

# CLUSTER ANALYSIS

# CLUSTER ANALYSIS

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# GENERAL APPLICATIONS OF CLUSTERING

- Pattern Recognition
- Spatial Data Analysis
  - Create thematic maps in GIS by clustering feature spaces
  - Detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

# EXAMPLES OF CLUSTERING APPLICATIONS

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- 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

# WHAT IS GOOD CLUSTERING?

- A good clustering method will produce high quality clusters with
  - High intra-class similarity
  - Low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

# REQUIREMENTS OF CLUSTERING IN DATA MINING

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

## CONT...

### i) Scalability:

- Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions of objects.
- Clustering on a sample of a given large data set may lead to biased results.

## CONT...

### ii) Ability to deal with different types of attributes

- Many algorithms are designed to cluster interval-based (numerical) data.
- However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.



## CONT...

### iii) Discovery of clusters with arbitrary shape:

- Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures.
- Algorithms based on such distance measures tend to find spherical clusters with similar size and density.

## CONT...

- iv) Minimal requirements for domain knowledge to determine input parameters:
- Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters).
  - The clustering results can be quite sensitive to input parameters.
  - Parameters are often difficult to determine especially for data sets containing high-dimensional objects.

## CONT...

- v) Ability to dealwith noisy data:
  - Most real-world databases contain outliers or missing, unknown, or erroneous data.
  - Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

## CONT...

- vi) Incremental clustering and insensitivity to the order of input records:
  - Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch.
  - Some clustering algorithms are sensitive to the order of input data.

## CONT...

### vii) High dimensionality:

- A database or a data warehouse can contain several dimensions or attributes.
- Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions.
- Human eyes are good at judging the quality of clustering for up to three dimensions.
- Finding clusters of data objects in highdimensional space is challenging, especially considering that such data can be sparse and highly skewed.

# CONT...

## viii) Constraint-based clustering:

- Real-world applications may need to perform clustering under various kinds of constraints.
- Suppose that your job is to choose the locations for a given number of new automatic banking machines (ATMs) in a city.
- To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks, and the type and number of customers per cluster.
- A challenging task is to find groups of data with good clustering behavior that satisfy specified constraints.

## CONT...

### ix) Interpretability and usability:

- Users expect clustering results to be interpretable, comprehensible, and usable.
- That is, clustering may need to be tied to specific semantic interpretations and applications.
- It is important to study how an application goal may influence the selection of clustering features and methods.

# CLUSTER ANALYSIS

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
  - Partitioning Methods
  - Hierarchical Methods
  - Density-Based Methods
  - Grid-Based Methods
  - Model-Based Clustering Methods
- Outlier Analysis
- Summary



# DATA STRUCTURES

- Data matrix
  - (two modes)

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

- Dissimilarity matrix
  - (one mode)

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

# MEASURE THE QUALITY OF CLUSTERING

June 19, 2020

- **Dissimilarity/Similarity metric** : Similarity is expressed in terms of a distance function, which is typically metric:  $d(i, j)$
- There is a separate “quality” function that measures the “goodness” of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, Boolean, Categorical, Ordinal and Ratio variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
  - the answer is typically highly subjective.

# TYPE OF DATA IN CLUSTERING ANALYSIS

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

# INTERVAL-VALUED VARIABLES

- Standardize data

- Calculate the mean absolute deviation:

$$s_f = \frac{1}{n} (|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

where

$$m_f = \frac{1}{n} (x_{1f} + x_{2f} + \dots + x_{nf}).$$

- Calculate the standardized measurement (*z-score*)

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

- Using mean absolute deviation is more robust than using standard deviation

# SIMILARITY AND DISSIMILARITY BETWEEN OBJECTS

June 19, 2020

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

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

where  $i = (x_{i1}, x_{i2}, \dots, x_{ip})$  and  $j = (x_{j1}, x_{j2}, \dots, x_{jp})$  are two  $p$ -dimensional data objects, and  $q$  is a positive integer

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

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

# SIMILARITY AND DISSIMILARITY BETWEEN OBJECTS (CONT.)

June 19, 2020

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

- Properties

- $d(i,j) \geq 0$
- $d(i,i) = 0$
- $d(i,j) = d(j,i)$
- $d(i,j) \leq d(i,k) + d(k,j)$

$$d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

- Also one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures.

# BINARY VARIABLES

- A Binary variable has only two states (0 or 1)
  - 0 means variable is absent
  - 1 means variable is present
  - Ex: variable *smoker* (1 indicates patient smokes and 0 indicates patient does not)
  - Treating binary variables as interval-scaled can lead to misleading results
  - There may be symmetric and asymmetric binary variables
- To compute the dissimilarity between two binary variables
  - If all binary variables are thought of as having same weight, construct a 2-by-2 contingency table.

# BINARY VARIABLES (CONTD...)

- A contingency table for binary data

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

Where,  $p$  is total no of variable and  $p = a + b + c + d$

- A binary variable is **symmetric** if both of states are **equally valuable and carry the same weight**, no preference on the outcome should be coded as 0 or 1.
- Ex: **gender** (male or female)
- **Dissimilarity** based on symmetric binary variable is called **symmetric binary dissimilarity**

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



# BINARY VARIABLES (CONTD...)

June 19, 2020

- A binary variable is **asymmetric** if the states are not **equally important**.
- Ex: **test** (positive or negative)
- By convention, we shall code the most important outcome, which is usually the rarest one by **1 (e.g. HIV positive)** and the other by **0 (e.g. HIV negative)**
- Given two asymmetric variables, the agreement of two 1s (a positive match) is considered **more significant** than two 0s (a negative match)
- **Dissimilarity** based on asymmetric binary variable is called **asymmetric binary dissimilarity** where the no. of –ve matches ***d*** is considered unimportant and thus ignored in the computation.

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

# BINARY VARIABLES (CONTD...)

June 19, 2020

- Complementarily , we can measure the distance between two binary variables based on notion of similarity instead of dissimilarity
- Jaccard coefficient ,

$$\text{sim}(i, j) = \frac{a}{a+b+c} = 1 - d(i, j)$$

# DISSIMILARITY BETWEEN BINARY VARIABLES

June 19, 2020

○

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

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

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

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

# NOMINAL VARIABLES (OR CATEGORICAL)

- A categorical variable (sometimes called a nominal variable) is one that has two or more categories.
- But there is no intrinsic ordering to the categories
- For example, gender is a categorical variable having two categories (male and female) and there is no intrinsic ordering to the categories
- Hair color is also a categorical variable having a number of categories (blonde, brown, brunette, red, etc.) and again, there is no agreed way to order these from highest to lowest.
- A purely categorical variable is one that simply allows you to assign categories but you cannot clearly order the variables. If the variable has a clear ordering, then that variable would be an ordinal variable

# NOMINAL VARIABLES (OR CATEGORICAL) CONTD...

June 19, 2020

- A generalization of the binary variable is that it can take more than 2 states, e.g., red, yellow, blue, green
- Let, no. of states of a categorical variable be M
- The states can be denoted as 1,2,...,M
- Method : Simple matching
  - $m$ : # of matches,  $p$ : total # of variables

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

# NOMINAL VARIABLES (OR CATEGORICAL) CONTD...

Example: Dissimilarity between categorical variables  
Suppose, we have object identifier and test-1 are the variables.

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

The dissimilarity matrix is :

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

Since, we have one categorical variable test-1, we set  $p=1$  in equation and  $d(i,j)$  is 0 if objects  $i$  and  $j$  match, and 1 if differ.

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

# ORDINAL VARIABLES

- An ordinal variable is similar to a categorical variable.
- The difference between the two is that there is a **clear ordering of the variables**
- For example, suppose you have a variable, economic status, with three categories (low, medium and high).
- In addition to **being able to classify people** into these three categories, you can **order the categories** as low, medium and high
- Now consider a variable like educational experience (with values such as elementary school graduate, high school graduate, some college and college graduate).
- Even though we can order these from lowest to highest, the spacing between the values may not be the same across the levels of the variables.

# ORDINAL VARIABLES (CONTD...)

- Say we assign scores 1, 2, 3 and 4 to these four levels of educational experience and we compare the difference in education between categories one and two with the difference in educational experience between categories two and three, or the difference between categories three and four.
- The difference between categories one and two (elementary and high school) is probably much bigger than the difference between categories two and three (high school and some college).
- In this example, we can order the people in level of educational experience but the size of the difference between categories is inconsistent (because the spacing between categories one and two is bigger than categories two and three).



# HOW ARE ORDINAL VARIABLES HANDLED ?

June 19, 2020

- Quite similar as interval-scaled variable while computing the dissimilarity between objects
- Let  $f$  is a variable from a set of ordinal variables describing  $n$  objects
- The dissimilarity w.r.t  $f$  involves the following steps:
  - replacing  $x_{if}$  by their rank  $r_{if} \in \{1, \dots, M_f\}$
  - map the range of each variable onto  $[0, 1]$  by replacing  $i$ -th object in the  $f$ -th variable by 
$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$
  - compute the dissimilarity using methods for interval-scaled variables

# ORDINAL VARIABLES (EXAMPLE)

- Example: Dissimilarity between ordinal variables
- Suppose, we have object identifier and test-2 are the variables.
- There are 3 states for test-2 i.e  $M_f = 3$
- *Step1 : Replace each value of test-2 by its rank, i.e 3,1,2,3*
- *Step2: Normalize the ranking by mapping rank 1 to 0, rank 2 to 0.5 and rank 3 to 1.*
- *Step3: Use Euclidian distance to find the dissimilarity matrix*

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

# RATIO-SCALED VARIABLES

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as  $Ae^{Bt}$  or  $Ae^{-Bt}$
- Methods:
  - treat them like interval-scaled variables — *not a good choice! (since it is likely that the scale may be distorted)*
  - apply logarithmic transformation

$$y_{if} = \log(x_{if})$$

- treat them as continuous ordinal data treat their rank as interval-scaled.

# RATIO-SCALED VARIABLES (EXAMPLE)

- Ratio variables are those in which the ratio of two of the numbers have meaning, such as miles per gallon, for example. If car A gets 15 mpg and car B gets 20 mpg, you can take the ratio of the two:  $15/20$  and compute 0.75, meaning car A gets 75% of the mileage of car B.

# RATIO-SCALED VARIABLES (EXAMPLE)

June 19, 2020

- Example: Dissimilarity between ratio-scaled variables
- Suppose, we have object identifier and test-3 are the variables.
- *Step1 : Let us try logarithmic transformation( the results are 2.65 , 1.34, 2.21 and 3.08)*
- *Step3: Use **Euclidian distance** to find the **dissimilarity matrix***

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

# WHY DOES IT MATTER WHETHER A VARIABLE IS CATEGORICAL, ORDINAL OR INTERVAL?

- Statistical computations and analyses assume that the variables have a specific levels of measurement.
- For example, it would not make sense to compute an average hair color.
- An average of a categorical variable does not make much sense because there is no intrinsic ordering of the levels of the categories.
- Moreover, if you tried to compute the average of educational experience as defined in the ordinal section above, you would also obtain a nonsensical result.
- Because the spacing between the four levels of educational experience is very uneven, the meaning of this average would be very questionable.

# WHY DOES IT MATTER WHETHER A VARIABLE IS CATEGORICAL, ORDINAL OR INTERVAL?

- In short, an average requires a variable to be interval.
- Sometimes you have variables that are "in between" ordinal and interval, for example, a five-point scale with values "strongly agree", "agree", "neutral", "disagree" and "strongly disagree".
- If we cannot be sure that the intervals between each of these five values are the same, then we would not be able to say that this is an interval variable, but we would say that it is an ordinal variable.
- However, in order to be able to use statistics that assume the variable is interval, we will assume that the intervals are equally spaced.

# VARIABLES OF MIXED TYPES

June 19, 2020

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio.

- One may use a weighted formula to combine their effects.

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

- $f$  is binary or nominal:  
 $d_{ij}^{(f)} = 0$  if  $x_{if} = x_{jf}$ , or  $d_{ij}^{(f)} = 1$  o.w.
- $f$  is interval-based: use the normalized distance
- $f$  is ordinal or ratio-scaled
  - compute ranks  $r_{if}$  and
  - and treat  $z_{if}$  as interval-scaled

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



# CLUSTER ANALYSIS

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# MAJOR CLUSTERING APPROACHES

June 19, 2020

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

# CLUSTER ANALYSIS

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# PARTITIONING ALGORITHMS: BASIC CONCEPT

June 19, 2020

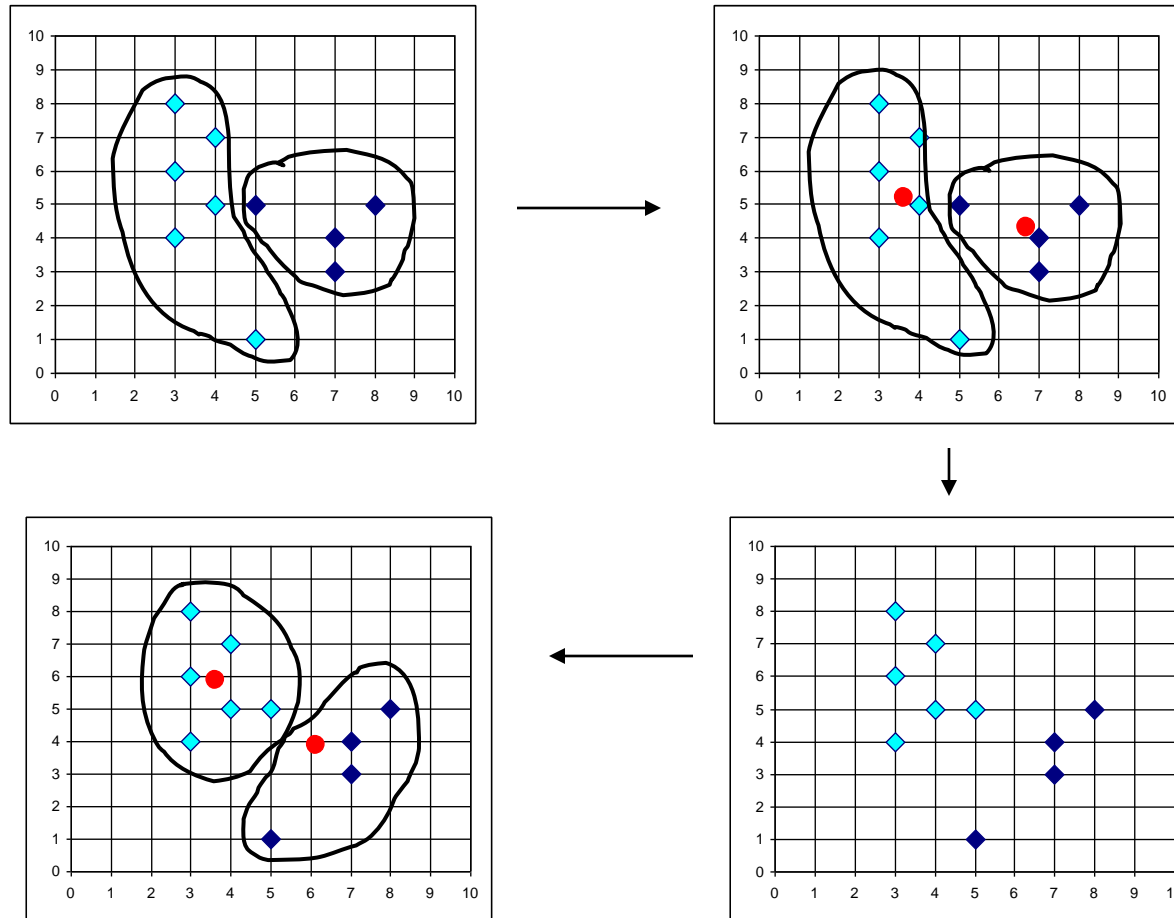
- Partitioning method: Construct a partition of a database  $D$  of  $n$  objects into a set of  $k$  clusters
- Given a  $k$ , find a partition of  $k$  clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): 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

# THE *K-MEANS* CLUSTERING METHOD

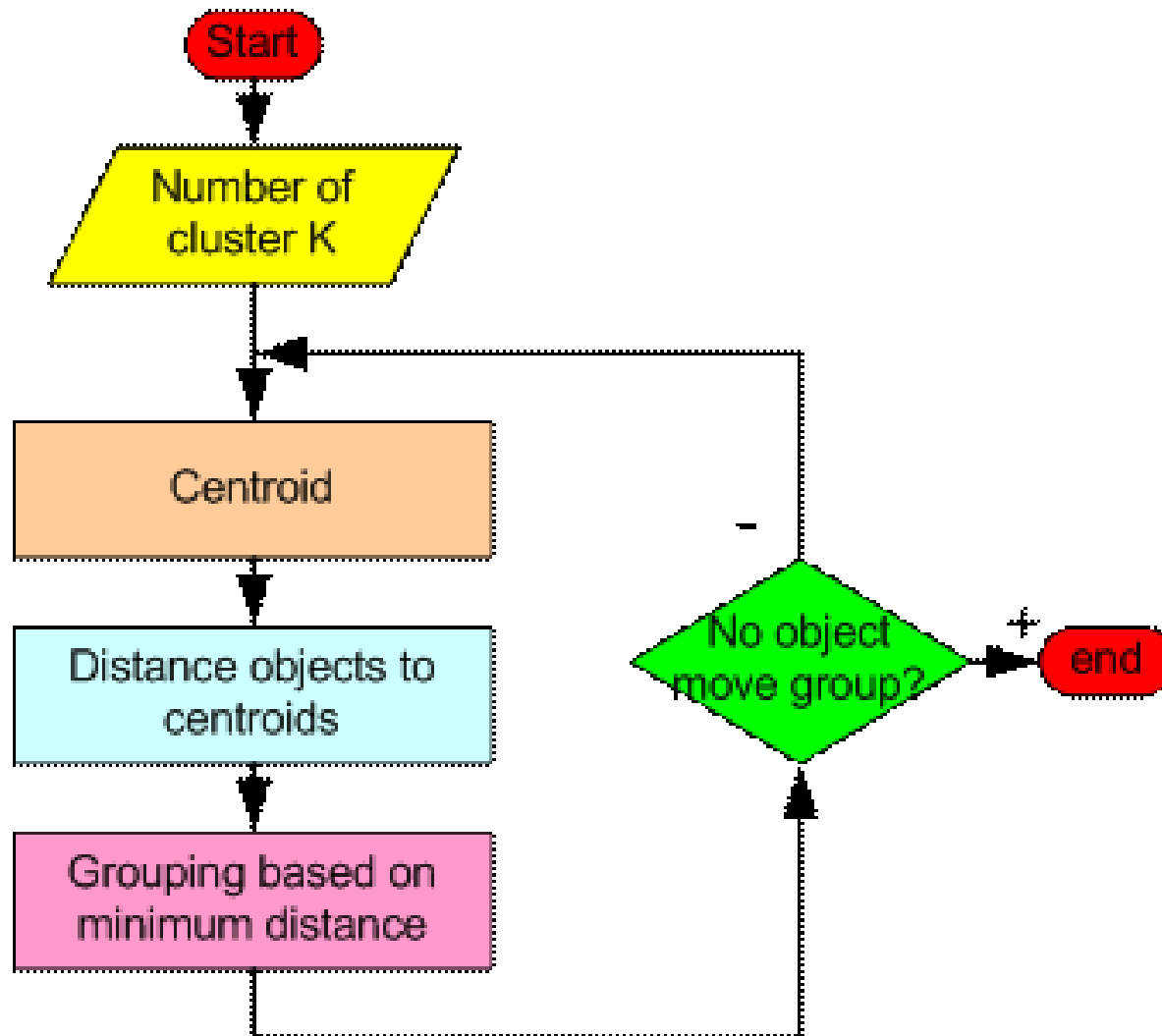
- Given  $k$ , the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into  $k$  nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  - Assign each object to the cluster with the nearest seed point.
  - Go back to Step 2, stop when no more new assignment.

