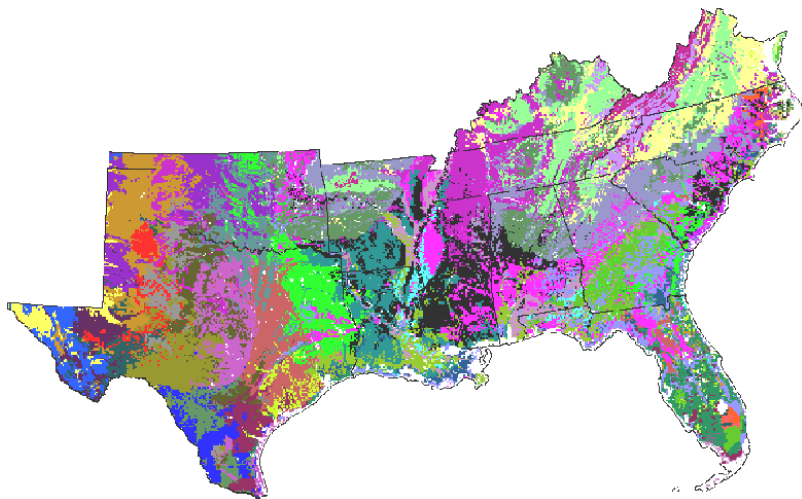


# Clustering

April 8, 2021  
Slides Courtesy of Jiayu Zhou

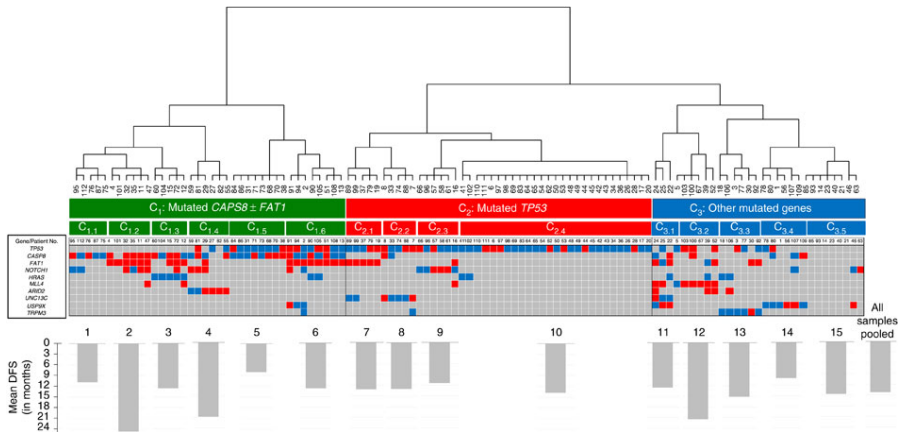
# Clustering Application - Geography

13 States Clustered into 51 Custom Eco-regions




# Clustering Application - Cancer Patients

Clustering of gingivo-buccal oral cancer patients based on mutational profiles





Mutational landscape of gingivo-buccal oral squamous cell carcinoma reveals new recurrently-mutated genes and molecular subgroups, Nature Communications, 2013.

# Clustering Application - Search Result Clustering




company | products | solutions | customers | demos | partners | press









 


[Other demos](#) | [Help!](#) | [Tell us what you think!](#)

**Clustered Results**

**Top 185 results retrieved for the query [jaguar](#) (Details)**

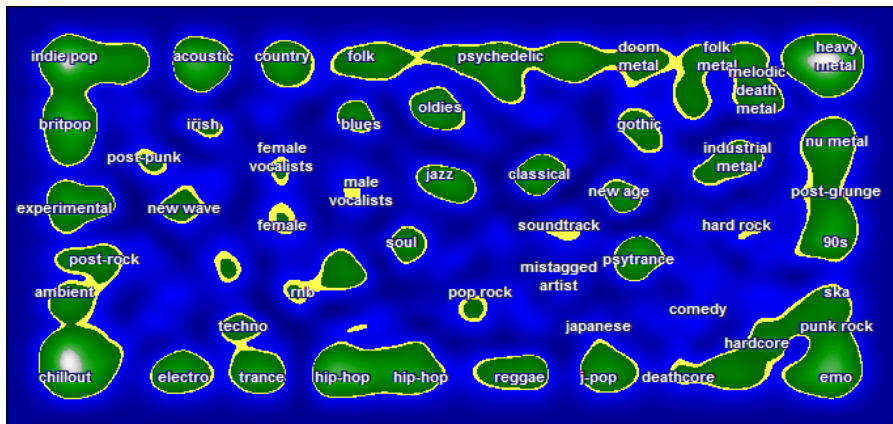
 [jaguar](#) (185)

