School of Computer Science

DATA MINING

COURSE FILE

Program Name : B.TECH- CSE
Course : DATA MINING
Course Code : (CS4102PC)
Year& Semester : IV B.TECH ,I SEM
Faculty Name : Bantu Mahesh

COURSE CONTENTS UNITS 1 TO 5

CHAPTER-1

What Is Data Mining?

Data mining refers to extracting or mining knowledge from large amountsof data. The term is actually a misnomer. Thus, data mining should have been more appropriately named as knowledge mining which emphasis on mining from large amounts of data. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. The key properties of data mining are Automatic discovery of patterns Prediction of likely outcomes Creation of actionable information Focus on large datasets and databases

The Scope of Data Mining

Data mining derives its name from the similarities between searching for valuable business information in a large database — for example, finding linked products in gigabytes of store scanner data — and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

Automated prediction of trends and behaviors. Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

Automated discovery of previously unknown patterns. Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

Tasks of Data Mining

Data mining involves six common classes of tasks:

Anomaly detection (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.

Association rule learning (Dependency modelling) – Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

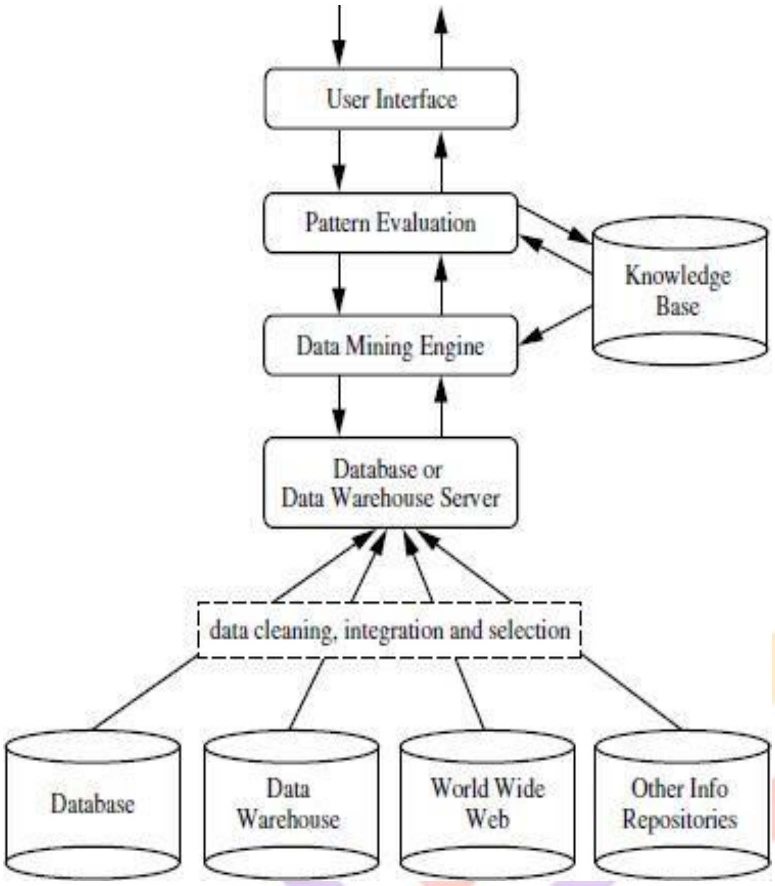
Clustering – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.

Classification – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam". Regression – attempts to find a function which models the data with the least error.

Summarization – providing a more compact representation of the data set, includingvisualization and report generation.

Architecture of Data Mining

A typical data mining system may have the following major components.



1. Knowledge Base:

This is the domain knowledge that is used to guide the search orevaluate the interestingness of resulting patterns. Such knowledge can include concepthierarchies,
used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).

2. Data Mining Engine:

This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.

3. Pattern Evaluation Module:

This component typically employs interestingness measures interacts with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.

4. User interface:

This module communicates between users and the data mining system,allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this componentallows the user to browse database and data warehouse schemas or data structures,evaluate mined patterns, and visualize the patterns in different forms.

Data Mining Process:

Data Mining is a process of discovering various models, summaries, and derived values from a given collection of data.

The general experimental procedure adapted to data-mining problems involves the following steps:

1. State the problem and formulate the hypothesis

Most data-based modeling studies are performed in a particular application domain. Hence, domain-specific knowledge and experience are usually necessary in order to come up with a meaningful problem statement. Unfortunately, many application studies tend to focus on the data-mining technique at the expense of a clear problem statement. In this step, a modeler usually specifies a set of variables for the unknown dependency and, if possible, a general form of this dependency as an initial hypothesis. There may be several hypotheses formulated for a single problem at this stage. The first step requires the combined expertise of an application domain and a data-mining model. In practice, it usually means a close interaction between the data-mining expert and the application expert. In successful data-mining applications, this cooperation does not stop in the initial phase; it continues during the entire data-mining process.

2. Collect the data

This step is concerned with how the data are generated and collected. In general, there are two distinct possibilities. The first is when the data-generation process is under the control of an expert (modeler): this approach is known as a designed experiment. The second possibility is when the expert cannot influence the data- generation process: this is known as the observational approach. An observational setting, namely, random data generation, is assumed in most data-mining applications. Typically, the sampling distribution is completely unknown after data are collected, or it is partially and implicitly given in the data-collection procedure. It is very important, however, to understand how data collection affects its theoretical distribution, since such a priori knowledge can be very useful for modeling and, later, for the final interpretation of results. Also, it is important to make sure that the data used for estimating a model and the data used later for testing and applying a model come from the same, unknown, sampling distribution. If this is not the case, the estimated model cannot be successfully used in a final application of the results.

Preprocessing the data

In the observational setting, data are usually "collected" from the existing databses, data warehouses, and data marts. Data preprocessing usually includes at least two common tasks:

1. Outlier detection (and removal) – Outliers are unusual data values that are not consistent with most observations. Commonly, outliers result from measurement errors, coding and recording errors, and, sometimes, are natural, abnormal values. Such nonrepresentative samples can seriously affect the model produced later. There are two strategies for dealing with outliers:

a. Detect and eventually remove outliers as a part of the preprocessing phase, or
b. Develop robust modeling methods that are insensitive to outliers.

2. Scaling, encoding, and selecting features – Data preprocessing includes several steps such as variable scaling and different types of encoding. For example, one feature with the range [0, 1] and the other with the range [−100, 1000] will not have the same weights in the applied technique; they will also influence the final data-mining results differently. Therefore, it is recommended to scale them and bring both features to the same weight for further analysis. Also, application-specific encoding methods usually achieve dimensionality reduction by providing a smaller number of informative features for subsequent data modeling.
These two classes of preprocessing tasks are only illustrative examples of a large spectrum of preprocessing activities in a data-mining process.

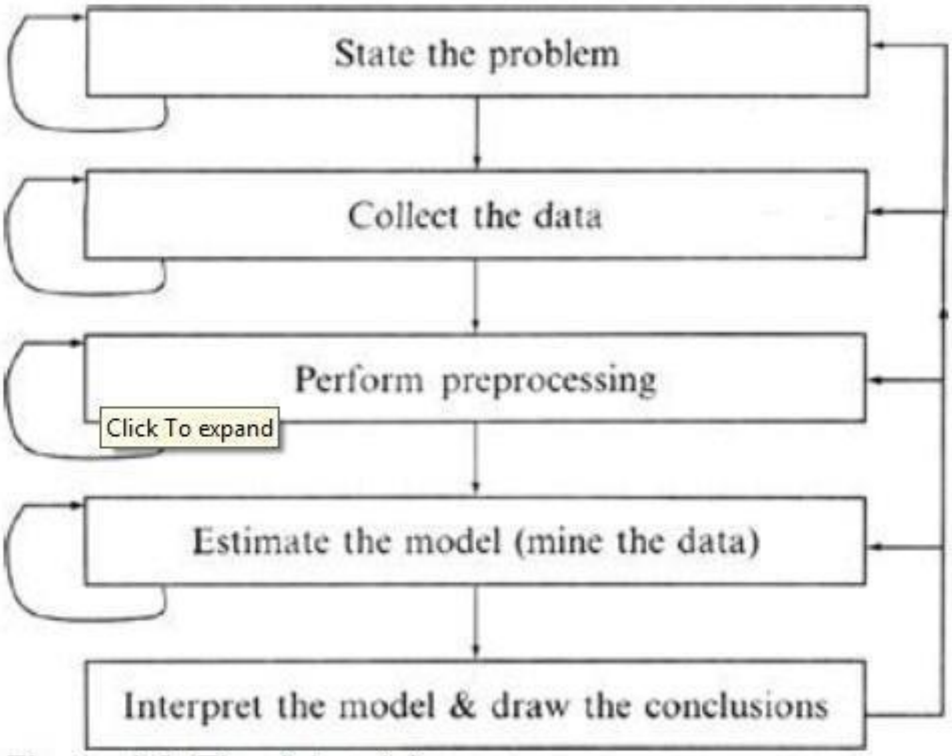
Data-preprocessing steps should not be considered completely independent from other data-mining phases. In every iteration of the data-mining process, all activities, together, could define new and improved data sets for subsequent iterations. Generally, a good preprocessing method provides an optimal representation for a data-mining technique by incorporating a priori knowledge in the form of application-specific scaling and encoding.

Estimate the model

The selection and implementation of the appropriate data-mining technique is the main task in this phase. This process is not straightforward; usually, in practice, the implementation is based on several models, and selecting the best one is an additional task. The basic principles of learning and discovery from data are given in Chapter 4 of this book. Later, Chapter 5 through 13 explain and analyze specific techniques that are applied to perform a successful learning process from data and to develop an appropriate model.

Interpret the model and draw conclusions

In most cases, data-mining models should help in decision making. Hence, such models need to be interpretable in order to be useful because humans are not likely to base their decisions on complex "black-box" models. Note that the goals of accuracy of the model and accuracy of its interpretation are somewhat contradictory. Usually, simple models are more interpretable, but they are also less accurate. Modern data-mining methods are expected to yield highly accurate results using highdimensional models. The problem of interpreting these models, also very important, is considered a separate task, with specific techniques to validate the results. A user does not want hundreds of pages of numeric results. He does not understand them; he cannot summarize, interpret, and use them for successful decision making.



Data Mining Process

Classification of Data mining Systems:

The data mining system can be classified according to the following criteria:

Database Technology Statistics

Machine Learning Information Science Visualization Other Disciplines

Some Other Classification Criteria:

Classification according to kind of databases mined Classification according to kind of knowledge mined Classification according to kinds of techniques utilized Classification according to applications adapted

Classification according to kind of databases mined

We can classify the data mining system according to kind of databases mined. Database system can be classified according to different criteria such as data models, types of data etc. And the data mining system can be classified accordingly. For example if we classify the database according to data model then we may have a relational, transactional, object-relational, or data warehouse mining system.

Classification according to kind of knowledge mined

We can classify the data mining system according to kind of knowledge mined. It is means data mining system are classified on the basis of functionalities such as:

Characterization Discrimination

Association and Correlation Analysis Classification

Prediction Clustering

Outlier Analysis Evolution Analysis

Classification according to kinds of techniques utilized

We can classify the data mining system according to kind of techniques used. We can describes these techniques according to degree of user interaction involved or the methods of analysis employed.

Classification according to applications adapted

We can classify the data mining system according to application adapted. These applications are as follows:

Finance Telecommunications

DNA

Stock Markets E-mail

Major Issues and Challenges In Data Mining:

Mining different kinds of knowledge in databases. - The need of different users is not the same. And Different user may be in interested in different kind of knowledge. Therefore it is necessary for data mining to cover broad range of knowledge discovery task.

Interactive mining of knowledge at multiple levels of abstraction. - The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on returned results.

Incorporation of background knowledge. - To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple level of abstraction.

Data mining query languages and ad hoc data mining. - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.

Presentation and visualization of data mining results. - Once the patterns are discovered it needs to be expressed in high level languages, visual representations. This representations should be easily understandable by the users.
Handling noisy or incomplete data. - The data cleaning methods are required that can handle the noise, incomplete objects while mining the data regularities. If data cleaning methods are not there then the accuracy of the discovered patterns will be poor.

Pattern evaluation. - It refers to interestingness of the problem. The patterns discovered should be interesting because either they represent common knowledge or lack novelty.

Efficiency and scalability of data mining algorithms. - In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.

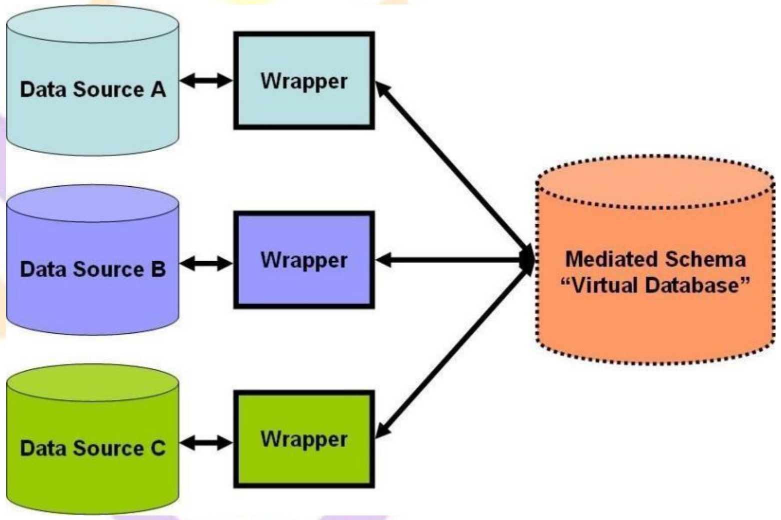
Parallel, distributed, and incremental mining algorithms. - The factors such as huge size of databases, wide distribution of data,and complexity of data mining methods

motivate the development of parallel and distributed data mining algorithms. These algorithm divide the data into partitions which is further processed parallel. Then the results from the partitions is merged. The incremental algorithms, updates databases without having mine the data again from scratch.

Knowledge Discovery in Databases(KDD)
Some people treat data mining same as Knowledge discovery while some people view data mining essential step in process of knowledge discovery. Here is the list of steps involved in knowledge discovery process:

Data Cleaning - In this step the noise and inconsistent data is removed.
Data Integration - In this step multiple data sources are combined.
Data Selection - In this step relevant to the analysis task are retrieved from the database.
Data Transformation - In this step data are transformed or consolidated into formsappropriate for mining by performing summary or aggregation operations.
Data Mining - In this step intelligent methods are applied in order to extract data patterns.
Pattern Evaluation - In this step, data patterns are evaluated.
Knowledge Presentation - In this step,knowledge is represented.

