Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-1
Introduction to **Data Mining (DM)**

Prof. Naimish R. Vadodariya
Computer Engineering Department
Darshan Institute of Engineering & Technology, Rajkot
naimish.vadodariya@darshan.ac.in
8866215253

Topics to be covered

- Motivation for Data Mining
- Data Mining - Definition and Functionalities
- Data Mining – On what kind of data?
- KDD Process (**K**nowledge **D**iscovery in **D**atabases)
- DM task primitives
- Classification of DM (Data Mining) Systems
- Issues in DM

Motivation for Data Mining

Section - 1

Just think: One Second on Internet

- 9,003 Tweets
- 4,705 Skype Calls
- 1,711 Tumblr Posts
- 83,378 Google Searches
- 84,388 YouTube videos viewed
- 996 Instagram photos uploaded
- & many more…

Are all these information is really important to us

?????????

Motivation: Why data mining?

- **"Necessity is the Mother of all Inventions"**
- "It has been estimated that the amount of **information** in the world **doubles** every **10** months."
- There is a tremendous increase in the amount of data recorded and stored on digital media as well as individual sources.

- Since the 1960's, database and information technology has been changed systematically from primitive file processing systems to powerful database systems.
- The research and development in database systems since the 1970's has led to the development of relational database systems.

"We are drowning in data, but starving for knowledge!"
"Data rich but Information poor"

Motivation: Why data mining? (Cont..)

Years	Evolutions
Since 1960's	Data collection, database creation, IMS (hierarchical database system by IBM) and network DBMS

Every day data **grows exponentially**,
but these **all data** are really **important to us**??

Motivation for Data Mining : An Example

Data → Knowledge → Action → Goal

Netflix collects user ratings of movies (**data**) → What types of movies you will like (**knowledge**) → Recommend new movies to you (**action**) → Users stay with Netflix (**goal**)

Gene sequences of cancer patients (**data**) → Which genes lead to cancer? (**knowledge**) → Appropriate treatment (**action**) → Save life (**goal**)

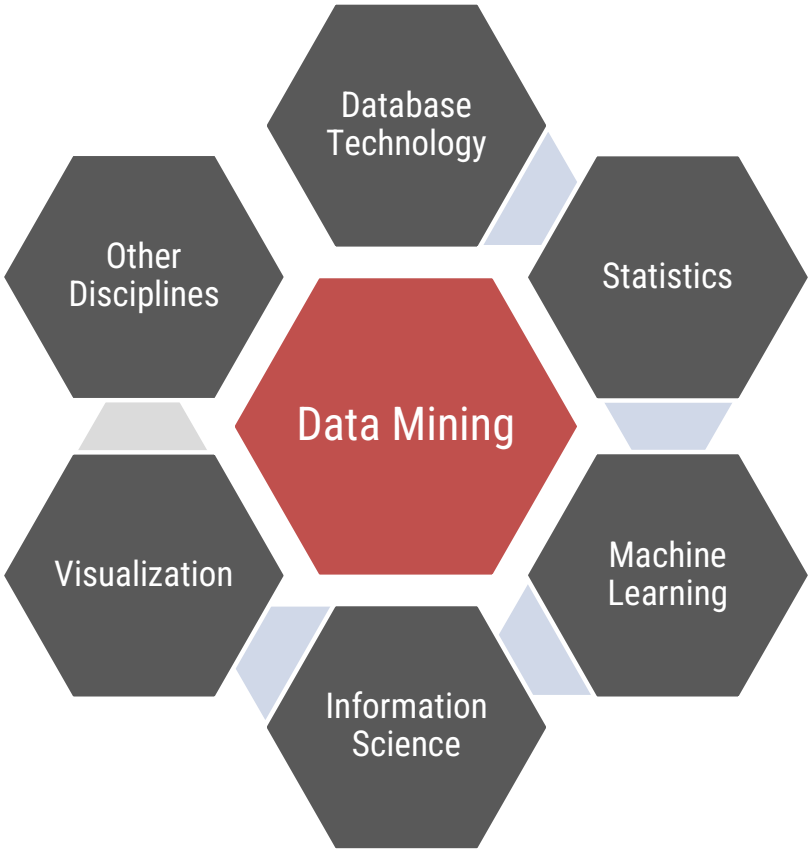
Road traffic **(data)** → Which road is likely to be congested? **(knowledge)** → Suggest better routes to drivers **(action)** → Save time and energy **(goal)**

Summary

The overall goal of the data mining process is to **extract information from a large data sets** or **databases** and **transform it into an understandable structure for further use**.

Data Mining Definition & Functionalities

Section - 2



What is Data Mining?

- ✓ Data mining refers to extracting or “mining” knowledge from large amounts of data.
- ✓ “Knowledge mining from data” or “Knowledge mining”
- ✓ “Extract knowledge from large data or databases”
- ✓ “Knowledge discovery from database (KDD)”

Data Mining Functionalities

- Data mining functionalities can be classified into two categories:
  1. **Descriptive**
  2. **Predictive**

- **Descriptive**
  - This task presents the general properties of data stored in a database.
  - The descriptive tasks are used to find out patterns in data.
  - **E.g.**: Cluster, Trends, etc.

- **Predictive**
  - These tasks predict the value of one attribute on the basis of values of other attributes.
  - **E.g.**: Festival Customer/Product Sell prediction at store

Data Mining – On what kind of data?

Section - 3

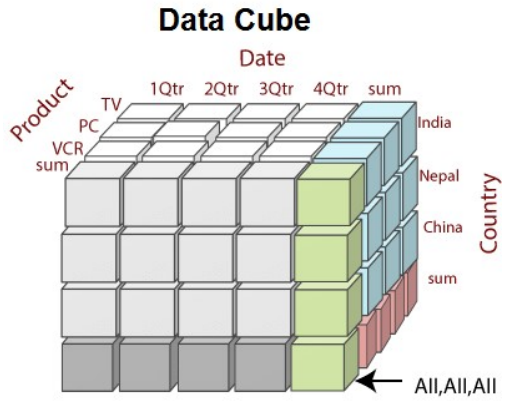
Data Mining—On what kind of data?

Relational Databases:

- A database system, also called a database management system (DBMS), consists of a collection of interrelated data, known as a database tables, and a set of software programs to manage and access these data.
- **E.g.** : SQL Server, Oracle etc.

Data Warehouses:

- A data warehouse is a repository of information collected from multiple sources.
- It is constructed after pre-processing of data. (Data cleaning, Data integration, Data transformation, Data loading, and Periodic data refreshing etc.)
- **E.g.** : Stock Market, D-Mart, Big Bazar etc.



Data Mining—On what kind of data? (Cont..)

Transactional Databases:

- Transactional database consists of a file where each record represents a transaction.
- A transaction typically includes a unique transaction identity number (TID) and a list of the items making up the transaction (such as items purchased in a store).
- **E.g.** : Online shopping on Flipkart, Amazon etc.

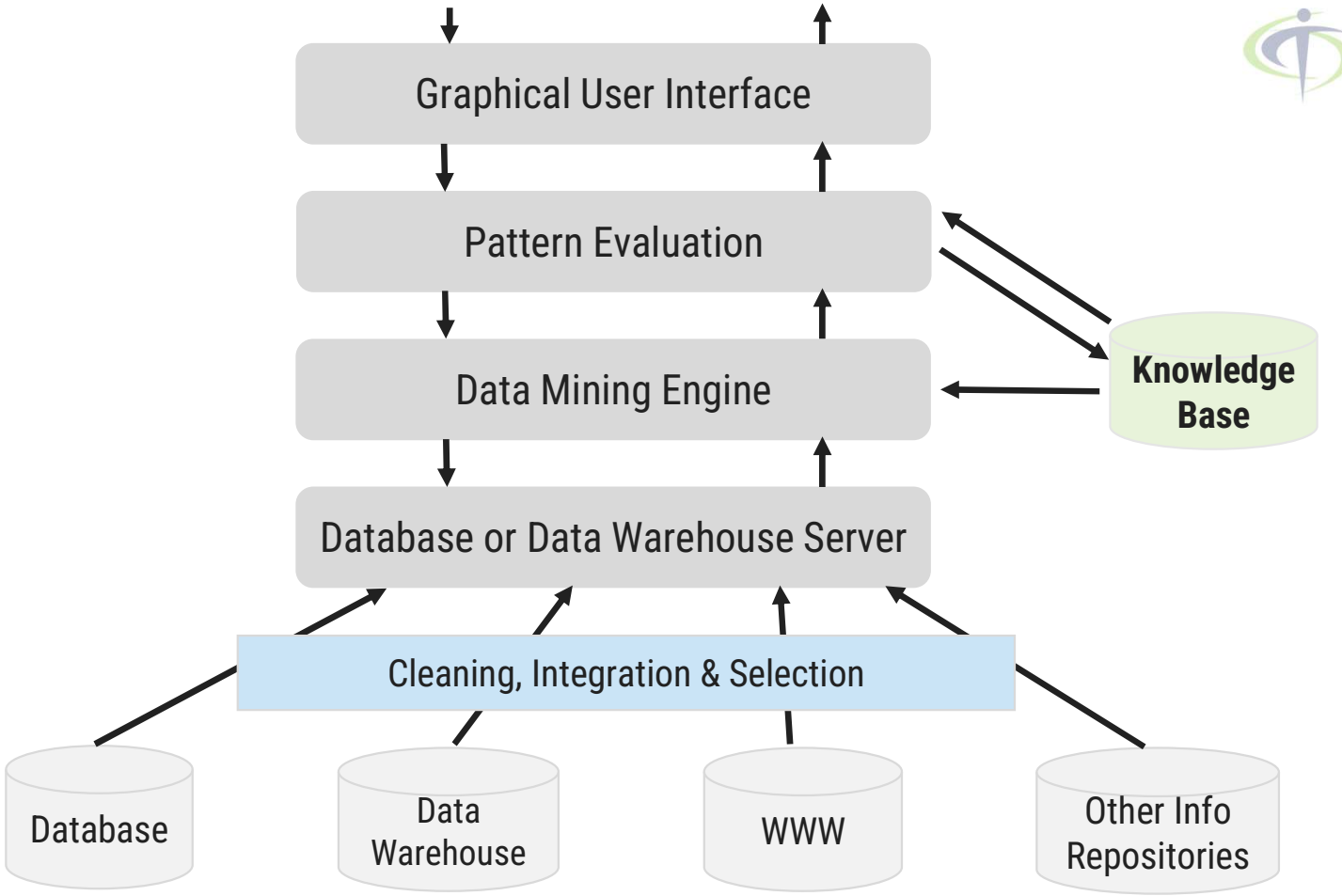
Other Data/Databases

- Spatial data (Maps or Location related data)
- Engineering design data (Designs of Buildings, Offices Structures data)
- Hypertext and multimedia data (Including text, image, video and audio data), the World Wide Web (WWW a huge, widely distributed information repository made available on the Internet).

KDD Process

Section - 4

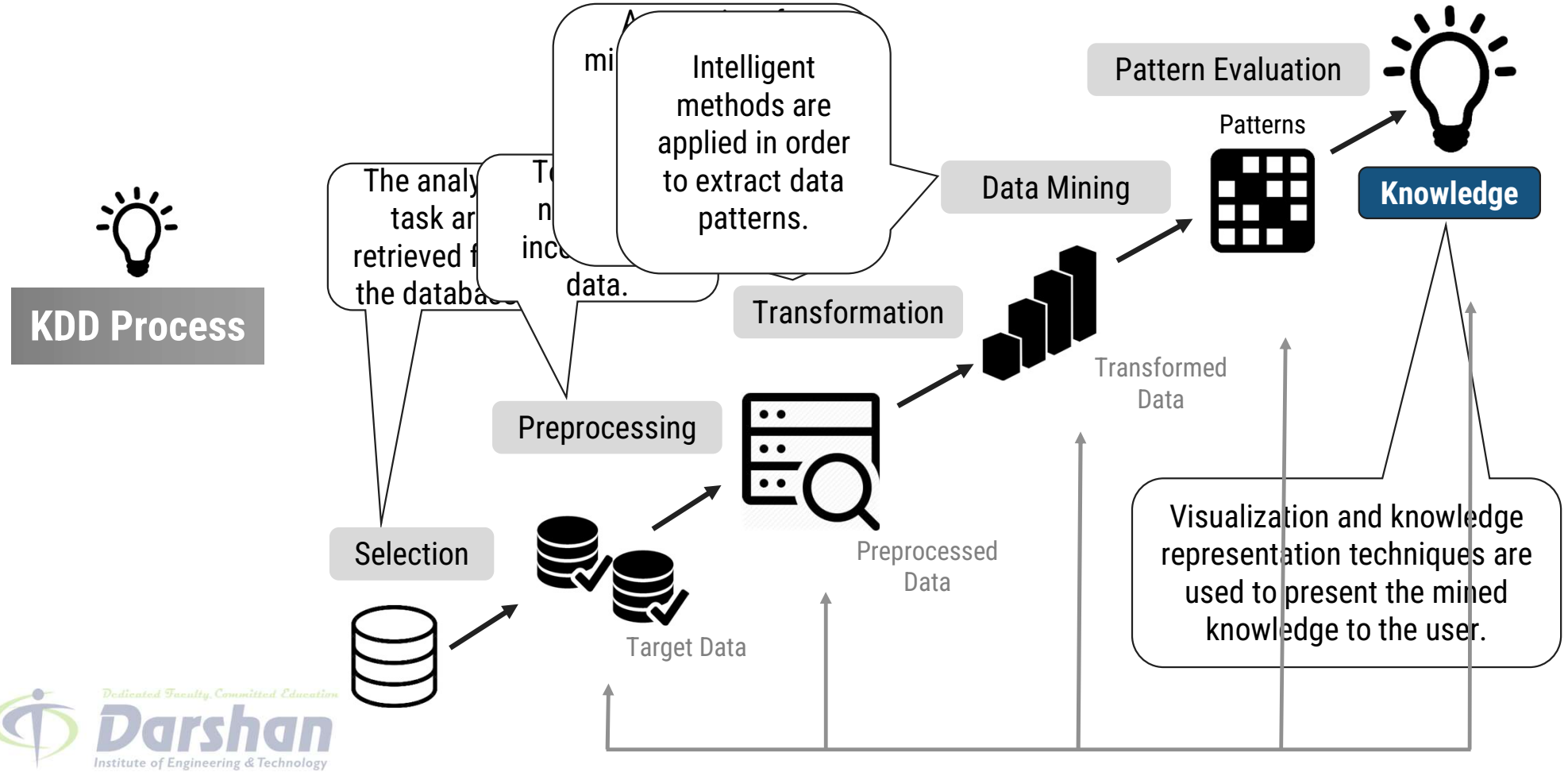
Data Mining Architecture



KDD (Knowledge Discovery in Databases) Process

- Knowledge discovery in databases is a process of an iterative sequence of the following steps:
  1. **Selection**
  2. **Preprocessing**
  3. **Transformation**
  4. **Data Mining**
  5. **Pattern Evaluation**
  6. **User Interface (Visualization of Pattern or Knowledge)**

KDD (Knowledge Discovery in Databases) Process (Cont..)



KDD (Knowledge Discovery in Databases) Process (Cont..)

- **Data Selection:** Where data relevant to the analysis task are retrieved from the database.
- **Data Cleaning:** To remove noise and inconsistent data.
- **Data Integration:** Where multiple data sources may be combined.
- **Data Transformation**: Where data are transformed or consolidated into appropriate forms for mining by performing summary or aggregation operations.
- **Data Mining**: An essential process where intelligent methods are applied in order to extract data patterns.
- **Pattern Evaluation**: To identify the truly interesting patterns representing knowledge based on some interestingness measures.
- **Knowledge Presentation**: Where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

Data Mining Task Primitives

Section - 5

Data Mining Task Primitives

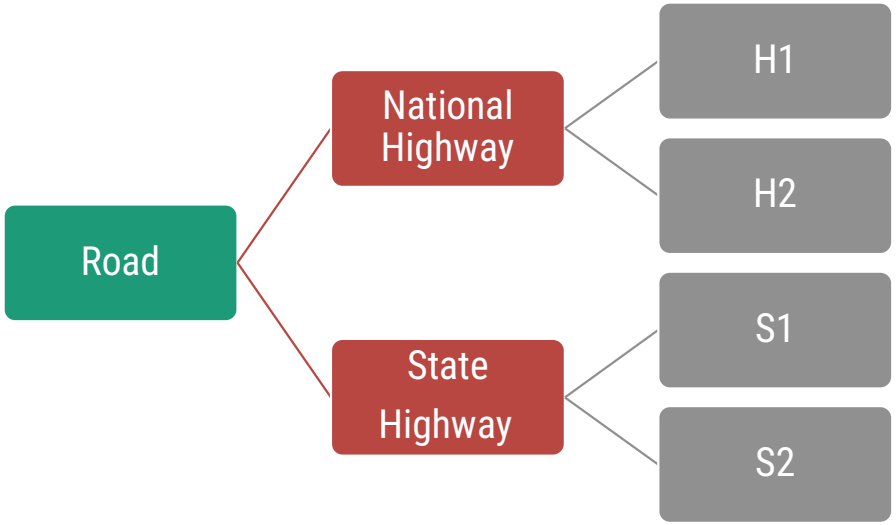
- Primitive (જૂના વખતનું સાદું Or પ્રાચીન ઢબનું)
- A data mining task can be specified in the form of a data mining query, which is input to the data mining system.
- A data mining query is defined in terms of data mining task primitives.
- These primitives allow the user to **inter- actively** communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths.
- The **data mining primitives** specify the following
  - ✓ The set of task-relevant data to be mined
  - ✓ The kind of knowledge to be mined
  - ✓ The background knowledge to be used in the discovery process
  - ✓ The interestingness measures and thresholds for pattern evaluation
  - ✓ The expected representation for visualizing the discovered patterns

Data Mining Task Primitives (Cont..)

- The set of task-relevant data to be mined
  - This specifies the portions of the database or the set of data in which the user is interested (Target Data)
  - This includes the database attributes or data warehouse dimensions of interest
- The kind of knowledge to be mined
  - This specifies the data mining functions to be performed, such as
    - ✓ **Characterization**: Summarization of the general characteristics or features of a target class of data**.**
    - ✓ **Association**: It discovers the probability of the co-occurrence of items in a collection.
    - ✓ **Correlation analysis**: It is used to find the association between the variables.
    - ✓ **Classification**: It discovers a model that defines the data classes or concepts.
    - ✓ **Prediction:** It represents the data classes to predict future data/trends.
    - ✓ **Cluster analysis**: To find out the group of objects which are similar to each other in the group but are different from the object in other groups.
    - ✓ **Outlier analysis**: It is a process that involves identifying the anomalous observation in the dataset

Data Mining Task Primitives (Cont..)

- The background knowledge to be used in the discovery process
  - This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found.
  - Concept hierarchies (It defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts) are a popular form of back- ground knowledge, which allow data to be mined at multiple levels of abstraction.



Data Mining Task Primitives (Cont..)

- The interestingness measures and thresholds for pattern evaluation
  - They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns.
  - Different kinds of knowledge may have different interestingness measures
  - For example,
    - Interestingness measures for association rules include support and confidence.
    - Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.

ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
…	…

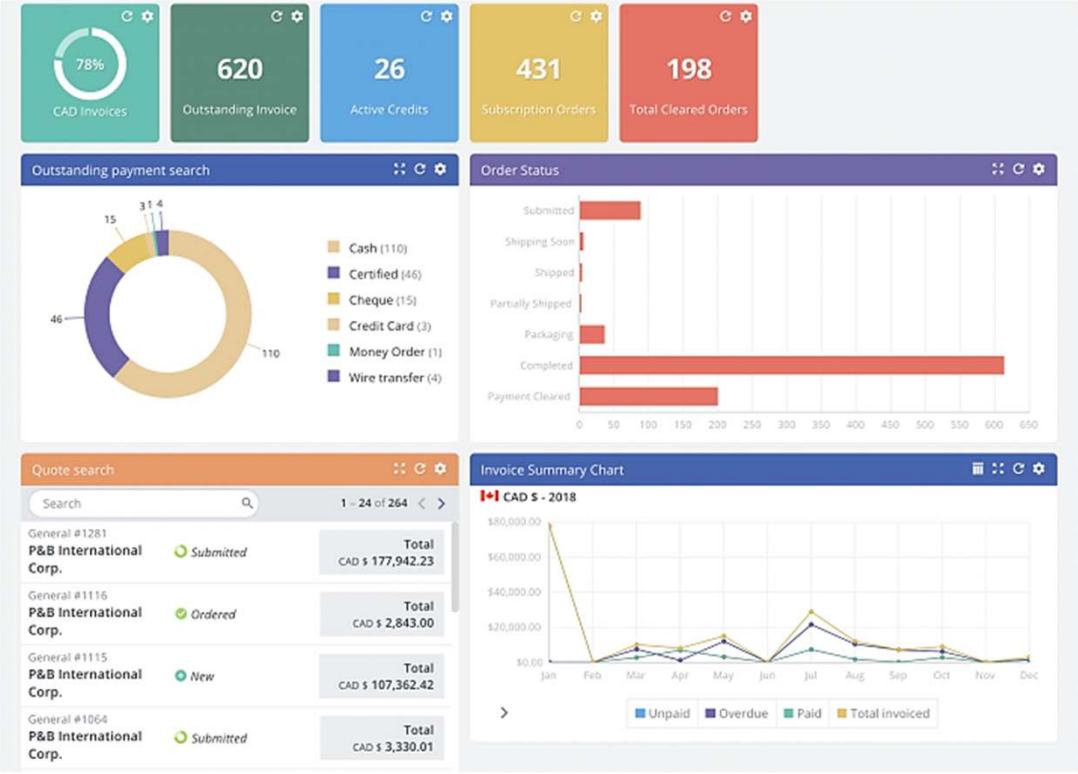
market basket transactions

{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule

Data Mining Task Primitives (Cont..)

- The expected representation for visualizing the discovered patterns
  - This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.



Classification of Data Mining Systems

Section - 6

Classification of Data Mining Systems

- Classification of data mining based on..
  1. **Databases** to be mined
  2. **Knowledge** to be mined
  3. **Techniques/Methods** utilized
  4. **Application** adapted

1. Classification according to the kinds of Databases mined

- Database models are important for classification according to the kinds of databases to be mined.
- Types of database models
  - Hierarchical database model
  - Relational model
  - Network model
  - Object-oriented database model
  - Entity-relationship model
  - Document model
  - Entity-attribute-value model
  - Star schema
  - Object-relational model

Classification of Data Mining Systems

1. Classification according to the kinds of Databases mined (Cont..)

- It can be classified as a 'type of data' or 'use of data' model or 'application of data'.
- Classified according to different criteria (such as data models, or the types of data or applications involved), each of which may require its own data mining technique.
- For instance, if classifying according to data models, we may have a relational, transactional, object-oriented, object-relational, or data warehouse mining system.
- If classifying according to the special data types, we may have a spatial, time-series, text or multimedia data mining system or a world-wide web mining system.
- Other system types include heterogeneous data mining systems and legacy data mining systems.

Classification of Data Mining Systems

2. Classification according to the kinds of Knowledge mined

- Based on data mining functionalities,
  - ✓ **Characterization**: Summarization of the general characteristics or features of a target class of data**.**
  - ✓ **Association**: It discovers the probability of the co-occurrence of items in a collection.
  - ✓ **Correlation analysis**: It is used to find the association between the variables.
  - ✓ **Classification**: It discovers a model that defines the data classes or concepts.
  - ✓ **Prediction:** It represents the data classes to predict future data/trends.
  - ✓ **Cluster analysis**: To find out the group of objects which are similar to each other in the group but are different from the object in other groups.
  - ✓ **Outlier analysis**: It is a process that involves identifying the anomalous observation in the dataset

Classification of Data Mining Systems

3. Classification according to the kinds of Techniques utilized

- These techniques can be described according to the degree of user interaction involved (e.g., autonomous systems, query-driven systems).
- The methods of data analysis employed (e.g., database-oriented or data warehouse–oriented techniques, machine learning, statistics, visualization, pattern recognition, neural networks etc.)
- A sophisticated data mining system will
  - Often adopt multiple data mining techniques for work out an effective
  - Integrated technique which combines the merits of a few individual approaches

Classification of Data Mining Systems

4. Classification according to the Applications adapted

- Retail
- Telecommunication
- Banking
- Fraud analysis
- Stock market analysis
- Text mining
- Web mining etc.