-  [Cars](#) (56)
-  [Club](#) (35)
-  [Parts](#) (26)
-  [Racing](#) (15)
-  [Models](#) (12)
-  [Atari](#) (11)
  - [History](#) (8)
  - [Classic Jaguar](#) (8)
-  [International Jaguar](#) (6)
-  [Jaguar Dealership](#) (7)
- [More](#)

Find in clusters:  
 

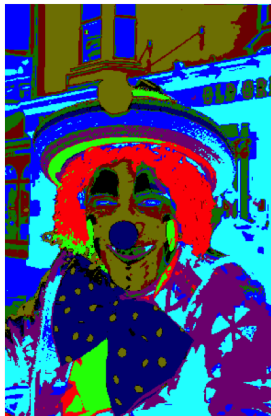
- [Jaguar Cars](#)** [\[new window\]](#) [\[frame\]](#) [\[preview\]](#)  
Official worldwide web site of **Jaguar** Cars. Gama actual, concesionarios, historia, noticias, anuncios y servicios fina  
URL: [www.jaguar.com](#) - [show in clusters](#)  
Sources: [Lycos 1](#)
- [Jaguar Cars](#)** [\[new window\]](#) [\[frame\]](#) [\[preview\]](#)  
URL: [www.jaguarcars.com](#) - [show in clusters](#)  
Sources: [Lycos 2](#), [Lycos 50](#), [Lycos 90](#), [Lycos 97](#), [Lycos 99](#)
- [www.jaguar-racing.com](#)** [\[new window\]](#) [\[frame\]](#) [\[preview\]](#)  
URL: [www.jaguar-racing.com](#) - [show in clusters](#)  
Sources: [Lycos 3](#), [Lycos 93](#), [Lycos 116](#)
- [Jaguar Cars](#)** [\[new window\]](#) [\[frame\]](#) [\[preview\]](#)  
United States United Kingdom Germany Japan France Italy Spain...  
URL: [www.jaguarehicles.com](#) - [show in clusters](#)  
Sources: [Lycos 4](#), [Lycos 8](#), [Lycos 41](#), [Lycos 102](#), [Lycos 188](#)
- [Apple - Mac OS X](#)** [\[new window\]](#) [\[frame\]](#) [\[preview\]](#)  
... queries to find your stuff, refining the list as you narrow options. Sure you could quantify that as up to six times fa:  
**Jaguar** , but you'll probably think Panthers done almost before you...  
URL: [www.apple.com/macosx](#) - [show in clusters](#)  
Sources: [Lycos 5](#)

# Clustering Application - Islands of Music

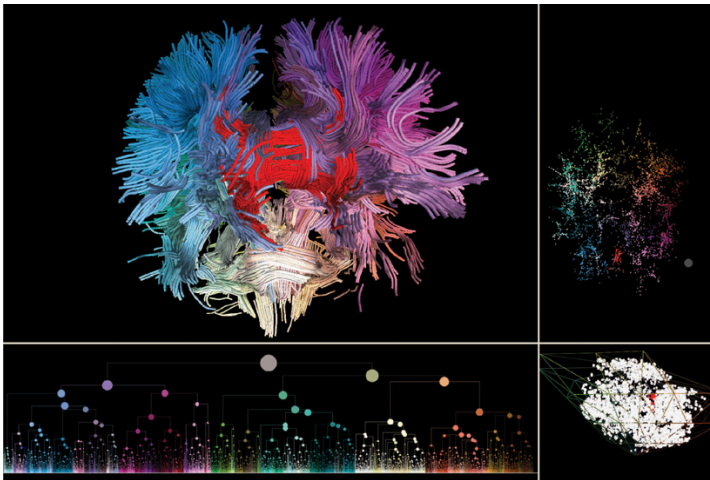


Pampalk, Elias, Andreas Rauber, and Dieter Merkl. "Content-based organization and visualization of music archives." Proceedings of the Tenth ACM International Conference on Multimedia. ACM, 2002.