Data Preprocessing:



Data Integration:

It combines datafrom multiple sources into a coherent data store, as in data warehousing. These sourcesmay include multiple databases, data cubes, or flat files.

The data integration systems are formally defined as triple<G,S,M> Where G: The global schema

S:Heterogeneous source of schemas

M: Mapping between the queries of source and global schema

Issues in Data integration:

Schema integration and object matching:

How can the data analyst or the computer be sure that customer id in one database and customer number in another reference to the same attribute.

Redundancy:

An attribute (such as annual revenue, forinstance) may be redundant if it can be derived from another attribute or set ofattributes. Inconsistencies in attribute or dimension naming can also cause redundanciesin the resulting data set.

Data Transformation:

In data transformation, the data are transformed or consolidated into forms appropriatefor mining.

Data transformation can involve the following:

Smoothing, which works to remove noise from the data. Such techniques includebinning,regression, and clustering.

Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annualtotal amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.

Generalization of the data, where low-level or ―primitive‖ (raw) data are replaced byhigher-level concepts through the use of concept hierarchies. For example, categoricalattributes, like street, can be generalized to higher-level concepts, like city or country.

Normalization, where the attribute data are scaled so as to fall within a small specifiedrange, such as 1:0 to 1:0, or 0:0 to 1:0.

Attribute construction (or feature construction),wherenewattributes are constructedand added from the given set of attributes to help the mining process.

Data Reduction:

Data reduction techniques can be applied to obtain a reduced representation of thedata set that ismuch smaller in volume, yet closely maintains the integrity of the originaldata. That is, mining on the reduced data set should be more efficient yet produce thesame (or almost the same) analytical results.

Strategies for data reduction include the following:

Data cube aggregation, where aggregation operations are applied to the data in theconstruction of a data cube.

Attribute subset selection, where irrelevant, weakly relevant, or redundant attributesor dimensions may be detected and removed.

Dimensionality reduction, where encoding mechanisms are used to reduce the dataset size.

Numerosityreduction,where the data are replaced or estimated by alternative, smallerdata representations such as parametric models (which need store only the modelparameters instead of the actual data) or nonparametric methods such as clustering,sampling, and the use of histograms.

Discretization and concept hierarchy generation,where rawdata values for attributesare replaced by ranges or higher conceptual levels. Data discretization is a form ofnumerosity reduction that is very useful for the automatic generation of concept hierarchies.Discretization and concept hierarchy generation are powerful tools for datamining, in that they allow the mining of data at multiple levels of abstraction.

Types of Data Mining

Each of the following data mining techniques serves several different business problems and provides a different insight into each of them. However, understanding the type of business problem you need to solve will also help in knowing which technique will be best to use, which will yield the best results. The Data Mining types can be divided into two basic parts that are as follows:

Predictive Data Mining Analysis

Descriptive Data Mining Analysis

1. Predictive Data Mining

As the name signifies, Predictive Data-Mining analysis works on the data that may help to know what may happen later (or in the future) in business. Predictive Data-Mining can also be further divided into four types that are listed below:

Classification Analysis

Regression Analysis

Time Serious Analysis

Prediction Analysis

2. Descriptive Data Mining

The main goal of the Descriptive Data Mining tasks is to summarize or turn given data into relevant information. The Descriptive Data-Mining Tasks can also be further divided into four types that are as follows:

Clustering Analysis

Summarization Analysis

Association Rules Analysis

Sequence Discovery Analysis

Here, we will discuss each of the data mining's types in detail. Below are several different data mining techniques that can help you find optimal outcomes as the results.

Tasks and Functionalities of Data Mining

Data mining tasks are designed to be semi-automatic or fully automatic and on large data sets to uncover patterns such as groups or clusters, unusual or over the top data called anomaly detection and dependencies such as association and sequential pattern. Once patterns are uncovered, they can be thought of as a summary of the input data, and further analysis may be carried out using Machine Learning and Predictive analytics. For example, the data mining step might help identify multiple groups in the data that a decision support system can use. Note that data collection, preparation, reporting are not part of data mining.

There is a lot of confusion between data mining and data analysis. Data mining functions are used to define the trends or correlations contained in data mining activities. While data analysis is used to test statistical models that fit the dataset, for example, analysis of a marketing campaign, data mining uses Machine Learning and mathematical and statistical models to discover patterns hidden in the data. In comparison, data mining activities can be divided into two categories:

Descriptive Data Mining: It includes certain knowledge to understand what is happening within the data without a previous idea. The common data features are highlighted in the data set. For example, count, average etc.

Predictive Data Mining: It helps developers to provide unlabeled definitions of attributes. With previously available or historical data, data mining can be used to make predictions about critical business metrics based on data's linearity. For example, predicting the volume of business next quarter based on performance in the previous quarters over several years or judging from the findings of a patient's medical examinations that is he suffering from any particular disease.

Functionalities of Data Mining

Data mining functionalities are used to represent the type of patterns that have to be discovered in data mining tasks. Data mining tasks can be classified into two types: descriptive and predictive. Descriptive mining tasks define the common features of the data in the database, and the predictive mining tasks act in inference on the current information to develop predictions.

Data mining is extensively used in many areas or sectors. It is used to predict and characterize data. But the ultimate objective in Data Mining Functionalities is to observe the various trends in data mining. There are several data mining functionalities that the organized and scientific methods offer, such as:

1. Class/Concept Descriptions

A class or concept implies there is a data set or set of features that define the class or a concept. A class can be a category of items on a shop floor, and a concept could be the abstract idea on which data may be categorized like products to be put on clearance sale and non-sale products. There are two concepts here, one that helps with grouping and the other that helps in differentiating.

Data Characterization: This refers to the summary of general characteristics or features of the class, resulting in specific rules that define a target class. A data analysis technique called Attribute-oriented Induction is employed on the data set for achieving characterization.

Data Discrimination: Discrimination is used to separate distinct data sets based on the disparity in attribute values. It compares features of a class with features of one or more contrasting classes.g., bar charts, curves and pie charts.

2. Mining Frequent Patterns

One of the functions of data mining is finding data patterns. Frequent patterns are things that are discovered to be most common in data. Various types of frequency can be found in the dataset.

Frequent item set:This term refers to a group of items that are commonly found together, such as milk and sugar.

Frequent substructure: It refers to the various types of data structures that can be combined with an item set or subsequences, such as trees and graphs.

Frequent Subsequence: A regular pattern series, such as buying a phone followed by a cover.

3. Association Analysis

It analyses the set of items that generally occur together in a transactional dataset. It is also known as Market Basket Analysis for its wide use in retail sales. Two parameters are used for determining the association rules:

It provides which identifies the common item set in the database.

Confidence is the conditional probability that an item occurs when another item occurs in a transaction.

4. Classification

Classification is a data mining technique that categorizes items in a collection based on some predefined properties. It uses methods like if-then, decision trees or neural networks to predict a class or essentially classify a collection of items. A training set containing items whose properties are known is used to train the system to predict the category of items from an unknown collection of items.

5. Prediction

It defines predict some unavailable data values or spending trends. An object can be anticipated based on the attribute values of the object and attribute values of the classes. It can be a prediction of missing numerical values or increase or decrease trends in time-related information. There are primarily two types of predictions in data mining: numeric and class predictions.

Numeric predictions are made by creating a linear regression model that is based on historical data. Prediction of numeric values helps businesses ramp up for a future event that might impact the business positively or negatively.

Class predictions are used to fill in missing class information for products using a training data set where the class for products is known.

6. Cluster Analysis

In image processing, pattern recognition and bioinformatics, clustering is a popular data mining functionality. It is similar to classification, but the classes are not predefined. Data attributes represent the classes. Similar data are grouped together, with the difference being that a class label is not known. Clustering algorithms group data based on similar features and dissimilarities.

7. Outlier Analysis

Outlier analysis is important to understand the quality of data. If there are too many outliers, you cannot trust the data or draw patterns. An outlier analysis determines if there is something out of turn in the data and whether it indicates a situation that a business needs to consider and take measures to mitigate. An outlier analysis of the data that cannot be grouped into any classes by the algorithms is pulled up.

8. Evolution and Deviation Analysis

Evolution Analysis pertains to the study of data sets that change over time. Evolution analysis models are designed to capture evolutionary trends in data helping to characterize, classify, cluster or discriminate time-related data.

9. Correlation Analysis

Correlation is a mathematical technique for determining whether and how strongly two attributes is related to one another. It refers to the various types of data structures, such as trees and graphs, that can be combined with an item set or subsequence. It determines how well two numerically measured continuous variables are linked. Researchers can use this type of analysis to see if there are any possible correlations between variables in their study.

Chapter-2

Association Rule Mining:

Association rule mining is a popular and well researched method for discovering interesting relations between variables in large databases.

It is intended to identify strong rules discovered in databases using different measures of interestingness.

Based on the concept of strong rules, RakeshAgrawal et al. introduced association rules.

Problem Definition:

The problem of association rule mining is defined as:

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items.

Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database.

Each transaction in D has a unique transaction ID and contains a subset of the items in I.

A rule is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$.

The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and

consequent (right-hand-side or RHS) of the rule respectively.

Example:

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is $I = \{\mathrm{milk}, \mathrm{bread}, \mathrm{butter}, \mathrm{beer}\}$ and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table.

An example rule for the supermarket could be $\{\mathrm{butter}, \mathrm{bread}\} \Rightarrow \{\mathrm{milk}\}$ meaning that if butter and bread are bought, customers also buy milk

Example database with 4 items and 5 transactions

Transaction ID	milk	bread	butter	beer
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Important concepts of Association Rule Mining:

The support $\mathrm{supp}(X)$ of an itemset X is defined as the proportion of transactions in the data set which contain the itemset. In the example database, the itemset

$\{\text{milk}, \text{bread}, \text{butter}\}$ has a support of $1/5 = 0.2$ since it occurs in 20% of all transactions (1 out of 5 transactions).

The confidenceof a rule is defined

$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X).$$

For example, the rule $\{\text{butter}, \text{bread}\} \Rightarrow \{\text{milk}\}$ has a confidence of $0.2/0.2 = 1.0$ in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well). Confidence can be interpreted as an estimate of the probability $P(Y|X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

Mining Multilevel Association Rules:

For many applications, it is difficult to find strong associations among data items at lower primitive levels of abstraction due to the sparsity of data at those levels.

Strong associations discovered at high levels of abstraction may represent commonsense knowledge.

Therefore, data mining systems should provide capabilities for mining association ruleset multiple levels of abstraction, with sufficient flexibility for easy traversal among different abstraction spaces.

Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules.

Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework.

In general, a top-down strategy is employed, where counts are accumulated for the calculation of frequent itemsets at each concept level, starting at the concept level 1 and working downward in the hierarchy toward the more specific concept levels, until no more frequent itemsets can be found.

A concept hierarchy defines a sequence of mappings froma set of low-level concepts to higher level, more general concepts. Data can be generalized by replacing low-level concepts within the data by their higher-level concepts, or ancestors, from a concept hierarchy.

The concept hierarchy has five levels, respectively referred to as levels 0to 4, starting with level 0 at the root node for all.

Here, Level 1 includes computer, software, printer camera, and computer accessory. Level 2 includes laptop computer, desktop computer, office software, antivirus software Level 3 includes IBM desktop computer, . . . , Microsoft office software, and so on.

Level 4 is the most specific abstraction level of this hierarchy.

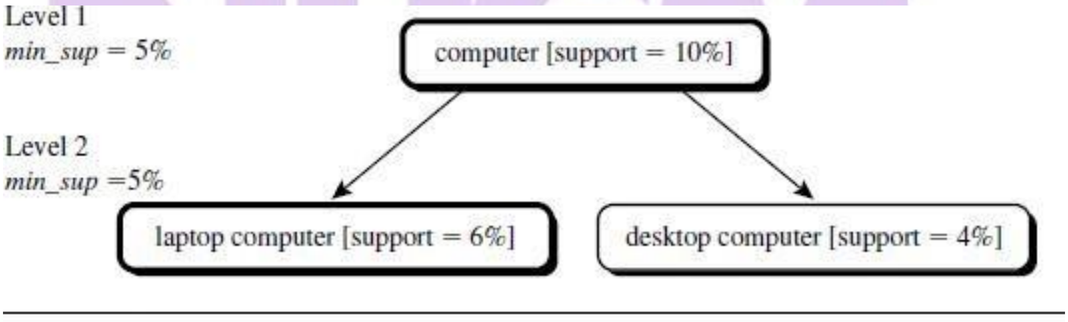
Approaches For Mining Multilevel Association Rules:

Uniform Minimum Support:

The same minimum support threshold is used when mining at each level of abstraction. When a uniform minimum support threshold is used, the search procedure is simplified. The method is also simple in that users are required to specify only one minimum support threshold.

The uniform support approach, however, has some difficulties. It is unlikely that items at lower levels of abstraction will occur as frequently as those at higher levels of abstraction.

If the minimum support threshold is set too high, it could miss some meaningful associations occurring at low abstraction levels. If the threshold is set too low, it may generate many uninteresting associations occurring at high abstraction levels.



Multilevel mining with uniform support.

Mining Quantitative Association Rules:

Quantitative association rules are multidimensional association rules in which the numeric attributes are dynamically discretized during the mining process so as to satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined.

In this section, we focus specifically on how to mine quantitative association rules having two quantitative attributes on the left-hand side of the rule and one categorical attribute on the right-hand side of the rule. That is

Aquan1 ^Aquan2 =>Acat

whereAquan1 and Aquan2 are tests on quantitative attribute interval

Acattests a categorical attribute fromthe task-relevantdata.

Such rules have been referred to as two-dimensional quantitative association rules, because they contain two quantitative dimensions.

For instance, suppose you are curious about the association relationship between pairs of quantitative attributes, like customer age and income, and the type of television (such as high-definition TV, i.e., HDTV) that customers like to buy.

What are Maximal Frequent Itemsets?

A maximal frequent itemset is represented as a frequent itemset for which none of its direct supersets are frequent. The itemsets in the lattice are broken into two groups such as those that are frequent and those that are infrequent. A frequent itemset border, which is defined by a dashed line.

Each item set situated above the border is frequent, while those located under the border (the shaded nodes) are infrequent. Between the itemsets residing near the border, {a, d}, {a, c, e}, and {b, c, d, e} are treated to be maximal frequent itemsets because their direct supersets are infrequent.

An itemset including {a, d} is maximal frequent because some direct supersets, {a, b, d}, {a, c, d}, and {a, d, e}, are infrequent. In contrast, {a, c} is non-maximal because the direct supersets, {a, c, e}, is frequent.