Data Mining Issues

Section - 7

Data Mining Issues

- Data mining issues can be classified into five categories:
  1. **Mining Methodology**
  2. **User Interaction**
  3. **Efficiency and Scalability (Algorithms)**
  4. **Diversity of Database Types**
  5. **Data Mining and Society**

1. Mining Methodology

- **Mining various and new kinds of knowledge**
  - Data mining covers a wide spectrum of data analysis and knowledge discovery tasks, so these tasks may use the same database in different ways and requires a development of numerous data mining techniques.
- **Mining knowledge in multidimensional space**
  - When searching for knowledge in large data sets, we can explore the data in multidimensional space.
  - That is, we can search for interesting patterns among combinations of dimensions (attributes) at varying levels of abstraction. Such mining is known as (exploratory) multidimensional data mining.
- **Data mining—an interdisciplinary effort**
  - The power of data mining can be substantially enhanced by integrating new methods from multiple disciplines.
  - For example, to mine data with natural language text, it makes sense to fuse data mining methods of information retrieval and natural language processing.
- **Handling uncertainty, noise, or incompleteness of data**
  - Data often contain noise, errors, exceptions, uncertainty or incomplete.
  - Errors and noise may confuse the data mining process, leading to the derivation of erroneous patterns.

2. User Interaction

- **Interactive mining**
  - The data mining process should be highly interactive. Thus, it is important to build flexible user interfaces and an exploratory mining environment, facilitating the user's interaction with the system.
- **Incorporation of background knowledge**
  - Background knowledge, constraints, rules, and other information regarding the domain under study should be incorporated into the knowledge discovery process.
- **Presentation and visualization of data mining results**
  - How any system can present data mining results, vividly(clear image in mind) and flexibly ?, so that the discovered knowledge can be easily understood and directly usable by humans.

3. Efficiency and Scalability

- **Efficiency and scalability of data mining algorithms**
  - Data mining algorithms must be efficient and scalable in order to effectively extract information from huge amounts of data lies in many data repositories or in dynamic data streams.
  - In other words, the running time of a data mining algorithm must be predictable, short, and acceptable by applications.
  - Efficiency, scalability, performance, optimization and the ability to execute in real time are key criteria for new mining algorithms.
- **Parallel, distributed, and incremental mining algorithms**
  - The giant size of many data sets, the wide distribution of data, and the computational complexity of some data mining methods are factors that motivate the development of parallel and distributed data-intensive mining algorithms.

4. Diversity of Database Types

- **Handling complex types of data**
  - Data mining is how to uncover knowledge from stream, time-series, sequence, graph, social network and multi-relational data.
  - In mining various types of attributes are available and also different types of data in database or dataset.
- **Mining dynamic, networked, and global data repositories**
  - Data from multiple sources are connected by the Internet and various kinds of networks like distributed and heterogeneous global information systems.
  - The discovery of knowledge from different sources of structured, semi-structured, or unstructured is challengeable.

5. Data Mining and Society

Social impacts of data mining

- With data mining penetrating our everyday lives, it is important to study the impact of data mining on society,
  - How can we used at a mining technology to benefit our society?
  - How can we guard against its misuse?

Privacy-preserving data mining

- Data mining will help in scientific discovery, business management, economy recovery, and security protection (e.g., the real-time discovery of intruders and cyber attacks).
- However, it poses the risk of disclosing an individual's personal information.

Invisible data mining

- We cannot expect everyone in society to learn and master in data mining techniques.
- For example, when purchasing items online, users may be unaware that the store is likely collecting data on the buying patterns of its customers, which may be used to recommend other items for purchase in the future.

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit – 1
Introduction to
Data Mining (DM)

Thank You

Any Questions ?

Prof. Naimish R. Vadodariya
Computer Engineering
Darshan Institute of Engineering & Technology, Rajkot
naimish.Vadodariya@darshan.ac.in
8866215253

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-2
Data Pre-processing

Prof. Naimish R. Vadodariya
Computer Engineering Department
Darshan Institute of Engineering & Technology, Rajkot
naimish.vadodariya@darshan.ac.in
8866215253

Topics to be covered

- Why to pre-process data?
- Mean, Median, Mode, Range & Standard Deviation
- Attribute Types
- Data Summarization
- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Why to pre-process data?

Section - 1

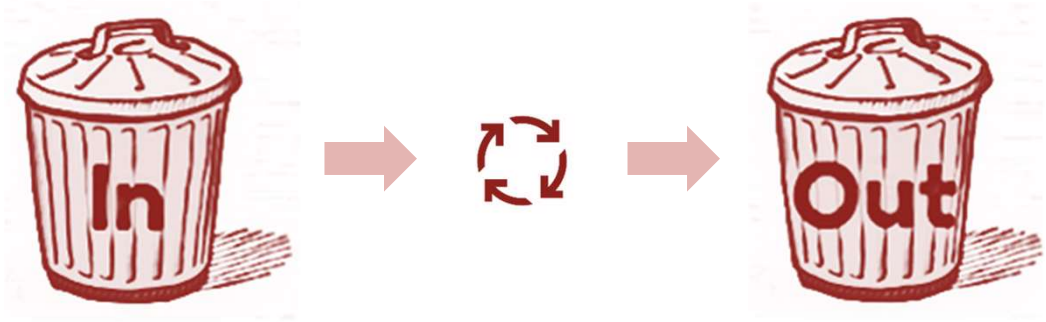
Why to pre-process data?

- Data pre-processing is a data mining technique that involves transforming raw data (real world data) into an understandable format.
- Real-world data is often incomplete, inconsistent, lacking in certain behaviors or trends and likely to contain many errors.
  - **Incomplete**: Missing attribute values, lack of certain attributes of interest, or containing only aggregate data.
    - E.g. Occupation = " "
  - **Noisy**: Containing errors or outliers.
    - E.g. Salary = "abcxy"
  - **Inconsistent**: Containing similarity in codes or names.
    - E.g. "Gujarat" & "Gujrat" (Common mistakes like spelling, grammar, articles)

Why to pre-process data? (Cont..)

No quality data, No quality results

- It looks like Garbage In Garbage Out (GIGO).



- Quality decisions must be based on quality data.
- Duplicate or missing data may cause incorrect or even misleading statistics.
- Data preprocessing **prepares** raw data for **further processing**.

Data preparation, cleaning and transformation are the majority task (90%) in data mining.

Mean, Median, Mode, Range & Standard Deviation

Section - 2

Mean (Average)

Mean is the average of a dataset

- Mean is the **average** of a dataset.
- The mean is the total of all the values, divided by the number of values.
- Formula to find mean $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x$
- Example
  - Find out mean for **12, 15, 11, 11, 7, 13** **(Here total data is = 6)**

First, find the **sum of the data**.

12 + 15 +11 + 11 + 7 + 13 = **69**

Then **divide by the total number of data**.

69 / 6 = **11.5** **(Mean)**

Median {Centre Or Middle Value}

- The median is the middle number in a list of numbers ordered from lowest to highest.

If count is **Odd** then **middle number** is **Median**

- Example
  - Find out Median for 12, 15, 11, 11, 7, 13, 15 **(Here total data is = 7 {odd})**

First, arrange the data in ascending order.

7, 11, 11, 12, 13, 15, 15

Partitioning data into equal half's

7, 11, 11, 12, 13, 15, 15

12 ⇦ Median

Median {Centre Or Middle Value} (Cont..)

If count is **Even** then take **average (mean) of middle two numbers** that is **Median**

- Example
  - Find out Median for 12, 15, 11, 11, 7, 13 **(Here total data is = 6 {even})**

First, arrange the data in ascending order.

7, 11, 11, 12, 13, 15

Calculate an average (mean) of the two numbers in the middle.

7, 11, 11, 12, 13, 15

(11 + 12)/2 = **11.5** ⇦ **Median**

Mode

- The mode is the number that occurs most often within a set of numbers.
- Example

12, 15, 11, 11, 7, 13	12, 12 15, 11, 11, 7, 13, 7
11 ⇦ Mode (Unimodal)	**7, 11, 12 ⇦ Mode (Trimodal)**
12, 15, 11, 11, 7, 12, 13	12, 15, 11, 10, 7, 14, 13
11, 12 ⇦ Mode (Bimodal)	**No Mode**

- If more than three numbers repeats within a set of numbers then it is called as **multimodal**.

Range

- The range of a set of data is the difference between the largest and the smallest number in the set.
- Example
  - Find the range for given data 40, 30, 43, 48, 26, 50, 55, 40, 34, 42, 47, 50

First, arrange the data in ascending order.

26, 30, 34, 40, 40, 42, 43, 47, 48, 50, 50, 55

  - In our example largest number is 55, and subtract the smallest number is 26.

55 – 26 ⇐ **29**

Range

Standard Deviation (σ)

- The Standard Deviation is a measure of how numbers are spread out.
- Its symbol is **σ** (the Greek letter sigma)
- In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values.
- A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.
- Formula to find standard deviation **σ** = $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X - Mean)^2}$

Standard Deviation (σ) Cont..

- Standard Deviation is Square root of sample variance.
- The Variance is defined as:
  - The average of the **squared** differences from the Mean.
- To calculate the variance follow these steps:
  1. Calculate the **mean**, $\bar{x}$.
  2. Write a table that **subtracts the mean from each observed value**.
  3. Square each of the differences, add this column in table.
  4. Divide by **n -1** where **n** is the number of items in the **sample**, this is the **variance** (In actual case take n).
  5. To get the **standard deviation** we take the square root of the variance.

Standard Deviation (σ) Cont..

- The owner of the Indian restaurant is interested in how much people spend at the restaurant.
- He examines **8** randomly selected receipts for parties and writes down the following data.

44, 50, 38, 96, 42, 47, 40, 39

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X - Mean)^2}$$

1. Find out Mean (Mean is **49.5** for given data)
2. Write a table that **subtracts the mean from each observed value**. (2nd step)

Step : 3

X	X - Mean	Value	(X – Mean)^2
44	44 - 49.5	-5.5	30.25
50	50 - 49.5	0.5	0.25
38	38 - 49.5	11.5	132.25
96	96 - 49.5	46.5	2162.25
42	42 - 49.5	-7.5	56.25
47	47 - 49.5	-2.5	6.25
40	40 - 49.5	-9.5	90.25
39	39 - 49.5	-10.5	110.25
		Total	**2588**

Step : 4

$$= \frac{2588}{8-1}$$

$$\mathbf{S^2 = 69.71}$$

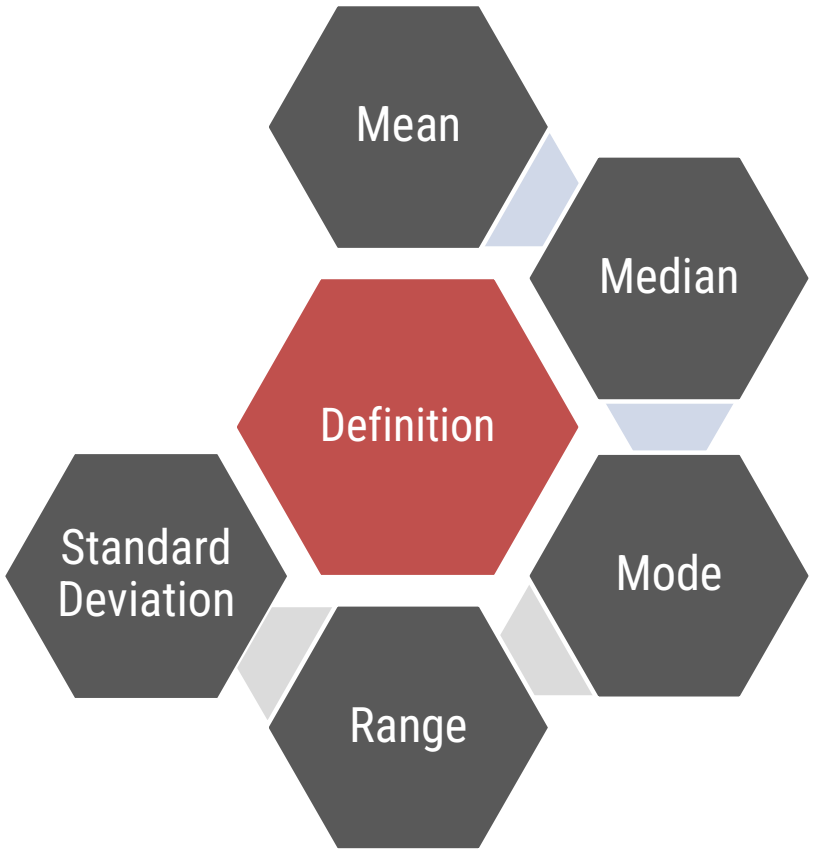
Step : 5

$$\sigma = \sqrt{69.71}$$

$$\boldsymbol{\sigma = 8.34 \sim 8}$$

Standard Deviation (σ) Cont..

- Standard deviation can be thought of measuring how far the data values lie from the mean, we take the mean and move on standard deviation in either direction.
- The mean for this example is **μ = 49.5** and the standard deviation is **σ = 8**.
- Now, we add & subtract values with mean like 49.5 - 8 = 41.5 and 49.5 + 8 = 57.5
- This means that most of the data probably spend between **41.5** and **57.5**.
  - ↪ 38, 39, 40, **42, 44, 47, 50**, 96
- **If all data are same then variance & standard deviation is 0 (zero).**



Summary

- **Mean**: Mean is the **average** of a dataset
- **Median**: Median is the **middle number** in a dataset when the data is arranged in numerical order (Sorted Order).
- **Mode**: The mode is the **number that occurs most often** within a set of numbers.
- **Range**: The range of a set of data is the **difference** between the **largest and the smallest number** in the set.
- **Standard Deviation**: The Standard Deviation is a **measure of how numbers are spread out in dataset**.

Attribute Types

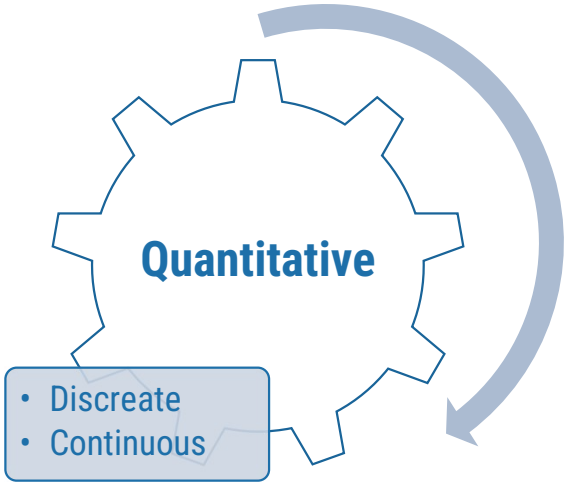
Section - 3

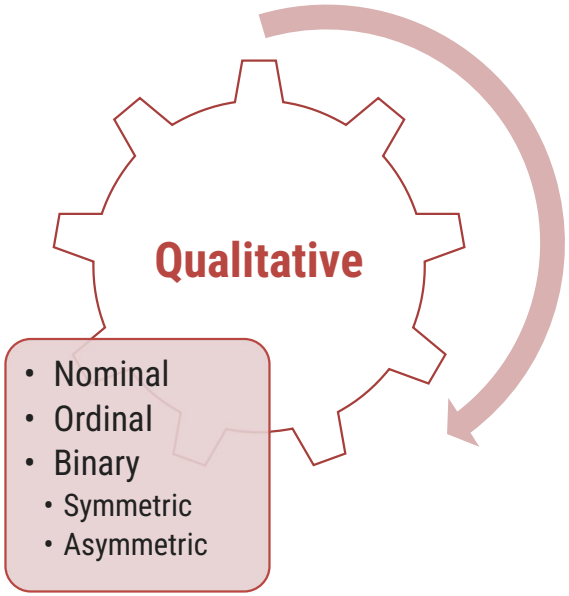
What is an Attribute?

- The attribute can be defined as a field for storing the data that represents the characteristics of a data object.
- It can also be viewed as a property, characteristics, feature or column of a data object.
- It represents the different features of an object (real world entity) like..
  - **Person** → Name, Age, Qualification, Birthdate etc.
  - **Computer** → Brand, Model, Processor, RAM etc.
  - **Book** → Book Name, Author, Price, ISBN etc.
- An attribute set defines an object.
- The object is also referred to as a record of the instances or entity.

Attribute Types

- Attribute types can be divided into mainly two categories.
  1. **Quantitative**
     1. Discrete
     2. Continuous
  2. **Qualitative**
     1. Nominal
     2. Ordinal
     3. Binary
        1. Symmetric
        2. Asymmetric





1. Quantitative Attribute

- Quantitative is an adjective that simply means something **that can be measured**.
- It is a special attribute that is used to compare two values, i.e., it is used to compare a user-defined value against an upper limit and a lower limit.
- Example
  - We can count the number of sheep on a farm or measure the liters of milk produced by a cow.
  - Consider a query to find all patients with low or high blood glucose levels. In database, for each patient a lower value and an upper value for blood glucose level is stored in the Result class.
  - To find patients with low/high level of blood glucose, without QA you would have to specify a limit on the Low attribute or the High attribute of the Result class.
  - While defining limit you can use Between, Equals, Less than, Less than or Equal to, Greater than, Greater than or Equal as relational operators.

1. Quantitative Attribute

1) Discrete Attribute

- A discrete attribute has a finite or countably infinite set of values, which may or may not be represented as integers.
- The attributes hair_color, smoker, medical_test, and drink_size each have a finite number of values, and so are discrete.
- CustomerID in a table has countably infinite set of values because over a time period it grows.

2) Continues Attribute

- Real numbers as attribute values.
- The attributes temperature, height, or weight are the examples of continuous attributes.
- Practically, real values can only be measured and represented using a finite number of digits.
- Continuous attributes are typically represented as **floating- point variables**.

2. Qualitative Attribute

- Qualitative data deals with characteristics and descriptors that can't be easily measured, but can be observed subjectively—such as smells, tastes, textures, attractiveness, and color.
- Simple arithmetic attributes that is named or described in words.
- It is represented in integer or real values.
- Results of qualitative attribute are often quoted on scales.
- Below are the qualitative Attributes.
  - Nominal
  - Ordinal
  - Binary
    - Symmetric
    - Asymmetric

2. Qualitative Attribute Cont..

1) Nominal Attribute

- Nominal attributes are **named** attributes which can be **separated into discrete (individual) categories** which do not overlap.
- Nominal attributes values also called as **distinct values**.
- Example

What is your gender?
Male
Female
Other

What is your hair color?
Black
Brown
Gray
Blonde
Other

2. Qualitative Attribute Cont..

2) Ordinal Attribute

- Ordinal attribute is the **order of the values,** that's important and significant, but the differences between each one is not really known.
- Example
  - **Rankings** → 1st, 2nd, 3rd
  - **Ratings** → ★ ★ ★ , ★ ★ ★ ★ ★
- We know that a 5 star is better than a 2 star or 3 star, but we don't know and cannot quantify–how much better it is?

3) Binary Attribute

- Binary attributes are the categorical attributes with only two possible values (yes or no), (true or false), (0 or 1).
- Symmetric binary attribute is the attribute which each value is equally valuable (male or female). The male here is not more important than the female value.
- Asymmetric is the attribute which the two states is not equally important, for example, the medical test (positive or negative), here, the positive results is more significant than the negative one.

Interval Attribute

- Interval attribute comes in the form of a numerical value where the difference between points is meaningful.
- Example
  - **Temperature** → 10°-20°, 30°-50°, 35°-45°
  - **Calendar Dates** → 15^{th} – 22^{nd}, 10^{th} – 30^{th}
- We can not find true zero (absolute) value with interval attributes.

Ratio Attribute

- Ratio attribute is looks **like interval attribute**, but it **must have** a **true zero (absolute)** value.
- It tells us about the order and the exact value between units or data.
- Example
  - **Age Group** → 10-20, 30-50, 35-45 (In years)
  - **Mass** → 20-30 kg, 10-15 kg
- It does have a true zero (absolute) so, it is possible to compute ratios.

Data Summarization

Section - 4

Why Data Summarization?

- As we are living in a digital world where data transfers in a second and it is much faster than a human capability.
- In the corporate field, employees work on a huge volume of data which is derived from different sources like Social Network, Media, Newspaper, Book, cloud media storage etc.
- But sometimes it may create difficulties for you, to summarize the data.
- Sometimes you do not expect data volume because when you retrieve data from relational sources you can not predict that how much data will be stored in the database.
- As a result, data becomes more complex and takes time to summarize information.

What is Data Summarization?

- Summarization is a key data mining concept which involves techniques for finding a compact description of a dataset.
- It is aimed at extracting useful information and general trends from the raw data.
- Two methods for data summarization are through tables and graphs.
  - Tables are row & column representation of the dataset, you can apply aggregate functions on it.
  - Graphs showing the relation between variable quantities, typically of two variables, each measured along one of a pair of axes at right angles.

Data Cleaning

Section - 4

Data Cleaning

1. **Fill in missing values**
   1. Ignore the tuple
   2. Fill missing value manually
   3. Fill in the missing value automatically
   4. Use a global constant to fill in the missing value
2. **Identify outliers and smooth out noisy data**
   1. Binning Method
   2. Regression
   3. Clustering
3. **Correct inconsistent data**
4. **Resolve redundancy caused by data integration**

1) Fill in missing values

- **Ignore the tuple (record/row)**:
  - Usually done when **class label is missing**.
  - **Example**
    - The task is to distinguish between two types of emails, “spam” and “non-spam” (Ham).
    - Spam & non-spam are called as class label.
    - If an email comes to you, in which class label is missing then it is discarded.
- **Fill missing value manually**:
  - Use the **attribute mean (average)** to **fill in the missing value** and **also use the attribute mean (average)** for **all samples belonging to the same class**.
- **Fill in the missing value automatically**:
  - **Predict** the **missing value** by using a **learning algorithm**:
    - Consider the attribute with the missing value as a dependent variable and run a learning algorithm (usually Naive Bayes or Decision tree) to predict the missing value.
- **Use a global constant to fill in the missing value**
  - Replace **all missing attribute values** by the same constant such as a label like **“Unknown”**.

2) Identify outliers and smooth out noisy data

There are three data smoothing techniques as follows..

1. **Binning :**
   - Binning methods smooth a sorted data value by consulting its “neighborhood” that is, the values around it.
2. **Regression :**
   - It conforms data values to a function.
   - Linear regression involves finding the “best” line to fit two attributes (or variables) so that one attribute can be used to predict the other.
3. **Outlier analysis :**
   - Outliers may be detected by clustering for example, where similar values are organized into groups or “clusters”.
   - In this, values that fall outside of the set of clusters may be considered as outliers.

1. Binning Method

- Binning method is a top-down splitting technique based on a specified number of bins.
- In this method the data is first sorted and then the sorted values are distributed into a number of buckets or bins.
- For example, attribute values can be discretized (separated) by applying equal-width or equal-frequency binning, and then replacing each value by the bin mean, median or boundaries.
- It can be applied recursively to the resulting partitions to generate concept hierarchies.
- It does not use class information, therefore it is called as unsupervised discretization technique.
- It used to minimize the effects of small observation errors.

Identify outliers and smooth out noisy data

1. Binning Method Cont..

There are basically two types of binning approaches..

1. Equal width (or distance) binning :

- The simplest binning approach is to partition the range of the variable into k equal-width intervals.
- The interval width is simply the range [Min, Max] of the variable divided by N,
- Width = Max – Min / N (Number of Bins)

- Example
  - **Data**: 5,10,11,13,15, 35, 50, 55, 72, 92, 204, 215
  - As per above formula we have Max=215, Min=5, Number of Bins=3
    - 70+5=75 (from 5 to 75) = **Bin 1**: 5,10,11,13,15, 35, 50, 55, 72
    - 70+75=145 (from 75 to 145) = **Bin 2**: 92
    - 70+145=215 (from 145 to 215) = **Bin 3**: 204, 215

2. Equal depth (or frequency) binning :

- In equal-frequency binning we divide the range [Max, Min] of the variable into intervals that contain (approximately) equal number of points; equal frequency may not be possible due to repeated values.

Identify outliers and smooth out noisy data

1. Binning Method Cont..

Bin Operations

1. **Smoothing by bin means**
   - In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.
2. **Smoothing by bin median**
   - In this method each bin value is replaced by its bin median value.
3. **Smoothing by bin boundary**
   - In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries.
   - Each bin value is then replaced by the closest boundary value.

