DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE MATERIALS



CS 402 DATAMINING AND WAREHOUSING

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT DEPARTMENT

- Established in: 2002
- Course offered : B.Tech in Computer Science and Engineering

  M.Tech in Computer Science and Engineering

  M.Tech in Cyber Security
- Approved by AICTE New Delhi and Accredited by NAAC
- Affiliated to the University of A P J Abdul Kalam Technological University.

DEPARTMENT VISION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.
5.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

PEO2: Graduates will be able to analyze, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

PEO3: Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

PEO4: Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Teamwork and leadership qualities.

PROGRAM OUTCOMES (POS)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

COURSE OUTCOMES

C402.1	To identify the key process of Data mining and Warehousing and its applications
C402.2	To apply appropriate techniques to convert raw data into suitable format for practical data mining tasks
C402.3	To analyze various classification algorithms and apply in appropriate domain
C402.4	To analysis and evaluate the performance of various classification methods using performance metrics
C402.5	To make use of the concept of association rule mining in real world scenario
C402.6	To select appropriate clustering and algorithms for various applications and extend data mining methods to the new domains of data

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES

CO'S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C402.1	3	3			2							2
C402.2	3	2	3	3	2							3
C402.3	3	3	3	3	3							2
C402.4	3	3	3	3	3							3
C402.5	3	3	3	3	3							2
C402.6	3	3	2	3	3							3
C402	3	2.83	2.8	3	2.67							2.5

Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

CO'S	PSO1	PSO2	PSO3
C402.1	3		
C402.2			3
C402.3		3	3
C402.4	3	3	
C402.5		3	3
C402.6		3	3
C402	3	3	3

SYLLABUS

Course code	Course Name	L-T-P Credits	Year of Introduction
CS402	DATA MINING AND WAREHOUSING	3-0-0-3	2016

Course Objectives:

- To introduce the concepts of data Mining and its applications
- To understand investigation ofe data using practical data mining tools.
- To introduce Association Rules Mining
- To introduce advanced Data Mining techniques

Syllabus:

Data Mining, Applications, Data Mining Models, Data Warehousing and OLAP, Challenges, Tools, Data Mining Principles, Data Preprocessing: Data Preprocessing Concepts, Data Visualization, Data Sets and Their Significance, Classification Models, Multi Resolution Spatial Data Mining, Classifiers, Association Rules Mining, Cluster Analysis, Practical Data Mining Tools, Advanced Data Mining Techniques, Web Mining, Text Mining, CRM Applications and Data Mining, Data warehousing.

Expected Outcome:

The Student will be able to :

i. identify the key process of Data mining and Warehousing
ii. apply appropriate techniques to convert raw data into suitable format for practical data mining tasks
iii. analyze and compare various classification algorithms and apply in appropriate domain
iv. evaluate the performance of various classification methods using performance metrics
v. make use of the concept of association rule mining in real world scenario
vi. select appropriate clustering and algorithms for various applications
vii. extend data mining methods to the new domains of data

Text Books:

1. Dunham M H, "Data Mining: Introductory and Advanced Topics", Pearson Education, New Delhi, 2003.
2. Jaiwei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Elsevier, 2006.

References:

1. M Sudeep Elayidom, "Data Mining and Warehousing", 1st Edition, 2015, Cengage Learning India Pvt. Ltd.
2. Mehmed Kantardzic, "Data Mining Concepts, Methods and Algorithms", John Wiley and Sons, USA, 2003.
3. Pang-Ning Tan and Michael Steinbach, "Introduction to Data Mining", Addison Wesley, 2006.

Course Plan			
Module	**Contents**	**Hours**	**End Sem Exam. Marks**
I	Data Mining:- Concepts and Applications, Data Mining Stages, Data Mining Models, Data Warehousing (DWH) and On-Line Analytical Processing (OLAP), Need for Data Warehousing, Challenges, Application of Data Mining Principles, OLTP Vs DWH, Applications of DWH	6	15%
II	Data Preprocessing: Data Preprocessing Concepts, Data Cleaning, Data integration and transformation, Data Reduction, Discretization and concept hierarchy.	6	15%
FIRST INTERNAL EXAM			
III	Classification Models: Introduction to Classification and Prediction, Issues regarding classification and prediction, Decision Tree- ID3, C4.5, Naive Bayes Classifier.	6	15%
IV	Rule based classification- IR. Neural Networks-Back propagation. Support Vector Machines, Lazy Learners-K Nearest Neighbor Classifier. Accuracy and error Measures-evaluation. Prediction:-Linear Regression and Non-Linear Regression.	6	15%
SECOND INTERNAL EXAM			
V	Association Rules Mining: Concepts, Apriori and FP-Growth Algorithm. Cluster Analysis: Introduction, Concepts, Types of data in cluster analysis, Categorization of clustering methods. Partitioning method: K-Means and K-Medoid Clustering.	8	20
VI	Hierarchical Clustering method: BIRCH. Density-Based Clustering –DBSCAN and OPTICS. Advanced Data Mining Techniques: Introduction, Web Mining- Web Content Mining, Web Structure Mining, Web Usage Mining. Text Mining. Graph mining:- Apriori based approach for mining frequent subgraphs. Social Network Analysis:- characteristics of social networks. Link mining:- Tasks and challenges.	8	20
END SEMESTER EXAMINATION			

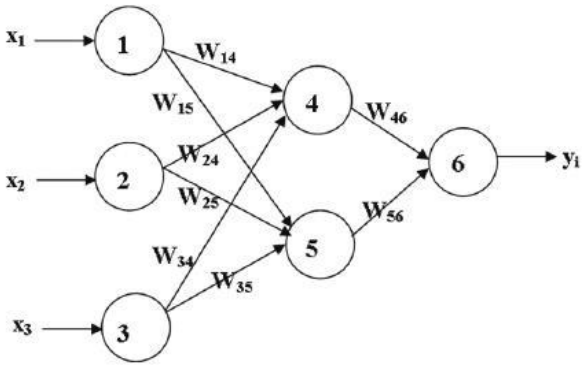
Question Paper Pattern

1. There will be ***FOUR*** parts in the question paper – **A, B, C, D**
2. **Part A**
   a. **Total marks : 40**
   b. ***TEN*** questions, each have **4 marks**, covering **all the SIX modules** (***THREE*** questions from **modules I & II**; ***THREE*** questions from **modules III & IV**; ***FOUR*** questions from **modules V & VI**).
   All the TEN questions have to be answered.
3. **Part B**
   a. **Total marks : 18**
   b. ***THREE*** questions, each having **9 marks**. One question is from **module I**; one question is from **module II**; one question ***uniformly*** covers **modules I & II**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
4. **Part C**
   a. **Total marks : 18**
   b. ***THREE*** questions, each having **9 marks**. One question is from **module III**; one question is from **module IV**; one question ***uniformly*** covers **modules III & IV**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
5. **Part D**
   a. **Total marks : 24**
   b. ***THREE*** questions, each having **12 marks**. One question is from **module V**; one question is from **module VI**; one question ***uniformly*** covers **modules V & VI**.
   c. ***Any TWO*** questions have to be answered.
   d. Each question can have ***maximum THREE*** subparts.
6. There will be ***AT LEAST 60%*** analytical/numerical questions in all possible combinations of question choices.

QUESTION BANK

CS402-DATAMINING AND DATA WAREHOUSING

SL NO.	QUESTIONS	COS	KL	PAGE NO.
	MODULE I			
1.	Explain steps in knowledge discovery	CO 1	K5	13
2.	Explain OLAP and OLTP	CO 1	K3	23
3	Describe about star and snow flake model	CO 1	K2	27
4.	Describe about datacube with example	CO 1	K2	24
5.	Explain OLAP operations	CO 1	K5	30
6.	Draw and explain the datamining architecture	CO 1	K3	14
7.	Compare OLTP and DW	CO 1	K3	33
	MODULE II			
1.	Describe about Data cleaning and transformation.	CO2	K2	40
2.	Distinguish between Data reduction and Transformation with example.	CO2	K5	54
3.	Explain different methods involving data transformation	CO2	K5	51
4.	Define the role of Concept Hierarchy	CO2	K2	66
5.	Explain different smoothing techniques	CO2	K3	41
6.	Explain the term classification with neat diagram	CO2	K5	63
7.	Analyze the different steps used in Data Pre-Processing	CO2	K4	38
8.	Describe about aggregation	CO2	K2	55
	MODULE III			
1.	Briefly explain about naïve bayes classifier	CO3	K5	101
2.	Explain in detail about classification by decision tree	CO3	K5	80
3.	Briefly explain about information gain	CO3	K3	86
4.	Explain in detail about classification and prediction	CO3	K3	75

5.	Briefly explain about CART	CO3	K3	93
6.	Explain about types of tree pruning	CO3	K5	97
7.	Explain GAIN Ratio	CO3	K3	91
8.	Explain about Gain index	CO3	K3	93
9.	Describe the issues regarding classification and prediction	CO3	K2	77
10	Explain about ID3		K3	86
	MODULE IV			
1.	Write a short note on support vector machine	CO4	K1	124
2.	Explain in detail about Lazy learners	CO4	K5	128
3.	Give a detailed note on Linear Regression and Non-Linear Regression.	CO4	K3	133
4.	Explain K-nearest Neighbor Classifier	CO4	K5	129
5.	The following figure shows a multilayer feed-forward neural network. Let the learning rate be 0.9. The initial weight and bias values of the network is given in the table below. The activation function used is the sigmoid function 	CO4	K3	122
6.	Explain in detail on BackPropagation neural network	CO4	K5	114
7.	Give a detailed note on accuracy and error measures	CO4	K1	136
	MODULE V			
1.	Write a short note on the concepts of Association Rule Mining.	CO5	K1	143
2.	Explain K-Means Partition algorithm	CO5	K5	178
3.	Consider the transaction database given below. Set minimum support count as 2 and minimum confidence threshold as 70% a) Find the frequent item-set using FP Growth Algorithm.	CO5	K3	148

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4

	b) Generate strong association rules. Transaction ID | List of Item_Ids T100 | I1,I2,I5 T200 | I2,I4 T300 | I2,I3 T400 | I1,I2,I4 T500 | I1,I3 T600 | I2,I3 T700 | I1,I3 T800 | I1,I2,I3,I5 T900 | I1,I2,I3			
4.	Explain in detail on the Partitioning method – K Medoid Clustering	CO5	K5	183
5.	What is meant by clustering? What are the major categories of clustering method?	CO5	K2	162
6.	Explain K- Medoid partition algorithm	CO5	K5	180
7.	Given two objects represented by the tuples (22,1,42,10) and (20,0,36,8) Compute the Manhattan distance between the two objects	CO5	K3	164
8.	How to compute the dissimilarity between objects described by binary variables?	CO5	K3	165
	MODULE VI			
1.	State the pros and cons of DBSCAN method.	C06	K2	196
2.	Based on the following data determine the gender of a person having height 6 ft., weight 130 lbs. and foot size 8 in. (use Naive Bayes algorithm).	C06	K5	101

person	height (feet)	weight (lbs)	foot size (inch
male	6.00	180	10
male	6.00	180	10
male	5.50	170	8
male	6.00	170	10
female	5.00	130	8
female	5.50	150	6
female	5.00	130	6
female	6.00	150	8

3.	Explain BIRCH clustering method.	C06	K5	191
4.	Explain in detail on Balanced Iterative Reducing and Clustering using Hierarchies	C06	K5	191
5.	What are the advantages of BIRCH compared to other clustering methods	C06	K4	195
6.	Distinguish between agglomerative and divisive hierarchial clustering	C06	K4	187
7.	Explain about OPTICS algorithm	C06	K5	199

APPENDIX 1

CONTENT BEYOND THE SYLLABUS

S:NO;	TOPIC	PAGE NO:
1	Data Mining tools	201

MODULE NOTES

.

