Data Mining

Cluster Analysis
Advanced Concepts and Algorithms

Slides by Tan, Steinbach, Kumar adapted by Pimprapai Thainiam

Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - Graph-Based Clustering
- Cluster Validity (Cluster Evaluation)
  - Clustering Tendency
  - Determining the Correct Number of Clusters
  - Unsupervised Cluster Evaluation
  - Supervised Cluster Evaluation

Fuzzy Clustering

- **Fuzzy clustering** is used when the objects in a data set cannot be partitioned into well-separated clusters, and there will be a certain arbitrariness in assigning an object to a particular cluster.
- It might be more appropriate to assign a weight (w_{ij}) to each object, $\mathbf{x}_i$, and each cluster, C_j, that indicates the degree to which the object belongs to the cluster.
- The membership weights (degrees), w_{ij}, have been assigned values between 0 and 1.
- Conditions on the clusters in order to ensure that the clusters form what is called **fuzzy pseudo-partition**.

  1) All the weights for a given point, $\mathbf{x}_i$, add up to 1

$$\sum_{j=1}^{k} w_{ij} = 1$$

  2) Each cluster, C_j, contains at least one point with non-zero weight, but does not contain all of the points with a weight of 1.

$$0 < \sum_{i=1}^{m} w_{ij} < m$$

ex obj clus 1 clus 2
1 0.8 0.2 = 1
2 0.6 0.4 = 1

ex obj clus 1
1 0.8
⋮ ⋮
m 0.2
< m

Fuzzy C-means Algorithm

- **Fuzzy c-means** is considered as a fuzzy version of K-means clustering which is attempting to minimize a fuzzy version of the sum of the squared error (SSE).

Basic fuzzy c-means algorithm

1: Select an initial fuzzy pseudo-partition, i.e., assign values to all the w_{ij}
2: **Repeat**
3: Compute the centroid of each cluster using the fuzzy pseudo-partition
4: Recompute the fuzzy pseudo-partition, i.e., the w_{ij}
5: **Unitil** The centroids don't change.
(Alternative stopping conditions are " if the change in the error is below a specified threshold" or "if the absolute change in any w_{ij} is below given threshold.")

Fuzzy C-means Algorithm

- The sum of squared errors for fuzzy clustering:

$$SSE(C_1, C_2, \dots, C_k) = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{p} \, dist(\mathbf{x}_i, \mathbf{c}_j)^2$$

- The **corresponding centroid**, $\mathbf{c}_j$, for a cluster, C_j, ➡

$$\mathbf{c}_j = \frac{\sum_{i=1}^{m} w_{ij}^{p} \, \mathbf{x}_i}{\sum_{i=1}^{m} w_{ij}^{p}}$$

- The **weights** can be derived by minimizing the SSE ➡

$$w_{ij} = \frac{\left(\frac{1}{dist(\mathbf{x}_i, \mathbf{c}_j)^2}\right)^{\frac{1}{p-1}}}{\sum_{q=1}^{k} \left(\frac{1}{dist(\mathbf{x}_i, \mathbf{c}_j)^2}\right)^{\frac{1}{p-1}}}$$

Normalized the weight

where

$\mathbf{c}_j$ denotes the centroid of the j^{th} cluster

p denotes the exponent that determines the influence of the weights (between 1 and ∞)

Fuzzy C-means Algorithm

- The weight w_{ij}, should be relatively high if $\mathbf{x}_i$ is close to centroid $\mathbf{c}_j$ (if $dist(\mathbf{x}_i, \mathbf{c}_j)$ is low) and relatively low if $\mathbf{x}_i$ is far from centroid $\mathbf{c}_j$ (if $dist(\mathbf{x}_i, \mathbf{c}_j)$ is high).
- Choosing the value of p :

  $p = 2$: This simplifies the weight update formula.

  $p \to 1$: This makes fuzzy c-means to behave like traditional K-means.

  $p \to \infty$: This makes all the cluster centroids approach the global centroid of all the data points.

Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - **Grid-Based Clustering**
  - Graph-Based Clustering
- Cluster Validity (Cluster Evaluation)
  - Clustering Tendency
  - Determining the Correct Number of Clusters
  - Unsupervised Cluster Evaluation
  - Supervised Cluster Evaluation

Grid-Based Clustering

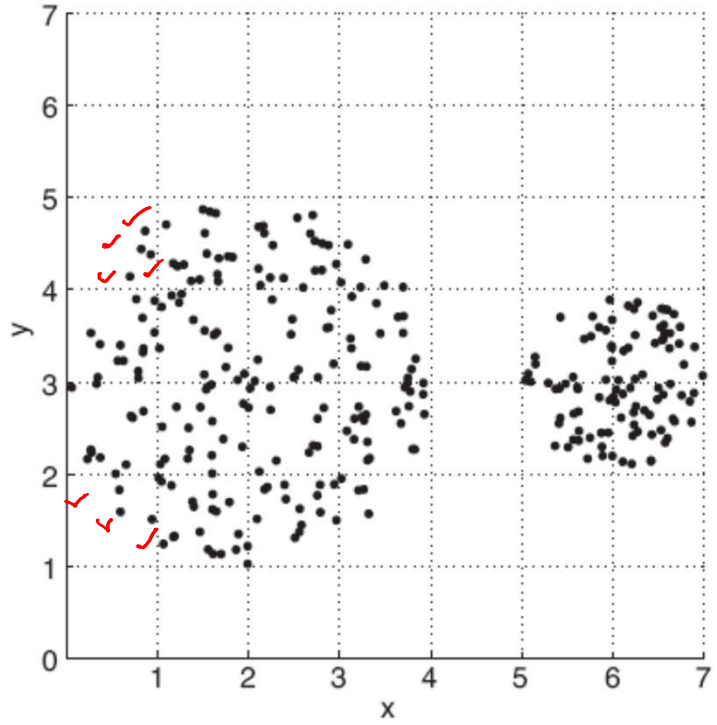
- **Grid-based clustering** breaks the data space into grid cells and then forms clusters from cells that are sufficiently dense.

Basic grid-based clustering algorithm
1: Define a set of grid cells.
2: Assign objects to the appropriate cells and compute the density of each cell.
3: Eliminate cells having a density below a specified threshold, τ.
4: Form clusters from contiguous (adjacent) groups of dense cells.

- One common approach to define grid cells is to break the values of an attribute into equal width intervals, then the resulting grid cells will all have the same volume and the density of a cell is defined as the number of points in the cell.
- The **density of a grid cell** is defined as the number of points per amount of space.

Grid-Based Clustering

- **Example:** Two sets containing 200 points and 100 points. Since the cells have equal volume (area), we can consider these values to be the densities of the cells.



0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	**17**	**18**	**6**	0	0	0
14	**14**	**13**	**13**	0	**18**	**27**
11	**18**	**10**	**21**	0	**24**	**31**
3	**20**	**14**	**4**	0	0	0
0	0	0	0	0	0	0

Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - **Graph-Based Clustering**
- Cluster Validity (Cluster Evaluation)
  - Clustering Tendency
  - Determining the Correct Number of Clusters
  - Unsupervised Cluster Evaluation
  - Supervised Cluster Evaluation

Graph-Based Clustering

- **Graph-based clustering** normally uses the proximity graph to create cluster; the **proximity graph** can be created as follows:

  Step 1: Compute the proximity matrix (either similarity or dissimilarity)

  Step 2: Consider each point as a node in a graph

  Step 3: Each edge between two nodes has a weight which is the proximity between the two points

  Step 4: Connect each node to all other nodes by adding links → CLIQUE

- One of the key approaches for graph-based clustering is to sparsify the proximity graph to keep only the connections of an object with its nearest neighbors.
- This **sparsification** is useful for handling noise and outliers.

