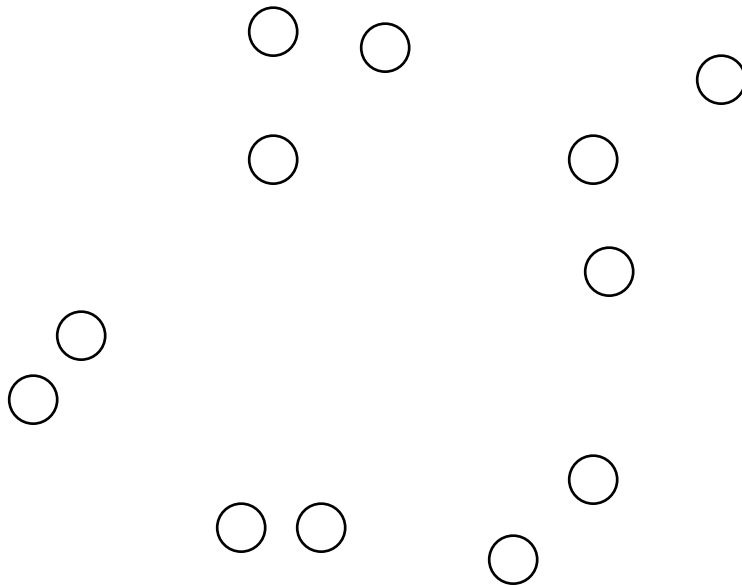


Choose the best



# Starting Situation

- Start with clusters of individual points and a proximity matrix



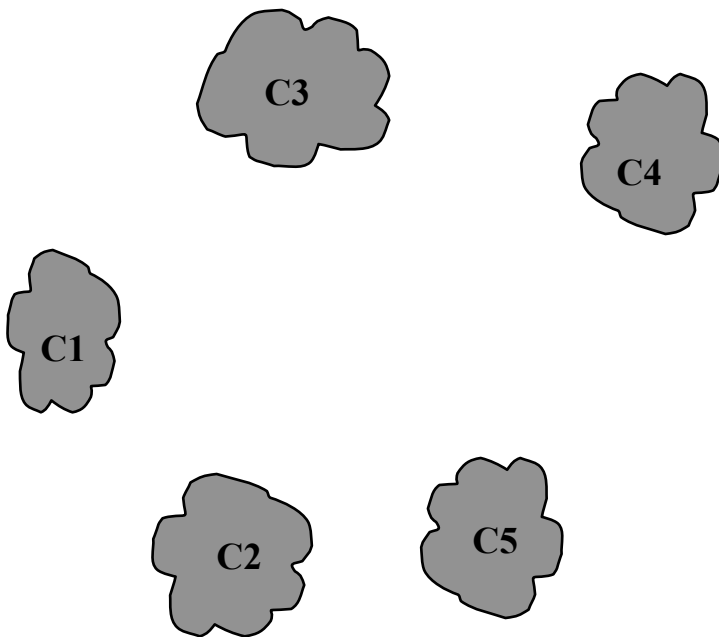
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

**Proximity Matrix**



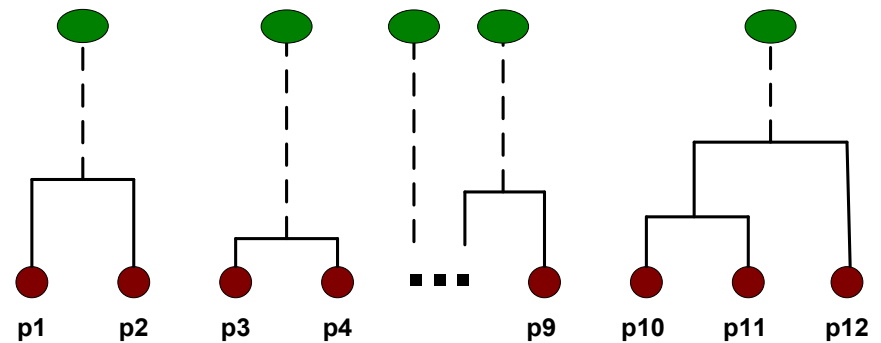
# Intermediate Situation

- After some merging steps, we have some clusters



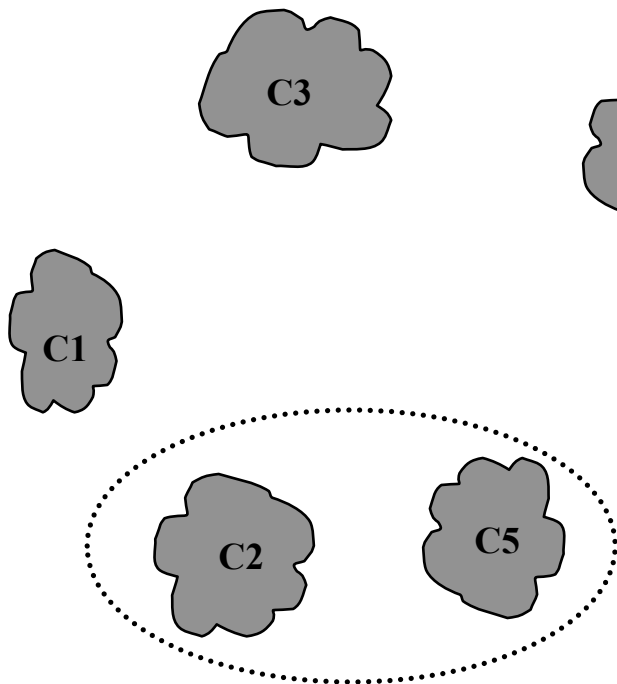
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

**Proximity Matrix**



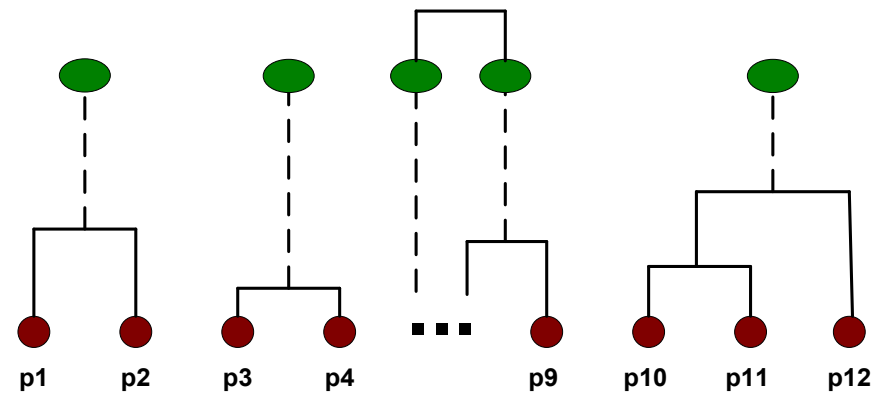
# Intermediate Situation

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



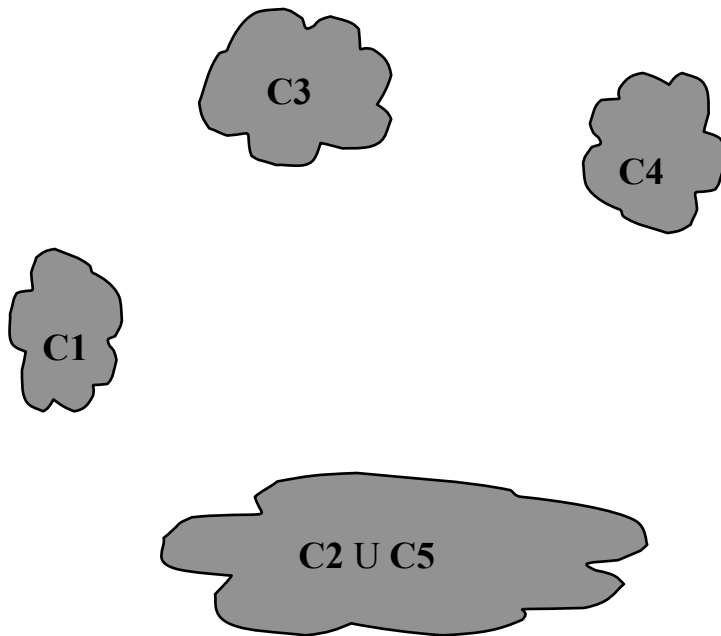
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

**Proximity Matrix**



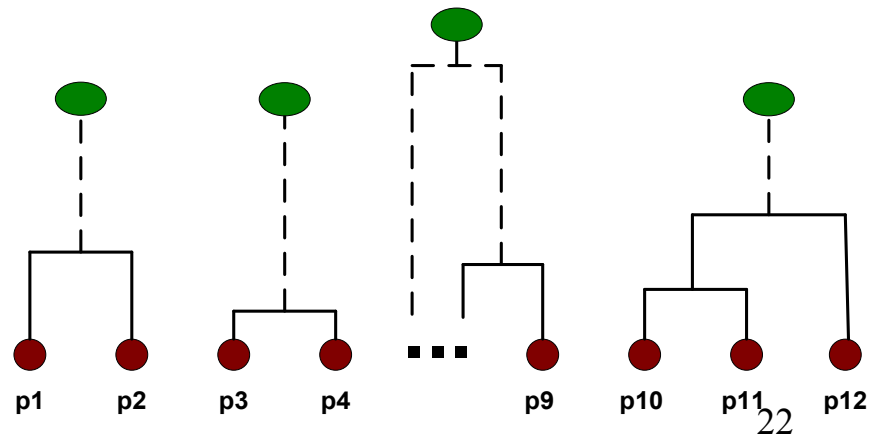
# After Merging

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



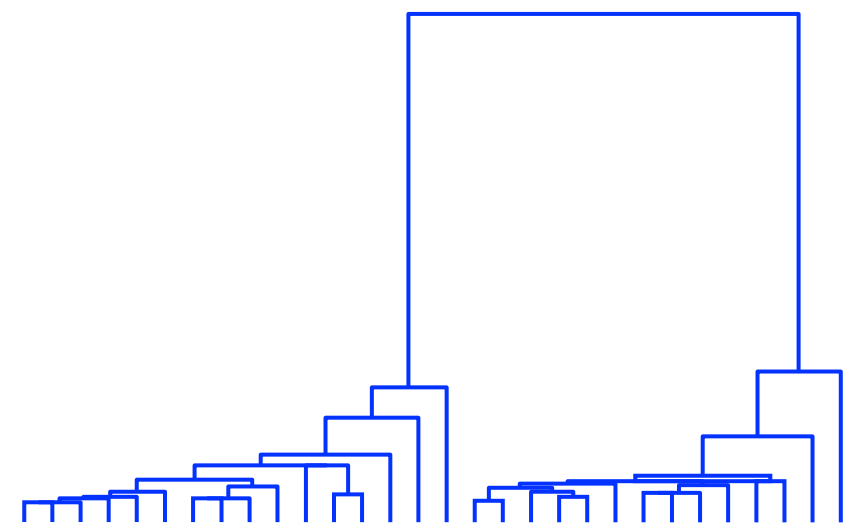
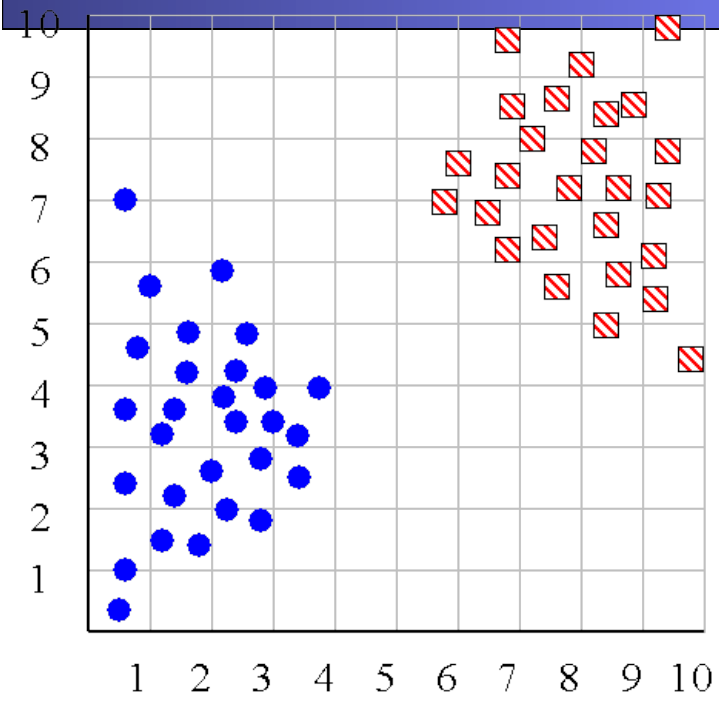
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix

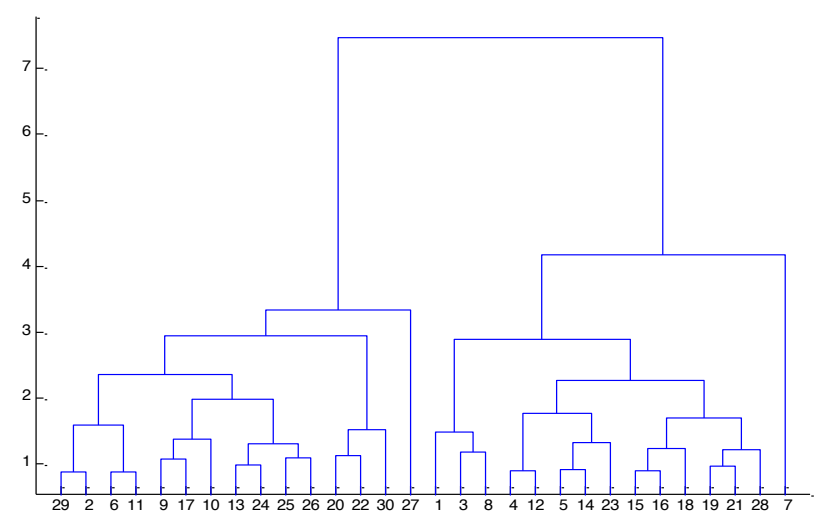


We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.