.

.

.

.

.

DDM-1

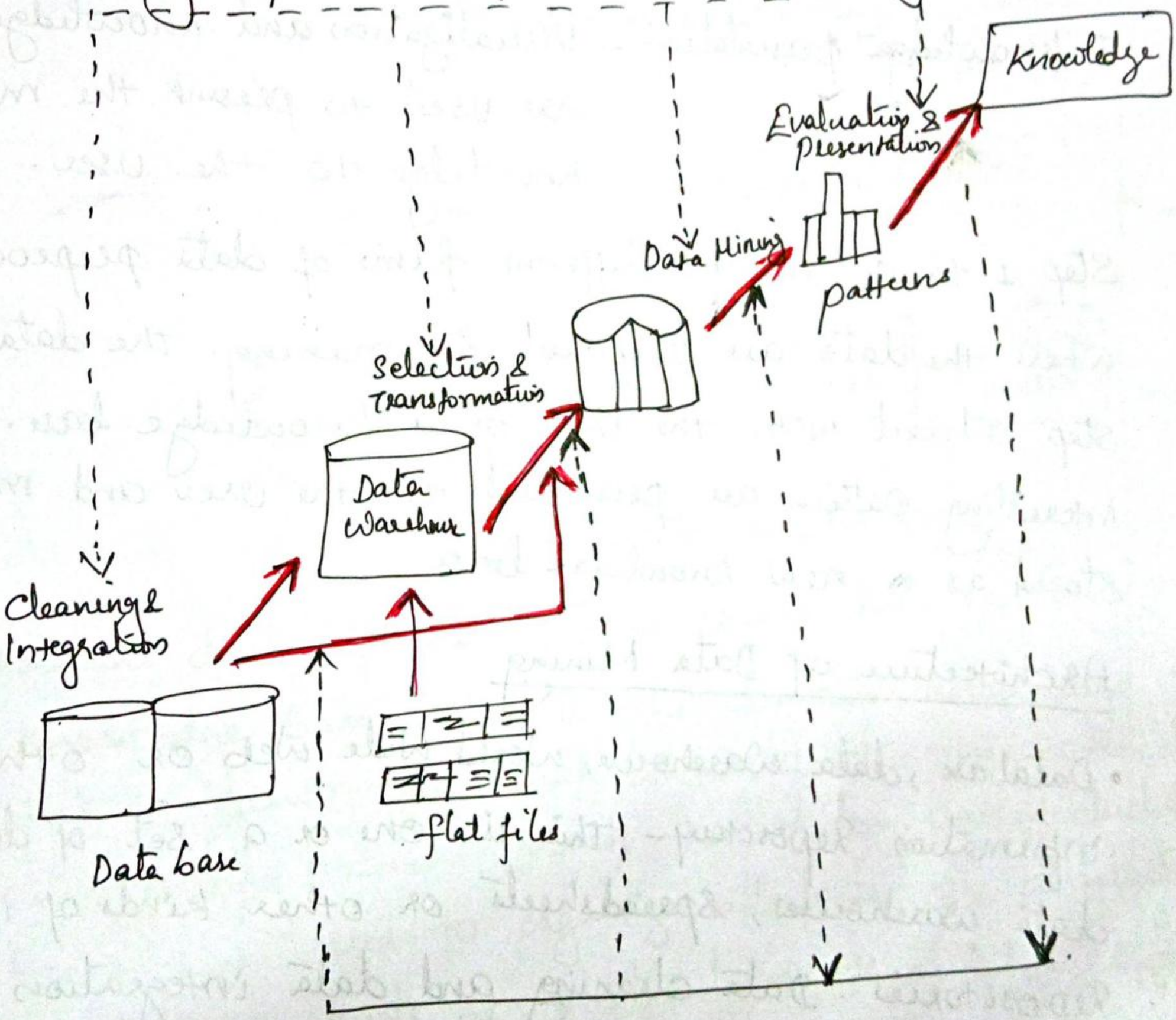
Data Mining - Concepts

Data Mining refers to extracting or "mining" knowledge from large amount of data.

Datamining ⟹ Knowledge Mining

Datamining can also treat as knowledge discovery from Data or KDD.

DataMining Steps is the process of knowledge discovery



1. Data cleaning - Removes noise and inconsistent data
2. Data Integration - multiple data sources may be combined
3. Data Selection - Data relevant to the analysis task are retrieved from the database

Scanned by TapScanner

4. Data Transformation – Data are transformed or consolidated into appropriate forms for mining by performing summary or aggregation operations

5. Data Mining – ~~Datt~~ Data patterns are extract by applying intelligent methods.

6. pattern Evaluation – Identify the truely interesting patterns representing knowledge

7. knowledge presentation – Visualization and knowledge techniques are used to present the mined knowledge to the user.

Step 1 to 4 are the different forms of data preprocessing where the data are prepared for mining. The data mining step interact with the user or a knowledge base. The interesting pattern are presented to the user and may be stored as a new knowledge base

Architecture of Data Mining

- Database, data warehouse, World Wide Web or other information Repository – This is one or a set of databases, data warehouses, spreadsheets or other kinds of information Repositories. Data cleaning and data integration techniques may be performed on the data
- Database or data warehouse Server – The database or data warehouse server is responsible for fetching the

Scanned by TapScanner

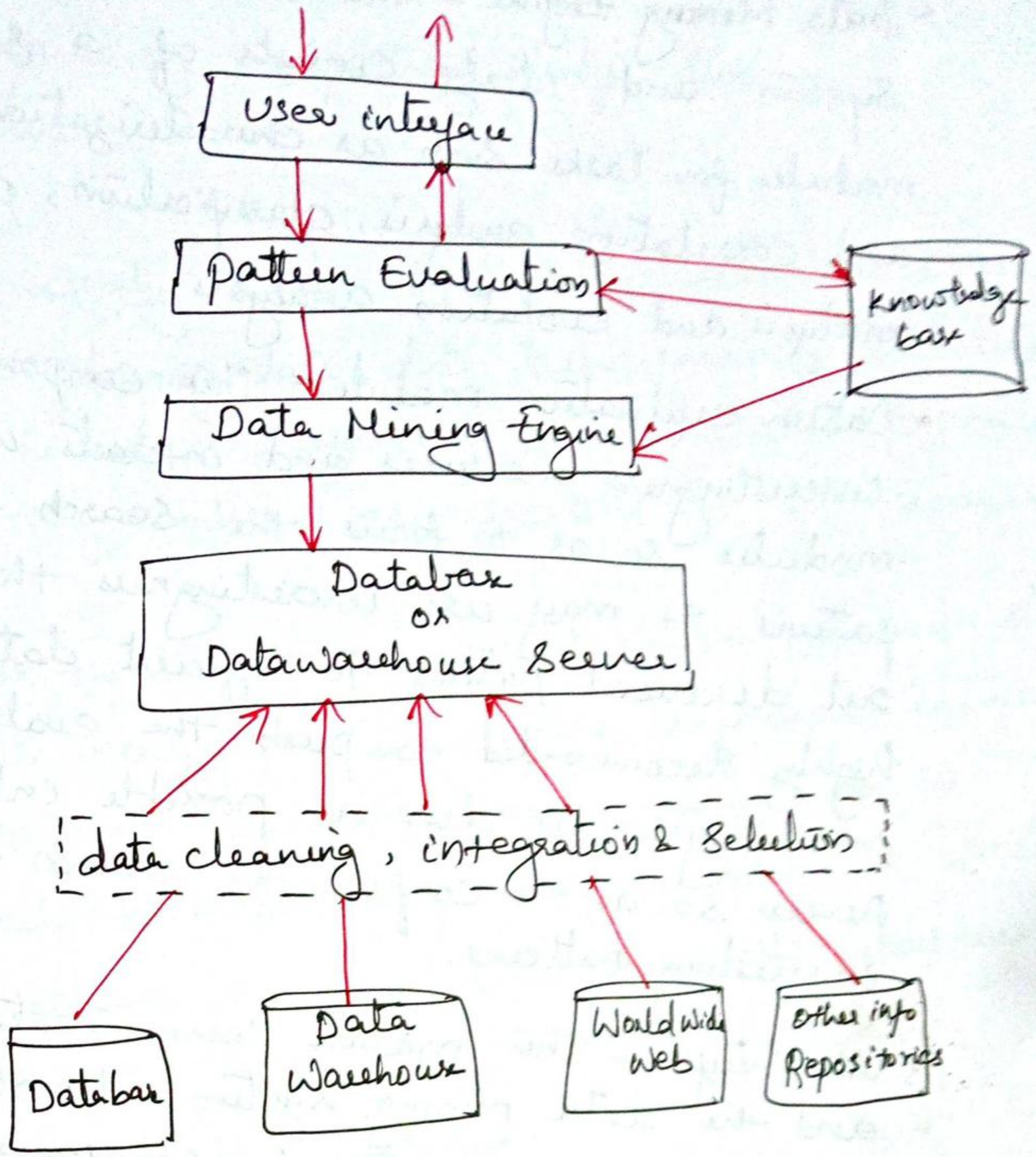


fig: Architecture of Data Mining

relevant data, based on the users data mining request.

- Knowledge Base - This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include hierarchies used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to access the pattern. Other examples of domain knowledge are additional interestingness constraints or threshold and metadata.

Scanned by TapScanner

- Data Mining Engine - This is essential to the data mining systems and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis and evolution analysis
- Pattern evaluation module - This component employs interestingness measures and interacts with the data mining modules so as to focus the search towards interesting patterns. It may use interestingness thresholds to filter out discovered patterns. 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.
- 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. This component allows the user to browse database and data warehouse schemas or data structures evaluate mined patterns, and visualize the patterns in different forms.

Stages in Datamining

- Data Mining is an analytic process designed to explore data. The ultimate goal of datamining is prediction. The process of data mining consists of 3 stages

Scanned by TapScanner

1) Initial exploration
2) Model building or pattern identification
3) Deployment

Exploration

This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records - performing some selection operations to bring the number of variables to a manageable range.

