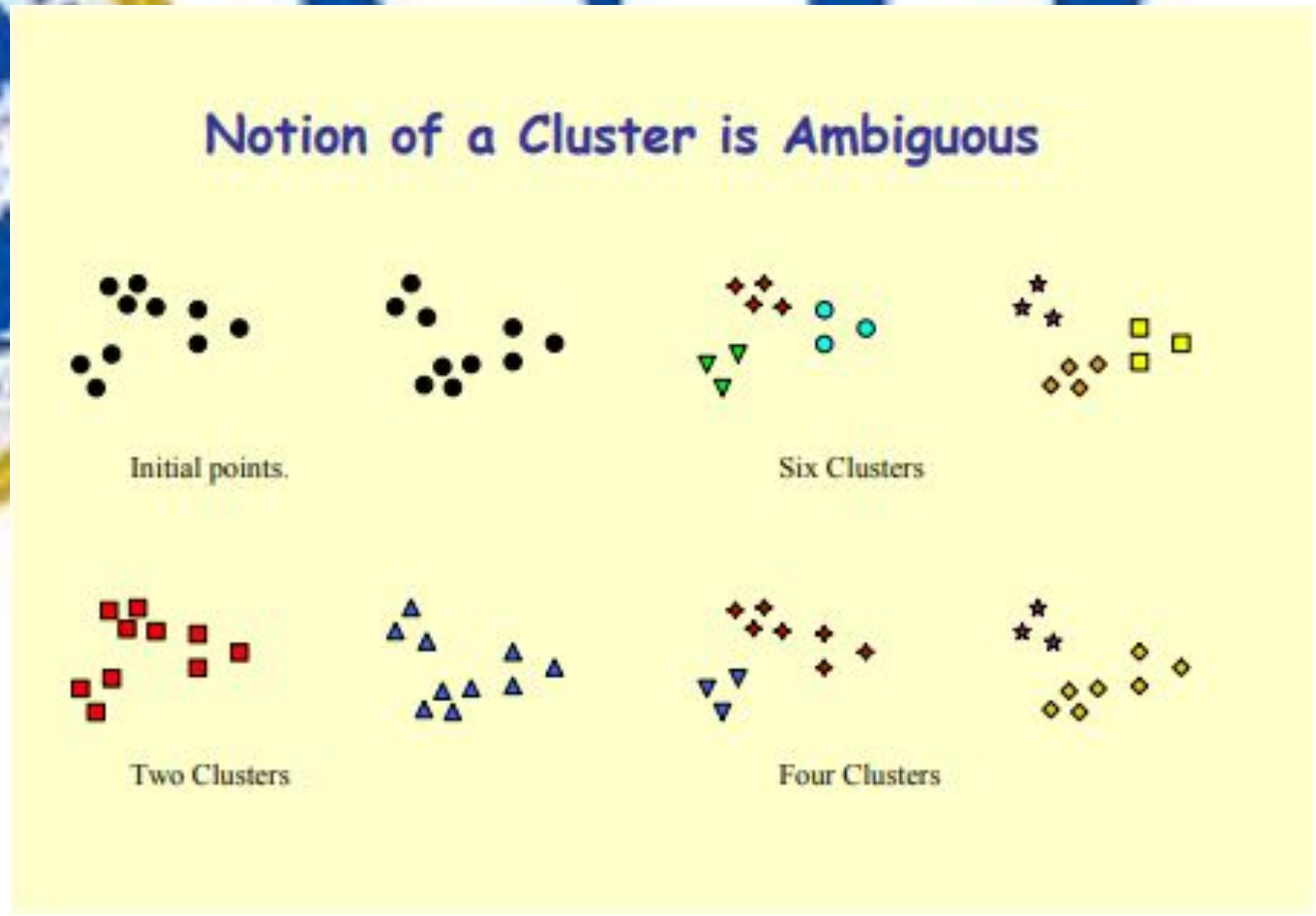
Cluster Analysis

Unit - 4



Syllabus

- *Cluster Analysis: Introduction*
- *Requirements and overview of different categories*
- *Partitioning method: Introduction*
- *k-means*
- *k-medoids*
- *Hierarchical method: Introduction*
- *Agglomerative vs. Divisive method*
- *Distance measures in algorithmic methods*
- *BIRCH technique*
- *DBSCAN technique*
- *STING technique*
- *CLIQUE technique*
- *Evaluation of clustering techniques*

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Session 1

Cluster Analysis: Introduction

Requirements and overview of different categories

- *Clustering* is the process of grouping a set of data objects intomultiple groups or *clusters*
- so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters.
- Dissimilarities and similarities are assessed based on the attribute values describing the objects and often involve distance measures.
- Clustering as a data mining tool has its roots in many application areas such as biology, security, business intelligence, and Web search.

Cluster Analysis

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Applications of Cluster Analysis

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research
- Owing to the huge amounts of data collected in databases, cluster analysis has recently become a highly active topic in data mining research.

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - It is hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective

Requirements and Challenges (of clustering Data mining)

- Scalability
  - Clustering all the data instead of only on samples // Example: Web search scenarios
  - Sample bias should not exist
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
  - complex data types such as graphs, sequences, images, and documents.
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters//the clustering results may be sensitive to such parameters
  - Quality of clusters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data //outlier. Erraneous,missing,unknown
    - need clustering methods that are robust to noise.
  - Incremental clustering and insensitivity to input order // D will be updated
  - High dimensionality

Types of Data in Cluster Analysis

Data matrix (or *object-by-variable structure*):

This represents *n* objects, such as persons, with *p* variables (also called *measurements* or *attributes*), such as age, height, weight, gender, and so on.

- The structure is in the form of a **relational table, or *n*-by-*p* matrix (*n* objects *p* variables)**

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Data matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Dissimilarity matrix

Types of Data in Cluster Analysis

Dissimilarity matrix (or *object-by-object structure*):

This stores a collection of proximities that are available for all pairs of *n* objects. It is often represented by an *n*-by-*n* table.

- d(i, j) - difference or dissimilarity between objects i and j.
- d(i, j) - **nonnegative number** that is **close to 0** when objects i and j are **highly similar or "near"** each other, and becomes larger the more they differ.
- Since d(i, j)=d(j, i), and d(i, i)=0.

- Data matrix - two-mode matrix.(Deals with different entities)
- Dissimilarity matrix - one-mode matrix.(Deals with same entity)

Types of Data in Cluster Analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

1. Interval-Scaled Variables

- Interval-scaled variables are continuous measurements of a roughly linear scale. Eg: weight and height
- The measurement unit used can affect the clustering analysis.
- For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to a very different clustering structure.
- To avoid dependence on the choice of measurement units, the data should be standardized.
- Standardizing measurements attempts to give all variables an equal weight.
- **Data Transformation by Normalization**
- The measurement unit used can affect the data analysis.

To help avoid dependence on the choice of measurement units, the data should be *normalized* or *standardized*. This involves transforming the data to fall within a smaller or common range

How can the data for a variable be standardized?

- To standardize measurements, one choice is to convert the original measurements to unit-less variables.

1. Calculate the mean absolute deviation, s_f:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|), \tag{7.3}$$

where $x_{1f}, \ldots, x_{nf}$ are n measurements of f, and m_f is the *mean* value of f, that is, $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf})$.

2. Calculate the standardized measurement, or z-score:

$$z_{if} = \frac{x_{if} - m_f}{s_f}. \tag{7.4}$$

Distance measures

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

where $i = (x_{i_1}, x_{i_2}, \ldots, x_{i_p})$ and $j = (x_{j_1}, x_{j_2}, \ldots, x_{j_p})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

- *If $q = 2$, d is* Euclidean distance

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

Distance Measure

- After standardization, or without standardization in certain applications, the dissimilarity (or similarity) between the objects described by interval-scaled variables is typically computed based on the distance between each pair of objects. The most popular distance measure is **Euclidean distance**

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2},$$

- Another well-known metric is **Manhattan (or city block) distance**, defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|.$$

Distance Measure

- Both the Euclidean distance and Manhattan distance satisfy the following mathematic requirements of a distance function

1. $d(i, j) \geq 0$: Distance is a nonnegative number.
2. $d(i, i) = 0$: The distance of an object to itself is 0.
3. $d(i, j) = d(j, i)$: Distance is a symmetric function.
4. $d(i, j) \leq d(i, h) + d(h, j)$: Going directly from object i to object j in space is no more than making a detour over any other object h (*triangular inequality*).

Distance Measure

- **Minkowski distance** is a generalization of both Euclidean distance and Manhattan distance. It is defined as

$$d(i, j) = (|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{in} - x_{jn}|^p)^{1/p},$$

- p - positive integer. Such a distance is also called Lp norm.
- It represents the Manhattan distance when $p = 1$ (i.e., $L1$ norm)
- Euclidean distance when p = 2 (i.e., L2 norm).

- If each variable is assigned a weight according to its perceived importance, the weighted Euclidean distance can be computed as

$$d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \cdots + w_m|x_{in} - x_{jn}|^2}.$$

Distance Measure

Euclidean distance and Manhattan distance. Let $x_1 = (1, 2)$ and $x_2 = (3, 5)$ represent two objects as in Figure 7.1. The Euclidean distance between the two is $\sqrt{(2^2 + 3^2)} = 3.61$. The Manhattan distance between the two is $2 + 3 = 5$. ■

Example: Data Matrix and Dissimilarity Matrix

Data Matrix

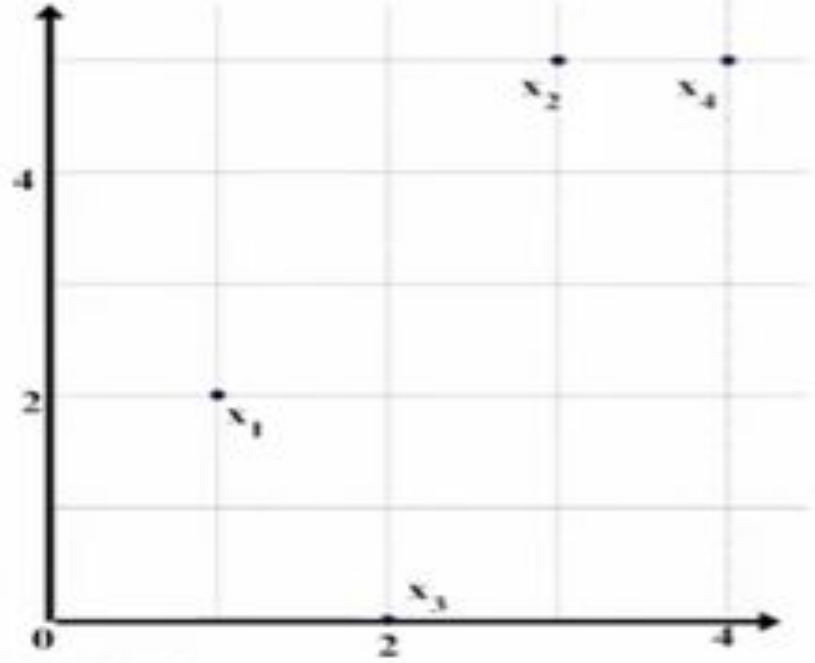
point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity Matrix (by Euclidean Distance)

	x1	*x2*	*x3*	*x4*
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Distance Measure

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

2. Binary Variables

- How to compute the dissimilarity between objects described by either *symmetric* or *asymmetric binary variables*.
- A binary variable has only **two states: 0 or 1**
- **0 – absent & 1- present**
- Treating binary variables as if they are interval-scaled can lead to misleading clustering results.
- If all binary variables are thought of as having the same weight, we have the 2-by-2 contingency table as

A contingency table for binary variables.

		object j		
		1	0	sum
object i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

Binary Variables

- A contingency table for binary data
- Distance measure for symmetric binary variables:
- Distance measure for asymmetric binary variables:
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

		Object j = 1	Object j = 0	sum
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

$$d(i,j) = \frac{b+c}{a+b+c}$$

$$sim(i,j) = 1 - d(i,j)$$

Binary Variables

- The total number of variables is p, where $p = q+r+s+t$.
- A binary variable is symmetric if both of its states are equally valuable and carry the same weight.
- There is no preference on which outcome should be coded as 0 or 1.

$$d(i, j) = \frac{r+s}{q+r+s+t}.$$

- Dissimilarity that is based on symmetric binary variables is called symmetric binary dissimilarity.
- A binary variable is asymmetric if the outcomes of the states are not equally important, such as the positive and negative outcomes of a disease test.

- A binary variable contains two possible outcomes: 1 (positive/present) or 0 (negative/absent). If there is no preference for which outcome should be coded as 0 and which as 1, the binary variable is called ***symmetric***.
- For example, the binary variable "is evergreen?" for a plant has the possible states "loses leaves in winter" and "does not lose leaves in winter." Both are equally valuable and carry the same weight when a proximity measure is computed.
- If the outcomes of a binary variable are not equally important, the binary variable is called ***asymmetric***.
- An example of such a variable is the presence or absence of a relatively rare attribute, such as "is color-blind" for a human being.
- While you say that two people who are color-blind have something in common, you cannot say that people who are not color-blind have something in common.

Jaccard Coefficient

- The number of negative matches, *t*, is considered unimportant and thus is ignored in the computation, as

$$d(i, j) = \frac{r+s}{q+r+s}.$$

- we can measure the distance between two binary variables based on the notion of similarity instead of dissimilarity.

$$sim(i, j) = \frac{q}{q+r+s} = 1 - d(i, j).$$

- The coefficient sim(i, j) is called the **Jaccard coefficient**.

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- gender is a symmetric attribute
- the remaining attributes are asymmetric binary
- let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	P	N	P	N	N	N
Mary	F	P	N	P	N	P	N
Jim	M	P	P	N	N	N	N

Object1,object 2=(1,1) or (p,p)=a
Object1,object 2=(1,0) or (p,n) =b
Object1,object 2=(0,1) or (n,p)= c
Object1,object 2=(0,0) or (n,n)=d
a= ? b=? c=?

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Binary Variables

		Object j 1	0	sum
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- A contingency table for binary data
- Distance measure for symmetric binary variables: $d(i,j) = \frac{b+c}{a+b+c+d}$
- Distance measure for asymmetric binary variables: $d(i,j) = \frac{b+c}{a+b+c}$
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables): $sim(i,j) = 1 - d(i,j)$

How can we compute the dissimilarity between objects described by categorical, ordinal, and ratio-scaled variables?"

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Categorical, Ordinal, and Ratio-Scaled Variables

- A **categorical variable** is a generalization of the binary variable in that it can take on more than two states.
- For example, *map color* is a categorical variable that may have, say, five states: *red, yellow, green, pink*, and *blue*.
- Let the number of **states** of a categorical variable be **M**. The states can be denoted by letters, symbols, or a set of integers, such as 1, 2, : : : , M.
- The dissimilarity between two objects i and j can be computed based on the ratio of mismatches (Eqn 7.3)

$$d(i, j) = \frac{p - m}{p},$$

- m - where m is the number of matches (i.e., the number of variables for which i and j are in the same state), and p is the total number of variables.

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

object identifier	*test-1* (categorical)	*test-2* (ordinal)	*test-3* (ratio-scaled)
1	code-A	excellent	445
2	code-B	fair	22
3	code-C	good	164
4	code-A	excellent	1,210

Table 7.3: A sample data table containing variables of mixed type.

Suppose that we have the sample data of Table 7.3, except that only the object-identifier and the variable (or attribute) test-1 are available, where test-1 is categorical. Let's compute the dissimilarity matrix

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

Since here we have one categorical variable, *test-1*, we set $p = 1$ in Equation (7.12) so that $d(i, j)$ evaluates to 0 if objects i and j match, and 1 if the objects differ. Thus, we get

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

3. Ordinal Variables

- An ordinal variable can be discrete or continuous. (we need to convert ordinal into ratio scale)
- Order is important, e.g. rank (junior, senior)
- Can be treated like interval-scaled
- Replace an ordinal variable value by its rank: $r_{if} \in \{1, \ldots, M_f\}$
- The distance can be calculated by treating ordinal as quantitative
- Map the range of each variable onto [0,1] by replacing i-th object in f-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - Normalized Rank

- Compute the dissimilarity using methods for interval-scaled variables.