Identify outliers and smooth out noisy data

Binning Method Example – {Bin Means}

- Given data: **4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34**
- Step: 1
- Partition into **equal-depth [n=4]**:

  Bin 1: 4, 8, 9, 15

  Bin 2: 21, 21, 24, 25

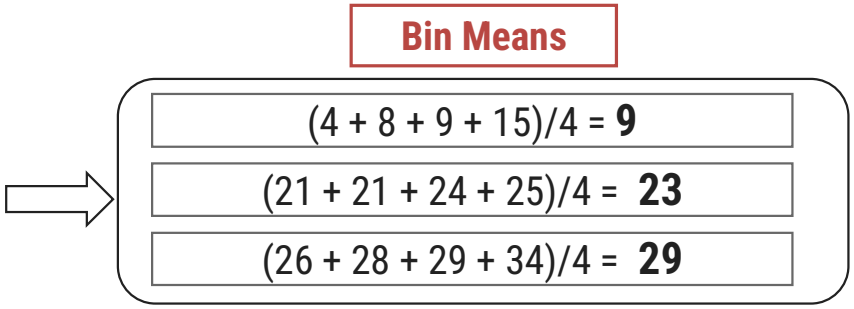
  Bin 3: 26, 28, 29, 34

- Step: 2
  - Smoothing by **bin means**:

    Bin 1: 9, 9, 9, 9

    Bin 2: 23, 23, 23, 23

    Bin 3: 29, 29, 29, 29



Identify outliers and smooth out noisy data

Binning Method Example – {Bin Boundaries}

- Given data: **4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34**
- Step: 1
- Partition into **equal-depth [n=4]**:

  Bin 1: 4, 8, 9, 15

  Bin 2: 21, 21, 24, 25

  Bin 3: 26, 28, 29, 34

- Step: 2
  - Smoothing by **bin boundaries**:

    Bin 1: **4, 4, 4, 15**

    Bin 2: **21, 21, 25, 25**

    Bin 3: **26, 26, 26, 34**

Identify outliers and smooth out noisy data

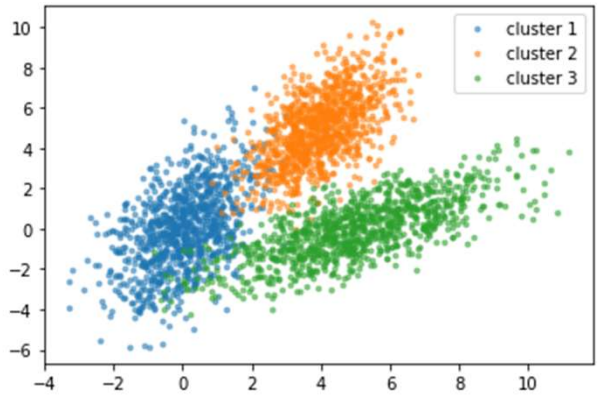
2. Regression

- Data smoothing can also be done by regression, a technique that conforms data values to a function.
- Regression analysis is a way to find trends in data & it is also called as mathematically describes the relationship between independent variables and the dependent variable.
- It can be divided into two categories..
  1. **Linear regression** :
     - It involves finding the "best" line to fit two attributes (or variables) so that one attribute can be used to predict the other.
     - In this, analysis on a single x variable for each dependent "y" variable. For example: (x_1, Y_1).
  2. **Multiple linear regression** :
     - An extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.
     - It uses multiple "x" variables for each independent variable: $(x1)_1, (x2)_1, (x3)_1, Y_1)$.

Identify outliers and smooth out noisy data

3. Clustering

- **Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (**clusters**).
- Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure.
- It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties.



Identify outliers and smooth out noisy data

Correct Inconsistent Data

- With larger datasets, it can be difficult to find all of the inconsistencies.
- It contains similarity in codes or names.
- We can manually solve common mistakes like spelling, grammar, articles or use other tools for it.

Resolve redundancy caused by data integration

- Data redundancy occurs in database systems **which have a field that is repeated in two or more tables**.
- When customer data is duplicated and attached with each product bought, then redundancy of data is known as **inconsistency**.
- So, the entity "customer" **might appear with different values**.
- Database **normalization** prevents redundancy and makes the best possible usage of storage.
- The proper use of **foreign keys** can minimize data redundancy and reduce the chance of destructive anomalies appearing.

Data Integration

Section - 5

Data Integration

- Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources
- Entity identification problem
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - Object identification: The same attribute or object may have different names in different databases
  - Derivable data: One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by correlation analysis and covariance analysis.
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality.

Data Transformation

Section - 6

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values that each old value can be identified with one of the new values
- **Methods**
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - Normalization: Scaled to fall within a smaller, specified range
    - Min-max normalization
    - Z-score normalization
    - Normalization by decimal scaling
  - Discretization: Concept hierarchy climbing

1. Min-Max Normalization

- Min max is a technique that helps to **normalizing the data**.
- It will **scale the data between 0 and 1 or within specified range**.
- Example

Age
16
20
30
40

Formula : $V' = \frac{v - Min_A}{Max_A - MinA}(NewMax_A - NewMin_A) + NewMin_A$

- **Given data**
  - Min : Minimum value = 16
  - Max : Maximum value = 40
  - V = Respective value of attributes. In our example V_1= 16, V_2=20, V_3=30 & V_4=40.
  - NewMax = **1**
  - NewMin = **0**

1. Min-Max Normalization Cont..

Example

Age
16
20
30
40

Formula : $V' = \frac{v - Min_A}{Max_A - MinA}(NewMax_A - NewMin_A) + NewMin_A$

For Age 16 :

MinMax (v') = (16 – 16)/(40-16) * (1 – 0) + 0
= 0 / 24 * 1
= **0**

For Age 30 :

MinMax (v') = (30 – 16)/(40-16) * (1 – 0) + 0
= 14 / 24 * 1
= **0.58**

For Age 20 :

MinMax (v') = (20 – 16)/(40-16) * (1 – 0) + 0
= 4 / 24 * 1
= **0.16**

For Age 40 :

MinMax (v') = (40 – 16)/(40-16) * (1 – 0) + 0
= 24 / 24 * 1
= **1**

Age	After Min-max normalization
16	**0**
20	**0.16**
30	**0.58**
40	**1**

2. Decimal Scaling

- In this technique we move the decimal point of values of the attribute.
- This movement of decimal points totally depends on the **maximum value among all values** in the attribute.
- Value V of attribute A can be normalized by the following formula
- Normalized value of attribute
  - ↪ $V' = \mathbf{V / 10^j}$

Where *j* is the smallest integer such that Max(|v'|) < 1

Example

CGPA	Formula	After Decimal Scaling
2	2 / 10	**0.2**
3	3 / 10	**0.3**

- We will check maximum value among our attribute CGPA.
- Maximum value is 3 so, we can convert it into decimal by dividing with 10. why 10?
- We will count total digits in our maximum value and then put 1.
- After 1 we can put zeros equal to the length of maximum value.
- Here 3 is maximum value and total digits in this value is only 1 so, we will put one zero after 1.

3. Z-Score Normalization

- It is also called zero-mean **normalization**.
- The essence of this technique is the data transformation by the **values** conversation to a common scale where an average number equals zero and a standard deviation is one.
- To find z-score values..

$$v' = \frac{v - \mu_A}{\sigma_A}$$

Where **μ**: Mean, **σ**: Standard deviation

Example

- Let μ = 54,000, σ = 16,000
- Find z-score for 73,600,

$$\frac{73{,}600 - 54{,}000}{16{,}000} = \mathbf{1.225}$$

Z-score for 73600: 1.225

Data Reduction

Section - 7

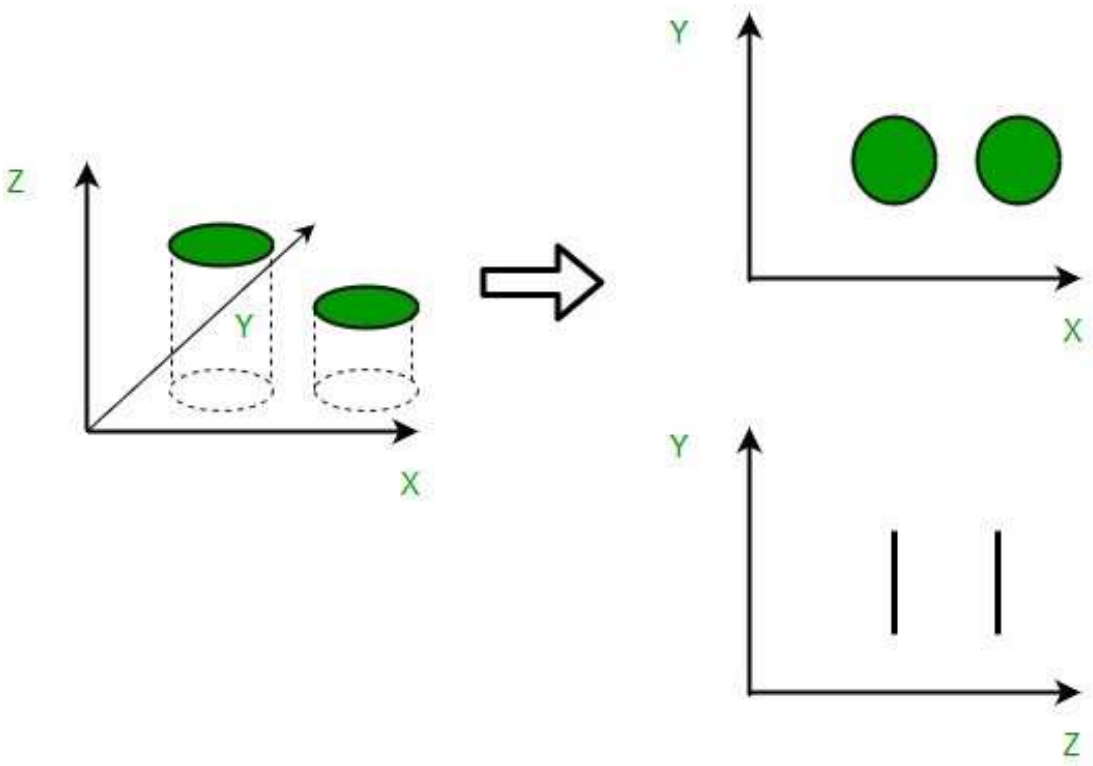
Data Reduction

- Why Data Reduction?
  - A database/data warehouse may store terabytes of data.
  - Complex data analysis may take a very long time to run on the complete data set.
- What is Data Reduction?
  - Data reduction process reduces the size of data and makes it suitable and feasible for analysis.
  - In the reduction process, integrity of the data must be preserved and data volume is reduced.
  - There are many techniques that can be used for data reduction like
    1. Dimensionality reduction
    2. Numerosity reduction
    3. Data compression

1. Dimensionality Reduction

- **Dimensionality reduction**, or **dimension reduction**, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension.
- The number of input variables or features for a dataset is referred to as its dimensionality.
- Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset.
- Example
  - Dimensional reduction can be discussed through a simple e-mail classification problem, where we need to classify whether the e-mail is spam or not.
  - This can involve a large number of features, such as whether or not the e-mail has a generic title, the content of the e-mail, whether the e-mail uses a template, etc.

1. Dimensionality Reduction Cont..



A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2 dimensional space, and a 1-D problem to a simple line.

2. Numerosity Reduction

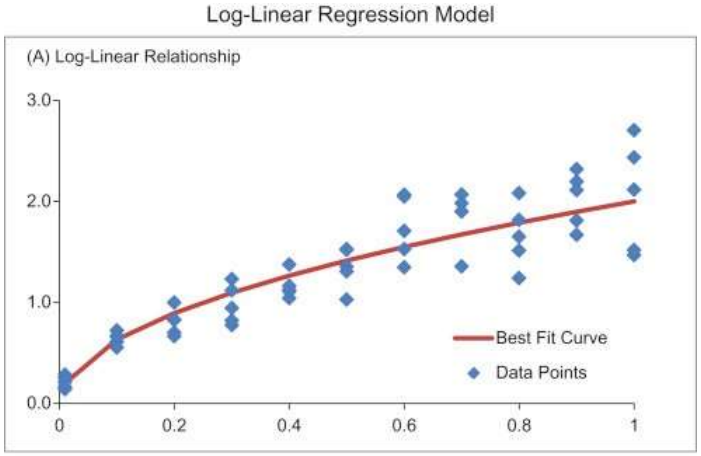
- Numerosity Reduction is a data reduction technique which replaces the original data by smaller form of data representation.
- There are two techniques for numerosity reduction- Parametric and Non-Parametric methods.
- **Parametric Methods**
  - For parametric methods, data is represented using some model.
  - The model is used to estimate the data, so that only parameters of data are required to be stored, instead of actual data.
  - Regression and Log-Linear methods are used for creating such models.
- **Non-Parametric Methods**
  - These methods are used for storing reduced representations of the data include **histograms**, **clustering**, **sampling** and **data cube aggregation**.

Regression

- Regression can be a simple linear regression or multiple linear regression.
- When there is only single independent attribute, such regression model is called simple linear regression and if there are multiple independent attributes, then such regression models are called multiple linear regression.
- In linear regression, the data are modeled to a fit straight line.
- For example, a random variable y can be modeled as a linear function of another random variable x with the equation y = ax+b
- Where a and b (regression coefficients) specifies the slope and y-intercept of the line, respectively.
- In multiple linear regression, y will be modeled as a linear function of two or more predictor (independent) variables.

Log-Linear Model

- Log-linear model can be used to estimate the probability of each data point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations.
- This allows a higher-dimensional data space to be constructed from lower-dimensional attributes.
- Regression and log-linear model can both be used on sparse data (most of the elements are zero), although their application may be limited.



Non-Parametric Methods

- **Histograms**
  - Histogram is the data representation in terms of frequency.
  - It uses binning to approximate data distribution and is a popular form of data reduction.
- **Clustering**
  - Clustering divides the data into groups/clusters, it partitions the whole data into different clusters.
  - In data reduction, the cluster representation of the data are used to replace the actual data, It also helps to detect outliers in data.
- **Sampling**
  - Sampling can be used for data reduction because it allows a large data set to be represented by a much smaller random data sample (or subset).
- **Data Cube Aggregation**
  - Data cube aggregation involves moving the data from detailed level to a fewer number of dimensions.
  - The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

3. Data Compression

- Data Compression is a reduction in the number of bits needed to represent data.
- Compressing data can save storage capacity, speed up file transfer, and decrease costs for storage hardware and network bandwidth.
- Compressing data can be a lossless or lossy process.
  - Lossless compression
    - It enables the restoration of a file to its original state, without the loss of a single bit of data, when the file is uncompressed.
    - Lossless compression is the typical approach with executables, as well as text and spreadsheet files, where the loss of words or numbers would change the information.
  - Lossy compression
    - It permanently eliminates bits of data that are redundant, unimportant or imperceptible.
    - Lossy compression is useful with graphics, audio, video and images, where the removal of some data bits has little or no discernible effect on the representation of the content.

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit – 2
Data
Pre-processing

Thank You

Any Questions ?

Prof. Naimish R. Vadodariya
Computer Engineering
Darshan Institute of Engineering & Technology, Rajkot
naimish.Vadodariya@darshan.ac.in
8866215253

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-3

Concept Description, Mining Frequent Patterns, Associations & Correlations

Prof. Naimish R. Vadodariya
Computer Engineering Department
Darshan Institute of Engineering & Technology, Rajkot
naimish.vadodariya@darshan.ac.in
8866215253

Topics to be covered

- Concept Description
- Market Basket Analysis
- Data Generalization and summarization-based characterization
- Attribute Relevance
- Class Comparisons
- Mining various kind of association rules
- From association mining to correlation analysis
- Advanced Association Rule Techniques
- Measuring the Quality of Rules

What is Concept Description?

Section - 1

What is Concept Description?

- As we know that data mining functionalities can be classified into two categories:
  1. **Descriptive**
  2. **Predictive**

- **Descriptive**
  - This task presents the general properties of data stored in a database.
  - The descriptive tasks are used to find out patterns in data.
  - **E.g.**: Cluster, Trends, etc.

- **Predictive**
  - These tasks predict the value of one attribute on the basis of values of other attributes.
  - **E.g.**: Festival Customer/Product Sell prediction at store

What is Concept Description? Cont..

- Descriptive data mining describes the data set in a concise and summative manner and presents interesting general properties of the data.
- Predictive data mining analyzes the data in order to construct one or a set of models, and attempts to predict the behavior of new data sets.
- Database is usually storing the large amounts of data in great detail. However users often like to view sets of summarized data in concise, descriptive terms.
- A concept usually refers to a collection of concise or summarized data such as grade wise students, products selling on festival etc.

What is Concept Description? Cont..

- Concept description generates descriptions for characterization and comparison of the data, it is also called class description.
- Characterization provides a concise and brief summarization of the data.
- While concept or class comparison (also known as discrimination) provides discriminations (inequity) comparing two or more collections of data.
- Example
  - Given the ABC Company database, for example, examining individual customer transactions.
  - Sales managers may prefer to view the data generalized to higher levels, such as summarized by customer groups according to geographic regions, frequency of purchases per group and customer income.

Market Basket Analysis

Section - 2

Market Basket Analysis

- Market Basket Analysis is a modelling technique to find frequent itemset.
- It is based on, if you buy a certain group of items, you are more (or less) likely to buy another group of items.
- For example, if you are in a store and you buy a car then you are more likely to buy insurance at the same time than somebody who don't buy insurance also.
- The set of items that a customer buys it referred as an **itemset**.
- Market basket analysis seeks to find relationships between purchases (Items).
  - **E.g.** IF {Car, Accessories} THEN {Insurance}

{Car, Accessories} → {Insurance}

The probability that a customer will buy car without an accessories is referred as the support for rule.	The conditional probability that a customer will purchase Insurance is referred to as the confidence.

Association Rule Mining

Section - 3

Association Rule Mining

- Given a set of transactions, we need rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.
- **Market-Basket transactions**

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

Example of Association Rules

{Chocolate} → {Pepsi},
{Milk, Bread} → {Eggs, Coke},
{Pepsi, Bread} → {Milk}

Association Rule Mining Cont..

- **Itemset**
  - A collection of one or more items
    - **E.g.** : {Milk, Bread, Chocolate}
  - k-itemset

    An itemset that contains **k** items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
    - **E.g.** σ({Milk, Bread, Chocolate}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
    - **E.g.** s({Milk, Bread, Chocolate}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a minimum support threshold

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

Association Rule Mining Cont..

- **Association Rule**
  - An implication expression of the form X → Y, where X and Y are item sets
    - **E.g.**: {Milk, Chocolate} → {Pepsi}
- **Rule Evaluation**
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

Example:

Find **support** & **confidence** for {Milk, Chocolate} ⇒ Pepsi

$$s = \frac{\sigma(Milk, Chocolate, Pepsi)}{|T|} = \frac{2}{5} = \mathbf{0.4}$$

$$c = \frac{\sigma(Milk, Chocolate, Pepsi)}{\sigma(Milk, Chocolate)} = \frac{2}{3} = \mathbf{0.67}$$

Association Rule Mining Cont..

Calculate **Support** & **Confidence**

Support (s) **: 0.4**

1. {Milk, Chocolate} ⟶ {Pepsi} c = **0.67**
2. {Milk, Pepsi} ⟶ {Chocolate} c = **1.0**
3. {Chocolate, Pepsi} ⟶ {Milk} c = **0.67**
4. {Pepsi} ⟶ {Milk, Chocolate} c = **0.67**
5. {Chocolate} ⟶ {Milk, Pepsi} c = **0.5**
6. {Milk} ⟶ {Chocolate, Pepsi} c = **0.5**

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

A common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks:

1. **Frequent Itemset Generation**
   - The objective is to find all the item-sets that satisfy the minimum support threshold.
   - These item sets are called frequent item sets.
2. **Rule Generation**
   - The objective is to extract all the high-confidence rules from the frequent item sets found in the previous step.
   - These rules are called strong rules.

Apriori Algorithm

Section - 4

Apriori Algorithm - Example

Minimum Support = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

ItemSet	Min. Sup
{1}	2
{2}	3
~~{3}~~	~~3~~
~~{4}~~	~~1~~
{5}	3

✗

→

L_1

ItemSet	Min. Sup
{1}	2
{2}	3
{3}	3
{5}	3

→

C_2

ItemSet
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D →

C_2

ItemSet	Min. Sup
~~{1 2}~~	~~1~~ ✗
~~{1 3}~~	~~2~~
~~{1 5}~~	~~1~~ ✗
{2 3}	2
{2 5}	3
{3 5}	2

←

L_2

ItemSet	Min. Sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Scan D ←

ItemSet	Min. Sup
~~{1 2 3}~~	~~1~~ ✗
~~{1 2 5}~~	~~1~~ ✗
~~{1 3 5}~~	~~1~~ ✗
~~{2 3 5}~~	~~2~~ ✓

Apriori Algorithm - Example Cont..

Minimum Support = 2

Rules Generation

Association Rule	Support	Confidence	Confidence (%)
2 ^ 3 → 5	2	2/2 = **1**	**100 %**
3 ^ 5 → 2	2	2/2 = **1**	**100 %**
2 ^ 5 → 3	2	2/3 = **0.66**	**66%**
2 = 3 ^ 5	2	2/3 = **0.66**	**66%**
3 = 2 ^ 5	2	2/3 = **0.66**	**66%**
5 = 2 ^ 3	2	2/3 = **0.66**	**66%**

Apriori Algorithm

- **Purpose:** The Apriori Algorithm is an influential algorithm for mining **frequent itemsets** for **Boolean association rules**.
- **Key Concepts:**
  - Frequent Itemsets:
    - The sets of item which has **minimum support** (denoted by L_i for ith-Itemset).
  - Apriori Property:
    - Any **subset of frequent itemset must be frequent**.
  - Join Operation:
    - To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} itself.

Apriori Algorithm Cont..

- **Find the frequent itemsets**
  - The sets of items that have minimum support and a subset of a frequent itemset **must** also be a frequent itemset (**Apriori Property**).
  - **E.g.** if **{AB}** is a frequent itemset, both {A} and {B} should be a frequent itemset.
  - Use the frequent item sets to generate association rules.
- **The Apriori Algorithm : Pseudo code**
  - **Join Step**: C_k is generated by joining $L_{k-1}$with itself
  - **Prune Step**: Any (k-1) itemset that is not frequent cannot be a subset of a frequent k-itemset

```
Ck: Candidate itemset of size k
Lk: Frequent itemset of size k
L1= {frequent items};
        for (k = 1; Lk != ∅; k++) do begin
                Ck+1 = candidates generated from Lk;
        for each transaction t in database do
                Increment the count of all candidates in Ck+1
                That are contained in t
                Lk+1 = candidates in Ck+1 with min_support
        end
        return ∪k Lk;
```

Apriori Algorithm Steps

- **Step 1:**
  - Start with itemsets containing just a single item (Individual items).
- **Step 2:**
  - Determine the support for itemsets.
  - Keep the itemsets that meet your minimum support threshold and remove itemsets that do not support minimum support.
- **Step 3:**
  - Using the itemsets you have kept from Step 1, generate all the possible itemset combinations.
- **Step 4:**
  - Repeat steps 1 & 2 until there are no more new itemsets.

Apriori Algorithm (Try Yourself!)