Depending on the nature of analytic problem, the first stage of the data mining process involve anywhere between a simple choice of straightforward predictors for a regression model to exploratory analyses using a wide variety of graphical and statistical methods inorder to identify the most relevant variables

Model Building and Validation

This stage involves considering various models and choosing the best one based on their predictive performance. There are variety of techniques developed to achieve the goal - many of which are based on so-called competitive evaluation of models - that is applying different models to same data set and comparing their performance to choose the best

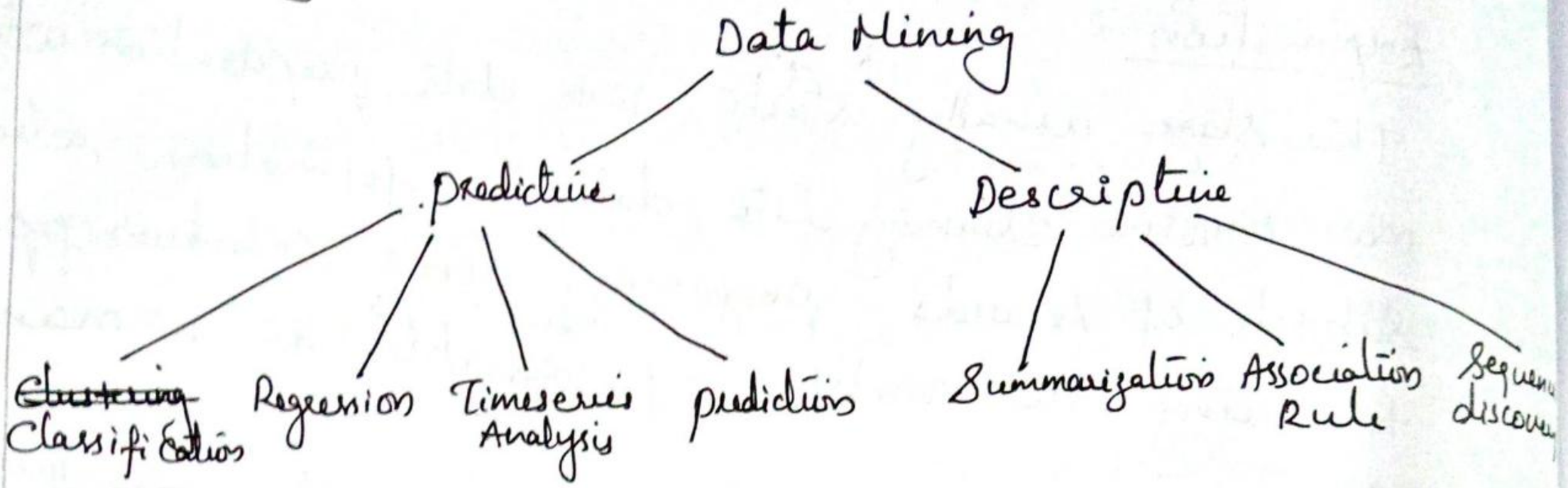
Deployment

This final stage involves using the selected best model in the previous stages and applying it to the new

Scanned by TapScanner

data in order to generate predictions and estimate of the expected outcome.

Data Mining Models



The data mining models are basically divided into two types

1. predictive
2. Descriptive

The Descriptive models recognizes the designs or relationships in data and discovers the properties of the studied data. It identifies patterns or relationship in Data

Summarization maps data into subsets with associated simple descriptions. Summarization is also called characterization or generalization. It extracts or derives representative information about the database. This may be accomplished by actually retrieving portions of the data. Alternatively summary type information can be derived from the data. Eg: Netflow data of a particular University can be calculated by considering a small portion of the dataflow and summarizing for the whole dataflow in the University.

Scanned by TapScanner

Association Rules refers to the data mining task of uncovering relationships among data. An association rule is a model that identifies specific types of data associations. These associations are often used in the retail sales community to identify items that are frequently purchased together. eg: In the supermarket if the customer bought tea then the probability of buying sugar is more. So there exists an association rule or link between tea and sugar.

Sequence discovery is used to determine sequential patterns in data. These patterns are based on a time sequence of actions. These patterns are similar to associations in that data are found to be related, but the relationship is based on time.

Eg: Webmaster periodically analyzes the web log data to determine how the users access web pages. ~~He~~ ~~noticed~~. The 70 percent of the users of page A follows one of the following patterns of behaviour (A,B,C) or (A,D,B,C) or (A,E,B,C). Then the webmaster determines to add a link directly from page A to page B & page C.

A predictive model makes a prediction about values of data using known results found from different data. predictive model is based on the use of other historical data. Eg: credit card use might be refused not because of the user's own credit history, but because the current purchase is similar to earlier purchases that were subsequently found

Scanned by TapScanner

to be made with the stolen cards.

Classification maps data into predefined groups or classes. It is often referred as supervised learning because the classes are determined before examining the data. Classification algorithms requires that the classes be defined based on data attribute values. Algorithms describe these classes by looking at the characteristics of data already known to belong to the classes.

Pattern Recognition is a type of classification where an input pattern is classified into one of the several classes based on its similarity to these predefined classes.

Eg: classification - Credit Card companies must determine whether to authorize credit card purchases. Each purchase is placed in one of the four classes

1) authorize
2) ask for further identification before authorization
3) do not authorize
4) do not authorize but contact police

Eg: pattern Recognition - Airport security screening station is used to determine if passengers are potential terrorists or criminals. To do this, the face of each passenger is scanned and its basic pattern is identified. This pattern is compared to entries in a database to see if it matches any patterns that are associated with known offenders.

Regression is used to map a data item to a real valued prediction variable. It involves predictor variable (the values which are known) and response variable (value to

Scanned by TapScanner

be predicted). Regression assumes that the target data fit into some known type of function (eg: linear, logistic etc) and then determines the best function of this type that models the given data.

Eg: A dataset contains information about relationship between 'number of hours studied' and 'mark obtained'. Many students have been observed and their hours of study and grade are recorded. This is considered as the training data. our goal is to design a model that can predict marks if the no: of studied hours is given.

Time Series Analysis - the value of an attribute is examined as it varies over time. The value usually are obtained as evenly spaced time points (daily, weekly, hourly etc.

Eg: A passenger need to book airticket for a journey after 6 months. He analyze the rate of tickets in the X, Y & Z flights. and found that Y and Z has the drastic variation in the tickets rate for the past 6 months and he go for the flight X.

Prediction can be viewed as a type of classification. It is a datamining task that is predicting a future state rather than a current state. Here we are refering to a type of application rather than to a type of data mining modeling approach. prediction application including flood prediction, Speech recognition, machine learning and pattern recognition.

Scanned by TapScanner

Eg:- predicting flood approach uses monitors placed at various points in the River. These monitors collect data relevant to flood prediction: water level, rain amount, time, humidity and so on. Then the water level at a potential flooding point in the river can be predicted based on the data collected by the sensors upriver from this point. The prediction must be made with respect to the time the data were collected.

Data Warehousing and OLAP Technology

Data warehouse generalize and consolidate data in multidimensional space. The construction of data warehouses involves data cleaning, data integration and data transformation and can be viewed as an important preprocessing step for data mining.

Data Warehouse provide online analytical processing (OLAP) tools for the interactive analysis of multidimensional data.

Data Warehousing provides architecture and tools for business executives to systematically organize, understand and use their data to make strategic decisions. It allows the integration of variety of application system.

Features of Data Warehouse

* Subject Oriented :-

A data warehouse is organized around major subjects. If it is a company's ongoing operation then subjects

Scanned by TapScanner

~~Such~~ are customer, Supplier, product and Sales. DataWarehouse focus on the modeling and analysis of data for decision makers rather than day-to day operation & transaction processing.

② Integrated :-

A datawarehouse is constructed by integrating multiple heterogenous sources such as relational database, flat files and online transaction records.

③ Time-Variant :-

Datas are stored to provide information from an historic perspective (5-10 years).

④ Non-Volatile :-

Datas stored on DataWarehouse are permanent.

Comparison of OLTP & OLAP Systems

Distinguish features of OLTP & OLAP are as follows.

→ User and system orientation :- An OLTP system is customer oriented and is used for transaction and query processing by clerks, clients and IT professionals. An OLAP is market oriented and is used for data analysis by knowledge workers, including managers, executives and analysts.

→ Data Contents :- An OLTP system manages current data and is typically too detailed for decision making. An OLAP System manages large amount of historic data

Scanned by TapScanner

provides facilities for summarization and aggregation and stores and manages information at different levels of granularity. These features ~~stores~~ makes the data easier to use for informed decision making

⇒ Database design :- An OLTP usually adopts an entity-Relationship (ER) model and an application-oriented database design. But an OLAP system typically adopts either a star or a snowflake model. and a subject-oriented database design.

⇒ View :- An OLTP system focuses mainly on the current data within an enterprises or department without referring to historic data. OLAP systems deal with information that originates from different organization, integrating information from many data stores.

⇒ Access patterns :- The access pattern of OLTP system consist mainly of short, atomic transaction. OLAP systems are mostly read-only operations. and the queries are complex.

Multidimensional Models

- Data warehouses and OLAP tools are based on a multidimensional data model. and the model views data in the form of a data cube.
- A data cube allows data to be modeled and viewed in multiple dimension. It is defined by dimensions & facts.

Scanned by TapScanner

Dimensions are the perspectives or entities with respect to which an organization wants to keep records. Each dimension may have a table associated with it called a dimension Table, which further describes the dimension.

Facts are the numeric measures. A multidimensional data model is organized around a central theme like Sales. This theme is represented by fact table. The fact table contains the names of the facts, or measures as well as keys to each of the related dimension tables

2-D View of Sales Data for Allelectronics According to time and item

location = "Vancouver"

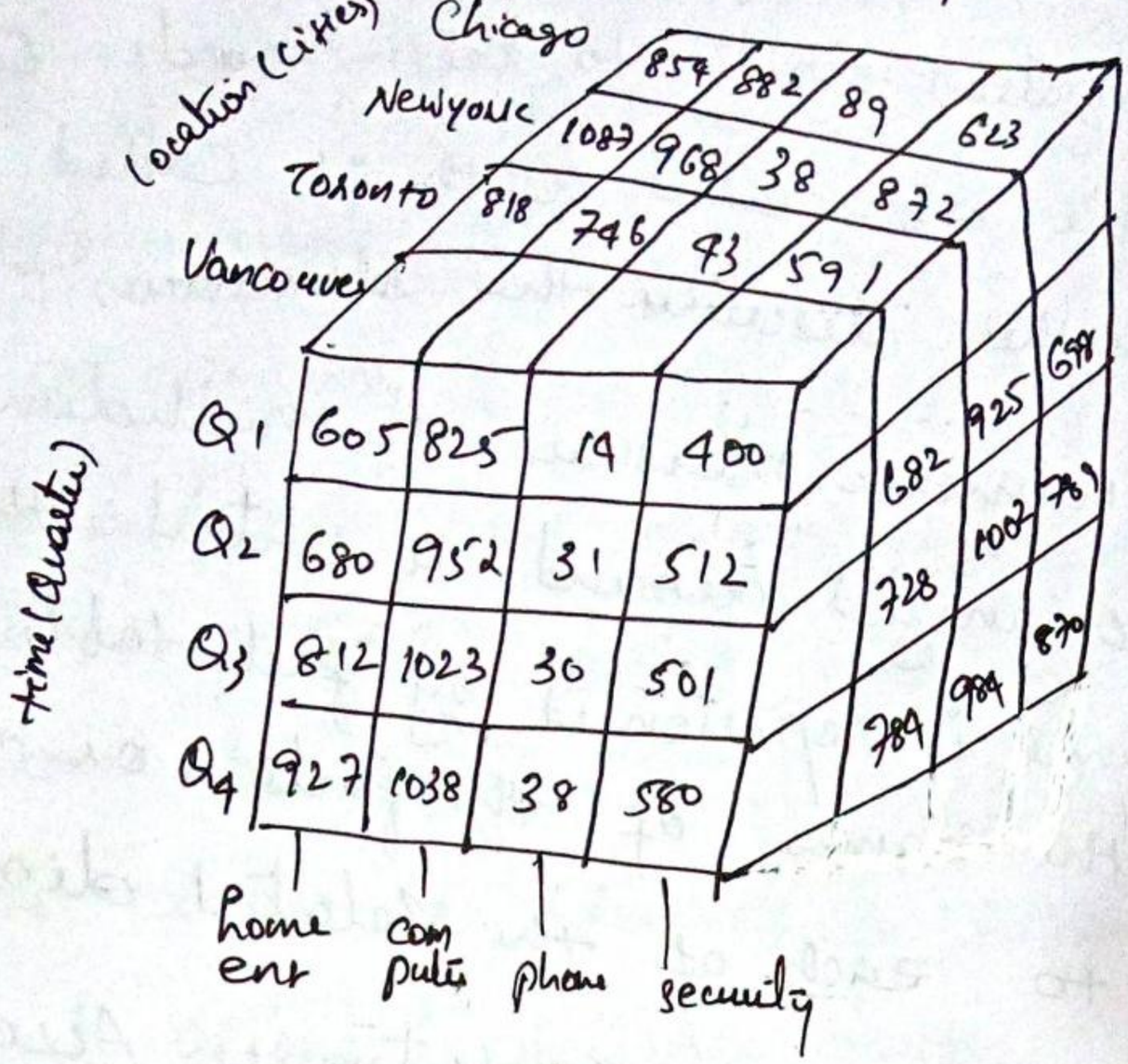
time (Quarter)	item (type) home entertainment.	Computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