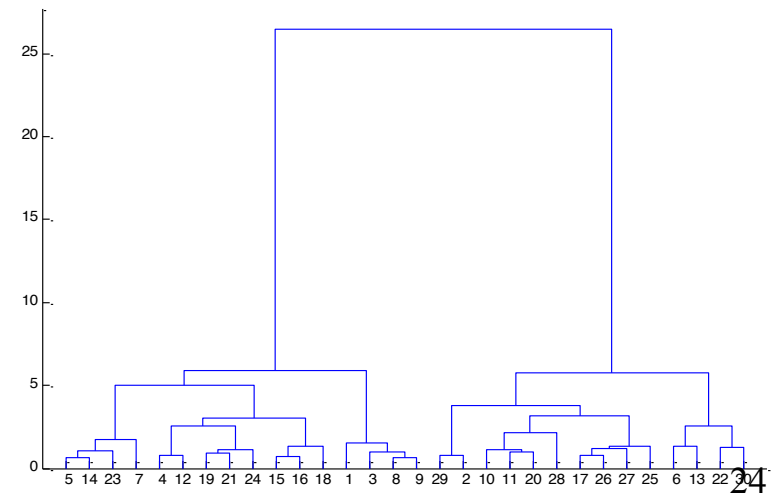
- **MIN or Single linkage (nearest neighbor):** In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.
- **MAX or Complete linkage (furthest neighbor):** In this method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors").
- **Group average linkage:** In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters.
- **Distance between centroids:** In this method, the distance between two clusters is determined by the distance between their respective centroids.
- **Wards Linkage:** In this method, we try to minimize the variance of the merged clusters



Single linkage

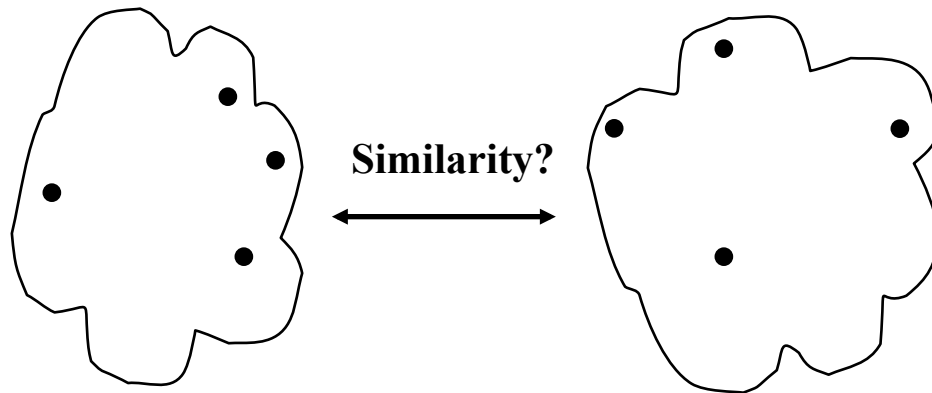


Average linkage



Wards linkage

# How to Define Inter-Cluster Similarity



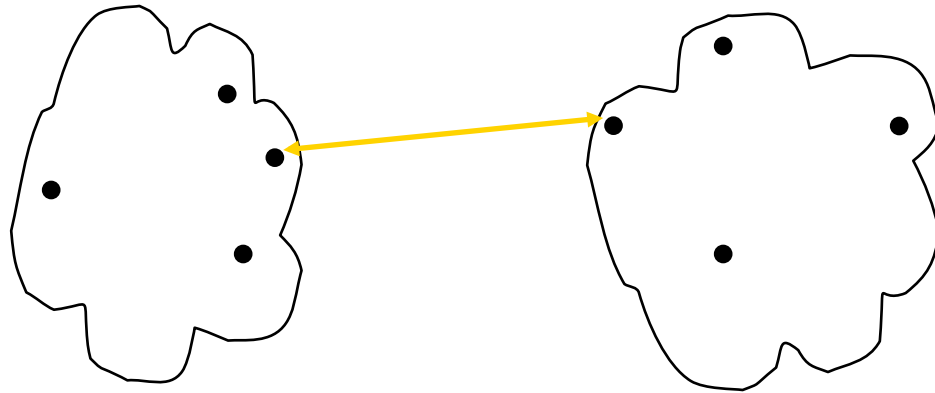
- MIN (single linkage)
- MAX (complete linkage)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

• **Proximity Matrix**



## How to Define Inter-Cluster Similarity

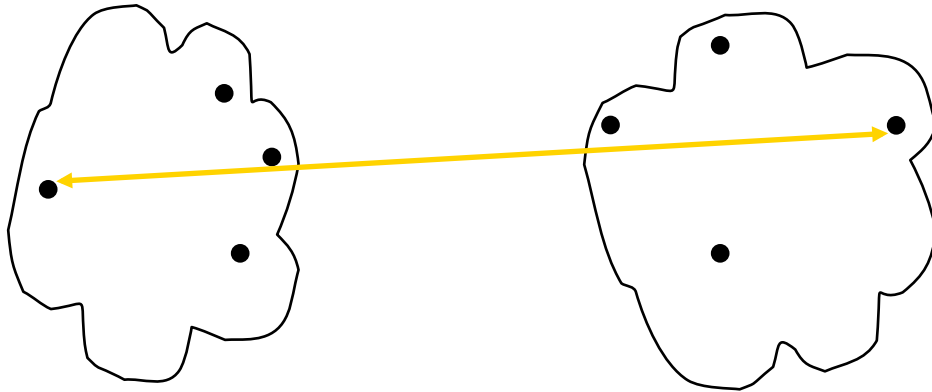


- **MIN (single linkage)**
- **MAX (complete linkage)**
- **Group Average**
- **Distance Between Centroids**
- **Other methods driven by an objective function**
  - Ward's Method uses squared error

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

**Proximity Matrix**

# How to Define Inter-Cluster Similarity

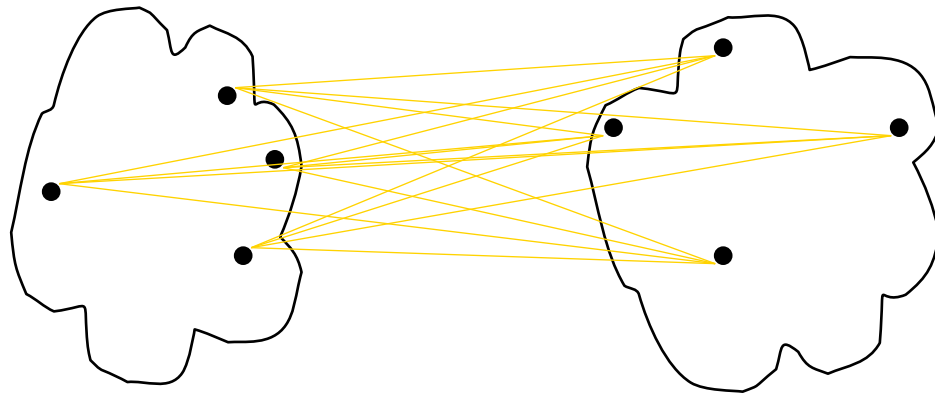


- MIN (single linkage)
- **MAX (complete linkage)**
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

# How to Define Inter-Cluster Similarity

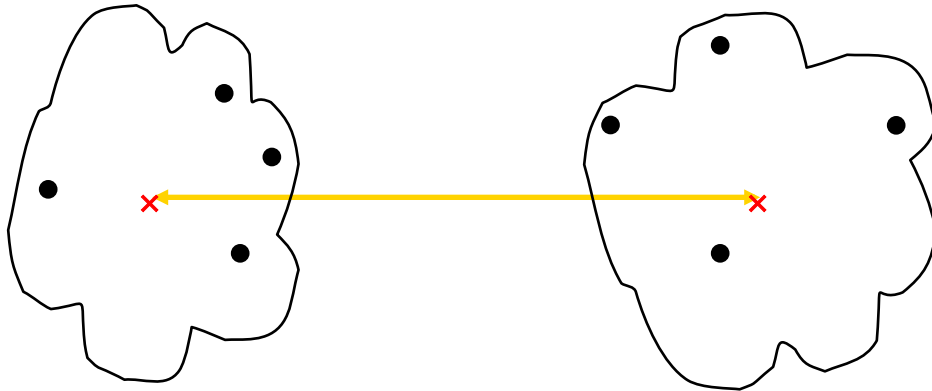


- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

	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

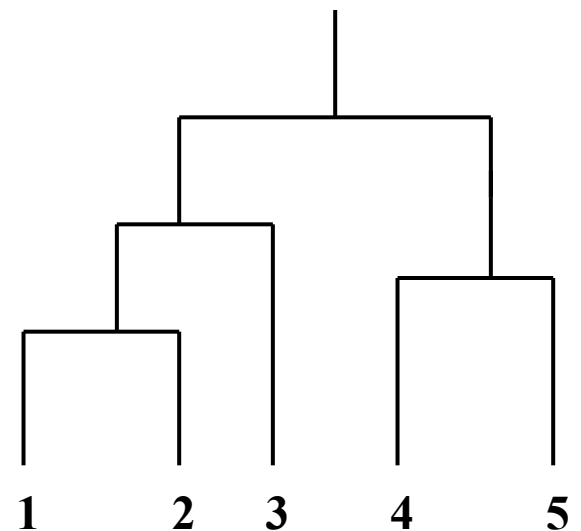
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

• **Proximity Matrix**

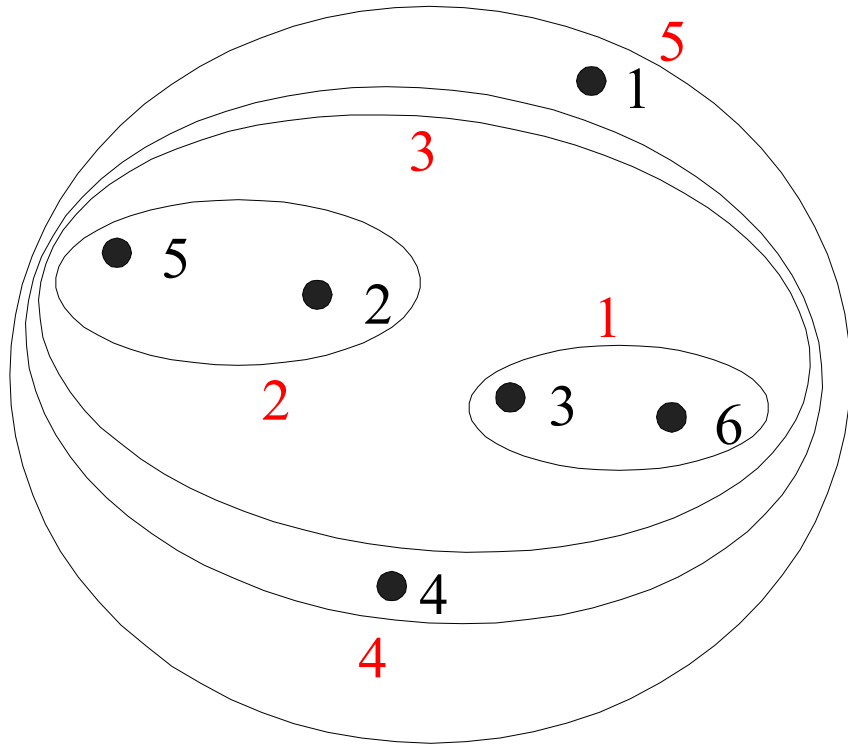
# Cluster Similarity: MIN or Single Link

- 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.

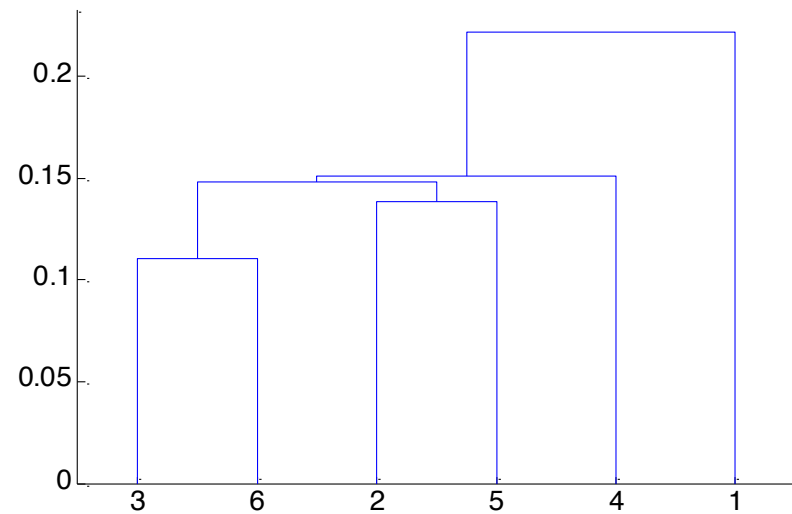
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