A database has 4 transactions. Let Min_sup = 50% and Min_conf = 75%

TID	Items
1000	Cheese, Milk, Cookies
2000	Butter, Milk, Bread
3000	Cheese, Butter, Milk, Bread
4000	Butter, Bread

Frequent Itemset	Sup
Butter,Milk,Bread	2

Sr.	Association Rule	Support	Confidence	Confidence (%)
Rule 1	**Butter^Milk → Bread**	**2**	**2/2 = 1**	**100%**
Rule 2	**Milk^Bread → Butter**	**2**	**2/2 = 1**	**100%**
Rule 3	Butter^Bread → Milk	2	2/3 = 0.66	66%
Rule 4	Butter→Milk^Bread	2	2/3 = 0.66	66%
Rule 5	Milk→Butter^Bread	2	2/3 = 0.66	66%
Rule 6	Bread→Butter^Milk	2	2/3 = 0.66	66%

Min_sup = 50%

How to convert support in integer?

$$\frac{\text{Given \% } \mathbf{X} \text{ Total Records}}{100}$$

So here, $\frac{50 \mathbf{X} 4}{100} = 2$

Classification

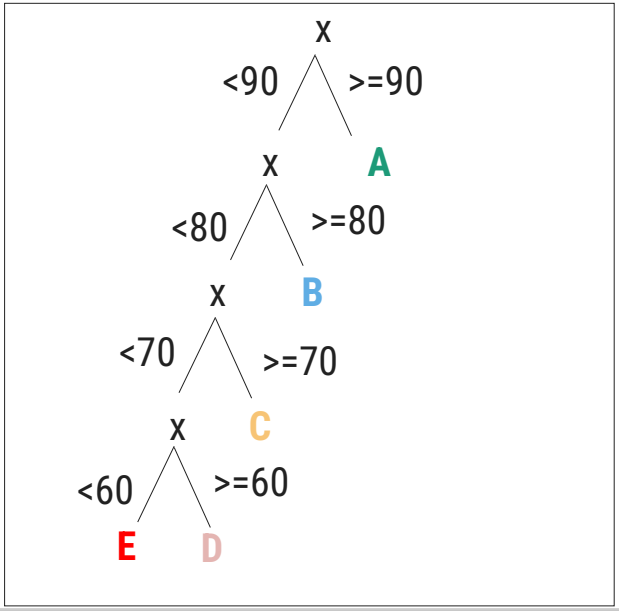
Section - 5

Classification

- Classification is a supervised learning method.
- It is a data mining function that assigns items in a collection to target categories or classes.
- The goal of classification is to accurately predict the target class for each case in the data.
- **For example**, a classification model could be used to identify loan applicants as low, medium, or high credit risks.
- In supervised learning, the learner(computer program) is provided with two sets of data, training data set and test data set.
- The idea is for the learner to "learn" from a set of labeled examples in the training set so that it can identify unlabeled examples in the test set with the highest possible accuracy.
- Suppose a Database D is given as $D = \{t_1, t_2, .. t_n\}$ and a set of desired classes are $C=\{C_1, ..., C_m\}$.
- The classification problem is to define the mapping m in such a way that which tuple of database D belongs to which class of C.
- Actually we divides D into equivalence classes.

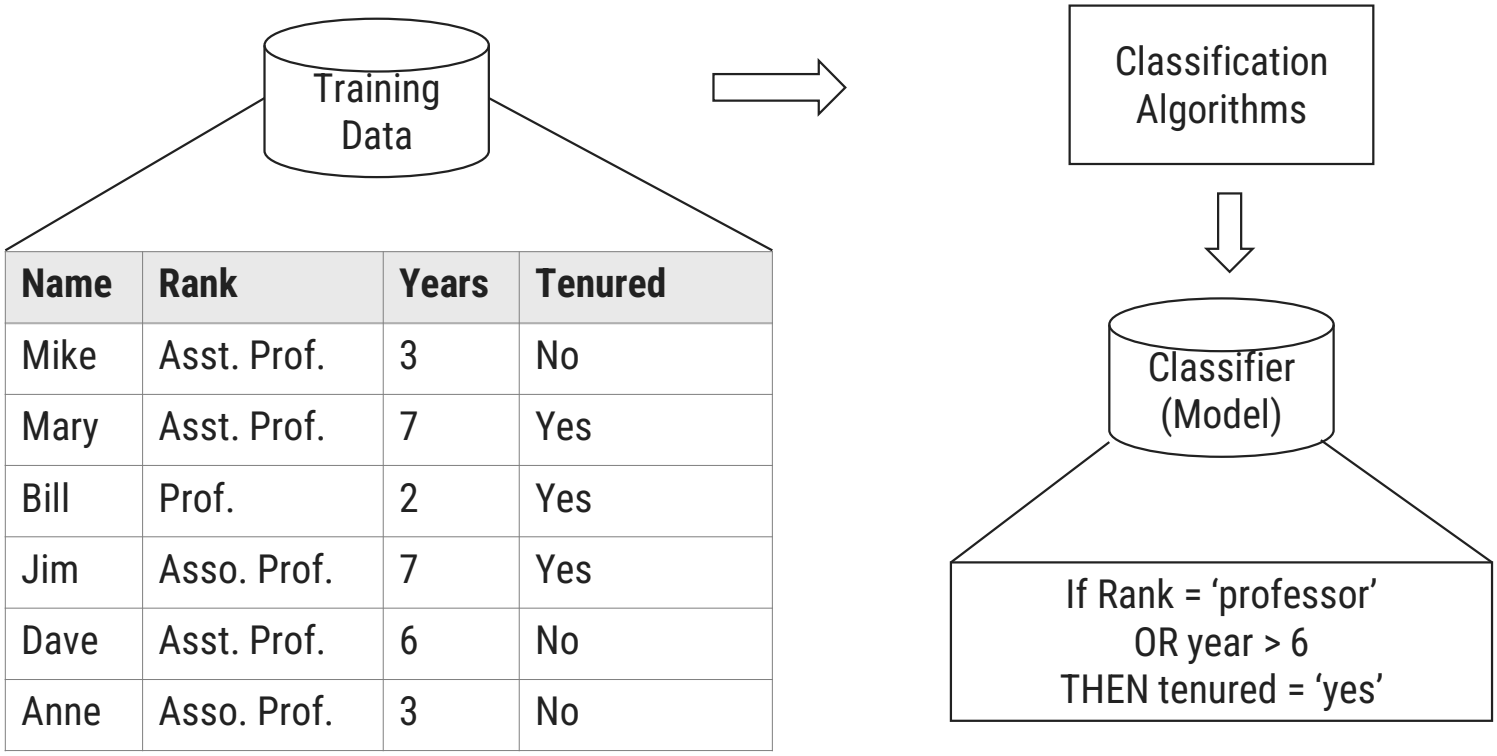
Classification Example

- Teachers classify students grades as **A,B,C,D or E**.
- Identify individuals with credit risks (**high, low, medium or unknown**).
- In cricket (**batsman, bowler, all-rounder**)
- Websites (**educational, sports, music**)
- How teachers give grades to students based on their obtained marks?
  - If **x >= 90** then **A** grade.
  - If **80 <= x < 90** then **B** grade.
  - If **70 <= x < 80** then **C** grade.
  - If **60 <= x < 70** then **D** grade.
  - If **x < 60** then **E** grade.



Classification : a two step process

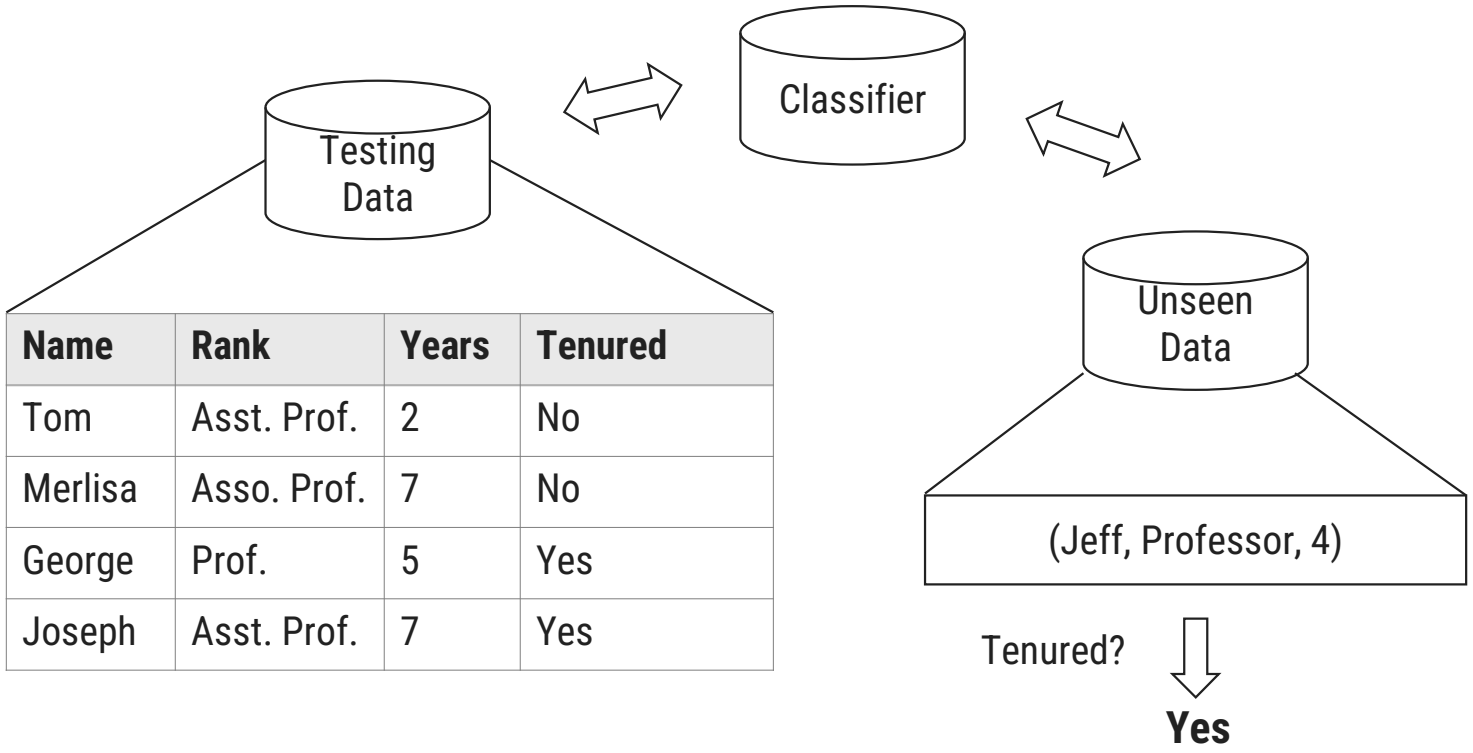
1. Model Construction



Name	Rank	Years	Tenured
Mike	Asst. Prof.	3	No
Mary	Asst. Prof.	7	Yes
Bill	Prof.	2	Yes
Jim	Asso. Prof.	7	Yes
Dave	Asst. Prof.	6	No
Anne	Asso. Prof.	3	No

Classification : a two step process Cont..

1. Model Usage



Name	Rank	Years	Tenured
Tom	Asst. Prof.	2	No
Merlisa	Asso. Prof.	7	No
George	Prof.	5	Yes
Joseph	Asst. Prof.	7	Yes

Classification : a two step process Cont..

- **Model Construction**
  - Describing a set of predetermined classes :
    - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute.
    - The set of tuples used for model construction is called as training set.
    - The model is represented as classification rules, decision trees, or mathematical formulae.

- **Model Usage**
  - For classifying future or unknown objects
    - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model.
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model.

Classification & Prediction Issues

Data Preparation

- **Data cleaning**
  - **Pre-process data** in order to **reduce noise and handle missing values**.
- **Relevance analysis** (Feature selection)
  - **Remove** the irrelevant or **redundant attributes**.
- **Data transformation**
  - Generalize the data to higher level concepts using **concept hierarchies** and/or normalize data which involves scaling the values.

Classification & Prediction Issues Cont..

Evaluating Classification Methods

- **Predict accuracy**
  - This refers the ability of the model **to correctly predict the class** label of new or previously unseen data.
- **Speed and scalability**
  - Time to **construct** model
  - Time to **use** the model
- **Robustness**
  - Handling noise and missing values
- **Interpretability**
  - Understanding and insight provided by model
- **Goodness of rules**
  - Decision **tree size**
  - Strongest rule or not

Bayesian Classification

Section - 6

What is Naive Bayes?

- The Naive Bayes classifier works on the principle of conditional probability, as given by the Bayes theorem.
- Consider the following example of tossing two coins.
- If we toss two coins and look at all the different possibilities, we have the sample space as:{HH, HT, TH, TT}
- While calculating the math on probability, we usually denote probability as P. Some of the probabilities in this event would be as follows:
- The probability of getting **two heads** = 1/4
- The probability of **at least one tail** = 3/4
- The probability of the **second coin being head given the first coin is tail** = 1/2
- The probability of **getting two heads given the first coin is a head** = 1/2
- The Bayes theorem gives us the conditional probability of event A, given that event B has occurred.
- In this case, the first coin toss will be B and the second coin toss A.
- This could be confusing because we've reversed the order of them and go from B to A instead of A to B.

Bayesian Classification

- Thomas Bayes, who proposed the Bayes Theorem so, it named Bayesian theorem.
- It is statistical method & supervised learning method for classification.
- It can solve problems involving both categorical and continuous valued attributes.
- Bayesian classification is used to find conditional probabilities.
- Consider a data set **D** with a tuple **X** in Bayes Theorem **X** works as an evidence. Let H be some hypothesis such as that the data tuple X belongs to a specified class C.
- Suppose in **D** our data tuple **X** is confined to income and age only (actually these are the attributes) and one of the tuple says that **X** is a customer with age 35 and income 40,000 Rs. and let **H** is the hypothesis that our customer will buy a computer then the following are probabilities we need to consider-

- **The Bayes Theorem**:
  - $P(H|X) = \frac{P(X|H)\,P(H)}{P(X)}$
- **P(H|X)** : The probability that customer X will buy a computer given that we know the customer's age and income.
- **P(X|H)** : The probability that the customer X is 35 years of age and earns 40,000 Rs., given that we know he/she will buy the computer.
- **P(H)** : The probability the customer X from a set of customers is 35 years old and earns 40,000 Rs.
- **P(X)** : The probability that the customer will buy the computer.
- Here P(H|X), P(X|H) are called as **posterior** probability and P(X) ,P(H) are called **prior** probability.

Naïve Bayes Classifier - Example

Age	Income	Student	Credit_Rating	Class : buys_computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31..40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31..40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31..40	Medium	No	Excellent	Yes
31..40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

Naïve Bayes Classifier - Example Cont..

P (**Yes**) = 9/14 | P (**No**) = 5/14

Age	
P (<=30 | **Yes**) = 2/9	P (<=30 | **No**) = 3/5
P (31..40 | **Yes**) = 4/9	P (31..40 | **No**) = 0/5
P (> 40 | **Yes**) = 3/9	P (> 40 | **No**) = 2/5

Income	
P (High | **Yes**) = 2/9	P (High | **No**) = 2/5
P (Medium | **Yes**) = 4/9	P (Medium | **No**) = 2/5
P (Low | **Yes**) = 3/9	P (Low | **No**) = 1/5

Student	
P (No | **Yes**) = 3/9	P (No | **No**) = 4/5
P (Yes | **Yes**) = 6/9	P (Yes | **No**) = 1/5

Credit_rating	
P (Fair | **Yes**) = 6/9	P (Fair | **No**) = 2/5
P (Excellent | **Yes**) = 3/9	P (Excellent | **No**) = 3/5

Age	Income	Student	Credit_Rating	buys_comp?
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31..40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31..40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31..40	Medium	No	Excellent	Yes
31..40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

Naïve Bayes Classifier – Example Cont..

Age	Income	Student	Credit_Rating	Class : buys_computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31..40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31..40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31..40	Medium	No	Excellent	Yes
31..40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

An unseen sample Y =
(<=30, Low, Yes, Excellent)

buys_computer
Yes OR No?

Naïve Bayes Classifier – Example Cont..

We take one unseen sample record

Y = (<=30, Low, Yes, Excellent)

P(Y|Yes).P(Yes) = P(<=30|Yes). P(Low|Yes). P(Yes|Yes).P(Excellent|Yes) . P(Yes)

= 2/9 * 3/9 * 6/9 * 3/9 * 9/14

= 0.010582

P(Y|No).P(No) = P(<=30|No). P(Low|No). P(Yes|No).P(Excellent|No) . P(No)

= 3/5 * 1/5 * 1/5 * 3/5 * 5/14

= 0.005142

- ✓ Choose the class with maximizes this probability, this means that new instance **Y in above example** will be classified as **Yes** (Buys_computer)

Y = (<=30, Low, Yes, Excellent) → Yes (buys_computer)

Advantages - Bayesian Classification

- It is simple and easy to implement.
- It doesn't require as much training data.
- It handles both continuous and discrete data.
- It is highly scalable with the number of predictors and data points.
- It is fast and can be used to make real-time predictions.

Naïve Bayes Classifier Example (Try yourself!)

Car No	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

An unseen sample Y
= <Red, Domestic, SUV>

Stolen?
Yes OR No?
0.024,0.072 (Unseen)

Actual Data
Y = <Red, Sports, Domestic>

Stolen?
Yes OR No?
0.192, 0.096 (Actual)

FP Growth Algorithm

Section - 7

FP-Growth Algorithm

- The FP-Growth Algorithm is proposed by Han.
- It is an efficient and scalable method for mining the complete set of frequent patterns.
- Using prefix-tree structure for storing information about frequent patterns named frequent-pattern tree (**FP-tree**).
- Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent item sets.

FP-Growth Algorithm - Example

Minimum Support = 2

FP-Tree Generation

TID	Items
1	A B C E F O
2	A C G
3	E I
4	A C D E G
5	A C E G L
6	E J
7	A B C E F P
8	A C D
9	A C E G M
10	A C E G N

Step:1
Freq. 1-Itemsets.
Min_Sup ≥ 2

Arranged Order
A : 8
C : 8
E : 8
G : 5
B : 2
D : 2
F : 2

Remaining all O,I,J,L,P,M & N is with min_sup = 1

Step:2
Transactions with items sorted based on frequencies, and ignoring the infrequent items.

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G

Building the FP-Tree

- ✓ Scan data to determine the support count of each item.
- ✓ Infrequent items are discarded, while the frequent items are sorted in decreasing support counts.
- ✓ Make a second pass over the data to construct the FP-tree.
- ✓ As the transactions are read, before being processed, their items are sorted according to the above order.

FP-Tree after reading 1st transaction

A C E B F

A C G

E

A C E G D

A C E G

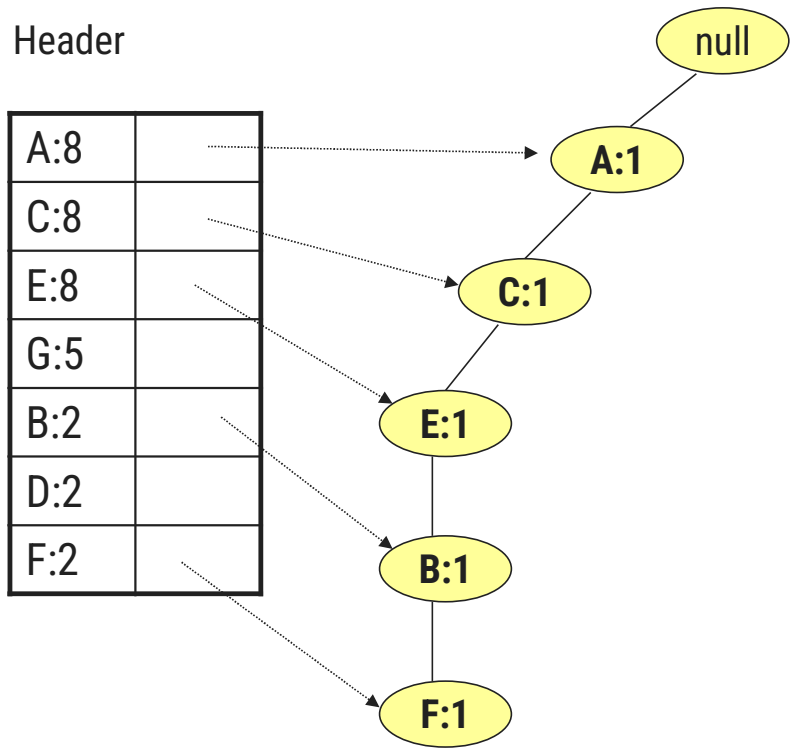
E

A C E B F

A C D

A C E G

A C E G



FP-Tree after reading 2nd transaction

A C E B F

A C G

E

A C E G D

A C E G

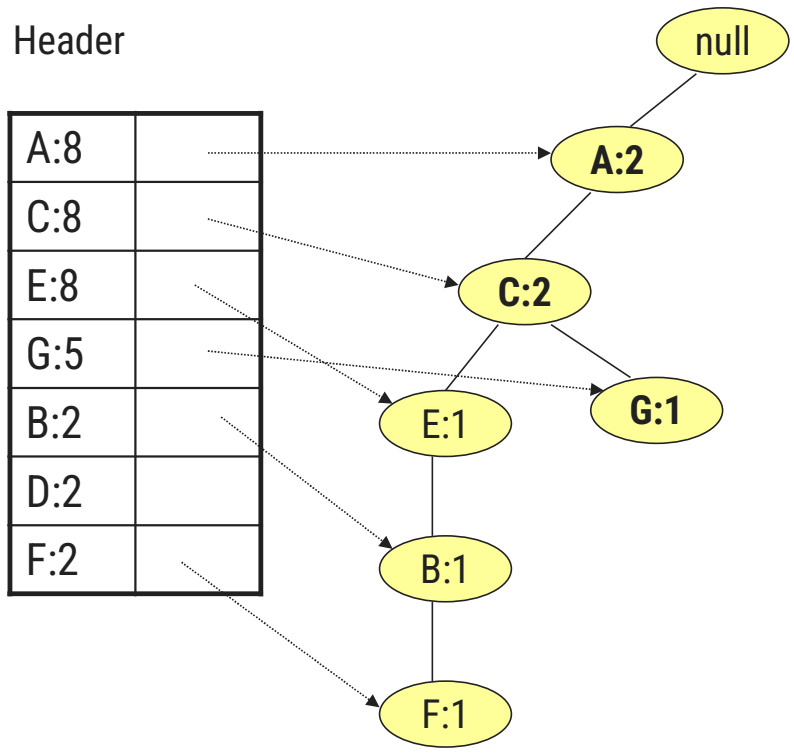
E

A C E B F

A C D

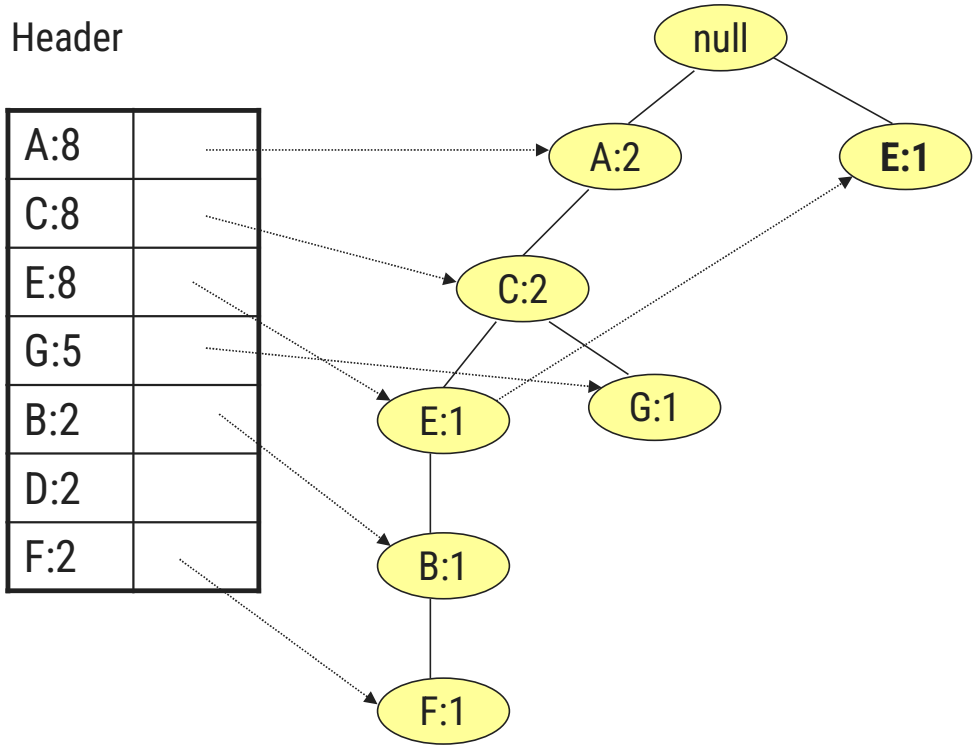
A C E G

A C E G



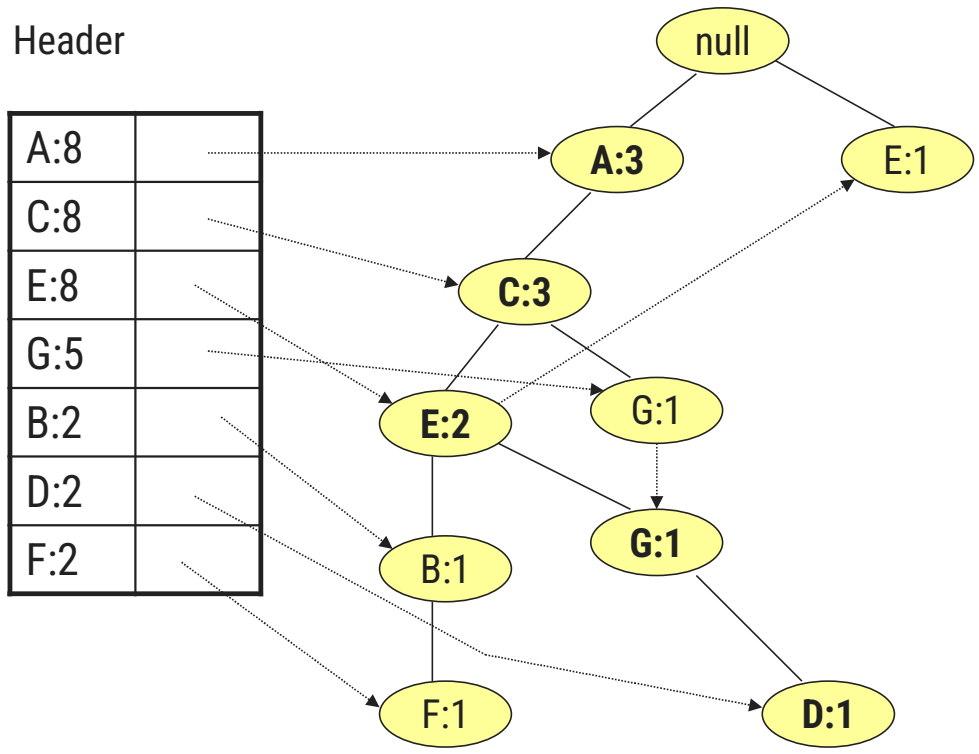
FP-Tree after reading 3rd transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