3D-View of Sales Data according to time, item & location is given below

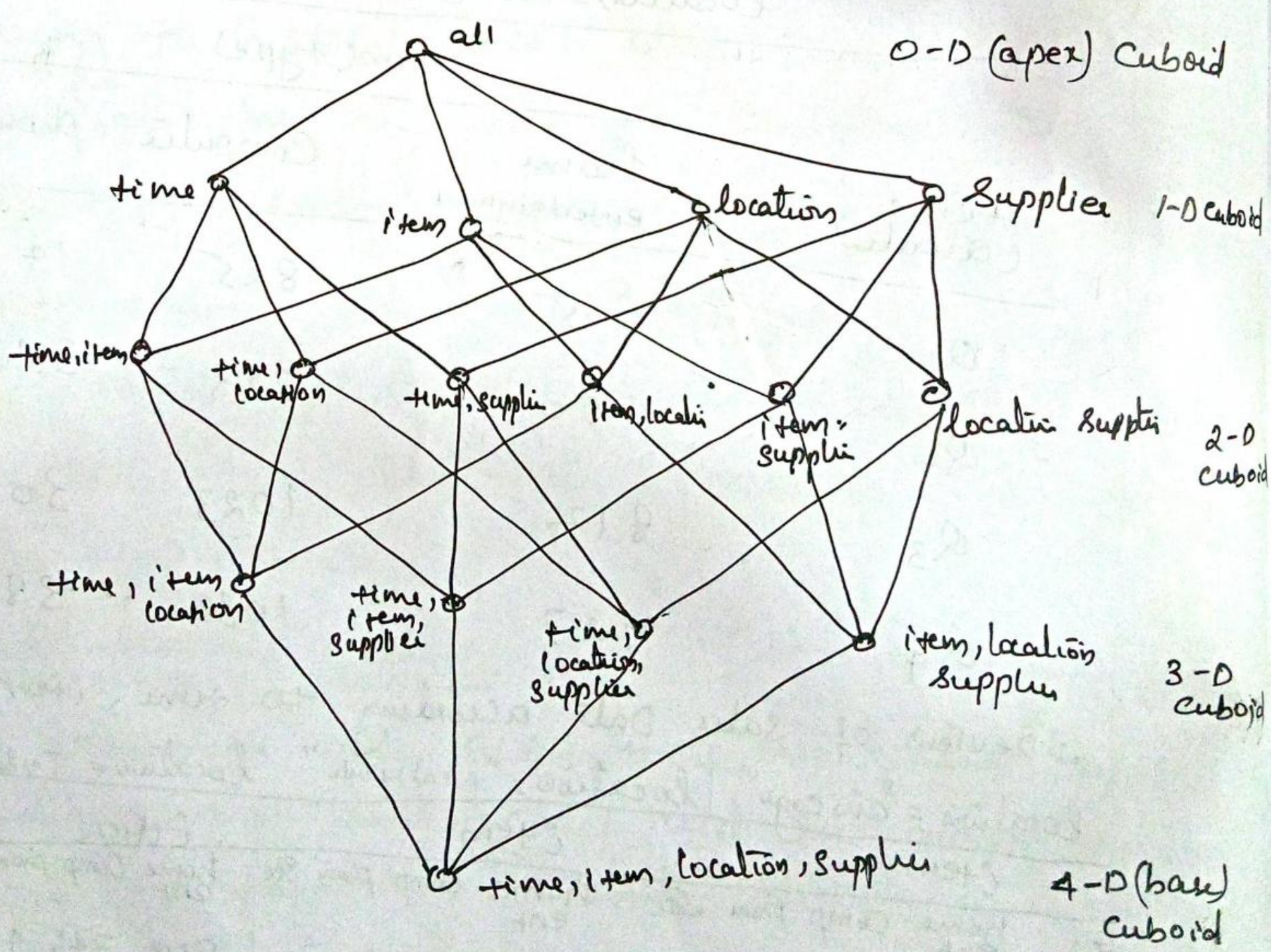
	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
	item				item				item				item			
Time	home ent	Comp	phone	Sec	home ent	Comp	phone	Sec	home ent	Comp	phone	Sec	home ent	comp	phone	sec
Q1	854	882	89	623	1087	968	32	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

Scanned by TapScanner

3-D Cuboid Representation of Data.



Lattice of cuboids making up a 4-D data cube for time, item, location and supplier is given below.



The above cuboid generates subsets of the given dimensions The result would form a lattice of cuboids each

Scanned by TapScanner

Showing the data at a different level of summarization, or group-by. The lattice of cuboid is referred as a data cube.

The cuboid that holds the lowest level of summarization is called the base cuboid

The 0-D cuboid which holds the highest level of summarization called the apex cuboid

Schemas For Multidimensional Data Models

Data Model for a datawarehouse is a multidimensional model, which exists in the following forms.

i) Star schema
ii) Snowflake schema
iii) fact Constellation schema.

Star Schema

The star schema contains
1) large central table (fact table) containing the bulk of data with no redundancy.
2) A set of smaller attendant tables (dimension tables), one for each dimension

The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table. This constraint may introduce some redundancy

The following figure is the schema contains a central fact table for sales that contains keys to each of the four dimensions namely time, item, branch and location

Scanned by TapScanner

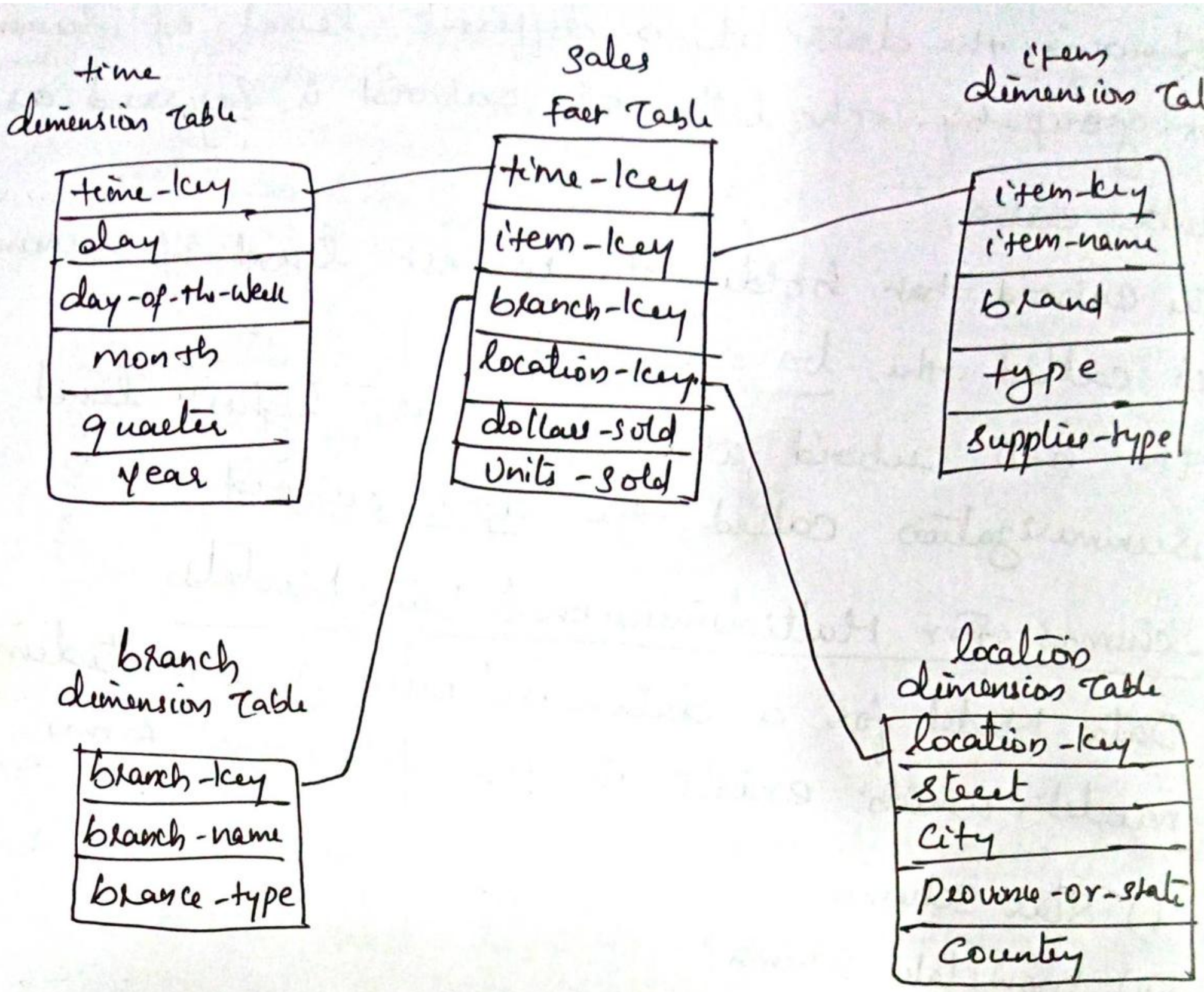


fig: star schema of a datawarehouse for sales

Snowflake schema :-

The Snowflake Schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graphs forms a shape similar to a snowflake.

The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space.