# THE *K*-MEANS CLUSTERING METHOD

## ○ Example



# THE *K*-MEANS CLUSTERING FLOWCHART



# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

June 19, 2020

Suppose we have several objects (4 types of medicines) and each object have two attributes or features as shown in table below. Our goal is to group these objects into  $K=2$  group of medicine based on the two features (pH and weight index).

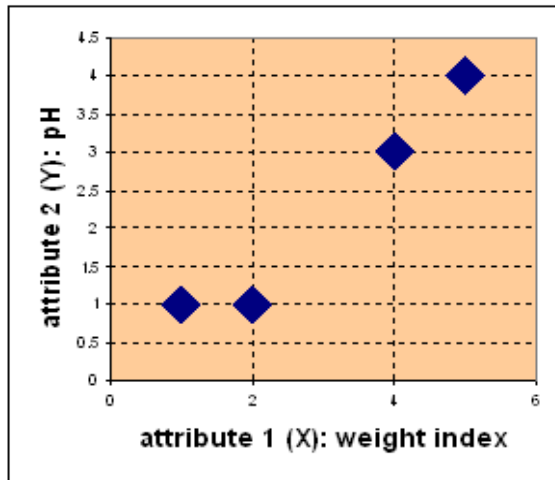
Object	attribute 1 (X): weight index	attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Each medicine represents one point with two attributes (X, Y) that we can represent it as coordinate in an attribute space as shown in the figure next slide.



# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

June 19, 2020

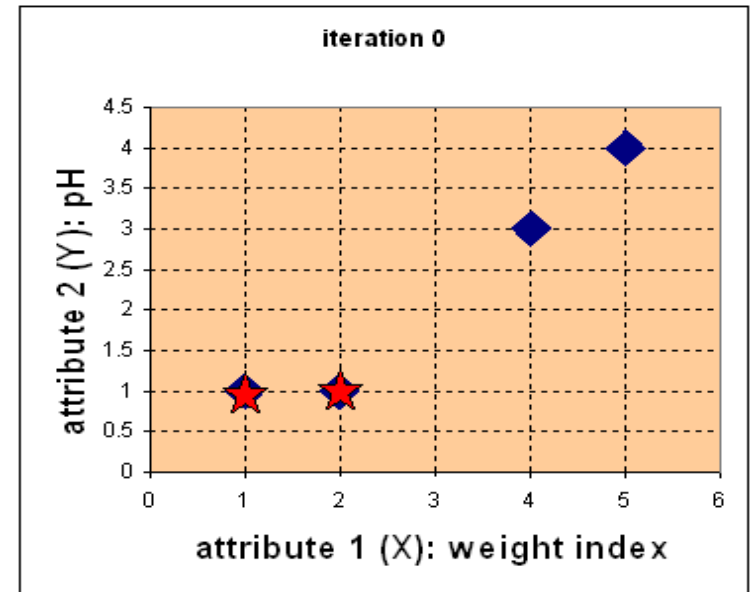


1. **Initial value of centroids** :  
 Suppose we use medicine A and medicine B as the first centroids. Let  $c_1$  and  $c_2$  denote the coordinate of the centroids, then  $c_1=(1,1)$  and  $c_2=(2,1)$

2. **Objects-Centroids distance** :  
 we calculate the distance between cluster centroid to each object. Let us use [Euclidean distance](#), then we have distance matrix at iteration 0 is

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{array}{l} c_1 = (1,1) \text{ group-1} \\ c_2 = (2,1) \text{ group-2} \end{array}$$

$A$	$B$	$C$	$D$	
$\begin{bmatrix} 1 & 2 & 4 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 3 & 4 \end{bmatrix}$			$X$
				$Y$



# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

June 19, 2020

- Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid.
- For example, distance from medicine  $C = (4, 3)$  to the first centroid  $c1=(1,1)$  is  $\sqrt{(4-1)^2 + (3-1)^2} = 3.61$
- and its distance to the second centroid  $c2= (2,1)$  is  $\sqrt{(4-2)^2 + (3-1)^2} = 2.83$

# THE *K-MEANS* CLUSTERING NUMERICAL EXAMPLE

3. **Objects clustering** : We assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group-1} \\ \text{group-2} \end{matrix}$$

A   B   C   D

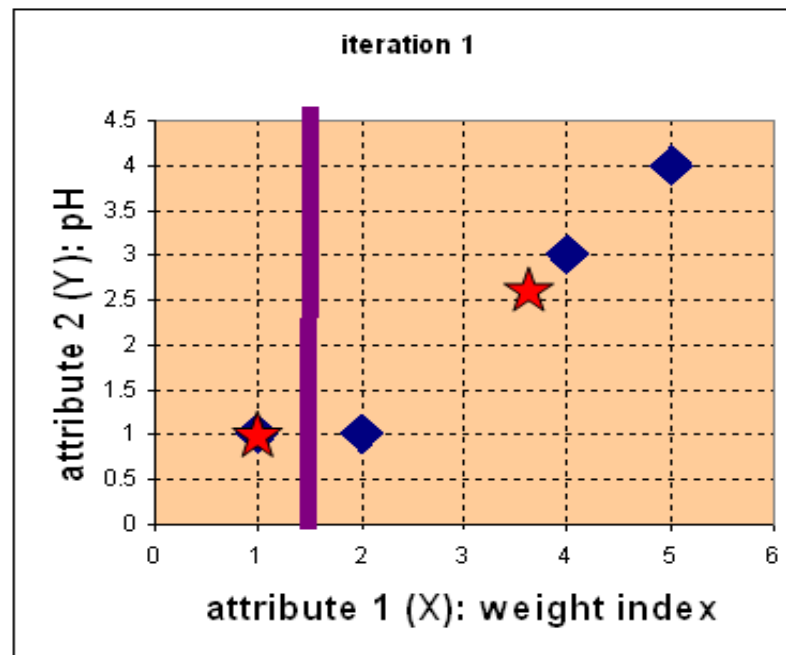
4. **Iteration-1, determine centroids** : Knowing the members of each group, now we compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in i.e  $c_1 = (1,1)$  and

# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

June 19, 2020

Group 2 now has three members, thus the centroid is the average coordinate among the three members:  $c_1=(1,1)$  and

$$c_2 = \left( \frac{2+4+5}{3}, \frac{1+3+4}{3} \right) = \left( \frac{11}{3}, \frac{8}{3} \right)$$



# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

June 19, 2020

**5. Iteration-1, Objects-Centroids distances** : The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

$$\mathbf{D}^1 = \begin{array}{cccc} \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} & \begin{array}{l} \mathbf{c}_1 = (1,1) \text{ group-1} \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) \text{ group-2} \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} & \\ \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} & \begin{array}{l} X \\ Y \end{array} \end{array}$$

# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

6. **Iteration-1, Objects clustering:** Similar to step 3, we assign each object based on the minimum distance. Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

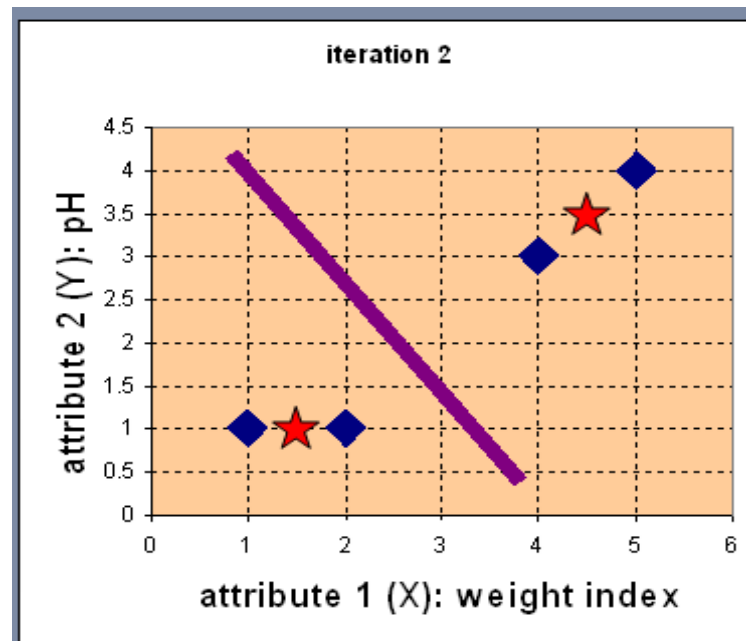
$$\mathbf{G}^1 = \begin{array}{cccc} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} & \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} & \end{array}$$

# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

7. **Iteration 2, determine centroids:** Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new centroids are

$$c_2 = \left( \frac{4+5}{2}, \frac{3+4}{2} \right) = \left( 4\frac{1}{2}, 3\frac{1}{2} \right)$$

$$c_1 = \left( \frac{1+2}{2}, \frac{1+1}{2} \right) = \left( 1\frac{1}{2}, 1 \right)$$



# THE *K-MEANS* CLUSTERING NUMERICAL EXAMPLE

June 19, 2020

8. **Iteration-2, Objects-Centroids distances** :  
Repeat step 2 again, we have new distance matrix at iteration 2 as

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{array}{l} \mathbf{c}_1 = (1\frac{1}{2}, 1) \text{ group-1} \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) \text{ group-2} \end{array}$$

$A$	$B$	$C$	$D$	
$\begin{bmatrix} 1 & 2 & 4 & 5 \end{bmatrix}$	$X$			
$\begin{bmatrix} 1 & 1 & 3 & 4 \end{bmatrix}$	$Y$			

9. **Iteration-2, Objects clustering** : Again, we assign each object based on the minimum distance.

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array}$$

$A$	$B$	$C$	$D$
-----	-----	-----	-----



# THE *K*-MEANS CLUSTERING NUMERICAL EXAMPLE

- We obtain result that  $G^2 = G^1$  . Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore. Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed. We get the final grouping as the results

Object	attribute 1 (X): weight index	attribute 2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

# STRENGTHS OF *K-MEANS* METHOD

- Strength
- It is sensitive with respect to data ordering
- Easily understood
- k-means is most used method
- *Relatively efficient:  $O(tkn)$* , where  $n$  is # objects,  $k$  is # clusters, and  $t$  is # iterations. Normally,  $k, t \ll n$ .

# WEAKNESSES OF *K*-MEANS METHOD

## ○ Weakness

- Applicable only when *mean* is defined, then what about categorical data?
- Need to specify  $k$ , the *number* of clusters, in advance
- It is not obvious what is a good  $k$  to use
- The process is sensitive with respect to outliers
- The algorithm lacks scalability
- Only numerical attributes are covered
- Resulting clusters can be unbalanced
- Unable to handle noisy data and outliers
- Not suitable to discover clusters with non-convex shapes
- The result strongly depends on the initial guess of centroids (or assignments)

# COMMENTS ON *K-MEANS* METHOD

- **Complexity:**

- K-Means:  $O(kn)$  per iteration

- **Uses:**

- All data sizes,
- Best with well separated clusters

- **Examples:**

PAM, CLARA, CLARANS, HMETIS, BAG

# VARIATIONS OF THE *K-MEANS* METHOD

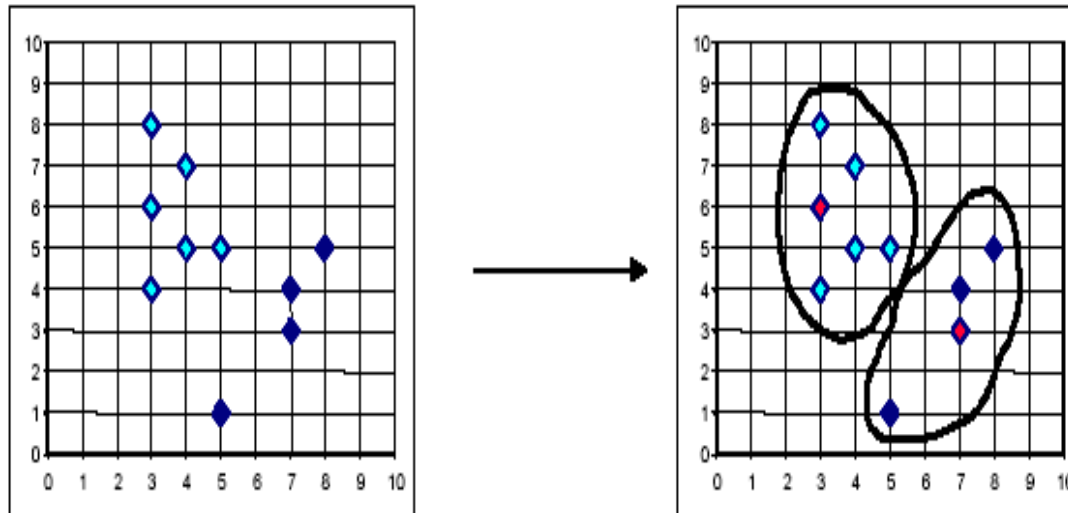
- **K-medoids** – instead of mean, use medians of each cluster
  - Mean of 1, 3, 5, 7, 9 is **5**
  - Mean of 1, 3, 5, 7, 1009 is **205**
  - Median of 1, 3, 5, 7, 1009 is **5**
  - Median advantage: not affected by extreme values
- For large databases, use sampling

# THE *K-MEDOIDS* CLUSTERING METHOD

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 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
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling

# K-MEDOIDS

K-medoids: the most centrally located object in a cluster



# PAM (PARTITIONING AROUND MEDOIDS)

## K-Medoids

- Handles outliers well.
- Ordering of input does not impact results.
- Does not scale well.
- Each cluster represented by one item, called the *medoid*.
- Initial set of k medoids randomly chosen.



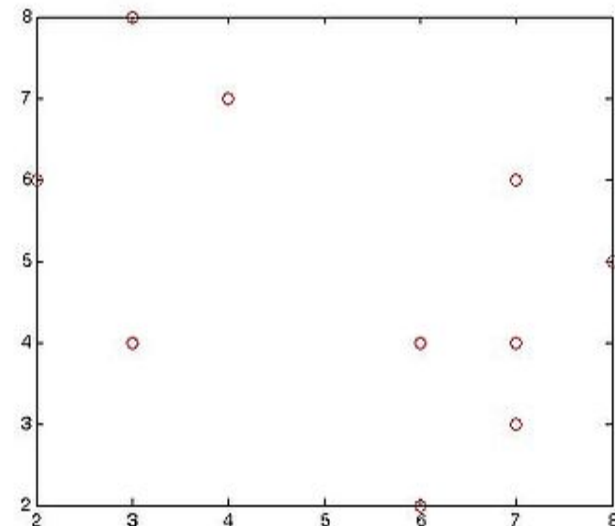
# PAM: BASIC STRATEGY

- First find a **representative object** (the medoid) for each cluster
- Each remaining object is clustered with the medoid to which it is most “similar”
- Iteratively replace one of the medoids by a non-medoid as long as the “quality” of the clustering is improved

# DEMONSTRATION OF PAM

- Cluster the following data set of ten objects into two clusters i.e  $k = 2$ .
- Consider a data set of ten objects as follows:

$X_1$	2	6
$X_2$	3	4
$X_3$	3	8
$X_4$	4	7
$X_5$	6	2
$X_6$	6	4
$X_7$	7	3
$X_8$	7	4
$X_9$	8	5
$X_{10}$	7	6



# DEMONSTRATION OF PAM (CONT...)

## ○ Step 1

- Initialise  $k$  centre
- Let us assume  $c_1 = (3,4)$  and  $c_2 = (7,4)$
- So here  $c_1$  and  $c_2$  are selected as medoid.
- Calculating distance so as to associate each data object to its nearest medoid. Cost is calculated using [Minkowski distance](#) metric with  $r = 1$ .

$c_1$		Data objects ( $\bar{X}_j$ )		Cost (distance)
3	4	2	6	3
3	4	3	8	4
3	4	4	7	4
3	4	6	2	5
3	4	6	4	3
3	4	7	3	5
3	4	8	5	6
3	4	7	6	6

# DEMONSTRATION OF PAM (CONT...)

c <sub>2</sub>		Data objects ( $\bar{X}_j$ )		Cost (distance)
7	4	2	6	7
7	4	3	8	8
7	4	4	7	6
7	4	6	2	3
7	4	6	4	1
7	4	7	3	1
7	4	8	5	2
7	4	7	6	2

Then so the clusters become:

Cluster<sub>1</sub> = {(3,4)(2,6)(3,8)(4,7)}

Cluster<sub>2</sub> = {(7,4)(6,2)(6,4)(7,3)(8,5)(7,6)}

Since the points (2,6) (3,8) and (4,7) are close to c<sub>1</sub> hence they form one cluster whilst remaining points form another cluster.

So the total cost involved is 20.