Graph-Based Clustering

- **Sparsification** uses the following property:

> Although every object has some level of similarity to every other object, for most data sets, object are highly similar to a small number of objects and weakly similar to most other objects.

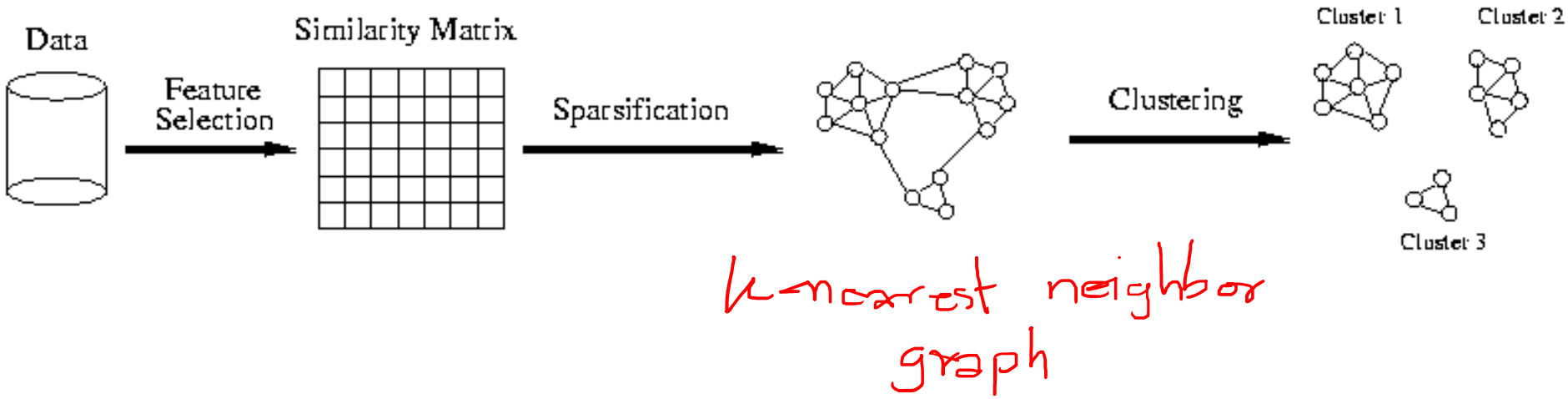
- The proximity graph (matrix) can be sparsified by setting many of these low-similarity (high-dissimilarity) values to 0 before beginning the actual clustering process.
- Setting low-similarity (high-dissimilarity) values to 0 (∞) can be performed by

  1) Breaking all links that have a similarity (dissimilarity) below (above) a **specified threshold**

  2) Keeping only links to the k nearest neighbors of point; this approach creates what is called the ***k*-nearest neighbor graph**

Graph-Based Clustering

- Sparsification of the proximity graph is an initial step before the use of actual clustering algorithms.
- The process of graph-based clustering using sparsification is summarized in the following figure.



Minimum Spanning Tree (MST) Clustering

- **Minimum spanning tree (MST) clustering** starts with the minimum spanning tree of the proximity graph and can be viewed as an application of sparsification for finding clusters.
- A minimum spanning tree of a graph is a subgraph that
  - Has no cycles
  - Contains all the nodes of the graph
  - Has the minimum total edge weight of all possible spanning trees
- The most famous algorithms to find MST are **Prim's algorithm** and **Kruskal's algorithm**.

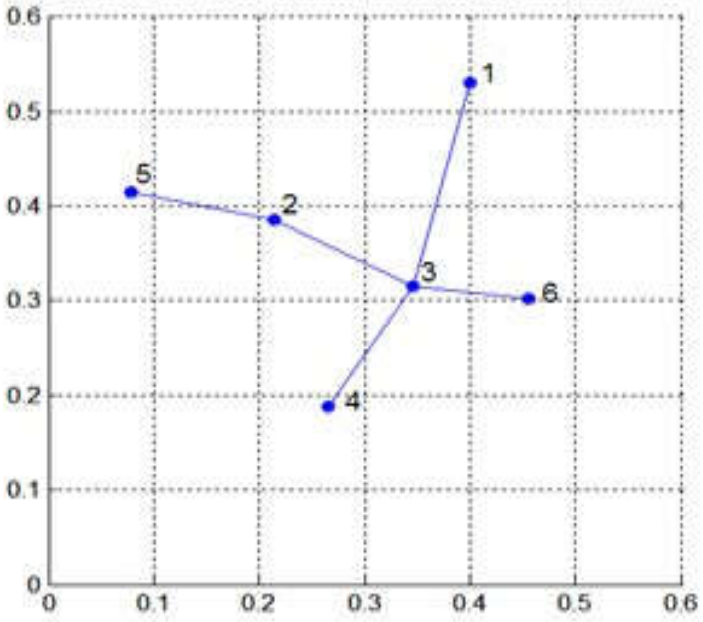
MST divisive hierarchical clustering algorithm

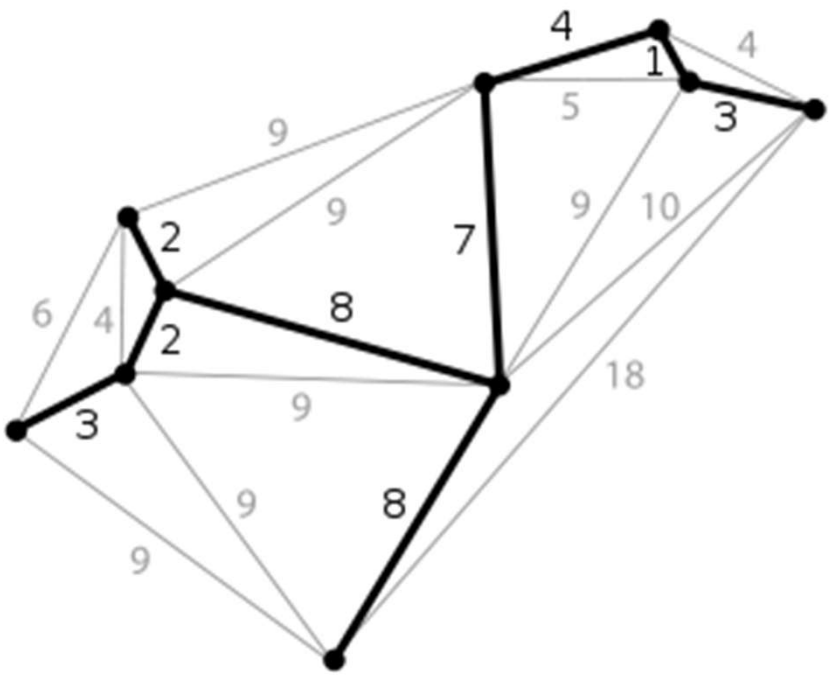
1: Compute a minimum spanning tree for the dissimilarity graph
2: **Repeat**
3: Create a new cluster by breaking the link corresponding to the largest dissimilarity
4: **Until** Only singleton clusters remain

Note: Data points that are not put into groups are called *singletons*.