Scanned by TapScanner

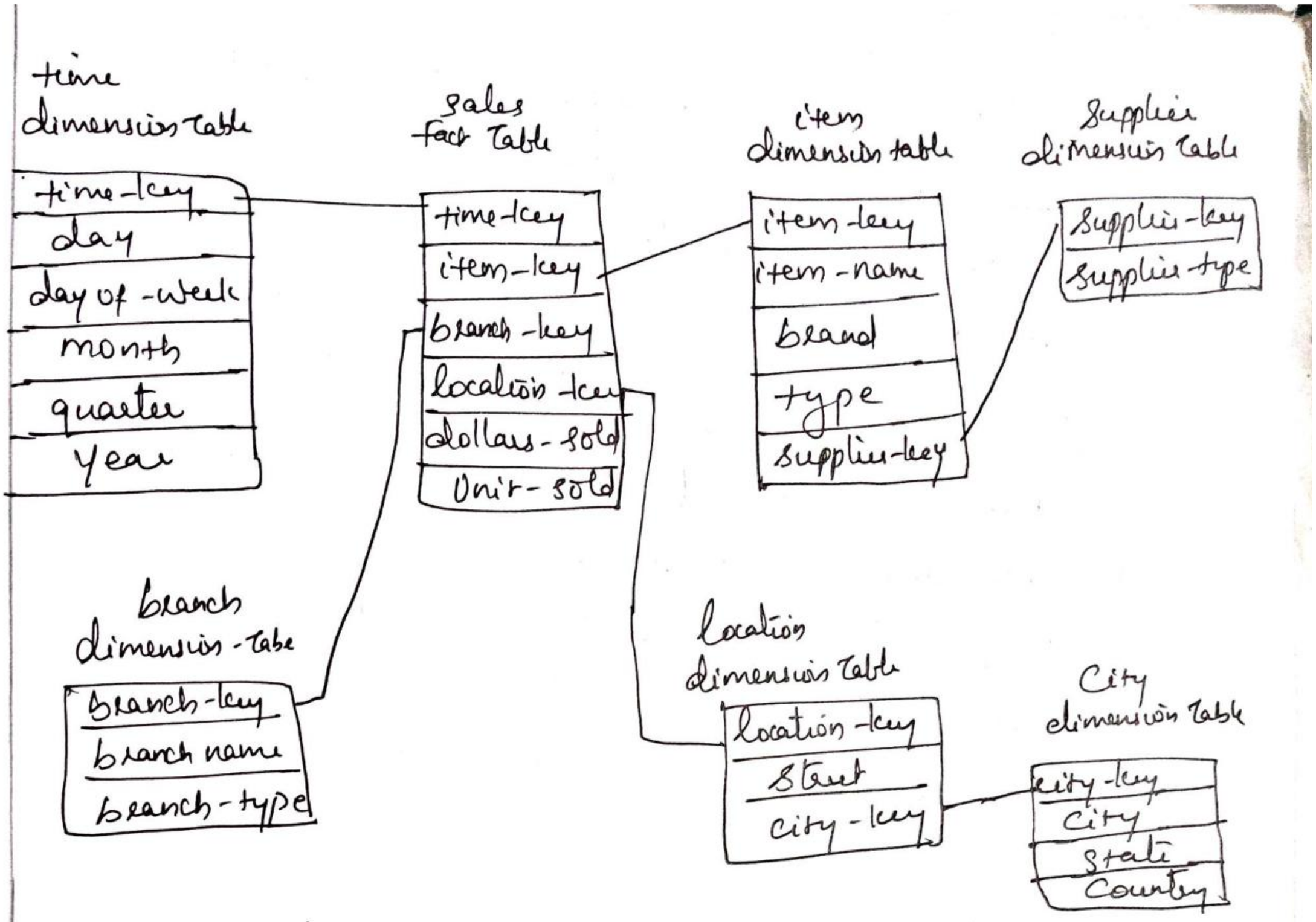
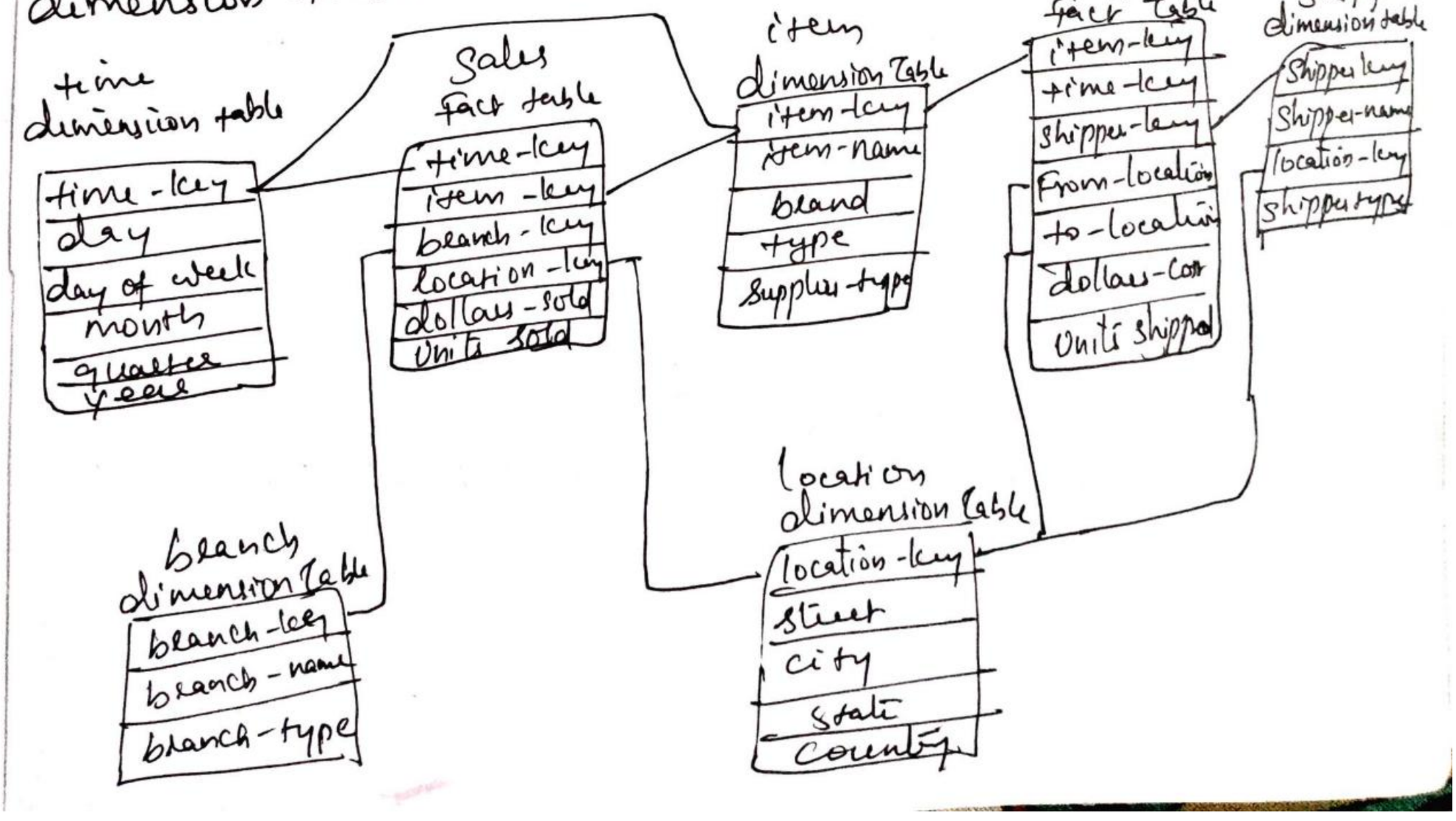


fig: Snowflake schema of datawarehouse for sales

Fact Constellation :-

A fact constellation schema is shown in the following figure. this schema specifies two fact tables, sales and shipping. A fact constellation schema allows dimension table to be shared between fact table.



Scanned by TapScanner

DataMart:- It is a department subset of the datawarehouses that focuses on selected subjects and thus its scope is department wide. Star and snowflake models are widely used for datamart

OLAP operations in Multidimensional Data Model

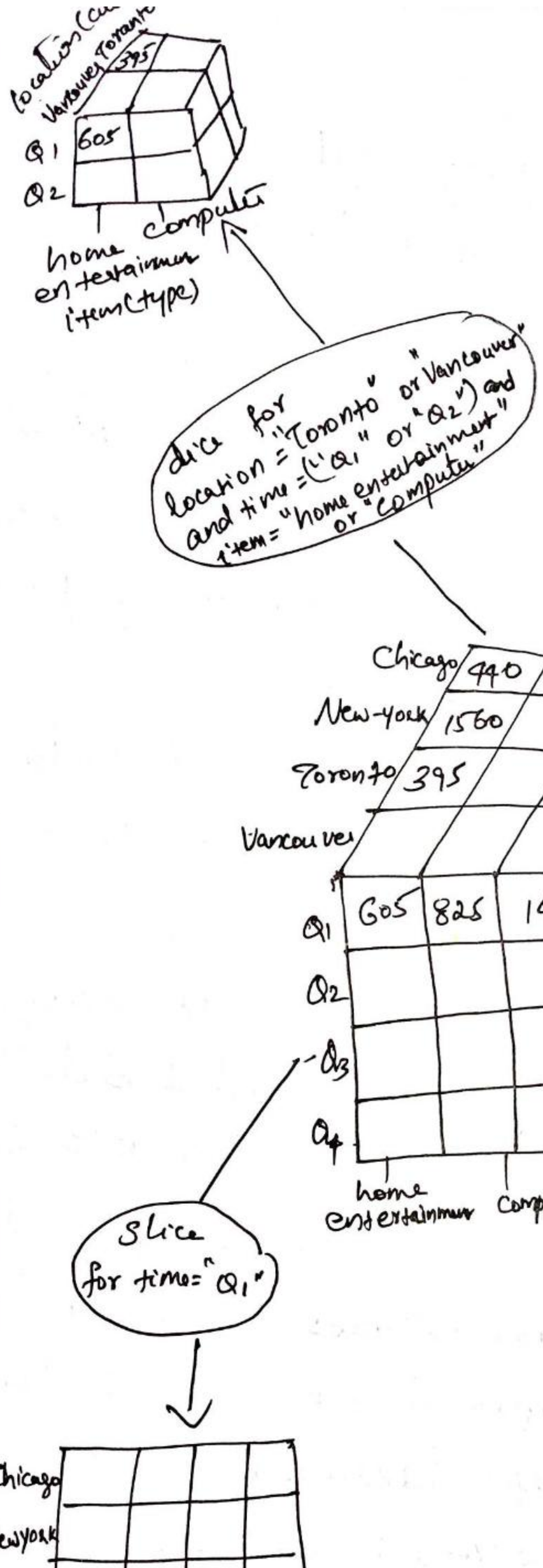
In a multidimensional model data are organized into multiple dimension and each dimension contains multiple levels of abstraction defined by concept hierarchis. This organization provides users with the flexibility to view data from different perspectives. The different OLAP operations are

→ Roll up
→ Drill-down
→ Slice and dice
→ pivot (Rotate)

Roll-up

The Roll-up operations performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension Reduction.
The following figure shows the result of a Roll-up operation performed on the central cube by climbing up the concept hierarchy.
When Roll-up is performed by dimension Reduction, one or more dimensions are removed from the given cube.

Scanned by TapScanner



Scanned by TapScanner

Roll-up

The Roll-up operation performs aggregation on a data cube either by climbing up a concept hierarchy for a dimension or by dimension Reduction.

The over figure shows the Roll-up operation performed on a central cube by climbing up the concept of hierarchy for the location. The Roll-up operation shown aggregates the data by ascending the location hierarchy from the level of city to the level of country.

When Roll-up is performed by dimension Reduction, one or more dimensions are removed from the given cube.

Drill-down

Drill-down is the reverse of Roll-up. It navigates from less detailed data to more detailed data. Drill-down can be Realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions. The above figure shows the Result of a drill down operation performed on the central cube by stepping down a concept hierarchy for time defined as day < month < quarter < year. Drill down occurs by descending the time hierarchy from level of quarter to the more detailed level of month.

Slice and Dice

The Slice operation performs a selection on one dimension of the given cube, Resulting in a subcube. The figure shows Slice operation where the Sales data

Scanned by TapScanner

are selected from the central cube for the dimension time using the criterion time = "Q1"

The dice operation defines a subcube by performing a selection on two or more dimensions. The figure shows a dice operation on the central cube based on the criteria that involve location, time & item dimension.

Pivot (Rotate)

Pivot is a visualization operation that Rotates the data axes in view in order to provide an alternative representation of data. The figure shows a Pivot operation where the item and location axes in a 2-D slice are Rotated

Other OLAP operation

Drill across - executes queries involving more than one fact table

Drill - Through - operation uses Relational SQL facilities to drill through the bottom level of data cube down to its back-end Relational tables.

OLTP Vs DW

Datawarehouse (OLAP)	operational Database (OLTP)
1. It involves historical processing of information	It involves day-to day processing
2. OLAP Systems are used by knowledge workers such as executives, managers & analysts	OLTP systems are used by clerks, DBA's or database professionals.