Dissimilarity between ordinal variables. Suppose that we have the sample data of Table 7.3, except that this time only the object-identi¯er and the continuous ordinal variable, test-2, are available. There are three states for test-2, namely fair, good, and excellent, that is Mf = 3. For step 1, if we replace each value for test-2 by its rank, the four objects are assigned the ranks 3, 1, 2, and 3, respectively. Step 2 normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0. For step 3, we can use, say, the Euclidean distance (Equation 7.5), which results in the following dissimilarity matrix:

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

object identifier	*test-1* (categorical)	*test-2* (ordinal)	*test-3* (ratio-scaled)
1	code-A	excellent	445
2	code-B	fair	22
3	code-C	good	164
4	code-A	excellent	1,210

Table 7.3: A sample data table containing variables of mixed types

Dissimilarity between ordinal variables

- There are three states for *test*-2, namely *fair*, *good*, and *excellent*, that is *Mf* =3.
- step 1, if we replace each value for *test*-2 by its rank, the four objects are assigned the ranks 3, 1, 2, and 3, respectively.
- Step 2 normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0.
- For step 3, we can use, say, the Euclidean distance, which results in the following dissimilarity matrix:

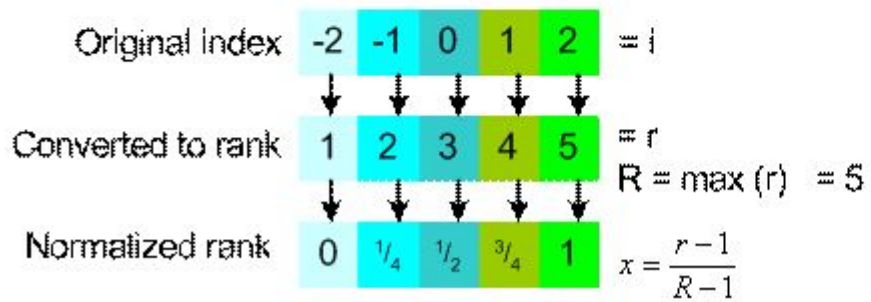
$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

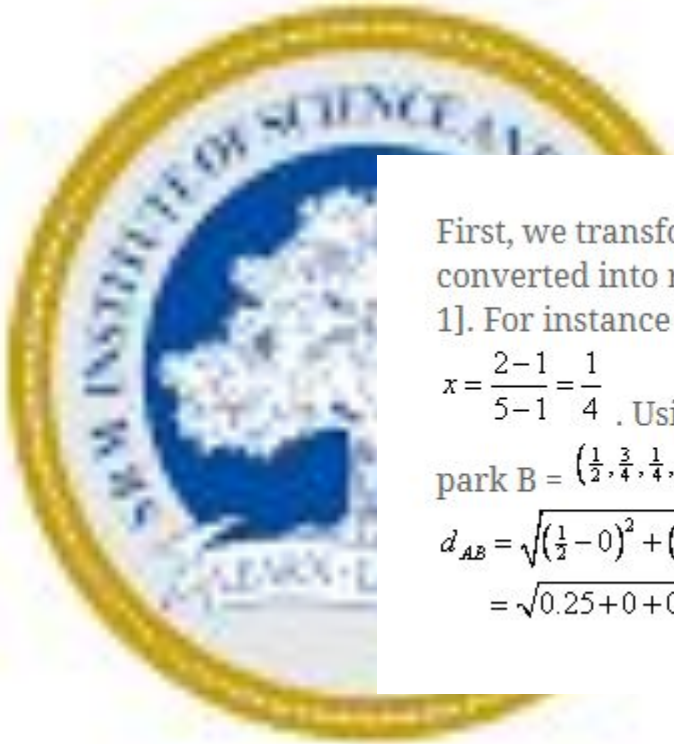
Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

We have questionnaire to ask level of satisfaction in term of safety, comfortable, convenient and proximity for two locations of park: park A and park B. Each level of satisfaction has 5 values: -2 = Very dissatisfied, - 1 = dissatisfied, 0 = indifference, 1 = satisfied, 2 = Very satisfied. Suppose the answers of respondent is as the following

	Safety	Comfortable	Convenient	Proximity
Park A	-2	1	0	2
Park B	0	1	-1	1

We want to measure dissimilarity of park A and B according to the respondent answer





First, we transform the ordinal scale into ratio scale. Original index (i = -2 to 2) is ordered and converted into rank (r = 1 to 5). The highest rank is R = 5. Then we normalized the rank into value [0, 1]. For instance in the position 2, we have i = -1, converted to rank become r = 2. Normalized rank is $x = \frac{2-1}{5-1} = \frac{1}{4}$. Using the normalized rank as new values, we have coordinates of Park A = $\left(0, \frac{3}{4}, \frac{1}{2}, 1\right)$ and park B = $\left(\frac{1}{2}, \frac{3}{4}, \frac{1}{4}, \frac{3}{4}\right)$. The Euclidean distance between park A and park B is

$$d_{AB} = \sqrt{\left(\tfrac{1}{2}-0\right)^2 + \left(\tfrac{3}{4}-\tfrac{3}{4}\right)^2 + \left(\tfrac{1}{4}-\tfrac{1}{2}\right)^2 + \left(\tfrac{3}{4}-1\right)^2}$$

$$= \sqrt{0.25+0+0.0625+0.0625} = 0.612$$

Ratio-Scaled Variables

- A ratio-scaled variable makes a positive measurement on a nonlinear scale, such as an exponential scale.

$$Ae^{Bt} \quad \text{or} \quad Ae^{-Bt}$$

- where *A* and *B* are positive constants, and *t* typically represents time. Eg: growth of a bacteria population or the decay of a radioactive element.

- There are three methods to handle ratio-scaled variables for computing the dissimilarity between objects.

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Ratio-Scaled Variables

- Treat ratio-scaled variables like interval-scaled variables.
- Apply logarithmic transformation to a ratio-scaled variable f having value x_{if} for object i by using the formula y_{if} = log(x_{if}). The y_{if} values can be treated as interval-valued.
- Treat x_{if} as continuous ordinal data and treat their ranks as interval-valued.
- The latter two methods are the most effective, although the choice of method used may depend on the given application.

Example 7.5 Dissimilarity between ratio-scaled variables. This time, we have the sample data of Table 7.3, except that only the *object-identifier* and the ratio-scaled variable, *test-3*, are available. Let's try a logarithmic transformation. Taking the *log* of *test-3* results in the values 2.65, 1.34, 2.21, and 3.08 for the objects 1 to 4, respectively. Using the Euclidean distance (Equation 7.5) on the transformed values, we obtain the following dissimilarity matrix:

$$\begin{bmatrix} 0 & & & \\ 1.31 & 0 & & \\ 0.44 & 0.87 & 0 & \\ 0.43 & 1.74 & 0.87 & 0 \end{bmatrix}$$

object identifier	*test-1* (categorical)	*test-2* (ordinal)	*test-3* (ratio-scaled)
1	code-A	excellent	445
2	code-B	fair	22
3	code-C	good	164
4	code-A	excellent	1,210

Table 7.3: A sample data table containing variables of mixed type.

Dissimilarity between ratio-scaled variables

- The ratio-scaled variable, *test-3*, are available. Let's try a logarithmic transformation.
- Taking the *log* of *test-3* results in the values 2.65, 1.34, 2.21, and 3.08 for the objects 1 to 4, respectively.

$$\begin{bmatrix} 0 & & & \\ 1.31 & 0 & & \\ 0.44 & 0.87 & 0 & \\ 0.43 & 1.74 & 0.87 & 0 \end{bmatrix}$$

Source :
http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

4. Variables of Mixed Types

- *how can we compute the dissimilarity between objects of mixed variable types?"*
- One approach is to group each kind of variable together, performing a separate cluster analysis for each variable type.
- A more preferable approach is to process all variable types together, performing a single cluster analysis.
- Suppose that the data set contains p variables of mixed type. The dissimilarity d(i, j) between objects i and j is defined as

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}},$$

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Variables of Mixed Types

where the indicator $\delta_{ij}^{(f)} = 0$ if either (1) x_{if} or x_{jf} is missing (i.e., there is no measurement of variable f for object i or object j), or (2) $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary; otherwise, $\delta_{ij}^{(f)} = 1$. The contribution of variable f to the dissimilarity between i and j, that is, $d_{ij}^{(f)}$, is computed dependent on its type:

- If f is interval-based: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$, where h runs over all nonmissing objects for variable f.
- If f is binary or categorical: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as interval-scaled.
- If f is ratio-scaled: either perform logarithmic transformation and treat the transformed data as interval-scaled; or treat f as continuous ordinal data, compute r_{if} and z_{if}, and then treat z_{if} as interval-scaled.

Source:
http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Variables of Mixed Types

- Apply logarithmic transformation to its values. Based on the transformed values of 2.65, 1.34, 2.21, and 3.08 obtained for the objects 1 to 4.
- $max_h x_h$ =3.08 and $min_h\ x_h$=1.34.
- Then normalize the values in the dissimilarity matrix obtained in Example 7.5 by dividing each one by (3.08 – 1.34) = 1.74.

$$\begin{bmatrix} 0 & & & \\ 0.75 & 0 & & \\ 0.25 & 0.50 & 0 & \\ 0.25 & 1.00 & 0.50 & 0 \end{bmatrix}$$

- We can now use the dissimilarity matrices for the three variables in our computation.

$$d(2,1) = \frac{1(1)+1(1)+1(0.75)}{3} = 0.92.$$

$$\begin{bmatrix} 0 & & & \\ 0.92 & 0 & & \\ 0.58 & 0.67 & 0 & \\ 0.08 & 1.00 & 0.67 & 0 \end{bmatrix}$$

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Vector Objects

- There are several ways to define such a similarity function, $s(\boldsymbol{x}, \boldsymbol{y})$, to compare two vectors ***x*** and ***y***.
- One popular way is to define the similarity function as a cosine measure

$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^t \cdot \boldsymbol{y}}{||\boldsymbol{x}|| ||\boldsymbol{y}||},$$

- where x' is a transposition of vector ***x***, ||x|| is the Euclidean norm of vector x' ,||y|| is the Euclidean norm of vector ***y***, and *s* is essentially the cosine of the angle between vectors ***x*** and ***y***.

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Session 2

Partitioning Method: Introduction

K-Means Algorithm

Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	– Can find arbitrarily shaped clusters – Clusters are dense regions of objects in space that are separated by low-density regions – Cluster density: Each point must have a minimum number of points within its "neighborhood" – May filter out outliers
Grid-based methods	– Use a multiresolution grid data structure – Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

The *K-Means* Clustering Method

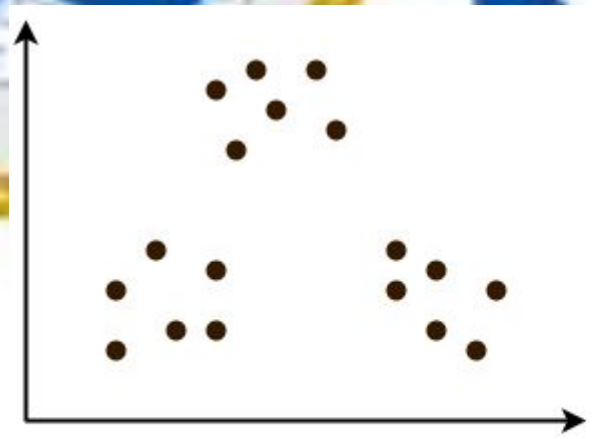
- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

K-Means Clustering-

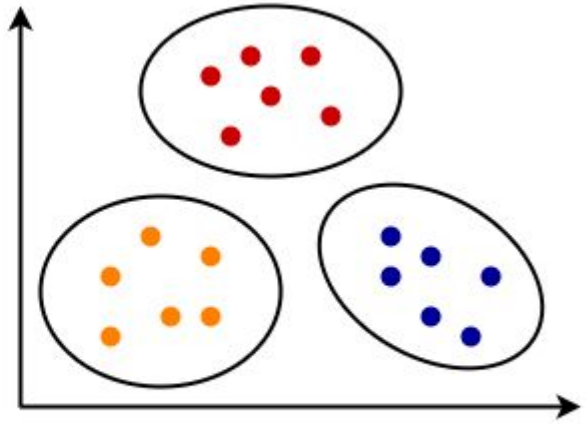
K-Means clustering is an unsupervised iterative clustering technique.
It partitions the given data set into k predefined distinct clusters.
A cluster is defined as a collection of data points exhibiting certain similarities.

Partitions each dataset such that

- Each data point belongs to a cluster with the nearest mean.
- Data points belonging to one cluster have high degree of similarity.
- Data points belonging to different clusters have high degree of dissimilarity.



Before K-Means



After K-Means

Step-01:

Choose the number of clusters K.

Step-02:

Randomly select any K data points as cluster centers.

•Select cluster centers in such a way that they are as farther as possible from each other.

Step-03:

Calculate the distance between each data point and each cluster center.

•The distance may be calculated either by using given distance function or by using euclidean distance formula.

Step-04:

Assign each data point to some cluster.

•A data point is assigned to that cluster whose center is nearest to that data point.

Step-05:

Re-compute the center of newly formed clusters.

•The center of a cluster is computed by taking mean of all the data points contained in that cluster.

Step-06:

Keep repeating the procedure from Step-03 to Step-05 until any of the following stopping criteria is met-

•Center of newly formed clusters do not change

•Data points remain present in the same cluster

•Maximum number of iterations are reached

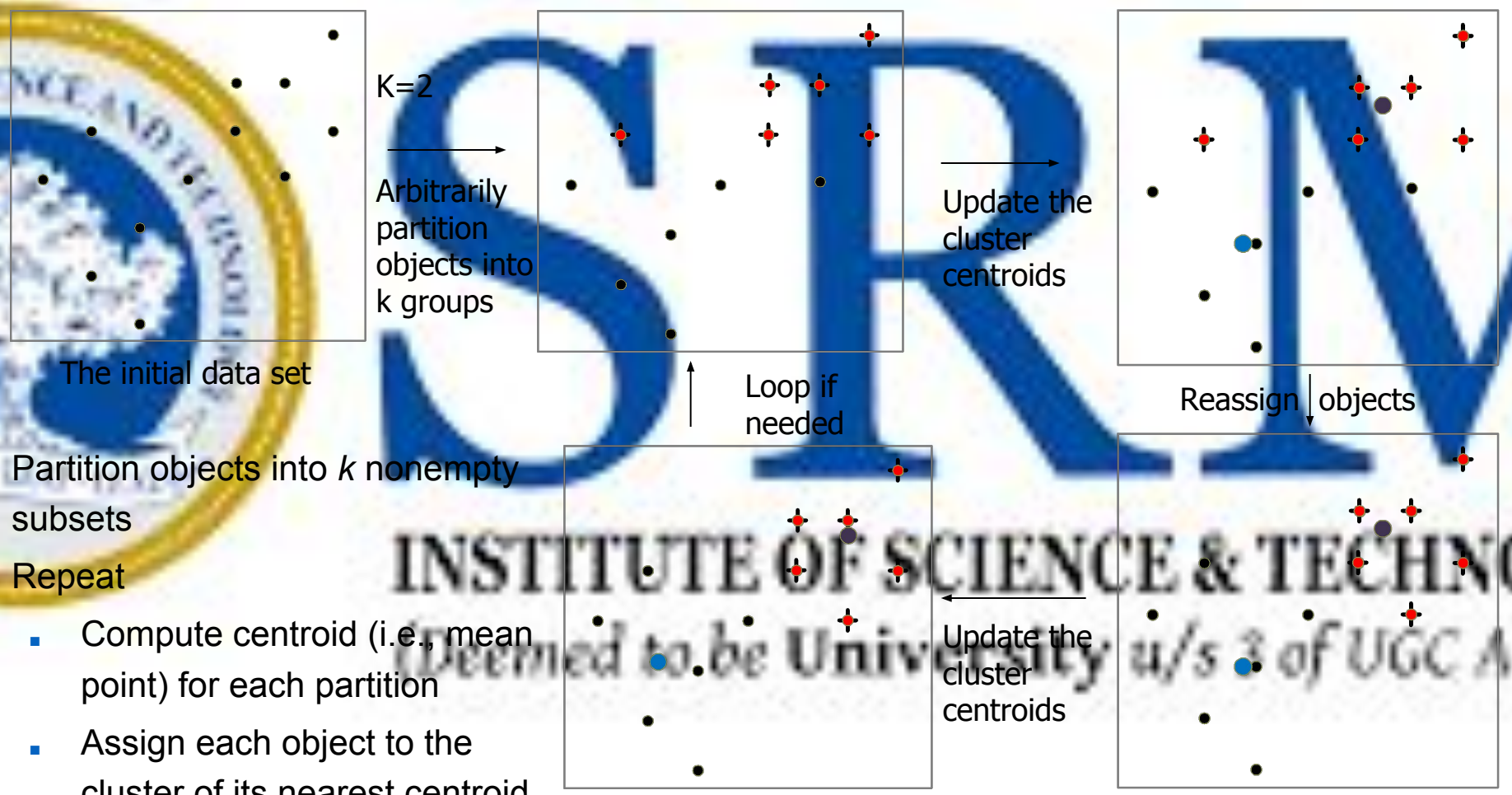
- K-Means Clustering Algorithm offers the following advantages-

- **Point-01:**
- It is relatively efficient with time complexity O(nkt) where-
- n = number of instances
- k = number of clusters
- t = number of iterations
- **Point-02:**
- It often terminates at local optimum.
- Techniques such as Simulated Annealing or Genetic Algorithms may be used to find the global optimum.

- **Disadvantages-**

- K-Means Clustering Algorithm has the following disadvantages-
- It requires to specify the number of clusters (k) in advance.
- It can not handle noisy data and outliers.
- It is not suitable to identify clusters with non-convex shapes.

An Example of *K-Means* Clustering



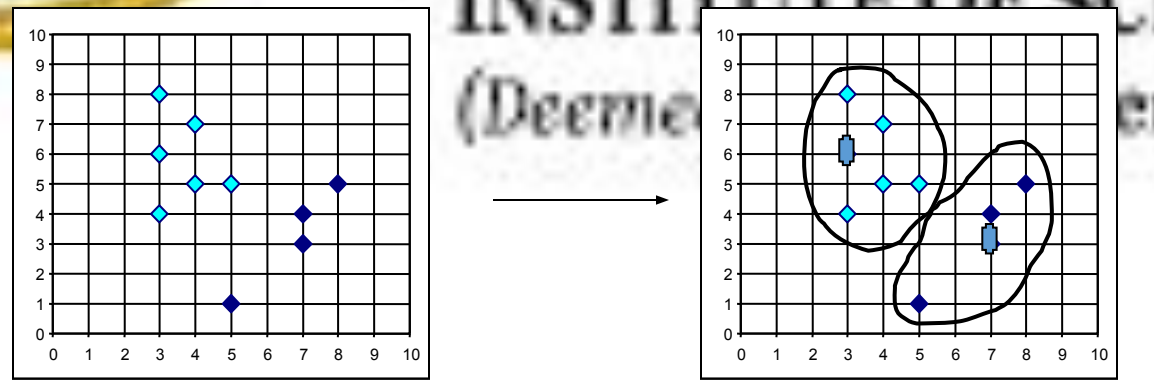
- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$ PAM// Partition around Mediods
  - Clustering in LARge Applications.
- Comment: Often terminates at a *local optimal*.
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
  - Using the k-modes method for categorical data
  - In comparison, k-medoids can be applied to a wide range of data
- Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
- Sensitive to noisy data and *outliers*
- Not suitable to discover clusters with *non-convex shapes*

What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster



- Another variant to k-means is the **k-modes method**, which extends the k-means paradigm to cluster categorical data by replacing the means of clusters with modes, using new dissimilarity measures to deal with categorical objects and a frequency-based method to update modes of clusters. The k-means and the k-modes methods can be integrated to cluster data with mixed numeric and categorical values.
- The **EM (Expectation-Maximization) algorithm** extends the k-means paradigm in a different way. Whereas the k-means algorithm assigns each object to a cluster,
- In EM,each object is assigned to each cluster according to a weight representing its probability of membership.
- In other words, there are no strict boundaries between clusters. Therefore, new means are computed based on weighted measures.

How can we make the k-means algorithm more scalable?"

A recent approach to scaling the k-means algorithm is based on the idea of identifying three kinds of regions in data:

1. regions that are compressible,
2. regions that must be maintained in main memory,
3. and regions that are discardable.

✔ An object is discardable if its membership in a cluster is ascertained.