# Clustering Application - Image Compression



# Clustering Application - MRI TDI Fibers



“Exploring 3D TDI fiber tracts with linked 2D representations.” Visualization and Computer Graphics, IEEE Transactions on 15.6 (2009): 1449-1456.

# Notion of a Cluster Can Be Ambiguous



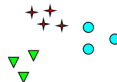
How many clusters?



# Notion of a Cluster Can Be Ambiguous



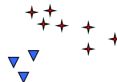
How many clusters?



Six Clusters



Two Clusters



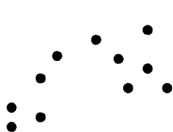
Four Clusters



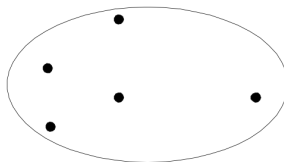
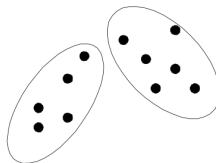
# Types of Clustering

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

# Partitional Clustering

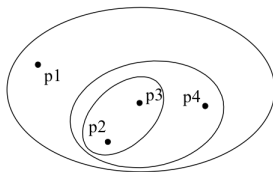


**Original Points**

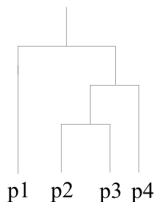


**A Partitional Clustering**

# Hierarchical clustering



**Traditional Hierarchical Clustering**



**Traditional Dendrogram**

# *K*-means for Clustering

# K-means 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

# K-means 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
- Optimization objective

$$\arg \min_{\{c_j, m_{i,j}\}} \sum_{j=1}^K \sum_{i=1}^n m_{i,j} \|x_i - c_j\|^2$$

where memberships  $\{m_{i,j}\}$  and centers  $\{c_j\}$  are correlated

# K-means Clustering Algorithm

$$\arg \min_{\{c_j, m_{i,j}\}} \sum_{j=1}^K \sum_{i=1}^n m_{i,j} \|x_i - c_j\|^2$$



# K-means Clustering Algorithm

$$\arg \min_{\{c_j, m_{i,j}\}} \sum_{j=1}^K \sum_{i=1}^n m_{i,j} \|x_i - c_j\|^2$$

Alternating procedure:

- Given centroids  $\{c_j\}$ ,  $m_{i,j} = \begin{cases} 1 & j = \arg \min_{j \in [1 \dots K]} \|x_i - c_j\|^2 \\ 0 & \text{otherwise} \end{cases}$
- Given memberships  $\{m_{i,j}\}$ ,  $c_j = \frac{\sum_{i=1}^n m_{i,j} x_i}{\sum_{i=1}^n m_{i,j}}$

# K-means Clustering Algorithm

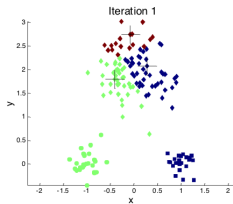
$$\arg \min_{\{c_j, m_{i,j}\}} \sum_{j=1}^K \sum_{i=1}^n m_{i,j} \|x_i - c_j\|^2$$

Alternating procedure:

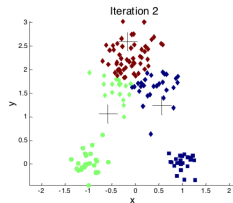
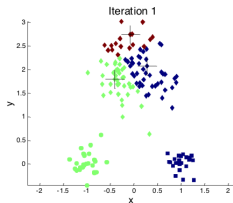
- Given centroids  $\{c_j\}$ ,  $m_{i,j} = \begin{cases} 1 & j = \arg \min_{j \in [1 \dots K]} \|x_i - c_j\|^2 \\ 0 & \text{otherwise} \end{cases}$
- Given memberships  $\{m_{i,j}\}$ ,  $c_j = \frac{\sum_{i=1}^n m_{i,j} x_i}{\sum_{i=1}^n m_{i,j}}$