Where cost between any two points is found using formula

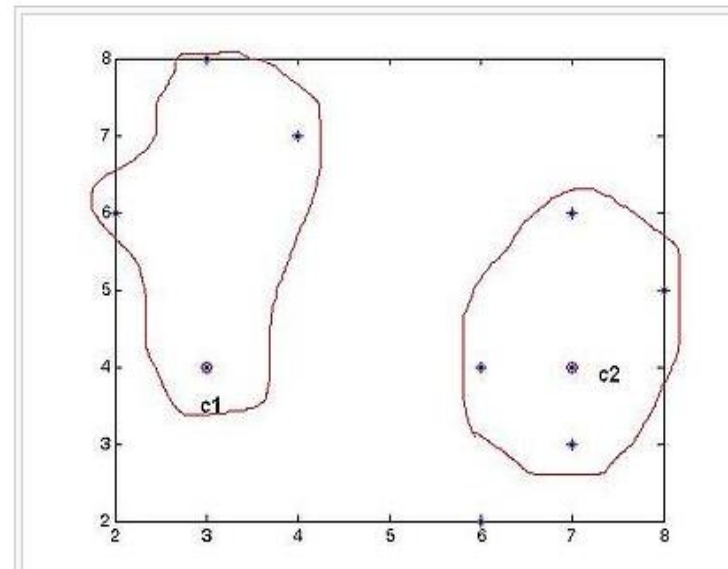
$$\text{cost}(x, c) = \sum_{i=1}^d |x - c|$$

# DEMONSTRATION OF PAM (CONT...)

where  $x$  is any data object,  $c$  is the medoid, and  $d$  is the dimension of the object which in this case is 2.

Total cost is the summation of the cost of data object from its medoid in its cluster so here:

$$\begin{aligned}\text{total cost} &= \{\text{cost}((3,4), (2,6)) + \text{cost}((3,4), (3,8)) + \text{cost}((3,4), (4,7))\} \\ &\quad + \{\text{cost}((7,4), (6,2)) + \text{cost}((7,4), (6,4)) + \text{cost}((7,4), (7,3)) \\ &\quad + \text{cost}((7,4), (8,5)) + \text{cost}((7,4), (7,6))\} \\ &= (3+4+4) + (3+1+1+2+2) \\ &= 20\end{aligned}$$



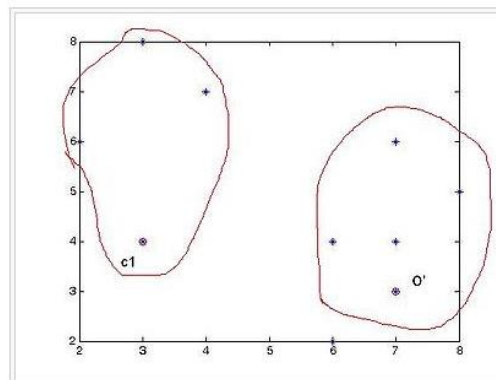
# DEMONSTRATION OF PAM (CONT...)

## Step 2

- Selection of nonmedoid  $O'$  randomly
- Let us assume  $O' = (7,3)$
- So now the medoids are  $c_1(3,4)$  and  $O'(7,3)$
- If  $c_1$  and  $O'$  are new medoids, calculate the total cost involved
- By using the formula in the step 1

$c_1$		Data objects ( $\bar{X}_i$ )		Cost (distance)
3	4	2	6	3
3	4	3	8	4
3	4	4	7	4
3	4	6	2	5
3	4	6	4	3
3	4	7	4	4
3	4	8	5	6
3	4	7	6	6

$O'$		Data objects ( $\bar{X}_i$ )		Cost (distance)
7	3	2	6	8
7	3	3	8	9
7	3	4	7	7
7	3	6	2	2
7	3	6	4	2
7	3	7	4	1
7	3	8	5	3
7	3	7	6	3



# DEMONSTRATION OF PAM (CONT...)

$$\begin{aligned}\text{total cost} &= 3 + 4 + 4 + 2 + 2 + 1 + 3 + 3 \\ &= 22\end{aligned}$$

So cost of swapping medoid from  $c_2$  to  $O'$  is

$$\begin{aligned}S &= \text{current total cost} - \text{past total cost} \\ &= 22 - 20 \\ &= 2 > 0.\end{aligned}$$

So moving to  $O'$  would be bad idea, so the previous choice was good and algorithm terminates here (i.e there is no change in the medoids).

It may happen some data points may shift from one cluster to another cluster depending upon their closeness to medoid.

# PAM (PARTITIONING AROUND MEDOIDS)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select  $k$  representative objects arbitrarily
  - For each pair of non-selected object  $h$  and selected object  $i$ , calculate the total swapping cost  $TC_{ih}$
  - For each pair of  $i$  and  $h$ ,
    - If  $TC_{ih} < 0$ ,  $i$  is replaced by  $h$
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change



# Adv & Disadvantages of PAM

June 19, 2020

- PAM is more robust than k-means in the presence of noise and outliers
- Medoids are less influenced by outliers
- PAM is efficient for small data sets but does not scale well for large data sets
- For each iteration Cost  $TC_{ih}$  for  $k(n-k)$  pairs is to be determined
- Sampling based method: CLARA

# CLARA (CLUSTERING LARGE APPLICATIONS) (1990)

June 19, 2020

- CLARA (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

## CLARANS (“RANDOMIZED” CLARA) (1994)

- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han’94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of  $k$  medoids

# CHAPTER 8. CLUSTER ANALYSIS

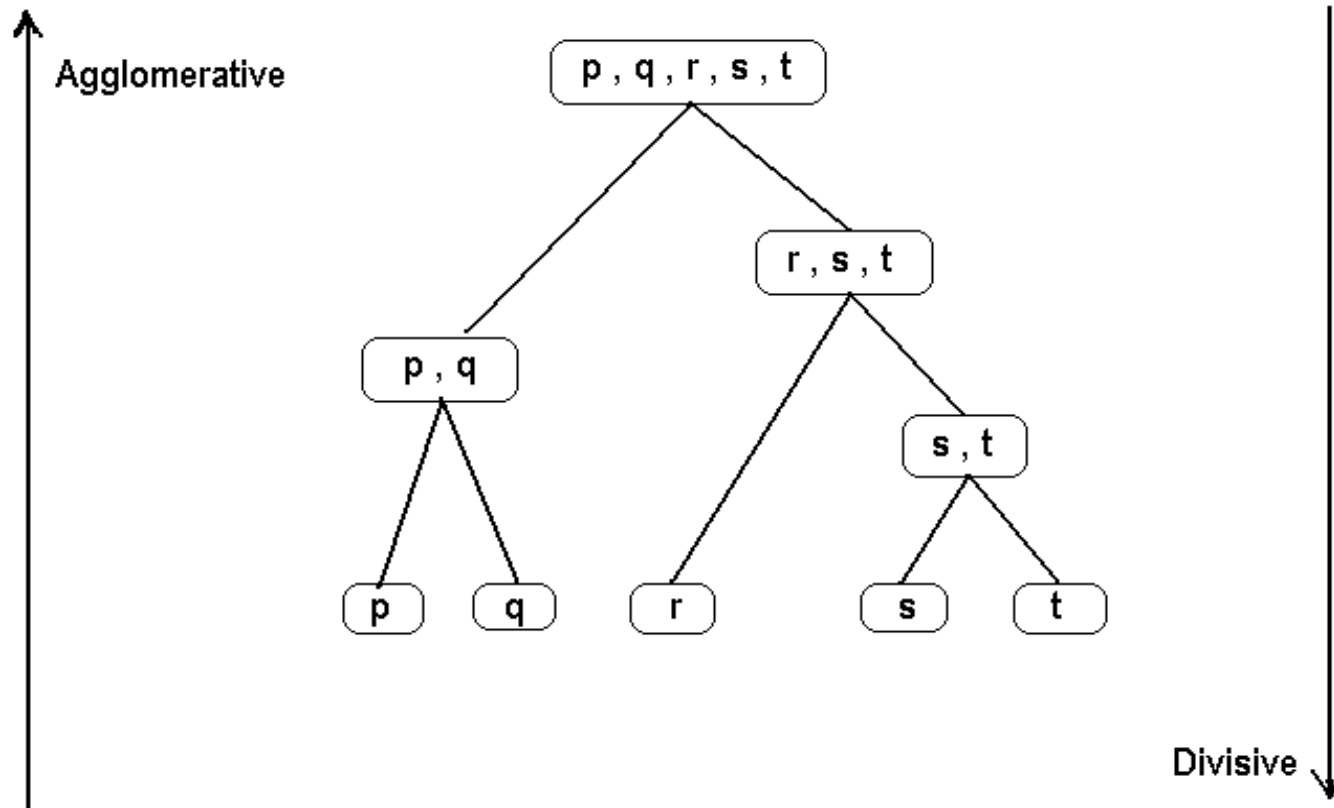
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# HIERARCHICAL CLUSTERING

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

# EXAMPLE

June 19, 2020



# DIFFICULTIES WITH HIERARCHICAL CLUSTERING

June 19, 2020

- Can never undo.
- No object swapping is allowed
- Merge or split decisions ,if not well chosen may lead to poor quality clusters.
- do not scale well: time complexity of at least  $O(n^2)$ , where  $n$  is the number of total objects.

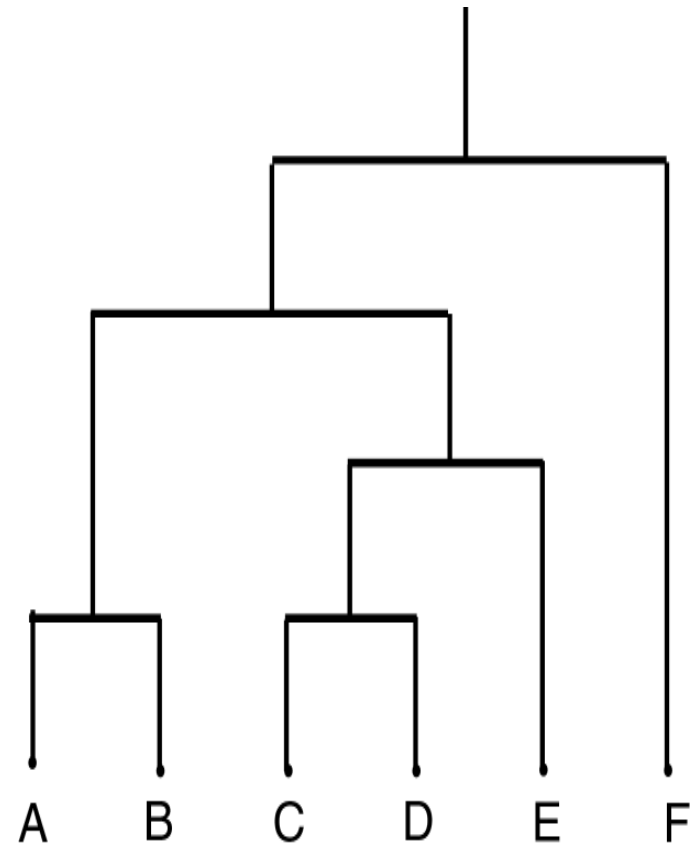
# HIERARCHICAL ALGORITHMS

- Single Link
- Complete Link
- Average Link



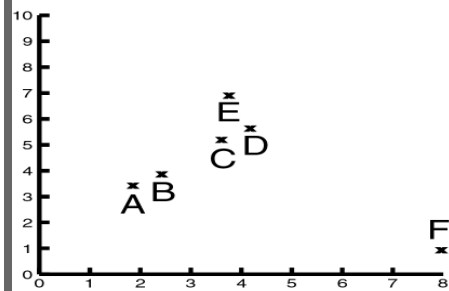
# DENDROGRAM

- ***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  $i+1$ .

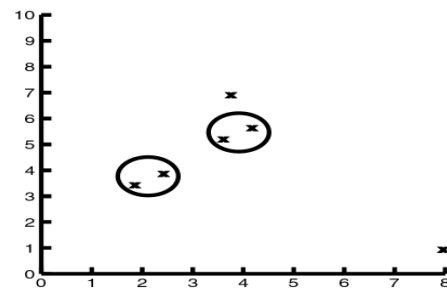


# LEVELS OF CLUSTERING

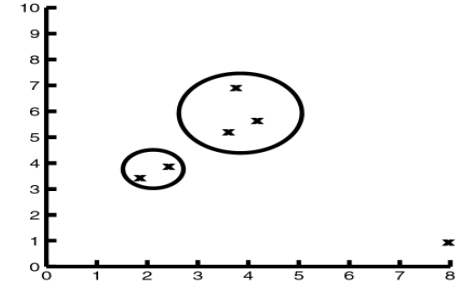
June 19, 2020



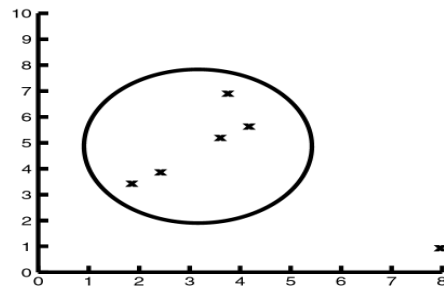
a) Six Clusters



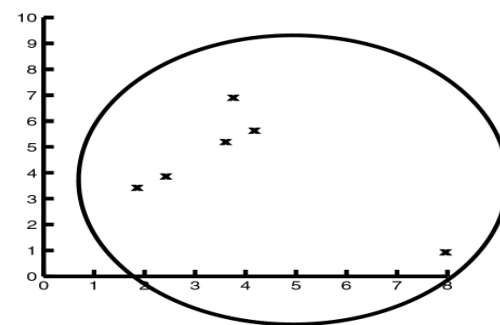
b) Four Clusters



c) Three Clusters



d) Two Clusters



e) One Cluster

# 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.

# SINGLE LINKAGE CLUSTERING

- It is an example of **agglomerative hierarchical** clustering.
- We consider the distance between one cluster and another cluster to be equal to the **shortest distance** from any member of one cluster to any member of the other cluster.

# ALGORITHM

Given a set of  $N$  items to be clustered, and an  $N \times N$  distance (or similarity) matrix, the basic process of single linkage clustering is this:

1. Start by assigning each item to its own cluster, so that if we have  $N$  items, we now have  $N$  clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size  $N$ .

# HOW TO COMPUTE GROUP SIMILARITY?

## Three Popular Methods:

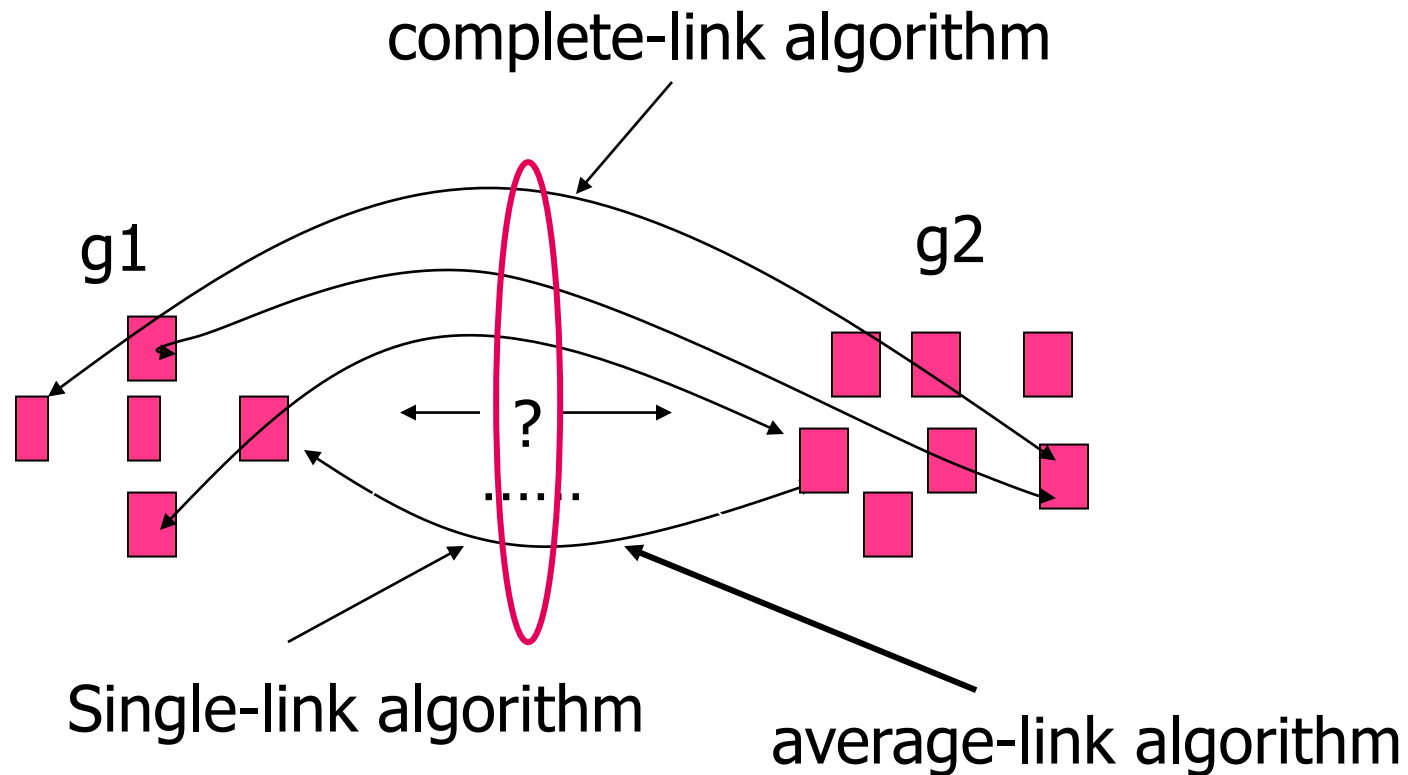
Given two groups  $g_1$  and  $g_2$ ,

*Single-link algorithm:*  $s(g_1, g_2) =$  similarity of the *closest* pair

*Complete-link algorithm:*  $s(g_1, g_2) =$  similarity of the *farthest* pair

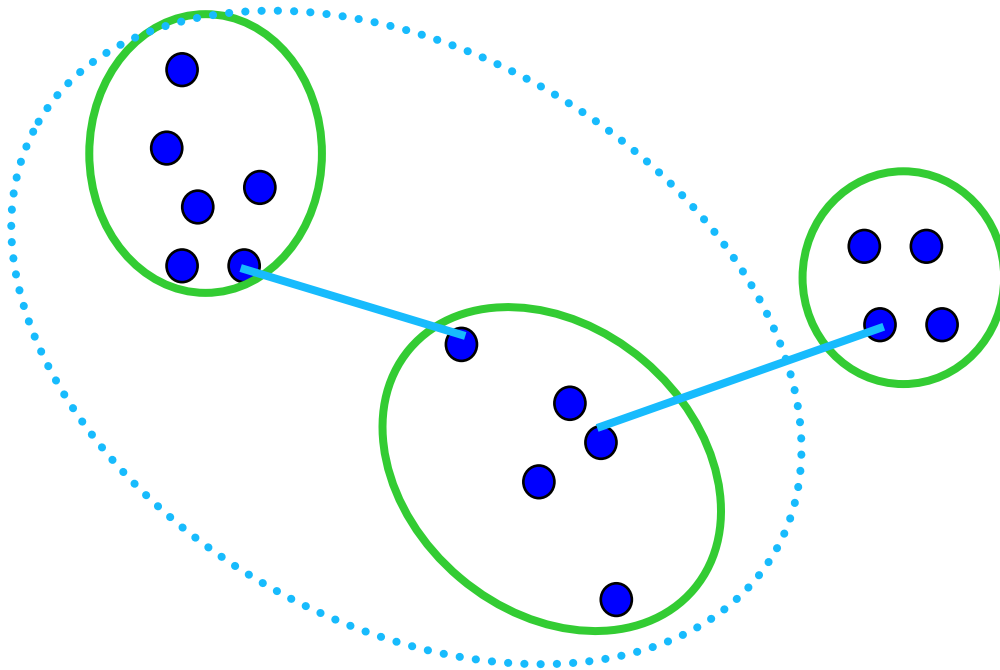
*Average-link algorithm:*  $s(g_1, g_2) =$  *average* of similarity of all pairs