# Hierarchical Clustering: MIN

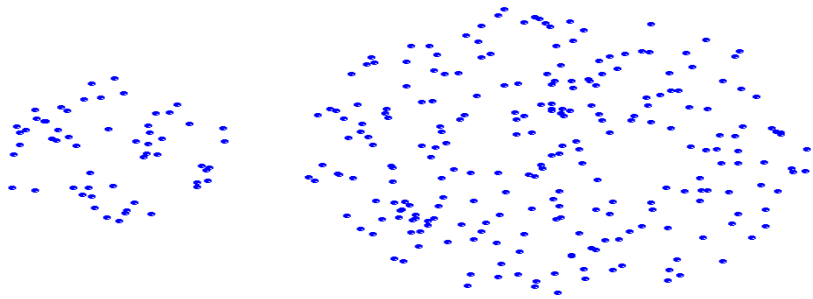


**Nested Clusters**

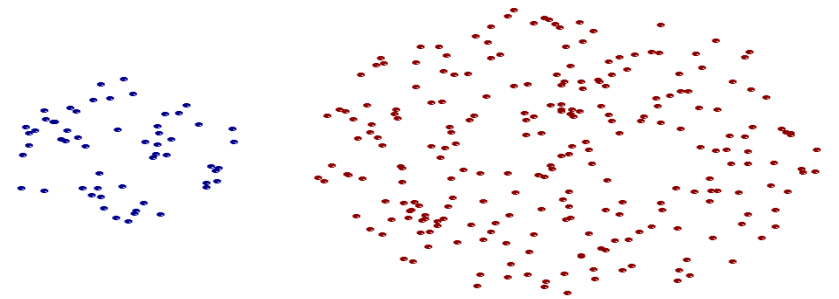


**Dendrogram**

# Strength of MIN



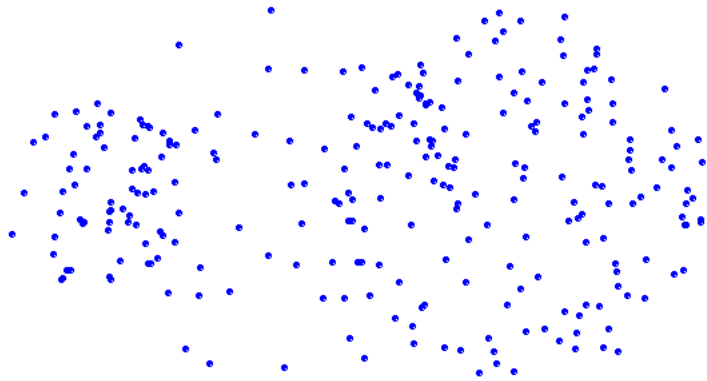
**Original Points**



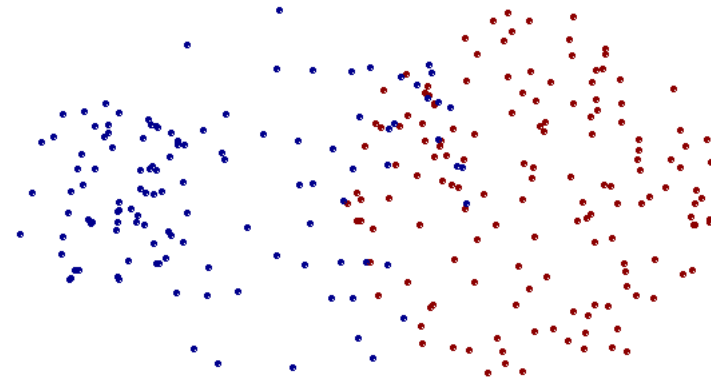
**Two Clusters**

- **Can handle non-elliptical shapes**

# Limitations of MIN



**Original Points**



**Two Clusters**

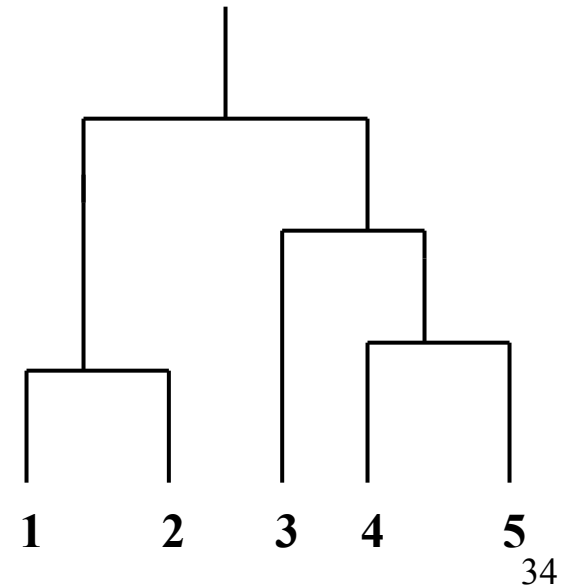
- **Sensitive to noise and outliers**



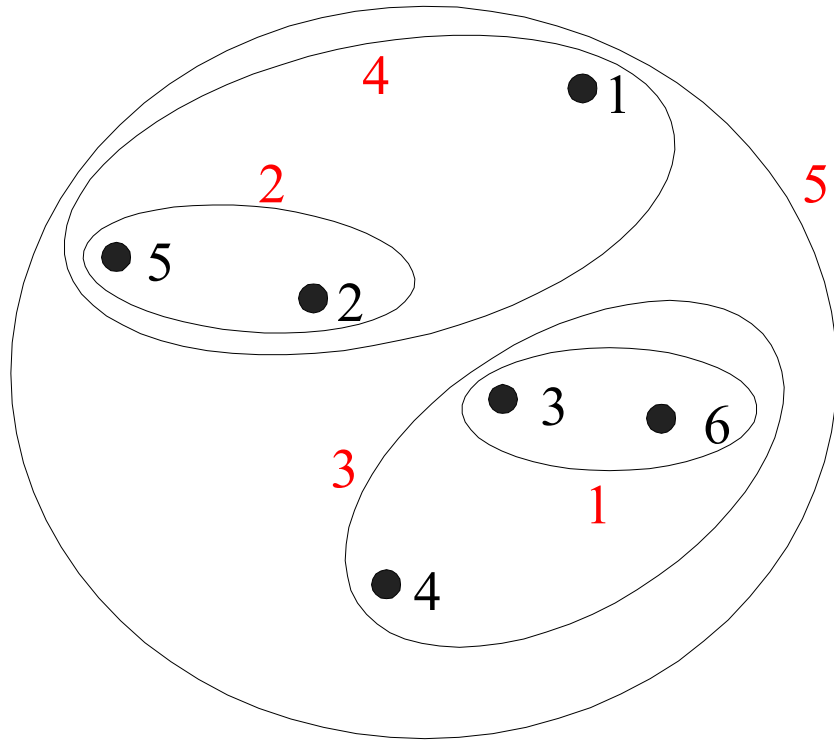
# Cluster Similarity: MAX or Complete Linkage

- 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

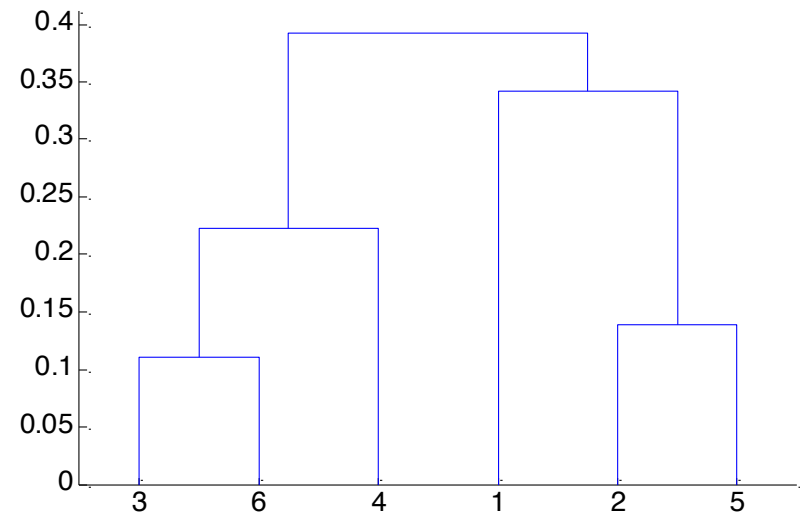
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



# Hierarchical Clustering: MAX

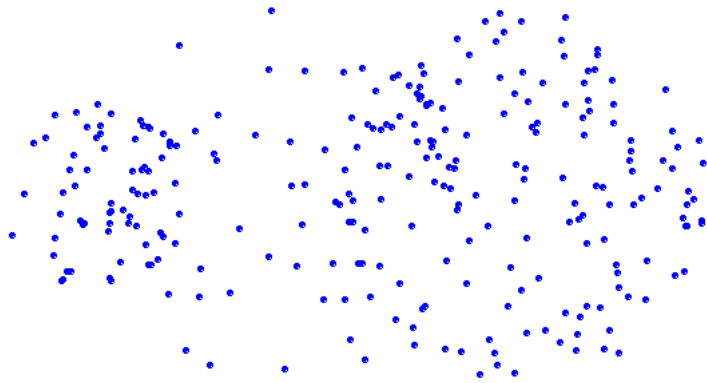


**Nested Clusters**

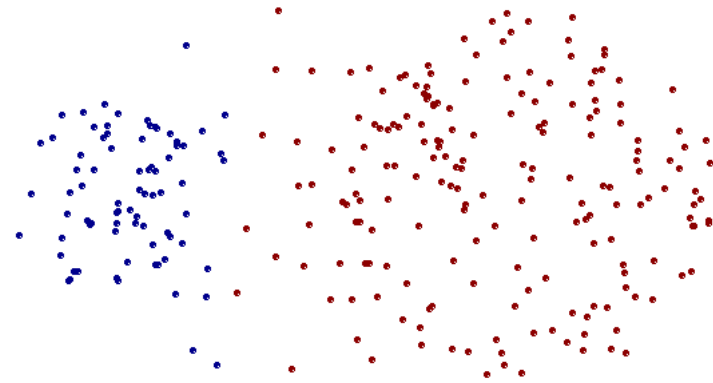


**Dendrogram**

# Strength of MAX



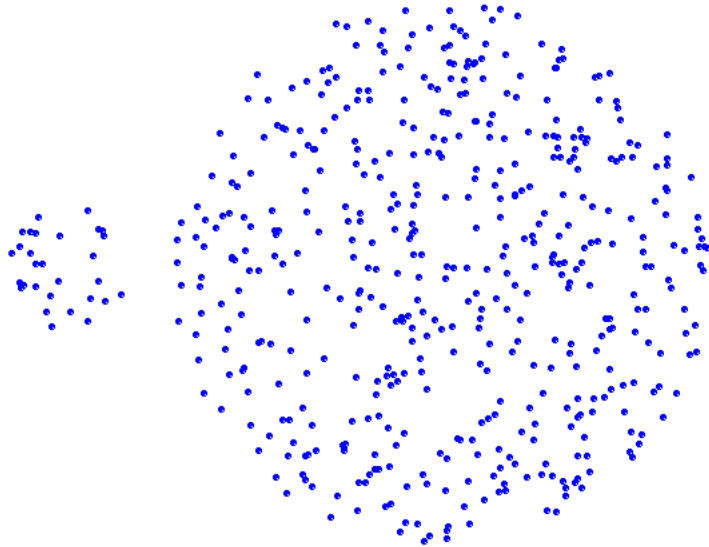
**Original Points**



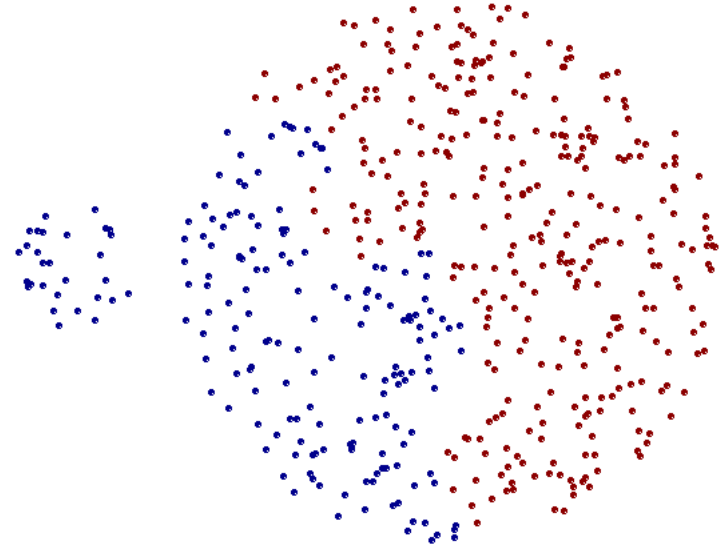
**Two Clusters**

- **Less susceptible to noise and outliers**

# Limitations of MAX



**Original Points**



**Two Clusters**

- **Tends to break large clusters**
- **Biased towards globular clusters**

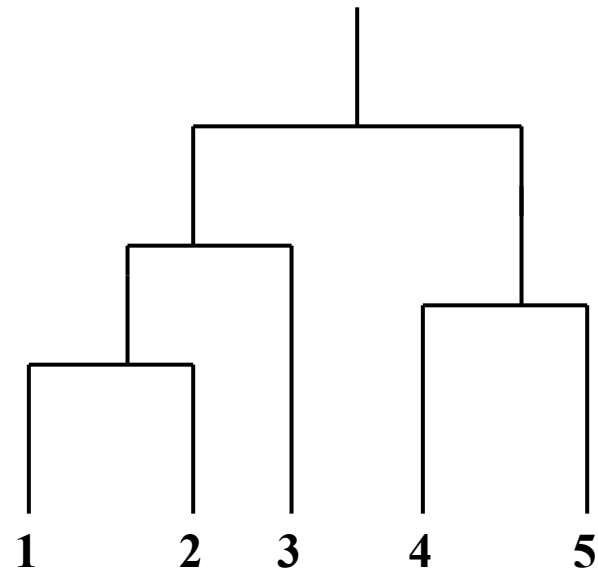
# Cluster Similarity: Group Average

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

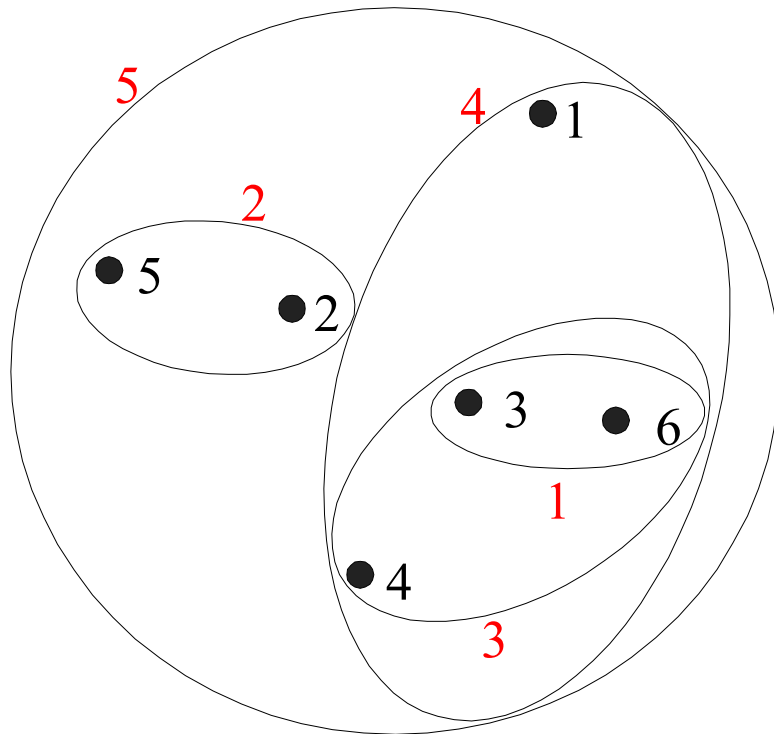
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

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

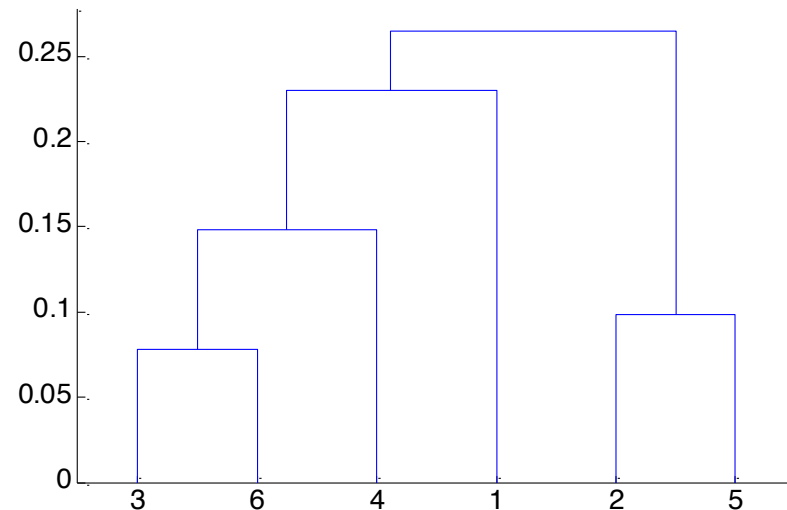
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



# Hierarchical Clustering: Group Average



**Nested Clusters**



**Dendrogram**

# Hierarchical Clustering: Group Average

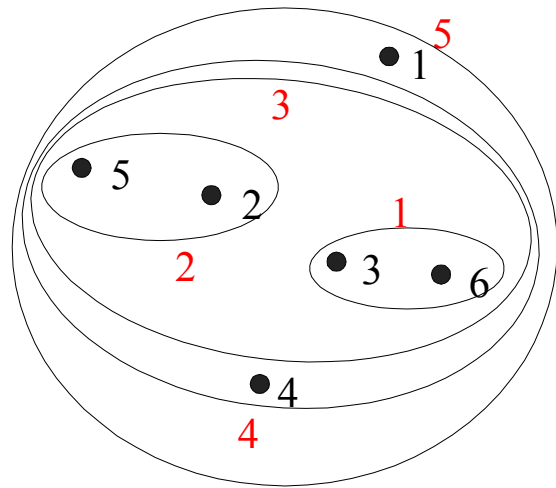
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