Maximal frequent itemsets adequately support a compact description of frequent itemsets. In other terms, they form the smallest set of itemsets from which some frequent itemsets can be derived. For instance, the frequent itemsets can be broken into two groups such as follows −

Frequent itemsets that start with item a and that can include items c, d, or e. This group contains itemsets including {a), {a, c), {a, d}, {a, e}, and {a, c, e}.

Frequent itemsets that start with items b, c, d, or e. This group contains itemsets including {b}, {b, c}, {c, d}, {b, c, d, e}, etc.

Frequent itemsets that apply in the first group are subsets of either {a, c, e} or {a, d}, while those that apply in the second group are subsets of {b, c, d, e}. Therefore, the maximal frequent itemsets {a, c, e}, {a, d}, and {b, c, d, e} support a compact description of the frequent itemsets. Maximal frequent itemsets support a valuable description for data sets that can make very high, frequent itemsets, as there are exponentially several frequent itemsets in such data. This method is practical only if an effective algorithm occurs to explicitly discover the maximal frequent itemsets without having to enumerate some subsets.

Despite supporting a compact description, maximal frequent itemsets do not include the support data of their subsets. For instance, the support of the maximal frequent itemsets {a,c,e}, {a,d}, and {b,c,d,e} do not give any idea about the support of their subsets.

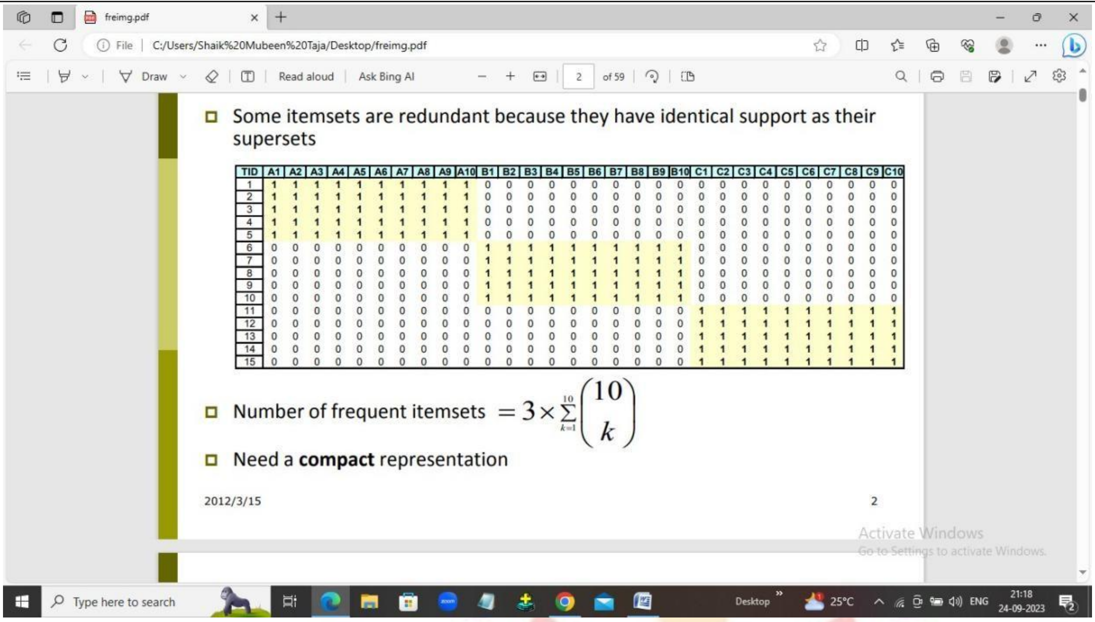
An additional pass over the data set is required to decide the support counts of the non- maximal frequent itemsets. In some cases, it can be desirable to have a minimal description of frequent itemsets that preserves the support data.

Relationship between Frequent Itemset Representations

In conclusion to this section it is important to point out the relationship between frequent itemsets, closed frequent itemsets and maximal frequent itemsets. As mentioned earlier closed and maximal frequent itemsets are subsets of frequent itemsets but maximal frequent itemsets are a more compact representation because it is a subset of closed frequent itemsets. The diagram to the right shows the relationship between these three types of itemsets. Closed frequent itemsets are more widely used than maximal frequent itemset because when efficiency is more important that space, they provide us with the support of the subsets so no additional pass is needed to find this information.

Compact Representation of Frequent Itemsets

Some item sets are redundant because they have identical support as their supersets

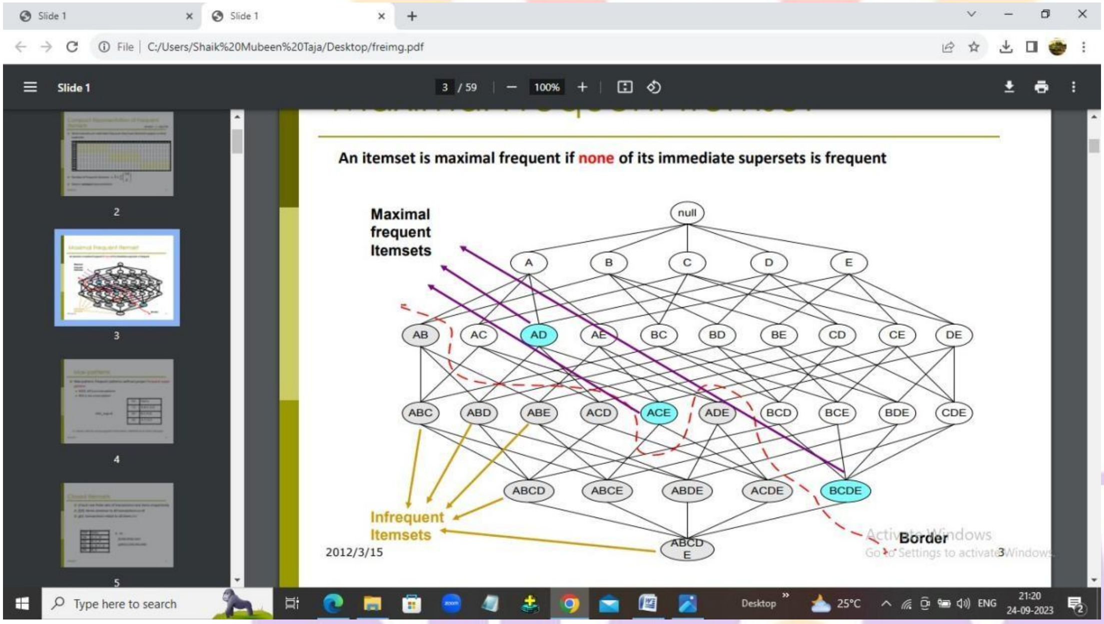


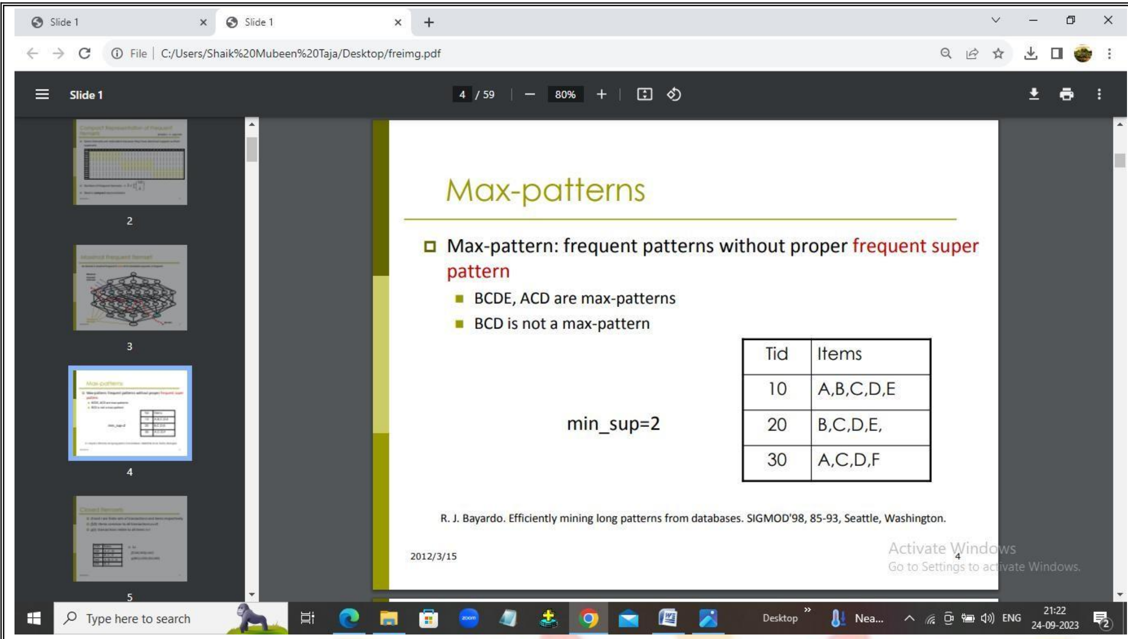
- Some itemsets are redundant because they have identical support as their supersets

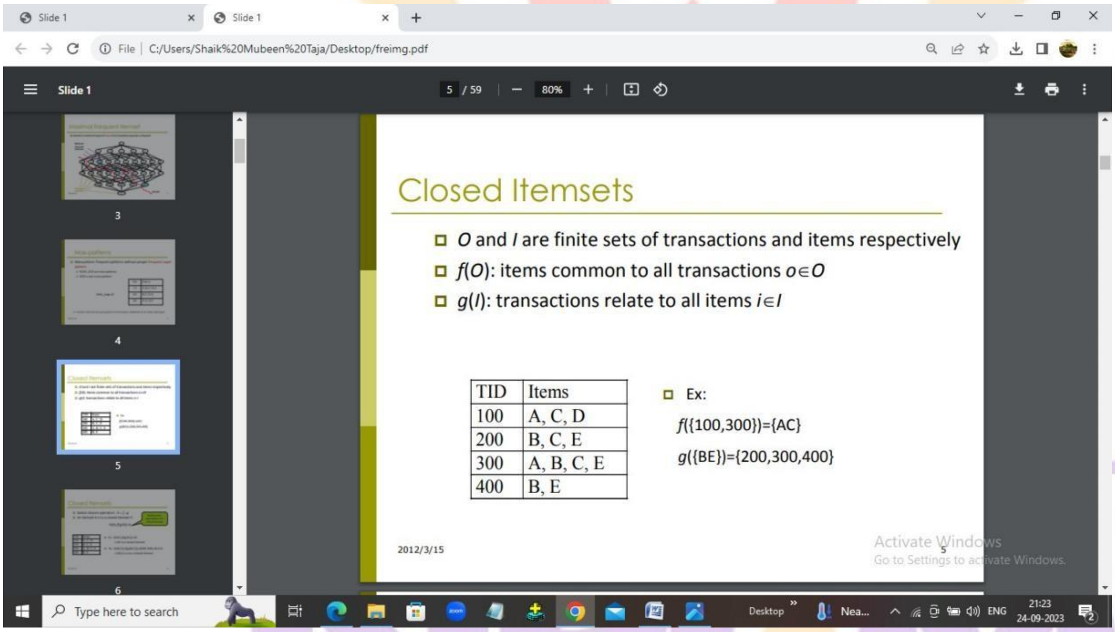
TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

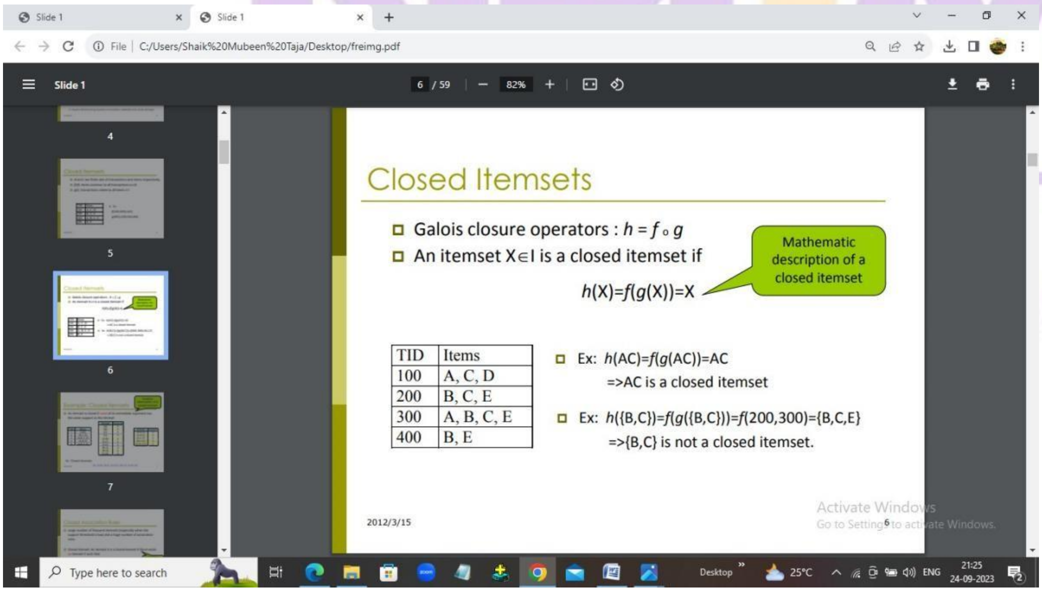
- Number of frequent itemsets $= 3 \times \sum_{k=1}^{10} \binom{10}{k}$
- Need a **compact** representation