✔ An object is compressible if it is not discardable but belongs to a tight subcluster.

✔ A data structure known as a clustering feature is used to summarize objects that have been discarded or compressed.

✔ If an objectis neither discardable nor compressible, then it should be retained in main memory.

To achieve scalability,

The iterative clustering algorithm only includes the clustering features of the compressible objects and the objects that must be retained in main memory,

- thereby turning a secondary-memory-based algorithm into a main-memory- based algorithm.
- An alternative approach to scaling the k-means algorithm explores the microclustering idea,

Cluster the following eight points (with (x, y) representing locations) into three clusters:
A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4, 9)

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).
The distance function between two points a = (x1, y1) and b = (x2, y2) is defined as-
P(a, b) = |x2 – x1| + |y2 – y1|

Use K-Means Algorithm to find the three cluster centers after the second iteration.

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1)

= |x2 – x1| + |y2 – y1|

= |2 – 2| + |10 – 10|

= 0

Calculating Distance Between A1(2, 10) and C2(5, 8)-

P(A1, C2)

= |x2 – x1| + |y2 – y1|

= |5 – 2| + |8 – 10|

= 3 + 2

= 5

Calculating Distance Between A1(2, 10) and C3(1, 2)-

P(A1, C3)

= |x2 – x1| + |y2 – y1|

= |1 – 2| + |2 – 10|

= 1 + 8

= 9

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

https://www.gatevidyalay.com/k-means-clustering-algorithm-example

- Calculate center of the clusters
- **For Cluster-01:**
  - We have only one point A1(2, 10) in Cluster-01.

  So, cluster center remains the same.
- **For Cluster-02:**

Center of Cluster-02

= ((8 + 5 + 7 + 6 + 4)/5, (4 + 8 + 5 + 4 + 9)/5)

= (6, 6)

- **For Cluster-03:**

Center of Cluster-03

= ((2 + 1)/2, (5 + 2)/2)

= (1.5, 3.5)

- This is completion of Iteration-01.

Calculating Distance Between A1(2, 10) and C1(2, 10)-

P(A1, C1)

= |x2 – x1| + |y2 – y1|

= |2 – 2| + |10 – 10|

= 0

Calculating Distance Between A1(2, 10) and C2(6, 6)-

P(A1, C2)

= |x2 – x1| + |y2 – y1|

= |6 – 2| + |6 – 10|

= 4 + 4

= 8

- **Calculating Distance Between A1(2, 10) and C3(1.5, 3.5)-**

P(A1, C3)

= |x2 – x1| + |y2 – y1|

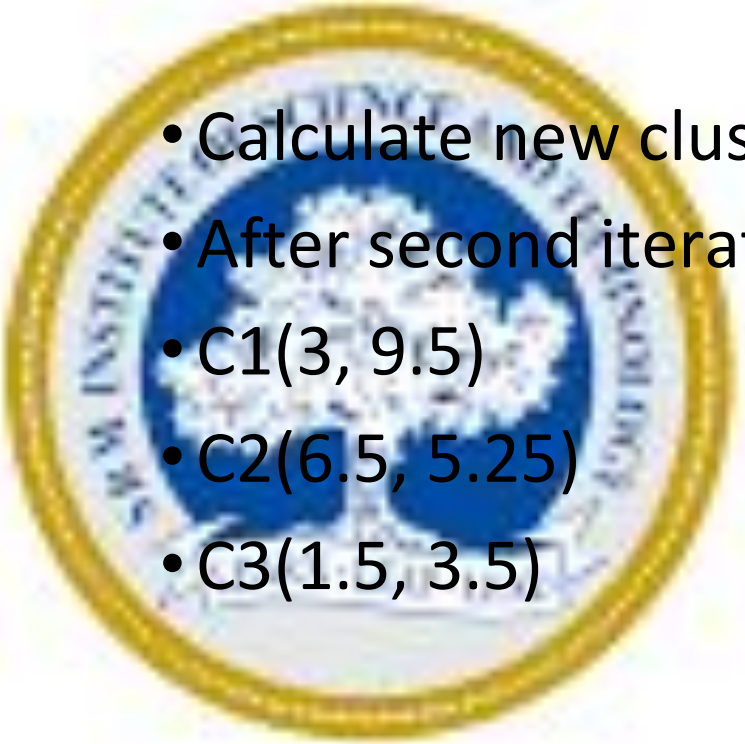
= |1.5 – 2| + |3.5 – 10|

= 0.5 + 6.5

= 7

We calculate the distance of each point from each of the center of the three clusters.
The distance is calculated by using the given distance function.

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (6, 6) of Cluster-02	Distance from center (1.5, 3.5) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	8	7	C1
A2(2, 5)	5	5	2	C3
A3(8, 4)	12	4	7	C2
A4(5, 8)	5	3	8	C2
A5(7, 5)	10	2	7	C2
A6(6, 4)	10	2	5	C2
A7(1, 2)	9	9	2	C3
A8(4, 9)	3	5	8	C1



- Calculate new cluster centers
- After second iteration, the center of the three clusters are-
- C1(3, 9.5)
- C2(6.5, 5.25)
- C3(1.5, 3.5)

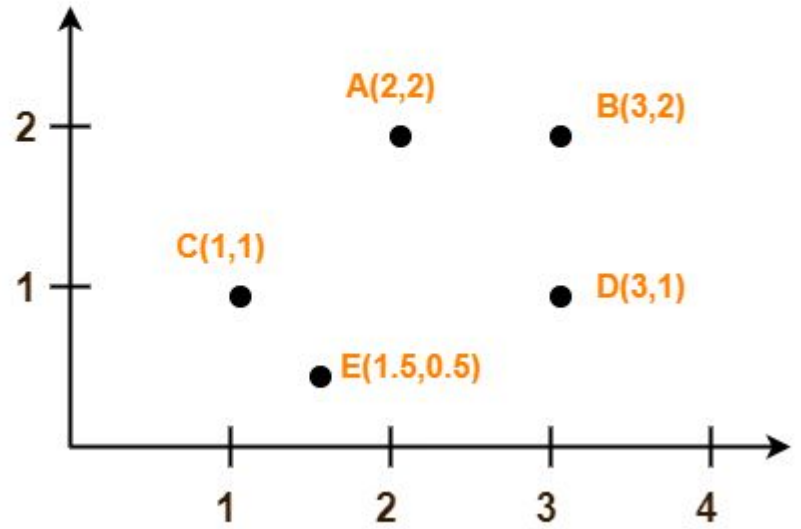
SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Practice Problem

Use K-Means Algorithm to create two clusters-



& TECHNOLOGY
3 of UGC Act, 1956)

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Session 3

K-Medoids

Hierarchical Method: Introduction

A drawback of *k*-means. Consider six points in 1-D space having the values 1, 2, 3, 8, 9, 10, and 25, respectively. Intuitively, by visual inspection we may imagine the points partitioned into the clusters {1, 2, 3} and {8, 9, 10}, where point 25 is excluded because it appears to be an outlier. How would *k*-means partition the values? If we apply *k*-means using $k = 2$ and Eq. (10.1), the partitioning $\{\{1,2,3\},\{8,9,10,25\}\}$ has the within-cluster variation

$$(1-2)^2 + (2-2)^2 + (3-2)^2 + (8-13)^2 + (9-13)^2 + (10-13)^2 + (25-13)^2 = 196,$$

given that the mean of cluster {1, 2, 3} is 2 and the mean of {8, 9, 10, 25} is 13. Compare this to the partitioning $\{\{1,2,3,8\},\{9,10,25\}\}$, for which *k*-means computes the within-cluster variation as

$$(1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (8-3.5)^2 + (9-14.67)^2 + (10-14.67)^2 + (25-14.67)^2 = 189.67,$$

given that 3.5 is the mean of cluster {1, 2, 3, 8} and 14.67 is the mean of cluster {9, 10, 25}. The latter partitioning has the lowest within-cluster variation; therefore, the *k*-means method assigns the value 8 to a cluster different from that containing 9 and 10 due to the outlier point 25. Moreover, the center of the second cluster, 14.67, is substantially far from all the members in the cluster. ■

Quality of clustering,
Variation within clustering
Error E=

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

"How can we modify the k-means algorithm to diminish such sensitivity to outliers?"

- Instead of taking the mean value of the objects in a cluster as a reference point,
- pick actual objects to represent the clusters, using one representative object per cluster.
- Each remaining object is assigned to the cluster of which the representative object is the most similar.
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object ***p*** and its correspondingrepresentative object.
- That is, an **absolute-error criterion** is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i),$$

The K-Medoid Clustering Method

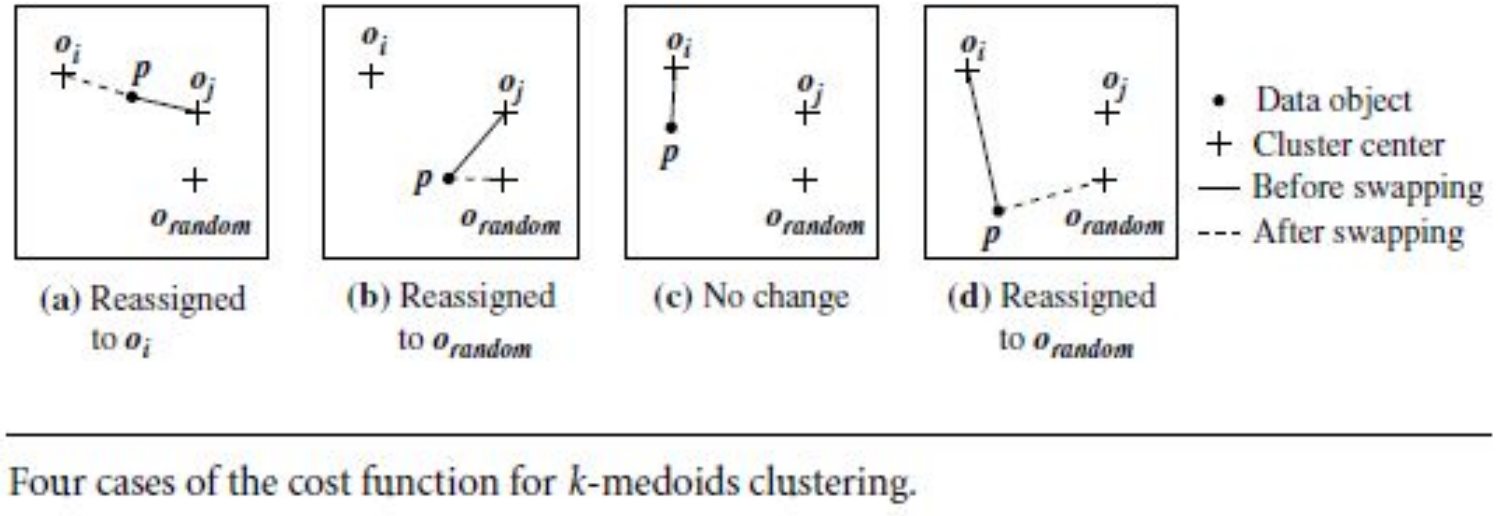
- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

- The **Partitioning Around Medoids (PAM)** algorithm is a popular realization of *k*-medoids clustering.
- It tackles the problem in an iterative, greedy way.
- Like the *k*-means algorithm, the initial representative objects (called seeds) are chosen arbitrarily.
- We consider whether replacing a representative object by a nonrepresentative object would improve the clustering quality.
- All the possible replacements are tried out. The iterative process of replacing representative objects by other objects
- continues until the quality of the resulting clustering cannot be improved by any replacement.

- Generalization of k-means
- But more robust to noise than k-means

A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal.

- The **Partitioning Around Medoids (PAM)** algorithm is a popular realization of *k*-medoids clustering. It tackles the problem in an iterative, greedy way.
- Like the *k*-means algorithm, the initial representative objects (called seeds) are chosen arbitrarily.
- It considers whether replacing a representative object by a nonrepresentative object would improve the clustering quality. All the possible replacements are tried out.
- The iterative process of replacing representative objects by other objects continues until the quality of the resulting clustering cannot be improved by any replacement.
- This quality is measured by a cost function of the average dissimilarity between an object and the representative object of its cluster



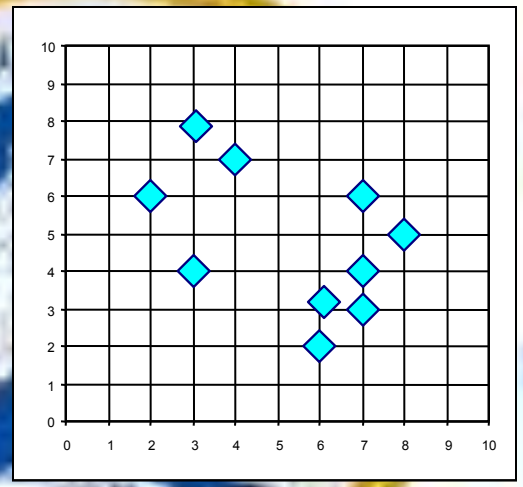
Four cases of the cost function for k-medoids clustering.

1. Specifically, let $\{o_1, \ldots, o_k\}$ be the current set of representative objects (i.e., medoids).
2. To determine whether a nonrepresentative object, denoted by o_{random}, is a good replacement for a current medoid o_j.
3. Calculate the distance from every object p to the closest object in the set $\{o_1, \ldots, o_{j-1}, o_{random}, o_{j+1}, \ldots, o_k\}$, and

use the distance to update the cost function.

- The reassignments of objects to new medoid are simple.

PAM: A Typical K-Medoids Algorithm

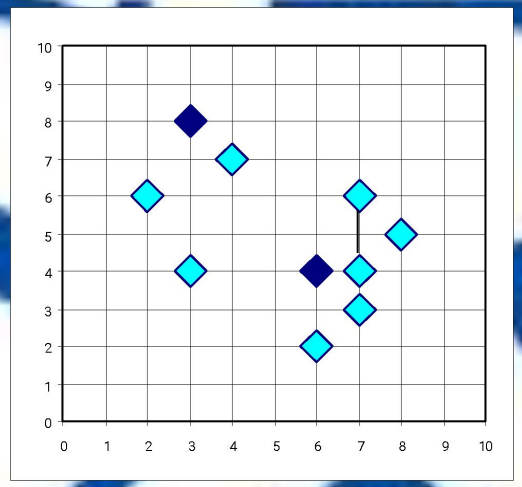


K=2

Do loop

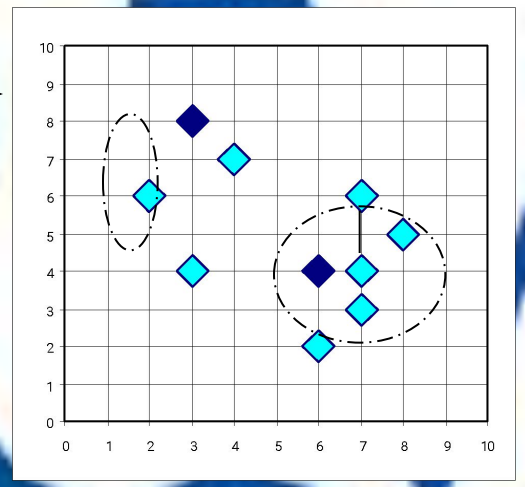
Until no change

Arbitrary choose k object as initial medoids

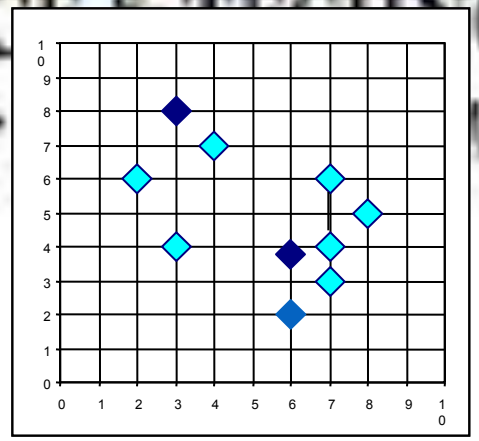


Assign each remaining object to nearest medoids

Total Cost = 20

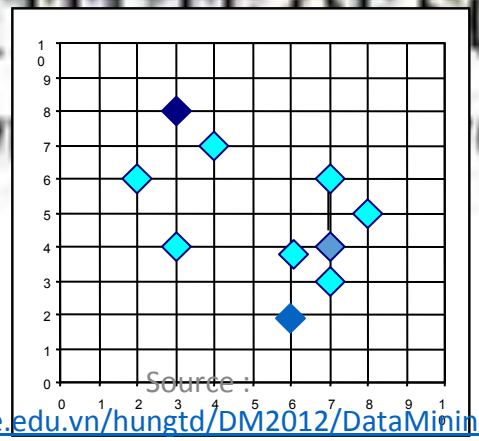


Randomly select a nonmedoid object,O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Algorithm: *k*-medoids. PAM, a *k*-medoids algorithm for partitioning based on medoid or central objects.

Input:

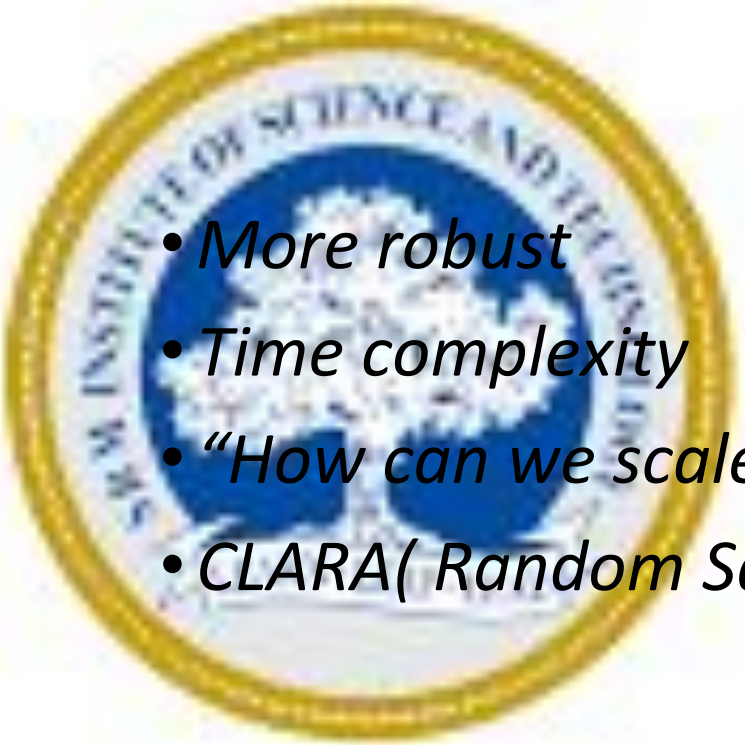
- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects in *D* as the initial representative objects or seeds;
(2) **repeat**
(3) assign each remaining object to the cluster with the nearest representative object;
(4) randomly select a nonrepresentative object, o_{random};
(5) compute the total cost, *S*, of swapping representative object, o_j, with o_{random};
(6) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of *k* representative objects;
(7) **until** no change;

PAM, a *k*-medoids partitioning algorithm.