# Cluster Similarity: Ward's Method

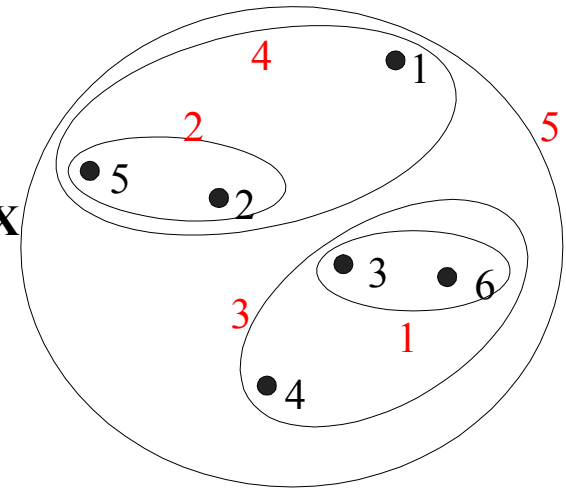
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means



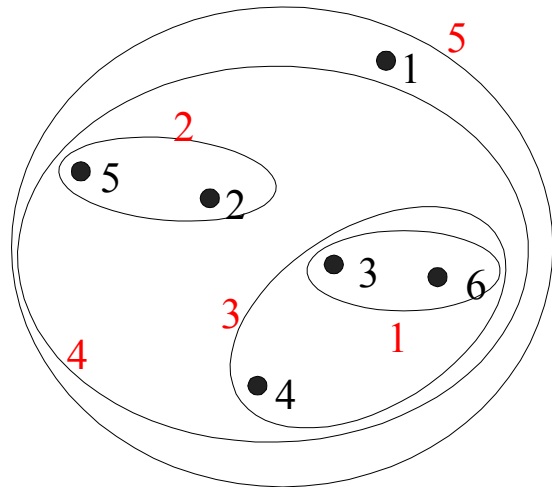
# Hierarchical Clustering: Comparison



MIN

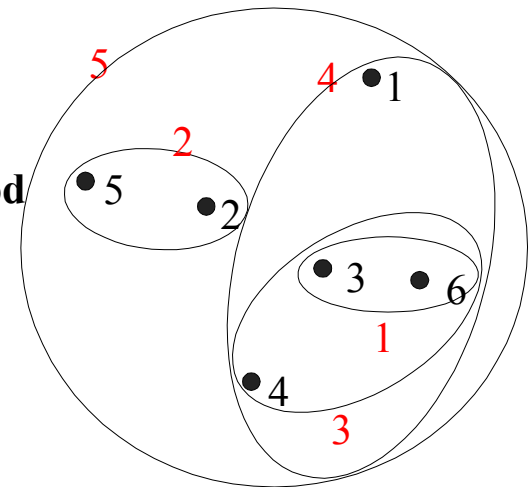


MAX



Group Average

Ward's Method



# Hierarchical Clustering: Time and Space requirements

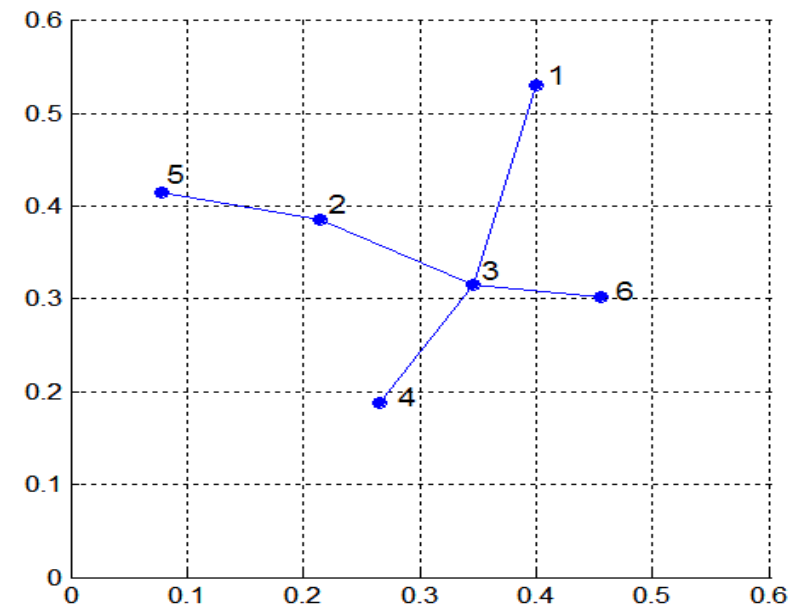
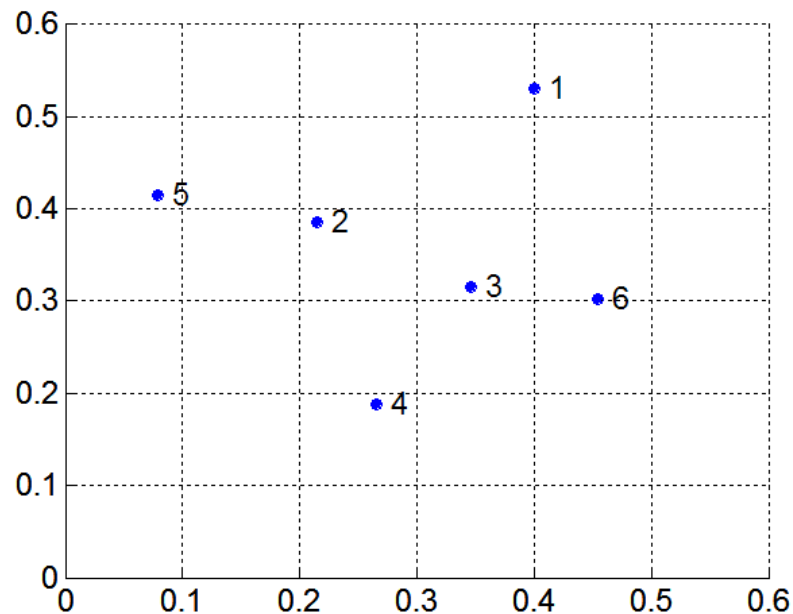
- $O(N^2)$  space since it uses the proximity matrix.
  - $N$  is the number of points.
- $O(N^3)$  time in many cases
  - There are  $N$  steps and at each step the size,  $N^2$ , proximity matrix must be updated and searched
  - Complexity can be reduced to  $O(N^2 \log(N))$  time for some approaches

# 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

# MST: Divisive Hierarchical Clustering

- 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



# MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

---

**Algorithm 7.5** MST Divisive Hierarchical Clustering Algorithm

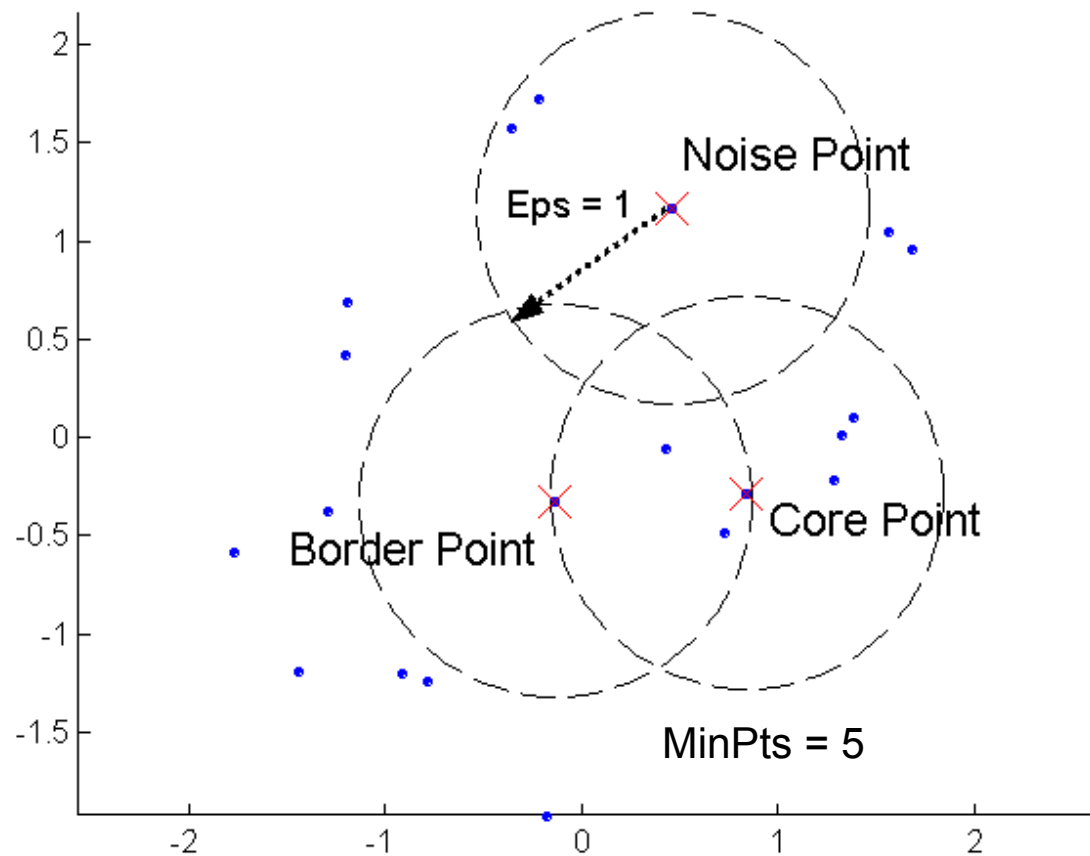
---

- 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
-

# DBSCAN

- 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

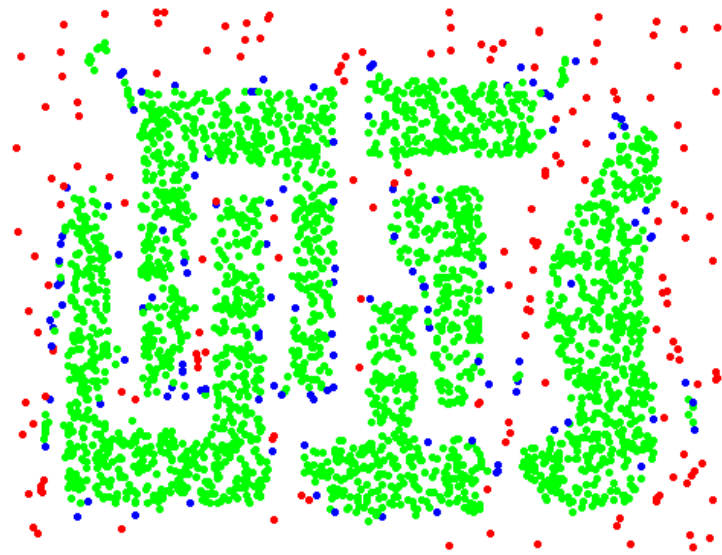
- Label all points as core, border or noise
- Eliminate noise points
- Put an edge between all core points that are within  $Eps$  of each other.
- Make each group of connected points into a separate cluster.
- Assign each border point to one of the clusters of its associated core points.



# DBSCAN: Core, Border and Noise Points



Original Points



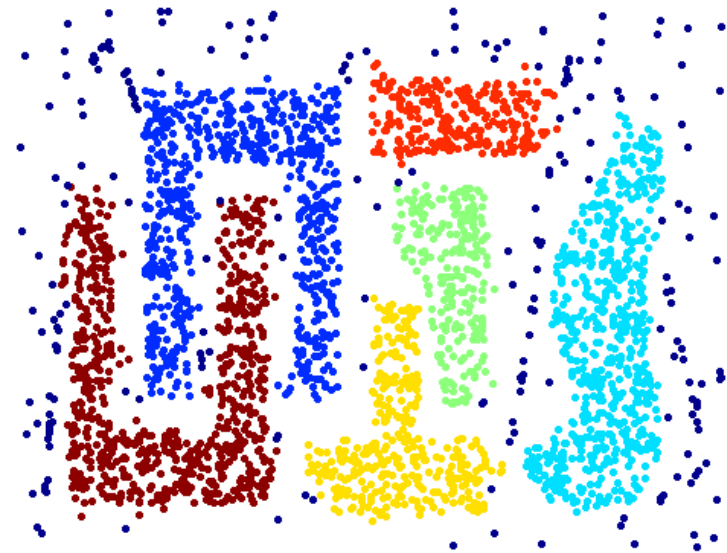
Point types: **core**,  
**border** and **noise**

Eps = 10, MinPts = 4

# When DBSCAN Works Well



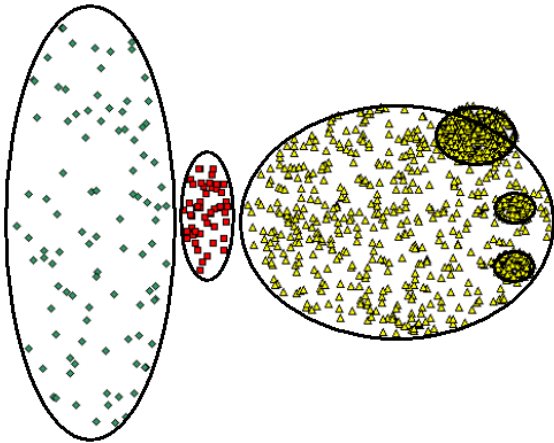
**Original Points**



**Clusters**

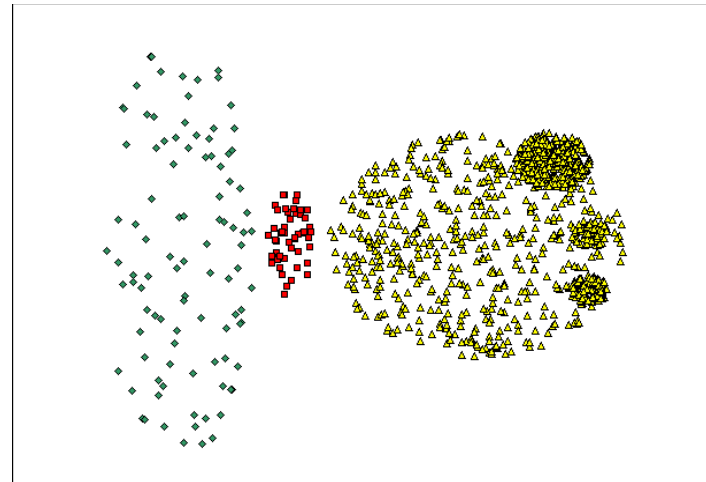
- **Resistant to Noise**
- **Can handle clusters of different shapes and sizes**