- 
- 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
-

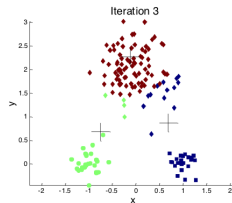
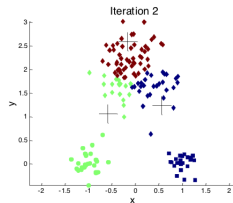
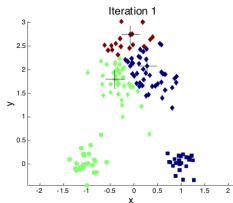
# K-means Illustration



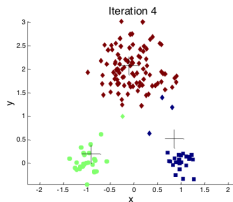
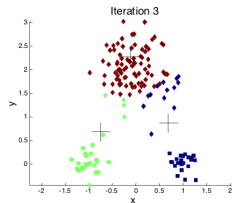
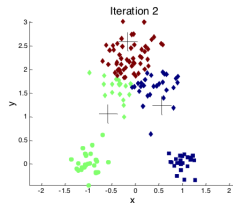
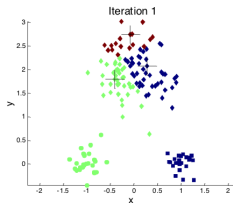
# K-means Illustration



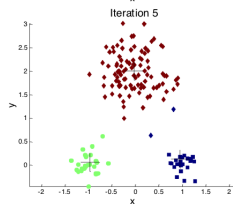
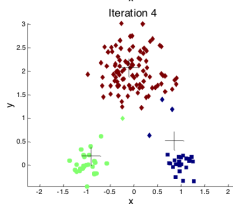
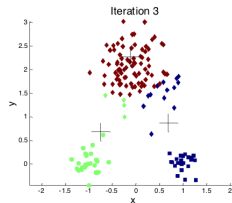
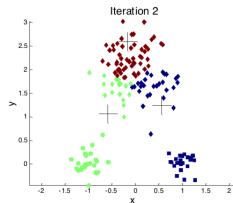
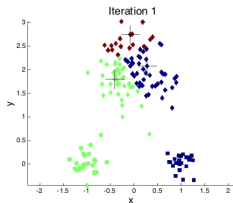
# K-means Illustration



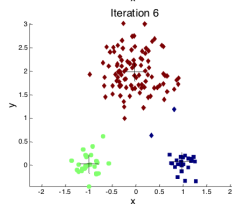
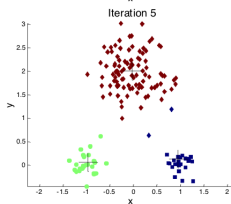
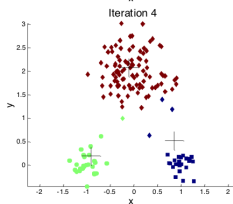
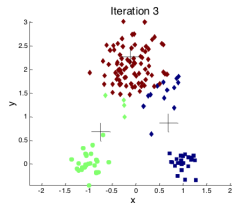
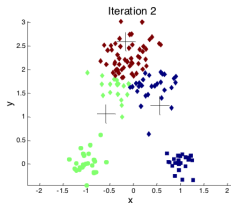
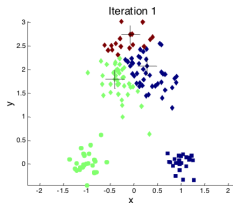
# K-means Illustration



# K-means Illustration



# K-means Illustration

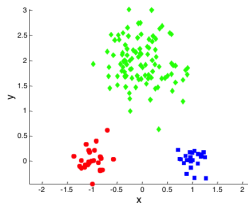




# K-means Clustering Details

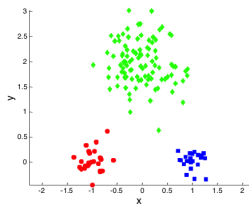
- Initial centroids are often chosen randomly
- Centroid is (typically) the mean of the points in the cluster
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- $K$ -means will converge for these common similarity measures
- Most of the convergence happens in the first few iterations.  
Typically the stopping condition is changed to “Until relatively few points change clusters”.
- Let  $n$  = number of points,  $K$  = number of clusters,  $I$  = number of iterations,  $d$  = number of attributes: complexity is  $O(n \times K \times I \times d)$