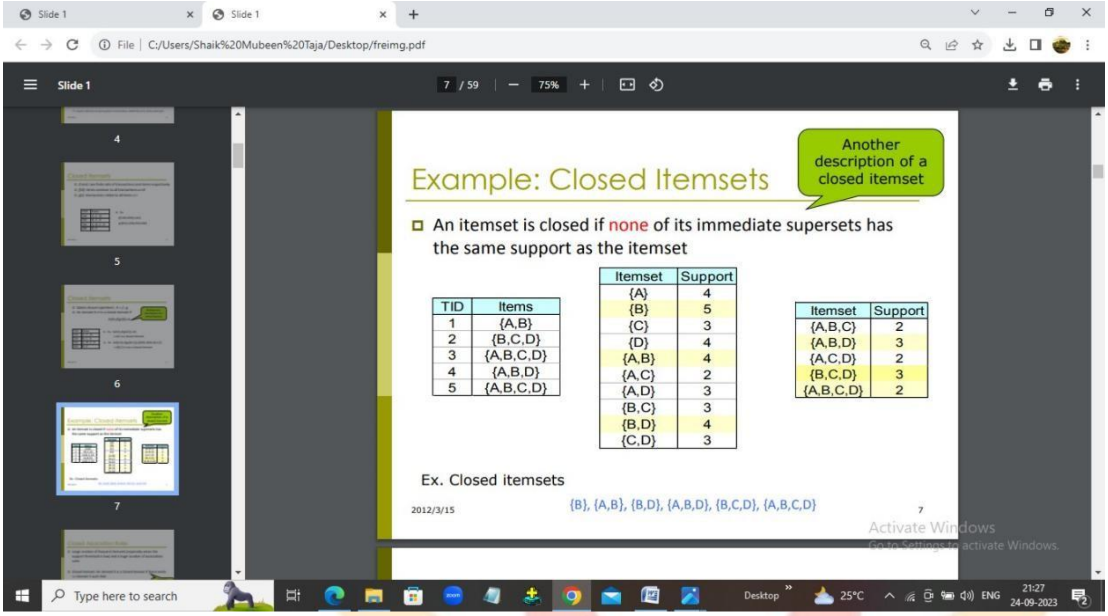
2012/3/15 2

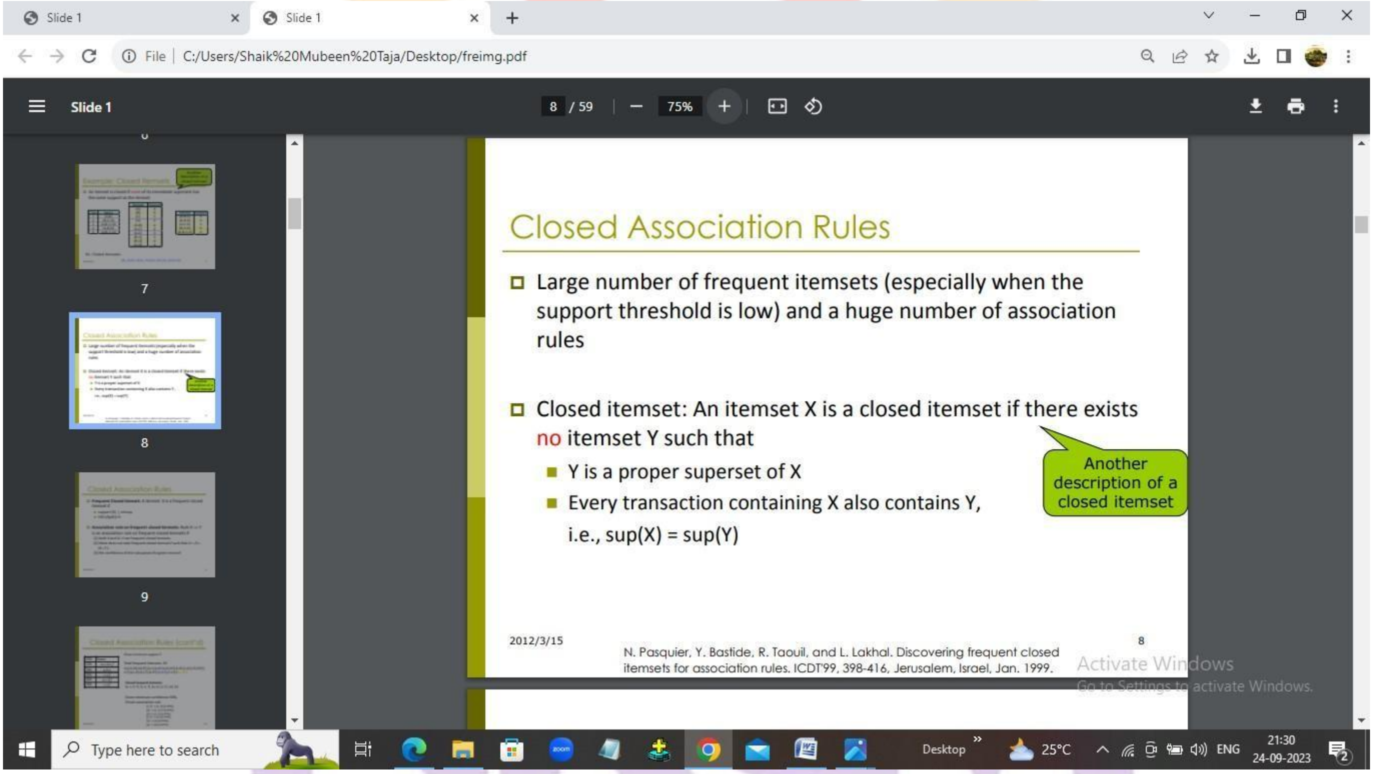


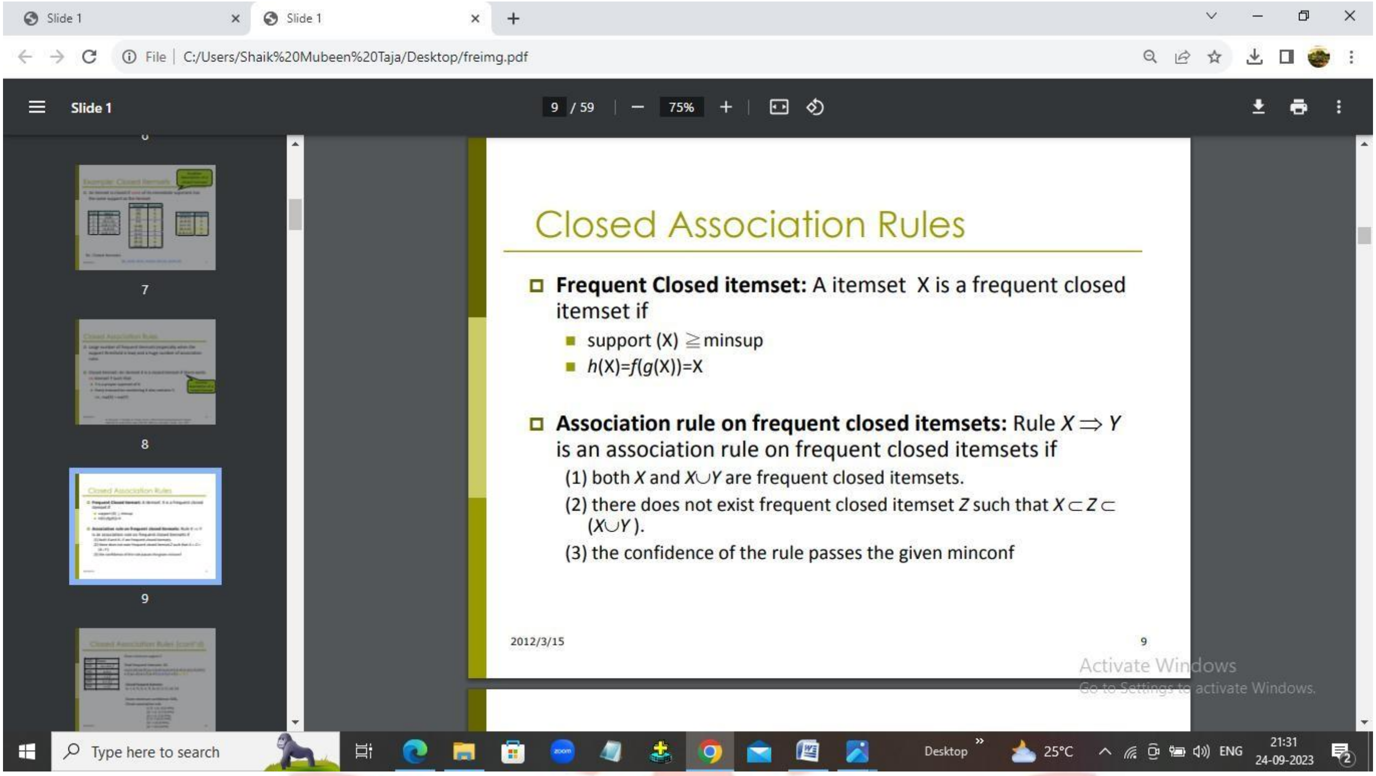

Max-patterns
Max-pattern: frequent patterns without proper frequent super pattern
BCDE, ACD are max-patterns
BCD is not a max-pattern
min_sup=2
Tid | Items
10 | A,B,C,D,E
20 | B,C,D,E,
30 | A,C,D,F
R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98, 85-93, Seattle, Washington.
2012/3/15

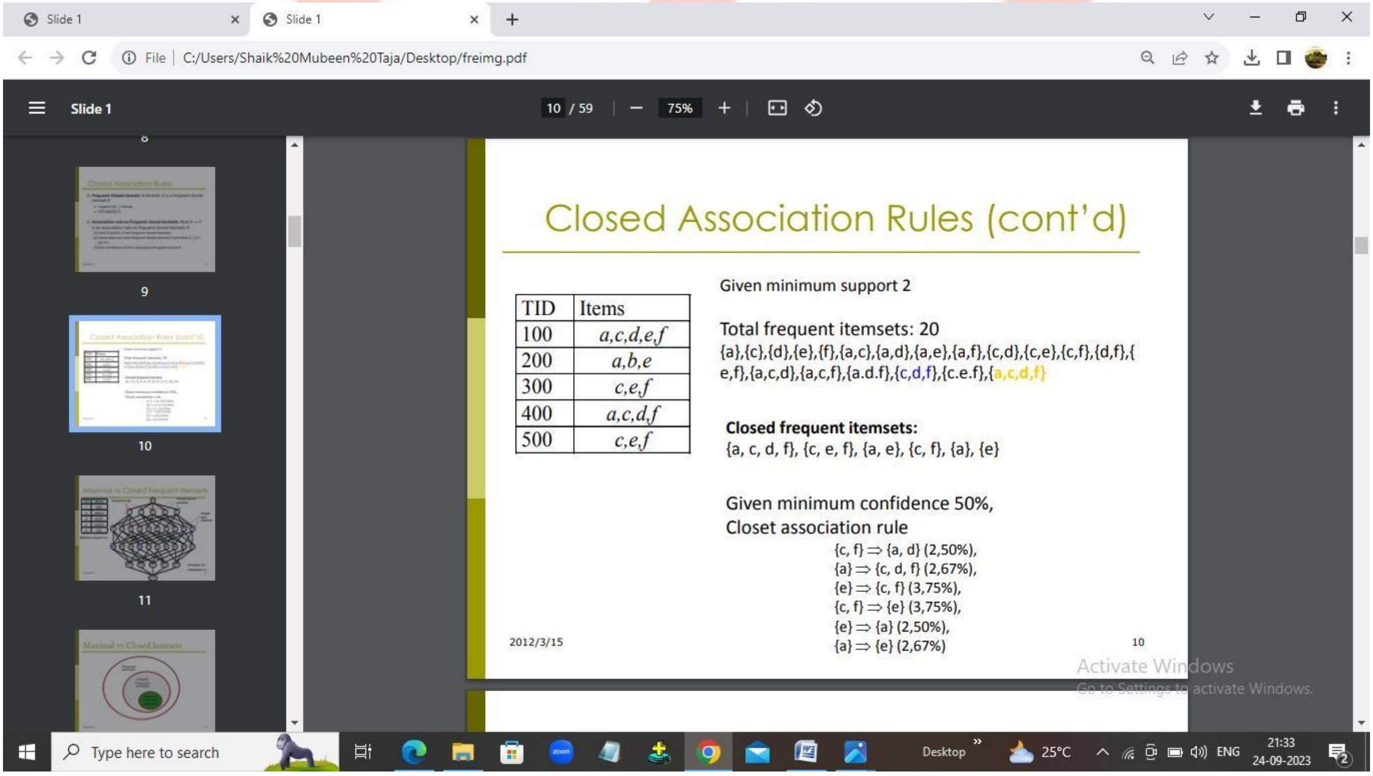

Closed Itemsets
O and I are finite sets of transactions and items respectively
f(O): items common to all transactions o∈O
g(I): transactions relate to all items i∈I
TID | Items
100 | A, C, D
200 | B, C, E
300 | A, B, C, E
400 | B, E
Ex:
f({100,300})={AC}
g({BE})={200,300,400}
2012/3/15

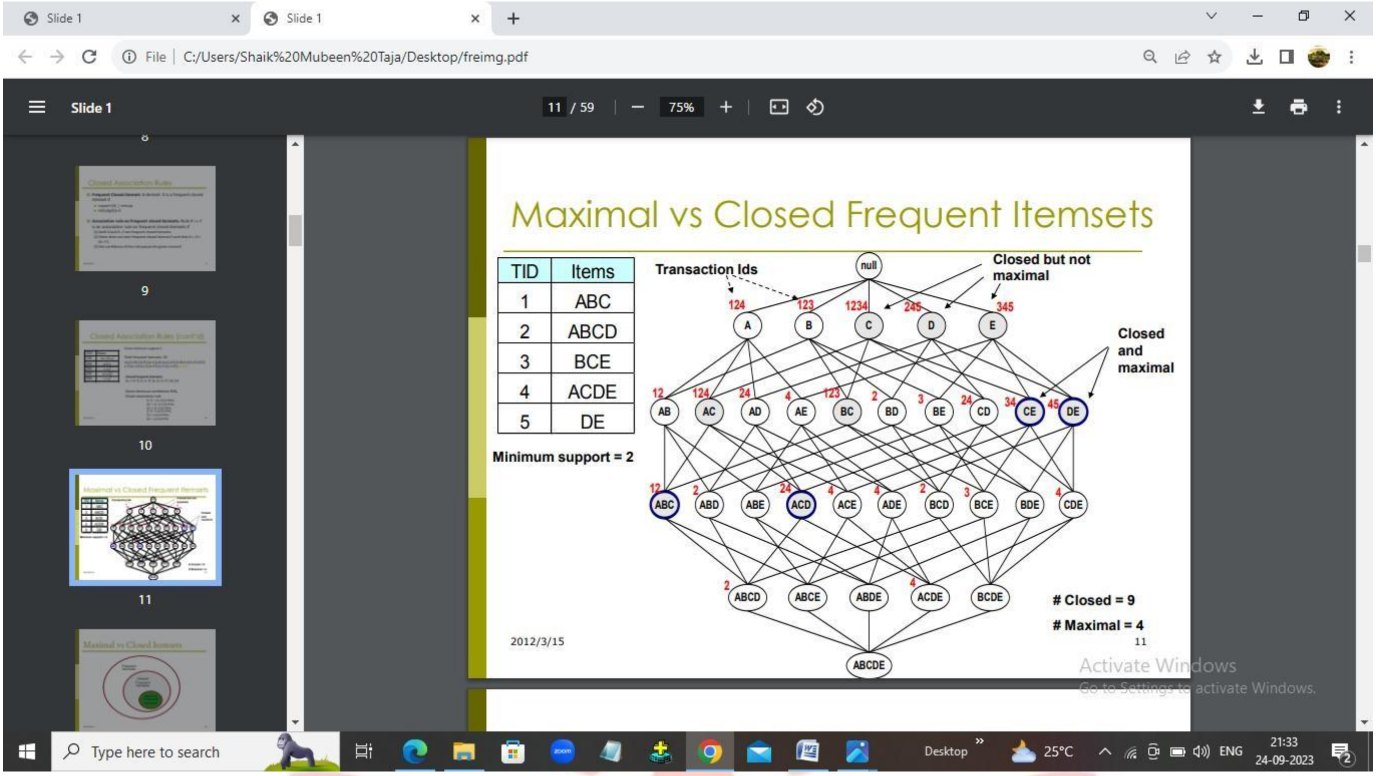

Closed Itemsets
Galois closure operators : h = f ∘ g
An itemset X∈I is a closed itemset if
h(X)=f(g(X))=X
Mathematical description of a closed itemset
TID | Items
100 | A, C, D
200 | B, C, E
300 | A, B, C, E
400 | B, E
Ex: h(AC)=f(g(AC))=AC
=>AC is a closed itemset
Ex: h({B,C})=f(g({B,C}))=f(200,300)={B,C,E}
=>{B,C} is not a closed itemset.
2012/3/15