# When DBSCAN Does NOT Work Well

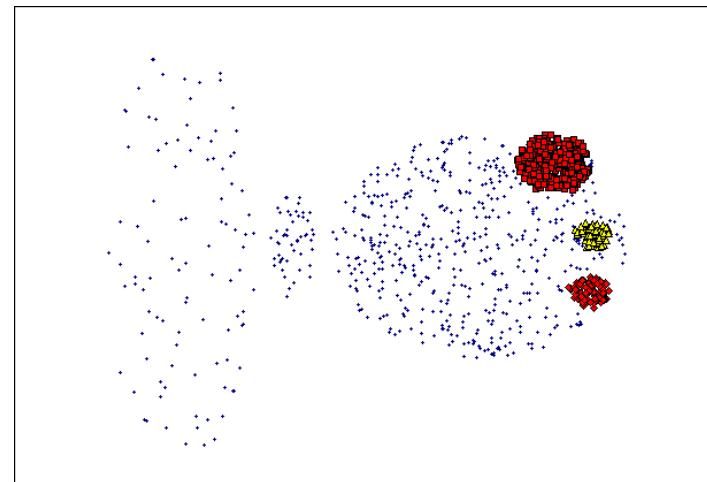


**Original Points**

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



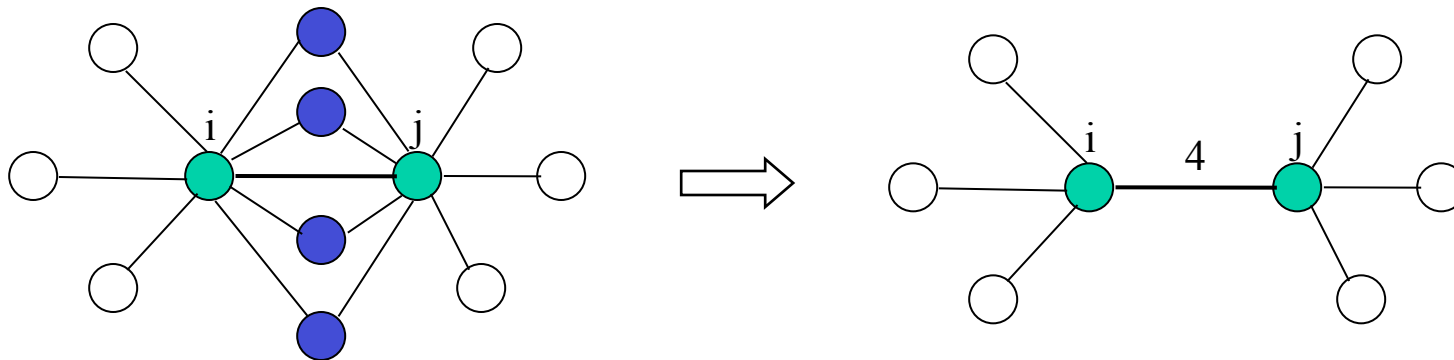
(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

# Shared Near Neighbor Approach

**SNN graph**: the weight of an edge is the number of shared neighbors between vertices given that the vertices are connected



# DBSCAN using SNN

## 1. Find the SNN density of each Point.

Using a user specified parameters,  $Eps$ , find the number points that have an SNN similarity of  $Eps$  or greater to each point. This is the SNN density of the point

## 5. Find the core points

Using a user specified parameter,  $MinPts$ , find the core points, i.e., all points that have an SNN density greater than  $MinPts$

## 6. Form clusters from the core points

If two core points are within a radius,  $Eps$ , of each other they are place in the same cluster

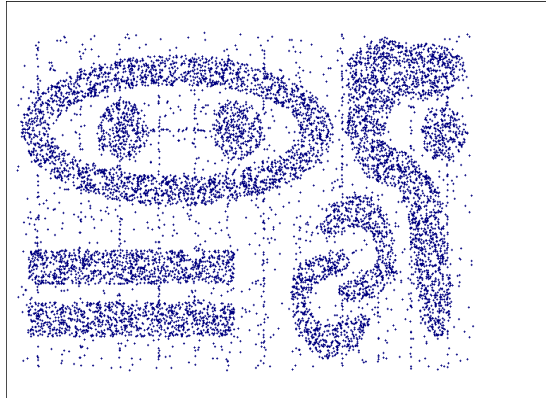
## 7. Discard all noise points

All non-core points that are not within a radius of  $Eps$  of a core point are discarded

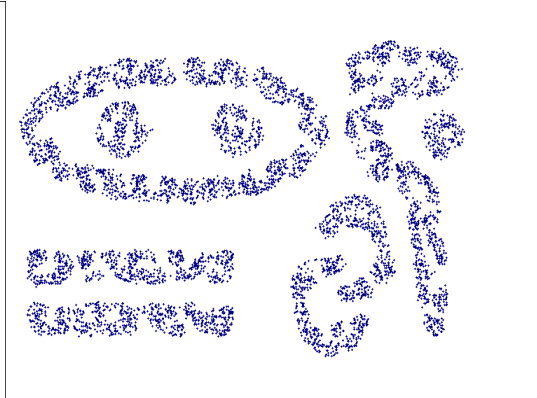
## 8. Assign all non-noise, non-core points to clusters

This can be done by assigning such points to the nearest core point

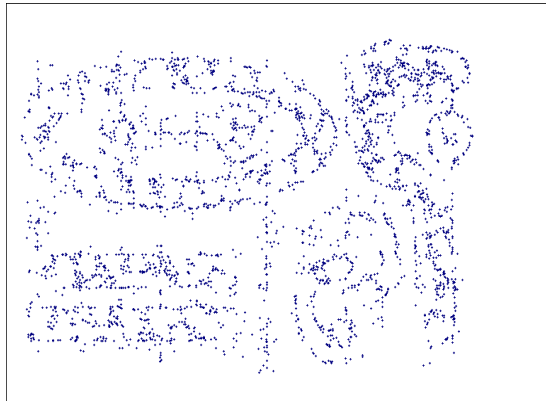
# SNN Density



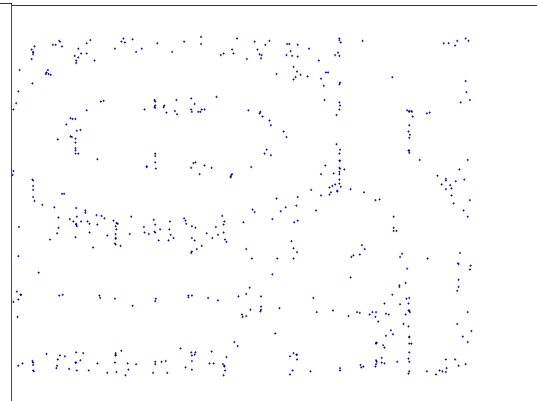
a) All Points



b) High SNN Density

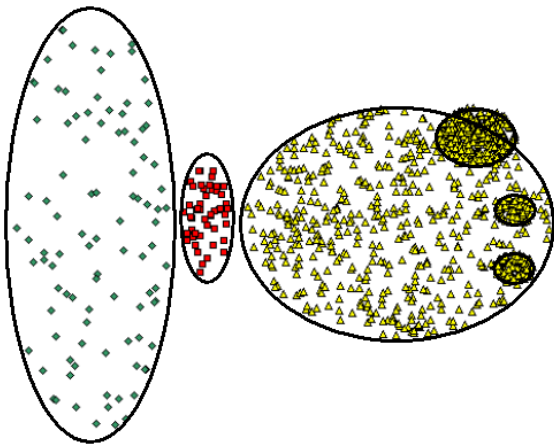


c) Medium SNN Density

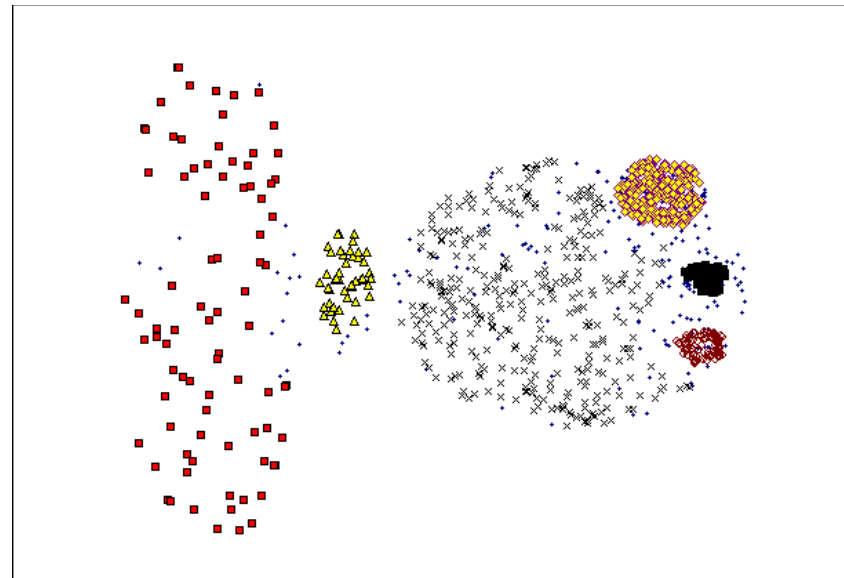


d) Low SNN Density

# SNN Clustering Can Handle Differing Densities

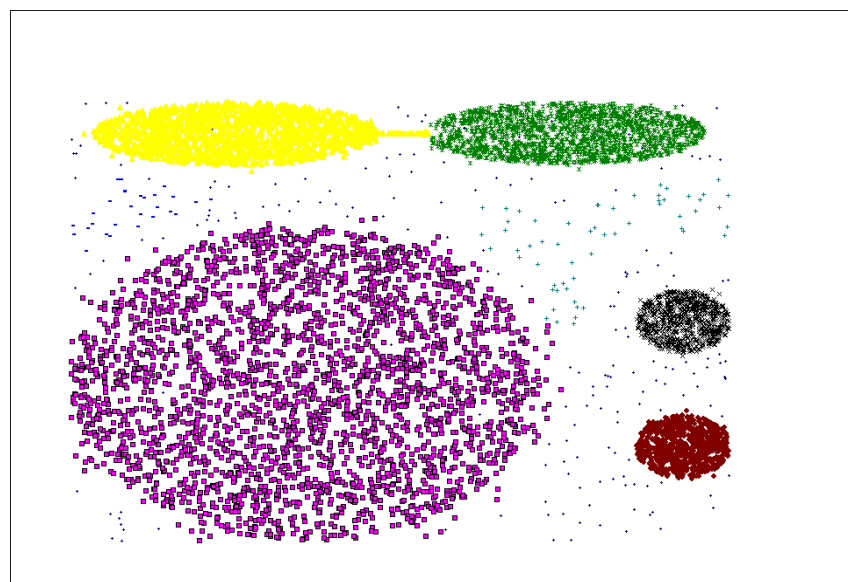
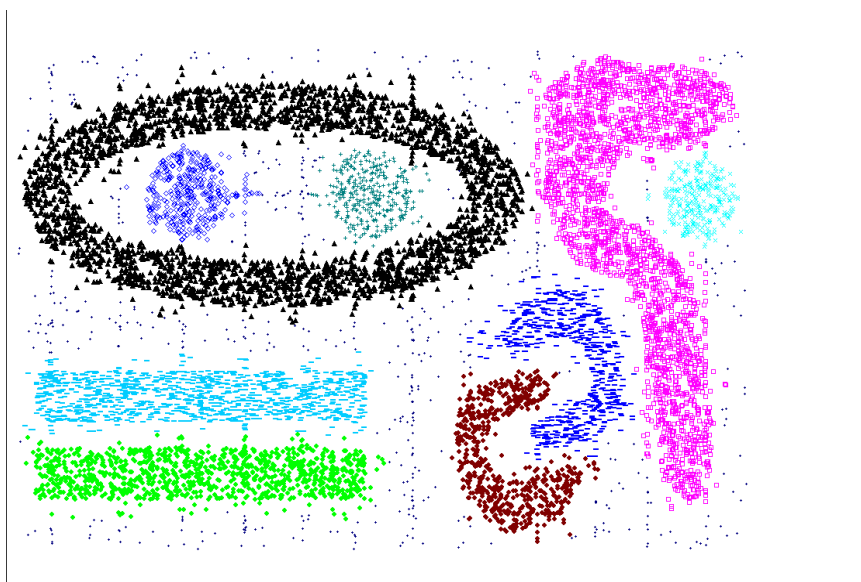


Original Points



SNN Clustering

# SNN Clustering Can Handle Other Difficult Situations



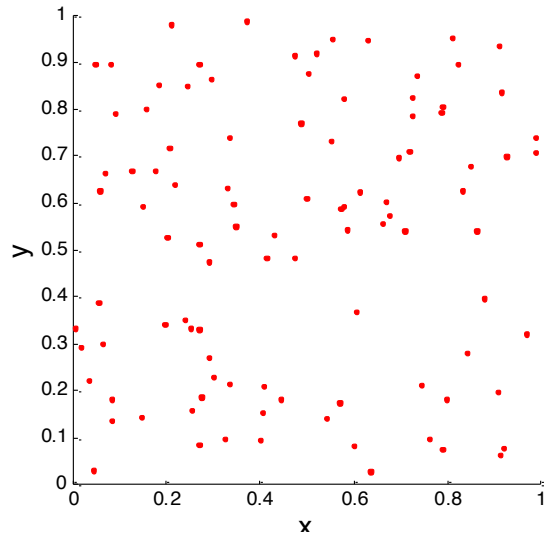


# Cluster Validity

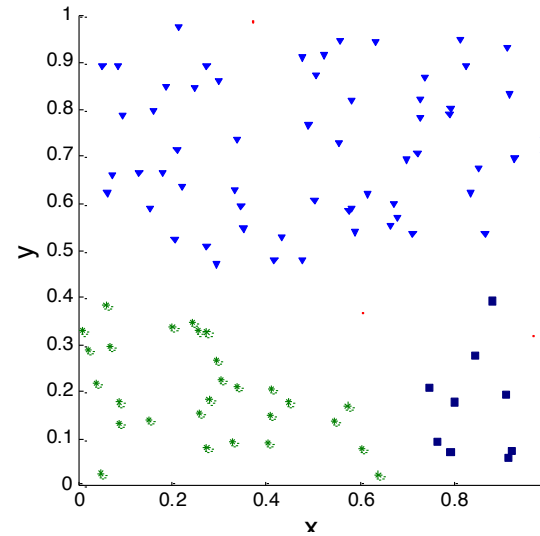
- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
- Then why do we 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