# THREE METHODS ILLUSTRATED



# HIERARCHICAL: SINGLE LINK

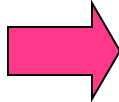
- Cluster similarity = similarity of two *most* similar members





# EXAMPLE: SINGLE LINK

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

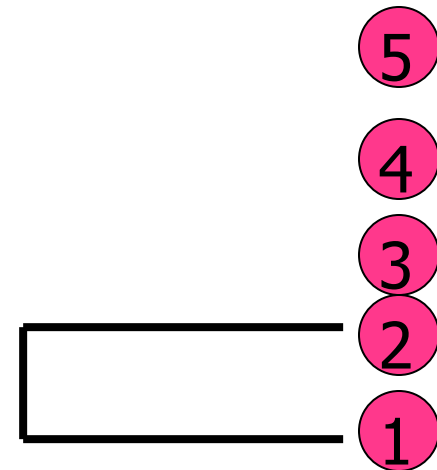


	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6, 3\} = 3$$

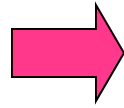
$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10, 9\} = 9$$

$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9, 8\} = 8$$

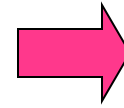


# EXAMPLE: SINGLE LINK

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



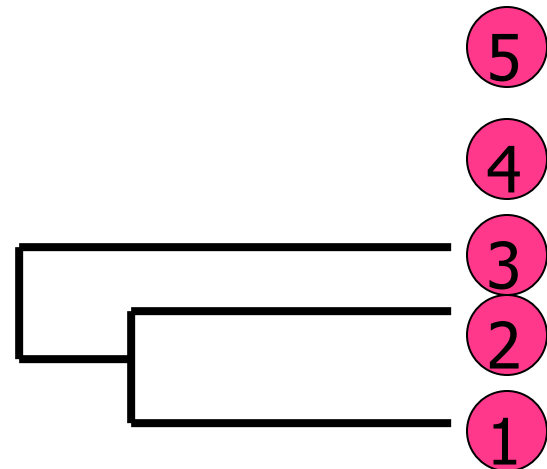
	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0



	(1,2,3)	4	5
(1,2,3)	0		
4	7	0	
5	5	4	0

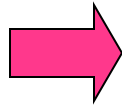
$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9, 7\} = 7$$

$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8, 5\} = 5$$

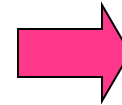


# EXAMPLE: SINGLE LINK

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

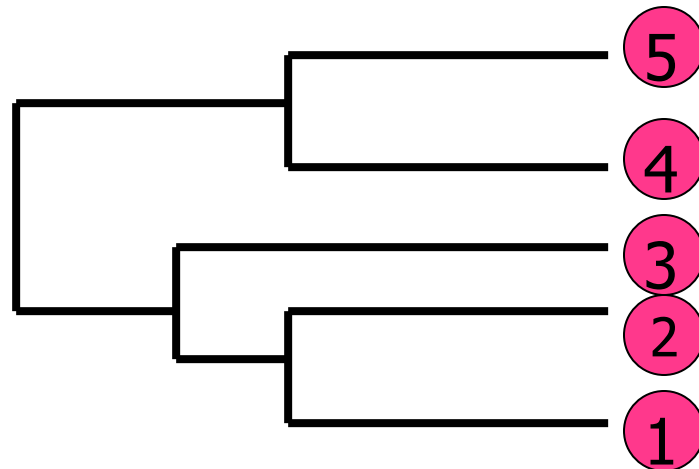


	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0



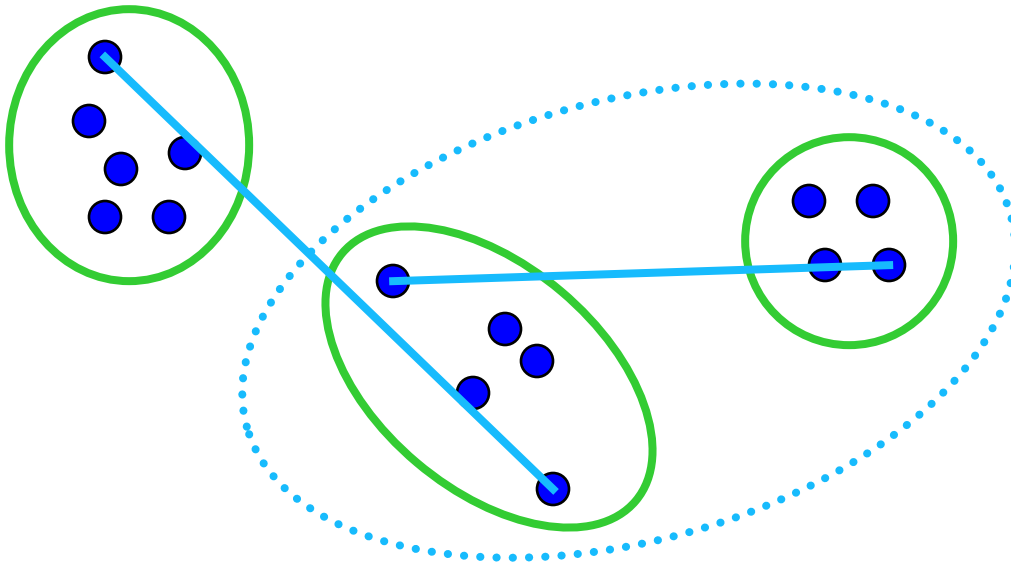
	(1,2,3)	4	5
(1,2,3)	0		
4	7	0	
5	5	4	0

$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$



# HIERARCHICAL: COMPLETE LINK

- Cluster similarity = similarity of *two least* similar members



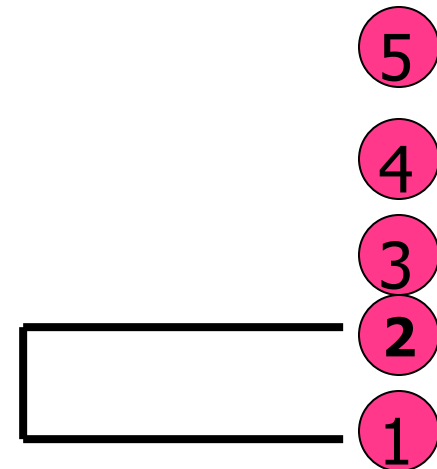
# EXAMPLE: COMPLETE LINK

$$\begin{array}{c}
 \begin{array}{ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix}
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{c}
 \begin{array}{ccccc}
 & (1,2) & 3 & 4 & 5 \\
 \begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 6 & 0 & & & \\ 10 & 7 & 0 & & \\ 9 & 5 & 4 & 0 & \end{bmatrix}
 \end{array}
 \end{array}$$

$$d_{(1,2),3} = \max\{d_{1,3}, d_{2,3}\} = \max\{6, 3\} = 6$$

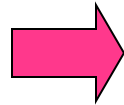
$$d_{(1,2),4} = \max\{d_{1,4}, d_{2,4}\} = \max\{10, 9\} = 10$$

$$d_{(1,2),5} = \max\{d_{1,5}, d_{2,5}\} = \max\{9, 8\} = 9$$

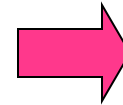


# EXAMPLE: COMPLETE LINK

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \quad 5 \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix}
 \end{array}$$



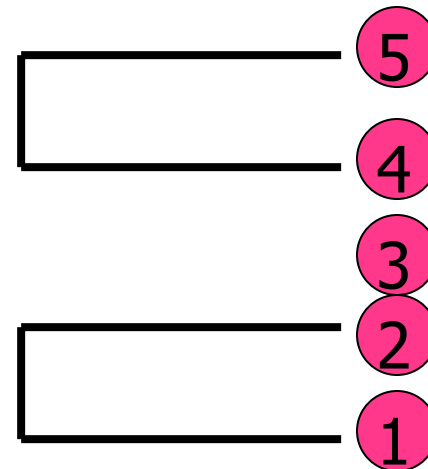
$$\begin{array}{c}
 (1,2) \quad 3 \quad 4 \quad 5 \\
 \begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & \\ 6 & 0 & & \\ 10 & 7 & 0 & \\ 9 & 5 & 4 & 0 \end{bmatrix}
 \end{array}$$



$$\begin{array}{c}
 (1,2) \quad 3 \quad (4,5) \\
 \begin{array}{c} (1,2) \\ 3 \\ (4,5) \end{array} \begin{bmatrix} 0 & & \\ 6 & 0 & \\ 10 & 7 & 0 \end{bmatrix}
 \end{array}$$

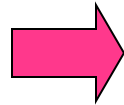
$$d_{(1,2),(4,5)} = \max\{d_{(1,2),4}, d_{(1,2),5}\} = \max\{10, 9\} = 10$$

$$d_{3,(4,5)} = \max\{d_{3,4}, d_{3,5}\} = \max\{7, 5\} = 7$$

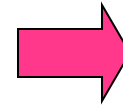


# EXAMPLE: COMPLETE LINK

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

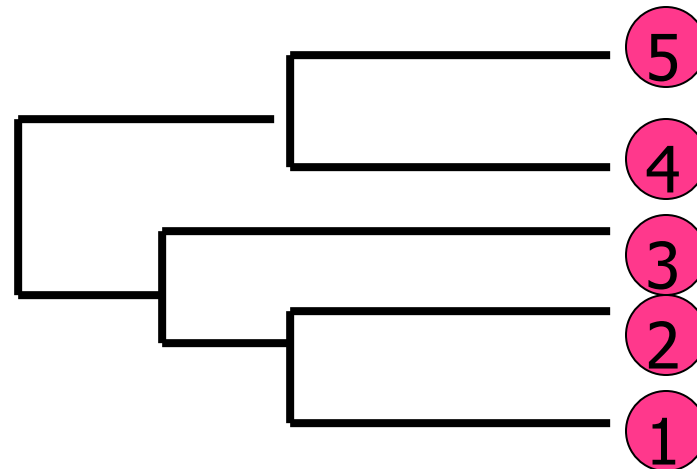


	(1,2)	3	4	5
(1,2)	0			
3	6	0		
4	10	7	0	
5	9	5	4	0



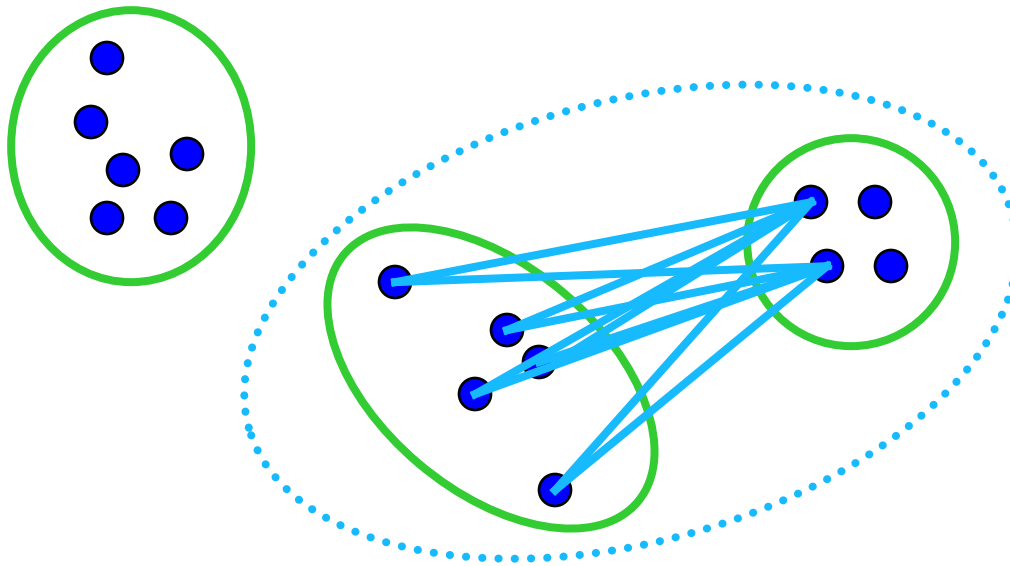
	(1,2)	3	(4,5)
(1,2)	0		
3	6	0	
(4,5)	10	7	0

$$d_{(1,2,3),(4,5)} = \max\{d_{(1,2),(4,5)}, d_{3,(4,5)}\} = 10$$



# HIERARCHICAL: AVERAGE LINK

- Cluster similarity = *average* similarity of all pairs





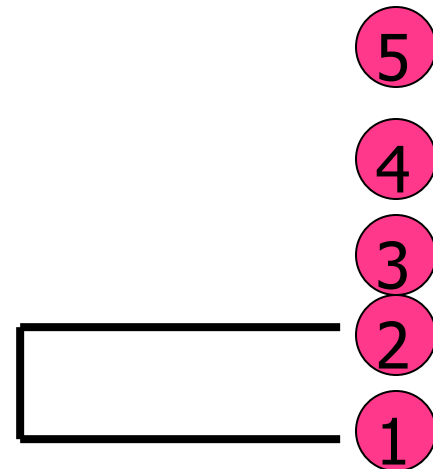
# EXAMPLE: AVERAGE LINK

$$\begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \quad \rightarrow \quad \begin{array}{c} (1,2) \quad 3 \quad 4 \quad 5 \\ \begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ 4.5 & 0 & & & \\ 9.5 & 7 & 0 & & \\ 8.5 & 5 & 4 & 0 & \end{bmatrix} \end{array}$$

$$d_{(1,2),3} = \frac{1}{2}(d_{1,3} + d_{2,3}) = \frac{6+3}{2} = 4.5$$

$$d_{(1,2),4} = \frac{1}{2}(d_{1,4} + d_{2,4}) = \frac{10+9}{2} = 9.5$$

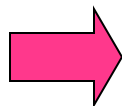
$$d_{(1,2),5} = \frac{1}{2}(d_{1,5} + d_{2,5}) = \frac{9+8}{2} = 8.5$$



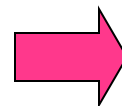
# EXAMPLE: AVERAGE LINK

June 19, 2020

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



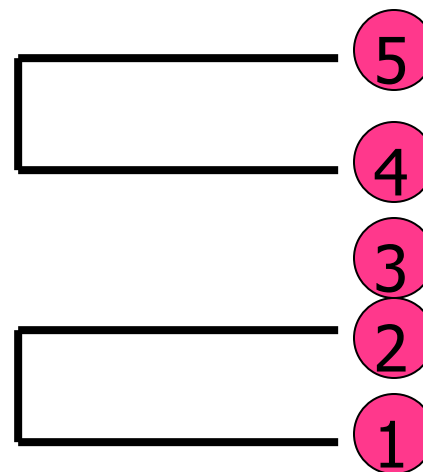
	(1,2)	3	4	5
(1,2)	0			
3	4.5	0		
4	9.5	7	0	
5	8.5	5	4	0



	(1,2)	3	(4,5)
(1,2)	0		
3	4.5	0	
(4,5)	9	6	0

$$d_{(1,2),(4,5)} = \frac{1}{4}(d_{1,4} + d_{1,5} + d_{2,4} + d_{2,5}) = 9$$

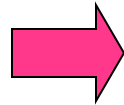
$$d_{3,(4,5)} = \frac{1}{2}(d_{3,4} + d_{3,5}) = 6$$



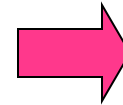
# EXAMPLE: AVERAGE LINK

June 19, 2020

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

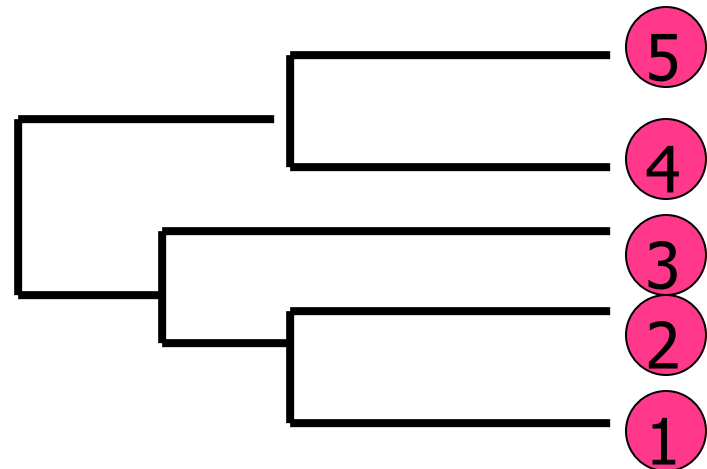


	(1,2)	3	4	5
(1,2)	0			
3	4.5	0		
4	9.5	7	0	
5	8.5	5	4	0



	(1,2)	3	(4,5)
(1,2)	0		
3	4.5	0	
(4,5)	9	6	0

$$d_{(1,2,3),(4,5)} = \frac{1}{6} (d_{1,4} + d_{1,5} + d_{2,4} + d_{2,5} + d_{3,4} + d_{3,5}) = 8$$



# MORE ON HIERARCHICAL CLUSTERING METHODS

June 19, 2020

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least  $O(n^2)$ , where  $n$  is the number of total objects
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

# BIRCH (1996)

June 19, 2020

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *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.