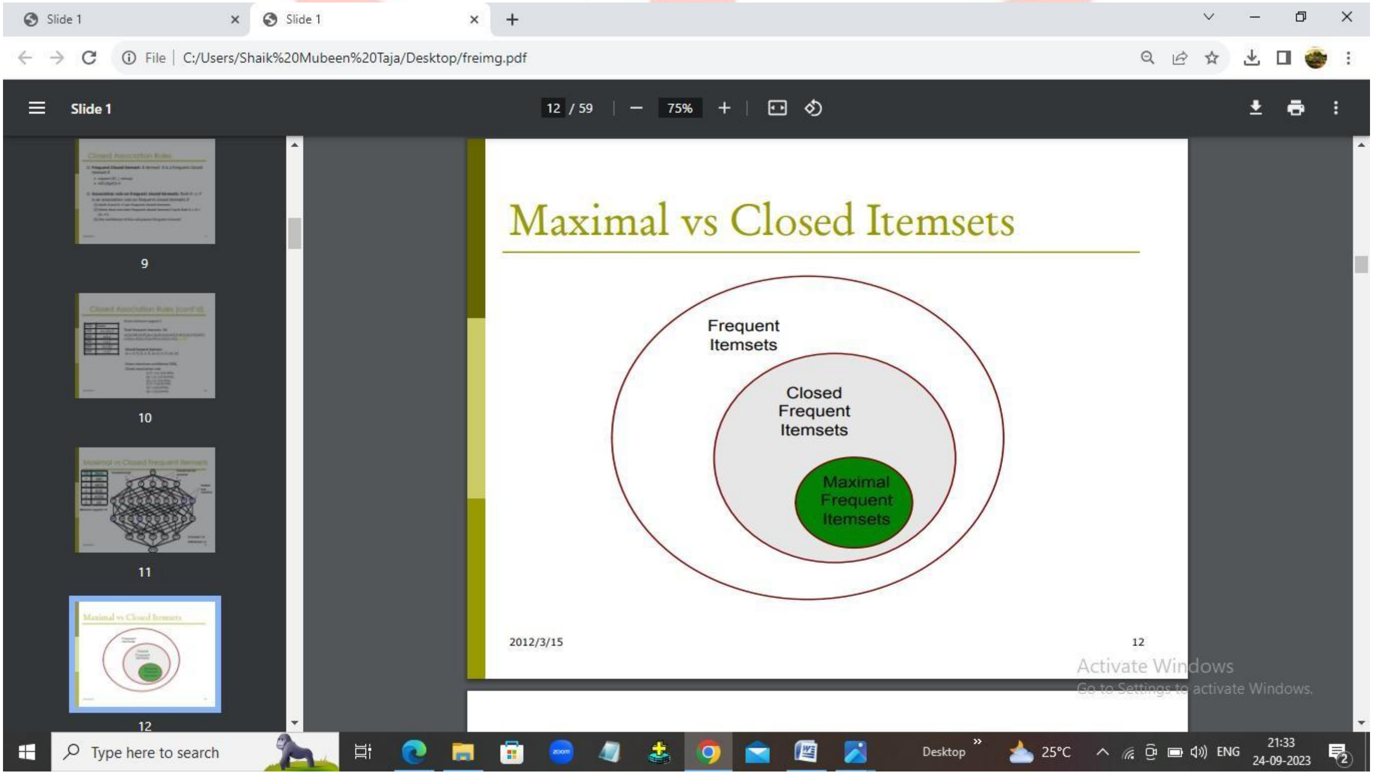

Example: Closed Itemsets
Another description of a closed itemset
An itemset is closed if none of its immediate supersets has the same support as the itemset
TID | Items
1 | {A,B}
2 | {B,C,D}
3 | {A,B,C,D}
4 | {A,B,D}
5 | {A,B,C,D}
Itemset | Support
{A} | 4
{B} | 5
{C} | 3
{D} | 4
{A,B} | 4
{A,C} | 2
{A,D} | 3
{B,C} | 3
{B,D} | 4
{C,D} | 3
Itemset | Support
{A,B,C} | 2
{A,B,D} | 3
{A,C,D} | 2
{B,C,D} | 3
{A,B,C,D} | 2
Ex. Closed itemsets
{B}, {A,B}, {B,D}, {A,B,D}, {B,C,D}, {A,B,C,D}
2012/3/15
7


Closed Association Rules
Large number of frequent itemsets (especially when the support threshold is low) and a huge number of association rules
Closed itemset: An itemset X is a closed itemset if there exists no itemset Y such that
Y is a proper superset of X
Every transaction containing X also contains Y, i.e., sup(X) = sup(Y)
Another description of a closed itemset
2012/3/15
N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99, 398-416, Jerusalem, Israel, Jan. 1999.
8


Closed Association Rules
Frequent Closed itemset: A itemset X is a frequent closed itemset if
support (X) ≧ minsup
h(X)=f(g(X))=X
Association rule on frequent closed itemsets: Rule X ⇒ Y is an association rule on frequent closed itemsets if
(1) both X and X∪Y are frequent closed itemsets.
(2) there does not exist frequent closed itemset Z such that X ⊂ Z ⊂ (X∪Y).
(3) the confidence of the rule passes the given minconf
2012/3/15
9


Closed Association Rules (cont'd)
TID Items
100 a,c,d,e,f
200 a,b,e
300 c,e,f
400 a,c,d,f
500 c,e,f
Given minimum support 2
Total frequent itemsets: 20
{a},{c},{d},{e},{f},{a,c},{a,d},{a,e},{a,f},{c,d},{c,e},{c,f},{d,f},{e,f},{a,c,d},{a,c,f},{a.d.f},{c,d,f},{c.e.f},{a,c,d,f}
Closed frequent itemsets:
{a, c, d, f}, {c, e, f}, {a, e}, {c, f}, {a}, {e}
Given minimum confidence 50%,
Closet association rule
{c, f} ⇒ {a, d} (2,50%),
{a} ⇒ {c, d, f} (2,67%),
{e} ⇒ {c, f} (3,75%),
{c, f} ⇒ {e} (3,75%),
{e} ⇒ {a} (2,50%),
{a} ⇒ {e} (2,67%)
2012/3/15
10


Maximal vs Closed Frequent Itemsets
TID Items
1 ABC
2 ABCD
3 BCE
4 ACDE
5 DE
Minimum support = 2
Transaction Ids
Closed but not maximal
Closed and maximal
null
A B C D E
AB AC AD AE BC BD BE CD CE DE
ABC ABD ABE ACD ACE ADE BCD BCE BDE CDE
ABCD ABCE ABDE ACDE BCDE
ABCDE
Closed = 9
Maximal = 4
2012/3/15
11


Maximal vs Closed Itemsets
Frequent Itemsets
Closed Frequent Itemsets
Maximal Frequent Itemsets
2012/3/15
12

CHAPTER-3

Classification and Prediction:

Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends.

Classification predicts categorical (discrete, unordered) labels, prediction models continuous valued functions.

For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures of potential customers on computer equipment given their income and occupation.

A predictor is constructed that predicts a continuous-valued function, or ordered value, as opposed to a categorical label.

Regression analysis is a statistical methodology that is most often used for numeric prediction.

Many classification and prediction methods have been proposed by researchers in machine learning, pattern recognition, and statistics.

Most algorithms are memory resident, typically assuming a small data size. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data.

Issues Regarding Classification and Prediction:

Preparing the Data for Classification and Prediction:

The following preprocessing steps may be applied to the data to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

Data cleaning:

This refers to the preprocessing of data in order to remove or reduce noise (by applying smoothing techniques) and the treatment of missingvalues (e.g., by replacing a missing value with the most commonly occurring value for that attribute, or with the most probable value based on statistics).

Although most classification algorithms have some mechanisms for handling noisy or missing data, this step can help reduce confusion during learning.

Relevance analysis:

Many of the attributes in the data may be redundant.

Correlation analysis can be used to identify whether any two given attributes are statistically related.

For example, a strong correlation between attributes A1 and A2 would suggest that one of the two could be removed from further analysis.

A database may also contain irrelevant attributes. Attribute subset selection can be used in these cases to find a reduced set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

Hence, relevance analysis, in the form of correlation analysis and attribute subset selection, can be used to detect attributes that do not contribute to the classification or prediction task.

Such analysis can help improve classification efficiency and scalability.

Data Transformation And Reduction

The data may be transformed by normalization, particularly when neural networks or methods involving distance measurements are used in the learning step.

Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1 to +1 or 0 to 1.

The data can also be transformed by generalizing it to higher-level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous valued attributes.

For example, numeric values for the attribute income can be generalized to discrete ranges, such as low, medium, and high. Similarly, categorical attributes, like street, can be generalized to higher-level concepts, like city.

Data can also be reduced by applying many other methods, ranging from wavelet transformation and principle components analysis to discretization techniques, such as binning, histogram analysis, and clustering.

Comparing Classification and Prediction Methods:

Accuracy:

The accuracy of a classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data (i.e., tuples without class label information).

The accuracy of a predictor refers to how well a given predictor can guess the value of the predicted attribute for new or previously unseen data.

Speed:

This refers to the computational costs involved in generating and using the given classifier or predictor.

Robustness:

This is the ability of the classifier or predictor to make correct predictions given noisy data or data with missing values.

Scalability:

This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.

Interpretability:

This refers to the level of understanding and insight that is provided by the classifier or predictor.

Interpretability is subjective and therefore more difficult to assess.

Classification by Decision Tree Induction:

Decision tree induction is the learning of decision trees from class-labeled training tuples.

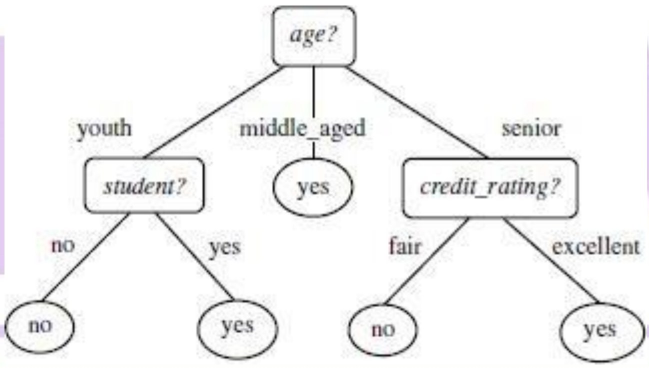
A decision tree is a flowchart-like tree structure, where

Each internal node denotes a test on an attribute.

Each branch represents an outcome of the test.

Each leaf node holds a class label.

The topmost node in a tree is the root node.



The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore I appropriate for exploratory knowledge discovery.

Decision trees can handle high-dimensional data.

Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans.

The learning and classification steps of decision tree induction are simple and fast. In general, decision tree classifiers have good accuracy.

Decision tree induction algorithms have been used for classification in many application areas, such as medicine,manufacturing and production, financial analysis, astronomy, and molecular biology.