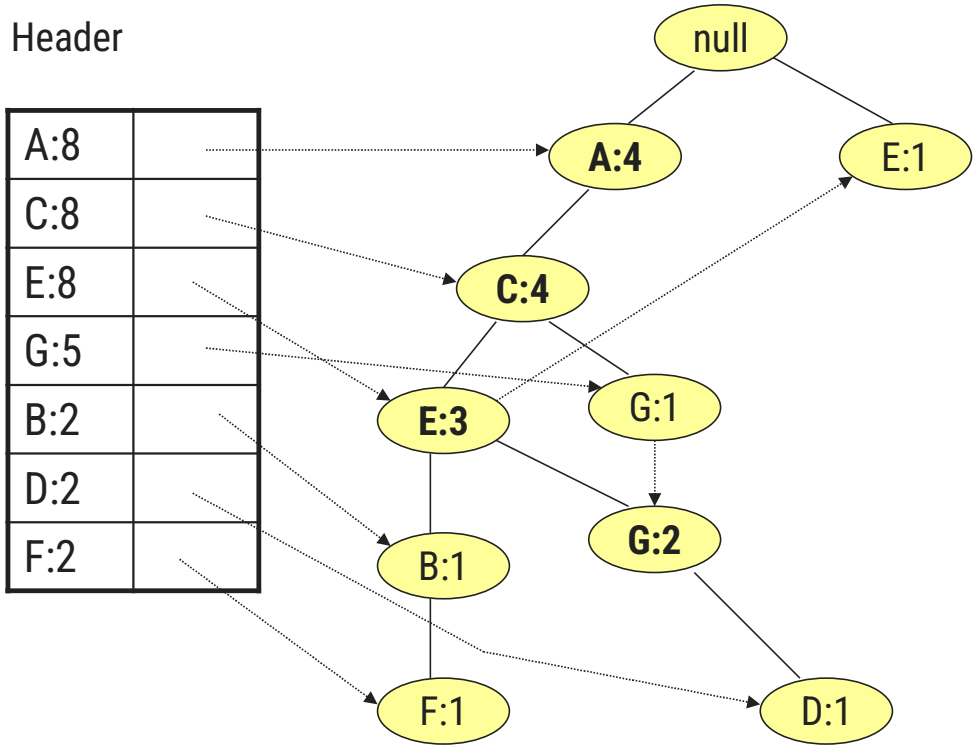
FP-Tree after reading 4th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



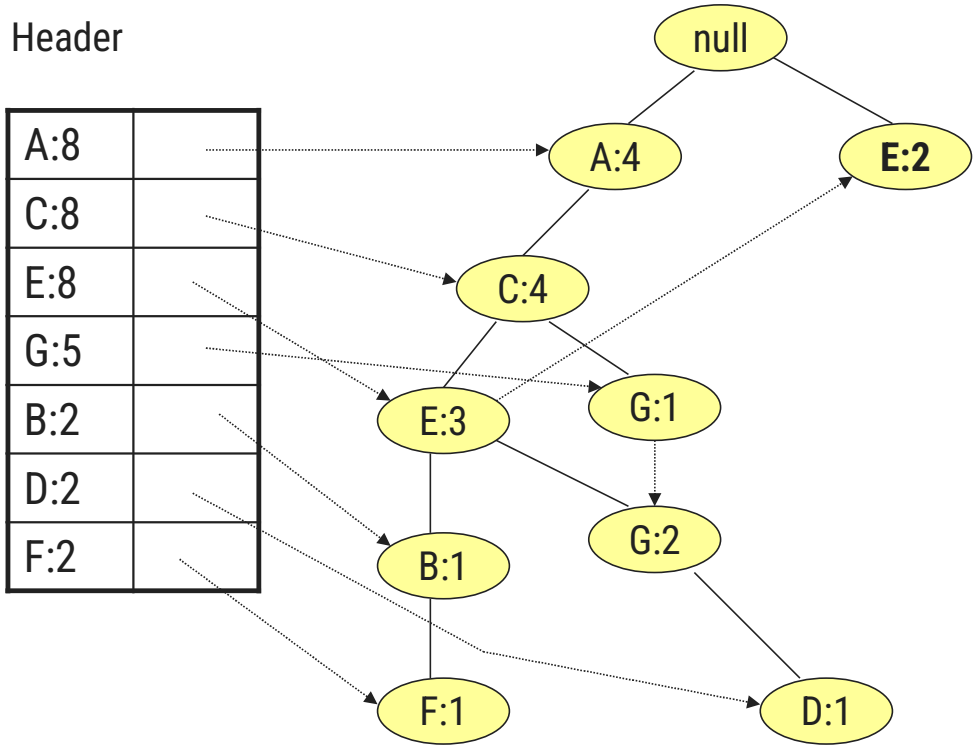
FP-Tree after reading 5th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



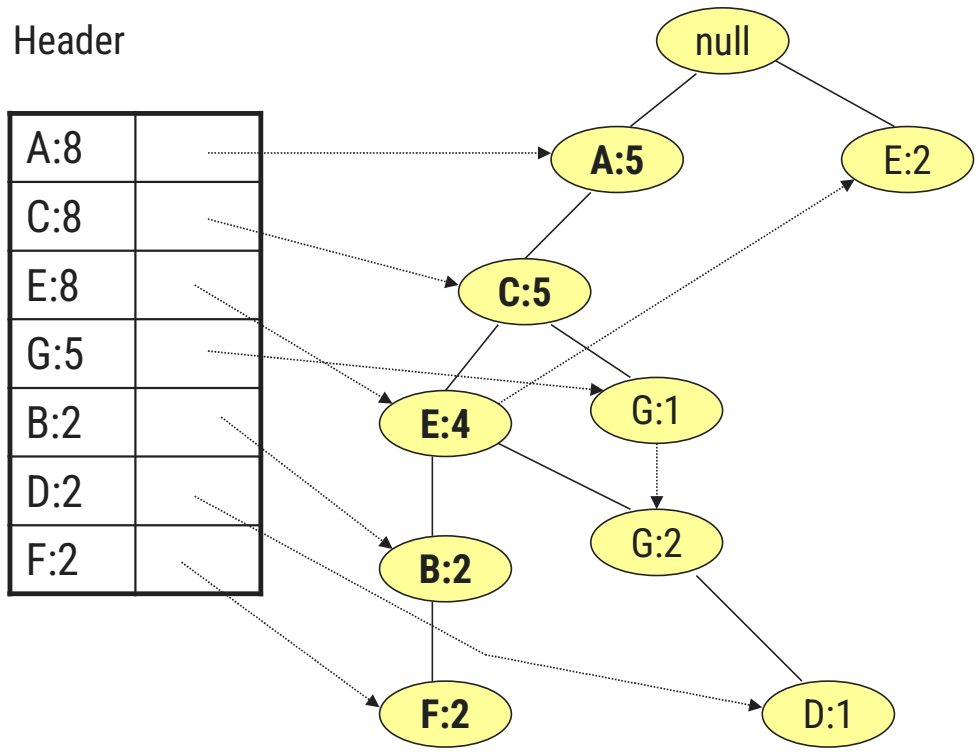
FP-Tree after reading 6th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



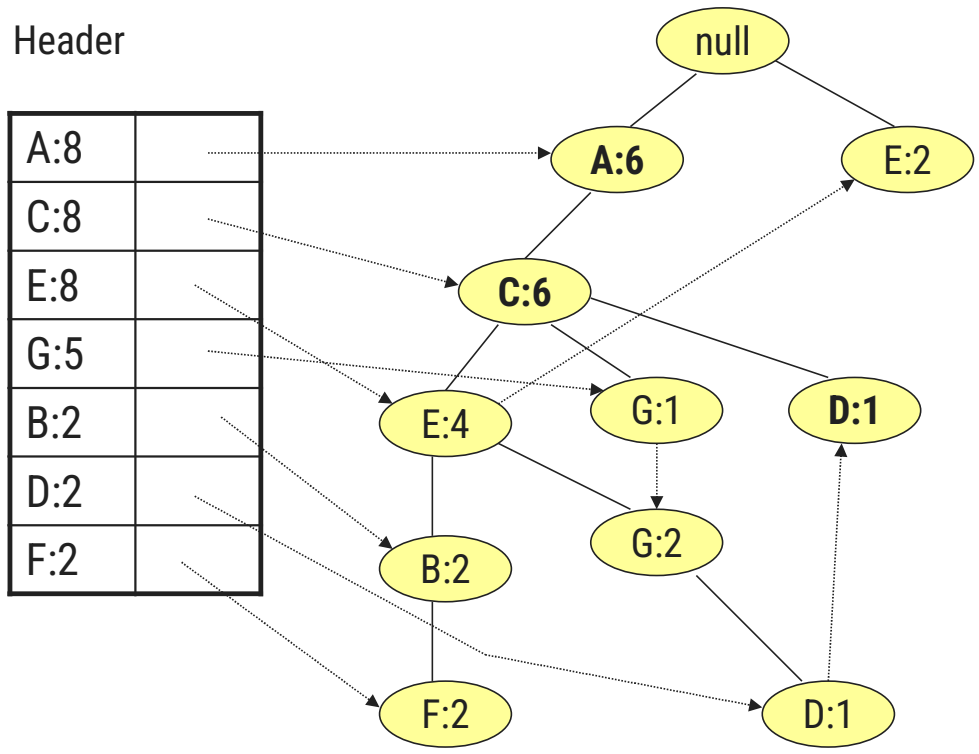
FP-Tree after reading 7th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



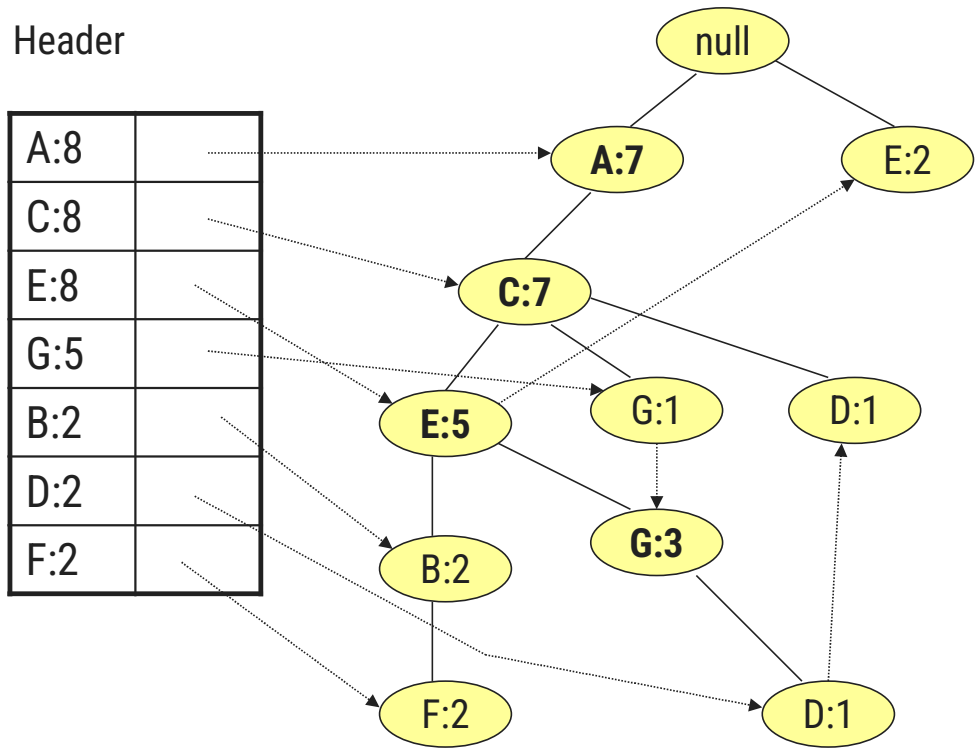
FP-Tree after reading 8th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



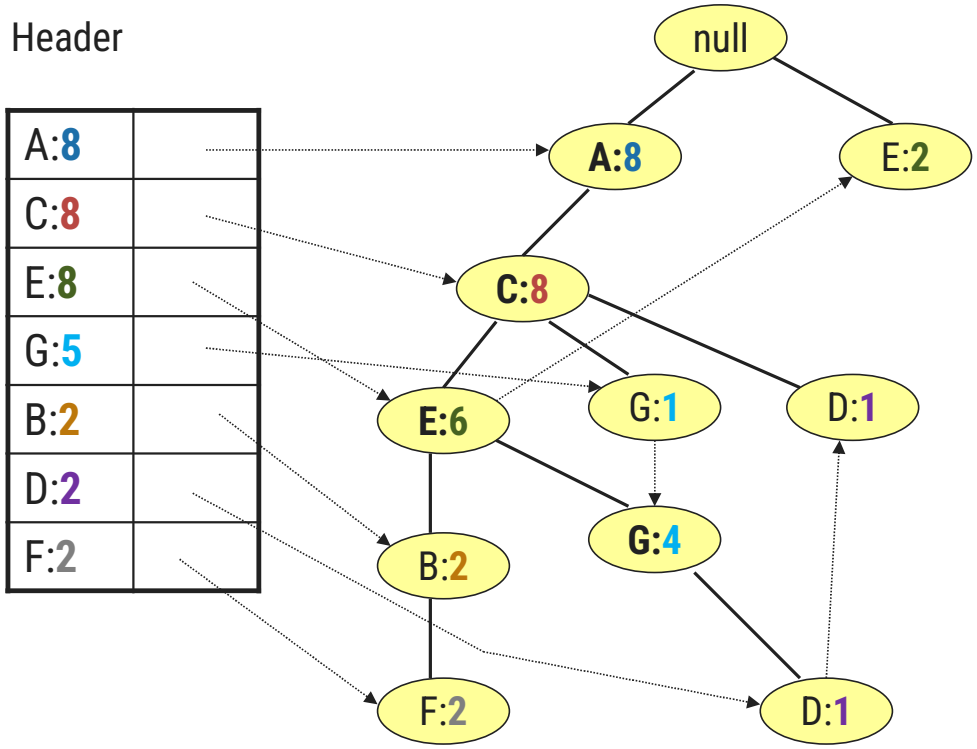
FP-Tree after reading 9th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



FP-Tree after reading 10th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



Examples

Apriori Algorithm – Example (Try Yourself!)

TID	Items
100	1,3,4,6
200	2,3,5,7
300	1,2,3,5,8
400	2,5,9,10
500	1,4

Itemset	Support
2,5	3

Min-support = 60% and Confidence = 80%

Rule	Support	Confidence	Conf. (%)
2 → 5	3	3/3 = 1	100%
5 → 2	3	3/3 = 1	100%

TID	Items
1000	A,B,C
2000	A,C
3000	A,D
5000	B,E,F

Itemset	Support
A,C	2

Min-support = 50%

Rule	Support	Confidence	Conf. (%)
A → C	2	2/3 = 0.66	66%
C → A	2	2/2 = 1	100%

FP-Growth Algorithm – Example (Try Yourself!)

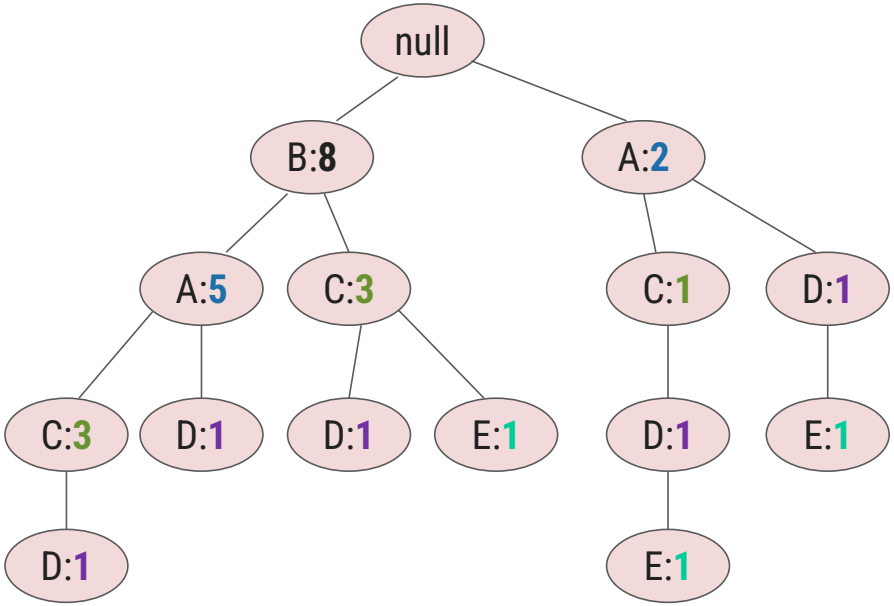
Minimum Support = 3

TID	Items
1	A B
2	B C D
3	A C D E
4	A D E
5	A B C
6	A B C D
7	B C
8	A B C
9	A B D
10	B C E

Header

Item	Support
B	**8**
A	**7**
C	**7**
D	**5**
E	**3**

FP-Tree



Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit – 3
Concept Description, Mining Frequent Patterns, Associations & Correlations

Thank You

Any Questions ?

Prof. Naimish R. Vadodariya
Computer Engineering
Darshan Institute of Engineering & Technology, Rajkot
naimish.Vadodariya@darshan.ac.in
8866215253

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-4
Classification & Prediction

Prof. Naimish R. Vadodariya
Computer Engineering Department
Darshan Institute of Engineering & Technology, Rajkot
naimish.vadodariya@darshan.ac.in
8866215253

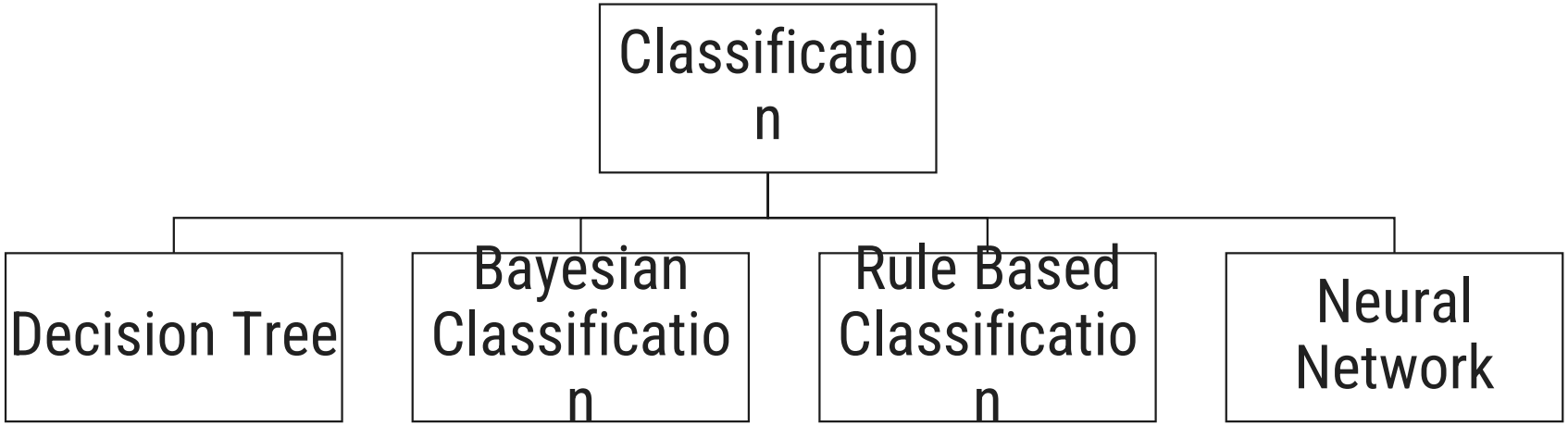
Topics to be covered

- Classification Methods
  - Decision Tree
  - Bayesian Classification
  - Rule Based Classification
  - Neural Network

Classification Methods

Section - 1

Classification Methods

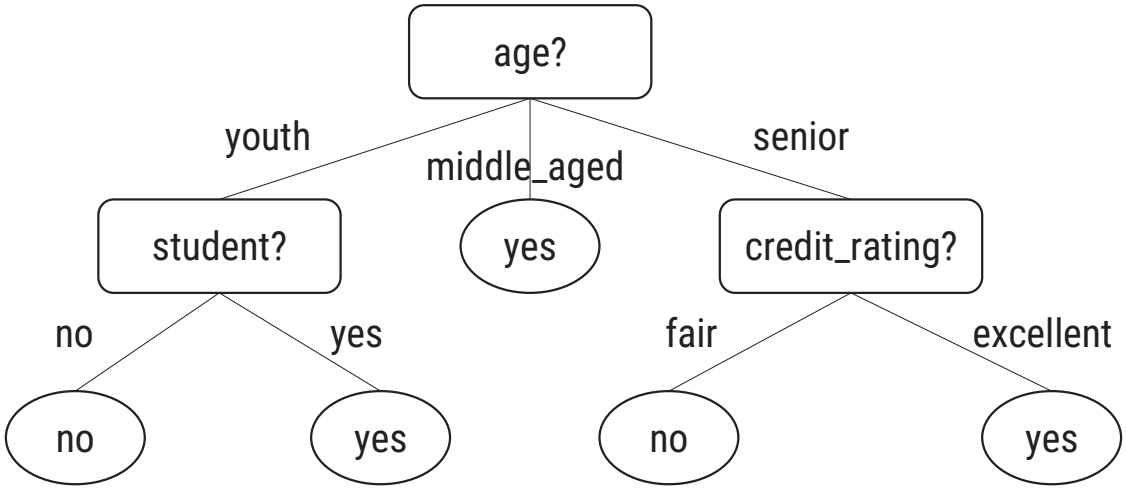


Decision Tree

- Decision tree induction is the learning of decision trees from class-labeled training tuples.
- A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes a test on an attribute.
- Each branch represents an outcome of the test.
- Each leaf node (or terminal node) holds a class label.
- The topmost node in a tree is the root node.

Decision Tree

- Decision Tree represents the concept *buys_computer*, i.e. it predicts whether a customer at AllElectronics is likely to purchase a computer.



History of Decision Tree

- During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser).
- This work expanded on earlier work on concept learning systems, described by E. B. Hunt, J. Marin, and P. T. Stone. Quinlan later presented C4.5 (a successor of ID3).
- In 1984, a group of statisticians (L. Breiman, J. Friedman, R. Olshen, and C. Stone) published the book Classification and Regression Trees (CART), which described the generation of binary decision trees.
- ID3, C4.5, and CART adopt a greedy (i.e., non-backtracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner.

Attribute Selection Measures

- An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D*, of class-labeled training tuples into individual classes.
- Also known as splitting rules as they determine how the tuples at a given node are to be split.
- The tree node created for partition *D* is labeled with the splitting criterion, branches are grown for each outcome of the criterion, and the tuples are partitioned accordingly.
- Three popular attribute selection measures
  1. Information gain
  2. Gain ratio
  3. Gini index

1. Information Gain

- ID3 uses information gain as its attribute selection measure.
- Let node N represent or hold the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N.
- This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions.
- The expected information needed to classify a tuple in D is given by

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) = -\sum_{i=1}^{m} |C_{i,D}|/|D| \log_2(|C_{i,D}|/|D|)$$

where p_i - nonzero probability that an arbitrary tuple in D belongs to class C_i = $|C_{i,D}|/|D|$

- *Info(D)* is just the average amount of information needed to identify the class label of a tuple in D.
- *Info(D)* is also known as the Entropy of D.