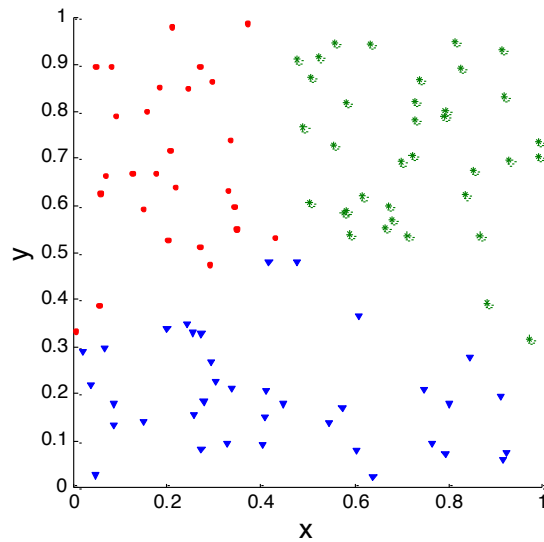
**Random Points**



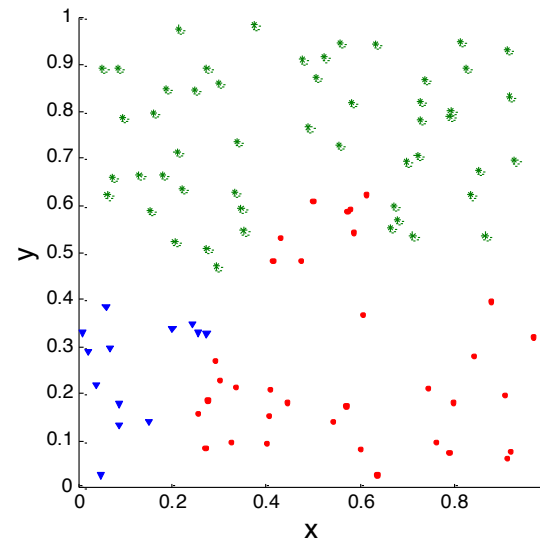
**DBSCAN**



**K-means**



**Complete Link**



# Different Aspects of Cluster Validation

1. Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
  - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the ‘correct’ number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

# Measures of Cluster Validity

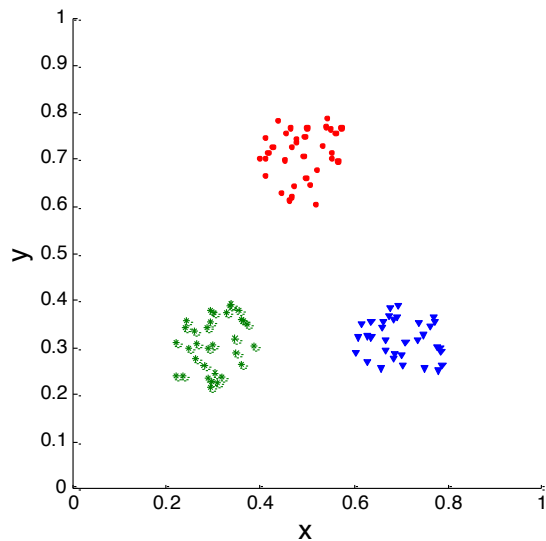
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - **External Index:** Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - **Internal Index:** Used to measure the 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

# Measuring Cluster Validity Via Correlation

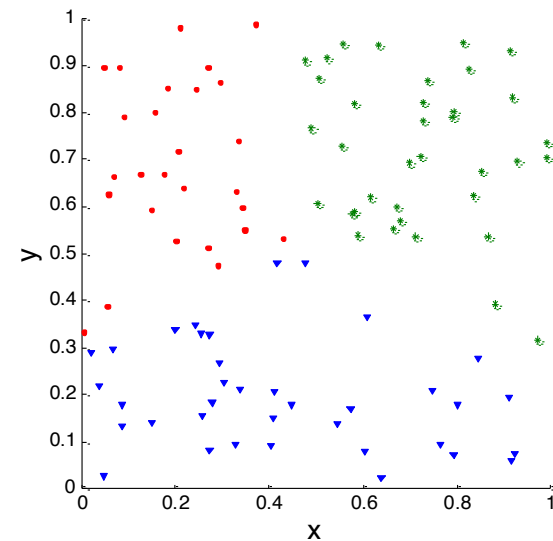
- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between 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.



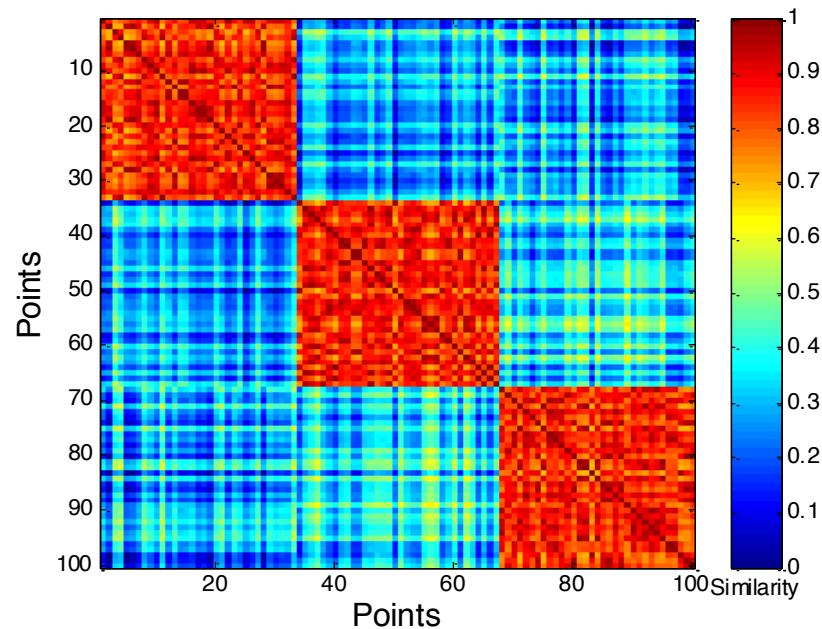
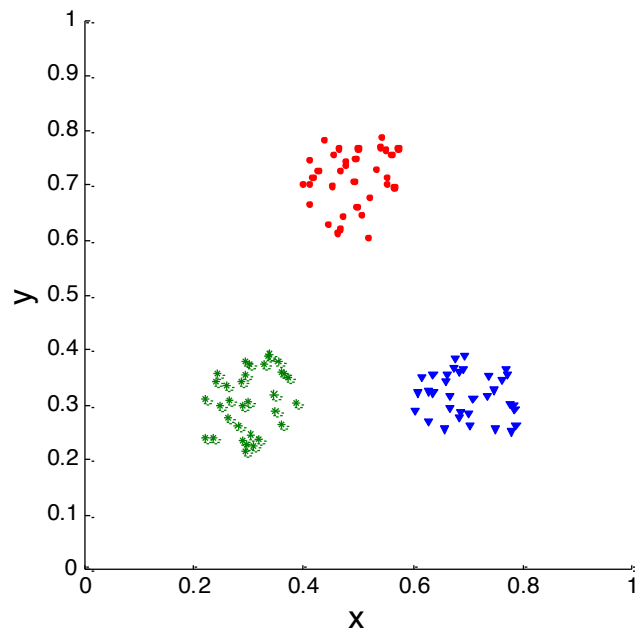
**Corr = -0.9235**



**Corr = -0.5810**

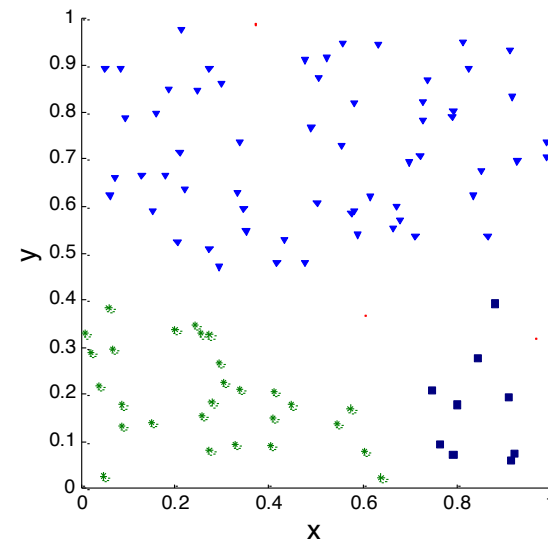
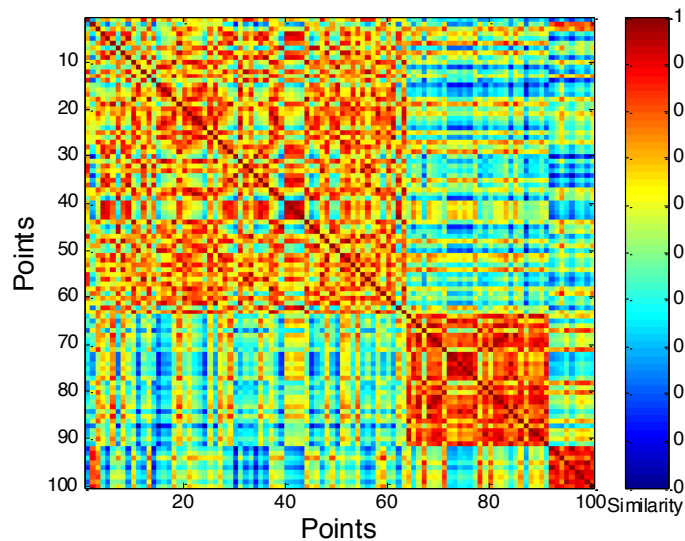
# Using Similarity Matrix for Cluster Validation

- 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

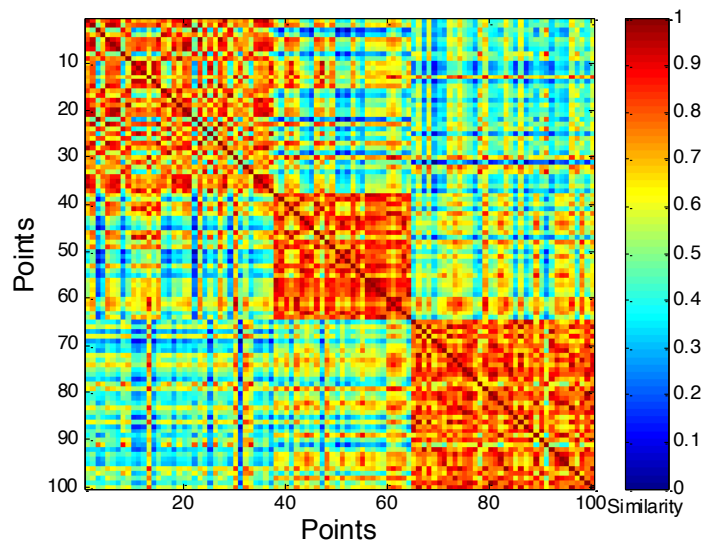


**DBSCAN**

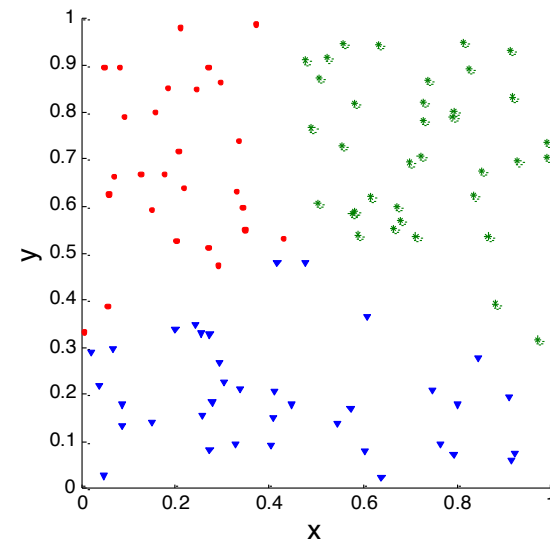


# Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

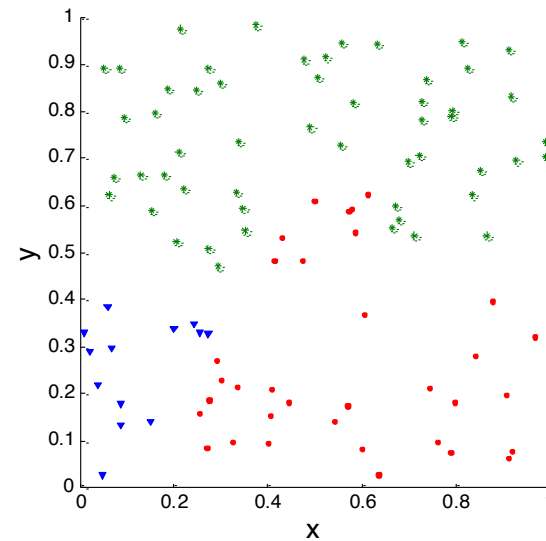
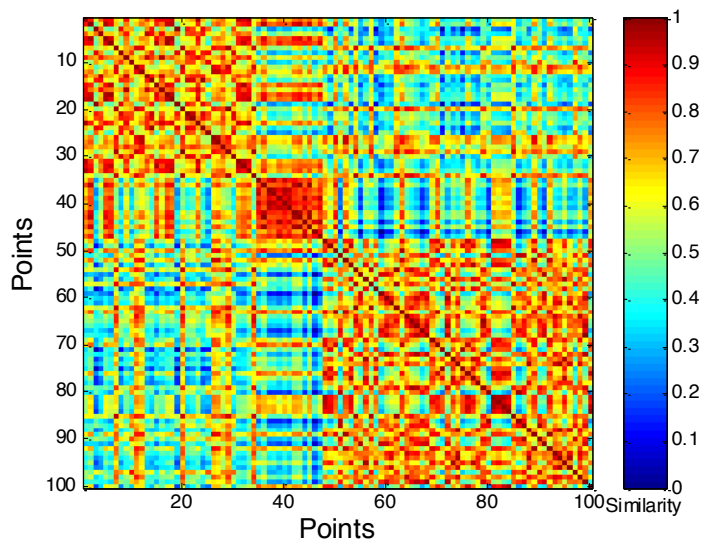