- *More robust*
- *Time complexity* $O(k(n-k)^2)$.
- *"How can we scale up the k-medoids method?"*
- *CLARA(Random Samples)*

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

- In some situations we may want to partition our data into groups at different levels such as in a hierarchy.
- A **hierarchical clustering method** works by grouping data objects into a hierarchy or “tree” of clusters.
- Representing data objects in the form of a hierarchy is useful for data summarization and visualization.
- Handwriting recognition, hierarchy of species(animals,birds etc),employee,gaming (chess).

- Agglomerative versus divisive hierarchical clustering,
- which organize objects into a hierarchy using a bottom-up or top-down strategy, respectively.
- ***Agglomerative methods*** start with individual objects as clusters, which are iteratively merged to form larger clusters.
- Conversely, ***divisive methods*** initially let all the given objectsform one cluster, which they iteratively split into smaller clusters.
- Hierarchical clustering methods can encounter difficulties regarding the selection of merge or split points. Such a decision is critical,

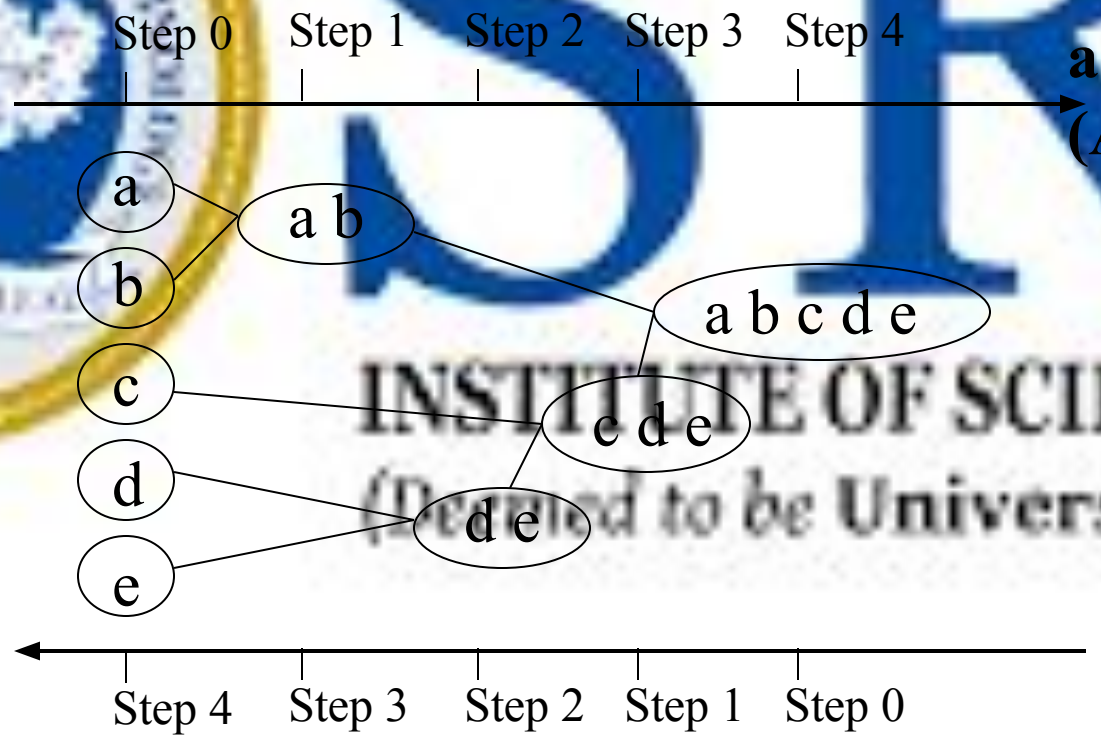
- merge or split decisions, if not well chosen, may lead to low-quality clusters.

Moreover, the methods do not scale well because each decision of merge or split needs to examine and evaluate many objects or clusters.

Solution: can be combined with Multiphase clustering

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



agglomerative
(AGNES-agglometriveNesting)

divisive
(DIANA) DIVisive ANAlysis

Source :
http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Session 4

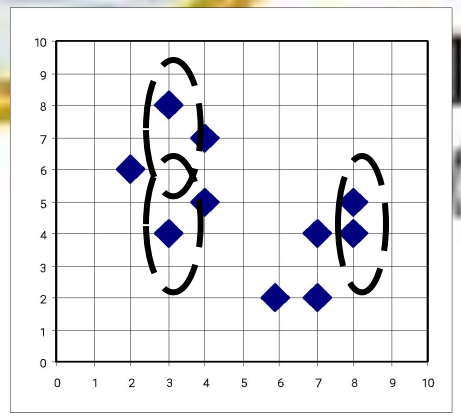
Agglomerative vs. Divisive method

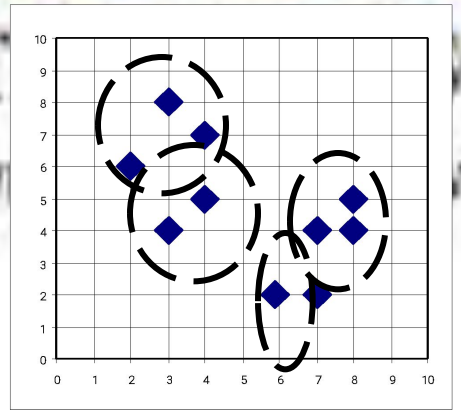
Distance measures in Algorithmic methods

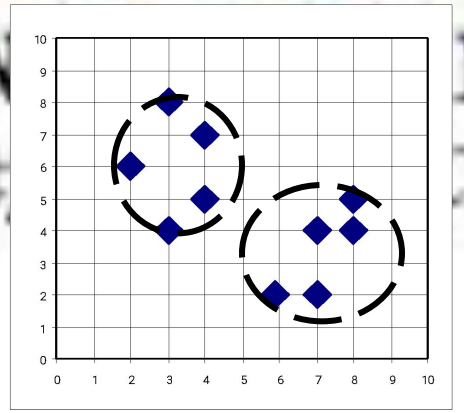
BIRCH technique
DBSCAN technique
STING technique
CLIQUE technique
Evaluation of clustering techniques

AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster







- **Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.**

- **A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.**

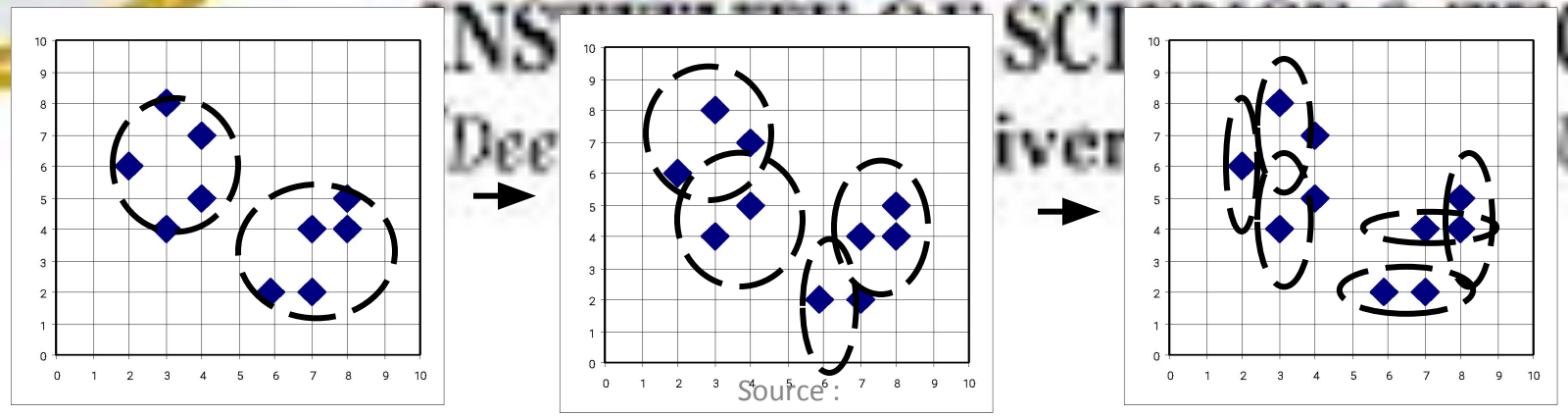
- A **dendrogram** is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering. The main use of a **dendrogram** is to work out the best way to allocate objects to clusters.

Dendrogram: Shows How Clusters are Merged

Source :
http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

DIANA (Divisive Analysis)

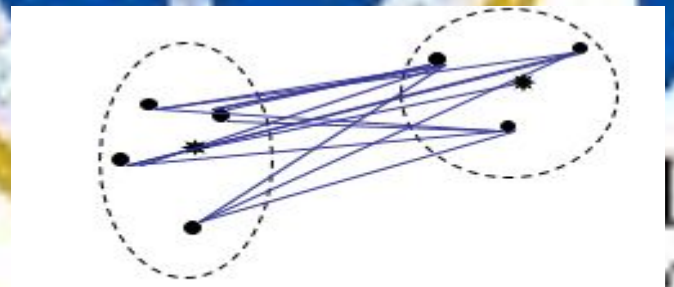
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own

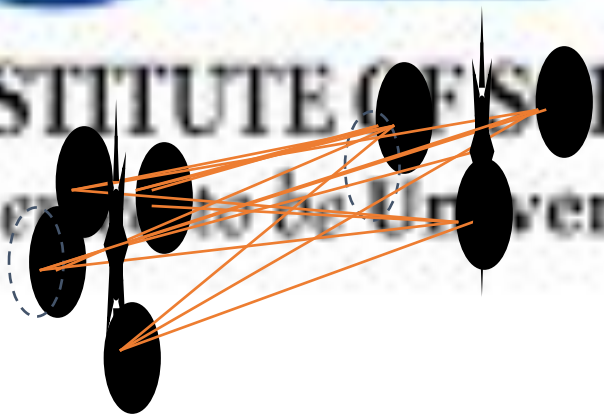


Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Distance Between Clusters

- ***Single Link***: smallest distance between points
- ***Complete Link:*** largest distance between points
- ***Average Link:*** average distance between points
- ***Centroid:*** distance between centroids





Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = min(t_{ip}, t_{jq})// updating distance matrix.
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = max(t_{ip}, t_{jq})
- Average: avg distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = avg(t_{ip}, t_{jq})
- Centroid: distance between the centroids of two clusters, i.e., dist(K_i, K_j) = dist(C_i, C_j)
- Medoid: distance between the medoids of two clusters, i.e., dist(K_i, K_j) = dist(M_i, M_j)
  - Medoid: a chosen, centrally located object in the cluster

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Hierarchical Clustering

- Clusters are created in levels actually creating sets of clusters at each level.
- ***Agglomerative***
  - Initially each item in its own cluster
  - Iteratively clusters are merged together
  - Bottom Up
- ***Divisive***
  - Initially all items in one cluster
  - Large clusters are successively divided
  - Top Down

Hierarchical Algorithms

- Single Link
- MST Single Link
- Complete Link
- Average Link

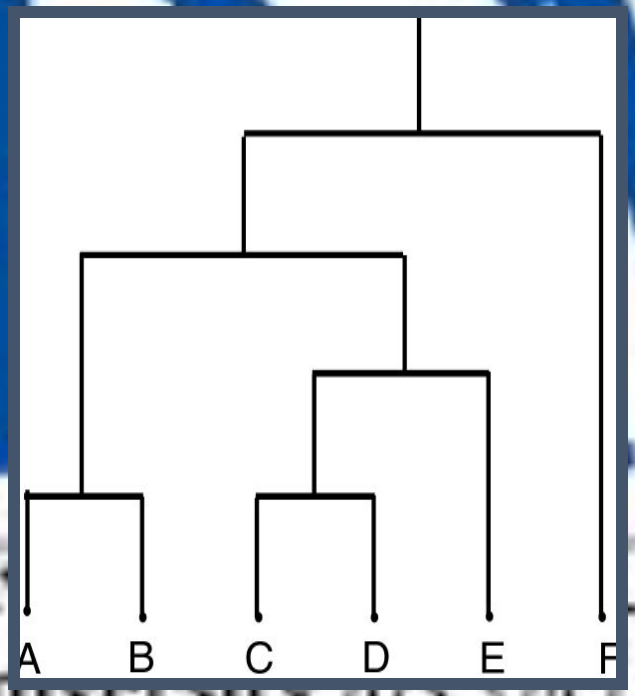
SRM

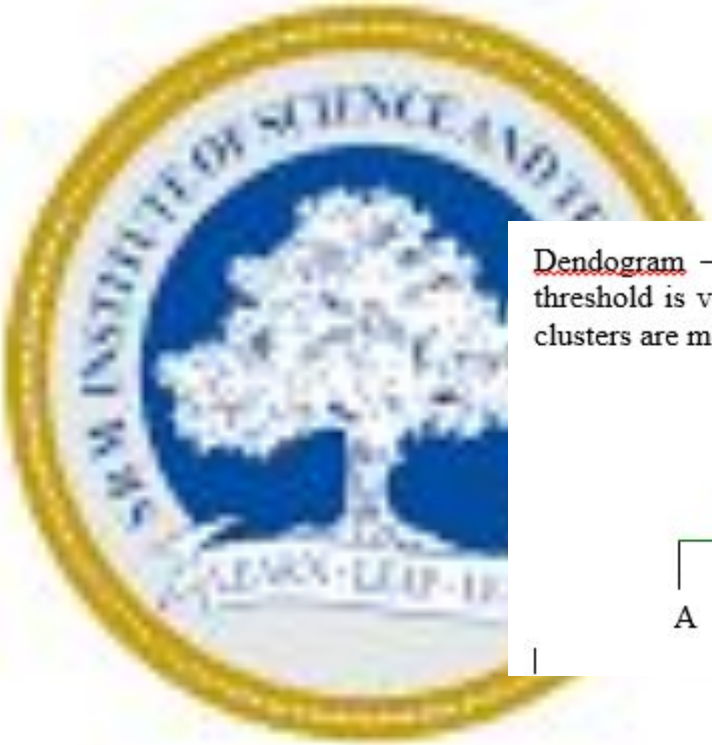
INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Dendrogram

- ***Dendrogram:*** a tree data structure which illustrates hierarchical clustering techniques.
- Each level shows clusters for that level.
  - Leaf – individual clusters
  - Root – one cluster
- A cluster at level i is the union of its children clusters at level i+1.



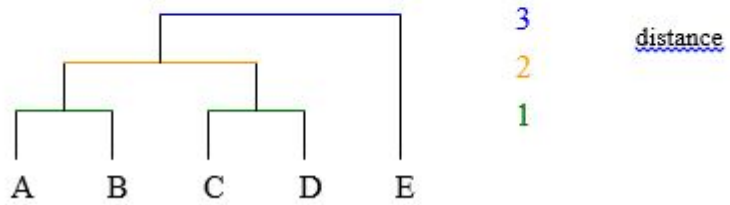


SRM

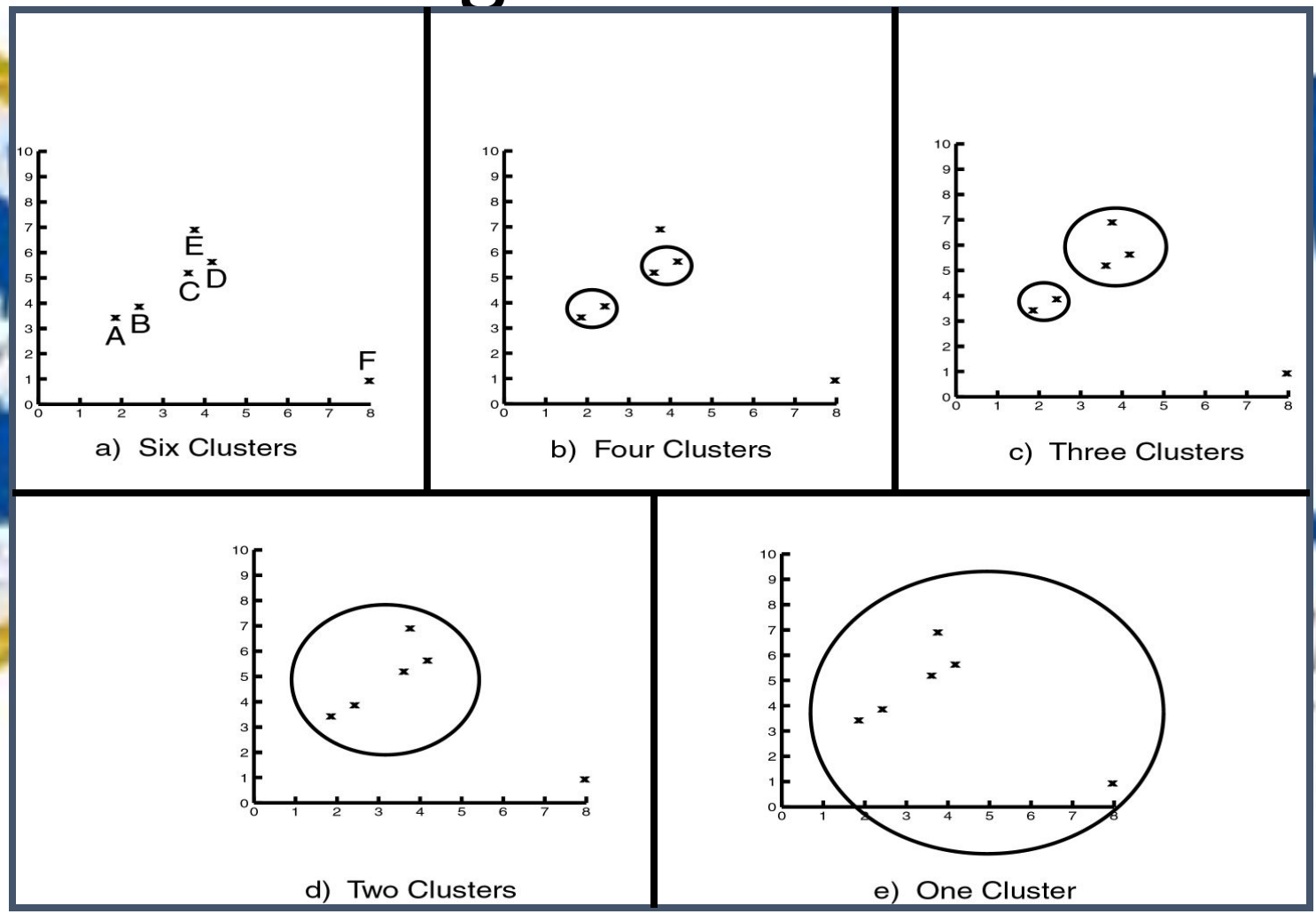
INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Dendogram – shows the same information as in the graph above, however distance threshold is vertical, and points are at the bottom (horizontal). The height at which two clusters are merged in the dendogram reflects the distance of the two clusters.