Algorithm For Decision Tree Induction:

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C then
(3)      return N as a leaf node labeled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     if Dj is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
(15) return N;
```


The algorithm is called with three parameters:

Data partition

Attribute list

Attribute selection method

The parameter attribute list is a list of attributes describing the tuples.
Attribute selection method specifies a heuristic procedure for selecting the attribute that —best‖ discriminates the given tuples according to class.
The tree starts as a single node, N, representing the training tuples in D.

If the tuples in D are all of the same class, then node N becomes a leaf and is labeled with that class .
All of the terminating conditions are explained at the end of the algorithm. Otherwise, the algorithm calls Attribute selection method to determine the splitting criterion.
The splitting criterion tells us which attribute to test at node N by determining the —best‖ way to separate or partition the tuples in D into individual classes.

There are three possible scenarios. Let A be the splitting attribute. A has v distinct values, {a1, a2, … ,av}, based on the training data.
A is discrete-valued:
In this case, the outcomes of the test at node N correspond directly to the known values of A.
A branch is created for each known value, aj, of A and labeled with that value. Aneed not be considered in any future partitioning of the tuples.
A is continuous-valued:
In this case, the test at node N has two possible outcomes, corresponding to the conditions A <=split point and A >split point, respectively
where split point is the split-point returned by Attribute selection method as part of the splitting criterion.
A is discrete-valued and a binary tree must be produced:
The test at node N is of the form—A€SA?‖ .
SA is the splitting subset for A, returned by Attribute selection method as part of the splitting criterion. It is a subset of the known values of A.

Bayesian Classification:

Bayesian classifiers are statistical classifiers.
They can predictclass membership probabilities, such as the probability that a given tuplebelongs toa particular class.

Bayesian classification is based on Bayes' theorem.

Bayes' Theorem:

Let X be a data tuple. In Bayesian terms, X is considered —evidence.‖ and it is described by measurements made on a set of n attributes.

Let H be some hypothesis, such as that the data tuple X belongs to a specified class C.
For classification problems, we want to determine P(H|X), the probability that the hypothesis H holds given the —evidence‖ or observed data tuple X.
P(H|X) is the posterior probability, or a posteriori probability, of H conditioned on X.
Bayes' theorem is useful in that it providesa way of calculating the posterior probability, P(H|X), from P(H), P(X|H), and P(X).

Naïve Bayesian Classification:

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X = (x1, x2, ...,xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2, ..., An.
Suppose that there are m classes, C1, C2, ..., Cm. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X.
That is, the naïve Bayesian classifier predicts that tuple X belongs to the class Ci if and only if
Thus we maximize P(CijX). The class Ci for which P(CijX) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem
As P(X) is constant for all classes, only P(X|Ci)P(Ci) need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P(C1) = P(C2) = ...= P(Cm), and we would therefore maximize P(X|Ci). Otherwise, we maximize P(X|Ci)P(Ci).
Given data sets with many attributes, it would be extremely computationally expensive to compute P(X|Ci). In order to reduce computation in evaluating P(X|Ci), the naive

assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple. Thus, We can easily estimate the probabilities P(x1|Ci), P(x2|Ci), : : : , P(xn|Ci) from the training tuples. For each attribute, we look at whether the attribute is categorical or continuous-valued. For instance, to compute P(X|Ci), we consider the following:

If Ak is categorical, then P(xk|Ci) is the number of tuples of class Ci in D having the value for Ak, divided by |Ci,D| the number of tuples of class Ci in D.

If Ak is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}.$$

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \leq j \leq m, j \neq i.$$

A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation , defined by In order to predict the class label of X, P(XjCi)P(Ci) is evaluated for each class Ci. The classifier predicts that the class label of tuple X is the class Ciif and only if A Multilayer Feed-Forward Neural Network:

The backpropagation algorithm performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples.

A multilayer feed-forward neural network consists of an input layer, one ormore hiddenlayers, and an output layer.

k-Nearest-Neighbor Classifier:

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuplewith training tuples that are similar to it.

The training tuples are describedby n attributes. Each tuple represents a point in an n-dimensional space. In this way,all of the training tuples are stored in an n-dimensional pattern space. When given anunknown tuple, a k-nearest-neighbor classifier searches the

pattern space for the k trainingtuples that are closest to the unknown tuple. These k training tuples are the k nearest neighbors of the unknown tuple. Closeness is defined in terms of a distance metric, such as Euclidean distance.

The Euclidean distance between two points or tuples, say, X1 = (x11, x12, ... , x1n) and X2 = (x21, x22, ... ,x2n).

In other words, for each numeric attribute, we take the difference between the corresponding values of that attribute in tuple X1and in tuple X2, square this difference and accumulate it. The square root is taken of the total accumulated distance count.

Min-Max normalization can be used to transform a value v of a numeric attribute A to v0 in the range [0, 1] by computing

For k-nearest-neighbor classification, the unknown tuple is assigned the most common class among its k nearest neighbors.

When k = 1, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space.

Nearestneighbor classifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple.

In this case, the classifier returns the average value of the real-valued labels associated with the k nearest neighbors of the unknown tuple.

Other Classification Methods:

Genetic Algorithms:

Genetic algorithms attempt to incorporate ideas of natural evolution. In general, genetic learning starts as follows.

An initial population is created consisting of randomly generated rules. Each rule can be represented by a string of bits. As a simple example, suppose that samples training set are described by two Boolean attributes, A1 and A2, and that there are two classes,C1 andC2. The rule —IF A1 ANDNOT A2 THENC2‖ can be encoded as the bit string —100,‖ where the two leftmost bits represent attributes A1 and A2, respectively, and the rightmost bit represents the class.

Similarly, the rule —IF NOT A1 AND NOT A2 THEN C1‖ can be encoded as —001.‖

If an attribute has k values, where k > 2, then k bits may be used to encode the attribute’s values.

Classes can be encoded in a similar fashion.

Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules in the current population, as well as offspring of these rules.

Typically, the fitness of a rule is assessed by its classification accuracy on a set of training samples.
Offspring are created by applying genetic operators such as crossover and mutation. In crossover, substrings from pairs of rules are swapped to form new pairs Inmutation, randomly selected bits in a rule's string are inverted.
The process of generating new populations based on prior populations of rules continues of rules until a population, P, evolves where each rule in P satisfies a pre specified fitness threshold.
Genetic algorithms are easily parallelizable and have been used for classification as well as other optimization problems. In data mining, they may be used to evaluate the fitness of other algorithms.

Fuzzy Set Approaches:
Fuzzy logic uses truth values between 0.0 and 1.0 to represent the degree of membership that a certain value has in a given category. Each category then represents afuzzy set.
Fuzzy logic systems typically provide graphical tools to assist users in converting attribute values to fuzzy truth values.
Fuzzy set theory is also known as possibility theory.

It was proposed by LotfiZadeh in1965 as an alternative to traditional two-value logic and probability theory.
It lets us work at a high level of abstraction and offers a means for dealing with imprecise measurement of data.
Most important, fuzzy set theory allows us to deal with vague or inexact facts.
Unlike the notion of traditional —crisp‖ sets where anelement either belongs to a set S or its complement, in fuzzy set theory, elements canbelong to more than one fuzzy set.
Fuzzy set theory is useful for data mining systems performing rule-based classification. It provides operations for combining fuzzy measurements.
Several procedures exist for translating the resulting fuzzy output into a defuzzified or crisp value that is returned by the system.
Fuzzy logic systems have been used in numerous areas for classification, including market research, finance, health care, and environmental engineering.

Regression Analysis:
Regression analysis can be used to model the relationship between one or more

independent or predictor variables and a dependent or response variable which is continuous-valued.

In the context of data mining, the predictor variables are theattributes of interest describing the tuple (i.e., making up the attribute vector).

In general,the values of the predictor variables are known.

The response variable is what we want to predict.

Linear Regression:

Straight-line regression analysis involves a response variable, y, and a single predictorvariable x.

It is the simplest form of regression, and models y as a linear function of x.

That is,$y = b+wx$

where the variance of y is assumed to be constant

$b$and w are regression coefficientsspecifying the Y-intercept and slope of the line.

The regression coefficients, w and b, can also be thought of as weights, so that we can equivalently write, $y = w_0+w_1x$

These coefficients can be solved for by the method of least squares, which estimates the best-fitting straight line as the one that minimizes the error between the actual data and the estimate of the line.

Let D be a training set consisting of values of predictor variable, x, for some population and their associated values for response variable, y. The training set contains $|D|$ data points of the form$(x_1, y_1), (x_2, y_2), \ldots , (x_{|D|}, y_{|D|})$.

The regressioncoefficients can be estimated using this method with the following equations:

where x is the mean value of $x_1, x_2, \ldots , x_{|D|}$, and y is the mean value of $y_1, y_2,\ldots, y_{|D|}$. The coefficients w_0 and w_1 often provide good approximations to otherwise complicated regression equations.

Multiple Linear Regression:

It is an extension of straight-line regression so as to involve more than one predictorvariable.

It allows response variable y to be modeled as a linear function of, say, n predictor variables or attributes, A1, A2, ..., An, describing a tuple, X.

An example of a multiple linear regression model basedon two predictor attributes orvariables, A1 and A2, isy = w0+w1x1+w2x2

where x1 and x2 are the values of attributes A1 and A2, respectively, in X.

Multiple regression problemsare instead commonly solved with the use of statistical software packages, such as SAS,SPSS, and S-Plus.

Nonlinear Regression:

It can be modeled by adding polynomial terms to the basic linear model.

By applying transformations to the variables, we can convert the nonlinear model into a linear one that can then be solved by the method of least squares.

Polynomial Regression is a special case of multiple regression. That is, the addition of high- order terms like x2, x3, and so on, which are simple functions of the single variable, x, can be considered equivalent to adding new independent variables.

Transformation of a polynomial regression model to a linear regression model:

Consider a cubic polynomial relationship given by y = w0+w1x+w2x2+w3x3

To convert this equation to linear form, we define new variables:

x1 = x, x2 = x2 ,x3 = x3

It can then be converted to linear formby applying the above assignments,resulting in the equationy = w0+w1x+w2x2+w3x3

which is easily solved by themethod of least squares using software for regression analysis.

Classifier Accuracy:

The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.

In the pattern recognition literature, this is also referred to as the overall recognition rate of theclassifier, that is, it reflects how well the classifier recognizes tuples of the various classes.

The error rate or misclassification rate of a classifier,M, which is simply 1-Acc(M), where Acc(M) is the accuracy of M.

The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of different classes.

True positives refer to the positive tuples that were correctly labeled by the classifier. True negatives are the negative tuples that were correctly labeled by the classifier.
False positives are the negative tuples that were incorrectly labeled.
How well the classifier can recognize, for this sensitivity and specificity measures can be used.
Accuracy is a function of sensitivity and specificity.

Chapter-4

Cluster Analysis:
The process of grouping a set of physical or abstract objects into classes of similar objectsis called clustering.
A cluster is a collection of data objects that are similar to one another within the samecluster and are dissimilar to the objects in other clusters.
A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.
Cluster analysis tools based on k-means, k-medoids, and several methods have also been built into many statisticalanalysis software packages or systems, such as S-Plus, SPSS, and SAS.

Applications:
Cluster analysis has been widely used in numerous applications, including market research, pattern recognition, data analysis, and image processing.
In business, clustering can help marketers discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns.
In biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionality, and gain insight into structures inherent in populations.
Clustering may also help in the identification of areas of similar land use in an earth observation database and in the identification of groups of houses in a city according to house type, value, and geographic location, as well as the identification of groups of automobile insurance policy holders with a high average claim cost.
Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity. Clustering can also be

used for outlier detection, Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce.

Typical Requirements Of Clustering In Data Mining:

Scalability:

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

Highly scalable clustering algorithms are needed.

Ability to deal with different types of attributes:

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

Discovery of clusters with arbitrary shape:

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

However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape.

Minimal requirements for domain knowledge to determine input parameters:

Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often difficult to determine, especially for data sets containing high-dimensional objects. This not only burdens users, but it also makes the quality of clustering difficult to control.

Ability to deal with noisy data:

Most real-world databases contain outliers or missing, unknown, or erroneous data.

Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

Incremental clustering and insensitivity to the order of input records:

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

That is, given a set of data objects, such an algorithm may return dramatically different clustering depending on the order of presentation of the input objects.

It is important to develop incremental clustering algorithms and algorithms that are insensitive to the order of input.

High dimensionality:

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

Constraint-based clustering:

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

Interpretability and usability:

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

Major Clustering Methods:

Partitioning Methods

Hierarchical Methods

Density-Based Methods

Grid-Based Methods

Model-Based Methods

Partitioning Methods:

A partitioning method constructs k partitions of the data, where each partition represents a cluster and k <= n. That is, it classifies the data into k groups, which together satisfy the following requirements:

Each group must contain at least one object, and Each object must belong to exactly one group.

A partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another.

The general criterion of a good partitioning is that objects in the same cluster are close or related to each other, whereas objects of different clusters are far apart or very different.

Hierarchical Methods:

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.

The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one or until a termination condition holds.

The divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.

Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. This rigidity is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices.

There are two approaches to improving the quality of hierarchical clustering:

Perform careful analysis of object ―linkages‖ at each hierarchical partitioning, such as in Chameleon, or

Integrate hierarchical agglomeration and other approaches by first using a hierarchical agglomerative algorithm to group objects into microclusters, and then performing macroclustering on the microclusters using another clustering method such as iterative relocation.

Density-based methods:

Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes.

Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers)and discover clusters of arbitrary shape.