Minimum Spanning Tree (MST) Clustering

Minimum spanning tree in 2-dimensional space



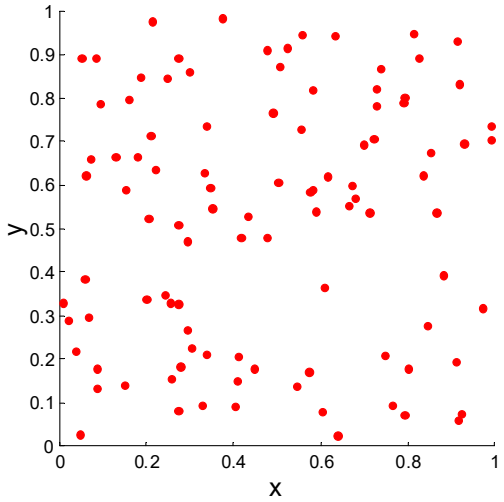


Topics

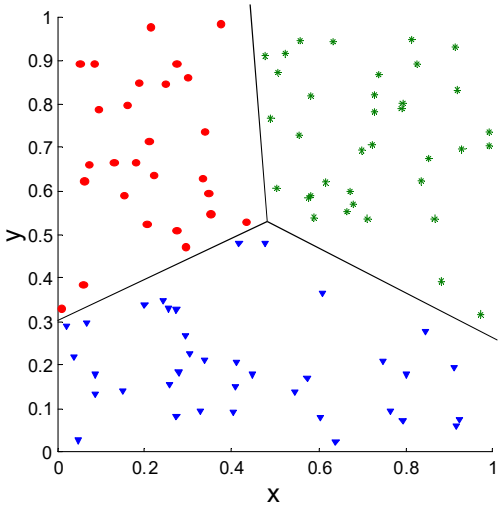
- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - Graph-Based Clustering
- **Cluster Validity (Cluster Evaluation)**
  - Clustering Tendency
  - Determining the Correct Number of Clusters
  - Unsupervised Cluster Evaluation
  - Supervised Cluster Evaluation

Cluster Validation

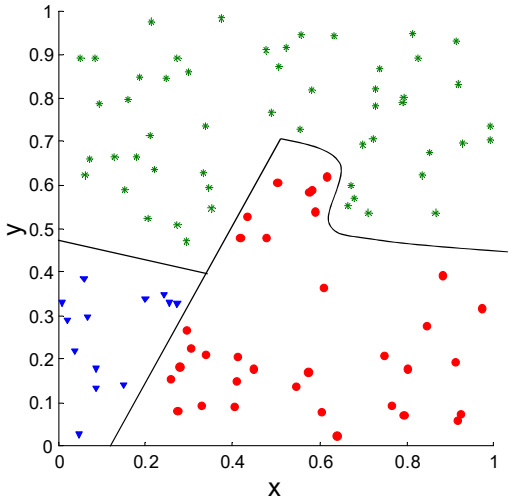
- **Cluster validation** should be a part of any cluster analysis because almost every clustering algorithm will find clusters in a data set, even if that data set has no natural cluster structure.



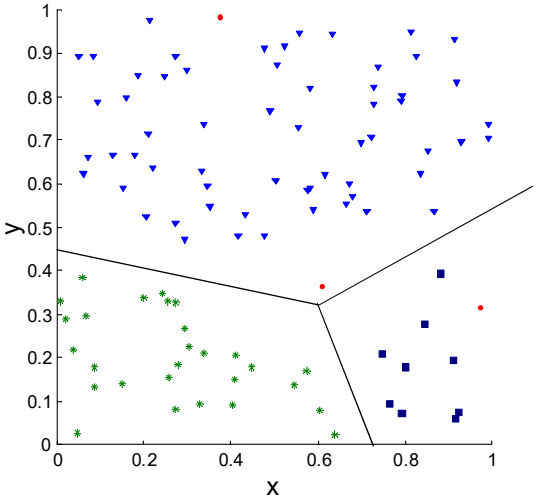
Random Points



K-means



Complete Link



DBSCAN

Different Aspects of Cluster Validation

Important Issues for 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. Determining the **correct number of clusters**.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information. $\Rightarrow$ **Internal Validation**
4. Comparing the results of a cluster analysis to externally known cluster labels (ground truth). $\Rightarrow$ **External Validation**
5. Comparing two sets of clusters to determine which is better.

Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - Graph-Based Clustering
- Cluster Validity (Cluster Evaluation)
  - **Clustering Tendency**
  - Determining the Correct Number of Clusters
  - Unsupervised Cluster Evaluation
  - Supervised Cluster Evaluation

Clustering Tendency

- The obvious approach to determine if a data set has clusters is to try to cluster it, and then evaluate the resulting clusters quality. This can be done by:

  Step 1: Cluster a given data set using multiple clustering algorithms

  Step 2: Evaluate the quality of the resulting clusters

  Step 3: Interpret the results:

  - If the clusters are uniformly poor, then this indicates that there are no clusters in the data.
  - If at least some of the clusters are good, this indicates that a data set has clusters.

- The other approach is to use statistical tests for spatial randomness; this method evaluates whether a data set has cluster without clustering, for example, Hopkins statistic.

Clustering Tendency

- **Hopkins Statistic (H)** steps:

  Step 1: Generate p points that are randomly distributed across the data space

  Step 2: Sample p actual data points

  Step 3: Find the distance to the nearest neighbor in the original data set for both sets obtained in step 1 and 2, then set them to u_i and w_i, respectively

  Step 4: Calculate Hopkins statistic

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

- Interpretation of H values:

  $H \approx 0.5 \Rightarrow$ The randomly generated points and the sample of data points have roughly the same nearest neighbor distances

  $H \approx 0 \Rightarrow$ The data is highly clustered.

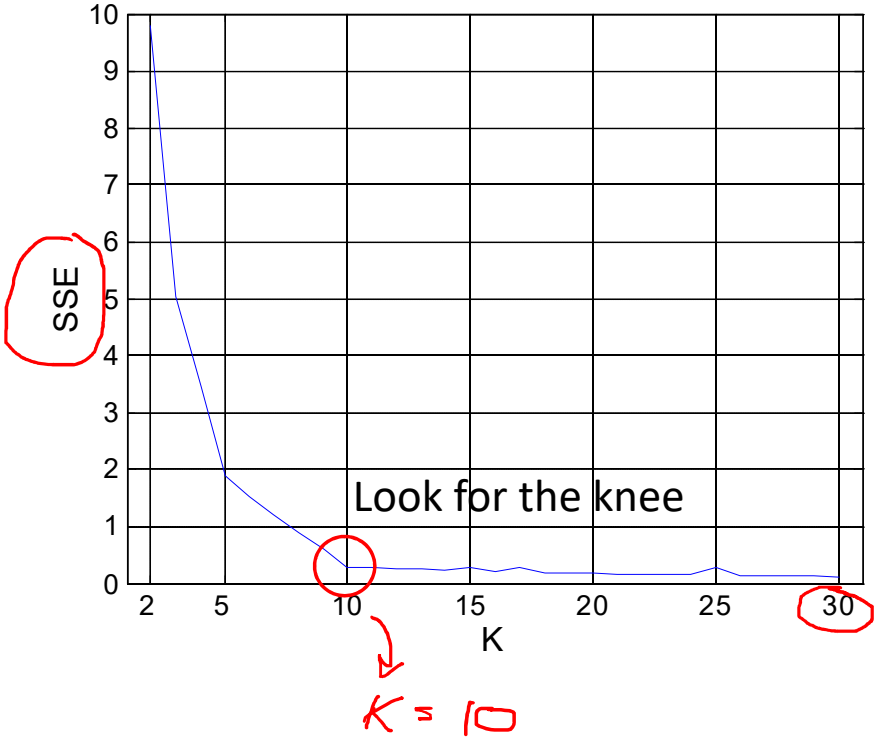
  $H \approx 1 \Rightarrow$ The data is regularly distributed in the data space.