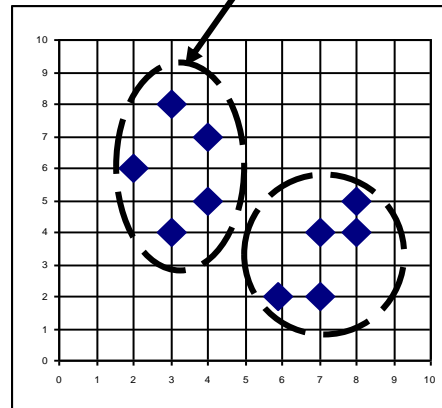
# Clustering Feature Vector

**Clustering Feature:**  $CF = (N, \overrightarrow{LS}, SS)$

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

$$LS: \sum_{i=1}^N \overrightarrow{X_i}$$

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



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

(3,4)

(2,6)

(4,5)

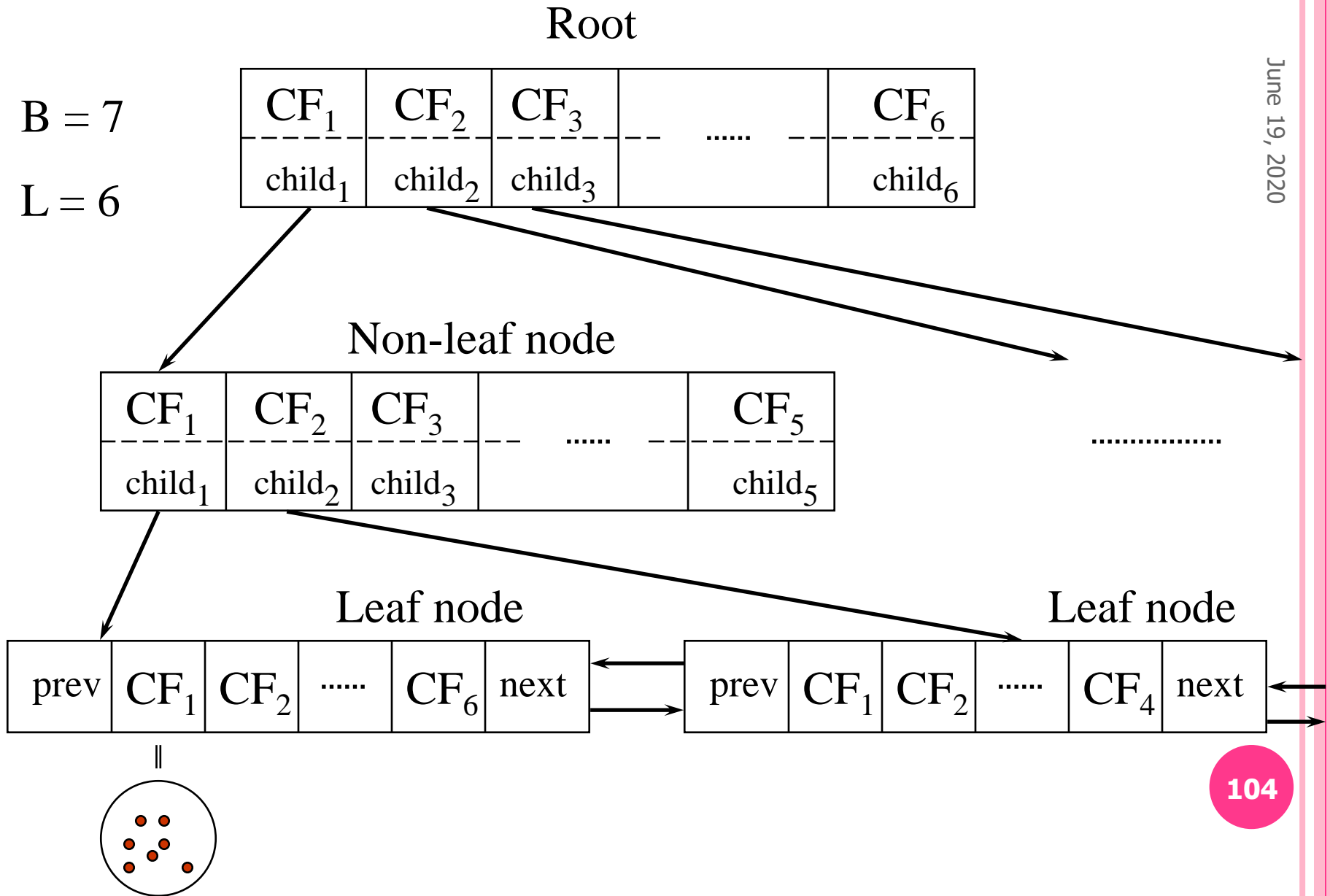
(4,7)

(3,8)

# CF TREE

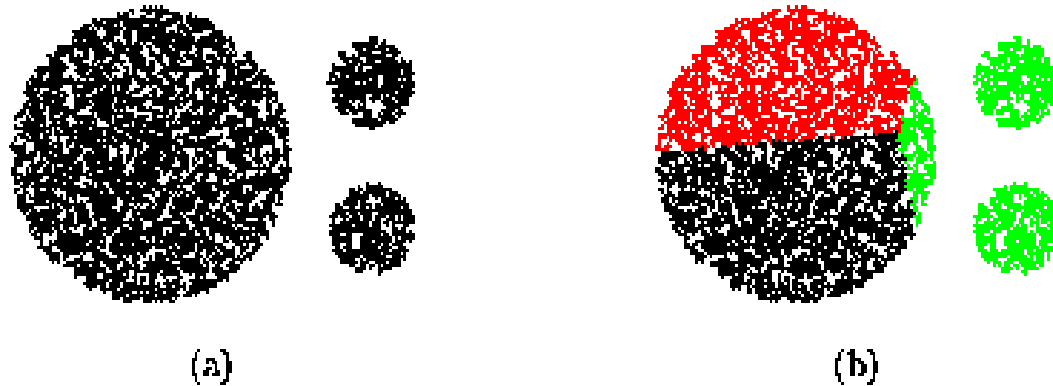
$B = 7$

$L = 6$



# CURE (CLUSTERING USING REPRESENTATIVES)

June 19, 2020



- CURE: proposed by Guha, Rastogi & Shim, 1998
  - Stops the creation of a cluster hierarchy if a level consists of  $k$  clusters
  - Uses multiple representative points to evaluate the distance between clusters, adjusts well to arbitrary shaped clusters and avoids single-link effect



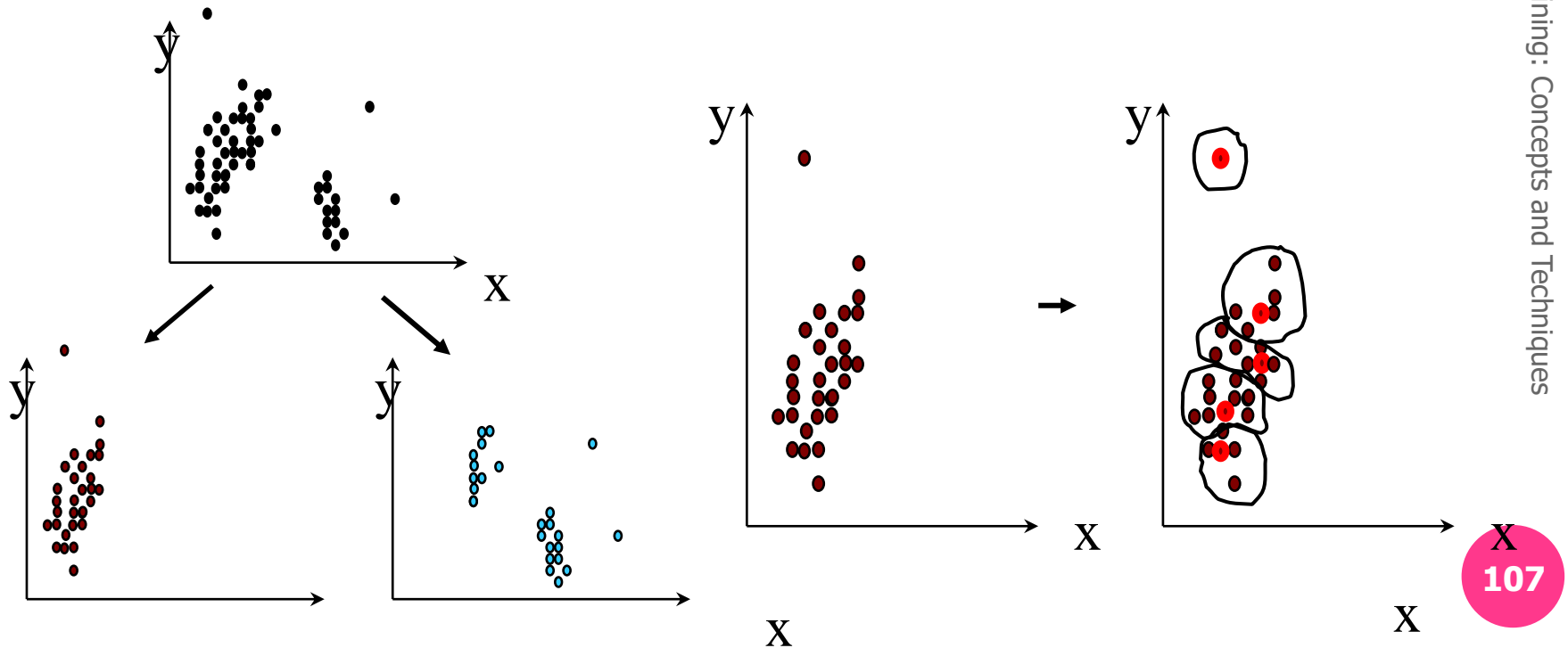
# CURE: THE ALGORITHM

- Draw random sample  $s$ .
- Partition sample to  $p$  partitions with size  $s/p$
- Partially cluster partitions into  $s/pq$  clusters
- Eliminate outliers
  - By random sampling
  - If a cluster grows too slow, eliminate it.
- Cluster partial clusters.
- Label data in disk

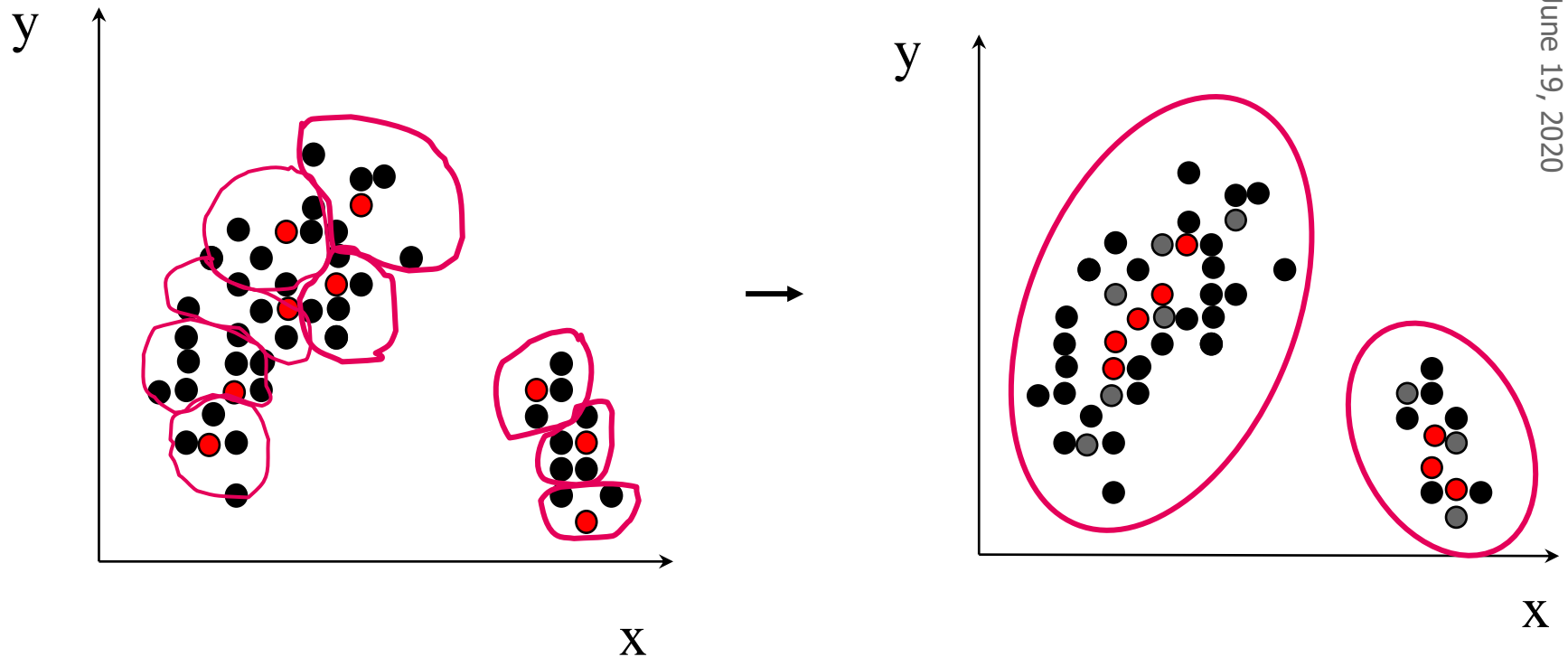
# DATA PARTITIONING AND CLUSTERING

- $s = 50$
- $p = 2$
- $s/p = 25$

$$\blacksquare s/pq = 5$$



# CURE: SHRINKING REPRESENTATIVE POINTS



June 19, 2020

- Shrink the multiple representative points towards the gravity center by a fraction of  $\alpha$ .
- Multiple representatives capture the shape of the cluster

# CLUSTERING CATEGORICAL DATA: ROCK

June 19, 2020

- ROCK: Robust Clustering using linKs, by S. Guha, R. Rastogi, K. Shim (ICDE'99).
  - Use links to measure similarity/proximity
  - Not distance based
  - Computational complexity:  $O(n^2 + nm_m m_a + n^2 \log n)$
- Basic ideas:

- Similarity function and neighbors:  $Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$

Let  $T_1 = \{1, 2, 3\}$ ,  $T_2 = \{3, 4, 5\}$

$$Sim(T_1, T_2) = \frac{|\{3\}|}{|\{1, 2, 3, 4, 5\}|} = \frac{1}{5} = 0.2$$

# ROCK: ALGORITHM

- Links: The number of common neighbours for the two points.

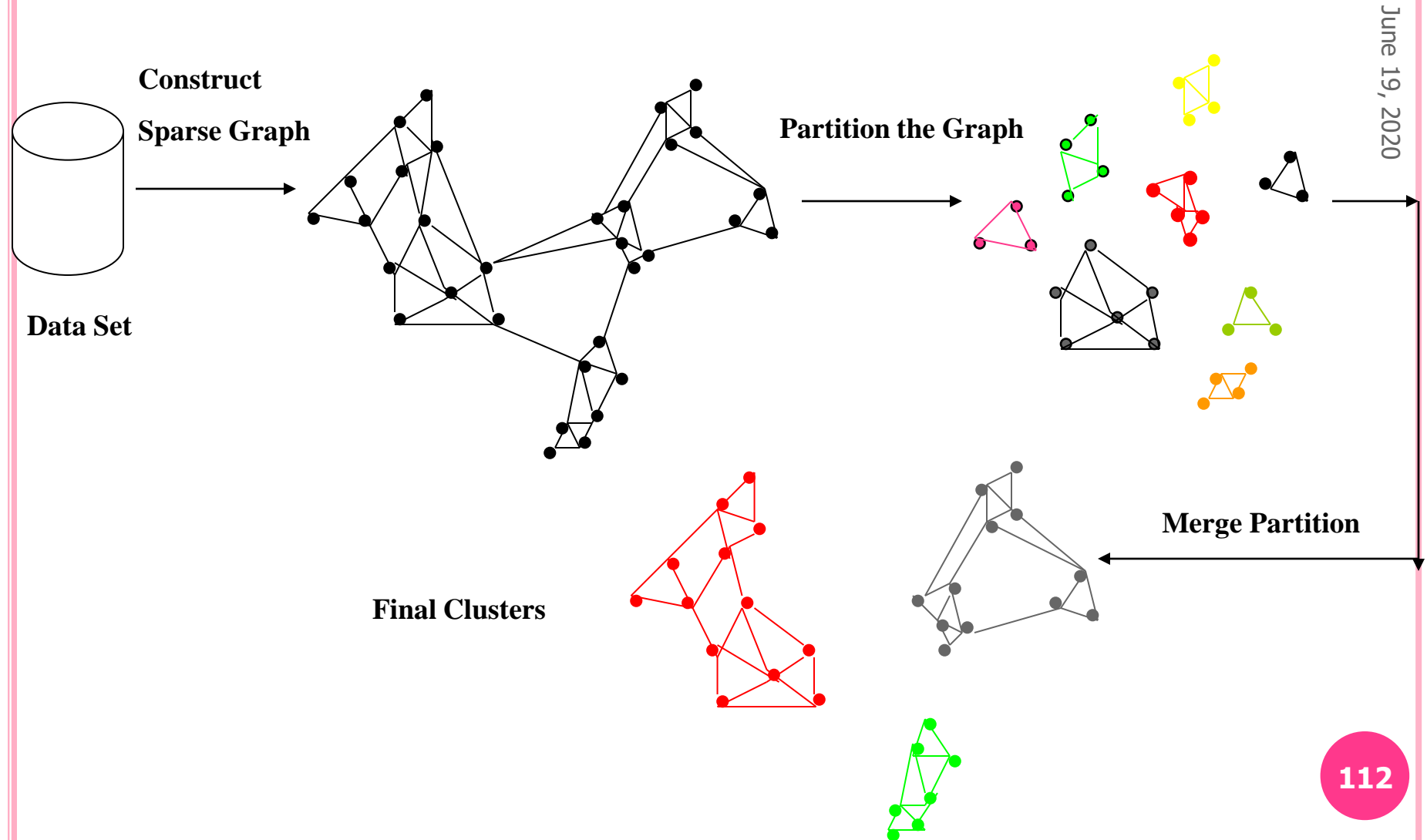
$\{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}$   
 $\{1,4,5\}, \{2,3,4\}, \{2,3,5\}, \{2,4,5\}, \{3,4,5\}$