Scanned by TapScanner

	Datawarehouse	OLTP
3.	It is used to analyze the business	It is used to Run business
4.	It focus on Information out	It focus on Data in
5-	It is based on star schema, Snowflake schema, and fact Constellation schema	It is based on Entity Relationship model
6.	It focus on historical data	It contains current data
7-	It provides summarized and Consolidated Data.	It provides primitive and highly detailed data
8.	The number of user is in hundreds.	The number of users is in thousands.
9.	The number of records accessed is in millions	The number of records accessed is in ten.
10.	The database size is from 100GB to 100 TB	The database size is from 100 MB to 100GB
11.	There are highly flexible	It provides high performance.

Scanned by TapScanner

Question banks

1. Give the application of data mining and data warehousing
2. Differentiate between OLTP & OLAP.
3. Describe the needs for Data warehousing
4. With the help of a neat diagram explain briefly about data mining models.
5. Describe about OLAP
6. Give the application of DWH.
7. Explain about OLAP operation with the help of example

Scanned by TapScanner

MOD - II

Data Preprocessing

Real world databases are highly susceptible to noisy, missing and inconsistent data due to their huge size. Several preprocessing techniques are used.

1) Data cleaning – Removes noise and correct inconsistencies of data
2) Data Integration – merges data from multiple sources into a coherent data store such as a data warehouse
3) Data Transformation – This is done by normalization which improves the accuracy and efficiency of mining algorithms.
4) Data Reduction – Reduce the data size by aggregating, eliminating redundant features.

Data preprocessing Concept

The datas used for datamining must possess certain qualities or it should satisfies the requirements of user needs.

Datamining dataset should maintain certain qualities. The factors that determine quality of datas are

1) Accuracy
2) completness
3) Consistency
4) Timeliness
5) Believability
6) Interpretability

Scanned by TapScanner

The dataset for datamining may contains

i) incomplete data - lack attribute values

ii) noisy data - contain error data or outlier values that deviate from the expected

iii) inconsistent - Contain discrepancies in the department code used to categorize items.

Reasons for incomplete datas are:-

→ Attribute information may not be available such as customer information for sale transaction data.

→ Datas may not be considered as important at the time of data entry

→ Relevant datas may not be entered due to misunderstanding or due to equipment malfunction

→ Datas that are inconsistent with other records are deleted

→ Missing data for tuples with missing values for some attributes may need to be inferred.

Reasons for noisy data :-

→ Fault of data collection instruments

→ Human or computer error during data entry

→ Errors in the data transmission

→ Limited buffer size for coordinate synchronized data transfer and consumption

→ Inconsistent format of input fields eg: date attribute fie

Scanned by TapScanner

Tasks in data preprocessing

There are mainly four task in data preprocessing

1) Data cleaning
2) Data Integration
3) Data Transformation
4) Data Reduction

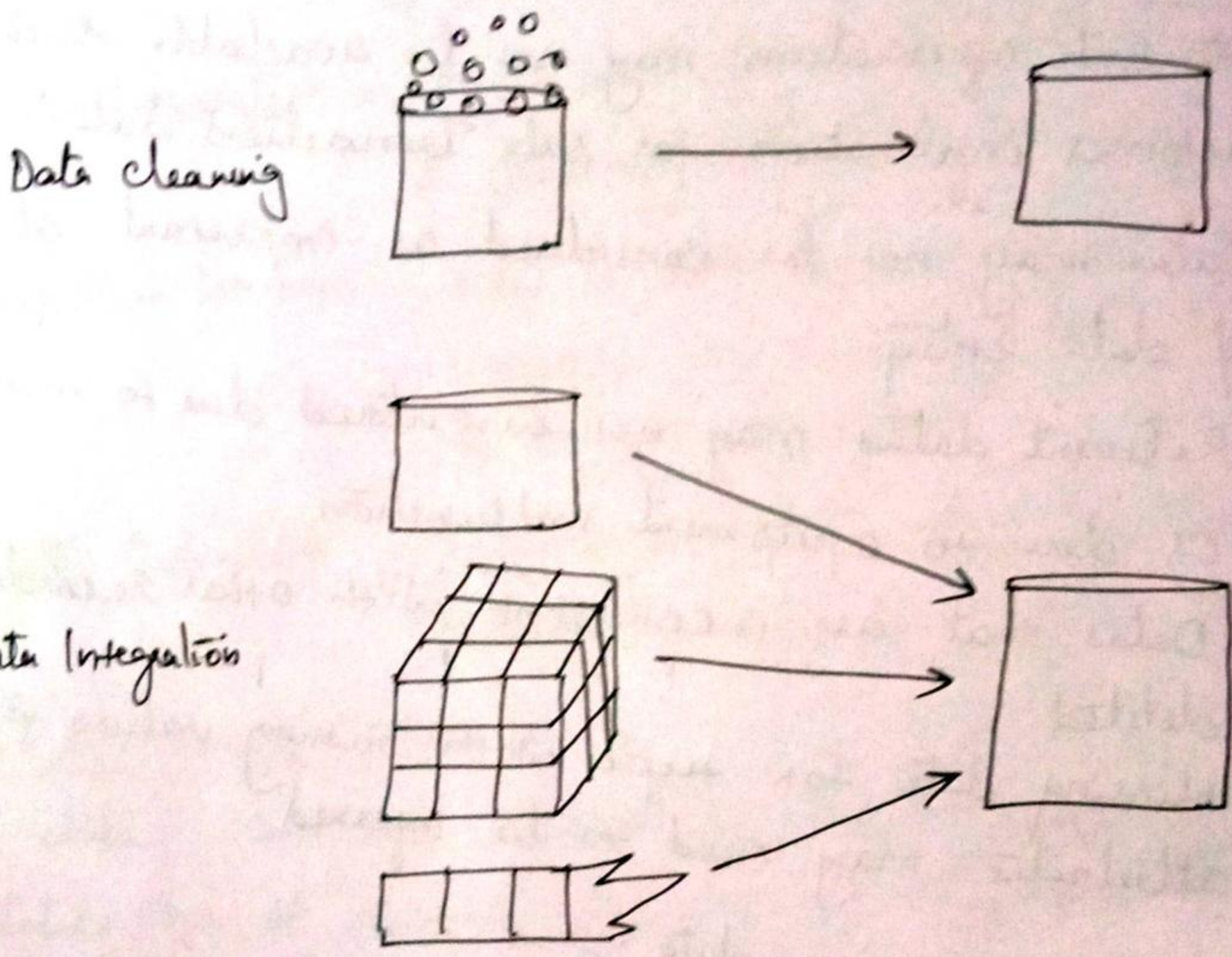


fig: Forms of data preprocessing

Scanned by TapScanner

Data cleaning

Data cleaning 'clean' the data by filling missing values, smoothing noisy data, identifying or removing outliers and resolving inconsistencies.

The dirty dataset results in unreliable datamining output and it cause confusion in datamining

Data Integration

This process integrate multiple sources like database, datacubes and files for preprocessing. while integrating the datas from different source there can be a chance of inconsistency and redundancy. The redundancy can be avoid by the further data cleaning.

Data cleaning and Data integration are performed as a preprocessing step when preparing data for a data warehouse.

Data Transformation

Data transformation is performed by normalization and aggregation and by concept hierarchy.

Data Reduction

Data reduction obtained a reduced representation of the data set that is much smaller in volume and produces the same analytical results. The main strategies used for data reduction are

Data aggregation — building a datacube

Scanned by TapScanner

attribute subset selection - Removing irrevalent attributes through correlation analysis

dimensionality Reduction - Using encoding schemes such as minimum length encoding or Wavelets

numerosity Reduction - Replacing the data by alternative smaller representations such as clusters or parametric models.

Data cleaning

Real World data tend to be incomplete, noisy and inconsistent.

Data cleaning ~~routine~~ Routines attempt to fill in missing values, smooth out noise while identifying outliers and correct inconsistencies in the data.

Basic methods of cleaning the data are as follows.

Missing Values

The missing values in the different attributes can be done by the following methods:-

1. Ignore the tuple:- This is done usually when class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values

2. Filling in the missing value manually:- This approach is time Consuming and may not be feasible for the large data set with missing values.

Scanned by TapScanner

3. Use a global constant to fill in the missing value:- Replace all missing attribute value by the same constant such as "Unknown" or "∝"

4 Use the attribute mean to fill in the missing value - Use average mean value to fill the missing attribute

5. Use the attribute mean for all samples belonging to the same class as the given tuple -

6. Use the most probable value to fill in the missing value:- This may be determined with Regression, Inference - based tools using decision tree induction.

Noisy Data

Noise is a random error or variance in a measured variable

Following are the different data smoothing techniques used to remove noise from the dataset.

Binning :-

Binning method smooth a sorted data value by consulting its neighbourhood, ie value around it - The ~~sho~~. sorted values are distributed into a number of buckets or bins. Binning method peroform local smoothing since it considers the neighbourhood values

Smoothing by bin means - Each value in the bin is replaced by the mean value of the bin

Smoothing by bin ~~medians~~ boundaries - Each bin value is replaced by the closest boundary values.

Smoothing by bin medians - Each bin value is replaced by bin medians

Scanned by TapScanner

In general, larger the width of sorted data set, the greater effect for the smoothing

Example binning...

Sorted data for price : 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

Smoothing by bin means

Bin 1: 9, 9, 9

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

Smoothing by bin boundaries

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

Bin 3: 25, 25, 34

fig: Binning method for data smoothing

Regression

Data can be smoothed by fitting the data to a function such as with regression

Linear Regression - involves finding the "best" line to fit two attributes, so that one attribute can be used to predict the other

Multiple Linear Regression - It is an extension of linear regression, where more than two attributes are involved

Scanned by TapScanner

and the data are fit to a multidimensional surface

clustering:

outliers may be detected by clustering, where the similar values are organized into groups or clusters. values that fall outside of the set of clusters may be considered as outliers