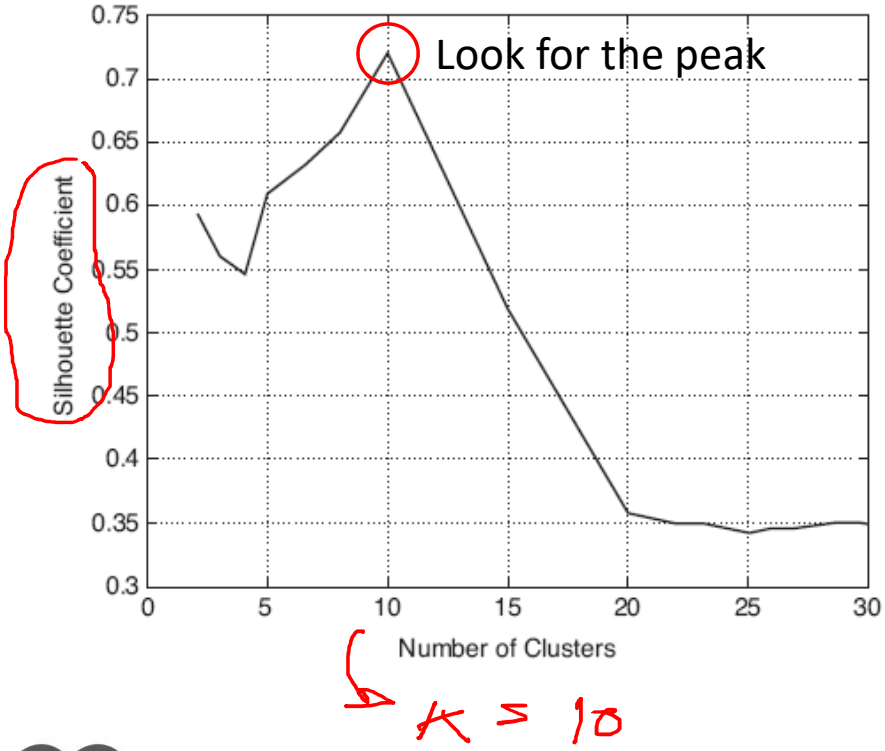
Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - Graph-Based Clustering
- Cluster Validity (Cluster Evaluation)
  - Clustering Tendency
  - **Determining the Correct Number of Clusters**
  - Unsupervised Cluster Evaluation
  - Supervised Cluster Evaluation

Determining the Correct Number of Clusters

- We can find the natural number of clusters in a data set by looking for the number of clusters at which there is a knee, peak, or dip in the plot of the evaluation measure when it is plotted against the number of clusters.





Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - Graph-Based Clustering
- Cluster Validity (Cluster Evaluation)
  - Clustering Tendency
  - Determining the Correct Number of Clusters
  - **Unsupervised Cluster Evaluation**
  - Supervised Cluster Evaluation

Measure of Cluster Validation

Measures can be classified into three types.

1. **Unsupervised (Internal indices):** It measures the goodness of a clustering structure without respect to external information. This type of measure can be divided into 2 classes.
   1) **Measures of cluster cohesion:** (compactness, tightness) They determine how closely related the objects in a cluster are.
   2) **Measures of cluster separation:** (isolation) They determine how distinct or well-separated a cluster is from other clusters.
2. **Supervised (External Indices):** It measures the extent to which the clustering structure discovered by a clustering algorithm matches some external structure, for example, class labels.
3. **Relative:** It compares different clusterings or clusters used for the purpose of comparison.

Unsupervised Cluster Evaluation

There are three main types for unsupervised cluster evaluation:

1. Unsupervised cluster evaluation using cohesion and separation
2. Unsupervised cluster evaluation using proximity matrix
3. Unsupervised cluster evaluation of hierarchical clustering

Unsupervised Cluster Evaluation Using Cohesion and Separation

- Overall cluster validity for a set of K clusters can be expressed as a weighted sum of the validity of individual clusters.

$$\Rightarrow \quad overall\ validity = \sum_{i=1}^{K} w_i(validity(C_i))$$

where $validity$ function can be cohesion, separation, or some combination of these quantities.

C_i is cluster i

K is number of cluster

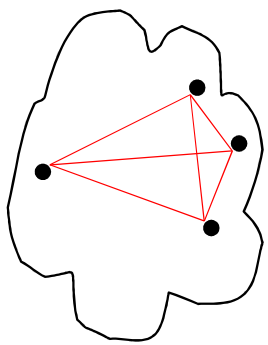
w_i is weight of cluster i which vary depending on the cluster validity measure. It could be 1 or the size of the cluster

- If $validity$ function is cohesion within a cluster, the higher values are better.
- If $validity$ function is separation within a cluster, the lower values are better.

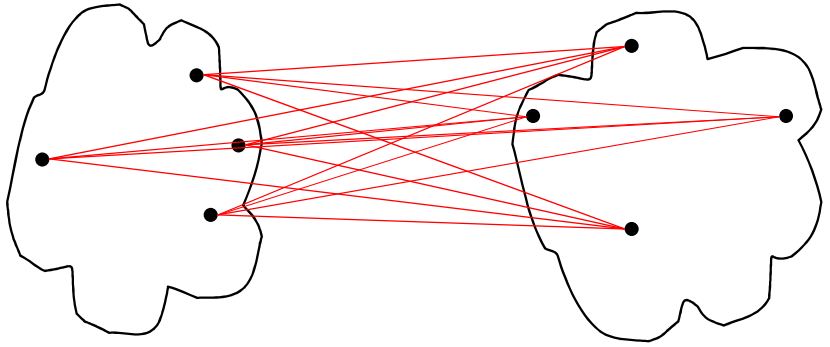
Unsupervised Cluster Evaluation Using Cohesion and Separation

Graph-Based View of Cohesion and Separation

- The **cohesion of a cluster** can be defined as the sum of the weights of the links in the proximity graph that connect points within the cluster.
- The **separation between two clusters** can be measured by the sum of the weights of the links from points in one cluster to points in the other cluster.



cohesion



separation

Unsupervised Cluster Evaluation Using Cohesion and Separation

$$cohesion(C_i) = \sum_{\substack{\mathbf{x} \in C_i \\ \mathbf{y} \in C_i}} proximity(\mathbf{x}, \mathbf{y})$$

$$separation(C_i, C_j) = \sum_{\substack{\mathbf{x} \in C_i \\ \mathbf{y} \in C_j}} proximity(\mathbf{x}, \mathbf{y})$$

where $proximity$ is a function which can be a similarity, a dissimilarity, or a simple function of these quantities