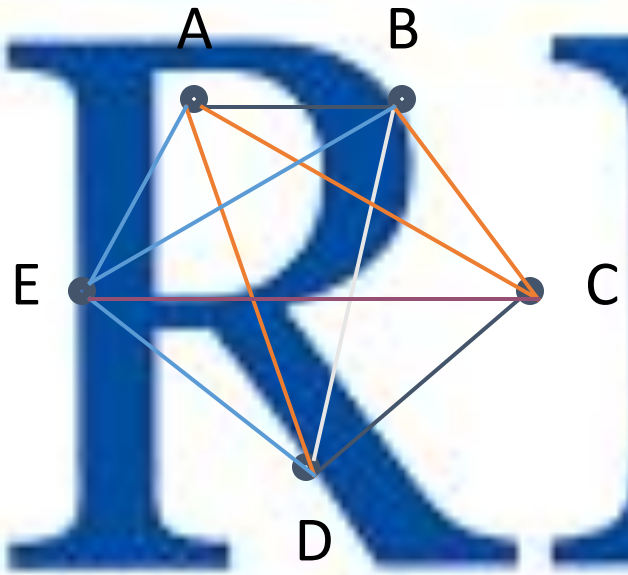


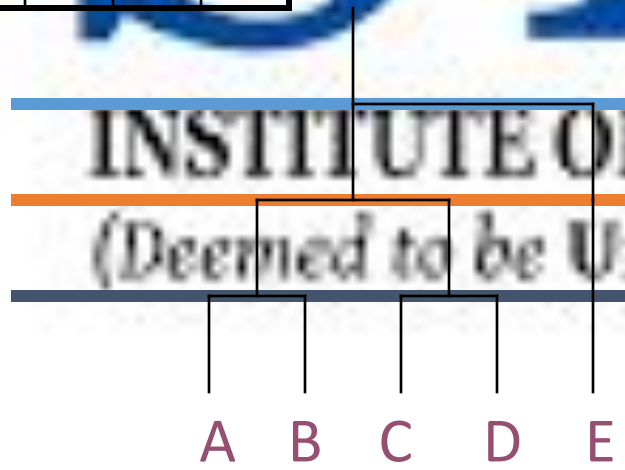
Levels of Clustering



Agglomerative Example

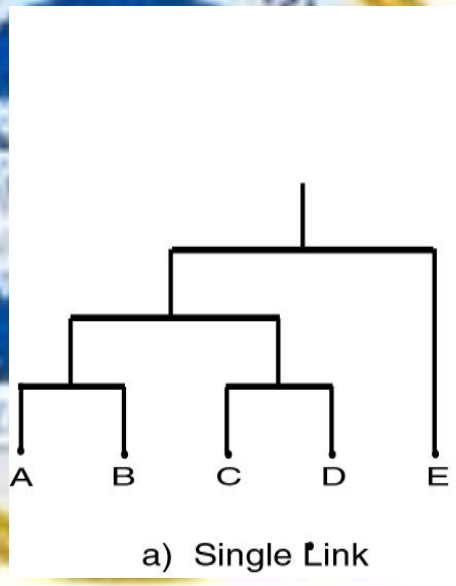
	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



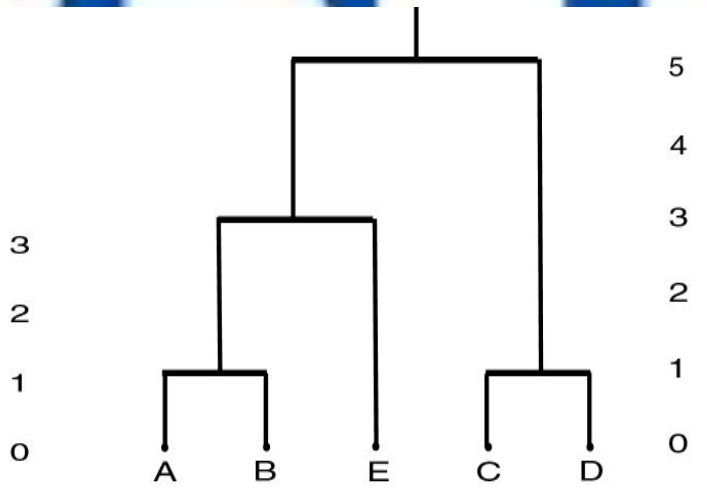


Threshold of

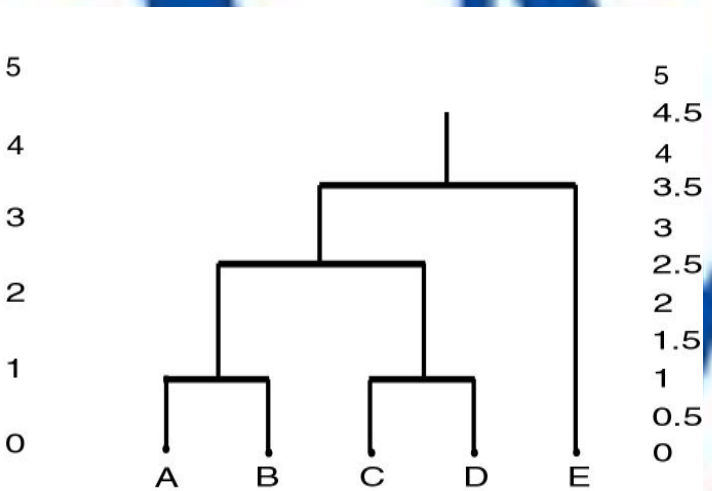
1 2 3 4 5



a) Single Link



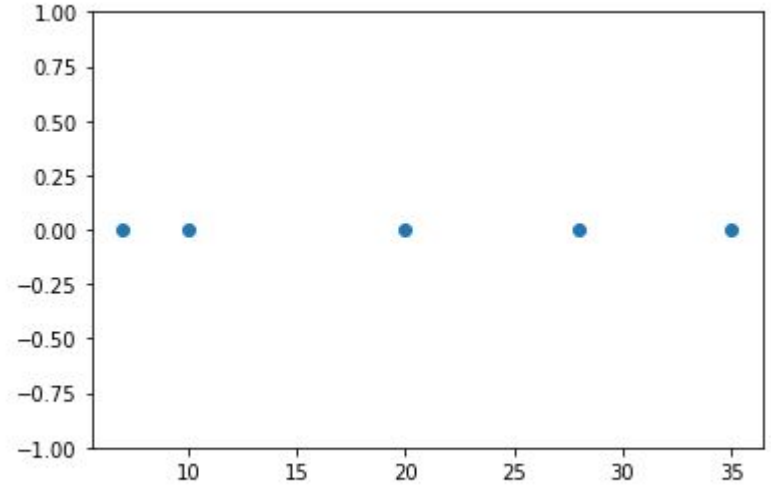
b) Complete Link

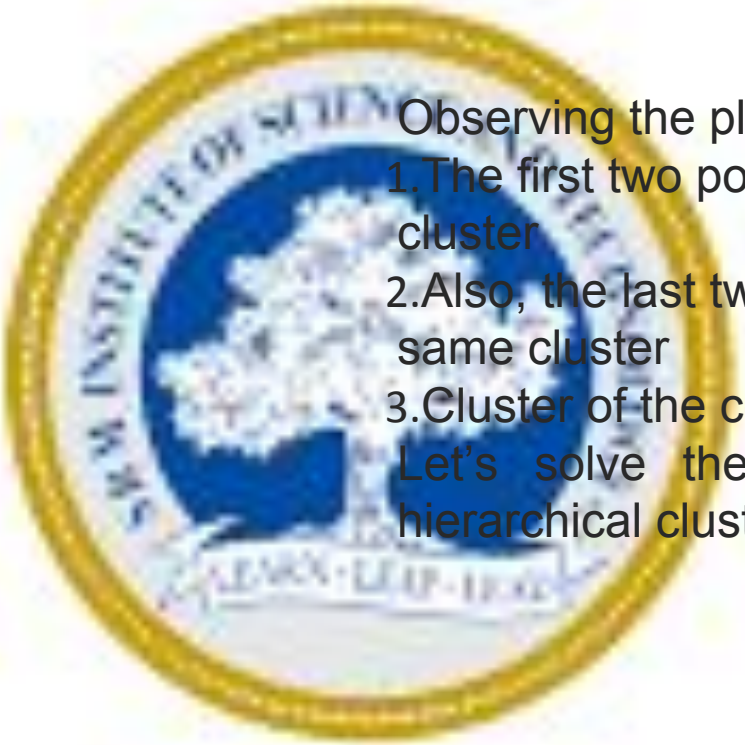


b) Average Link

Problem: For the one dimensional data set **{7,10,20,28,35}**, perform hierarchical clustering and plot the dendogram to visualize it.

Solution : First, let's the visualize the data.



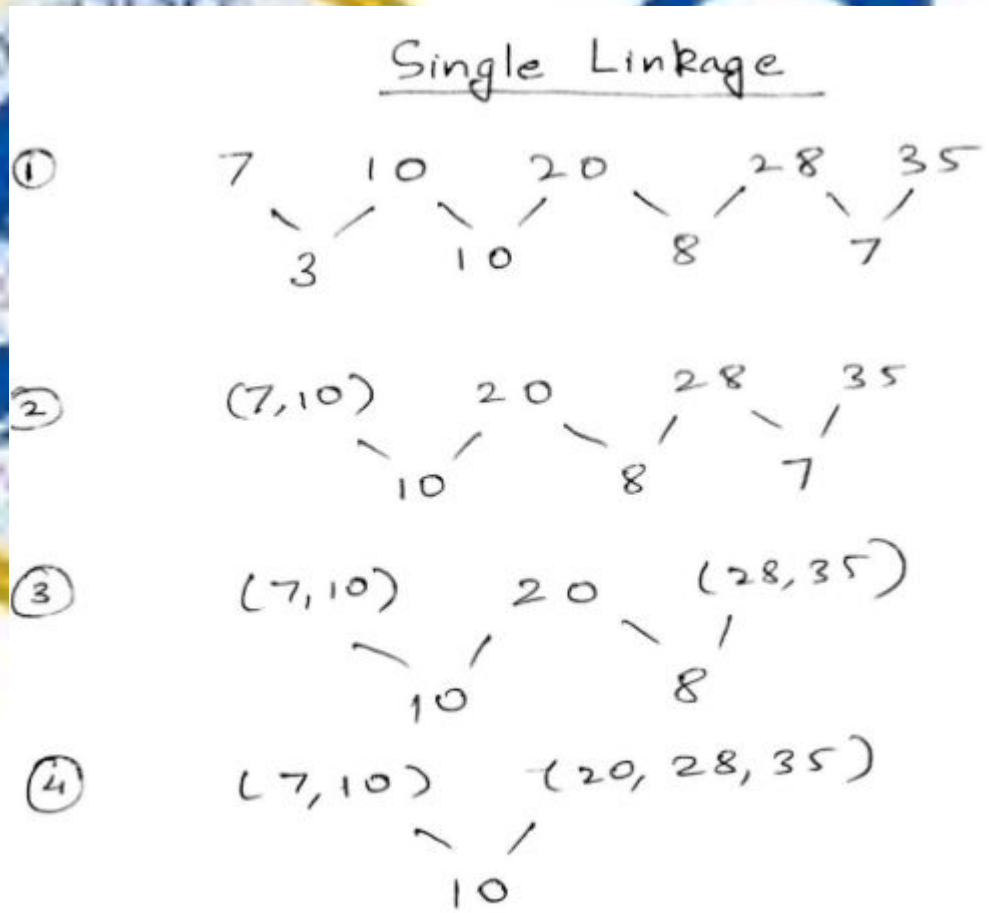


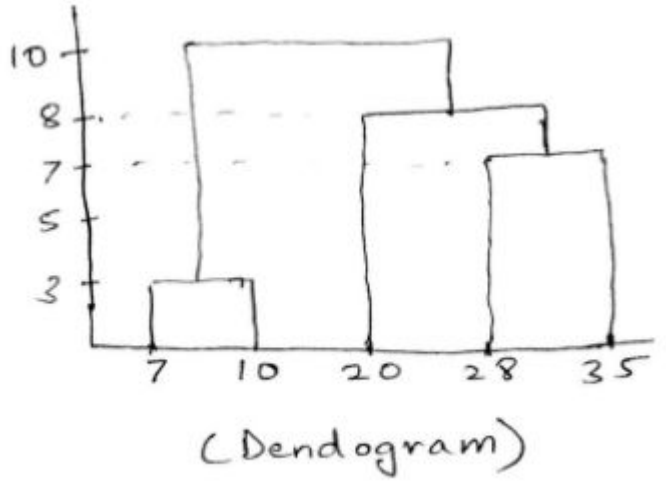
Observing the plot above, we can intuitively conclude that:

1. The first two points (7 and 10) are close to each other and should be in the same cluster
2. Also, the last two points (28 and 35) are close to each other and should be in the same cluster
3. Cluster of the center point (20) is not easy to conclude

Let's solve the problem by hand using both the types of agglomerative hierarchical clustering :

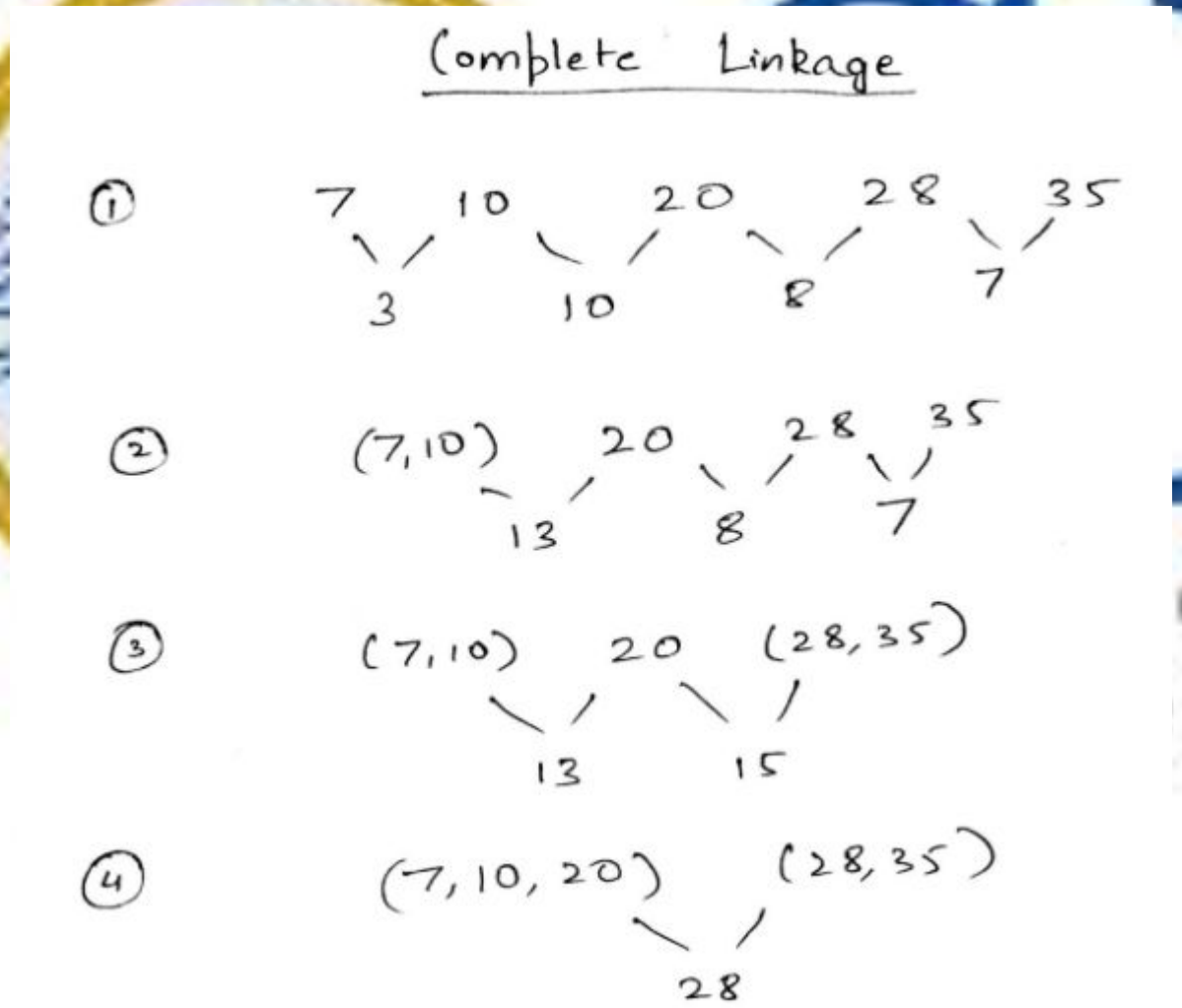
- **Single Linkage :** In single link hierarchical clustering, we merge in each step the two clusters, whose two closest members have the smallest distance.