- Algorithm  $\{1,2,3\} \xleftrightarrow{3} \{1,2,4\}$ 
  - Draw random sample
  - Cluster with links
  - Label data in disk

# CHAMELEON

- CHAMELEON: Hierarchical clustering using dynamic modeling, by G. Karypis, E.H. Han and V. Kumar'99
- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- A two phase algorithm
  - 1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  - 2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

# OVERALL FRAMEWORK OF CHAMELEON



# CHAPTER 8. CLUSTER ANALYSIS

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary



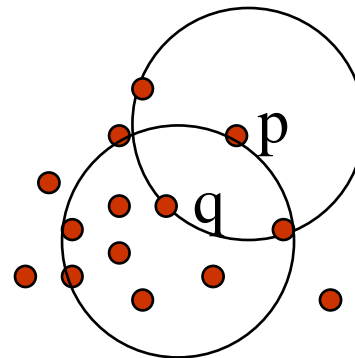
# DENSITY-BASED CLUSTERING METHODS

- 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)

# DENSITY-BASED CLUSTERING: BACKGROUND

- Two parameters:
  - **Eps**: Maximum radius of the neighbourhood
  - **MinPts**: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$ :  $\{q \text{ belongs to } D \mid dist(p,q) \leq Eps\}$
- Directly density-reachable: A point **p** is directly density reachable from a point **q** wrt. **Eps**, **MinPts** if
  - 1) **p** belongs to  $N_{Eps}(q)$
  - 2) core point condition:

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



MinPts = 5

Eps = 1 cm

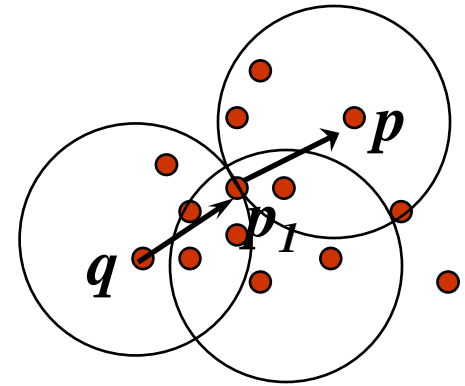
# DENSITY-BASED CLUSTERING: BACKGROUND (II)

June 19, 2020

Data Mining: Concepts and Techniques

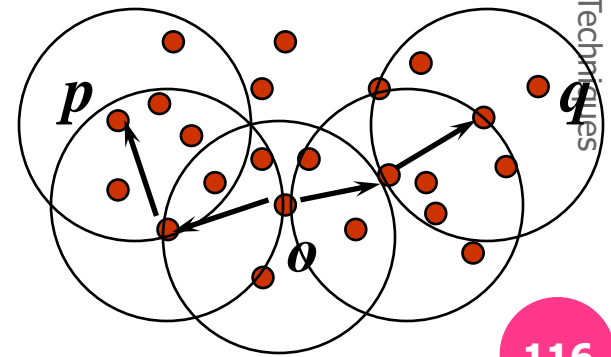
## ○ Density-reachable:

- A point  $p$  is density-reachable from a point  $q$  wrt.  $Eps$ ,  $MinPts$  if there is a chain of points  $p_1, \dots, p_n$ ,  $p_1 = q$ ,  $p_n = p$  such that  $p_{i+1}$  is directly density-reachable from  $p_i$



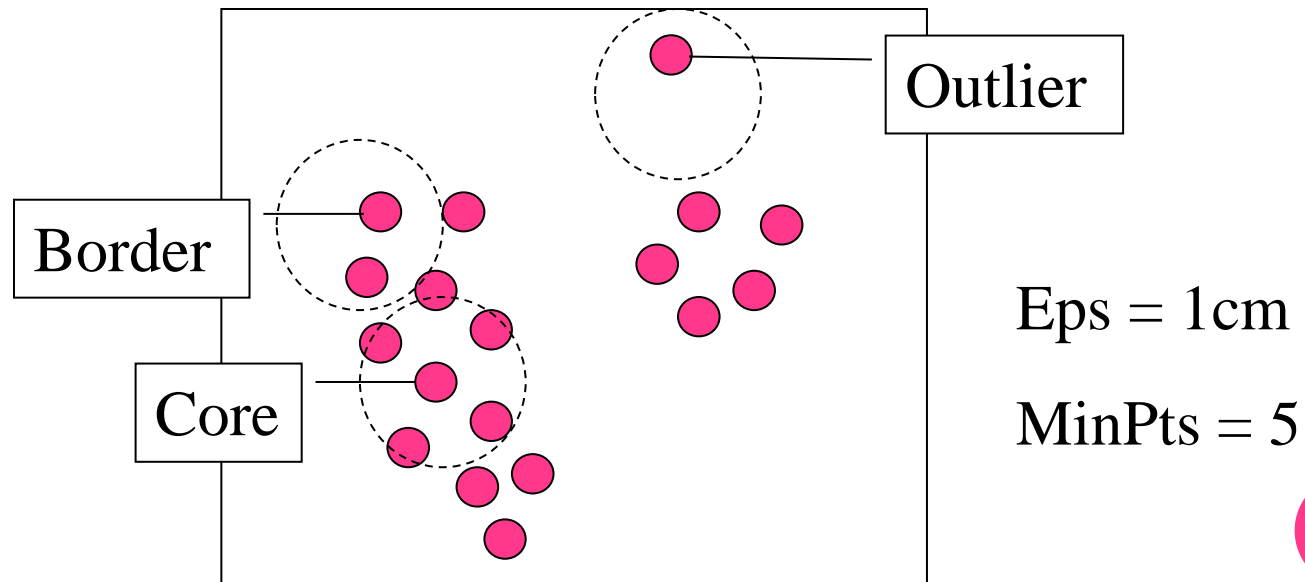
## ○ Density-connected

- A point  $p$  is density-connected to a point  $q$  wrt.  $Eps$ ,  $MinPts$  if there is a point  $o$  such that both,  $p$  and  $q$  are density-reachable from  $o$  wrt.  $Eps$  and  $MinPts$ .



# 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



## DBSCAN: THE ALGORITHM

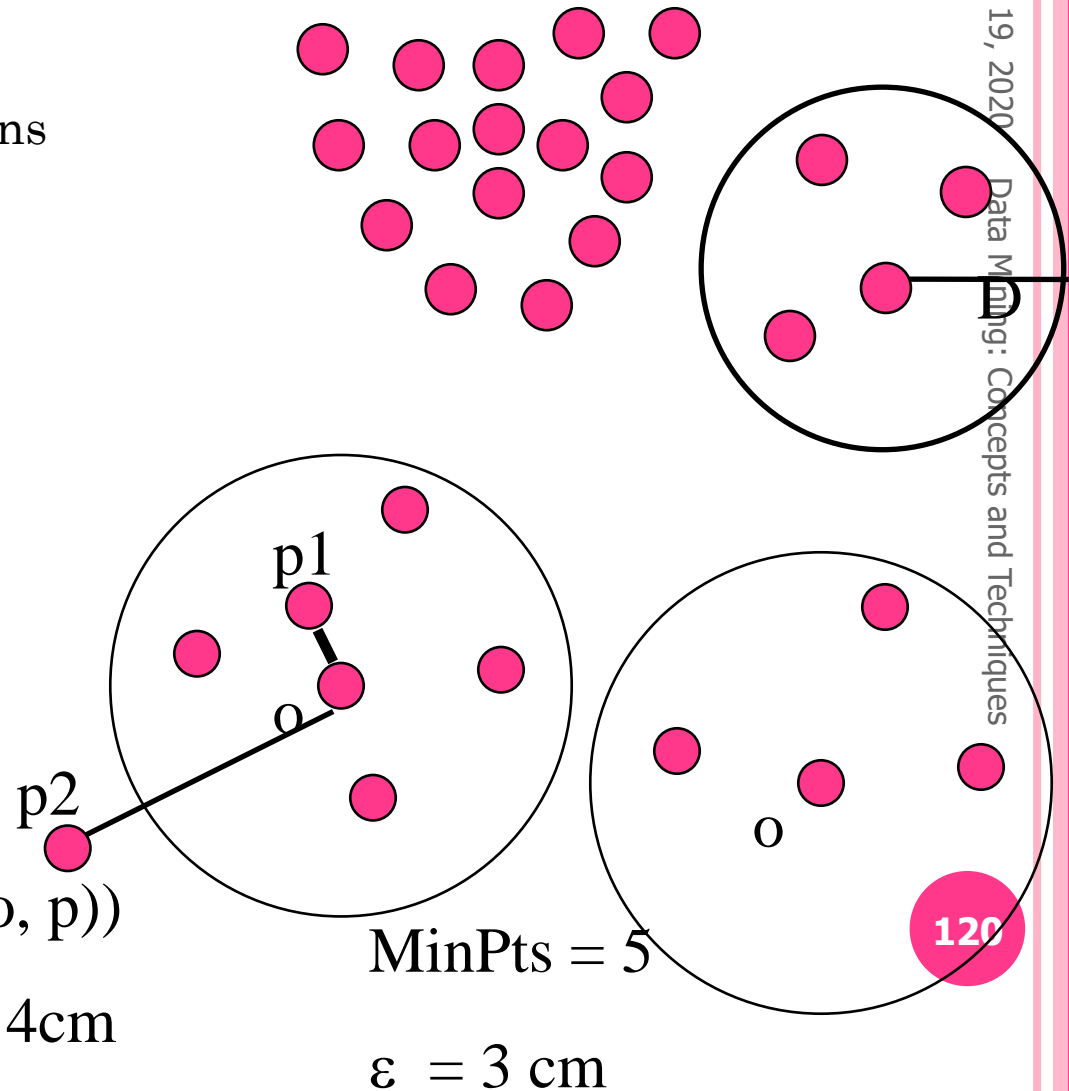
- Arbitrary select a point  $p$
- Retrieve all points density-reachable from  $p$  wrt  $Eps$  and  $MinPts$ .
- If  $p$  is a core point, a cluster is formed.
- If  $p$  is a border point, no points are density-reachable from  $p$  and DBSCAN visits the next point of the database.
- Continue the process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

# OPTICS: A CLUSTER-ORDERING METHOD (1999)

- OPTICS: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
  - Produces a special order of the database wrt its density-based clustering structure
  - This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically or using visualization techniques

# OPTICS: SOME EXTENSION FROM DBSCAN

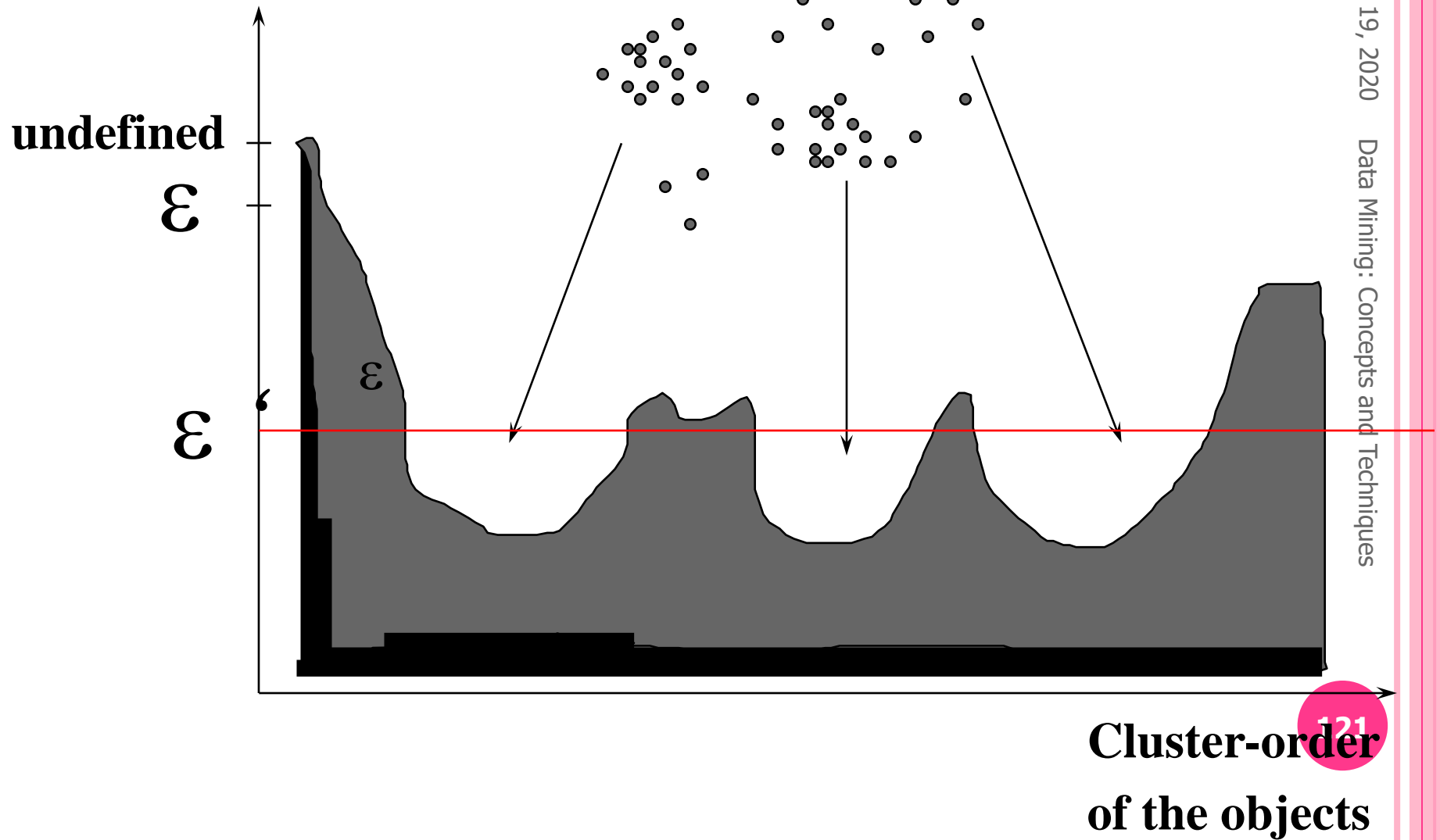
- Index-based:
  - $k$  = number of dimensions
  - $N = 20$
  - $p = 75\%$
  - $M = N(1-p) = 5$
  - Complexity:  $O(kN^2)$
- Core Distance
- Reachability Distance



Max (core-distance ( $o$ ),  $d(o, p)$ )

$r(p1, o) = 2.8\text{cm}$ .  $r(p2, o) = 4\text{cm}$

# Reachability-distance





# DENCLUE: USING DENSITY FUNCTIONS

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
- Major features
  - Solid mathematical foundation
  - Good for data sets with large amounts of noise
  - Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
  - Significant faster than existing algorithm (faster than DBSCAN by a factor of up to 45)
  - But needs a large number of parameters

# DENCLUE: TECHNICAL ESSENCE

- Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure.
- Influence function: describes the impact of a data point within its neighborhood.
- Overall density of the data space can be calculated as the sum of the influence function of all data points.
- Clusters can be determined mathematically by identifying density attractors.
- Density attractors are local maximal of the overall density function.

# GRADIENT: THE STEEPNESS OF A SLOPE

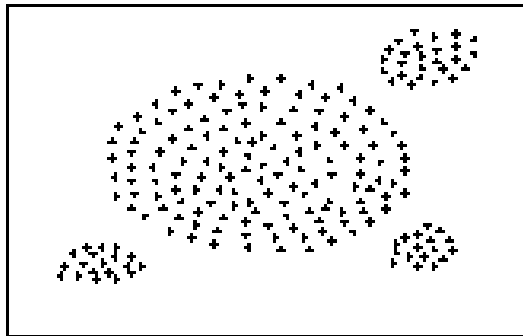
- Example

$$f_{\text{Gaussian}}(x, y) = e^{-\frac{d(x, y)^2}{2\sigma^2}}$$

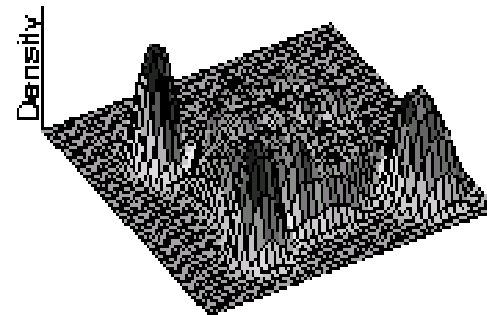
$$f_{\text{Gaussian}}^D(x) = \sum_{i=1}^N e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

$$\nabla f_{\text{Gaussian}}^D(x, x_i) = \sum_{i=1}^N (x_i - x) \cdot e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