**K-means**



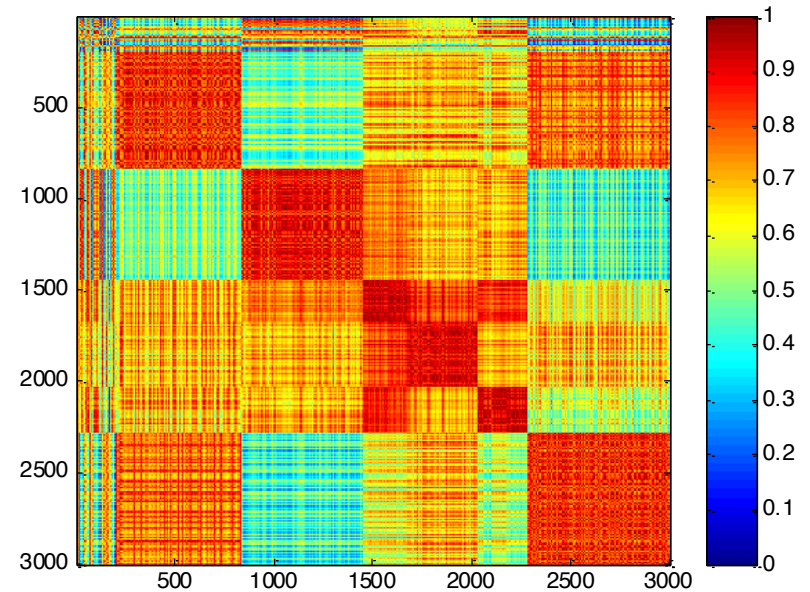
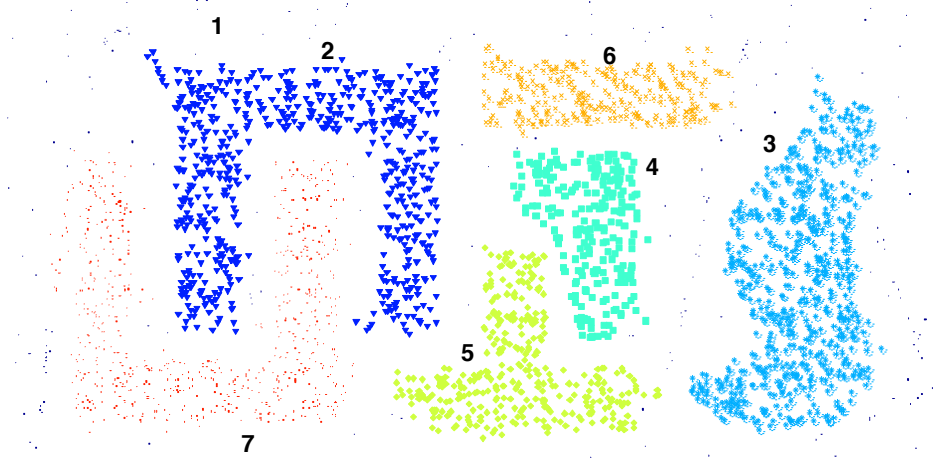
# Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp



**Complete Link**

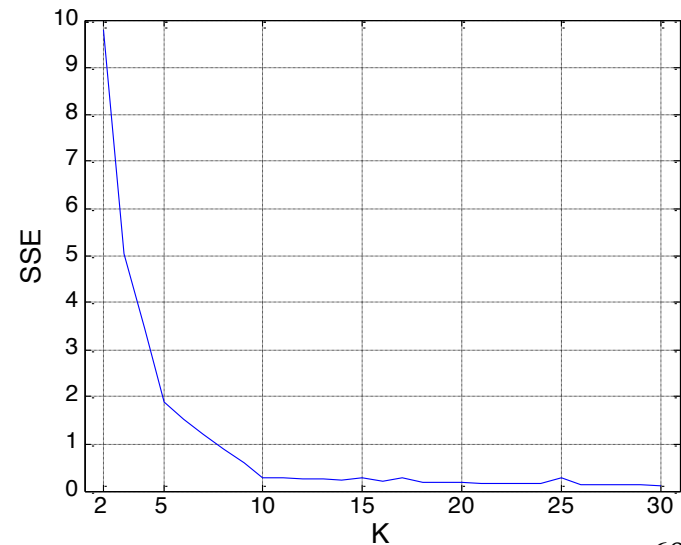
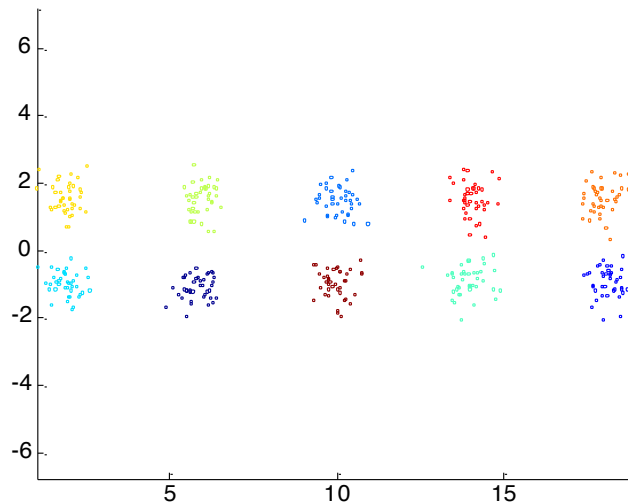
# Using Similarity Matrix for Cluster Validation



**DBSCAN**

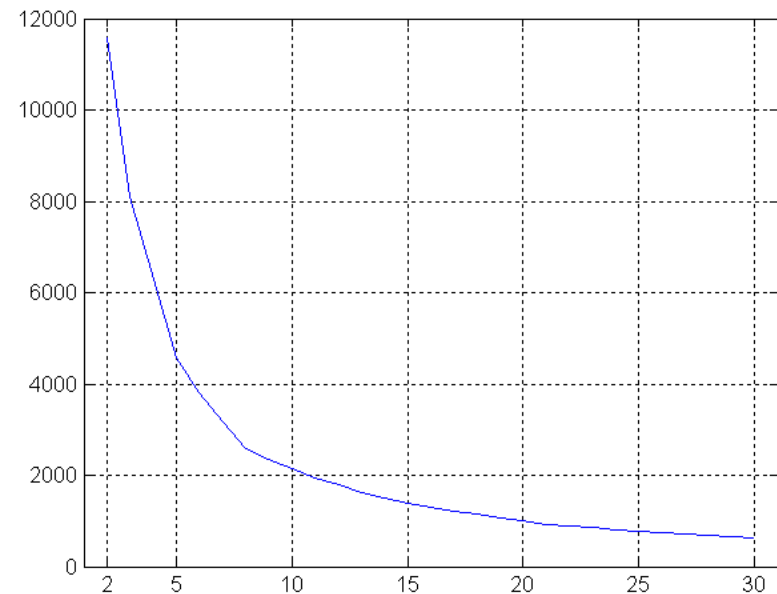
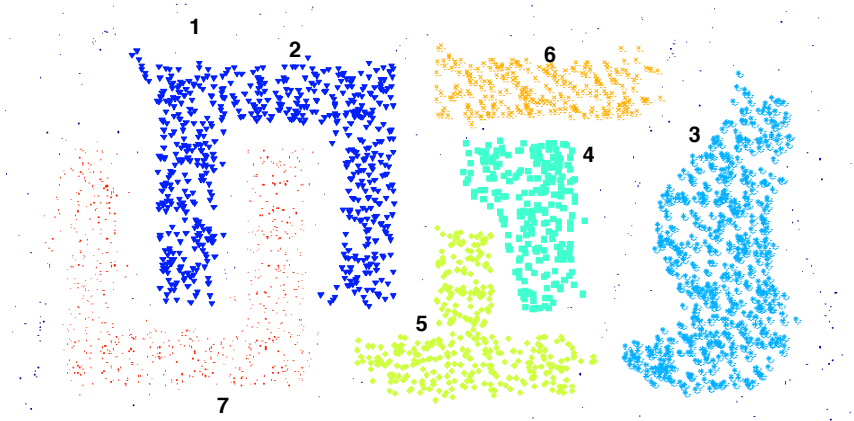
# Internal Measures: SSE

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters



# Internal Measures: SSE

- SSE curve for a more complicated data set



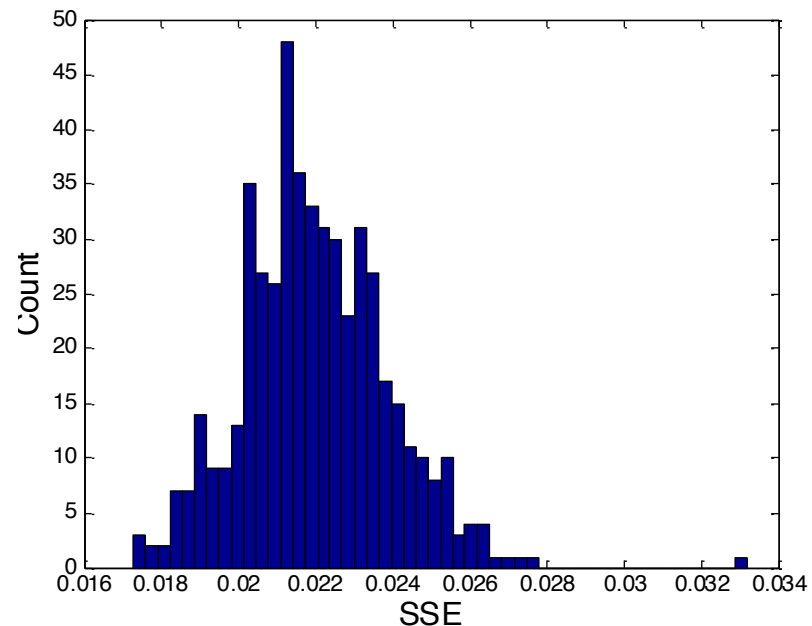
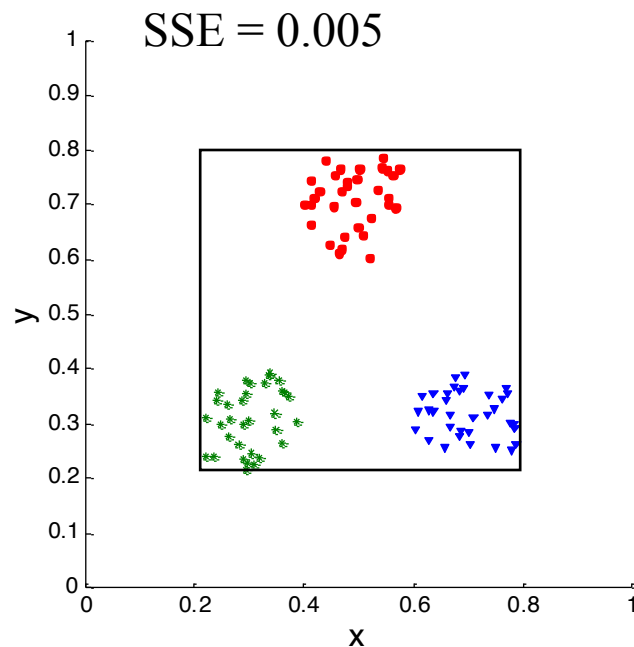
**SSE of clusters found using K-means**

# Framework for Cluster Validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
  - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
    - If the value of the index is unlikely, then the cluster results are valid
  - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant

# Statistical Framework for SSE

- Example
  - Compare SSE of 0.005 against three clusters in random data
  - Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values



# 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)

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

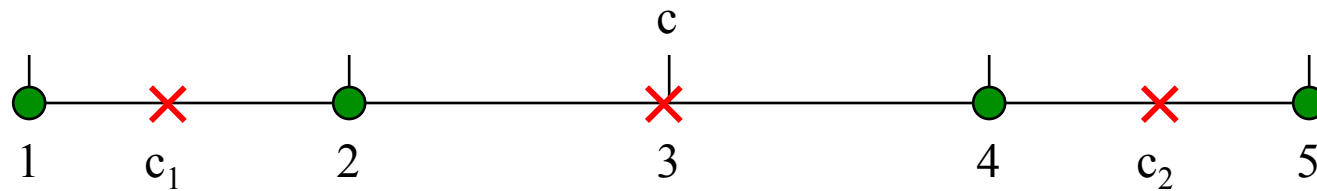
- Separation is measured by the between cluster sum of squares, or by between cluster to overall prototype sum of squares (shown)

$$SSB = \sum_i |C_i| (c - c_i)^2$$

where  $|C_i|$  is the size of cluster  $i$ ,  $c_i$  is the centroid of cluster  $i$ , and  $c$  is the overall centroid



# Total Sum of Squares (TSS)



K=1 cluster:

$$TSS = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$$

$$SSE = (3 - 1)^2 + (3 - 2)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$$

$$SSB = 4 \times (3 - 3)^2 = 0$$

K=2 clusters:

$$TSS = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$$

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

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

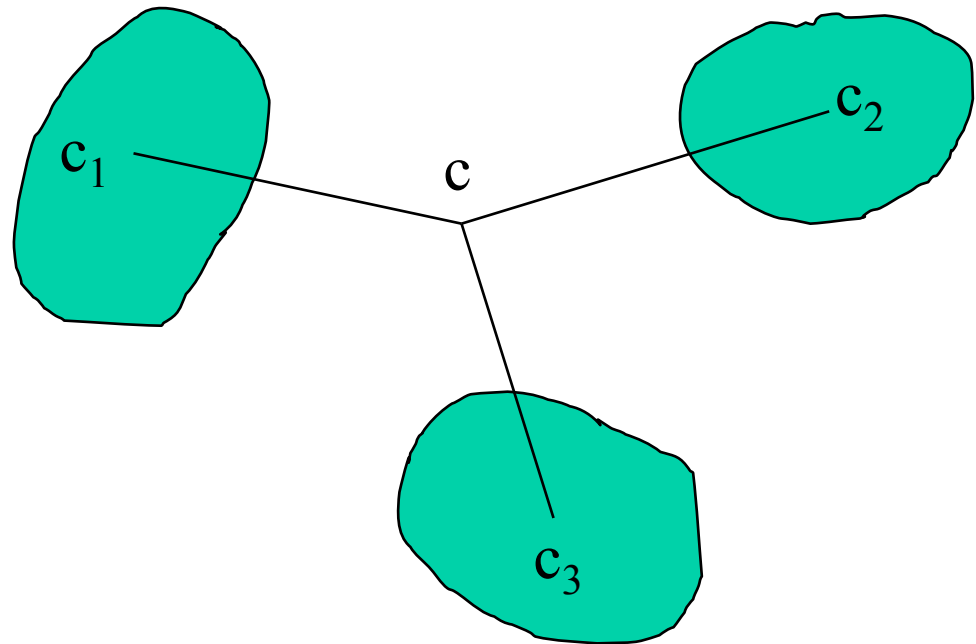
$$\mathbf{TSS = SSE + SSB}$$

# Total Sum of Squares (TSS)

$$TSS = \sum dist(x, c)^2$$

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

$$SSB = \sum_{i=1}^k |C_i| dist(c_i, c)^2$$



$c$ : overall mean

$c_i$ : centroid of each cluster  $C_i$

$|C_i|$ : number of points in cluster  $C_i$

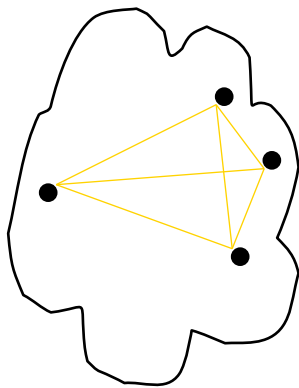
# Total Sum of Squares (TSS)

$$\mathbf{TSS = SSE + SSB}$$

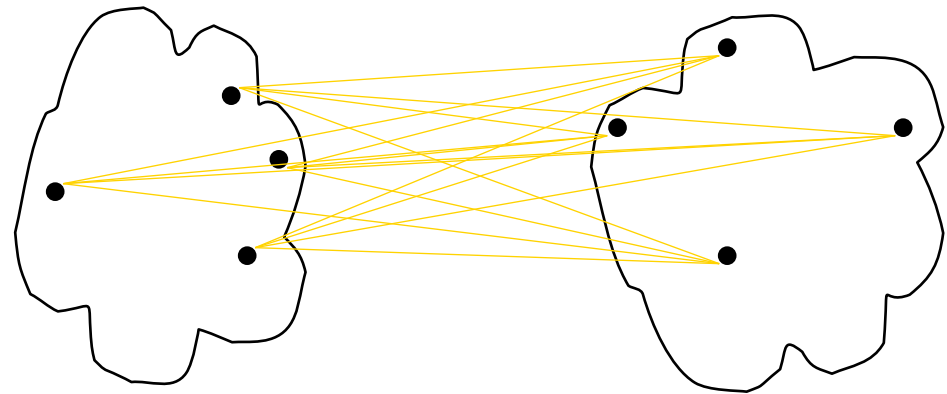
- Given a data set, TSS is fixed
- A clustering with large SSE has small SSB, while one with small SSE has large SSB
- Goal is to minimize SSE and maximize SSB