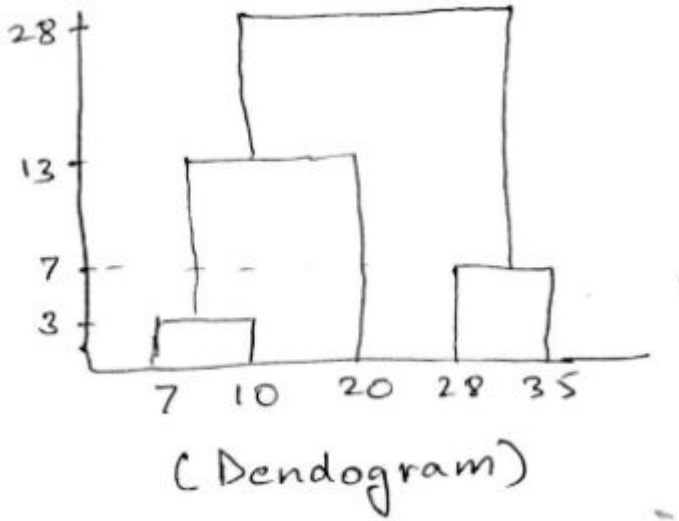




Using single linkage two clusters are formed :
Cluster 1 : (7,10)
Cluster 2 : (20,28,35)

Complete Linkage : In complete link hierarchical clustering, we merge in the members of the clusters in each step, which provide the smallest maximum pairwise distance.





Using complete linkage two clusters are formed :

Cluster 1 : (7,10,20)

Cluster 2 : (28,35)

- https://online.stat.psu.edu/stat555/node/86

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	**2**	8	0

Agglomerative Algorithm

Input:

$D = \{t_1, t_2, ..., t_n\}$ // Set of elements

A // Adjacency matrix showing distance between elements.

Output:

DE // Dendrogram represented as a set of ordered triples.

Agglomerative Algorithm:

$d = 0$;

$k = n$;

$K = \{\{t_1\}, ..., \{t_n\}\}$;

$DE = \{< d, k, K >\}$; // Initially dendrogram contains each element in its own cluster.

repeat

$oldk = k$;

$d = d + 1$;

$A_d =$ Vertex adjacency matrix for graph with threshold distance of d;

$< k, K >= NewClusters(A_d, D)$;

if $oldk \neq k$ then

$DE = DE \cup < d, k, K >$; // New set of clusters added to dendrogram.

until $k = 1$

Single Link

- View all items with links (distances) between them.
- Finds maximal connected components in this graph.
- Two clusters are merged if there is at least one edge which connects them.
- Uses threshold distances at each level.
- Could be agglomerative or divisive.

MST Single Link Algorithm

Input:

$D = \{t_1, t_2, ..., t_n\}$ // Set of elements

A // Adjacency matrix showing distance between elements.

Output:

DE // Dendrogram represented as a set of ordered triples.

MST Single Link Algorithm:

$d = 0$;

$k = n$;

$K = \{\{t_1\}, ..., \{t_n\}\}$;

$DE =< d, k, K >$; // Initially dendrogram contains each element in its own cluster.

$M = MST(A)$;

repeat

$oldk = k$;

$K_i, K_j =$ **two clusters closest together in MST;**

$K = K - \{K_i\} - \{K_j\} \cup \{K_i \cup K_j\}$;

$k = oldk - 1$;

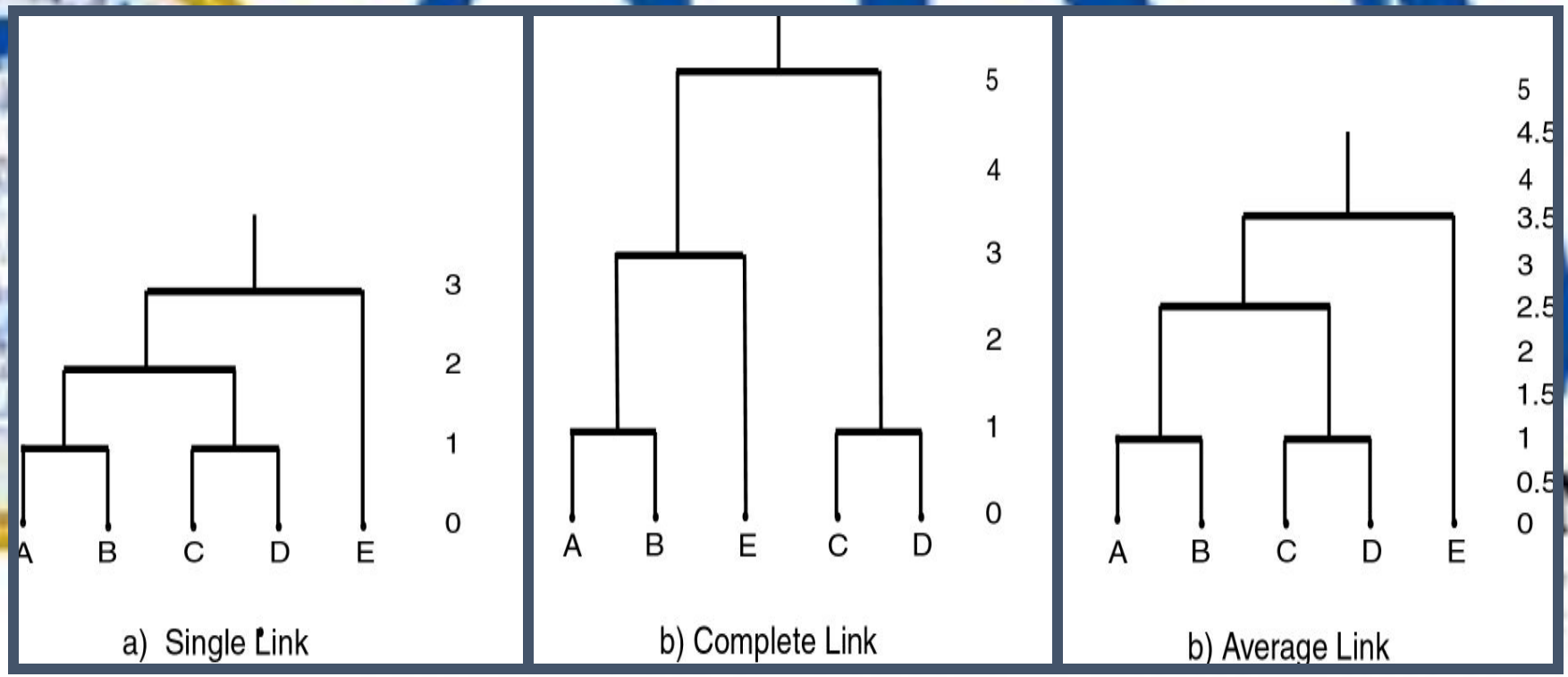
$d = dis(K_i, K_j)$;

$DE = DE \cup < d, k, K >$; // New set of clusters added to dendrogram.

$dis(K_i, K_j) = \infty$;

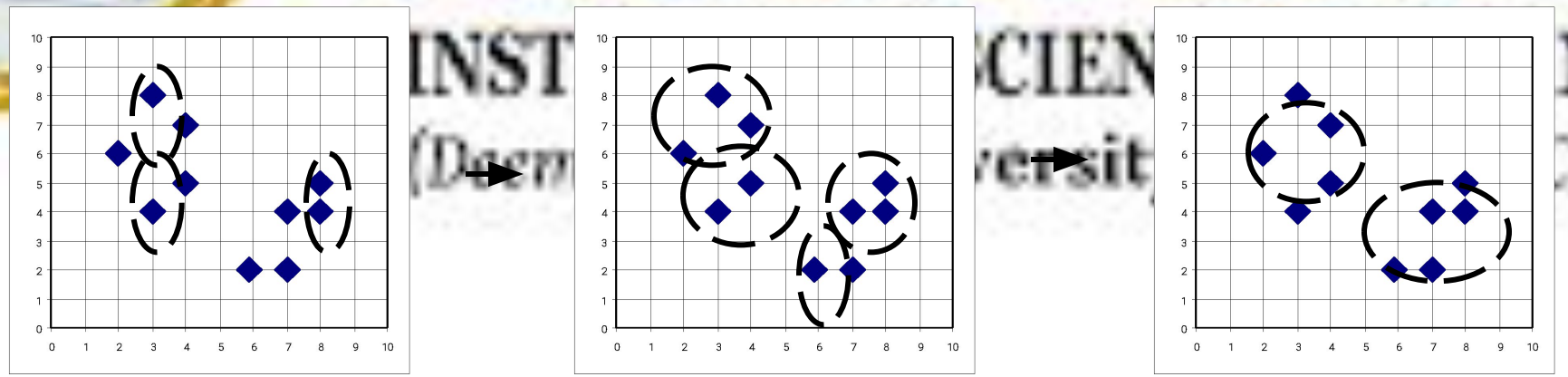
until $k = 1$

Single Link Clustering



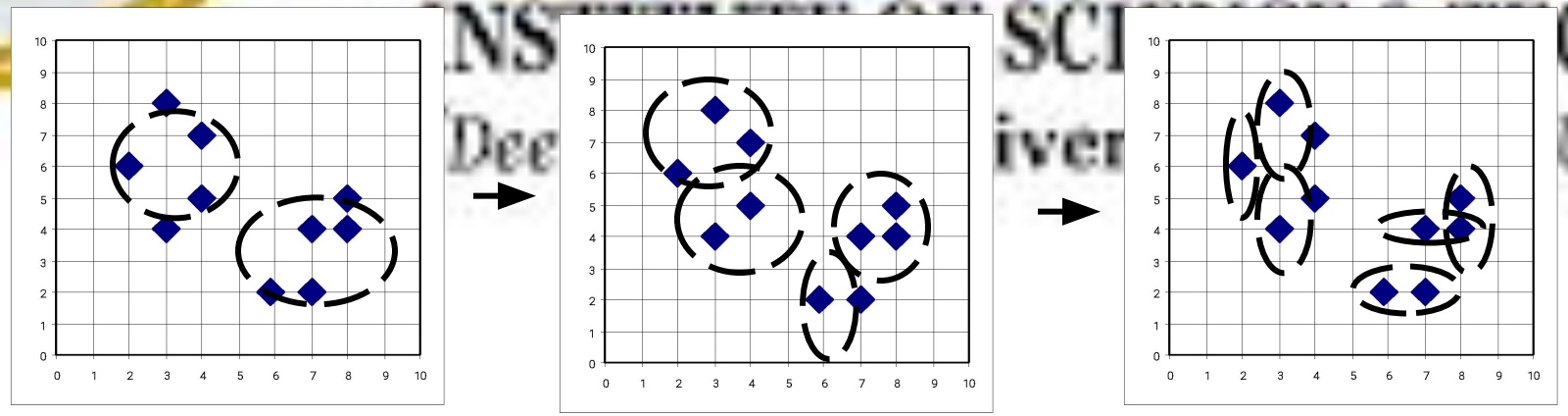
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

Session 5

BIRCH

Multiphase Hierarchical Clustering Using Clustering Feature Trees

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is designed for

- clustering a large amount of numeric data by integrating hierarchical clustering (at the initial *microclustering* stage) and other clustering methods such as iterative partitioning (at the later *macroclustering* stage).
- It overcomes the two difficulties in agglomerative clustering methods:
- (1) scalability and (2) the inability to undo what was done in theprevious step.

- BIRCH uses the notions of *clustering feature* to summarize a cluster, and
- *Clustering feature tree* (*CF-tree*) to represent a cluster hierarchy.
- Given a limited amount of main memory, an important consideration in BIRCH is to minimize the time required for input/output (I/O).
- BIRCH applies a *multiphase* clustering technique,
  - A single scan of the data set yields a basic, good clustering, and
  - one or more additional scans can optionally be used to further improve the quality

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
- **Phase 1:** BIRCH scans the database to build an initial in-memory CF-tree, whichcan be viewed as a multilevel compression of the data that tries to preserve the data's
- inherent clustering structure.
- **Phase 2:** BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of
- the CF-tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

Clustering Feature (CF): ***CF = (N, LS, SS)***

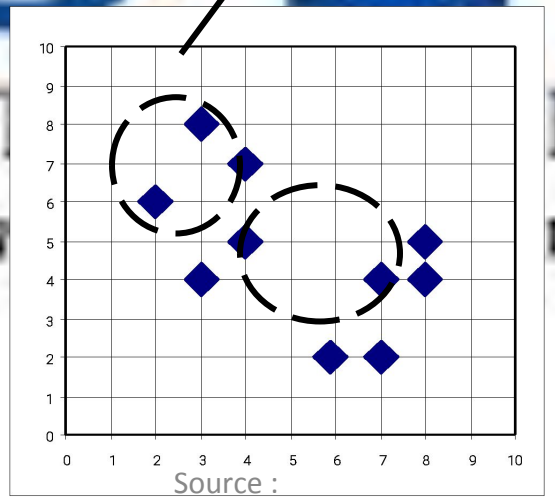
***N*: Number of data points**

LS: linear sum of N points: $\sum_{i=1}^{N} X_i$

SS: square sum of N points

$\sum_{i=1}^{N} X_i^2$

CF = (5, (16,30),(54,190))



(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

CF-Tree in BIRCH

- Clustering feature:
  - Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
  - Registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or “children”
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: max # of children
  - Threshold: max diameter of sub-clusters stored at the leaf nodes
- The branching factor specifies
- the maximum number of children per nonleaf node.
- The threshold parameter specifies

the maximum diameter of subclusters stored at the leaf nodes of the tree.

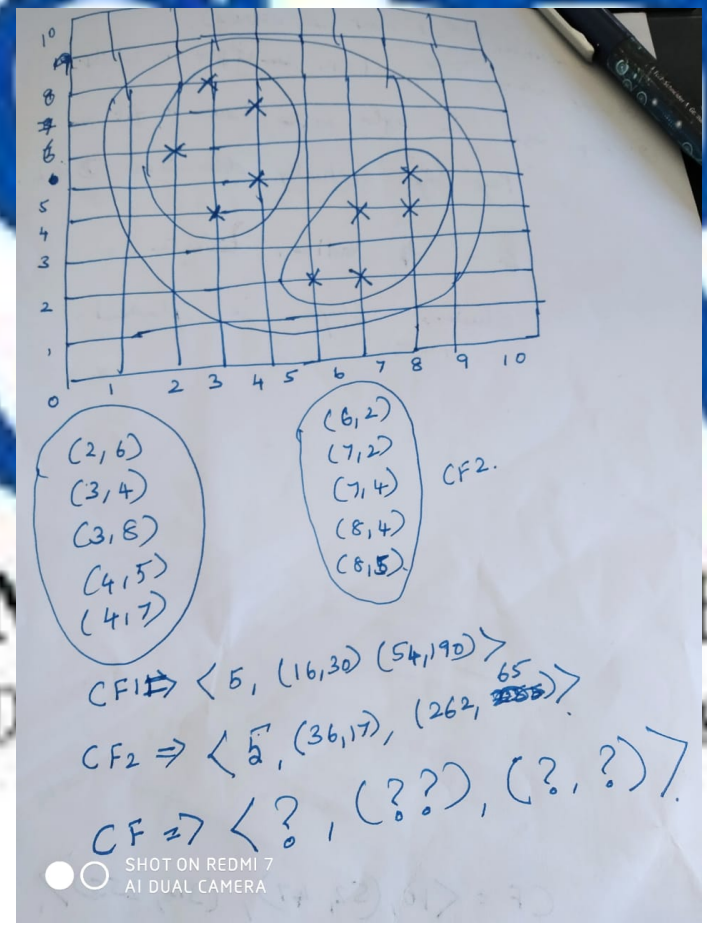
These two parameters implicitly control the resulting tree’s size.

Clustering feature. Suppose there are three points, $(2,5), (3,2)$, and $(4,3)$, in a cluster, C_1. The clustering feature of C_1 is

$$CF_1 = \langle 3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2) \rangle = \langle 3, (9,10), (29,38) \rangle.$$

Suppose that C_1 is disjoint to a second cluster, C_2, where $CF_2 = \langle 3, (35,36), (417,440) \rangle$. The clustering feature of a new cluster, C_3, that is formed by merging C_1 and C_2, is derived by adding CF_1 and CF_2. That is,

$$CF_3 = \langle 3+3, (9+35, 10+36), (29+417, 38+440) \rangle = \langle 6, (44,46), (446,478) \rangle. \blacksquare$$

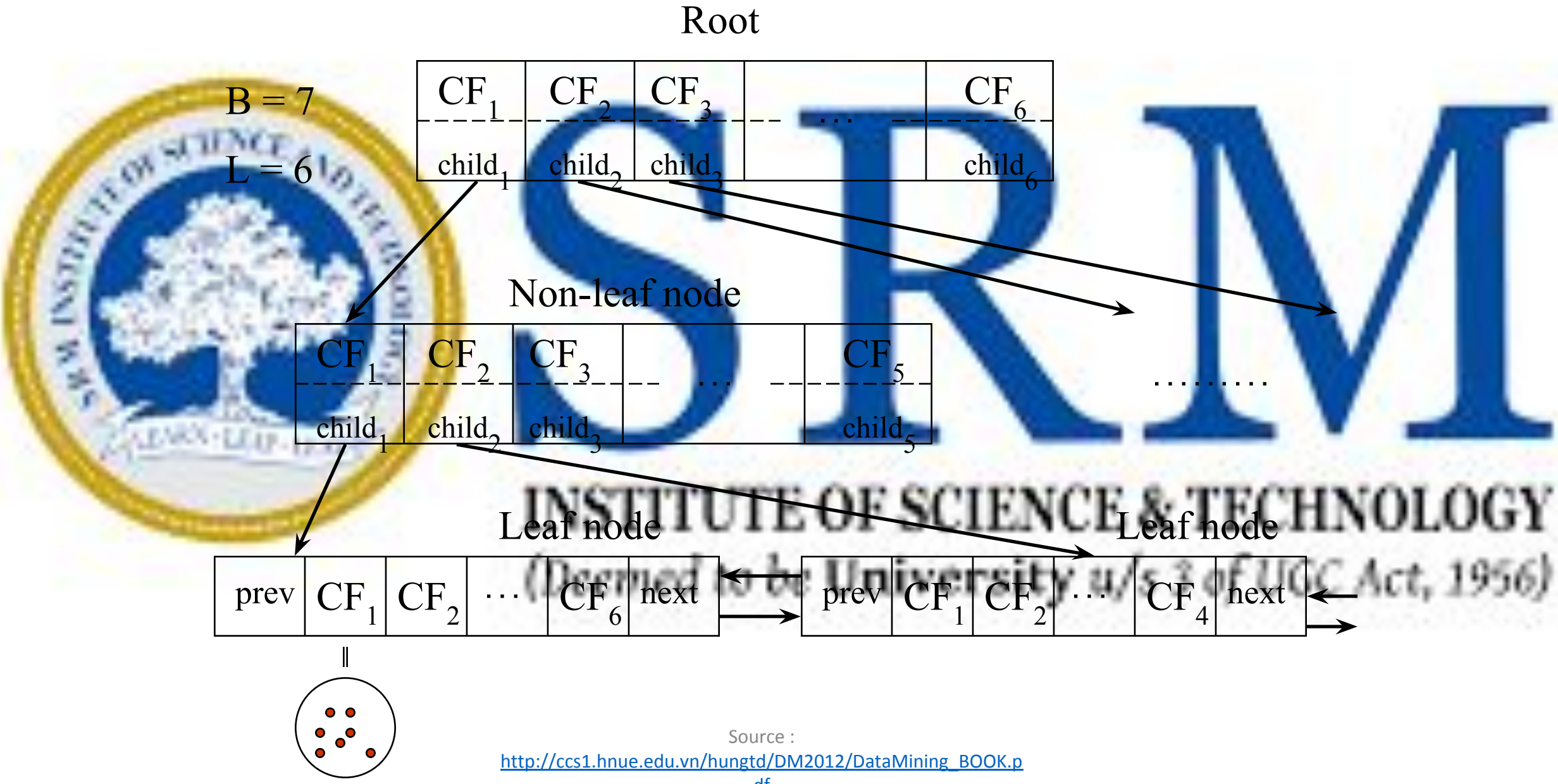


SRM

IN ENCE & TECHNOLOGY

(D sity u/s 3 of UGC Act, 1956)