1. Information Gain Cont..

- How much more information would we still need (after the partitioning) to arrive at an exact classification? This amount is measured by

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

where $\frac{|D_j|}{|D|}$ - weight of the j^{th} partition

- $Info_A(D)$ - the expected information required to classify a tuple from *D* based on the partitioning by *A*.
- The smaller the expected information (still) required, the greater the purity of the partitions.

1. Information Gain Cont..

- Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A).

$$Gain(A) = Info(D) - InfoA(D)$$

- The attribute A with the highest information gain is chosen as the splitting attribute at node N.

Information Gain - Example

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Information Gain - Example

- The class label attribute, *buys_computer*, has two distinct values namely, {yes, no}, therefore, there are two distinct classes i.e., m = 2.
- Let class C_1 correspond to *yes* and class C_2 correspond to *no*.
- C_1 has 9 tuples & C_2 has 5 tuples.
- $Info(D)$ is computed as

$$Info(D) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940\ bits$$

Information Gain - Example

- Computing for *age* attribute
- For the *age* category "*youth*" – *2 yes* tuples & *3 no* tuples
- For the category "*middle_aged*" – *4 yes* tuples & *0 no* tuples
- For the category "*senior*" - *3 yes* tuples & *2 no* tuples

- $$Info_{age}(D)$$
$$= \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$
$$= 0.694\ bits$$

Information Gain - Example

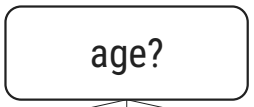
- Gain in information for *age* attribute is given as

$$Gain(age) = Info(D) - Infoage(D) = 0.940 - 0.694 = 0.246\ bits$$

- Similarly,
  - Gain(income) = 0.029 bits
  - Gain(student) = 0.151 bits
  - Gain(credit_rating) = 0.048 bits
- *age* attribute has highest Information Gain among all attributes.
- Therefore **node N is labelled with age** and **branches grow for each of the attributes value**.

Information Gain - Example

Splitting Attribute *age* at Root node



youth

income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

middle_aged

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

senior

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

2. Gain Ratio

- The information gain measure is biased toward tests with many outcomes.
- For example, consider an attribute that acts as a unique identifier such as *product_ID*.
- A split on product_ID would result in a large number of partitions each one containing just one tuple.
- $Info_{product\ ID}(D) = 0$ for *product_ID* attribute which results in maximum information gain. Clearly, such a partitioning is useless for classification.

2. Gain Ratio Cont..

- C4.5 uses an extension to information gain known as gain ratio.
- It applies a kind of normalization to information gain using a "split information" value defined with $Info(D)$ as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- This value represents the potential information generated by splitting the training data set, D, into v partitions, corresponding to the v outcomes of a test on attribute A.
- Gain Ratio is defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_{A(D)}}$$

- The attribute with the maximum gain ratio is selected as the splitting attribute.

2. Gain Ratio - Example

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Gain Ratio for the attribute *income* - Example

- To compute the gain ratio of *income*

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

- $Gain(income) = 0.029$

$$GainRatio(income) = \frac{0.029}{1.557} = 0.019$$

- Similarly, *GainRatio(age), GainRatio(student), GainRatio(credit_rating)* is to be computed.

3. Gini Index

- CART uses Gini Index.
- The Gini index measures the impurity of D, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

where p_i is the probability that a tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$.

- The Gini index considers a binary split for each attribute.

3. Gini Index Cont..

- Consider the case where A is a discrete-valued attribute having v distinct values, $\{a_1,a_2,\ldots,a_v\}$.
- Examine all the possible subsets that can be formed using known values of A.
- Each subset S_A, can be considered as a binary test for attribute A of the form "$A \in S_A$"
- For example, if *income* has three possible values, namely {low, medium, high}, then the possible subsets are {low, medium, high}, {low, medium}, {low, high}, {medium, high}, {low}, {medium}, {high}, and {}.
- Excluding the power set and the empty set, there are $2^V - 2$ possible ways to form two partitions of the data, D, based on a binary split on A.

3. Gini Index Cont..

- Compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on *A* partitions *D* into D_1 and D_2, the Gini index of *D* given that partitioning is

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

- For a discrete-valued attribute, the subset that gives the minimum Gini index for that attribute is selected as its splitting subset.

3. Gini Index - Example

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

3. Gini Index – Example Cont..

- Considering the data of AllElectronics,
- *buys_computer = yes* - 9 tuples
- *buys_computer = no* - 5 tuples
- Gini index to compute the impurity of *D* is

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

3. Gini Index – Example Cont..

- Consider each of the possible splitting subsets for income attribute.
- Consider the subset {low, medium}.
  - 10 tuples in partition D_1 satisfying the condition "$income \in \{low, medium\}$"
  - 4 tuples of D would be assigned to partition D_2.
- The Gini index value computed based on

$$Gini_{income \in \{low,medium\}}(D) = \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$

$$= 0.443$$

$$= Gini_{income \in \{high\}}(D)$$

3. Gini Index – Example Cont..

- $Gini_{income \in \{low,high\}}(D) = 0.458 = Gini_{income \in \{medium\}}(D)$
- $Gini_{income \in \{medium,high\}}(D) = Gini_{income \in \{low\}}(D) = 0.450$
- Best binary split for attribute income is on {low, medium} (or {high}) because it minimizes the Gini index.
- $Gini_{age \in \{youth,senior\}}(D) = Gini_{age \in \{middle\_aged\}}(D) = 0.357$
- $Gini_{student \in \{yes,no\}}(D) = 0.367$
- $Gini_{credit\_rating \in \{fair,excellent\}}(D) = 0.429$
- From above all Gini index of age is minimum which results in binary split $age \in \{youth, senior\}$ and $age \in \{middle\_aged\}$

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-4
Classification & Prediction

Thank You

Any Questions ?

Prof. Naimish R. Vadodariya
Computer Engineering
Darshan Institute of Engineering & Technology, Rajkot
naimish.Vadodariya@darshan.ac.in
8866215253

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-5
Clustering

Prof. Naimish R. Vadodariya
Computer Engineering Department
Darshan Institute of Engineering & Technology, Rajkot
naimish.vadodariya@darshan.ac.in
8866215253

Topics to be covered

- Clustering
  - Problem Definition, Clustering Overview, Evaluation of Clustering Algorithms
- Partitioning Clustering -K-Means Algorithm
- PAM Algorithm
- Hierarchical Clustering – Agglomerative Methods and divisive methods
- Basic Agglomerative Algorithm

Clustering : Introduction

Section - 1

What is Clustering?

- Clustering is one of the most important research areas in the field of data mining.
- **Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters).
- It is an unsupervised learning technique.
- Data clustering is the subject of active research in several fields such as statistics, pattern recognition and machine learning.
- From a practical perspective clustering plays an outstanding role in data mining applications in many domains.
- **The main advantage of clustering is that interesting patterns and structures can be found directly from very large data sets with little or none of the background knowledge**.
- Clustering algorithms an be applied in many areas, like marketing, biology, libraries, insurance, city-planning, earthquake studies and www document classification.

Applications of Clustering

- Real life examples where we use clustering:
  - **Marketing**
    - Finding group of customers with similar behavior given a large data-base of customers.
    - Data containing their properties and past buying records (Conceptual Clustering).
  - **Biology**
    - Classification of Plants and Animals Based on the properties under observation (Conceptual Clustering).
  - **Insurance**
    - Identifying groups of car insurance policy holders with a high average claim cost (Conceptual Clustering).
  - **City-Planning**
    - Groups of houses according to their house type, value and geographical location it can be both (Conceptual Clustering and Distance Based Clustering)
  - **Libraries**
    - It is used in clustering different books on the basis of topics and information.
  - **Earthquake studies**
    - By learning the earthquake-affected areas we can determine the dangerous zones.

Partitioning Clustering
K-Means Algorithm

Section - 2

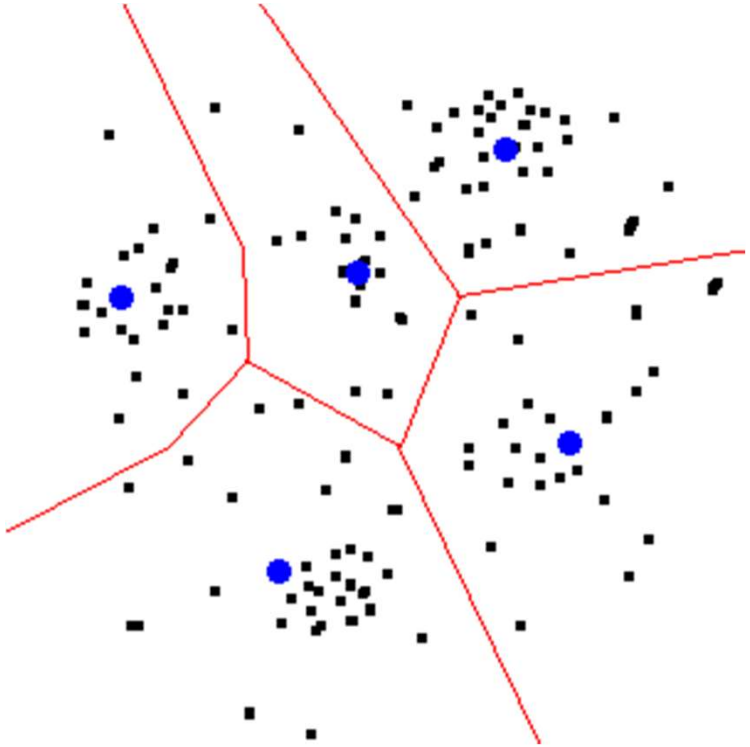
What is Partitioning?

- Clustering is a division of data into groups of similar objects.
- Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups.
- It represents many data objects by few clusters and hence, it models data by its clusters.
- A **cluster** is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.

K-MEANS Algorithm

- K-Means is one of the simplest unsupervised learning algorithm that solve the well known clustering problem.
- The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (k-clusters).
- The main idea is to define k centroids, one for each cluster.
- A centroid is “the center of mass of a geometric object of uniform density”, though here, we'll consider mean vectors as centroids.
- It is a method of classifying/grouping items into k groups (where k is the number of pre-chosen groups).
- The grouping is done by minimizing the sum of squared distances between items or objects and the corresponding centroid.

K-MEANS Algorithm Cont..



- A clustered scatter plot.
- The black dots are data points.
- The red lines illustrate the partitions created by the k-means algorithm.
- The blue dots represent the centroids which define the partitions.

K-MEANS Algorithm Cont..

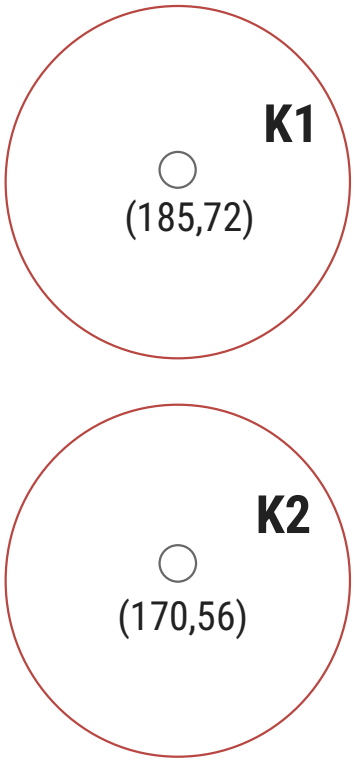
- The initial partitioning can be done in a variety of ways.
- **Dynamically Chosen**
  - This method is good when the amount of data is expected to grow.
  - The initial cluster means can simply be the first few items of data from the set.
  - For instance, if the data will be grouped into 3 clusters, then the initial cluster means will be the first 3 items of data.
- **Randomly Chosen**
  - Almost self-explanatory, the initial cluster means are randomly chosen values within the same range as the highest and lowest of the data values.
- **Choosing from Upper and Lower Bounds**
  - Depending on the types of data in the set, the highest and lowest of the data range are chosen as the initial cluster means.

K-Means Algorithm - Example

Section - 3

K-Means Algorithm - Example

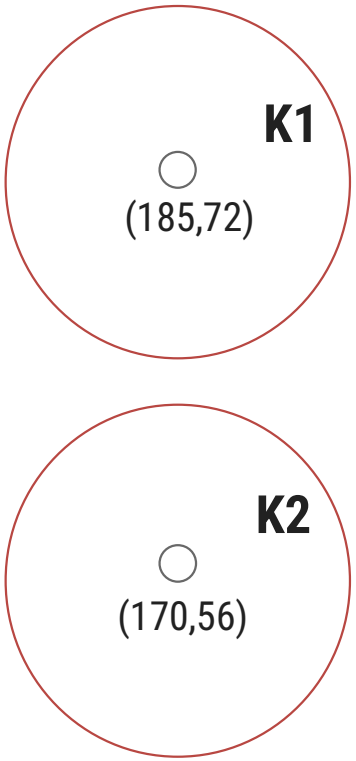
Sr.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



- First we take **K=2** So, two clusters or groups.
- We choose first (185,72) & second (170,56) row as centroid of each cluster or group.
- Now, we have to find Euclidean Distance,
  - $ED = \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$
- Where
  - X_o & Y_o = Observed Value
  - X_c & Y_c = Centroid Value

K-Means Algorithm – Example Cont..

Sr.	Height	Weight
1	185	72
2	170	56
3	**168**	**60**
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



➥ ED From **K1** ***to*** $(\mathbf{168, 60})$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(168 - 185)^2 + (60 - 72)^2}$

$= \sqrt{(-17)^2 + (-12)^2}$

$= \sqrt{289 + 144}$

$= \sqrt{433}$

$= 20.80$

➥ ED From **K2** ***to*** $(\mathbf{168, 60})$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(168 - 170)^2 + (60 - 56)^2}$

$= \sqrt{(-2)^2 + (-4)^2}$

$= \sqrt{4 + 16}$

$= \sqrt{20}$

$= 4.48$

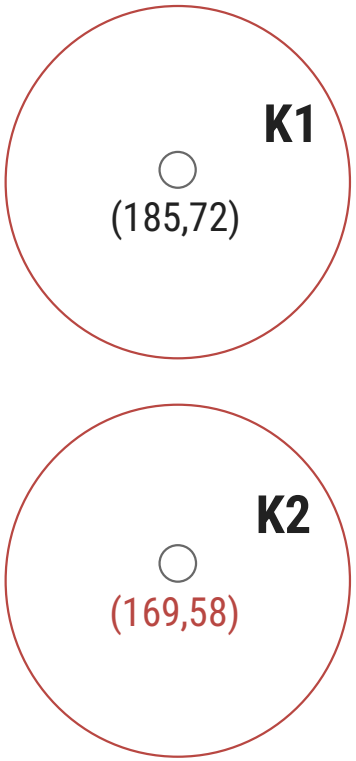
Now, data (168,60) nearer to K2, so it belongs to K2.

K1 = {1}

K2 = {2,3}

K-Means Algorithm – Example Cont..

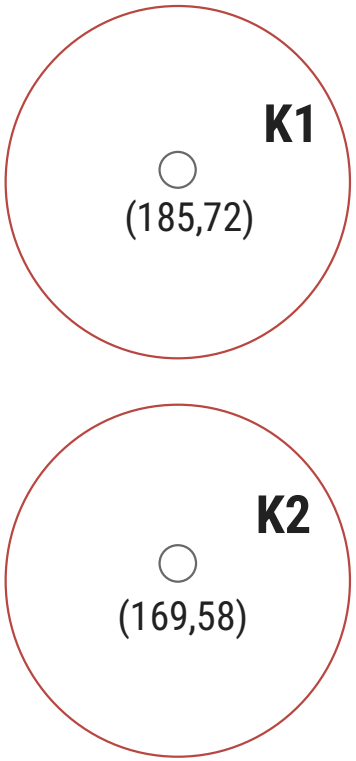
Sr.	Height	Weight
1	185	72
2	**170**	**56**
3	**168**	**60**
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



Now, New Centroid Calculation
For K2 = {2,3}
So, **K2** = {(170,56),(168,60)}
= 170+168/2 & 56+60/2
We get new centroid **C** = (169,58)

K-Means Algorithm – Example Cont..

Sr.	Height	Weight
1	185	72
2	170	56
3	168	60
4	**179**	**68**
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



➥ ED From **K1** ***to*** $(179, 68)$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(179 - 185)^2 + (68 - 72)^2}$

$= \sqrt{(-6)^2 + (-4)^2}$

$= \sqrt{36 + 16}$

$= \sqrt{52}$

$= 7.21$

➥ ED From **K2** ***to*** $(179, 68)$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(179 - 169)^2 + (68 - 58)^2}$

$= \sqrt{(10)^2 + (10)^2}$

$= \sqrt{100 + 100}$

$= \sqrt{200}$

$= 14.14$

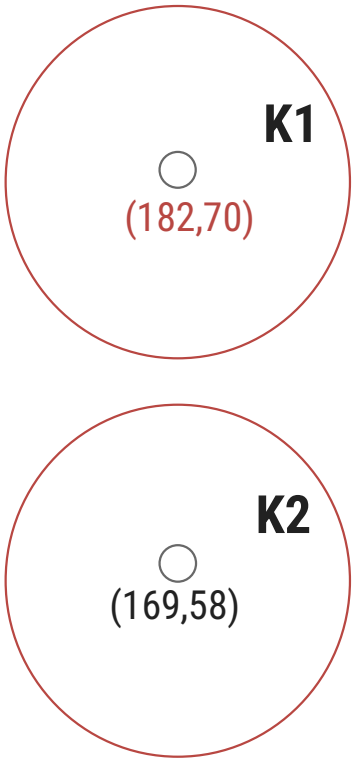
Now, data (179,68) nearer to K1, so it belongs to K1.

K1 = {1,4}

K2 = {2,3}

K-Means Algorithm – Example Cont..

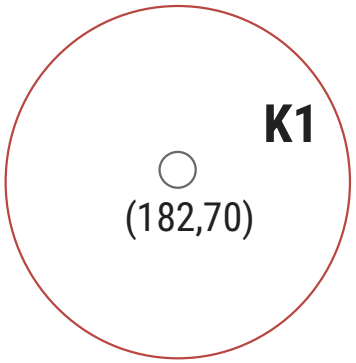
Sr.	Height	Weight
1	**185**	**72**
2	170	56
3	168	60
4	**179**	**68**
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76

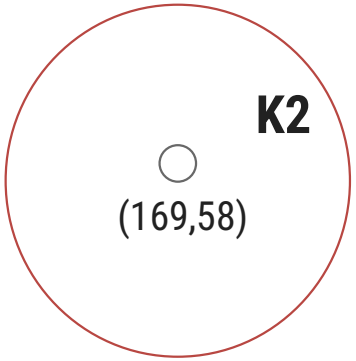


Now, New Centroid Calculation
For K1 = {1,4}
So, **K2** = {(185,72),(179,68)}
= 185+179/2 & 72+68/2
We get new centroid **C** = (182,70)

K-Means Algorithm – Example Cont..

Sr.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	**182**	**72**
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76





➥ ED From **K1** ***to*** $(182, 72)$

$$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$$
$$= \sqrt{(182 - 182)^2 + (72 - 70)^2}$$
$$= \sqrt{(0)^2 + (2)^2}$$
$$= \sqrt{0 + 4}$$
$$= \sqrt{4}$$
$$= 2$$

➥ ED From **K2** ***to*** $(182, 72)$

$$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$$
$$= \sqrt{(182 - 169)^2 + (72 - 58)^2}$$
$$= \sqrt{(-13)^2 + (-14)^2}$$
$$= \sqrt{169 + 196}$$
$$= \sqrt{365}$$
$$= 19.10$$

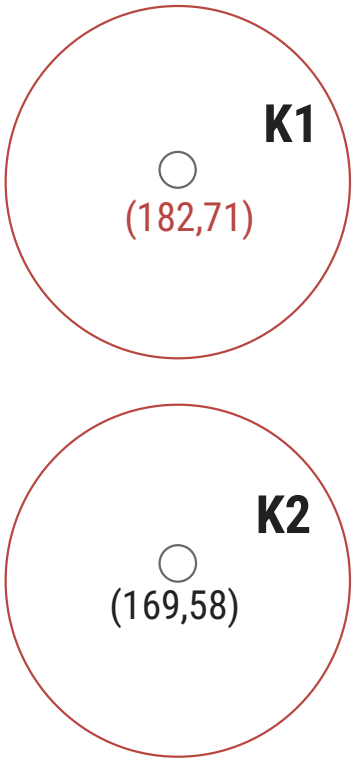
Now, data (182,72) nearer to K1, so it belongs to K1.

K1 = {1,4,5}

K2 = {2,3}

K-Means Algorithm – Example Cont..

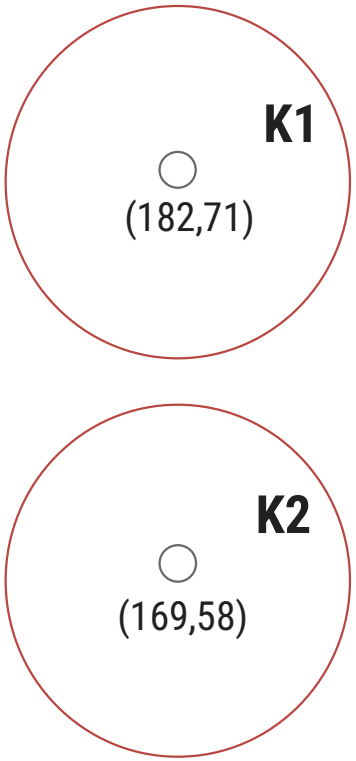
Sr.	Height	Weight
1	**185**	**72**
2	170	56
3	168	60
4	**179**	**68**
5	**182**	**72**
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



Now, New Centroid Calculation
For K1 = {1,4,5}
So, **K2** = {(185,72),(179,68),(182,72)}
= 185+179+182/3 & 72+68+72/3
We get new centroid **C** = (182,70.666) ~ (182,71)

K-Means Algorithm – Example Cont..

Sr.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	**188**	**77**
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



➥ ED From **K1** ***to*** $(\mathbf{188, 77})$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(188 - 182)^2 + (77 - 71)^2}$

$= \sqrt{(6)^2 + (6)^2}$

$= \sqrt{36 + 36}$

$= \sqrt{72}$

$= 8.48$

➥ ED From **K2** ***to*** $(\mathbf{188, 77})$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(188 - 169)^2 + (77 - 58)^2}$

$= \sqrt{(19)^2 + (19)^2}$

$= \sqrt{361 + 361}$

$= \sqrt{722}$

$= 26.87$

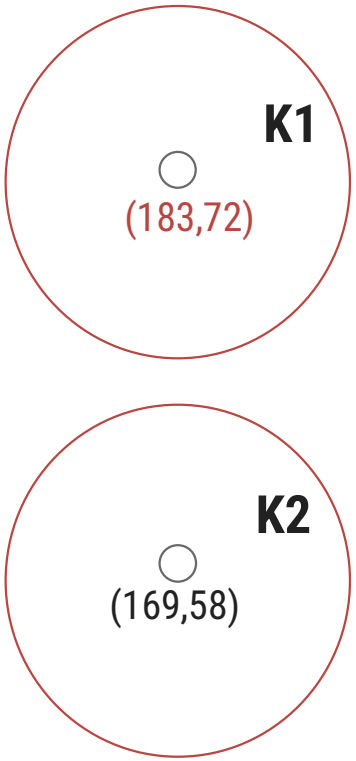
Now, data (188,77) nearer to K1, so it belongs to K1.

K1 = {1,4,5,6}

K2 = {2,3}

K-Means Algorithm – Example Cont..

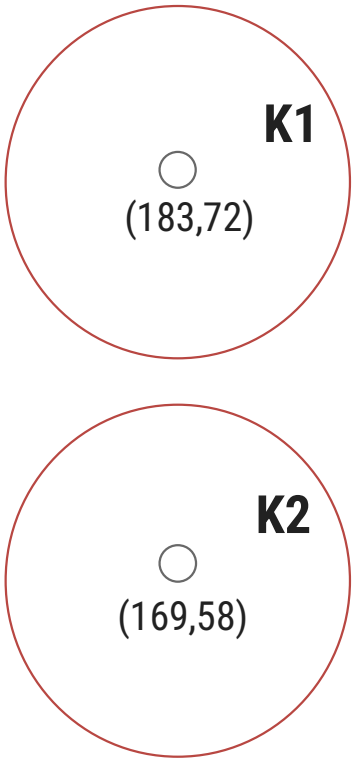
Sr.	Height	Weight
1	**185**	**72**
2	170	56
3	168	60
4	**179**	**68**
5	**182**	**72**
6	**188**	**77**
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



Now, New Centroid Calculation
For K1 = {1,4,5,6}
So, **K2** = {(185,72),(179,68),(182,72),(188,77)}
= 185+179+182+188/4 & 72+68+72+77/7
We get new centroid **C** = (183.50,72.25) ~ (183,72)

K-Means Algorithm – Example Cont..