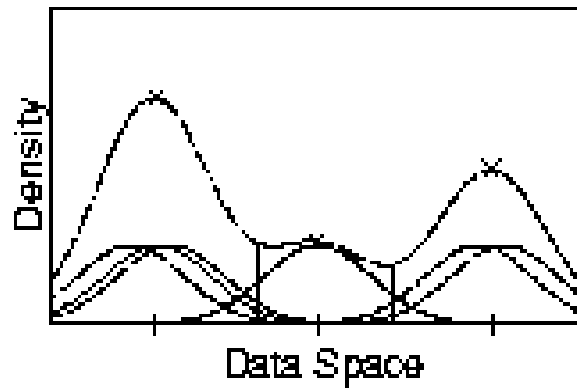
# DENSITY ATTRACTOR



(a) Data Set



(c) Gaussian



# CENTER-DEFINED AND ARBITRARY

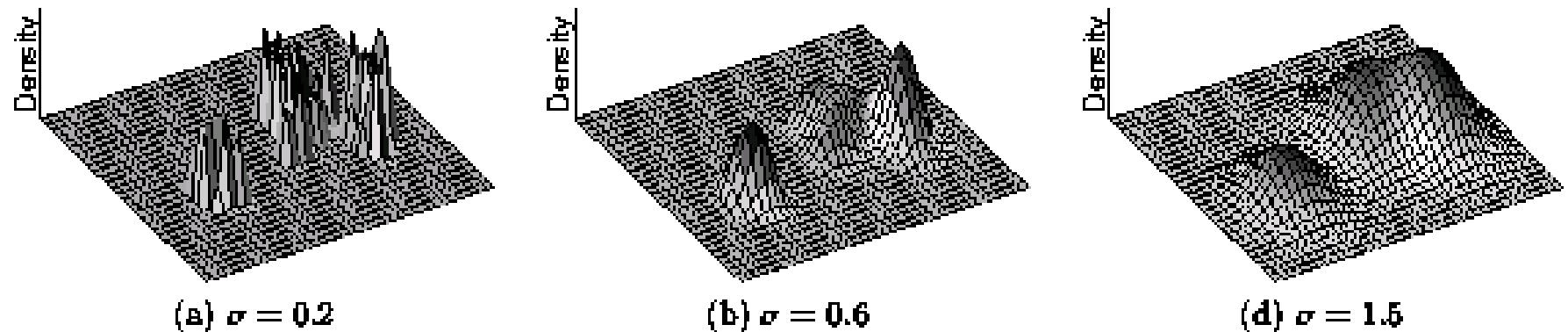


Figure 3: Example of Center-Defined Clusters for different  $\sigma$

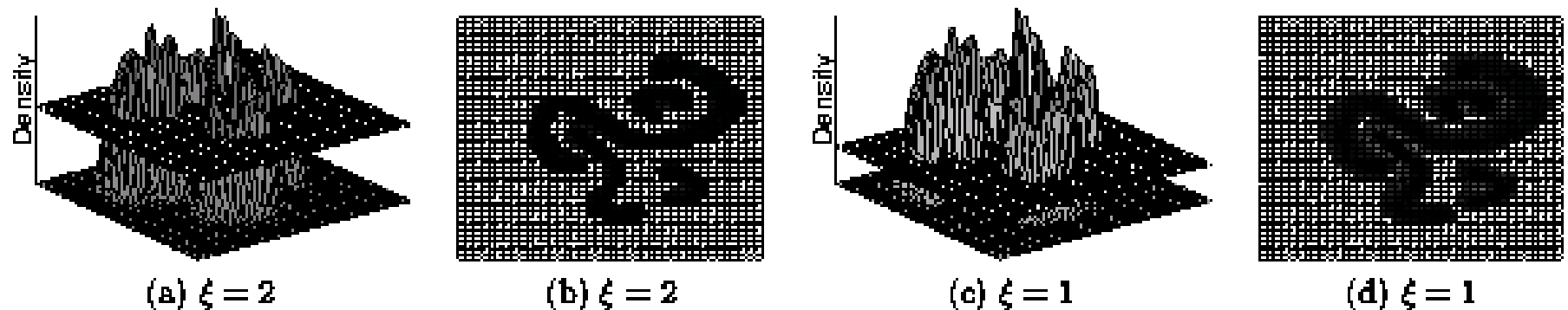


Figure 4: Example of Arbitrary-Shape Clusters for different  $\xi$

# CHAPTER 8. CLUSTER ANALYSIS

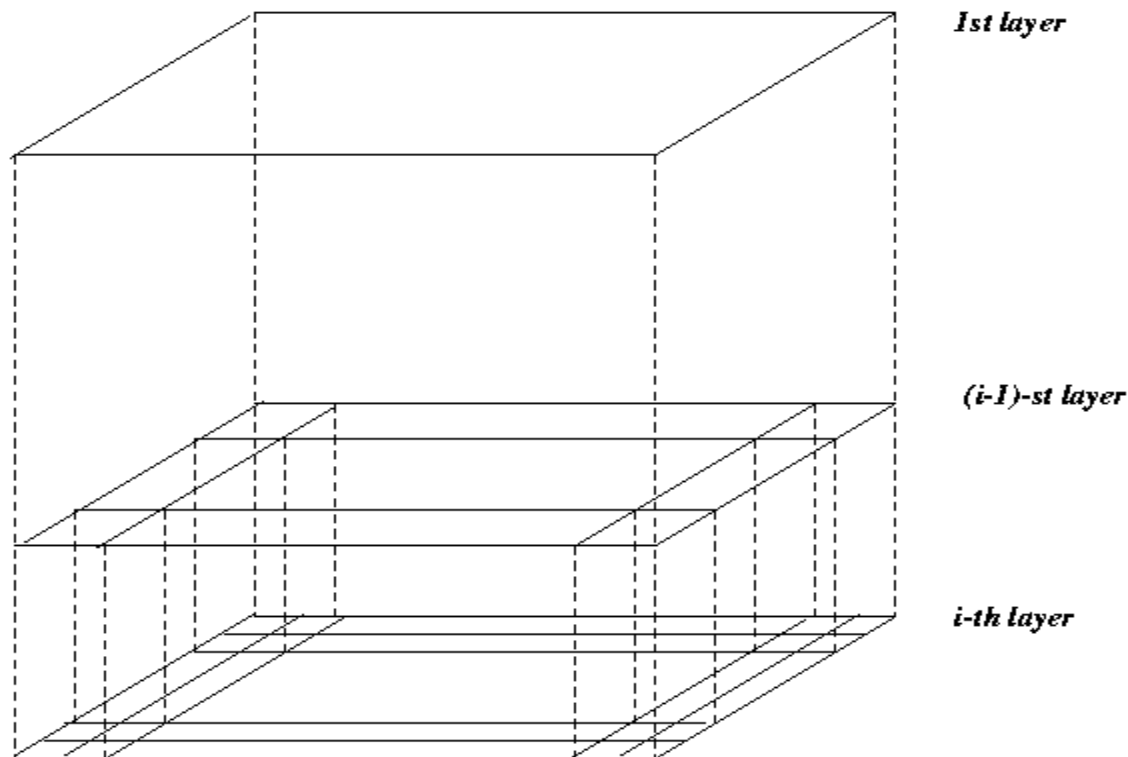
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# 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)

# 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





# STING: A STATISTICAL INFORMATION GRID APPROACH (2)

- 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: A STATISTICAL INFORMATION GRID APPROACH (3)

- 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



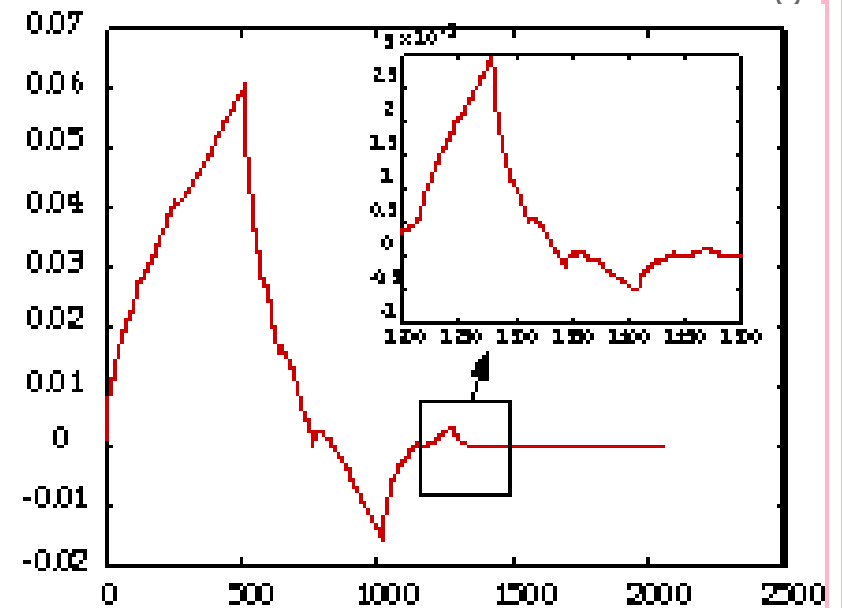
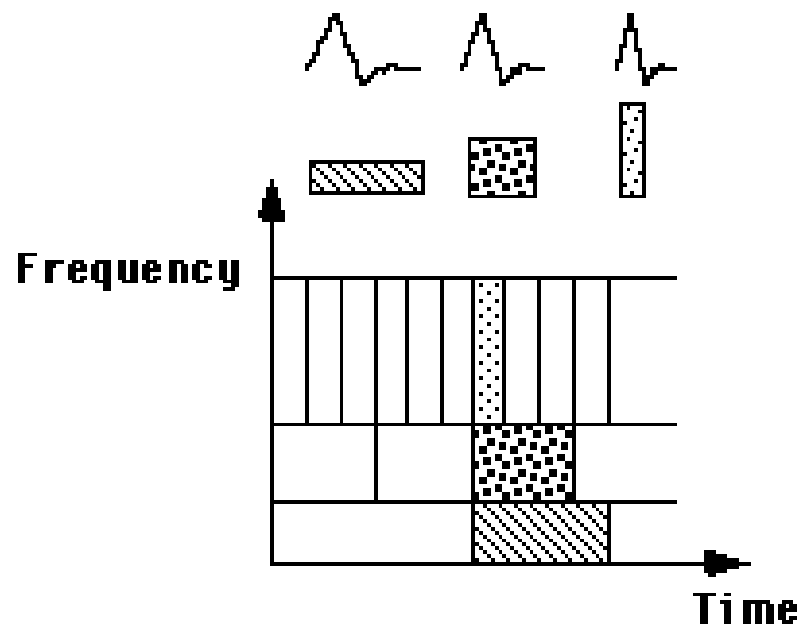
## WAVECLUSTER (1998)

- Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
- A multi-resolution clustering approach which applies wavelet transform to the feature space
  - A wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band.
- Both grid-based and density-based
- Input parameters:
  - # of grid cells for each dimension
  - the wavelet, and the # of applications of wavelet transform.

## WAVECLUSTER (1998)

- How to apply wavelet transform to find clusters
  - Summarizes the data by imposing a multidimensional grid structure onto data space
  - These multidimensional spatial data objects are represented in a n-dimensional feature space
  - Apply wavelet transform on feature space to find the dense regions in the feature space
  - Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse

# WHAT IS WAVELET (2)?



# QUANTIZATION

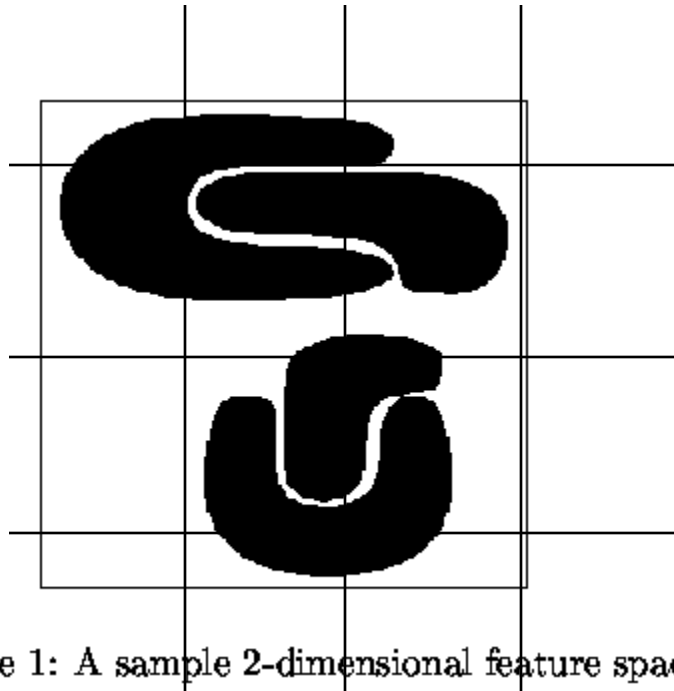
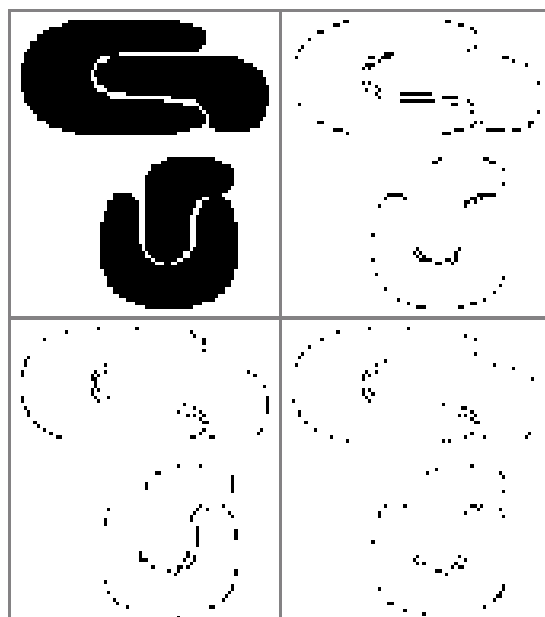


Figure 1: A sample 2-dimensional feature space.

# TRANSFORMATION

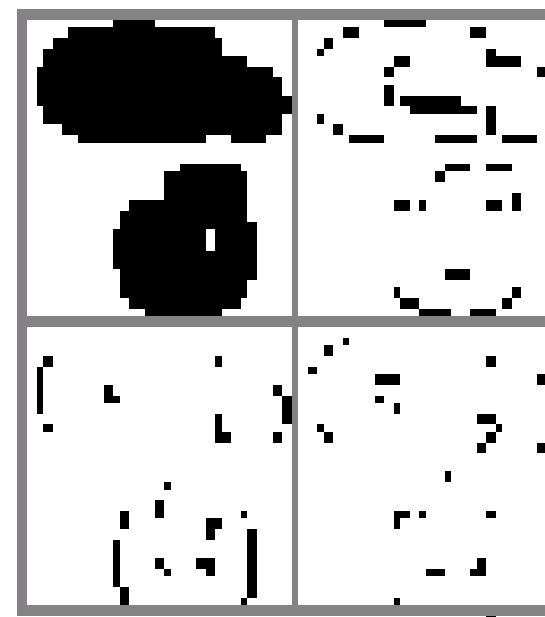
June 19,



a)



b)



c)

# WAVECLUSTER (1998)

- Why is wavelet transformation useful for clustering
  - Unsupervised clustering
    - It uses hat-shape filters to emphasize region where points cluster, but simultaneously to suppress weaker information in their boundary
  - Effective removal of outliers
  - Multi-resolution
  - Cost efficiency
- Major features:
  - Complexity  $O(N)$
  - Detect arbitrary shaped clusters at different scales
  - Not sensitive to noise, not sensitive to input order
  - Only applicable to low dimensional data

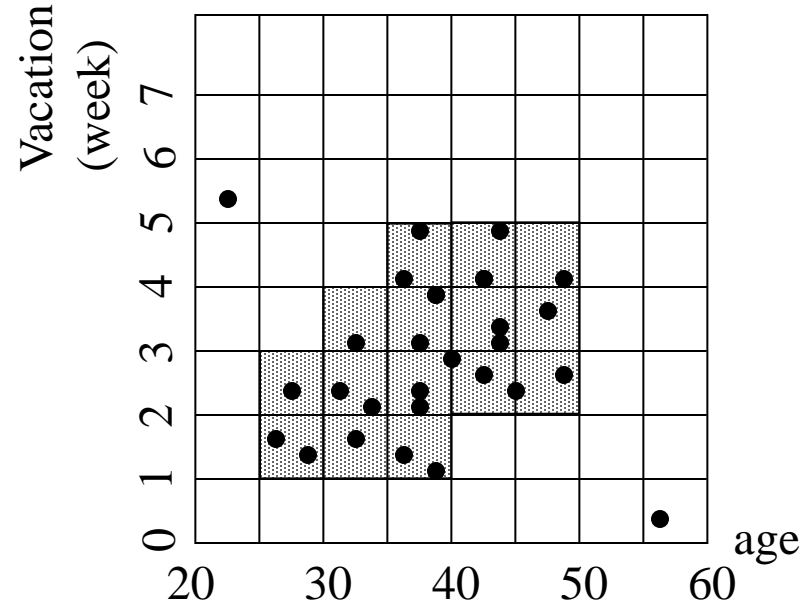
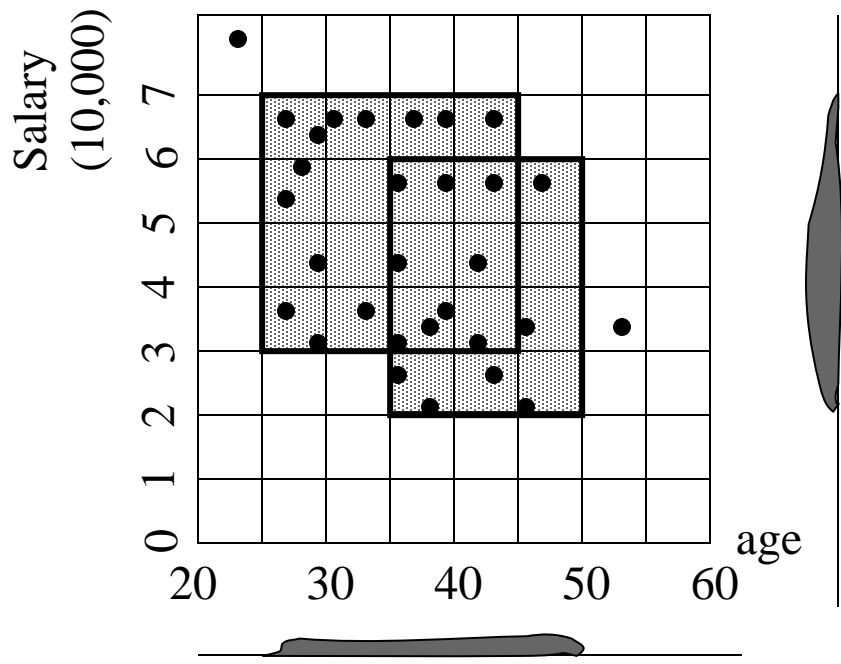