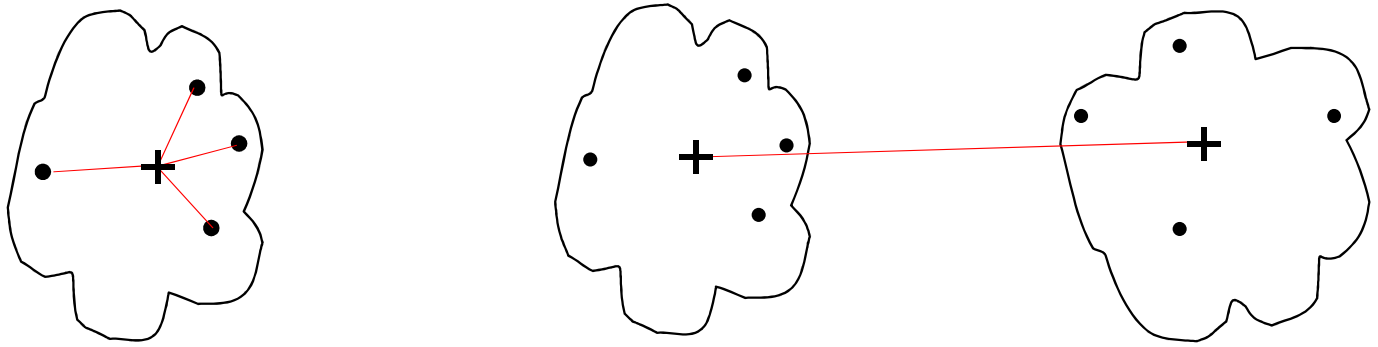
C_i , C_j are clusters

$\mathbf{x}$, $\mathbf{y}$ are points

Unsupervised Cluster Evaluation Using Cohesion and Separation

Prototype-Based View of Cohesion and Separation

- The **cohesion of a cluster** can be defined as the sum of the proximities with respect to the prototype (centroid or medoid) of the cluster.
- The **separation between two clusters** can be measured by the proximity of the two cluster prototypes.



cohesion separation

Unsupervised Cluster Evaluation Using Cohesion and Separation

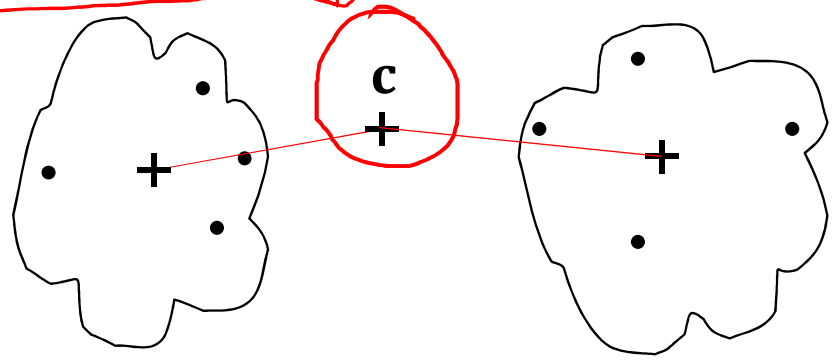
$$cohesion(C_i) = \sum_{\mathbf{x} \in C_i} proximity(\mathbf{x}, \mathbf{c_i})$$

$$separation(C_i, C_j) = proximity(\mathbf{c_i}, \mathbf{c_j})$$

$$separation(C_i) = proximity(\mathbf{c_i}, \mathbf{c})$$

where $\mathbf{c_i}$, $\mathbf{c_j}$ are the prototypes (centroids) of clusters C_i, C_j

$\mathbf{c}$ is the overall prototype (centroid)



Unsupervised Cluster Evaluation Using Cohesion and Separation

Overall Measures of Cohesion and Separation

- Cluster validities can be combined into an overall measure of cluster validity by using a weighted sum, but the important issue is that we need to decide what weights to use.

Name	Cluster Measure (Validity Function)	Cluster Weight	Type
$\mathcal{L}_1$	$\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_i}} proximity(\mathbf{x},\mathbf{y})$ cohesion ↑	$\frac{1}{m_i}$ ← No of points in a cluster	Graph-based cohesion
$\mathcal{L}_2$	$\sum_{\mathbf{x}\in C_i} proximity(\mathbf{x},\mathbf{c_i})$	1	Prototype-based cohesion
$\mathcal{E}_1$	$proximity(\mathbf{c_i},\mathbf{c})$	m_i	Prototype-based separation
$\mathcal{G}_1$	$\sum_{j=1}^{k}\sum_{\substack{\mathbf{x}\in C_i \\ j\neq i \ \mathbf{y}\in C_i}} proximity(\mathbf{x},\mathbf{y})$	$\frac{1}{\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_i}} proximity(\mathbf{x},\mathbf{y})}$	Graph-based separation and cohesion

Note: m_i is size of cluster C_i

Unsupervised Cluster Evaluation Using Cohesion and Separation

The Silhouette Coefficient

- The popular method of silhouette coefficients combines both cohesion and separation.
- Silhouette coefficient for an individual point can be calculated as the following steps:

  Step 1: Calculate its average distance to all other points in its cluster, then set to a_i

  Step 2: Calculate its average distance to all other points in a given cluster

  Step 3: Repeat step 2 for all other clusters

  Step 4: Find the minimum value found in step 2 and 3, then set to b_i

  Step 5: Calculate silhouette coefficient

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

Unsupervised Cluster Evaluation Using Cohesion and Separation

- **Silhouette coefficient** can vary between -1 (undesirable) and 1 (desirable).
- We want the silhouette coefficient to be as close to 1 as possible (if $a_i = 0$, then $s_i = 1$), thus

$$a_i < b_i$$

$$a_i \to 0$$

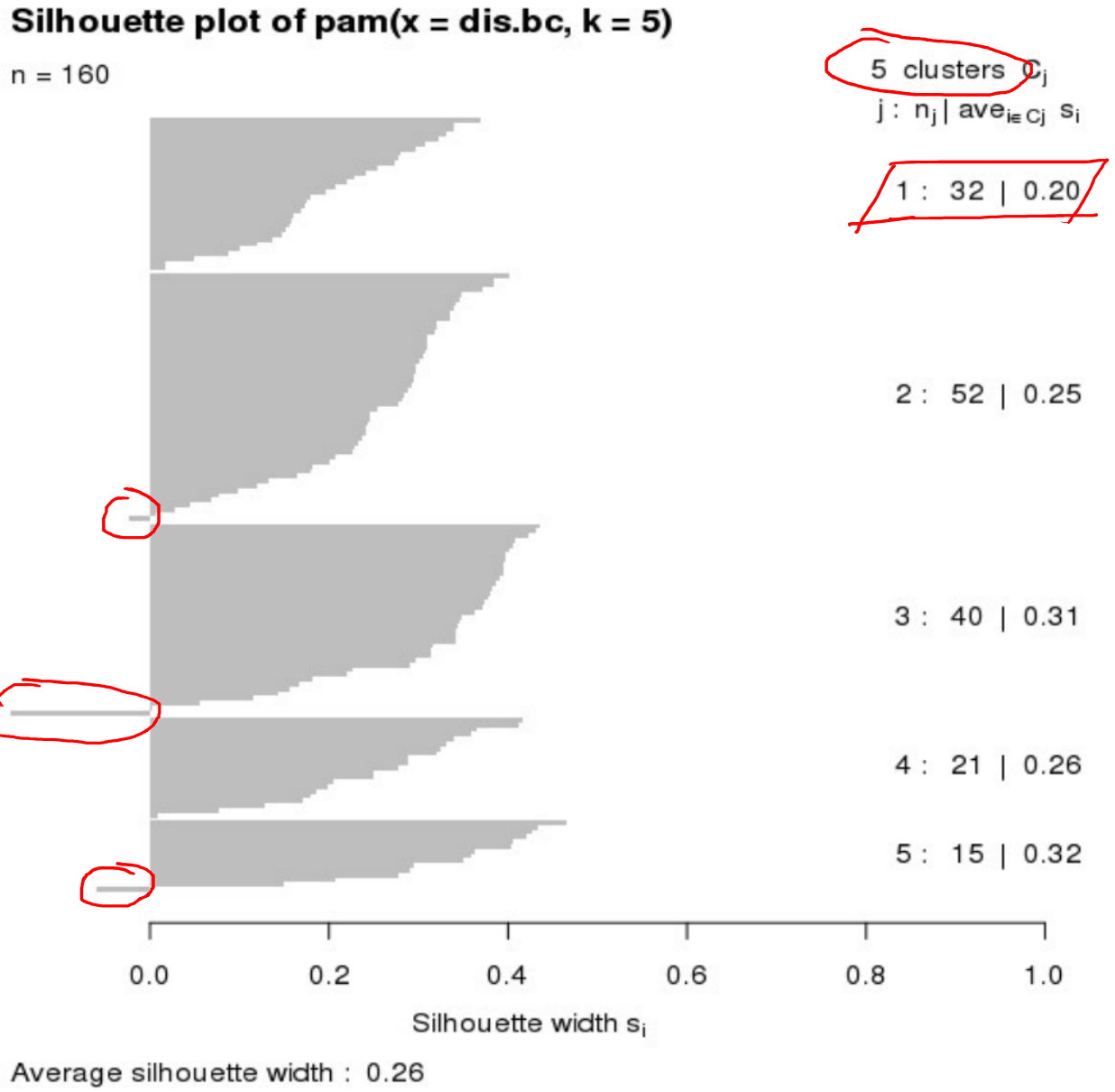
- The average silhouette coefficient of a cluster is the average of the silhouette coefficients of points belong to the cluster.
- An **overall measure of the goodness of a clustering** can be obtained by computing the average silhouette coefficient of all points.