Sr.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	**180**	**71**
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



➥ ED From **K1** ***to*** $(\mathbf{180, 71})$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(180 - 183)^2 + (71 - 72)^2}$

$= \sqrt{(-3)^2 + (-1)^2}$

$= \sqrt{9 + 1}$

$= \sqrt{10}$

$= 3.16$

➥ ED From **K2** ***to*** $(\mathbf{180, 71})$

$= \sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2}$

$= \sqrt{(180 - 169)^2 + (71 - 58)^2}$

$= \sqrt{(11)^2 + (13)^2}$

$= \sqrt{121 + 169}$

$= \sqrt{290}$

$= 17.02$

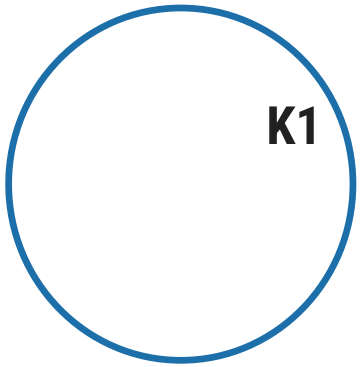
Now, data (180,71) nearer to K1, so it belongs to K1.

K1 = {1,4,5,6,7}

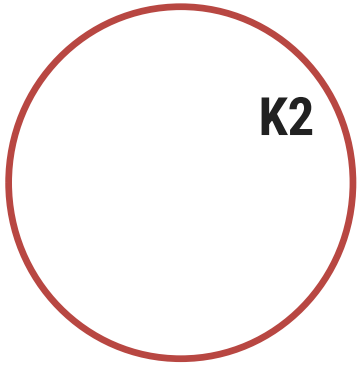
K2 = {2,3}

K-Means Algorithm – Example Cont..

Sr.	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



Cluster K1 = {1,4,5,6,7,8,9,10,11,12}



Cluster K2 = {2,3}

K-Means Algorithm Cont..

- Let us assume two clusters, and each individual's scores include two variables.
- **Step-1**
  - Choose the number of clusters.
- **Step-2**
  - Set the initial partition, and the initial mean vectors for each cluster.
- **Step-3**
  - For each remaining individual...
- **Step-4**
  - Get averages for comparison to the Cluster 1:
    - Add individual's A value to the sum of A values of the individuals in Cluster 1, then divide by the total number of scores that were summed.
    - Add individual's B value to the sum of B values of the individuals in Cluster 1, then divide by the total number of scores that were summed.

K-Means Algorithm Cont..

- **Step-5**
  - Get averages for comparison to the Cluster 2:
    - Add individual's A value to the sum of A values of the individuals in Cluster 2, then divide by the total number of scores that were summed.
    - Add individual's B value to the sum of B values of the individuals in Cluster 2, then divide by the total number of scores that were summed.
- **Step-6**
  - If the averages found in Step 4 are closer to the mean values of Cluster 1, then this individual belongs to Cluster 1, and the averages found now become the new mean vectors for Cluster 1.
  - If closer to Cluster 2, then it goes to Cluster 2, along with the averages as new mean vectors.
- **Step-7**
  - If there are more individual's to process, continue again with Step 4. Otherwise go to Step 8.
- **Step-8**
  - Now compare each individual's distance to its own cluster's mean vector, and to that of the opposite cluster.
  - The distance to its cluster's mean vector should be smaller than it distance to the other vector.
  - If not, relocate the individual to the opposite cluster.

K-Means Algorithm Cont..

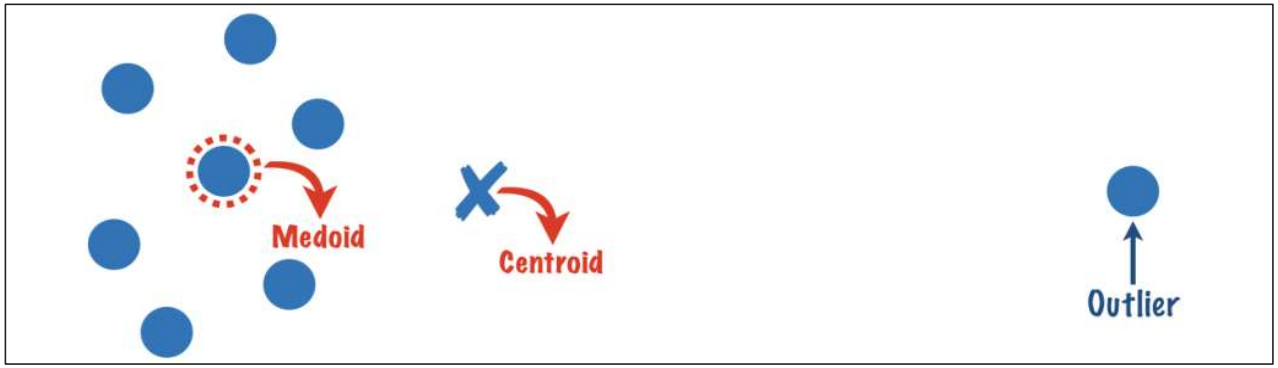
- **Step-9**
  - If any relocations occurred in Step 8, the algorithm must continue again with Step 3, using all individuals and the new mean vectors.
  - If no relocations occurred, stop. Clustering is complete.

K-Medoids Clustering Algorithm (PAM)

Section - 4

What is Medoid?

- **Medoids** are similar in concept to means or centroids, but medoids are always restricted to be members of the data set.
- **Medoids** are most commonly used on data when a mean or centroid cannot be defined, such as graphs.
- **Note:** A medoid is not equivalent to a median.



K-Medoids Clustering Algorithm (PAM)

- The **k-medoids algorithm** is a clustering algorithm related to the k-means algorithm also called as the medoid shift algorithm.
- Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups).
- In contrast to the k-means algorithm, k-medoids chooses datapoints as centers (medoids or exemplars).
- K-medoids is also a partitioning technique of clustering that clusters the data set of n objects into k clusters with k known a priori.
- It could be more robust to noise and outliers as compared to k-means because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances.
- A medoid of a finite dataset is a data point from this set, whose average dissimilarity to all the data points is minimal i.e. it is the most centrally located point in the set.

K-Medoids Clustering Algorithm (PAM) Cont..

- It was proposed in 1987 by Kaufman and Rousseeuw.
- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.
- The dissimilarity of the medoid(Ci) and object(Pi) or items is calculated by using E = |Pi - Ci|
- The cost in K-Medoids algorithm is given as $C = \sum_{Ci}^{n} \sum_{Pi \in Ci}^{n} |Pi - Ci|$
- Steps for k-medoid clustering (**Partitioning Around Medoids (PAM)**) algorithm follows as..

1. **Initialize**: randomly select *k* of the *n* data points as the medoids.
2. **Assignment step**: Associate each data point to the closest medoid.
3. **Update step**:
   - For each medoid *m* and each data point *o* associated to *m* swap *m* and *o* and compute the total cost of the configuration (that is, the average dissimilarity of *o* to all the data points associated to *m*).
   - Select the medoid *o* with the lowest cost of the configuration.
4. Repeat alternating steps 2 and 3 until there is no change in the assignments.

K-Medoids Clustering Algorithm - Example

Sr.	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5

Step 1:

Let the randomly selected 2 **medoids**, so select k = 2 and let **C1 -(4, 5)** and **C2 -(8, 5)** are the two medoids.

The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

Sr.	X	Y	Dissimilarity From C1	Dissimilarity From C2
0	8	7	|(8-4)|+|(7-5)| = 6	|(8-8)|+|(7-5)| = 2
1	3	7		
2	4	9		
3	9	6		
5	5	8		
6	7	3		
7	8	4		
8	7	5		

K-Medoids Clustering Algorithm – Example Cont..

Sr.	X	Y	Dissimilarity From C1	Dissimilarity From C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1

- Each point is assigned to the cluster of that medoid whose dissimilarity is less.
- The points **1, 2, 5 go to cluster C1** and **0, 3, 6, 7, 8 go to cluster C2**.
- The Cost = (3 + 4 + 4) + (2 + 2 + 3 + 1 + 1) = 20

K-Medoids Clustering Algorithm – Example Cont..

Sr.	X	Y	Dissimilarity From C1	Dissimilarity From C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
8	7	5	3	2

- **Step 3: randomly select one non-medoid point and recalculate the cost.**
- Let the randomly selected point be (8, 4).
- The dissimilarity of each non-medoid point with the medoids – **C1 (4, 5)** and **C2 (8, 4)** is calculated and tabulated.

K-Medoids Clustering Algorithm – Example Cont..

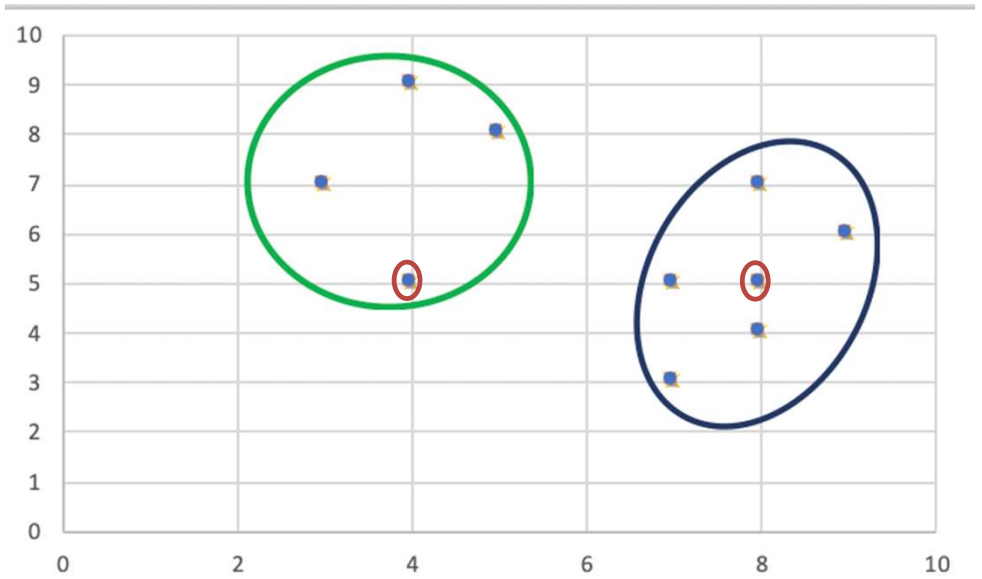
Sr.	X	Y	Dissimilarity From C1	Dissimilarity From C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
8	7	5	3	2

- Each point is assigned to that cluster whose dissimilarity is less. **So, the points 1, 2, 5 go to cluster C1** and **0, 3, 4, 6, 8 go to cluster C2**.
- The New cost,
  = (3 + 4 + 4) + (3 + 3 + 1 + 2 + 2) = 22
- Swap Cost = New Cost – Previous Cost
  = 22 – 20
  = 2
- So, 2>0 that is positive, now our previous medoid is best.
- **The total cost of Medoid (8,4) > the total cost when (8,5) was the medoid earlier & it generates the same clusters as earlier.**
- If you get negative then you have to take new medoid and recalculate again.

K-Medoids Clustering Algorithm – Example Cont..

Sr.	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	**8**	**5**
5	5	8
6	7	3
7	8	4
8	7	5
9	**4**	**5**

- As the swap cost is not less than zero, we undo the swap.
- Hence (4, 5) and (8, 5) are the final medoids.
- The clustering would be in the following way



K-Medoids Clustering Algorithm (Try Yourself!!)

Sr.	X	Y
0	2	6
1	3	4
2	3	8
3	4	7
4	6	2
5	6	4
6	7	3
7	7	4
8	8	5
9	7	6

Hierarchical Clustering Techniques

Section - 5

Hierarchical Clustering

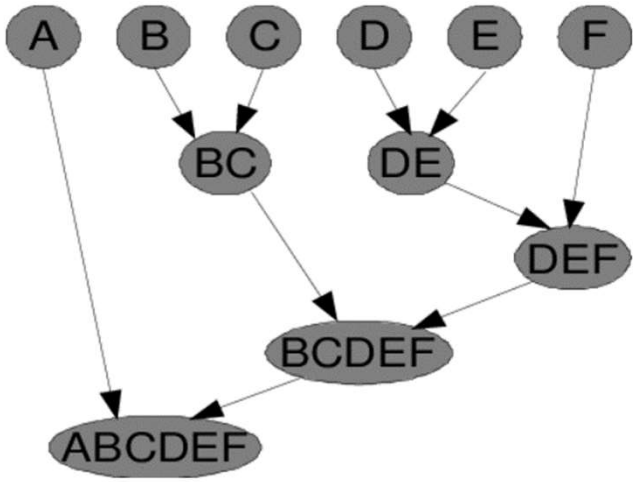
- Hierarchical Clustering is a technique to group objects based on distance or similarity.
- Hierarchical Clustering is called as unsupervised learning.
- Because, the machine (computer) learns mere from objects with their features and then the machine will automatically categorize those objects into groups.
- This clustering technique is divided into two types:
  - **Agglomerative**
    - In this technique, initially each data point is considered as an individual cluster.
    - At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.
  - **Divisive**
    - Divisive Hierarchical clustering is exactly the opposite of the Agglomerative Hierarchical clustering.
    - In Divisive Hierarchical clustering, we consider all the data points as a single cluster and in each iteration, we separate the data points from the cluster which are not similar.
    - Each data point which is separated is considered as an individual cluster. In the end, we'll be left with n clusters.

Agglomerative Hierarchical Clustering Technique

- In this technique, initially each data point is considered as an individual cluster.
- At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.
- The basic algorithm of Agglomerative is straight forward.
  - Compute the proximity matrix
  - Let each data point be a cluster
  - Repeat: Merge the two closest clusters and update the proximity matrix
  - Until only a single cluster remains
- Key operation is the computation of the proximity of two clusters.

Agglomerative Hierarchical Clustering - Example

- To understand better let's see a pictorial representation of the Agglomerative Hierarchical clustering Technique.
- Lets say we have six data points {A,B,C,D,E,F}.
- **Step- 1**:
  - In the initial step, we calculate the proximity of individual points and consider all the six data points as individual clusters as shown in the image below.

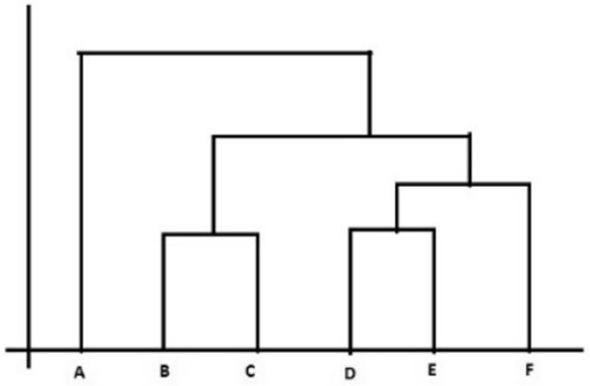


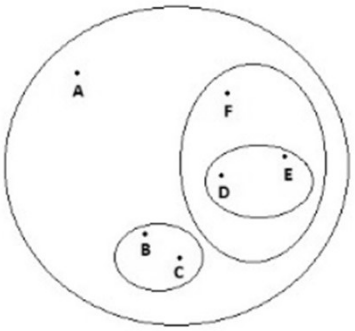
Agglomerative Hierarchical Clustering - Example

- **Step- 2**:
  - In step two, similar clusters are merged together and formed as a single cluster.
  - Let's consider B,C, and D,E are similar clusters that are merged in step two.
  - Now, we're left with four clusters which are A, BC, DE, F.
- **Step- 3**:
  - We again calculate the proximity of new clusters and merge the similar clusters to form new clusters A, BC, DEF.
- **Step- 4**:
  - Calculate the proximity of the new clusters.
  - The clusters DEF and BC are similar and merged together to form a new cluster.
  - We're now left with two clusters A, BCDEF.
- **Step- 5**:
  - Finally, all the clusters are merged together and form a single cluster.

Agglomerative Hierarchical Clustering - Example

- The Hierarchical clustering Technique can be visualized using a **Dendrogram**.
- A **Dendrogram** is a tree-like diagram that records the sequences of merges or splits.





Divisive Hierarchical Clustering Technique

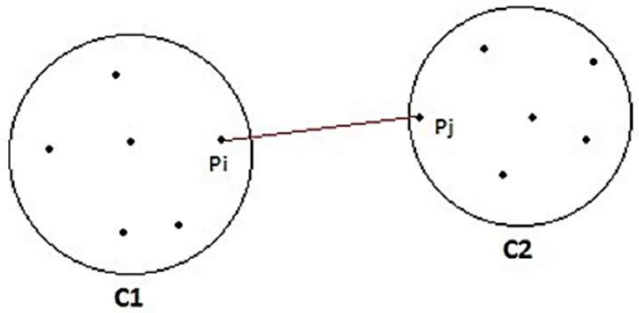
- Divisive Hierarchical clustering is exactly the opposite of the Agglomerative Hierarchical clustering.
- In Divisive Hierarchical clustering, we consider all the data points as a single cluster and in each iteration, we separate the data points from the cluster which are not similar.
- Each data point which is separated is considered as an individual cluster.
- In the end, we'll be left with **n** clusters.
- As we are dividing the **single** clusters into **n** clusters, it is named as Divisive Hierarchical clustering.
- It is not much used in the real world.

Divisive Hierarchical Clustering Technique Cont..

- Calculating the similarity between two clusters is important to merge or divide the clusters.
- There are certain approaches which are used to calculate the similarity between two clusters:
  - MIN
  - MAX
  - Group Average
  - Distance Between Centroids
  - Ward's Method

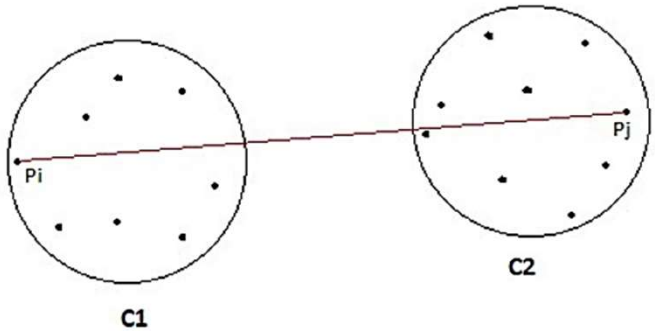
MIN

- Min is known as **single-linkage algorithm** can be defined as the similarity of two clusters C1 and C2 is equal to the minimum of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.
- Mathematically this can be written as,
  - Sim(C1,C2) = Min Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2
- In simple words, pick the two closest points such that one point lies in cluster one and the other point lies in cluster 2 and takes their similarity and declares it as the similarity between two clusters.



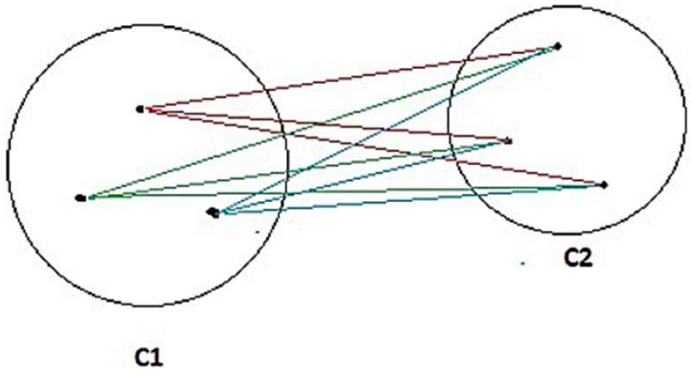
MAX

- Max is known as the **complete linkage algorithm**, this is exactly opposite to the MIN approach.
- The similarity of two clusters C1 and C2 is equal to the maximum of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.
- Mathematically this can be written as,
  - Sim(C1,C2) = Max Sim(Pi,Pj) such that Pi ∈ C1 & Pj ∈ C2
- In simple words, pick the two farthest points such that one point lies in cluster one and the other point lies in cluster 2 and takes their similarity and declares it as the similarity between two clusters.



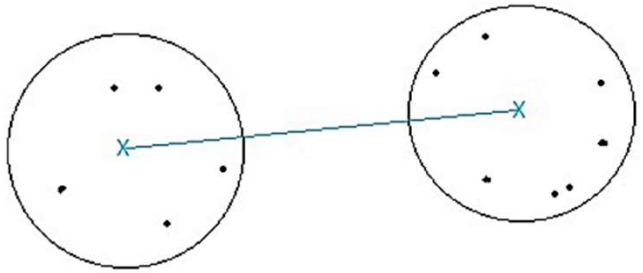
Group Average

- Take all the pairs of points and compute their similarities and calculate the average of the similarities.
- Mathematically this can be written as,
  - Sim(C1,C2) = ∑ Sim(Pi, Pj)/|C1|*|C2|, where, Pi ∈ C1 & Pj ∈ C2



Distance between centroids

- Compute the centroids of two clusters C1 & C2 and take the similarity between the two centroids as the similarity between two clusters.
- This is a less popular technique in the real world.