# K-means Revisited

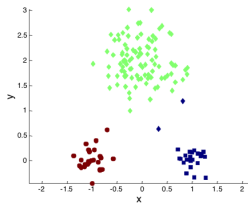


**Original Points**

# K-means Revisited

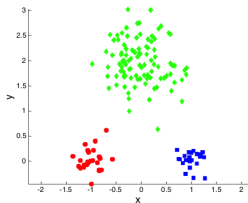


**Original Points**

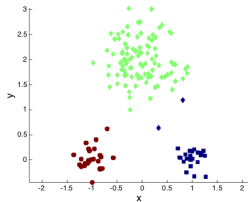


**Optimal Clustering**

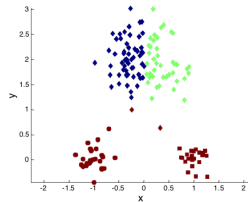
# K-means Revisited



**Original Points**



**Optimal Clustering**



**Sub-optimal Clustering**

# Problems with Selecting Initial Points

If there are  $K$  “real” clusters then the chance of selecting one centroid from each cluster is small

- Chance is relatively small when  $K$  is large
- If clusters are the same size,  $n$ , then the probability is

$$P = \frac{\text{ways to select one centroid from each cluster}}{\text{ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

- Example: if  $K = 10$ , then probability  $= 10!/10^{10} = 0.00036$
- Sometimes the initial centroids will readjust themselves in “right” way, and sometimes they don’t

# Solutions to Initial Centroids Problem

- Multiple runs  
Helps, but probability is not on your side

# Solutions to Initial Centroids Problem

- Multiple runs  
Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids

# Solutions to Initial Centroids Problem

- Multiple runs  
Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than  $K$  initial centroids and then select among these initial centroids



# Solutions to Initial Centroids Problem

- Multiple runs  
Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than  $K$  initial centroids and then select among these initial centroids
- Bisecting K-means
  - 1 Pick a cluster to split.
  - 2 Find 2 sub-clusters using the basic k-Means algorithm (Bisecting step)
  - 3 Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity.
  - 4 Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

Not as susceptible to initialization issues

# 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

$$SSE = \sum_{i=1}^K \sum_{x \in c_i} d^2(m_i, x)$$

- $x$  is a data point in cluster  $c_i$  and  $m_i$  is the representative point (center/mean) for cluster  $c_i$ .
- Given two clusters, we can choose the one with the smaller error
- One easy way to reduce SSE is to increase  $K$  (# of clusters)
  - A good clustering with smaller  $K$  can have a lower SSE than a poor clustering with higher  $K$

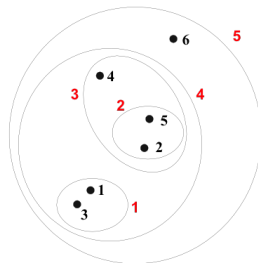
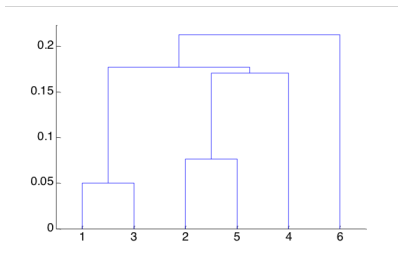
# Limitations of $K$ -means

- $K$ -means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-spherical shapes
- $K$ -means has problems when the data contains outliers
- Number of clusters ( $K$ ) is difficult to determine

# Hierarchical Clustering

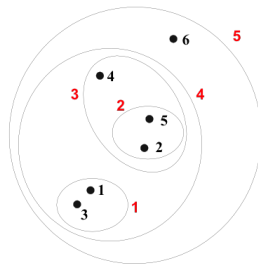
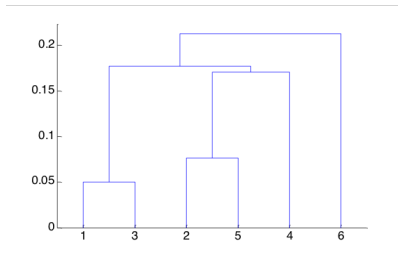
# Hierarchical Clustering

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



# Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by “cutting” the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - E.g., in biological sciences the animal kingdom