# Internal 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.



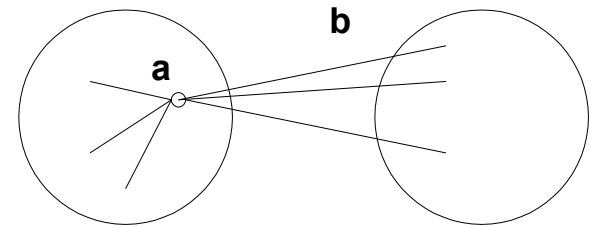
cohesion



separation

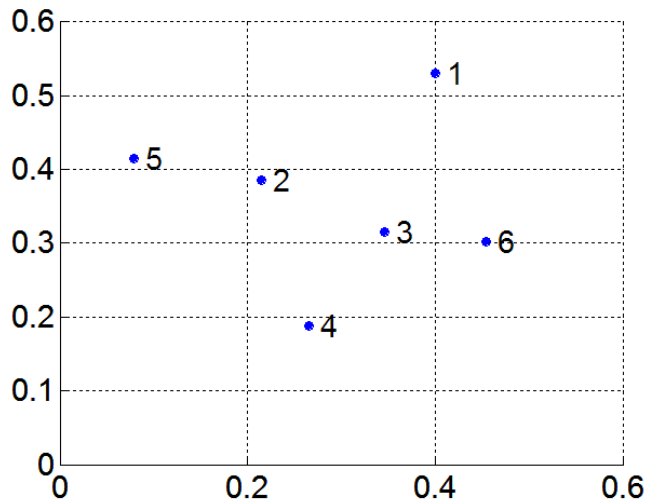
# Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point,  $i$ 
  - Calculate  $a$  = average distance of  $i$  to the points in its cluster
  - Calculate  $b$  = min (average distance of  $i$  to points in another cluster)
  - The silhouette coefficient for a point is then given by  
 $s = 1 - a/b$  if  $a < b$ , (or  $s = b/a - 1$  if  $a \geq b$ , not the usual case)
  - Typically between 0 and 1 (but can be negative if  $a \geq b$ ).
  - The closer to 1 the better.



- Can calculate the Average Silhouette width for a cluster or a clustering

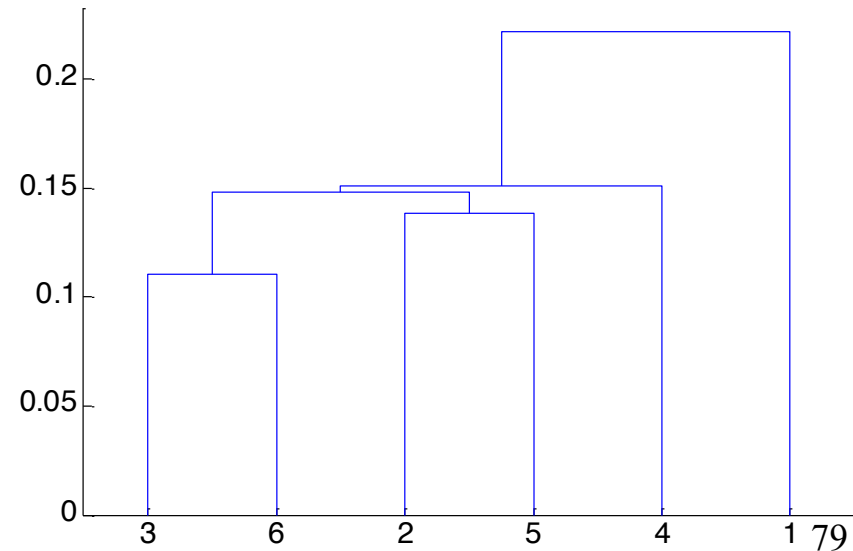
# Unsupervised Evaluation of Hierarchical Clustering



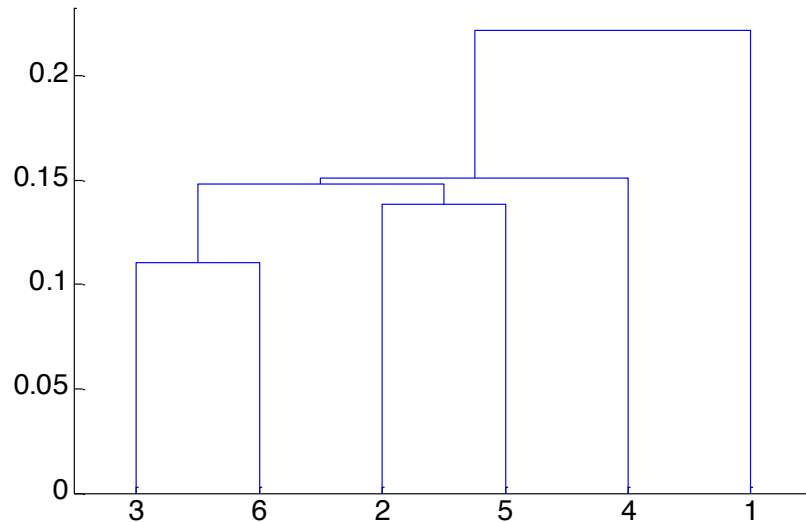
Single Link

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



# Unsupervised Evaluation of Hierarchical Clustering



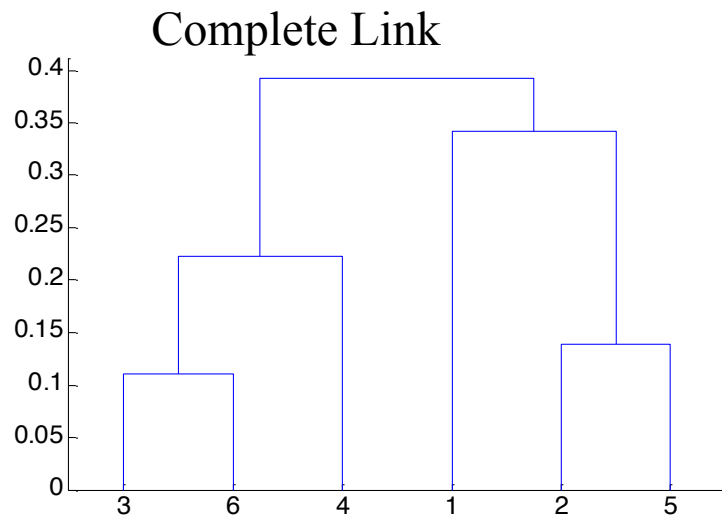
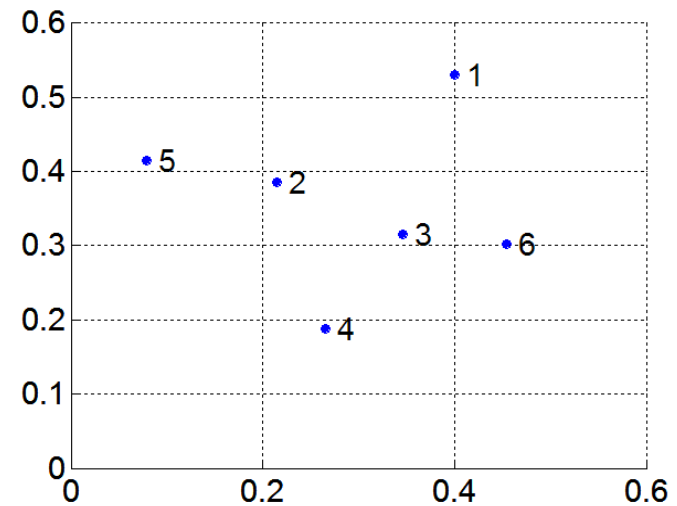
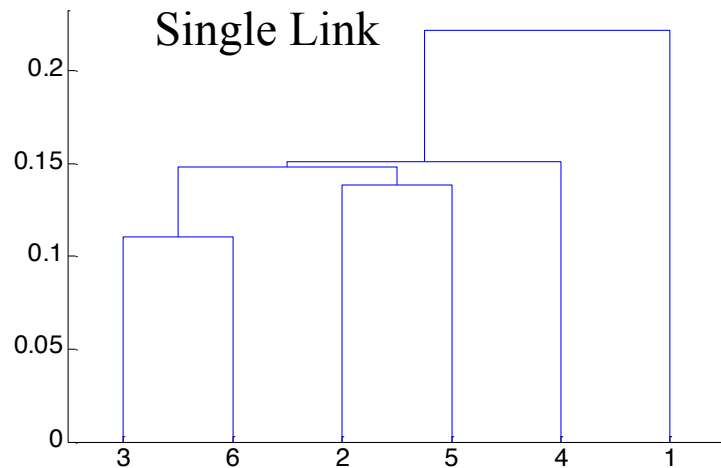
Single Link

Point	P1	P2	P3	P4	P5	P6
P1	0	0.222	0.222	0.222	0.222	0.222
P2	0.222	0	0.148	0.151	0.139	0.148
P3	0.222	0.148	0	0.151	0.148	0.110
P4	0.222	0.151	0.151	0	0.151	0.151
P5	0.222	0.139	0.148	0.151	0	0.148
P6	0.222	0.148	0.110	0.151	0.148	0

Cophenetic Distance Matrix for Single Link

- Cophenetic distance
  - the proximity at which the clustering technique puts the objects in the same cluster for the first time.
  - E.g. if two clusters are merged with distance = 0.1, then all points in one cluster have a cophenetic distance of 0.1 wrt the points in the other cluster.
- CPCC (CoPhenetic Correlation Coefficient)
  - Correlation between original distance matrix and cophenetic distance matrix

# Unsupervised Evaluation of Hierarchical Clustering



Technique	CPCC
Single Link	0.44
Complete Link	0.63
Group Average	0.66
Ward's	0.64



# External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

**entropy** For each cluster, the class distribution of the data is calculated first, i.e., for cluster  $j$  we compute  $p_{ij}$ , the ‘probability’ that a member of cluster  $j$  belongs to class  $i$  as follows:  $p_{ij} = m_{ij}/m_j$ , where  $m_j$  is the number of values in cluster  $j$  and  $m_{ij}$  is the number of values of class  $i$  in cluster  $j$ . Then using this class distribution, the entropy of each cluster  $j$  is calculated using the standard formula  $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$ , where the  $L$  is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e.,  $e = \sum_{i=1}^K \frac{m_i}{m} e_j$ , where  $m_j$  is the size of cluster  $j$ ,  $K$  is the number of clusters, and  $m$  is the total number of data points.

**purity** Using the terminology derived for entropy, the purity of cluster  $j$ , is given by  $purity_j = \max p_{ij}$  and the overall purity of a clustering by  $purity = \sum_{i=1}^K \frac{m_i}{m} purity_j$ .

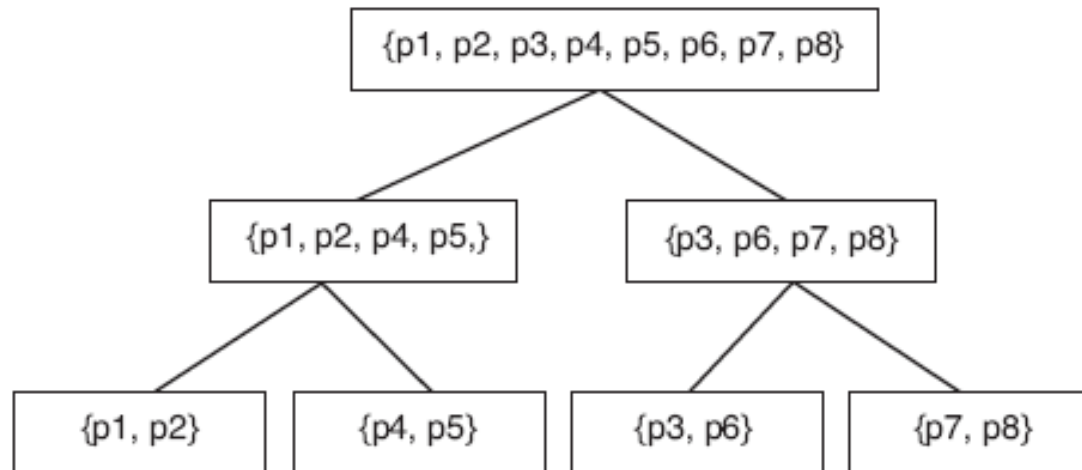
# Supervised Cluster Validation: Precision and Recall

Cluster i  
 $m_{i1}$ : class 1  
 $m_{i2}$ : class 2

Overall Data  
 $m_1$ : class 1  
 $m_2$ : class 2

- Precision for cluster i w.r.t. class j =  $\frac{m_{ij}}{\sum_k m_{ik}}$
- Recall for cluster i w.r.t. class j =  $\frac{m_{ij}}{\sum_k m_{kj}} = \frac{m_{ij}}{m_j}$

# Supervised Cluster Validation: Hierarchical Clustering



Hierarchical F-measure:

$$F = \sum_j \frac{m_j}{m} \max_i F(i, j)$$

where the maximum is taken over all clusters  $i$  at all levels,  $m_j$  is the number of objects in class  $j$ , and  $m$  is the total number of objects.

# Supervised Cluster Validation: Binary Similarity

- Consider all pairs of distinct objects
  - $f_{00}$  = # of pairs of objects having a different class and a different cluster
  - $f_{01}$  = # of pairs of objects having a different class and the same cluster
  - $f_{10}$  = # of pairs of objects having the same class and a different cluster
  - $f_{11}$  = # of pairs of objects having the same class and the same cluster

# Supervised Cluster Validation: Binary Similarity

- Rand Statistic (Simple matching coefficient):

$$\frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

- Jaccard Coefficient:

$$\frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

	Same Cluster	Different Cluster
Same Class	f11	f10
Different Class	f01	f00

# Final Comment on Cluster Validity

- “The validation of clustering structures is the most difficult and frustrating part of cluster analysis.
- Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

*Algorithms for Clustering Data*, by Jain and Dubes