DBSCAN and its extension, OPTICS, are typical density-based methods that grow clusters according to a density-based connectivity analysis. DENCLUE is a method that clusters objects based on the analysis of the value distributions of density functions.

Grid-Based Methods:

Grid-based methods quantize the object space into a finite number of cells that form a grid structure.

All of the clustering operations are performed on the grid structure i.e., on the quantized space. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.

STING is a typical example of a grid-based method. Wave Cluster applies wavelet transformation for clustering analysis and is both grid-based and density-based.

Model-Based Methods:

Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model.

A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.

It also leads to a way of automatically determining the number of clusters based on standard statistics, taking ―noise‖ or outliers into account and thus yielding robust clustering methods.

Tasks in Data Mining:

Clustering High-Dimensional Data

Constraint-Based Clustering

Clustering High-Dimensional Data:

It is a particularly important task in cluster analysis because many applications require the analysis of objects containing a large number of features or dimensions.

For example, text documents may contain thousands of terms or keywords as features, and DNA micro array data may provide information on the expression levels of thousands of genes under hundreds of conditions.

Clustering high-dimensional data is challenging due to the curse of dimensionality. Many dimensions may not be relevant. As the number of dimensions increases, the data become increasingly sparse so that the distance measurement between pairs of points become meaningless and the average density of points anywhere in the data is likely to be low. Therefore, a different clustering methodology needs to be developed for high-dimensional data.

CLIQUE and PROCLUS are two influential subspace clustering methods, which search for clusters in subspaces ofthe data, rather than over the entire data space.

Frequent pattern–based clustering, another clustering methodology, extracts distinct frequent patterns among subsets of dimensions that occur frequently. It uses such patterns to group objects and generate meaningful clusters.

Constraint-Based Clustering:

It is a clustering approach that performs clustering by incorporation of user-specified or application-oriented constraints.

A constraint expresses a user's expectation or describes properties of the desired clustering results, and provides an effective means for communicating with the clustering process.

Various kinds of constraints can be specified, either by a user or as per application requirements.

Spatial clustering employs with the existence of obstacles and clustering under user- specified constraints. In addition, semi-supervised clustering employs for pairwise constraints in order to improve the quality of the resulting clustering.

Classical Partitioning Methods:

The most well-known and commonly used partitioning methods are

The k-Means Method

k-Medoids Method

Centroid-Based Technique: The K-Means Method:

The k-means algorithm takes the input parameter, k, and partitions a set of n objects in to k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low.

Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity.

The k-means algorithm proceeds as follows.

First, it randomly selects k of the objects, each of which initially represents a clustermean or center.

For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.

It then computes the new mean for each cluster.

This process iterates until the criterion function converges.

Typically, the square-error criterion is used, defined as where E is the sum of the square error for all objects in the data set pis the point in space representing a given object is the mean of cluster Ci The k-means partitioning algorithm: The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Clustering of a set of objects based on the k-means method

The k-Medoids Method:

The k-means algorithm is sensitive to outliers because an object with an extremely large value may substantially distort the distribution of data. This effect is particularly exacerbated due to the use of the square-error function.

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

The initial representative objects are chosen arbitrarily. The iterative process of replacing representative objects by non representative objects continues as long as the quality of the resulting clustering is improved. This quality is estimated using a cost function that measures the average dissimilarity between an object and the representative object of its cluster.

To determine whether a non representative object, oj random, is a good replacement for a current representative object, oj, the following four cases are examined for each of the nonrepresentative objects.

Case 1:

pcurrently belongs to representative object, oj. If ojis replaced by or and representative object and p is closest to one of the other representative objects, oi,i≠j, then p is reassigned to oi.

Case 2:

pcurrently belongs to representative object, oj. If ojis replaced by orandomasa representative object and p is closest to orandom, then p is reassigned to orandom.

Case 3:

pcurrently belongs to representative object, oi, i≠j. If ojis replaced by orandomas a representative object and p is still closest to oi, then the assignment does notchange.

Case 4:

pcurrently belongs to representative object, oi, i≠j. If ojis replaced byorandomas a representative object and p is closest to orandom, then p is reassigned

toorandom.

Four cases of the cost function for k-medoids clustering

Thek-MedoidsAlgorithm:

The k-medoids algorithm for partitioning based on medoid or central objects.

The k-medoids method ismore robust than k-means in the presence of noise and outliers, because a medoid is lessinfluenced by outliers or other extreme values than a mean. However, its processing ismore costly than the k-means method.

Hierarchical Clustering Methods:

A hierarchical clustering method works by grouping data objects into a tree of clusters.

The quality of a pure hierarchical clusteringmethod suffers fromits inability to performadjustment once amerge or split decision hasbeen executed. That is, if a particular merge or split decision later turns out to have been apoor choice, the method cannot backtrack and correct it.

Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion.

Agglomerative hierarchical clustering:

This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.

Most hierarchical clustering methods belong to this category. They differ only in their definition of intercluster similarity.

Divisive hierarchical clustering:

This top-down strategy does the reverse of agglomerativehierarchical clustering by starting with all objects in one cluster.

It subdividesthe cluster into smaller and smaller pieces, until each object forms a cluster on itsown or until it satisfies certain termination conditions, such as a desired number ofclusters is obtained or the diameter of each cluster is within a certain threshold.

Constraint-Based Cluster Analysis:

Constraint-based clustering finds clusters that satisfy user-specified preferences orconstraints. Depending on the nature of the constraints, constraint-based clusteringmay adopt rather different approaches.

There are a few categories of constraints.

Constraints on individual objects:

We can specify constraints on the objects to beclustered. In a real estate application, for example, one may like to spatially cluster only those luxury mansions worth over a million dollars. This constraint confines the setof objects to be clustered. It can easily be handled by preprocessing after which the problem reduces to an instance ofunconstrained clustering.

Constraints on the selection of clustering parameters:

A user may like to set a desired range for each clustering parameter. Clustering parameters are usually quite specific to the given clustering algorithm. Examples of parameters include k, the desired numberof clusters in a k-means algorithm; or e the radius and the minimum number of points in the DBSCAN algorithm. Although such user-specified parameters may strongly influence the clustering results, they are usually confined to the algorithm itself. Thus, their fine tuning and processing are usually not considered a form of constraint- based clustering.

Constraints on distance or similarity functions:

We can specify different distance orsimilarity functions for specific attributes of the objects to be clustered, or differentdistance measures for specific pairs of objects.When clustering sportsmen, for example,we may use different weighting schemes for height, body weight, age, and skilllevel. Although this will likely change the mining results, it may not alter the clusteringprocess per se. However, in some cases, such changes may make the evaluationof the distance function nontrivial, especially when it is tightly intertwined with the clustering process.

User-specified constraints on the properties of individual clusters:

A user may like tospecify desired characteristics of the resulting clusters, which may strongly influencethe clustering process.

Semi-supervised clustering based on partial supervision:

The quality of unsupervisedclustering can be significantly improved using some weak form of supervision.This may be in the formof pairwise constraints (i.e., pairs of objects labeled as belongingto the same or different cluster). Such a constrained clustering process is calledsemi-supervised clustering.

Outlier Analysis:

There exist data objects that do not comply with the general behavior or model of the data. Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers.

Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. This, however, could result in the loss of important hidden information because one person’s noise could be another person’s signal. In other words, the outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining.

It can be used in fraud detection, for example, by detecting unusual usage of credit cards or telecommunication services. In addition, it is useful in customized marketing for identifying the spending behavior of customers with extremely low or extremely high incomes, or in medicalanalysis for finding unusual responses to various medical treatments.

Outlier mining can be described as follows: Given a set of n data points or objectsand k, the expected number of outliers, find the top k objects that are considerablydissimilar, exceptional, or inconsistent with respect to the remaining data. The outliermining problem can be viewed as two subproblems:

Define what data can be considered as inconsistent in a given data set, andFind an efficient method to mine the outliers so defined.

Types of outlier detection:

Statistical Distribution-Based Outlier Detection

Distance-Based Outlier Detection

Density-Based Local Outlier Detection

Deviation-Based Outlier Detection

Statistical Distribution-Based Outlier Detection:

The statistical distribution-based approach to outlier detection assumes a distributionor probability model for the given data set (e.g., a normal or Poisson distribution) andthen identifies outliers with respect to the model using a discordancy test. Application ofthe test requires knowledge of the data set parameters knowledge of distribution parameters such as the mean and variance and theexpected number of outliers.

A statistical discordancy test examines two hypotheses: A working hypothesis

An alternative hypothesis

A working hypothesis, H, is a statement that the entire data set of n objects comes from an initial distribution model, F, that is, The hypothesis is retained if there is no statistically significant evidence supporting its rejection. A discordancy test verifies whether an object, oi, is significantly large (or small) in relation to the distribution F. Different test statistics have been proposed for use as a discordancy test, depending on the available knowledge of the data. Assuming that some statistic, T, has been chosen for discordancy testing, and the value of the statistic for object oi is vi, then the distribution of T is constructed. Significance probability, SP(vi)=Prob(T > vi), is evaluated. If SP(vi) is sufficiently small, then oi is discordant and the working hypothesis is rejected.

An alternative hypothesis, H, which states that oi comes from another distribution model, G, is adopted. The result is very much dependent on which model F is chosen because oimay be an outlier under one model and a perfectly valid value under another. The

alternative distribution is very important in determining the power of the test, that is, the probability that the working hypothesis is rejected when oi is really an outlier.

There are different kinds of alternative distributions.

Inherent alternative distribution:

In this case, the working hypothesis that all of the objects come from distribution F is rejected in favor of the alternative hypothesis that all of the objects arise from another distribution, G:

H :oi € G, where i = 1, 2,..., n

F and G may be different distributions or differ only in parameters of the same distribution. There are constraints on the form of the G distribution in that it must have potential to produce outliers. For example, it may have a different mean or dispersion, or a longer tail.

Mixture alternative distribution:

The mixture alternative states that discordant values are not outliers in the F population, but contaminants from some other population,

G. In this case, the alternative hypothesis is

Slippage alternative distribution:

This alternative states that all of the objects (apart from some prescribed small number) arise independently from the initial model, F, with its given parameters, whereas the remaining objects are independent observations from a modified version of F in which the parameters have been shifted.

There are two basic types of procedures for detecting outliers:

Block procedures:

In this case, either all of the suspect objects are treated as outliersor all of them are accepted as consistent.

Consecutive procedures:

An example of such a procedure is the insideoutprocedure. Its main idea is that the object that is least likely to be an outlier istested first. If it is found to be an outlier, then all of the

more extreme values are alsoconsidered outliers; otherwise, the next most extreme object is tested, and so on. Thisprocedure tends to be more effective than block procedures.

Distance-Based Outlier Detection:

The notion of distance-based outliers was introduced to counter the main limitationsimposed by statistical methods. An object, o, in a data set, D, is a distance-based (DB)outlier with parameters pct and dmin,that is, a DB(pct;dmin)-outlier, if at least a fraction,pct, of the objects in D lie at a distance greater than dmin from o. In other words, rather thatrelying on statistical tests, we can think of distance-based outliers as thoseobjects that do not have enoughneighbors, where neighbors are defined based ondistance from the given object. In comparison with statistical-based methods, distancebased outlier detection generalizes the ideas behind discordancy testing for various standarddistributions. Distance-based outlier detection avoids the excessive computationthat can be associated with fitting the observed distribution into some standard distributionand in selecting discordancy tests. For many discordancy tests, it can be shown that if an object, o, is an outlier accordingto the given test, then o is also a DB(pct, dmin)-outlier for some suitably defined pct anddmin.

For example, if objects that lie three or more standard deviations from the mean are considered to be outliers, assuming a normal distribution, then this definition can be generalized by a DB(0.9988, 0.13s) outlier.

Several efficient algorithms for mining distance-based outliers have been developed.

Index-based algorithm:

Given a data set, the index-based algorithm uses multidimensionalindexing structures, such as R-trees or k-d trees, to search for neighbors of eachobject o within radius dminaround that object. Let Mbe the maximum number ofobjects within the dmin- neighborhood of an outlier. Therefore, onceM+1 neighborsof object o are found, it is clear that o is not an outlier. This algorithm has a worst-casecomplexity of O(n2k), where n is the number of objects in the data set and k is thedimensionality. The index-based algorithm scales well as k increases. However, thiscomplexity evaluation takes only the search time into account, even though the taskof building an index in itself can be computationally intensive.

Nested-loop algorithm:

The nested-loop algorithm has the same computational complexityas the index-based algorithm but avoids index structure construction and triesto minimize the number of I/Os. It divides the memory buffer space into two halvesand the data set into several logical blocks. By carefully choosing the order in whichblocks are loaded into each half, I/O efficiency can be achieved.

Cell-based algorithm:

To avoidO(n2) computational complexity, a cell-based algorithm was developed for memory-resident data sets. Its complexity is O(ck+n), where c is a constant depending on the number of cells and k is the dimensionality.

In this method, the data space is partitioned into cells with a side length equal to Eachcell has two layers surrounding it. The first layer is one cell thick, while the secondis

cells thick, rounded up to the closest integer. The algorithm countsoutliers on a cell-by-cell rather than an object-by-object basis. For a given cell, itaccumulates three counts—the number of objects in the cell, in the cell and the firstlayer together, and in the cell and both layers together. Let's refer to these counts ascell count, cell + 1 layer count, and cell + 2 layers count, respectively.

Let Mbe the maximum number ofoutliers that can exist in the dmin-neighborhood of an outlier.

An object, o, in the current cell is considered an outlier only if cell + 1 layer countis less than or equal to M. If this condition does not hold, then all of the objectsin the cell can be removed from further investigation as they cannot be outliers.