The CF Tree Structure

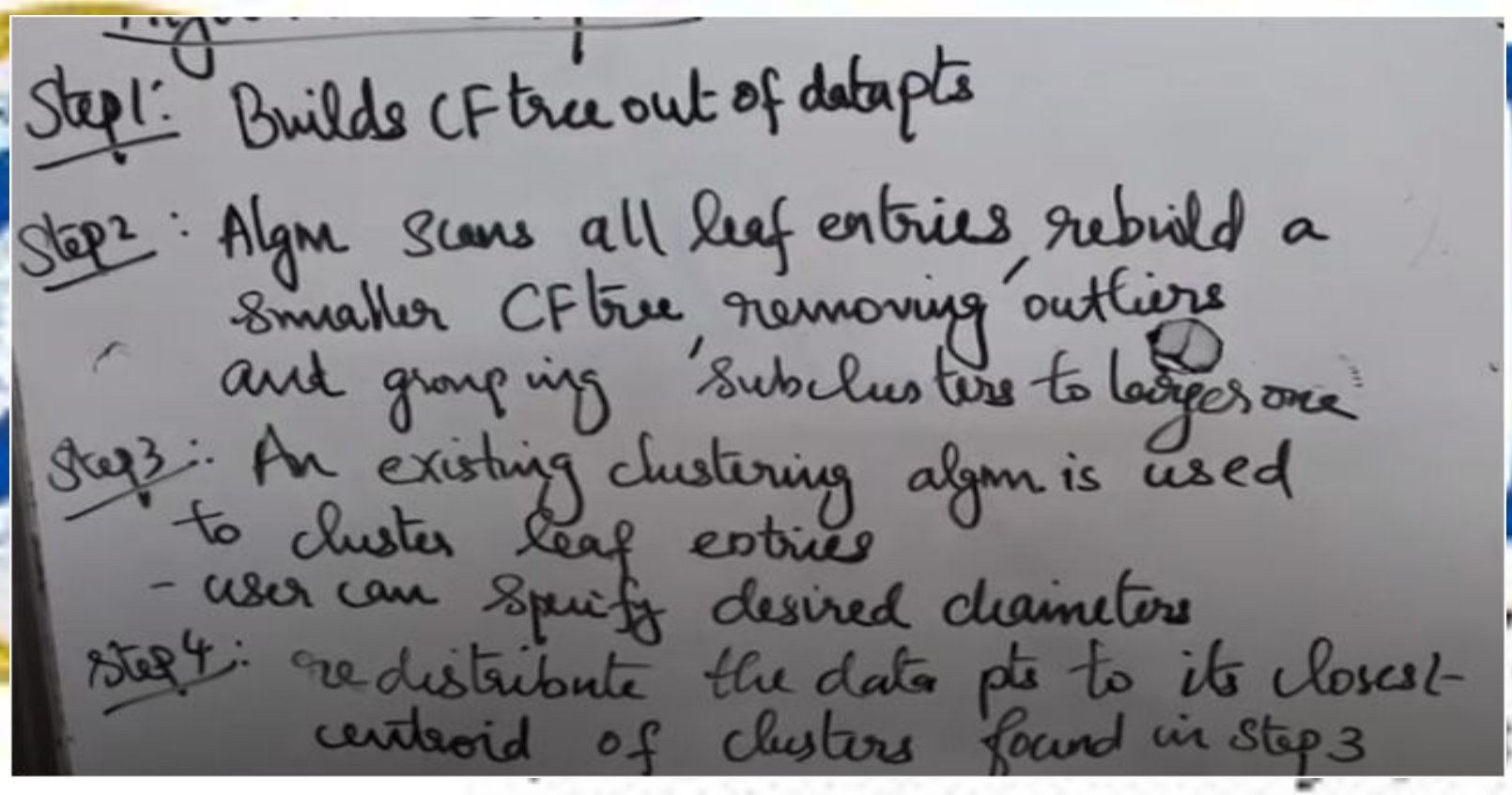


Source :
http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

The Birch Algorithm

- Cluster Diameter $\sqrt{\frac{1}{n(n-1)}\sum(x_i - x_j)^2}$
- For each point in the input
  - Find closest leaf entry
  - Add point to leaf entry and update CF
  - If entry diameter > max_diameter, then split leaf, and possibly parents
- Algorithm is O(n)
- Concerns
  - Sensitive to insertion order of data points
  - Since we fix the size of leaf nodes, so clusters may not be so natural
  - Clusters tend to be spherical given the radius and diameter measures

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

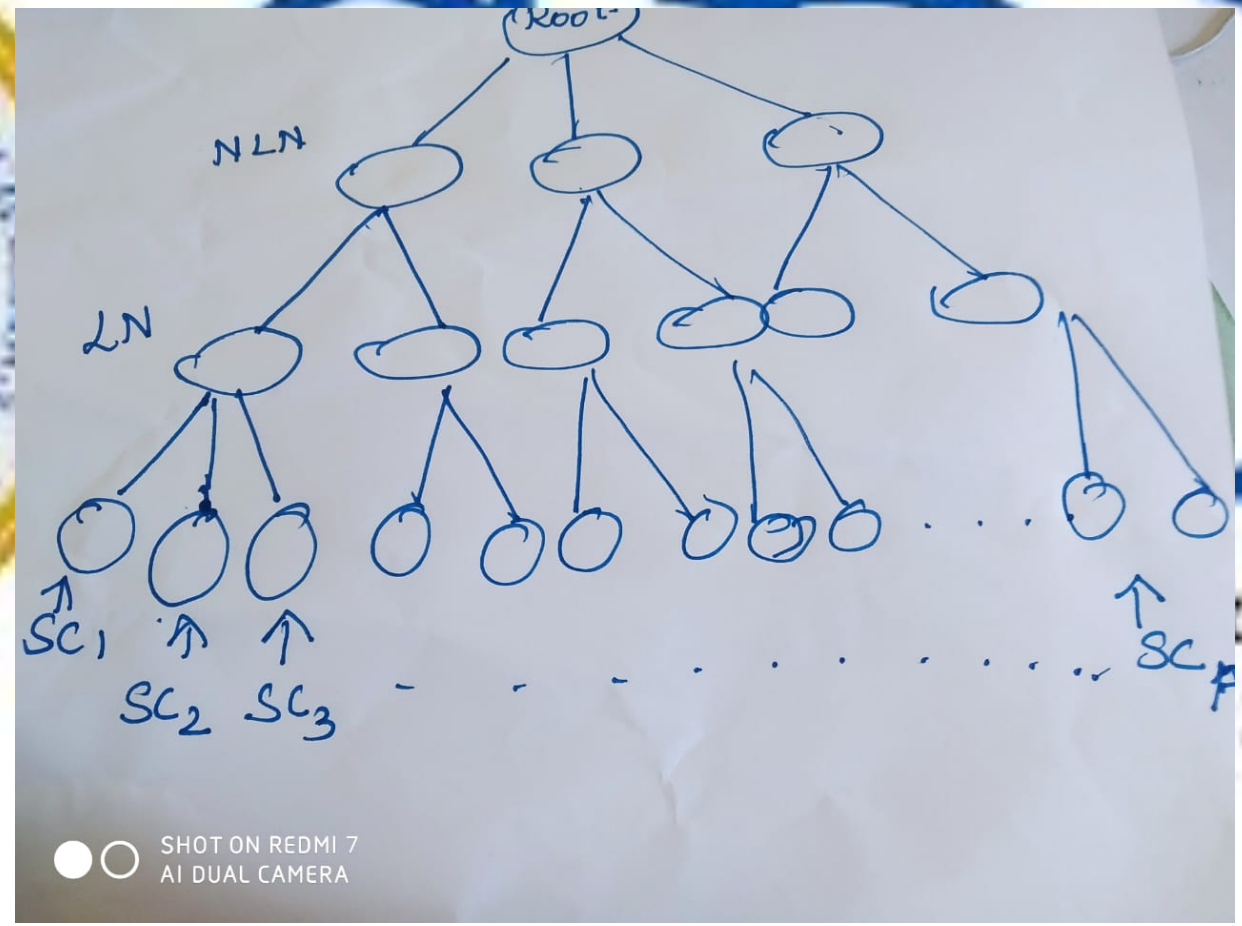


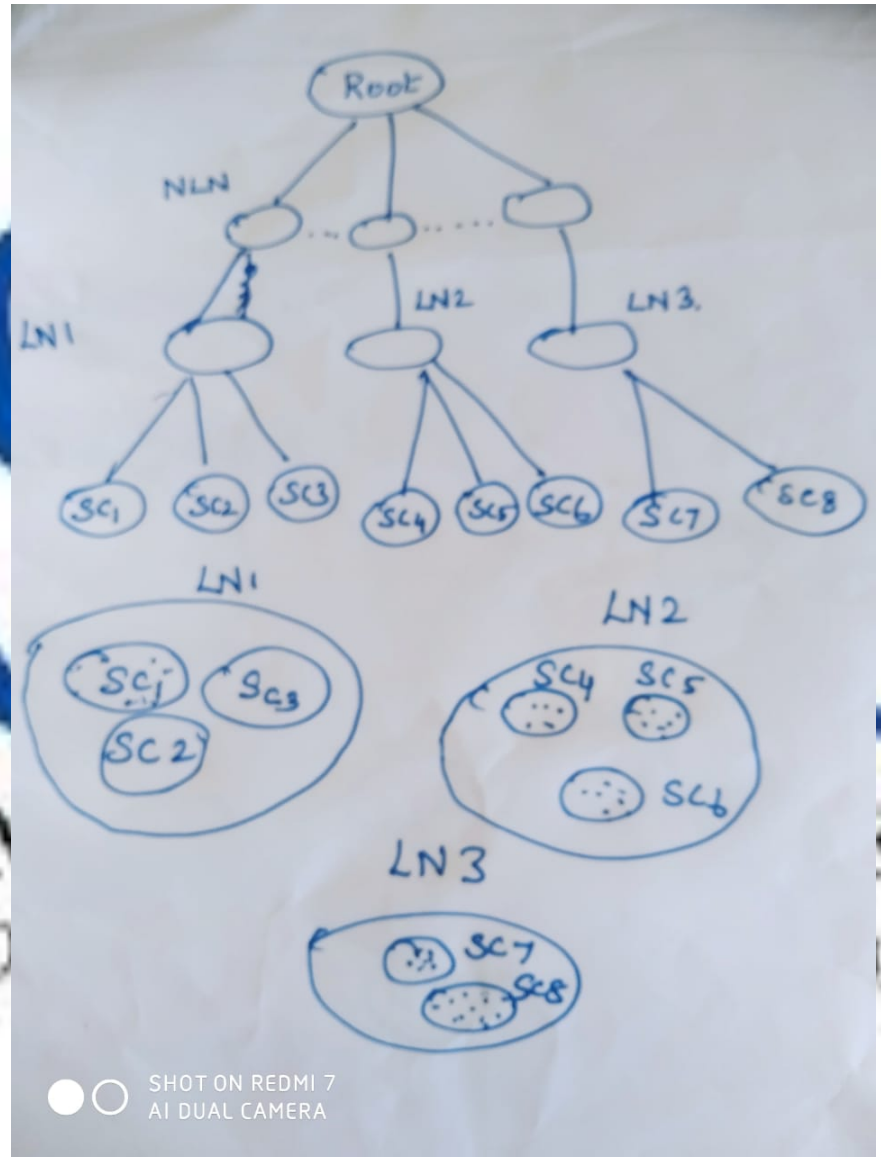
Step 1: Builds CF tree out of datapts

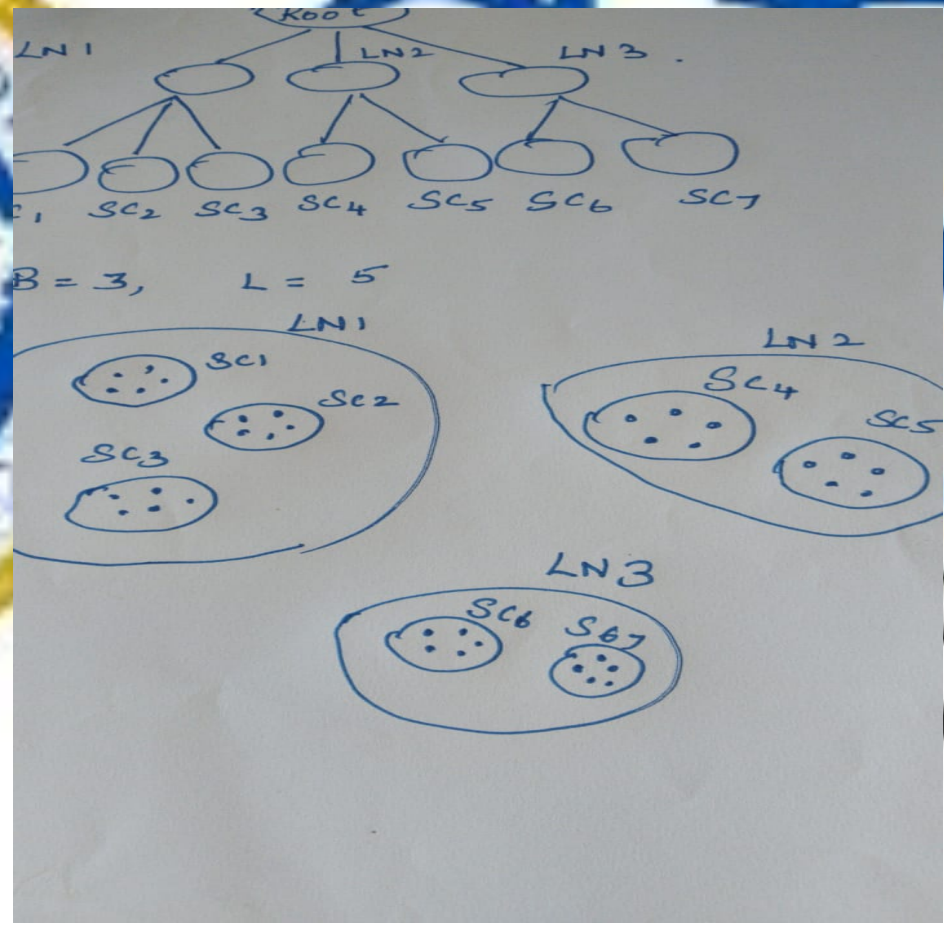
Step 2: Algm scans all leaf entries, rebuild a smaller CF tree, removing outliers and grouping 'subclusters to larger ones

Step 3: An existing clustering algm is used to cluster leaf entries
- user can specify desired diameters

Step 4: redistribute the data pts to its closest centroid of clusters found in step 3


NLN
LN
SC1
SC2
SC3
SC
SHOT ON REDMI 7
AI DUAL CAMERA


Root
NLN
LN1
LN2
LN3.
SC1
SC2
SC3
SC4
SC5
SC6
SC7
SC8
LN1
SC1
SC3
SC2
LN2
SC4
SC5
SC6
LN3
SC7
SC8
SHOT ON REDMI 7
AI DUAL CAMERA



CIENCE & TECHNOLOGY

ersity u/s 3 of UGC Act, 1956)

Partitioning and hierarchical methods are designed to find spherical-shaped clusters.

Session 6

DBSCAN

Density-based clusters are dense areas in the data space separated from each other by sparser areas.

Clusters of arbitrary shape.

Given such data, portioning and hierarchical would likely inaccurately identify convex regions, where noise or outliers are included in the clusters.

- To find clusters of arbitrary shape, alternatively, we may model clusters as dense
- regions in the data space, separated by sparse regions.
  This is the main strategy behind *density-based clustering methods*, which can discover clusters of nonspherical shape.

- *"How can we find dense regions in density-based clustering?"*
- The *density* of an object ***o*** can be measured by the number of objects close to ***o***.
- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) finds *core objects*, that is, objects that have dense neighborhoods.
- It connects core objects and their neighborhoods to form dense regions as clusters.