$$Average\ Silhouette = \frac{\sum_{i=1}^{n} s_i}{n}$$

where s_i denotes silhouette coefficient of any point i

n denotes number of points in the data set

The average distance of this point to all other points in its cluster is greater than the average distance of this point to all other points in the closest cluster



Unsupervised Cluster Evaluation Using Proximity Matrix

There are two schemes for unsupervised cluster evaluation using proximity matrix:

1. Measuring Cluster Validity via Correlation

- The goodness of the clustering can be measured by looking at the **correlation** between the actual similarity matrix and an ideal version of the similarity matrix based on the cluster labels.
- An ideal similarity matrix is the one whose points have
  - A similarity of 1 to all points in the cluster
  - A similarity of 0 to all points in other clusters
- High correlation $\Rightarrow$ The points that belong to the same cluster are close to each other.
- Low correlation $\Rightarrow$ The points that belong to the same cluster are spread out from each other.
- Since these matrices are symmetric, the correlation can be calculated only among the $n(n-1)/2$ entries below or above the diagonal of the matrices.

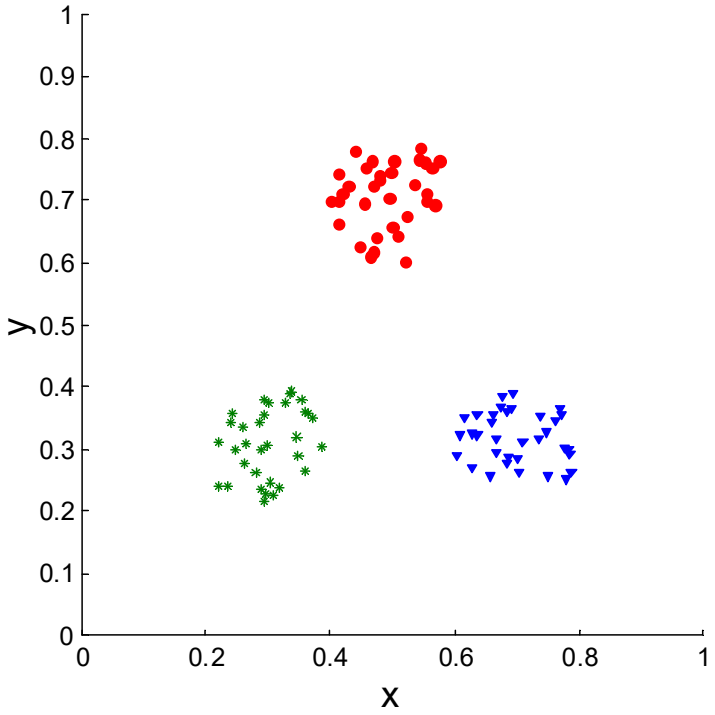
Unsupervised Cluster Evaluation Using Proximity Matrix

An ideal similarity matrix after rearrange the rows and columns so that all objects belonging to the same class are together (block diagonal structure):

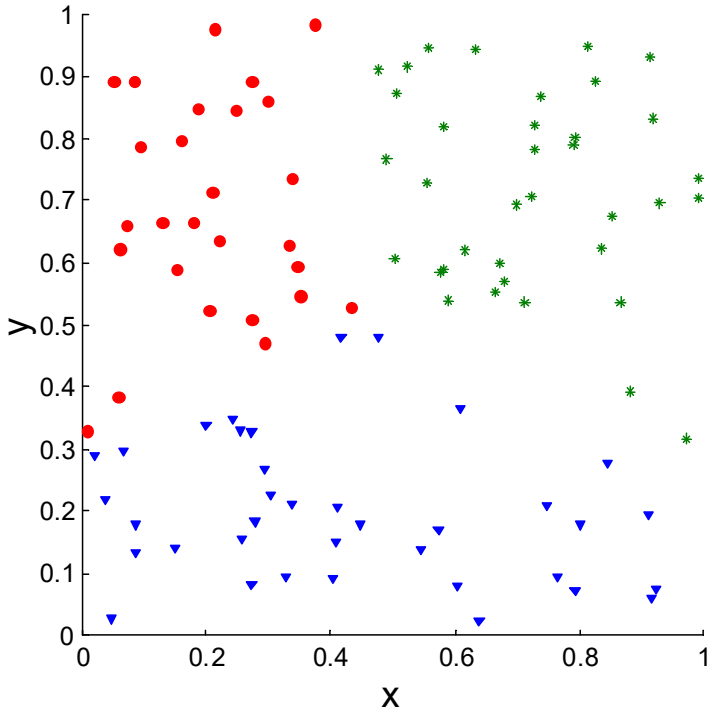
Object number	7	5	3	1	9	2	4	10	6	8
7	1	1	1	0	0	0	0	0	0	0
5	1	1	1	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0
9	0	0	0	1	1	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1
4	0	0	0	0	0	1	1	1	1	1
10	0	0	0	0	0	1	1	1	1	1
6	0	0	0	0	0	1	1	1	1	1
8	0	0	0	0	0	1	1	1	1	1

Unsupervised Cluster Evaluation Using Proximity Matrix

Examples: Correlations of a well-separated data set and a random data set; the results obtained from K-means clustering