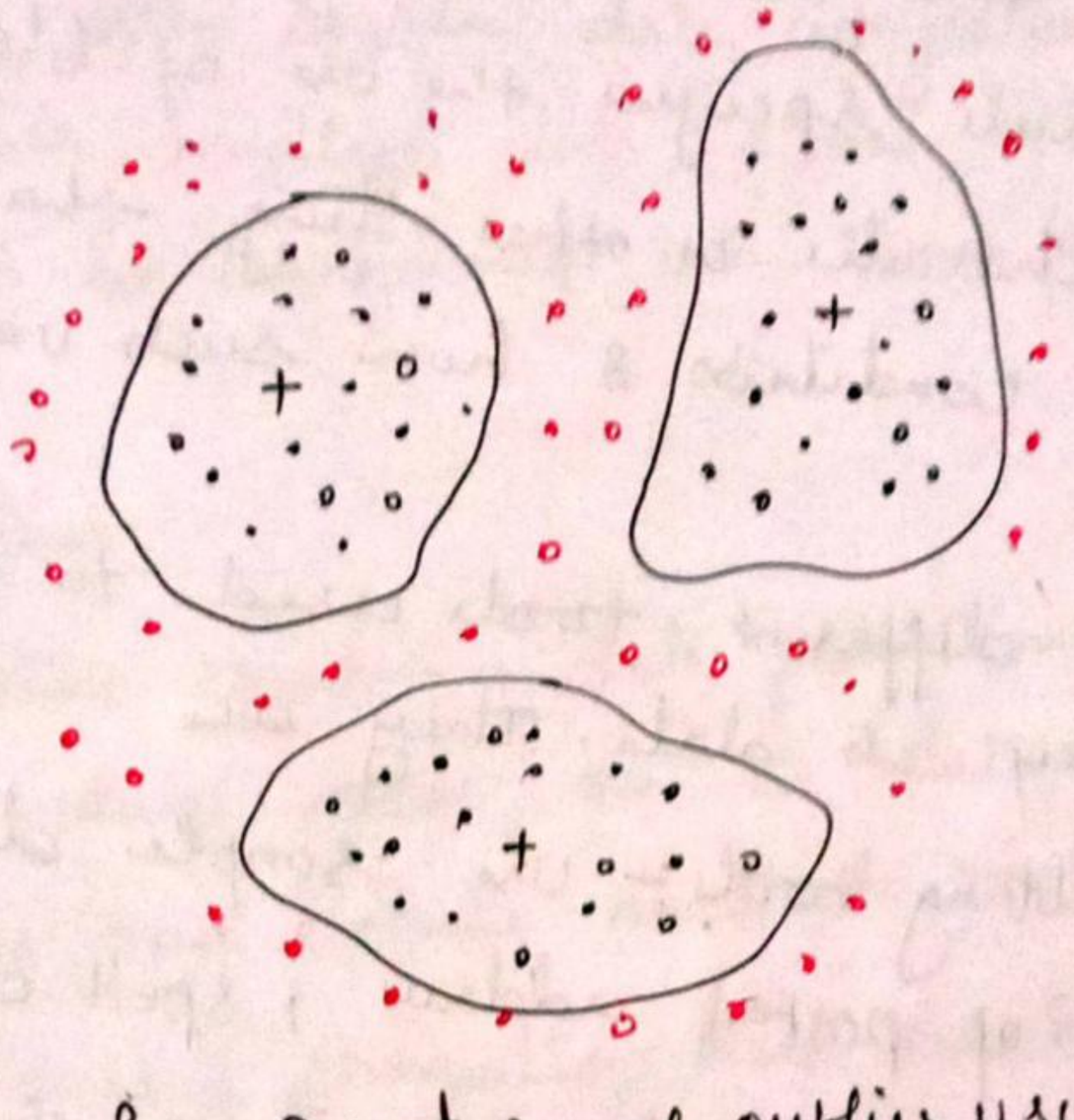


fig: Detection of outlier using clustering

Data cleaning as a process

Missing values, noise and inconsistencies contribute to inaccurate data.

The first step in data cleaning as a process is discrepancy detection. Discrepancies can be caused by several factors including poorly designed data entry forms that have many optional field, human error in data entry, deliberate errors, data decay and inconsistent use of data.

The data should be examined by unique rule, consecutive rule and null rule.

Scanned by TapScanner

A Unique Rule says that each value of the given attribute must be different from all other values for that attribute.

A consecutive Rule says that there can be no missing values between the lowest and highest values for the attributes and that all values must also be unique

A null Rule Specifies the use of blanks, question mark Special character or other string that may indicate the null condition & how such values should be handled

There are different tools used to found the discrepency in data. They are

Data Scrubbing tools:- use simple domain knowledge (knowledge of postal address, spell checking etc) to detect errors & make correction in the data.

Data auditing tools:- Find discrepency by analyzing the data to discover Rules and Relationships and detecting data that violates such conditions

Most of the errors requires data transformation. Commercial tools can assist in the transformation step are Data migration tools- which allow simple transformation such as replace the string 'gender' by 'sex'. and

ETL (Extraction/transformation/Loading tools allow user to Specify transforms through a graphical user interface (GUI)

Another approach to increased interactivity in data cleaning is the development of declarative languages for the specifi

Scanned by TapScanner

of data transformation operators. Such work focuses on defining powerful extension of SQL & algorithms that enable user to express data cleaning, specifications efficiently.

Data Integration and Transformation

Data mining often requires data integration - the merging of data from multiple data stores. The data may also need to be transformed into forms appropriate for mining

Data Integration

Data Integration combines data from multiple sources into a coherent data store as in data warehousing. These sources may include multiple databases, datacubes and files. Careful integration help to reduce and avoid redundancies and inconsistencies in the resulting dataset. This helps to improve the speed and accuracy of the subsequent datamining process.

Entity identification problem in Data Integration

There are certain issues in the data integration. One example of entity identification problem is that attribute 'customer-id' and 'cust-number' ≠ ~~attribute~~ present in different database possess same value for the same tuple. This should be solved during integration.

Metadata helps to avoid the errors in Schema integration

During integration it is to ensure that any attribute functional dependencies and referential constraints in the source system match those in the target system.

Redundancy & Correlation Analysis

Redundancy is another important issue in data integration. An attribute may be redundant if it can be derived from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

Some redundancies can be detected by correlation analysis. Given, two attribute correlation analysis can measure how strongly one attribute implies the other, based on available data.

For numerical attributes, the correlation b/w two attributes A and B are evaluated by computing the correlation coefficient (pearson's product moment coefficient) can be given as.

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \bar{A})(b_i - \bar{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i b_i) - N\bar{A}\bar{B}}{N\sigma_A\sigma_B}$$

→ Covariance

where N is the number of tuples a_i and b_i are the respective values of A and B in the tuple i, $\bar{A}$ and $\bar{B}$ are the respective mean value of A and B, σ_A and σ_B are the respective standard deviations of A and

Scanned by TapScanner

$\Sigma(a_i b_i)$ is the sum of AB cross product (i.e each value of A is multiplied by the value of B in that tuple.

Conditions for two attributes correlated is if.

$r_{A,B}$ lies b/w -1 & +1 i.e $-1 \leq r_{A,B} \leq +1$. then $r_{A,B}$ is not correlated & both are independent. But

If $r_{A,B}$ is greater than 0, then A & B are positively correlated, meaning the value of A increases as the value of B increased. Higher the value stronger the correlation

If the resulting value is 0, then A & B are independent

If the resulting value is less than 0, then A & B are negatively correlated, where the value of one attribute increases and other decreases.

For categorical (discrete) data, a correlation relationship between two attributes A and B found by a χ^2 test

Suppose A has c distinct values $a_1, a_2 \ldots a_c$ and B has r distinct values $b_1, b_2, \ldots b_r$. The χ^2 value can be computed by

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(O_{ij} - e_{ij})^2}{e_{ij}}$$

where O_{ij} is observed frequency i.e (actual count) of the joint event (A_i, B_j) and e_{ij} is the expected frequency

Scanned by TapScanner

of (A_i, B_j) which can be computed as

$$e_{ij} = \frac{\text{Count}(A = a_i) \times \text{count}(B = b_j)}{n}$$

where 'n' is the no: of data tuples

Count $(A = a_i)$ is the no: of tuples having value a_i for A

Count $(B = b_j)$ is the no: of tuples having value b_j for B

The sum is computed over all of the $r \times c$ cells.

The cells that contribute the most to the χ^2 value are those for which the actual count is very different from that expected. The test is based on $(r-1)(c-1)$ degree of freedom.

Example of correlation analysis of categorical attributes

Using χ^2

	male	female	Total
fiction	250(90)	200(360)	450
non-fiction	50(210)	1000(840)	1050
Total	300	1200	1500

Suppose a group of 1500 are surveyed and was polled against their preferred type of reading material fiction and non-fiction.

Expected frequency for the cell (male, fiction)

$$e_{11} = \frac{\text{Count(male)} \times \text{count(fiction)}}{N}$$

$$= \frac{300 \times 450}{1500} = 90$$

Scanned by TapScanner

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= 284.44 + 121.90 + 71.11 + 30.48 = \underline{507.93}$$

In the example a 2×2 table is used so the degree of freedom is $(2-1)(2-1) = 1$

for 1 degree of freedom the χ^2 distribution is 10.828 (from Statistics of χ^2 distribution)

Our evaluated value is 507.93 which is larger than the Significance level 10.828. so the two attributes gender & preferred-reading are independend and are strongly correlated.

Example of Covariance analysis of numeric attribute

Time point	Allelectro	HighTech
t_1	6	20
t_2	5	10
t_3	4	14
t_4	3	5
t_5	2	5

We have to find the covariance of two companies Allelectro and HighTech price rate

E(Allectro) = $\frac{6+5+4+3+2}{5} = \frac{20}{5} = 4$

Scanned by TapScanner

$$E(\text{High Tech}) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = 10.80$$

According to covariance equation,

$$Cov(A,B) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{N} \quad \text{OR} \quad Cov(A,B) = \frac{E(A \cdot B) - \bar{A}\bar{B}}{N}$$

$$Cov(\text{Allectro}, \text{High Tech}) = \frac{6\times20 + 5\times10 + 4\times14 + 3\times5 + 2\times5}{5} - 9\times8$$

$$= 50.2 - 43.2 = 7$$

For two attributes A and B, that tend to change together if A is larger than $\bar{A}$ (is expected value) then B is likely to be larger than $\bar{B}$ and the value of covariance between A & B is positive

Covariance b/w A and B is negative if one value of attribute is above the value of expected value and otherone is below the value of expected value.

If Covariance is 0. then A & B are independent and assumed to be not correlated to each other. But this case is not always true. Only under some additional assumptions only it will work.

Tuple duplication

Tuple duplication is also an issue in integration it should be detected at the tuple level. when two or more identical tuples for a ~~genuis~~ unique data entry case is the tuple duplication. The use of denormalized

Scanned by TapScanner

table is another source of data redundancy. Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences

Data Value Conflict Detection & Resolution

Data integration involves the detection and resolution of data value conflicts. eg: price of the hotel rooms in different cities are stored in different currencies, different services and taxes.

Data Transformation

In data transformation data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following.

- Smoothing - which works to remove noise from the data. Such techniques involves, binning, regression & clustering
- Aggregation - summary or aggregation operations are applied to the data to compute monthly and annual total amounts. This step is typically used in constructing data cube for analysis of the data at multiple granularities
- Generalization - where the low level or raw data are replaced higher level concepts through the use of concept hierarchies. eg: street can be generalized to higher level concept city or country.

Scanned by TapScanner

* Normalization - where the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0 or 0.0 to 1.0

* Attribute Construction - where new attributes are constructed and added from the given set of attributes to help the mining process.

An attribute is normalized by scaling its values, so that they fall within a small specified range such as 0.0 to 1.0. Normalization is useful for classification algorithm involving neural networks.

There are many methods for data normalization. They are

i) min-max normalization

ii) Z-score normalization

iii) normalization by decimal scaling

Min-Max Normalization performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum value of an attribute the Min-Max normalization maps a value v_i of A to v' in the range $[new\text{-}min_A, new\text{-}max_A]$ by computing

$$v' = \frac{v - min_A}{max_A - min_A}((new\text{-}max_A) - (new\text{-}min_A)) + new\text{-}min_A$$

Min-Max normalization preserves the relationships among the original data values.

Scanned by TapScanner

eg: Suppose the minimum & maximum value for the attribute income are $12,000 & $98,000. We have to map the income to the range [0.0, 1.0].

By min-max normalization a value of $73,600 for income is transformed to $\frac{73,600 - 12,000}{98,000 - 12,000}(1.0 - 0) + 0 = 0.716$

Z-Score Normalization (zero-mean normalization) - the value of an attribute A, are normalized based on the mean ~~of~~ and standard deviation of A.

A value v, of A is normalized to v' by computing

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

where $\bar{A}$ and σ_A are the mean and standard deviation of attribute A. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown or there are outliers that dominate the min-max normalization.