# Hierarchical Clustering

- 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

# 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 (or  $k$ ) remains

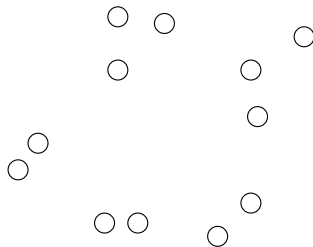


# Agglomerative Clustering Algorithm

- 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 (or  $k$ ) 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



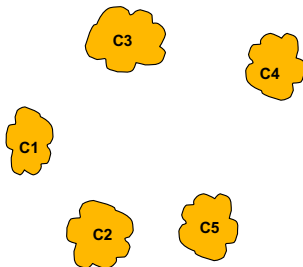
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

**Proximity Matrix**



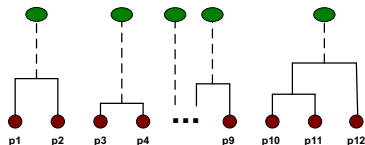
# Intermediate Situation

- After some merging steps, we have some clusters



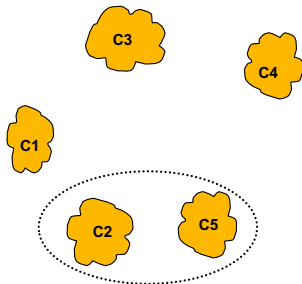
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



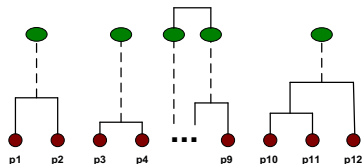
# Intermediate Situation

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



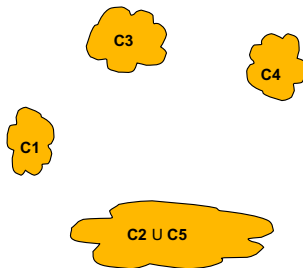
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



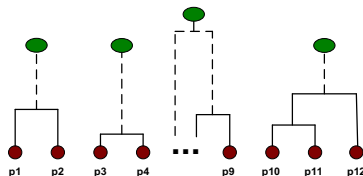
# After Merging

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



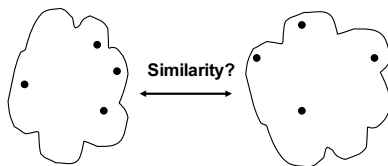
		C2 U C5			
		C1	C5	C3	C4
C1			?		
C2 U C5		?	?	?	?
C3			?		
C4			?		

**Proximity Matrix**



# How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids

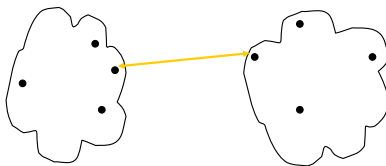


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

**Proximity Matrix**

# How to Define Inter-Cluster Similarity

- **MIN**
- **MAX**
- **Group Average**
- **Distance Between Centroids**

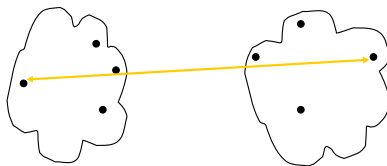


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

**Proximity Matrix**

# How to Define Inter-Cluster Similarity

- MIN
- **MAX**
- Group Average
- Distance Between Centroids



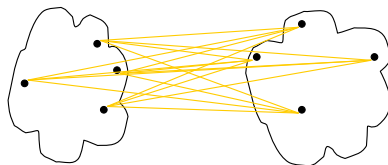
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



# How to Define Inter-Cluster Similarity

- MIN
- MAX
- **Group Average**
- Distance Between Centroids

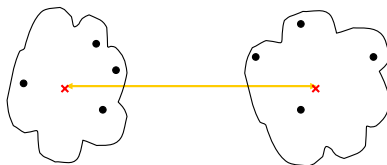


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

**Proximity Matrix**

# How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- **Distance Between Centroids**



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

**Proximity Matrix**

# Cluster Similarity: MIN (Single Link)

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters

$$D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$$

- 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

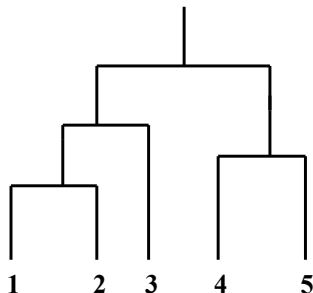
# Cluster Similarity: MIN (Single Link)

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters

$$D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$$

- 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



Note: Clusters formed via single linkage clustering may be forced together due to single elements being close to each other, even though many of the elements in each cluster may be very distant to each other.