Correlation = 0.9235



Correlation = 0.5810

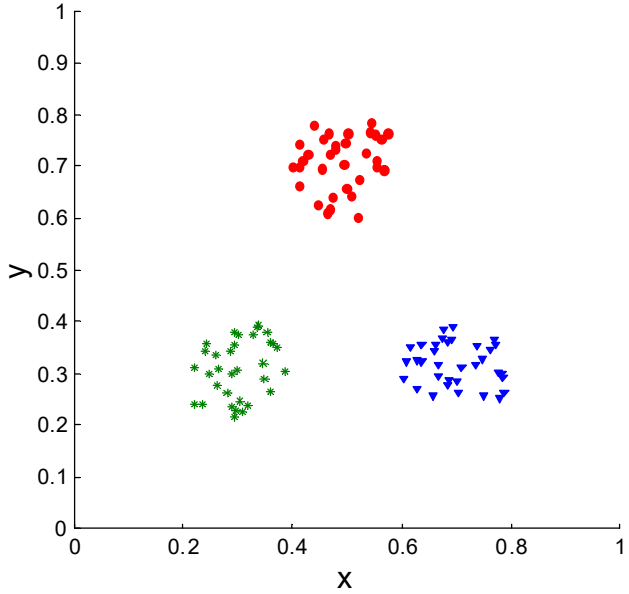
Unsupervised Cluster Evaluation Using Proximity Matrix

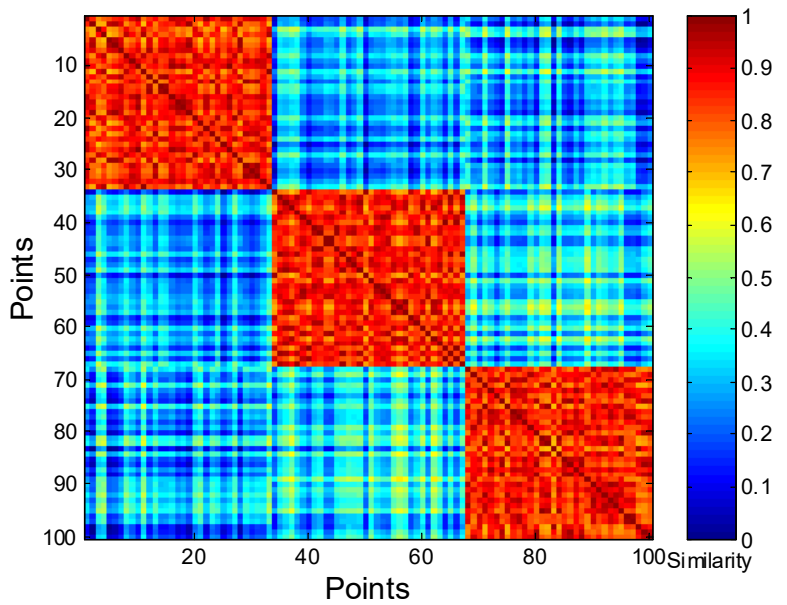
2. Judging a Clustering Visually by Its Similarity Matrix

- This approach is a qualitative approach. It can be done by

  Step 1: Order the similarity matrix with respect to cluster labels

  Step 2: Plot the ordered similarity matrix → all objects in the same cluster must be close to each other.

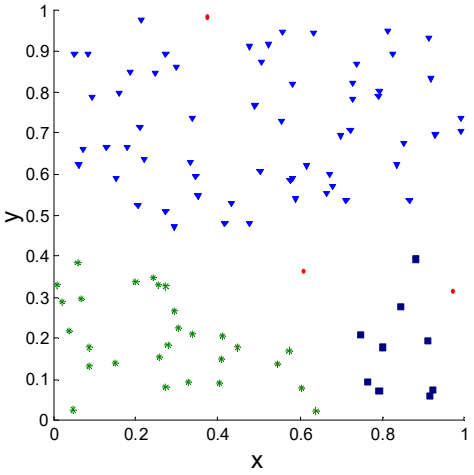




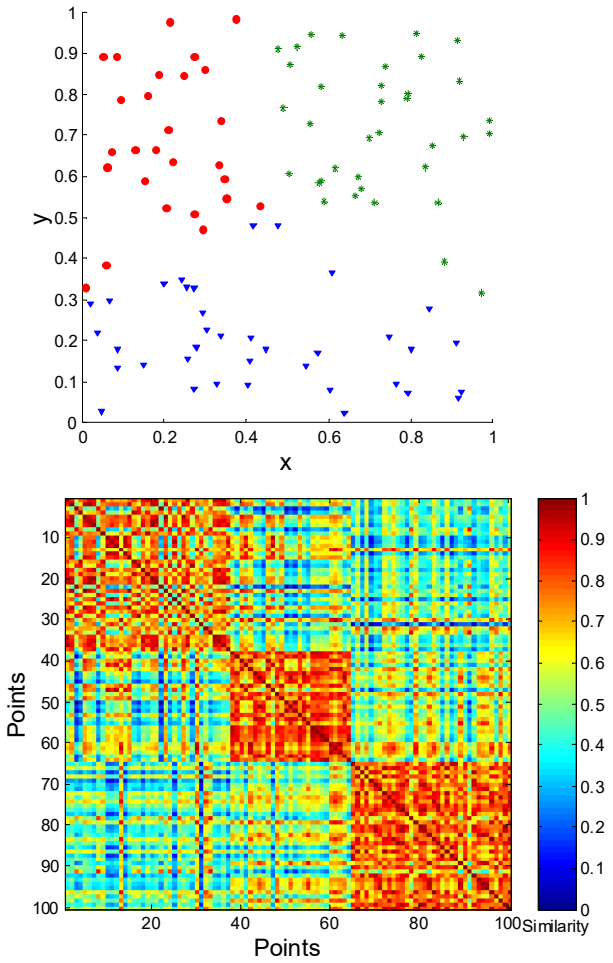
A very strong block-diagonal pattern

Unsupervised Cluster Evaluation Using Proximity Matrix

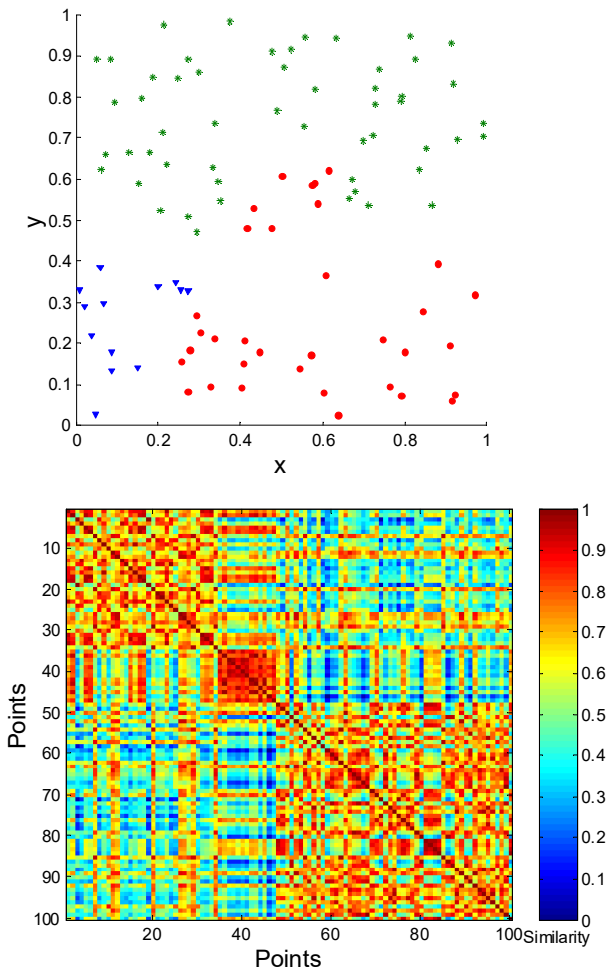
DBSCAN



K-means



Complete Link



Unsupervised Cluster Evaluation of Hierarchical Clustering

- **Cophenetic correlation** is the well-known measure to evaluate cluster validation for hierarchical clustering.
- The **cophenetic distance between two objects** is the proximity at which agglomerative hierarchical clustering technique puts these two objects in the same cluster for the first time.
- Each entry in **cophenetic distance matrix** can be created by taking the smallest distance between the two points from two clusters.
- The **CoPhenetic Correlation Coefficient (CPCC)** is the correlation between the entries of **cophenetic distance matrix** and the entries of **original dissimilarity matrix**.
- The most common use of this measure is to evaluate which type of hierarchical clustering (Single link, Complete link, Group average, Ward's) is best for a particular type of data.