If cell_+ 2_layers_count is less than or equal to M, then all of the objects in thecell are considered outliers. Otherwise, if this number is more than M, then itis possible that some of the objects in the cell may be outliers. To detect theseoutliers, object-by-object processing is used where, for each object, o, in the cell,objects in the second layer of o are examined. For objects in the cell, only thoseobjects having no more than M points in their dmin-neighborhoods are outliers.The dmin-neighborhood of an object consists of the object's cell, all of its firstlayer, and some of its second layer.

A variation to the algorithm is linear with respect to n and guarantees that no morethan three passes over the data set are required. It can be used for large disk-residentdata sets, yet does not scale well for high dimensions.

Density-Based Local Outlier Detection:

Statistical and distance-based outlier detection both depend on the overall or globaldistribution of the given set of data points, D. However, data are usually not uniformlydistributed. These methods encounter difficulties when analyzing data with rather different
density distributions.

To define the local outlier factor of an object, we need to introduce the concepts ofk-distance, k-distance neighborhood, reachability distance,13 and local reachability density. These are defined as follows:

The k-distance of an object p is the maximal distance that p gets from its k- nearestneighbors. This distance is denoted as k-distance(p). It is defined as the distance, d(p, o), between p and an object o 2 D, such that for at least k objects, o0 2 D, it holds that d(p, o')_d(p, o). That is, there are at least k objects inDthat are as close asor closer to p than o, and for at most k-1 objects, o00 2 D, it holds that d(p;o'') <d(p, o).

That is, there are at most k-1 objects that are closer to p than o. You may bewondering at this point how k is determined. The LOF method links to density-basedclustering in that it sets k to the parameter rMinPts,which specifies the minimumnumberof points for use in identifying clusters based on density.

Here, MinPts (as k) is used to define the local neighborhood of an object, p.

The k-distance neighborhood of an object p is denoted Nkdistance(p)(p), or Nk(p)for short. By setting k to MinPts, we get NMinPts(p). It contains the MinPts-nearestneighbors of p. That is, it contains every object whose distance is not greater than theMinPts-distance of p. The reachability distance of an object p with respect to object o (where o is within theMinPts-nearest neighbors of p), is defined as reach
distMinPts(p, o) = max{MinPtsdistance(o), d(p, o)}.

Intuitively, if an object p is far away , then the reachabilitydistance between the two is simply their actual distance. However, if they are sufficientlyclose (i.e., where p is within the MinPts-distance neighborhood of o), thenthe actual distance is replaced by the MinPts-

distance of o. This helps to significantlyreduce the statistical fluctuations of d(p, o) for all of the p close to o.
The higher thevalue of MinPts is, the more similar is the reachability distance for objects withinthe same neighborhood.
Intuitively, the local reachability density of p is the inverse of the average reachability density based on the MinPts-nearest neighbors of p. It is defined as
The local outlier factor (LOF) of p captures the degree to which we call p an outlier. It is defined as

It is the average of the ratio of the local reachability density of p and those of p's MinPts- nearest neighbors. It is easy to see that the lower p's local reachability density is, and the higher the local reachability density of p's MinPts-nearest neighbors are, the higher LOF(p) is.
Deviation-Based Outlier Detection:
Deviation-based outlier detection does not use statistical tests or distance-basedmeasures to identify exceptional objects. Instead, it identifies outliers by examining themain characteristics of objects in a group.Objects that ―deviate‖ fromthisdescription areconsidered outliers. Hence, in this approach the term deviations is typically used to referto outliers. In this section, we study two techniques for deviation-based outlier detection.The first sequentially compares objects in a set, while the second employs an OLAPdata cube approach.

Sequential Exception Technique:

The sequential exception technique simulates the way in which humans can distinguishunusual objects from among a series of supposedly like objects. It uses implicit redundancyof the data. Given a data set, D, of n objects, it builds a sequence of subsets,{D1, D2, ...,Dm}, of these objects with 2<=m <= n such that

Dissimilarities are assessed between subsets in the sequence. The technique introducesthe following key terms.
Exception set:

This is the set of deviations or outliers. It is defined as the smallestsubset of objects whose removal results in the greatest reduction of dissimilarity in the residual set.

Dissimilarity function:

This function does not require a metric distance between theobjects. It is any function that, if given a set of objects, returns a lowvalue if the objectsare similar to one another. The greater the dissimilarity among the objects, the higherthe value returned by the function. The dissimilarity of a subset is incrementally computedbased on the subset prior to it in the sequence. Given a subset of n numbers, {x1, ...,xn}, a possible dissimilarity function is the variance of the numbers in theset, that is,

where x is the mean of the n numbers in the set. For character strings, the dissimilarityfunction may be in the form of a pattern string (e.g., containing wildcard charactersthat is used to cover all of the patterns seen so far. The dissimilarity increases when the pattern covering all of the strings in Dj-1 does not cover any string in Dj that isnot in Dj-1.

Cardinality function:

This is typically the count of the number of objects in a given set.

Smoothing factor:

This function is computed for each subset in the sequence. Itassesses how much the dissimilarity can be reduced by removing the subset from theoriginal set of objects

WEBSITES: PPT's

https://www.slideshare.net/akannshat/data-mining-15329899
https://www.slideshare.net/smj/data-mining-slides
https://www.slideshare.net/zafarjcp/data-mining-association-rules-basics
https://www.slideshare.net/Tommy96/data-mining-concepts-and-techniques-4036310
https://www.slideshare.net/Tommy96/data-miningppt-4035580

QUESTION BANK

Course Title : Data Mining
Course Code : DS3103PC
Regulation : R21

UNIT-1

S.No	Questions	BT	CO	PO
Part – A (Short Answer Questions)				

UNIT 2	1		What is Data Mining?	L4		
	2		What is KDD process?	L1		
	3		Explain Data cleaning method.	L1		
	4		What is Data Preprocessing?	L2		
	5		What is data reduction?	L1		
	6		Draw a neat diagram of Data Mining architecture.	L1		
	7		List different types of Data Mining tasks.	L2		
	8		Deference between Data cleaning and Data reduction.	L2		
	9		Explain data transformation method.	L1		
	1 0		What is Data transformation? Explain.	L2		
	Part – B (Long Answer Questions)					
	11	a)	What is KDD process? Explain with neat diagram.	L4		
		b)	Explain the Noisy and missing data.	L1		
	12	a)	What is data mining? Explain architecture of data mining.	L1		
		b)	Explain Dimensionality reduction process with diagram.	L2		
	13		Explain the following: a. Data cleaning	L1		
		b)	Data transformation	L1		
	14	a)	Explain different types of data preprocessing techniques.	L2		
		b)	Explain in detail about measures of similarity and Dissimilarity.	L2		
	15		Discuss about the following: a. missing data	L1		
		b)	Dimensionality Reduction	L2		
	S.No		**Questio ns**	**BT**	**CO**	**P O**
	Part – A (Short Answer Questions)					
	1		How the Association rule is helpful to growth of business.	L4	2	
	2		What are disadvantages of Apriori Algorithm?	L1	2	
	3		What is market basket analysis? Explain.	L1	2	
	4		Discuss the applications of association analysis.	L2	2	
	5		What are the advantages of FP-Growth algorithm?	L1	2	
	6		Write a step by step process of FP Growth Algorithm.	L1	2	
	7		Write about basic concept in Association Rule Mining.	L2	2	
	8		Explain the process of Apriori Algorithm.	L2	2	
	9		What are the advantages and disadvantages of ECLAT algorithm?	L1	2	
	1 0		Explain the process of ECLAT algorithm.	L2	2	
	Part – B (Long Answer Questions)					
	11	a)	Explain about the Apriori algorithm for finding frequent item sets with an example.	L4	2	
		b)	Write a step by step process of Apriori algorithm.	PO sL1	2	

				BT	CO	PO
UNIT 3	12	a)	Discuss about basic concepts of frequent item set mining.	L1	2	
		b)	Write the advantages and disadvantages of Apriori Algorithm.	L2	2	
	13	a)	Write about basic concept in Association Rule Mining.	L1	2	
		b)	Can we overcome the draw backs of Apriori algorithm? Discuss.	L1	2	
	14		Explain the following: a. Frequent item set.	L2	2	
		b)	Closed Frequent item set	L2	2	
	15	a)	What are the drawbacks of Apriori Algorithm? Explain.	L1	2	
		b)	Write a FP Growth Algorithm and explain.	L2	2	
UNIT 4	**S.No**		**Questions**	**BT**	**CO**	**PO**
			Part – A (Short Answer Questions)			
	1		What is difference between classification and prediction?	L4	3	
	2		What is Bayes theorem? Explain.	L1	3	
	3		Write a note attribute selection measures.	L1	3	
	4		what is prediction?	L2	3	
	5		Discuss about Naïve Bayesian Classification.	L1	3	
	6		Draw the structure of Decision tree with example.	L1	3	
	7		Explain Baysain belief network.	L2	3	
	8		Discuss about Accuracy and Error measures.	L2	3	
	9		What is classification? Explain.	L1	3	
	10		What are the uses of classification technic?	L2	3	
			Part – B (Long Answer Questions)			
	11		Describe the data classification process with a neat diagram. How does the Naive Bayesian Classification works? Explain.	L1	3	
	12	a)	Describe the data classification process with a neat diagram.	L2	3	
		b)	How does the Naive Bayesian Classification works? Explain.	L2	3	
	13		Write and explain about Classification by Back propagation Algorithm.	L1	3	
	14	a)	What is Bayesian belief network? Explain in detail.	L2	3	
		b)	Write a note attribute selection measures.	L1	3	
	15	a)	Explain decision tree induction algorithm for classifying data tuples and discuss suitable Example.	L1	3	
		b)	Explain construction of decision tree with example.	L2	3	
	S.No		**Questions**	**BT**	**CO**	**PO**
			Part – A (Short Answer Questions)			
	1		Explain about Types of Data in Cluster Analysis?	L4	4	

UNIT 5

* Blooms Taxonomy Level (BT) (L1 – Remembering; L2 – Understanding; L3 – Applying; L4 – Analyzing; L5 – Evaluating; L6 – Creating)

S.No		Questions	BT	CO	PO
2		Differentiate between AGNES and DIANA algorithms.	L1	4	
3		Write a K-means clustering algorithm.	L1	4	
4		Explain Grid based clustering methods.	L2	4	
5		Write the key issue in hierarchical clustering algorithm.	L1	4	
6		What is the use of clustering?	L1	4	
7		Explain inter clustering.	L2	4	
8		What is clustering? How it is useful to business.	L2	4	
9		Explain un supervised data.	L1	4	
10		What is intra clustering?	L2	4	
		Part – B (Long Answer Questions)			
11	a)	Classify various Clustering methods.	L4	4	
	b)	Write any one Partitioning based clustering methods.	L1	4	
12	a)	What is the goal of clustering? Explan.	L1	4	
	b)	How does partitioning around medoids algorithm achieve this?	L2	4	
13		Explain the following: a) Density based clustering methods	L1	4	
	b)	Grid based clustering methods.	L1	4	
14	a)	Give a brief note on PAM Algorithm.	L2	4	
	b)	What is the drawback of k-means algorithm? How can we modify thealgorithm to diminish?	L2	4	
15	a)	What is outlier detection? Explain distance based outlier detection.			
	b)	Write partitioning around mediods algorithm.			

JNTU OLD QUESTION PAPERS

Code No: 117CD
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech IV Year I Semester

S.No		Questions	BT	CO	PO
		Part – A (Short Answer Questions)			
1		What is web mining?	L1	5	
2		What is Text mining?	L1	5	
3		Explain text clustering.	L2	5	
4		What is web content mining?	L1	5	
5		Describe hierarchy of categories	L1	5	
6		Difference between web mining and text mining.	L2	5	
7		Explain the different data mining methods.	L1	5	
8		What is the use of Text mining?	L1	5	
9		Explain types of text mining.	L2	5	
10		Explain applications of web mining.	L2	5	
		Part – B (Long Answer Questions)			
11	a)	Discuss about the Web mining	L1	5	
	b)	Write note about the web content mining.	L1	5	
12	a)	Explain in detail about text mining.	L2	5	
	b)	What is Text clustering? Explain in details.	L1	5	
13	a)	What is mining? Explain web structure mining.	L1	5	
	b)	Explain how data mining is useful to Data Science.	L2	5	
14		Explain the following: a) Unstructured text	L2	5	
	b)	Hierarchical categories.	L1	5	
15		Discuss about the episode rule discovery for texts.	L2	5	

Examinations, November/December – 2017 DATA WAREHOUSING AND DATA MINING (Computer Science and Engineering) Time: 3 Hours
Max. Marks: 75

Note: This question paper contains two parts A and B. Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART - A

1.a) Define data warehouse. [2]
b) List the Data warehouse Characteristics. [3]
c) How can you go about filling in the missing values for this attribute? [2]
d) Why is the word data mining a misnomer? [3]
e) Give a note on Closed Frequent Item Set. [2]
f) Write the FP-graph algorithm. [3]
g) How prediction is different from classification? [2]
h) What is rule classification? [3]
i) Give a note on k means algorithm. [2]
j) List the Key Issues in Hierarchical Clustering. [3]

PART – B (50 Marks)

2.a) Make a comparisons between the MOLAP and HOLAP
. b) Discuss the star and snowflake schema in detail with suitable example.
[5+5]
OR
3.a) Write the difference between designing a data warehouse and an OLAP cube.
b) Give a brief note on ROLAP. [5+5]
4. Explain concept hierarchy generation for the nominal data. [10]
OR
5.a) Describe the Feature Subset Selection. b) Illustrate the Data Transformation by Normalization. [5+5]
6. Make a comparison of Apriori and ECLAT algorithms for frequent item set mining in transactional databases.
Apply these algorithms to the following data:

TID LIST OF ITEMS
1Bread, Milk, Sugar, TeaPowder, Cheese, Tomato
2Onion, Tomato, Chillies, Sugar, Milk
3Milk, Cake, Biscuits, Cheese, Onion
4Chillies, Potato, Milk, Cake, Sugar, Bread
5Bread, Jam, Mik, Butter, Chilles
6Butter, Cheese, Paneer, Curd, Milk, Biscuits
7Onion, Paneer, Chilies, Garlic, Milk
8Bread, Jam, Cake, Biscuits, Tomato [10]

OR

7. Briefly explain the Partition Algorithms. [10]

8. Discuss K- Nearest neighbor classification-Algorithm and Characteristics. [10]

OR

9. How does the Naïve Bayesian classification works? Explain in detail. [10]

10. a) Give a brief note on PAM Algorithm.

b) What is the drawback of k-means algorithm? How can we modify the algorithm to diminish that problem? [5+5]

OR

11. What are the different clustering methods? Explain in detail. [10]

Code No: 117CD

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech IV Year I Semester Examinations, April/May - 2018 DATA WAREHOUSING AND DATA MINING

(Computer Science and Engineering)

Time: 3 Hours Max.

Marks: 75

Note: This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART A

1.a) List out the operations of OLAP. [2]

b)What is fact table? Write its uses. [3]

c)Define discretization. [2]

d)What is predictive mining? Explain it briefly [3]

e)Write the purpose of Apriori algorithm. [2]

f)Define support and confidence measure. [3]
g)What is boosting? [2]
h)Define decision tree. [3]
i)Write the strengths of hierarchical clustering. [2]
j)Compare agglomerative and divisive methods. [3]

PART-B (50 Marks)

2.a) With a neat sketch, Explain three tier architecture of data ware housing.
b) Explain various data warehouse models. [5+5]
OR
3.Write a note on
a)Relational OLAP
b)Multi dimensional OLAP. [5+5]
4.a) Discuss in detail about the steps of knowledge discovery?
b) Write a note on subset selection in attributes for data reduction
OR
5.a) Explain various data mining tasks.
b) Discuss briefly about data cleaning techniques. [5+5]
6.a) Write FP- growth algorithm.
b) Explain how association rules are generated from frequent item sets. [5+5]
OR
7.a) Explain the procedure to mining closed frequent data item sets.
b) Explain, how can you improve the performance of Apriori algorithm.
8.a) What is Bayesian belief network? Explain in detail.
b) Write a note attribute selection measures. [5+5]
OR
9.a) Write k-nearest neighbor classification algorithm and its characteristics.
b) Write decision tree induction algorithm. [5+5]
10.a) What is outlier detection? Explain distance based outlier detection.
b) Write partitioning around mediods algorithm. [5+5]
OR
11.a) Write K-means clustering algorithm.
b) Write the key issue in hierarchical clustering algorithm

Code No: 157BC

R18

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech IV Year I Semester Examinations, February/March - 2022
DATA MINING
(Common to CSE, IT)

Time: 3 Hours **Max. Marks: 75**

Answer any Five Questions
All Questions Carry Equal Marks

- - -

1. Explain the need of data preprocessing and various forms of preprocessing. [15]

2. What is a data warehouse? Demonstrate integrating data mining system with a data warehouse with a neat diagram. [15]

3. Apply FP-Growth algorithm to the following data for finding frequent item sets, consider support threshold as 30%. [15]

TID	List of ItemIDs
1	I1, i2, i4, i5
2	I2, i4, i7
3	I2,i3,i4,i5
4	I1,i3,i4,i7
5	I1,i2,i3,i4,i5
6	I3,i4,i5,i6

4.a) How to identify sub graphs in a graph?
b) Give an overview of correlation analysis. [8+7]

5.a) Explain classification as a two step process.
b) State Bayes theorem. How this concept is used in classification. [8+7]

6. What is a decision tree? Explain decision tree induction algorithm. [15]

7.a) Contrast k-means clustering with k-medoids clustering approach.
b) Discuss the merits and demerits of hierarchical approaches for clustering. [8+7]

8. How to apply mining techniques to unstructured text database? Explain with example. [15]

VIDEO LINKS:

https://www.youtube.com/watch?v=0FLmrC3-P1A&list=PLERZXVMwiajp7_9-1f0-vhUNtXmc62xxB
https://www.youtube.com/watch?v=JQ0j_aoIOIE&list=PLERZXVMwiajp7_9-1f0-vhUNtXmc62xxB&index=2
https://www.youtube.com/watch?v=HNqr4YYbV0w&list=PLERZXVMwiajp7_9-1f0-vhUNtXmc62xxB&index=3
https://www.youtube.com/watch?v=iCJmTYu6Inw&list=PLERZXVMwiajp7_9-1f0-vhUNtXmc62xxB&index=4
https://www.youtube.com/watch?v=xEmrFePGjEg&list=PLmAmHQ-_5ySxFoIGmY1MJao-XYvYGxxgj