# Cluster Similarity: MAX (Complete Link)

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

$$D(X, Y) = \max_{x \in X, y \in Y} d(x, y)$$

- 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

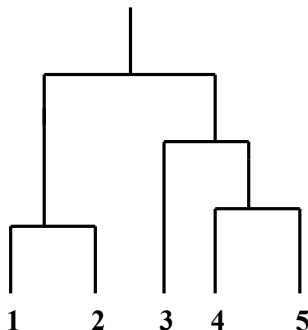
# Cluster Similarity: MAX (Complete Link)

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

$$D(X, Y) = \max_{x \in X, y \in Y} d(x, y)$$

- 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



# Cluster Similarity: Group Average

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

$$D(X, Y) = \sum_{x \in X, y \in Y} d(x, y) / (|X| \cdot |Y|)$$

- 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

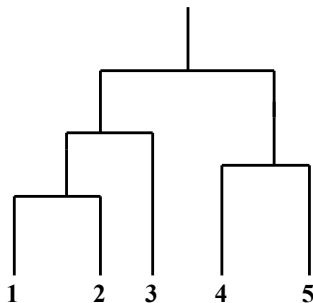
# Cluster Similarity: Group Average

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

$$D(X, Y) = \sum_{x \in X, y \in Y} d(x, y) / (|X| \cdot |Y|)$$

- 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 (Average Link)

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

# Hierarchical Clustering: Time & 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 & 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 (MIN)
  - Difficulty handling different sized clusters and non-convex shapes (Group average, MAX)
  - Breaking large clusters (MAX)

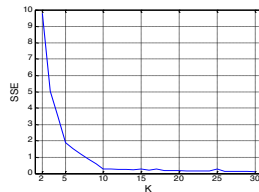
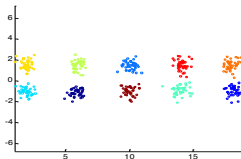
## Evaluating Clusters

# Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following 3 types
  - **External Index:** Used to measure the extent to which cluster labels match externally supplied class labels (e.g., entropy)
  - **Internal Index:** Used to measure the goodness of a clustering structure without respect to external information (e.g., 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

# Internal Measures: SSE

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



# Internal Measures: Cohesion & Separation

## ● Cluster Cohesion

- Measures:
  - How closely related objects are in a cluster (e.g., SSE)
  - Variability of observations within the cluster
- Smaller cohesion score (WCSS) = more compact cluster

## ● Cluster Separation

- Measures:
  - How distinct or well-separated a cluster is from other clusters
  - Variation between clusters
- Smaller separation score (BCSS) = clusters are closer together

# Internal Measures: Cohesion & Separation

- **Cohesion** is measured by the **within** cluster sum of squares

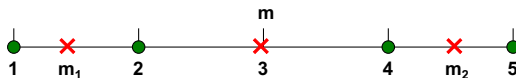
$$WCSS = \sum_i \sum_{x \in c_i} (x - m_i)^2$$

- **Separation** is measured by the **between** cluster sum of squares

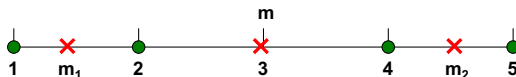
$$BCSS = \sum_i |c_i| (m - m_i)^2$$



# Internal Measures: Cohesion and Separation



# Internal Measures: Cohesion and Separation



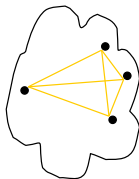
$K = 2$  clusters

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

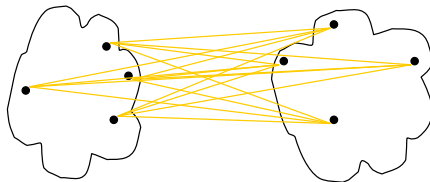
$$BCSS = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9$$

# Internal Measures: Cohesion and Separation

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



cohesion



separation