# CLIQUE (CLUSTERING IN QUEST)

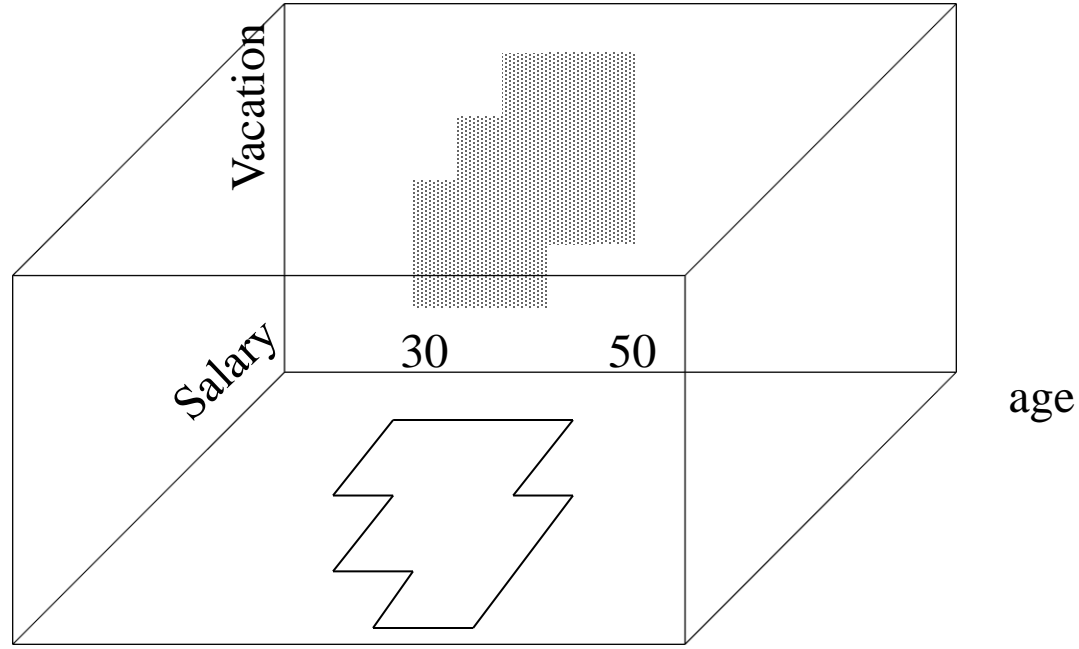
- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length 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



$$\tau = 3$$



# STRENGTH AND WEAKNESS OF *CLIQUE*

## ○ Strength

- It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
- It is *insensitive* to the order of records in input and does not presume some canonical data distribution
- It 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

# CHAPTER 8. CLUSTER ANALYSIS

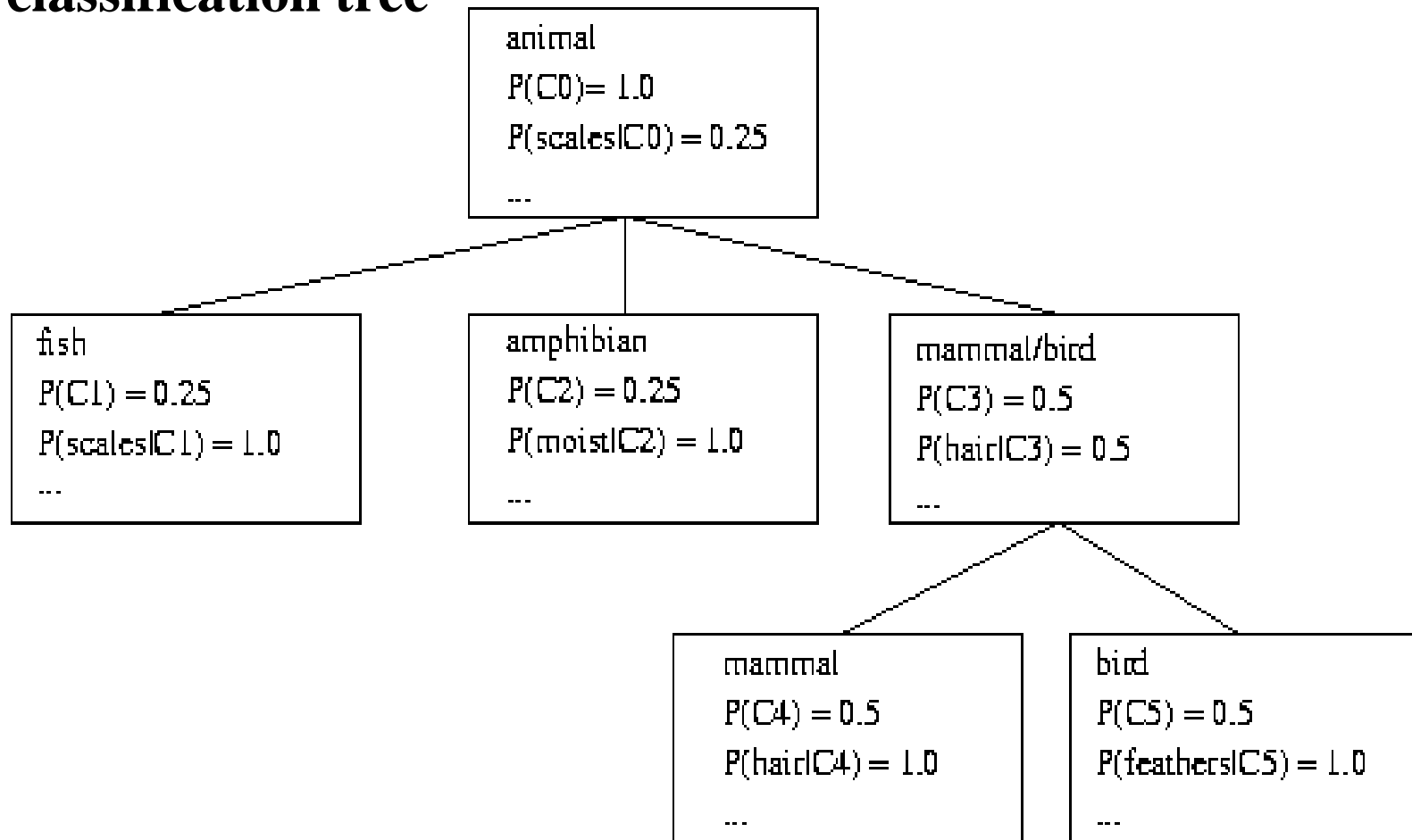
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# MODEL-BASED CLUSTERING METHODS

- Attempt to optimize the fit between the data and some mathematical model
- Statistical and AI approach
  - Conceptual clustering
    - A form of clustering in machine learning
    - Produces a classification scheme for a set of unlabeled objects
    - Finds characteristic description for each concept (class)
  - COBWEB (Fisher'87)
    - A popular a simple method of incremental conceptual learning
    - Creates a hierarchical clustering in the form of a **classification tree**
    - Each node refers to a concept and contains a probabilistic description of that concept

# COBWEB CLUSTERING METHOD

## A classification tree



# MORE ON STATISTICAL-BASED CLUSTERING

- Limitations of COBWEB
  - The assumption that the attributes are independent of each other is often too strong because correlation may exist
  - Not suitable for clustering large database data – skewed tree and expensive probability distributions
- CLASSIT
  - an extension of COBWEB for incremental clustering of continuous data
  - suffers similar problems as COBWEB
- AutoClass (Cheeseman and Stutz, 1996)
  - Uses Bayesian statistical analysis to estimate the number of clusters
  - Popular in industry



# OTHER MODEL-BASED CLUSTERING METHODS

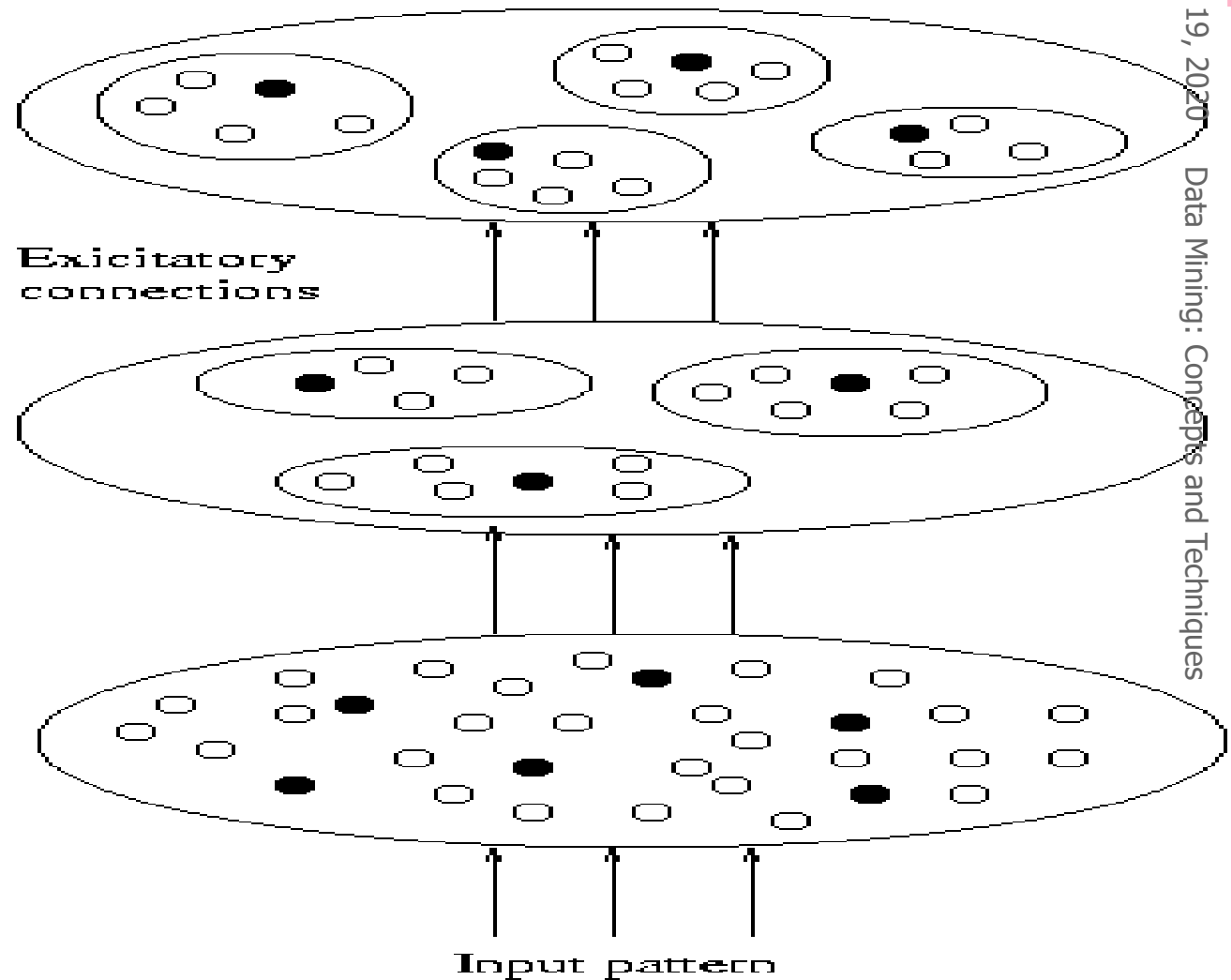
- Neural network approaches
  - Represent each cluster as an exemplar, acting as a “prototype” of the cluster
  - New objects are distributed to the cluster whose exemplar is the most similar according to some distance measure
- Competitive learning
  - Involves a hierarchical architecture of several units (neurons)
  - Neurons compete in a “winner-takes-all” fashion for the object currently being presented

# MODEL-BASED CLUSTERING METHODS

**Layer 3**  
**Inhibitory**  
**clusters**

**Layer 2**  
**Inhibitory**  
**clusters**

**Layer 1**  
**Input units**



# SELF-ORGANIZING FEATURE MAPS (SOMs)

- Clustering is also performed by having several units competing for the current object
- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

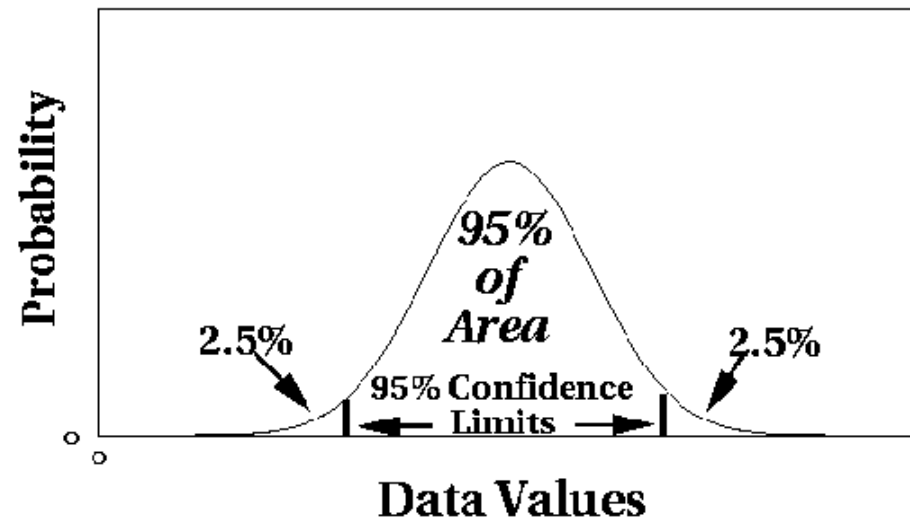
# CHAPTER 8. CLUSTER ANALYSIS

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# WHAT IS OUTLIER DISCOVERY?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Example: Sports: Michael Jordon, Wayne Gretzky, ...
- Problem
  - Find top n outlier points
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis

# OUTLIER DISCOVERY: STATISTICAL APPROACHES



- Assume a model underlying distribution that generates data set (e.g. normal distribution)
- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers
- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known

# OUTLIER DISCOVERY: DISTANCE-BASED APPROACH

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution.
- Distance-based outlier: A  $DB(p, D)$ -outlier is an object  $O$  in a dataset  $T$  such that at least a fraction  $p$  of the objects in  $T$  lies at a distance greater than  $D$  from  $O$
- Algorithms for mining distance-based outliers
  - Index-based algorithm
  - Nested-loop algorithm
  - Cell-based algorithm



# OUTLIER DISCOVERY: DEVIATION-BASED APPROACH

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that “deviate” from this description are considered outliers
- sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data



# CHAPTER 8. CLUSTER ANALYSIS

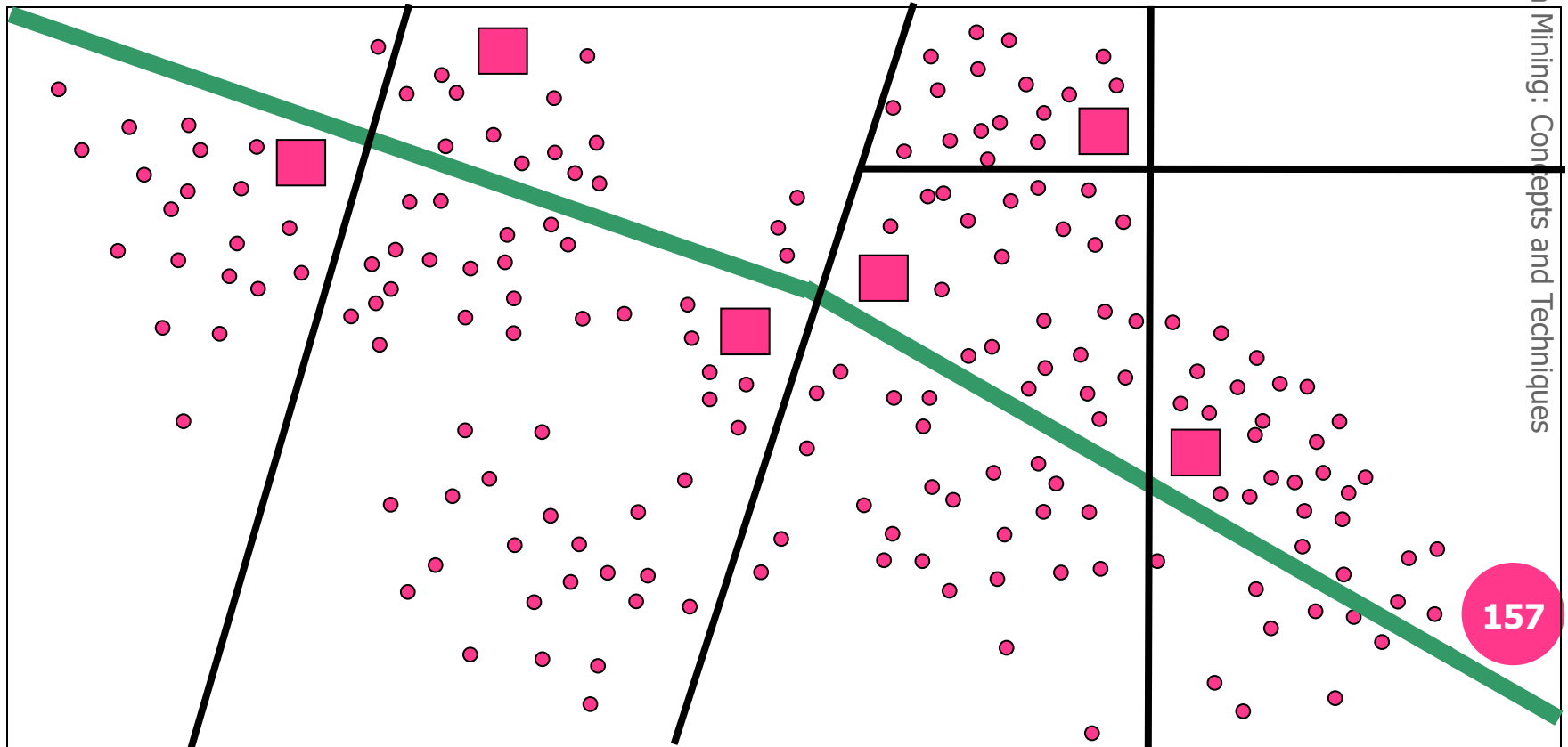
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

# PROBLEMS AND CHALLENGES

- Considerable progress has been made in scalable clustering methods
  - Partitioning: k-means, k-medoids, CLARANS
  - Hierarchical: BIRCH, CURE
  - Density-based: DBSCAN, CLIQUE, OPTICS
  - Grid-based: STING, WaveCluster
  - Model-based: Autoclass, Denclue, Cobweb
- Current clustering techniques do not address all the requirements adequately
- Constraint-based clustering analysis: Constraints exist in data space (bridges and highways) or in user queries

# CONSTRAINT-BASED CLUSTERING ANALYSIS

- Clustering analysis: less parameters but more user-desired constraints, e.g., an ATM allocation problem



# SUMMARY

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis, such as constraint-based clustering

# REFERENCES (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scientific, 1996
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.

# REFERENCES (2)

- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.
- G. J. McLachlan and K.E. Bkaford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. Future Generation Computer systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition, 101-105.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.



Thank you !!!