- *"How does* **DBSCAN** *quantify the neighborhood of an object?"*

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

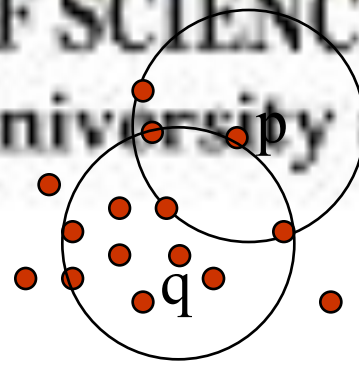
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
  - *Eps*: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
  - $N_{Eps}(p)$: {q belongs to D | dist(p,q) ≤ Eps}
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:

$$|N_{Eps}(q)| \geq MinPts$$



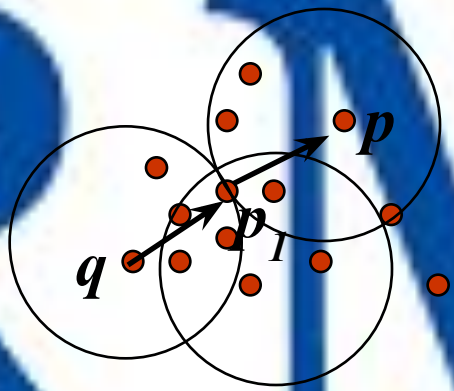
MinPts = 5

Eps = 1 cm

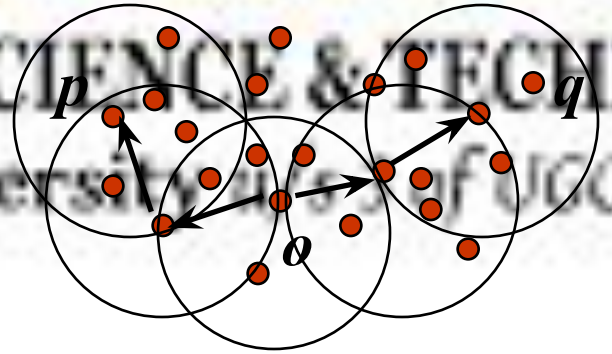
Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Density-Reachable and Density-Connected

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i



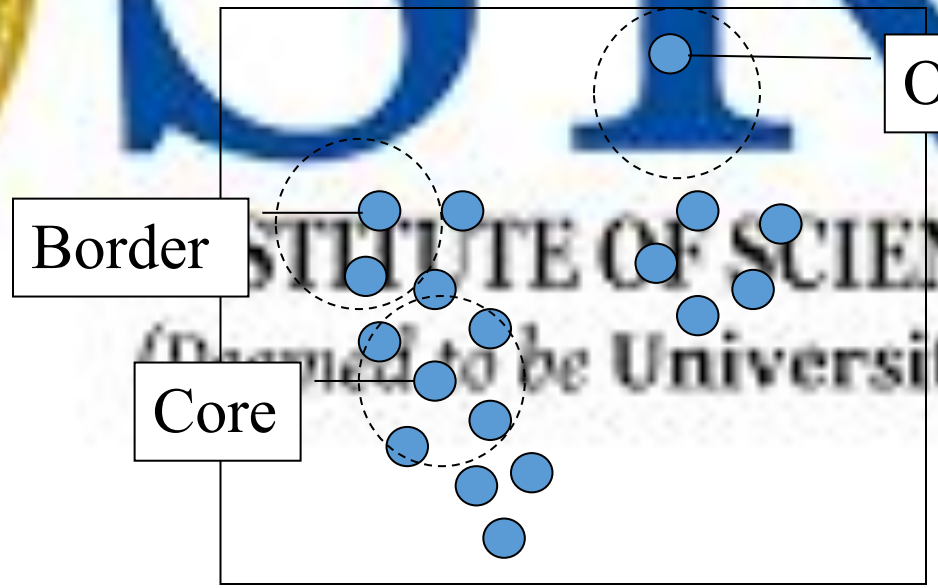
- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

DBSCAN: The Algorithm

- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
- If *p* is a core point, a cluster is formed
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

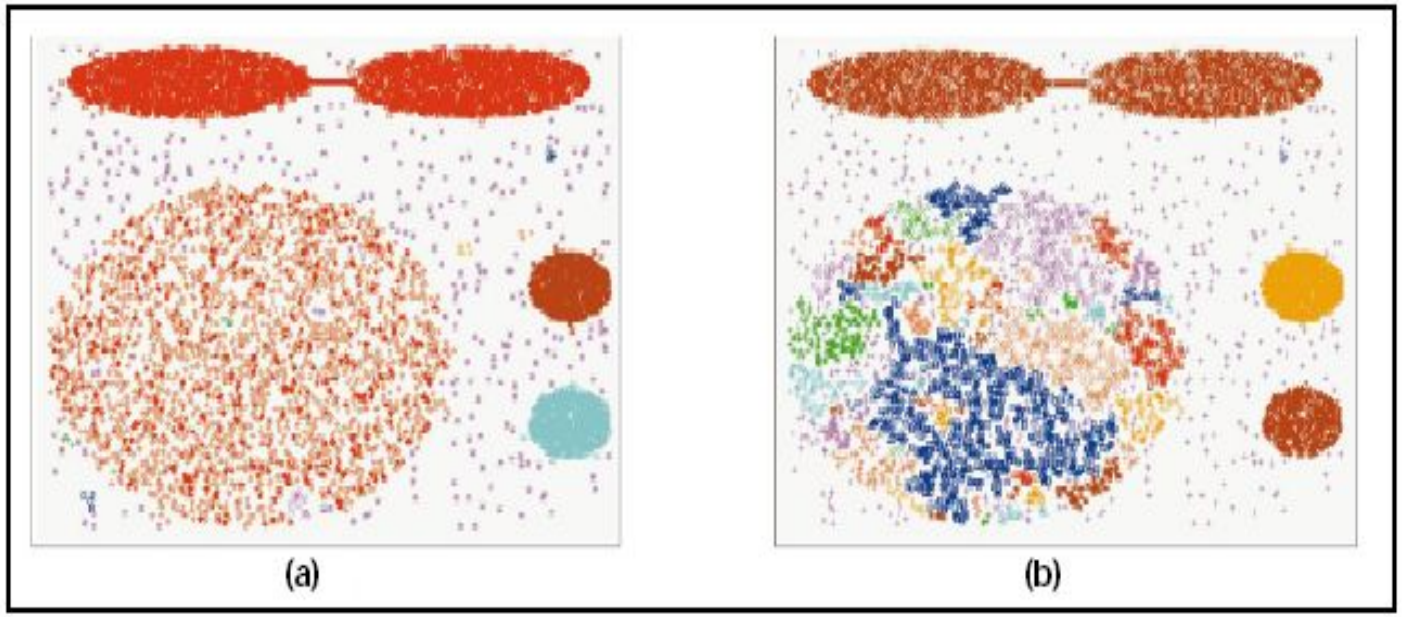
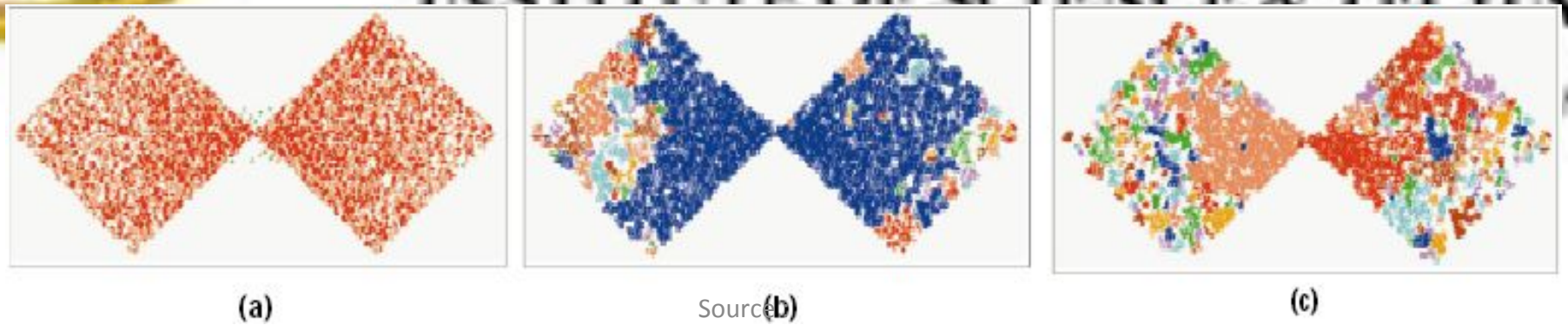


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



Source: http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Session 7

STING

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

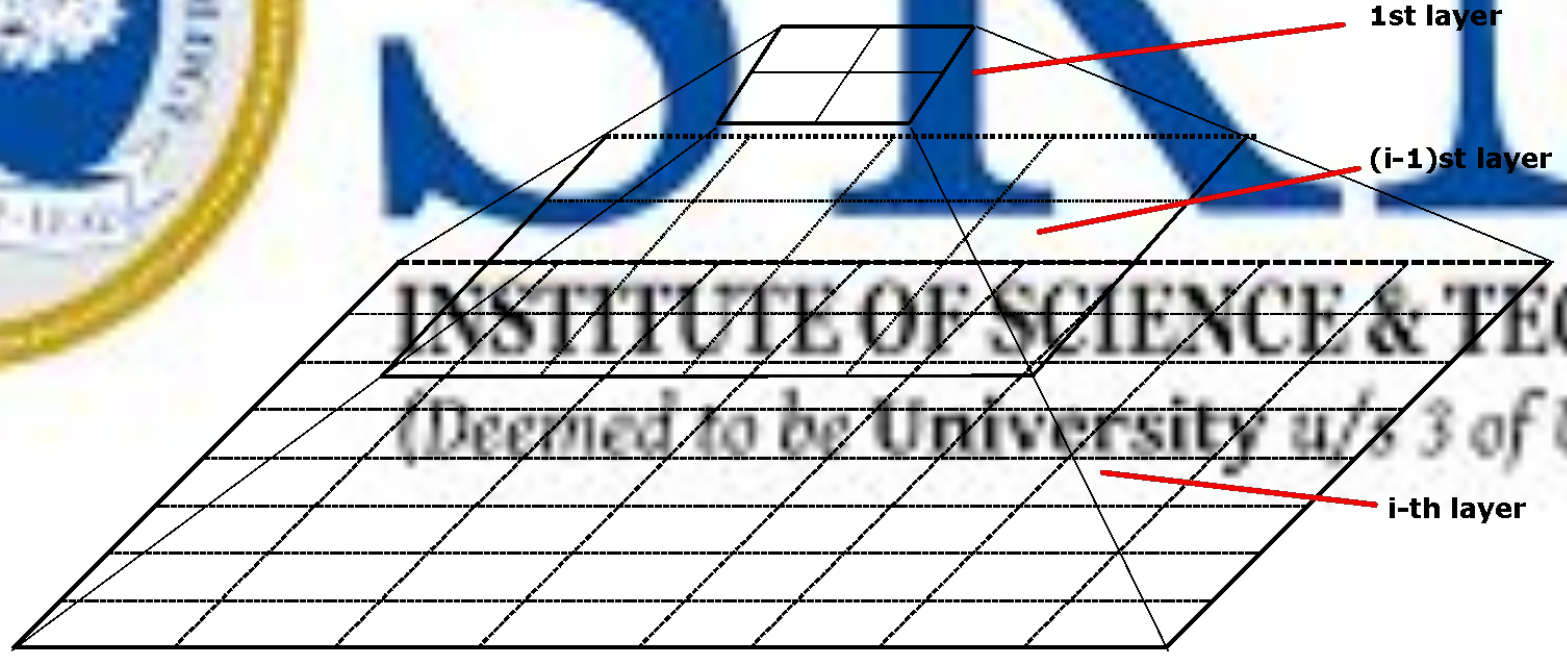
(Deemed to be University u/s 3 of UGC Act, 1956)

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD'98)
    - Both grid-based and subspace clustering

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

The STING Clustering Method

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—*normal*, *uniform*, etc.
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

STING Algorithm and Its Analysis

- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O(K),* where *K* is the number of grid cells at the lowest level
- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

Session 8

CLIQUE

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

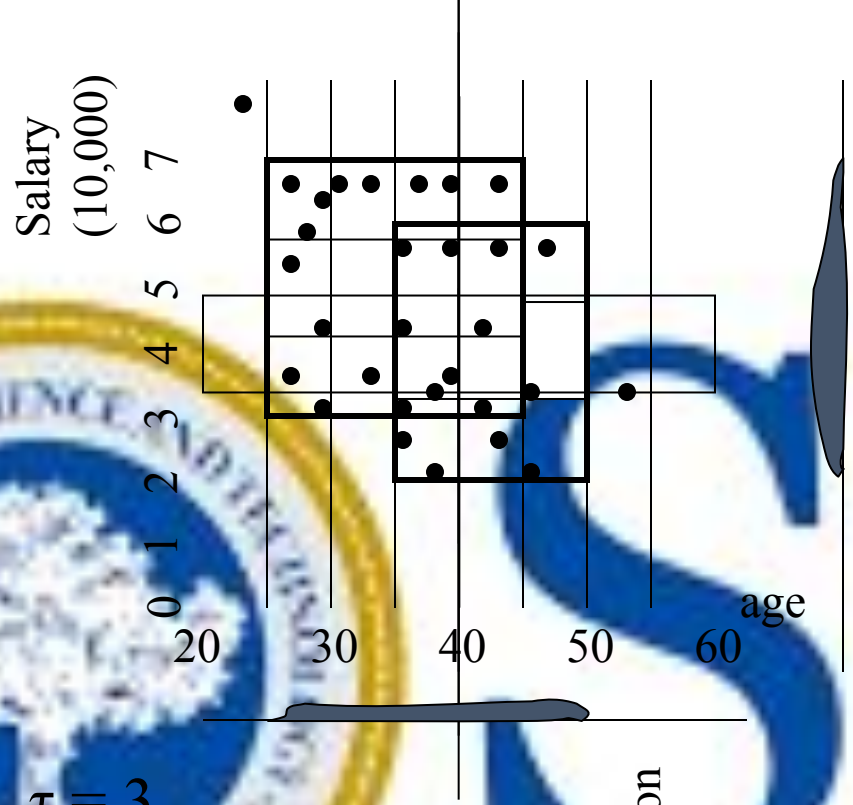
(Deemed to be University u/s 3 of UGC Act, 1956)

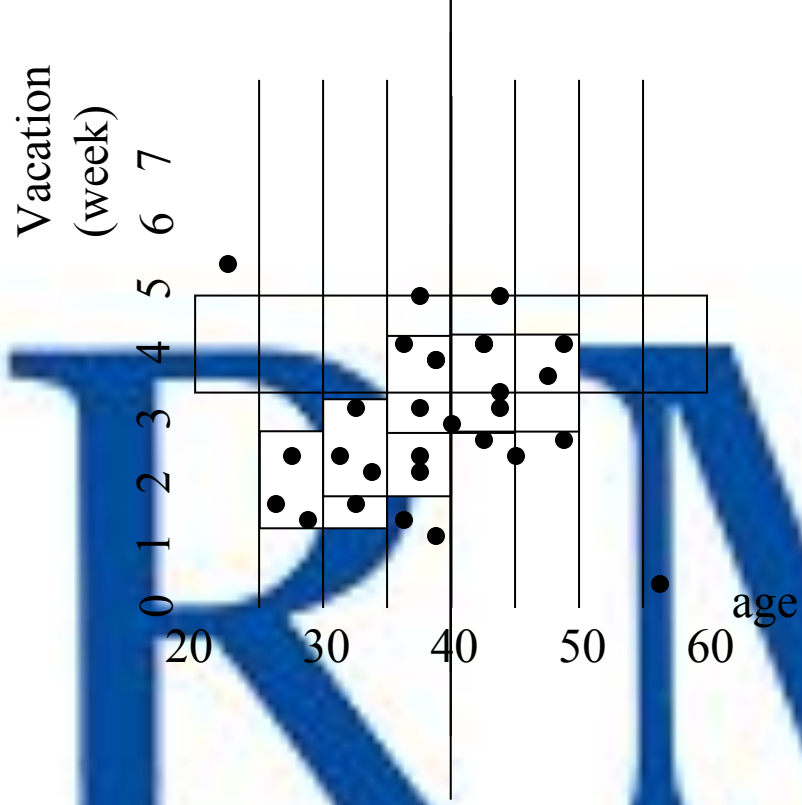
CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length interval
  - It partitions an m-dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace

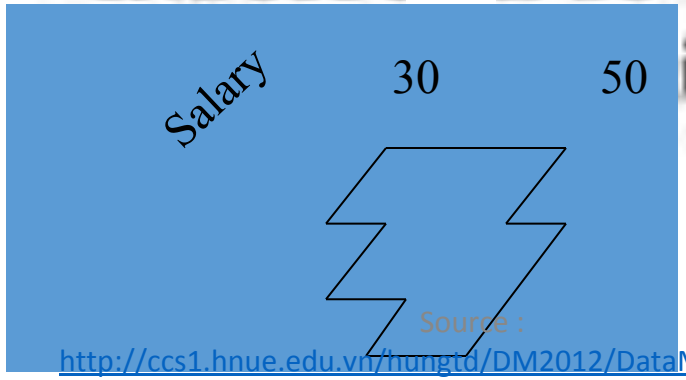
CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster





$\tau = 3$



Source :
http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Strength and Weakness of *CLIQUE*

- Strength
  - *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - *insensitive* to the order of records in input and does not presume some canonical data distribution
  - scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Session 9

Evaluation of Clustering Techniques

Assessing Clustering Tendency

- Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Test spatial randomness by statistic test: Hopkins Static
  - Given a dataset D regarded as a sample of a random variable o, determine how far away o is from being uniformly distributed in the data space
  - Sample n points, $p_1, \ldots, p_n$, uniformly from D. For each p_i, find its nearest neighbor in D: $x_i = min\{dist(p_i, v)\}$ where v in D
  - Sample n points, $q_1, \ldots, q_n$, uniformly from D. For each q_i, find its nearest neighbor in $D - \{q_i\}$: $y_i = min\{dist(q_i, v)\}$ where v in D and $v \neq q$
  - Calculate the Hopkins Statistic:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

  - If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5. If D is highly skewed, H is close to 0

Source : http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

Determine the Number of Clusters

- Empirical method
  - # of clusters ≈√n/2 for a dataset of n points
- Elbow method
  - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- Cross validation method
  - Divide a given data set into *m* parts
  - Use *m* – 1 parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
- For any k > 0, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best

Measuring Clustering Quality

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
  - Compare a clustering against the ground truth using certain clustering quality measure
  - Ex. BCubed precision and recall metrics
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
  - Ex. Silhouette coefficient

Measuring Clustering Quality: Extrinsic Methods

- Clustering quality measure: $Q(C, C_g)$, for a clustering C given the ground truth C_g.
- Q is good if it satisfies the following **4** essential criteria
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., "miscellaneous" or "other" category)
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

References

- *Jiawei Han and Micheline Kamber, " Data Mining: Concepts and Techniques", 3rd Edition, Morgan Kauffman Publishers, 2011.*
- http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)