Ward's Method

- This approach of calculating the similarity between two clusters is exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances Pi and Pj.
- Mathematically this can be written as,
  - $Sim(C1,C2) = \sum (dist(Pi, Pj))^2/|C1|*|C2|$

Hierarchical Clustering Example

Agglomerative Hierarchical Clustering - Example

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

- Calculate Euclidean distance, create the distance matrix.
- Distance [(x,y),(a,b)] = $\sqrt{(x \quad -a \quad)^2 + (y \quad -b \quad)^2}$

↪ ED $\mathbf{P1}$ & $\boldsymbol{P2}$ $(\mathbf{0.40}, \mathbf{0.53}), (\mathbf{0.22}, \mathbf{0.38})$

$= \sqrt{(0.40 - 0.22)^2 + (0.53 - 0.38)^2}$

$= \sqrt{(0.18)^2 + (0.15)^2}$

$= \sqrt{0.0324 + 0.0225}$

$= \sqrt{0.0549}$

$= 0.23$

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3			0			
P4				0		
P5					0	
P6						0

Agglomerative Hierarchical Clustering - Example

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

➥ ED **P1** & $\boldsymbol{P3}$ $(\mathbf{0.40, 0.53}), (\mathbf{0.35, 0.32})$

$= \sqrt{(0.40 - 0.35)^2 + (0.53 - 0.32)^2}$

$= \sqrt{(0.05)^2 + (0.21)^2}$

$= \sqrt{0.0025 + 0.0441}$

$= \sqrt{0.0466}$

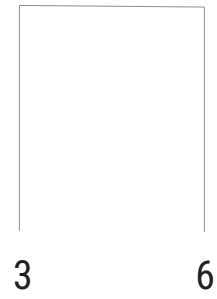
$= 0.22$

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22		0			
P4				0		
P5					0	
P6						0

Agglomerative Hierarchical Clustering - Example

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



Agglomerative Hierarchical Clustering - Example

- To Update the distance matrix MIN[dist(P3,P6),P1]
- MIN (dist(P3,P1),(P6,P1))
- Min[(0.22,0.23)]
- 0.22

- To Update the distance matrix MIN[dist(P3,P6),P2]
- MIN (dist(P3,P2),(P6,P2))
- Min[(0.15,0.25)]
- 0.15

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	**0.22**	**0.15**	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

Agglomerative Hierarchical Clustering - Example

- To Update the distance matrix MIN[dist(P3,P6),P4]
- MIN (dist(P3,P4),(P6,P4))
- Min[(0.15,0.22)]
- 0.15

- To Update the distance matrix MIN[dist(P3,P6),P5]
- MIN (dist(P3,P5),(P6,P5))
- Min[(0.28,0.39)]
- 0.28

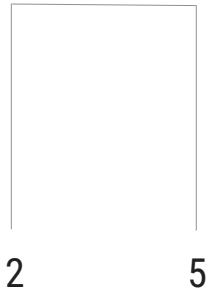
	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	**0.15**	0		
P5	0.34	0.14	**0.28**	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

Agglomerative Hierarchical Clustering - Example

- The Updated distance matrix for cluster P3, P6

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



Agglomerative Hierarchical Clustering - Example

- To Update the distance matrix MIN[dist(P2,P5),P1]
- MIN (dist(P2,P1),(P5,P1))
- Min[(0.23,0.34)]
- 0.23

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

- To Update the distance matrix MIN[dist(P2,P5),(P3,P6)]
- MIN [(dist(P2,(P3,P6)),(P5,(P3,P6))]
- Min[(0.15,0.28)]
- 0.15

Agglomerative Hierarchical Clustering - Example

- To Update the distance matrix MIN[dist(P2,P5),P4]
- MIN (dist(P2,P4),(P5,P4))
- Min[(0.20,0.29)]
- 0.20

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

Agglomerative Hierarchical Clustering - Example

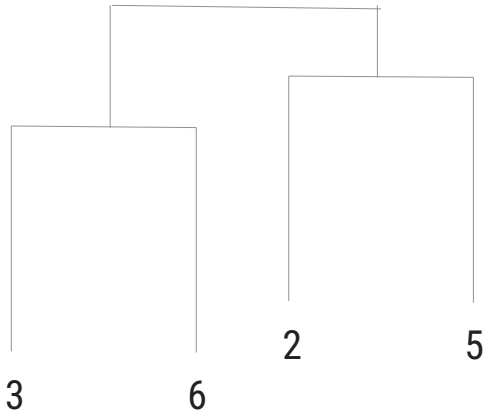
	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

Agglomerative Hierarchical Clustering - Example

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

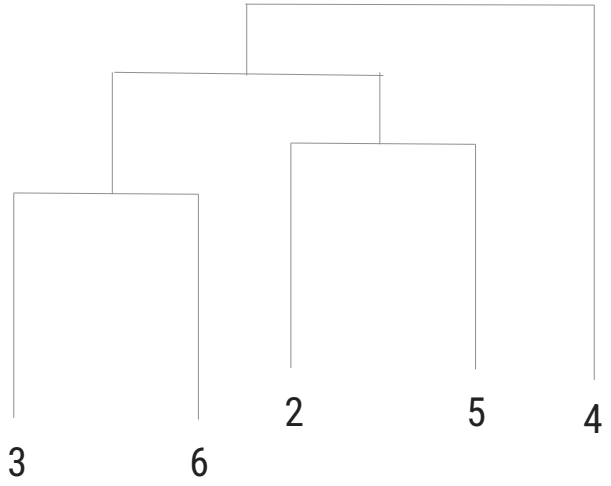


Agglomerative Hierarchical Clustering - Example

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

To Update the distance matrix MIN[dist(P2,P5),(P3,P6)),P1]
MIN (dist(P2,P5),P1),((P3,P6),P1)]
Min[(0.23,0.22)]
0.22

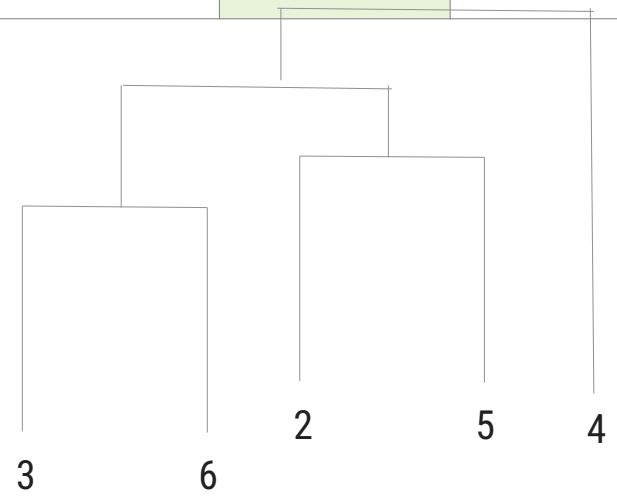
To Update the distance matrix MIN[dist(P2,P5),(P3,P6)),P4]
MIN (dist(P2,P5),P4),((P3,P6),P4)]
Min[(0.20,0.15)]
0.15



	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

Agglomerative Hierarchical Clustering - Example

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

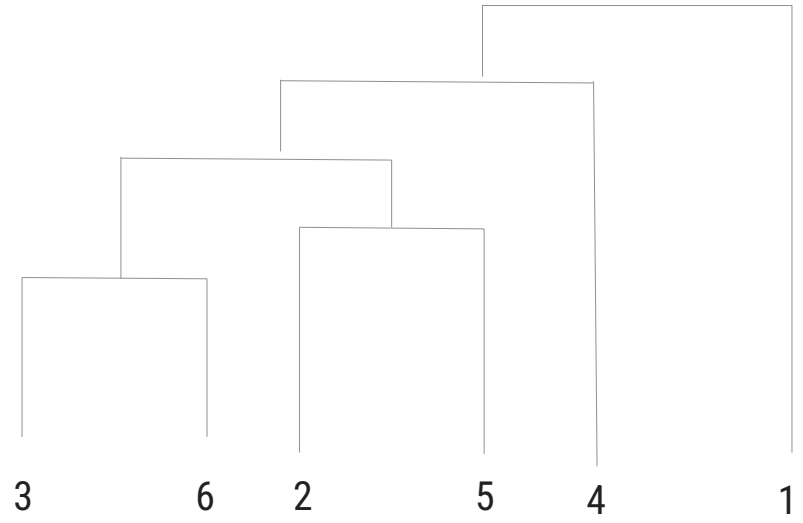


To Update the distance matrix MIN[dist(P2,P5,P3,P6),P4]
MIN (dist(P2,P5,P3,P6),P1),(P4,P1)]
Min[(0.22,0.37)]
0.22

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0

Agglomerative Hierarchical Clustering - Example

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30



Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit – 5
Cluster Analysis

Thank You

Any Questions ?

Prof. Naimish R. Vadodariya
Computer Engineering
Darshan Institute of Engineering & Technology, Rajkot
naimish.Vadodariya@darshan.ac.in
8866215253

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit-6
Web mining and Other data mining

Prof. Naimish R. Vadodariya
Computer Engineering Department
Darshan Institute of Engineering & Technology, Rajkot
naimish.vadodariya@darshan.ac.in
8866215253

Topics to be covered

- Web Mining: Introduction to Web Mining
  - Web content mining
  - Web structure mining
  - Web usage mining
- Text Mining
- Spatial Data Mining
- Temporal Mining
- Multimedia Mining
- Big Data
- Web log structure
- Applications of Distributed and parallel Data Mining

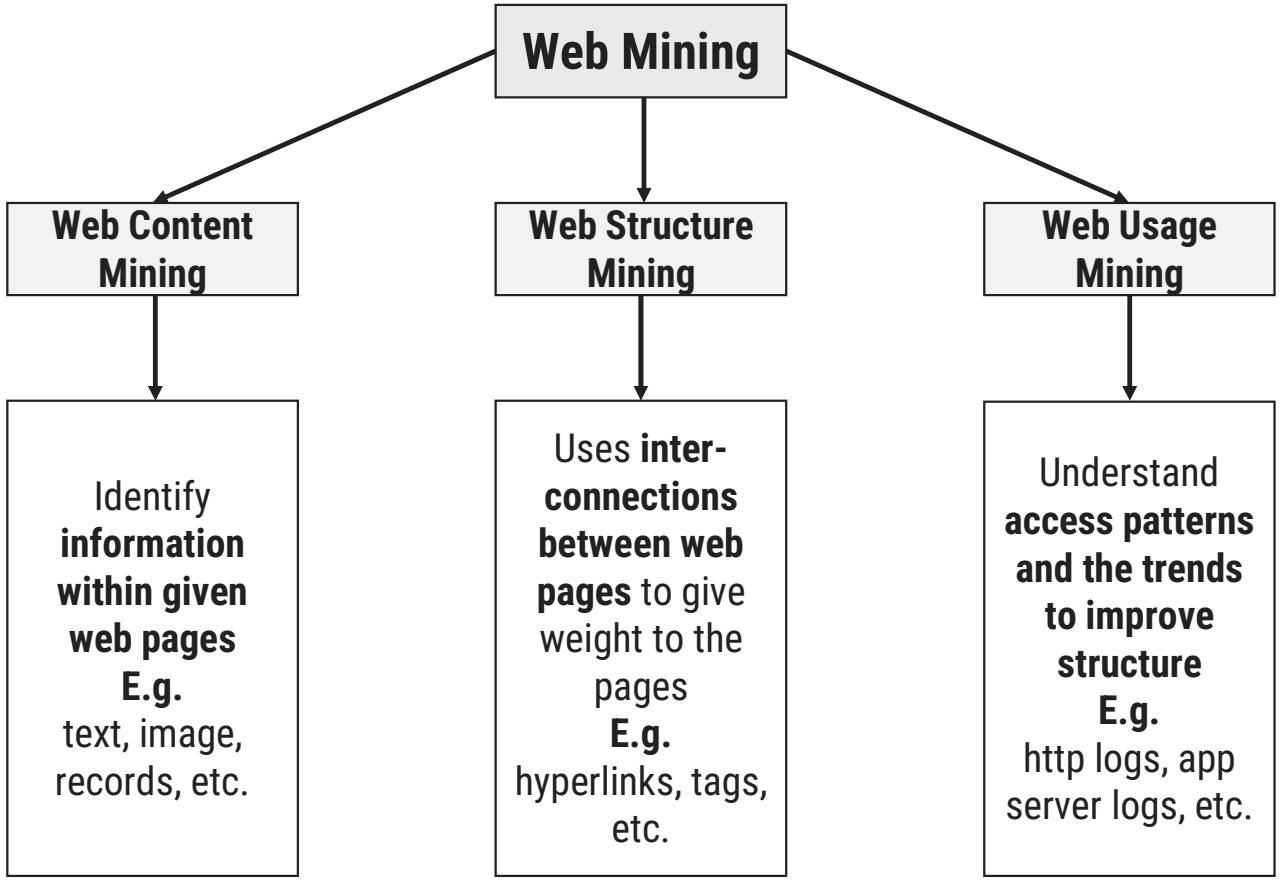
Web Mining: Introduction to Web Mining

Section - 1

What is Web Mining?

- Web mining is the use of data mining techniques to automatically discover and **extract information from web documents** and services.
- There are general classes of information that can be discovered in web mining: **web activity, from server logs** and **web browser activity tracking**.
- Web mining can be broadly divided into three categories, according to the kinds of data to be mined.
  - **Web Content Mining**
  - **Web Structure Mining**
  - **Web Usage Mining**

What is Web Mining?



Text Mining

Section - 2

Text Mining

- **Text mining**, also referred to as **text data mining**, roughly equivalent to **text analytics**, is the process of deriving **high-quality information from text**.
- With the advancement of technology, more and more data is available in digital form, among them, **most of the data** (approx. 85%) is in **unstructured textual form**.
- Compared with the kind of data stored in databases, text is unstructured, ambiguous, and difficult to process.
- Nevertheless, in modern culture, **text** is the **most communal way for the formal exchange of information**.
- It has become essential to develop better techniques and algorithms to **extract useful and interesting information** from this large amount of **textual data**.

Text Mining Cont..

Information Retrieval

- The ability to query a computer system to return relevant results.
- The most widely used example is the **google web search engine**.

Natural Language Processing (NLP)

- Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.
- NLP is one of the oldest and most challenging problems in the field of **artificial intelligence**.
- It is related to **study of human language** so that computers can understand **natural languages** as humans do.
- NLP research pursues the question of how we understand the **meaning of a sentence or a document**.
- While **words - nouns, verbs, adverbs and adjectives** - are the building blocks of meaning, it is their correlation to each other within the structure of a sentence in a document.

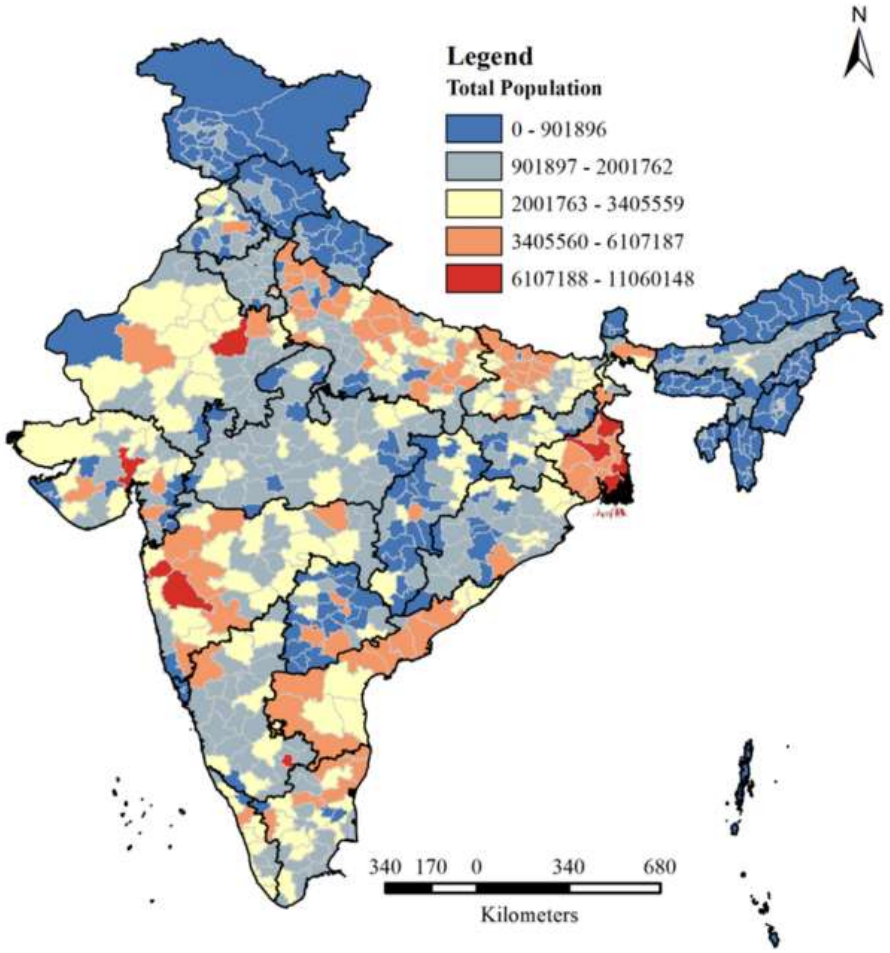
Spatial Data Mining

Section - 3

Spatial Data Mining

- Spatial data mining is the application of **data mining** to **spatial models**.
- Spatial data mining is based on **geographical analysis.**
- In spatial data mining, **analysts use geographical or spatial information to produce business intelligence** or other results.
- It requires **specific techniques and resources to get the geographical data** into relevant and useful formats.
- The task is to search for **spatial patterns**.

Spatial Data Mining Cont..



Temporal Mining

Section - 4

Temporal Mining

- **Temporal** refers to time.
- It describes a phenomenon in a certain location and time — for example, shipping movements across a geographic area over time.
- **Temporal data mining** refers to the extraction of implicit, non-trivial, and potentially useful abstract information from large collections of **temporal data**.
- There are several **mining** tasks that can be applied on **temporal data**, most of which directly extend from the corresponding mining tasks on **general data** types.

Big Data

Section - 5

Big Data

- Big Data is a term used for any **data that is large in quantity**.
- It is used to refer to any kind of data that is **difficult to be represent** using conventional methods like database management systems or MS excel.
- Big data challenges include **capturing data**, **data storage**, **data analysis**, **data search**, **data sharing**, **data transfer**, **visualization of data**, **querying**, **updating** and **data security**.
- Big data is an evolving term that describes any **large amount of structured**, **semi structured and unstructured data** that has the potential to be mined for information.
- 3 v's of big data can be described as follows.
  - **Volume**
  - **Velocity**
  - **Veracity**

1) Volume

- A typical PC might have had **10 gigabytes(GB) of storage in 2000**.
- Today, Facebook ingests **700 terabytes(TB) of new data every day**.
- **Boeing 737** will generate 240 terabytes(TB) of flight data during a single flight across the US.
- The smart phones, it creates and consumes data.
- Sensors embedded into every objects will soon result in billions of new data
- Constantly-updated data feeds containing environmental, location, and other information, including videos, images etc.

2) Velocity

- **Clickstreams and ad impressions** capture user behavior at millions of events per second.
- High-frequency stock trading algorithms reflect **market changes within microseconds**.
- Machine to machine processes exchange data between **billions of devices.**
- **Infrastructure** and **sensors** generate **massive log data** in real-time.
- **On-line gaming systems** support millions of concurrent users, each producing multiple inputs per second.

3) Veracity

- Big Data isn't just a numbers, dates, and strings or text.
- It is also **geospatial data, 3D data, audio and video, unstructured text, including log files and social media**.
- Traditional database systems were designed to address smaller volumes of structured data, fewer updates or a predictable consistent data structure.
- Big Data analysis includes **different types of data.**

Multimedia Mining

Section - 6

Multimedia Mining

- Multimedia data mining refers to the **mining of multimedia content**.
- In other words, it is study of large amounts of multimedia information in order to find patterns or statistical relationships.
- Once data is collected, computer programs are used to analyze it and look for meaningful connections.
- **Examples**
  - Multimedia data mining for Traffic Video Sequences
    - Traffic camera footage to analyze traffic flow.
    - This would come in handy while planning new streets, expanding existing streets, or diverting traffic.
    - The same can be used by the Government organizations and city planners to help traffic flow more smoothly and quickly.
  - Multimedia data mining in Digital Libraries
    - The Digital library retrieves, stores and preserves the digital data.
    - For this purpose, there is a need to convert different formats of information such as text, images, video, audio, etc. Thus, in the process of conversion of the multimedia files in the libraries, the data mining techniques are popular.

Web Log Structure

Section - 7

What is Web Log File?

- Web log file is log file automatically created and maintained by a web server.
- Every "hit" to the web site, including each view of a HTML document, image or other object, is logged.
- The raw web log file format is essentially one line of text for each hit to the web site.
- This contains information about who was visiting the site, where they came from, and exactly what they were doing on the web site.
- **Example**

```
66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html HTTP/1.1" 200 11179 "-
" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET / HTTP/1.1" 200 10801
"http://www.google.com/search?q=log+analyzer&ie=utf-8&oe=utf-8
&aq=t&rls=org.mozilla:en-US:official&client=firefox-a" "Mozilla/5.0 (Windows; U; Windows NT
5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7"

111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET /style.css HTTP/1.1" 200 3225
"http://www.loganalyzer.net/" "Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7)
Gecko/20070914 Firefox/2.0.0.7"
```

Web log file structure

- **Fields**
  - Different servers have different log formats. Nevertheless the data in this log fragment is pretty typical of the information available. Let's look at one line from the above fragment (split for easier viewing).
- 111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET / HTTP/1.1" 200 10801 "http://www.google.com/search?q=log+analyzer&ie=utf-8&oe=utf-8 &aq=t&rls=org.mozilla:en-US:official&client=firefox-a" "Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7“

Fields
"111.111.111.111"
"-"
"-"
[08/Oct/2007:11:17:55 -0400]
"GET / HTTP/1.1"
"200"
10801
http://www.google.com/search?q=log+analyzer&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a
"Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7"

Web log file structure Cont..

Fields	Description
IP address : "111.111.111.111"	• IP address of the machine that contacted our site

Web log file structure Cont..

Fields	Description
Bytes transferred : "10801"	• The number of bytes transferred. • This tells you how many bytes were transferred to the user, i.e. the bandwidth used.

Distributed and Parallel Data Mining

Section - 8

Distributed Data Mining

- Data mining algorithms deal predominantly with simple data formats (typically flat files).
- There is an increasing amount of focus on mining complex and advanced data types such as object-oriented, spatial and temporal data.
- Another aspect of this growth and evolution of data mining systems is the move from stand-alone systems using centralized and local computational resources towards supporting increasing levels of distribution.
- Databases in today's information age are inherently distributed.
- An Organizations that operate in global markets need to perform data mining on distributed data sources (homogeneous / heterogeneous) and require cohesive and integrated knowledge from this data.
- Such organizational environments are characterized by a geographical separation of users from the data sources.

Distributed Data Mining Cont..

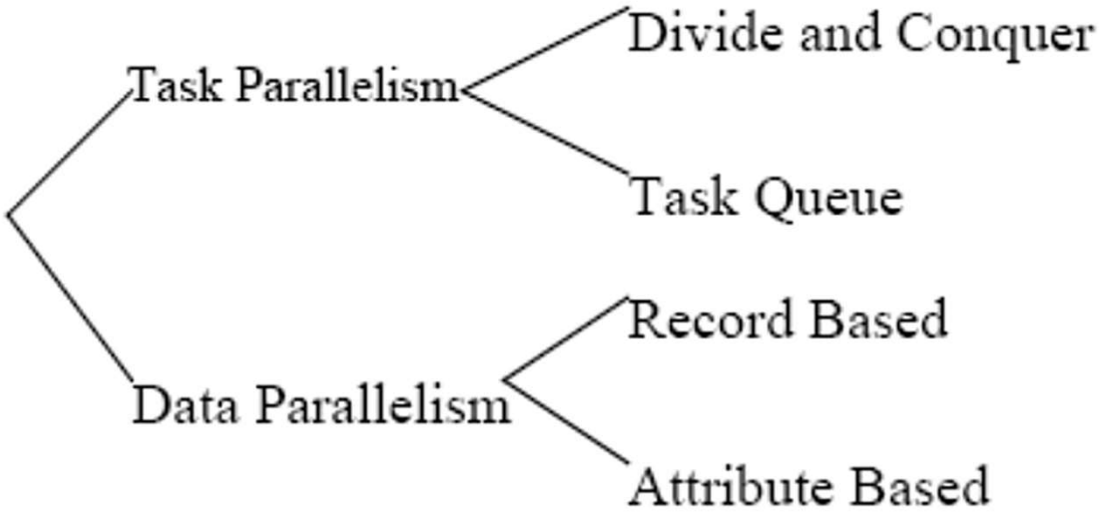
- This inherent distribution of data sources and large volumes of data involved inevitably leads to exorbitant communications costs.
- Broadly, data mining environments consist of users, data, hardware and the mining software (this includes both the mining algorithms and any other associated programs).
- Distributed data mining addresses the impact of distribution of users, software and computational resources on the data mining process.
- There is general consensus that distributed data mining is the process of mining data that has been partitioned into one or more physically/geographically distributed subsets.

Distributed Data Mining Cont..

- The significant factors, which have led to the emergence of distributed data mining from centralized mining, are as follows:
  - The need to mine **distributed subsets of data**, the integration of which is non- trivial and expensive.
  - The performance and scalability bottle necks of data mining.
  - Distributed data mining provides a framework for scalability, which allows the splitting up of larger datasets with high dimensionality into smaller subsets that require computational resources individually.

Parallel Data Mining

- For compute-intensive applications, parallelization is an obvious means for improving performance and achieving scalability.
- A variety of techniques may be used to distribute the workload involved in data mining over multiple processors.
- Four major classes of parallel implementations are distinguished.



Parallel Data Mining Cont..

- Task-parallel algorithms assign portions of the search space to separate processors
- The task parallel approaches can again be divided into two groups.
- The first group is based on a Divide and Conquer strategy that divides the search space and assigns each partition to a specific processor.
- The second group is based on a task queue that dynamically assigns small portions of the search space to a processor whenever it becomes available.
- A task parallel implementation of decision tree induction will form tasks associated with branches of the tree.
- A Divide and Conquer approach seems a natural reflection of the recursive nature of decision trees.

Parallel Data Mining Cont..

- However the task of parallel implementation suffers from load balancing problems caused by uneven distributions of records between branches.
- The success of a task parallel implementation of decision trees seems to be highly dependent on the structure of the data set. The second class of approaches, called data parallel, distributes the data set over the available processors.
- Data-parallel approaches come in two flavors.
  - A partitioning based on records will assign non-overlapping sets of records to each of the processors.
  - Alternatively a partitioning of attributes will assign sets of attributes to each of the processors.
- Attribute based approaches are based on the observation that many algorithms can be expressed in terms of primitives that consider every attribute in turn.
- If attributes are distributed over multiple processors, these primitives may be executed in parallel. For example, when constructing decision trees, at each node in the tree, all independent attributes are considered, in order to determine the best split at that point.

Data Mining (DM)
GTU #3160714

Dedicated Faculty, Committed Education
Darshan
Institute of Engineering & Technology

Unit – 6
Web mining and
Other data mining

Thank You

Any Questions ?

Prof. Naimish R. Vadodariya
Computer Engineering
Darshan Institute of Engineering & Technology, Rajkot
naimish.Vadodariya@darshan.ac.in
8866215253