Unsupervised Cluster Evaluation of Hierarchical Clustering

- **Example**: Cophenetic Distance

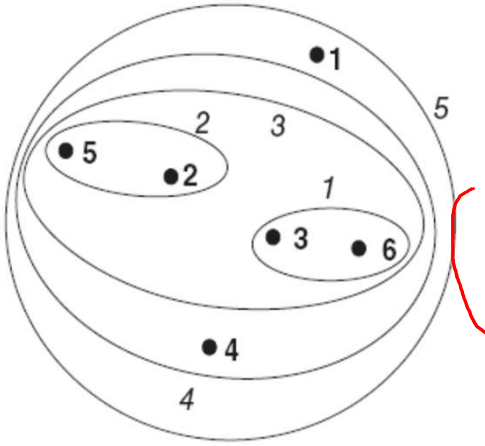
Euclidean distance matrix for 6 points

	P1	P2	P3	P4	P5	P6
P1	0	0.236	0.222	0.369	0.342	0.235
P2	0.236	0	0.148	0.204	0.139	0.254
P3	0.222	0.148	0	0.151	0.284	0.11
P4	0.369	0.204	0.151	0	0.293	0.222
P5	0.342	0.139	0.284	0.293	0	0.392
P6	0.235	0.254	0.11	0.222	0.392	0

Cophenetic distance matrix for 6 points

	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	0.11
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	0.11	0.151	0.148	0

Single link clustering



The followings are how to find cophenetic distance:

1) Points 3 and 6 are clustered together with the cophenetic distance of 0.11 → Cluster 1
2) Points 2 and 5 are clustered together with the cophenetic distance of 0.139 → Cluster 2
3) Clusters 1 and 2 are clustered together with the cophenetic distance of 0.148 → Cluster 3
4) Cluster 3 and point 4 are clustered together with the cophenetic distance of 0.151 → Cluster 4
5) Cluster 4 and point 1 are clustered together with the cophenetic distance of 0.222 → Cluster 5

→ min distance {(5,3), (5,6), (2,3), (2,6)} = 0.148

Topics

- Advanced Clustering Algorithms
  - Fuzzy Clustering
  - Grid-Based Clustering
  - Graph-Based Clustering
- Cluster Validity (Cluster Evaluation)
  - Clustering Tendency
  - Determining the Correct Number of Clusters
  - Unsupervised Cluster Evaluation
  - **Supervised Cluster Evaluation**

Supervised Measures of Cluster Validity

- This kind of measure is used when we have external information about data set which is usually in the form of class labels for the data objects.
- The usual procedure is to measure the degree of correspondence between the cluster labels and the class labels.
- The main objectives of this measure are
  1) To compare clustering techniques with the "ground truth"
  2) To evaluate which a manual classification process can be automatically produced by cluster analysis
- There are 2 types of approaches:
  1) **Classification-Oriented Measures:** The measures that are used in classification, such as, entropy, purity and the F-measure.
  2) **Similarity-Oriented Measures:** The measures that are related to the similarity measures for binary data, such as the Jaccard measure.

Supervised Measures of Cluster Validity

1. Classification-Oriented Measures of Cluster Validity

- There are a number of measures, such as, entropy, purity, precision, recall, and the F-measure.

Entropy

$$p_{ij} = \frac{m_{ij}}{m_i}$$

$$e_i = -\sum_{j=1}^{L} p_{ij} \log_2 p_{ij}$$

$$\boxed{e = \sum_{i=1}^{K} \frac{m_i}{m} e_i}$$

where m_i is the number of objects in cluster i
m_{ij} is the number of objects of class j in cluster i
m is the total number of data points
L is the number of classes
K is the number of clusters

$$\boxed{Purity = \sum_{i=1}^{K} \frac{m_i}{m} \max_j p_{ij}}$$

$$\boxed{Precision(i,j) = p_{ij}}$$

$$\boxed{Recall(i,j) = \frac{m_{ij}}{m_j}}$$

$$\boxed{F(i,j) = \frac{2\big(precision(i,j)\big) \times recall(i,j)}{\big(precision(i,j) + recall(i,j)\big)}}$$

Supervised Measures of Cluster Validity

2. Similarity-Oriented Measures of Cluster Validity

- The measures in this approach are based on the premise that any two objects that are in the same cluster should be in the same class and vice versa.
- This approach involves the comparison of two matrices:
  1) The **ideal cluster similarity matrix** which is defined with respect to cluster labels
  2) The **ideal class similarity matrix** which is defined with respect to class labels
- The measures that are most frequently used to evaluate cluster validity are
  1) Correlation
  2) Simple matching coefficient (aka Rand statistic)
  3) Jaccard coefficient

  Note that for measure 2) and 3), we can take only values lying above the diagonal.

original labels

resulting labels

Supervised Measures of Cluster Validity

$$Rand\ statistic = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

$$Jaccard\ coefficient = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

where f_{00} = Number of pairs of objects having a different class and a different cluster

f_{01} = Number of pairs of objects having a different class and the same cluster

f_{10} = Number of pairs of objects having the same class and a different cluster

f_{11} = Number of pairs of objects having the same class and the same cluster

Supervised Measures of Cluster Validity

- **Example:** The data set of 5 points having 2 classes and 2 clusters

$L_1 = \{p1,\ p2\}$
$C_1 = \{p1,\ p2,\ p3\}$

$L_2 = \{p3,\ p4,\ p5\}$ → original labels
$C_2 = \{p4,\ p5\}$ → resulting labels

Ideal class similarity matrix

	P1	P2	P3	P4	P5
P1	1	1	0	0	0
P2	1	1	0	0	0
P3	0	0	1	1	1
P4	0	0	1	1	1
P5	0	0	1	1	1

Ideal cluster similarity matrix

	P1	P2	P3	P4	P5
P1	1	1	1	0	0
P2	1	1	1	0	0
P3	1	1	1	0	0
P4	0	0	0	1	1
P5	0	0	0	1	1

The correlation between the entries of these two matrices is 0.359.

Supervised Measures of Cluster Validity

- **Example:**

Ideal class similarity matrix

	P1	P2	P3	P4	P5
P1	1	1	0	0	0
P2	1	1	0	0	0
P3	0	0	1	1	1
P4	0	0	1	1	1
P5	0	0	1	1	1

Ideal cluster similarity matrix

	P1	P2	P3	P4	P5
P1	1	1	1	0	0
P2	1	1	1	0	0
P3	1	1	1	0	0
P4	0	0	0	1	1
P5	0	0	0	1	1

Ideal class similarity vector : 1 0 0 0 0 0 0 1 1 1

Ideal cluster similarity vector : 1 1 0 0 1 0 0 0 0 1

$f_{00} = 4$, $f_{01} = 2$, $f_{10} = 2$, $f_{11} = 2$

$$Rand\ statistic = \frac{4+2}{4+2+2+2} = 0.6$$

$$Jaccard\ coefficient = \frac{2}{4+2+2+2} = 0.33$$