Eg: Suppose that the mean and standard deviation of the values for the attribute income are $54,000 and $16,000. With z-score normalization a value of $73,600 for income is transformed to $\frac{73,600 - 54000}{16,000} = 1.225$

Decimal Scaling Normalization - normalizes by moving the decimal point of values of attribute A. The number of decimal point moved depends on maximum absolute

Scanned by TapScanner

value of A. A value v_i of A is normalized to v' by computing $v' = \frac{v}{10^j}$ where 'j' is the smallest integer such that $Max(|v'|) < 1$

eg: Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, we divide each value by 1000 (i.e j=3) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.

Data Reduction

Data Reduction Techniques are applied to obtain a reduced representation of the data set that is much smaller in volume and closely maintains the integrity of the original data. Mining on the reduced data set should be more efficient and produce the same analytical result. Different Strategies for the data Reduction are as follows.

1. Data cube aggregation - aggregation operations are applied to the data in the construction of a data cube.
2. Attribute Subset Selection - ~~irrelevant~~ irrelevant, Weakly relevant, or redundant attributes or dimensions may be detected or removed.
3. Dimensionality Reduction - where encoding mechanisms are used to reduce the dataset size.
4. Numerosity Reduction - where the data are replaced or estimated by alternative smaller data representation

Scanned by TapScanner

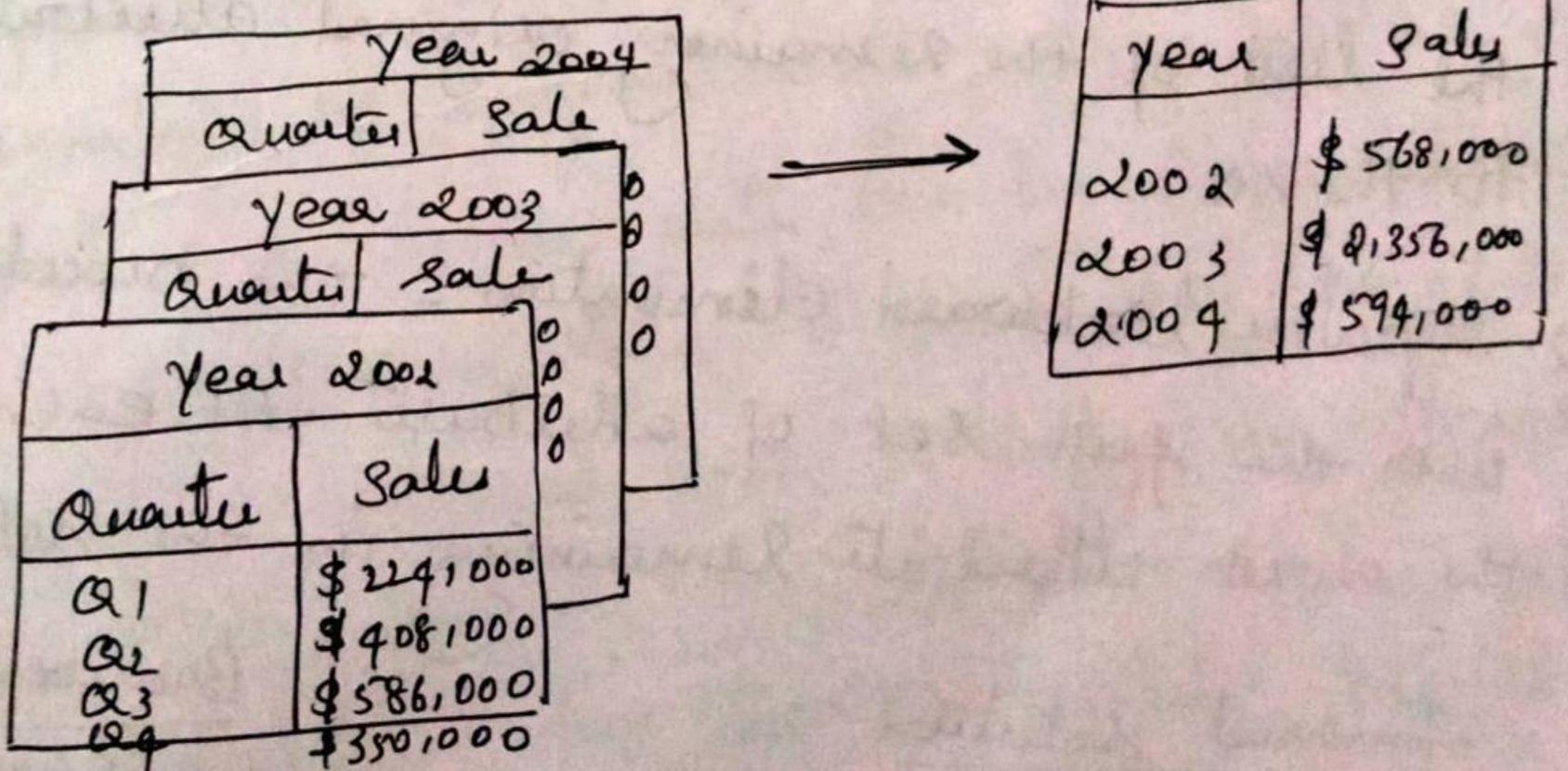
Such as parametric models or non parametric methods such as clustering, sampling and the use of histograms.

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

Data Cube Aggregation

The data is aggregates from the data set and the resulting data is smaller in volume, without loss of information necessary for the analysis task.

Example:-



Attribute Subset Selection

Attribute Subset Selection reduces the data set size by removing irrelevant or redundant attributes. The goal of attribute subset selection is to find a minimum set of

Scanned by TapScanner

attributes such that the resulting probability distribution of the data classes is as close as possible to the orig distribution obtained using all attributes.

Mining on a reduced set of attributes reduces the no. of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

Heuristic methods are used for the subset selection. The methods are greedy, they always makes the best choice at the time

Basic heuristic methods of attribute subset selection are as follows:-

1. Stepwise forward selection - The procedure starts with an empty set of attributes as reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step the best of the remaining original attributes is added to the set.

2. Stepwise backward elimination - The procedure starts with the full set of attributes. At each step it remov the worst attribute remaining in the set.

Forward selection

Initial attribute set:
$\{A_1, A_2, A_3, A_4, A_5, A_6\}$
Initial Reduced set
$\{\}$
$\Rightarrow \{A_1\}$
$\Rightarrow \{A_1, A_4\}$
$\rightarrow$ Reduced attribute set
$\{A_1, A_4, A_6\}$

Backward elimination

Initial attribute set
$\{A_1, A_2, A_3, A_4, A_5, A_6\}$
$\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$
$\Rightarrow \{A_1, A_4, A_5, A_6\}$
$\rightarrow$ Reduced attribute set
$\{A_1, A_4, A_6\}$

Scanned by TapScanner

3. Combination of forward selection & backward elimination - The stepwise forward selection and backward elimination methods can be combined so that, at each step the procedure selects the best attribute and removes the worst from among the remaining attributes.

4. Decision tree induction : Decision tree induction constructs a flowchart like structure where each internal node denotes a test on an attribute each branch corresponds to an outcome of the test each leaf node denotes a class prediction. when decision tree induction is used for attribute subset selection, a tree is constructed from the given data. The attributes which do not appear on the tree is considered as irrelevant.

Dimensionality Reduction

In dimensionality Reduction data encoding or transformations are applied to obtain a reduced or compressed representation of original data. They are of 2 types

lossless - If the original data can be reconstructed from compressed data without any loss of information

lossy - If the original data is only an approximation then it is lossy reduction

Two methods used for dimensionality reduction are

i) Wavelet transforms

ii) principal component analysis

Scanned by TapScanner

Wavelet Transforms

The discrete Wavelet transforms (DWT) is a linear signal processing technique that, when applied to a data vector x, transforms it to a numerically different vector x'. of wavelet coefficients. The two vectors are of same length.

Wavelet transforms can be truncated (make shorter) this technique also works to remove noise without smoothing out the main features of the data, making it effective for data cleaning

DWT is closely related to the discrete Fourier transform a signal processing technique involves sines & cosines.

The procedure for applying a discrete wave transform uses a hierarchical pyramid algorithm that halves the data at each iteration resulting in fast computational speed. The method is as follows

1. The length L, of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary.
2. Each transform involves applying two functions. The first applies some data smoothing, such as sum or weighted average. The second performs a weighted difference which acts to bring out the detailed features of the data.

Scanned by TapScanner

3. The two function are applied to pairs of data points in X_j, that is to all pairs of measurements (x_{2i}, x_{2i+1}). This Results in a set of data of length $L/2$.

4. The two function is ~~Recursively~~ Recursively applied to the set of data obtained in the previous loop, until the resulting data set obtained for length 2.

5. Selected values from the data sets obtained in the above iterations are designated the wavelet coefficients of the transformed data.

Principal Components Analysis (PCA)

Principal component analysis Searches for k n-dimensional orthogonal vectors that can be used to represent the data, where $K \leq n$. The original data are projected to smaller space, resulting in dimensionality Reduction. PCA combines the essence of attributes by creating an alternative, Smaller set of variables. The initial data can then be projected onto this Smaller set.

The procedure of PCA is as follows.

1. The input data are normalized, so that each attribute falls within the same range. This Step helps to ensure that attributes with large domains will not dominate attributes with Smaller domains

2. PCA computes k-orthogonal normal vectors that ~~each~~ provides a basis for the normalized input data. These are unit vectors that each point is a direction 1^{st} to the others. The vectors are referred as the principal components. The i/p

Scanned by TapScanner

data are a linear combination of the principal compon

3. The principal components are sorted in order of decreasing "significance" or strength. The principal compon essentially serve as a new set of axes for the dat providing important information about variance. Sorted axes are in the form such that first axis show the most variance among the data, the second axis shows the next higher variance and so on.

4. Since the components are sorted according to the decreasing order of significance the size of the data can be reduced by eliminating the weaker components that is those with low variance

Numerosity Reduction

Numerosity Reduction can be applied to reduce the datavolum by choosing alternative smaller forms of data representation there are 2 method

i) parametric
ii) nonparametric

parametric method is used to estimate the data so that only the data parameters need to be stored instead of the actual data. Example Log-linear Model.

Non parametric methods for storing reduced representation of the data including histogram, clustering & sampling

i) Regression and Log-linear Model.

Regression and log linear models can be used to approximate the given data. In linear regression, the

Scanned by TapScanner

data are modeled to fit a straight line. eg: a random (response) variable 'y' can be modeled as a linear function of another random variable x, (predictor variable) with the equation

$$y = wx + b$$

where variance of y is constant.

x & y are numerical database attributes

w & b (Regression coefficients) specify the slop of line & y-intercept

Multiple linear Regression is an extension of linear Regression, which allows a response variable 'y' to be modeled as a linear function of two or more predictor variables.

Log-linear Models can be used to estimate the probability of each tuple in an multidimensional space for a set of discretized attributes based on a smaller subset of dimensional combinations. This allows higher dimensional data space to be constructed from lower dimensional space. Log-linear models are useful for dimensionality reduction & data smoothing

Histogram

Histograms use binning to approximate data distributions and are a popular form of data reduction. A histogram for an attribute A, partitions the data distribution of A into disjoint subsets or buckets. If each bucket represents only a single attribute-value/frequency pair the buckets are Singleton buckets. Generally buckets represent continuous range for the given attribute.

Scanned by TapScanner

Example:-

The following data are a list of prices of commonly sold items of a company. The numbers are sorted as

1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 21, 21, 21, 25, 25, 25, 25, 28, 28, 30, 30, 30

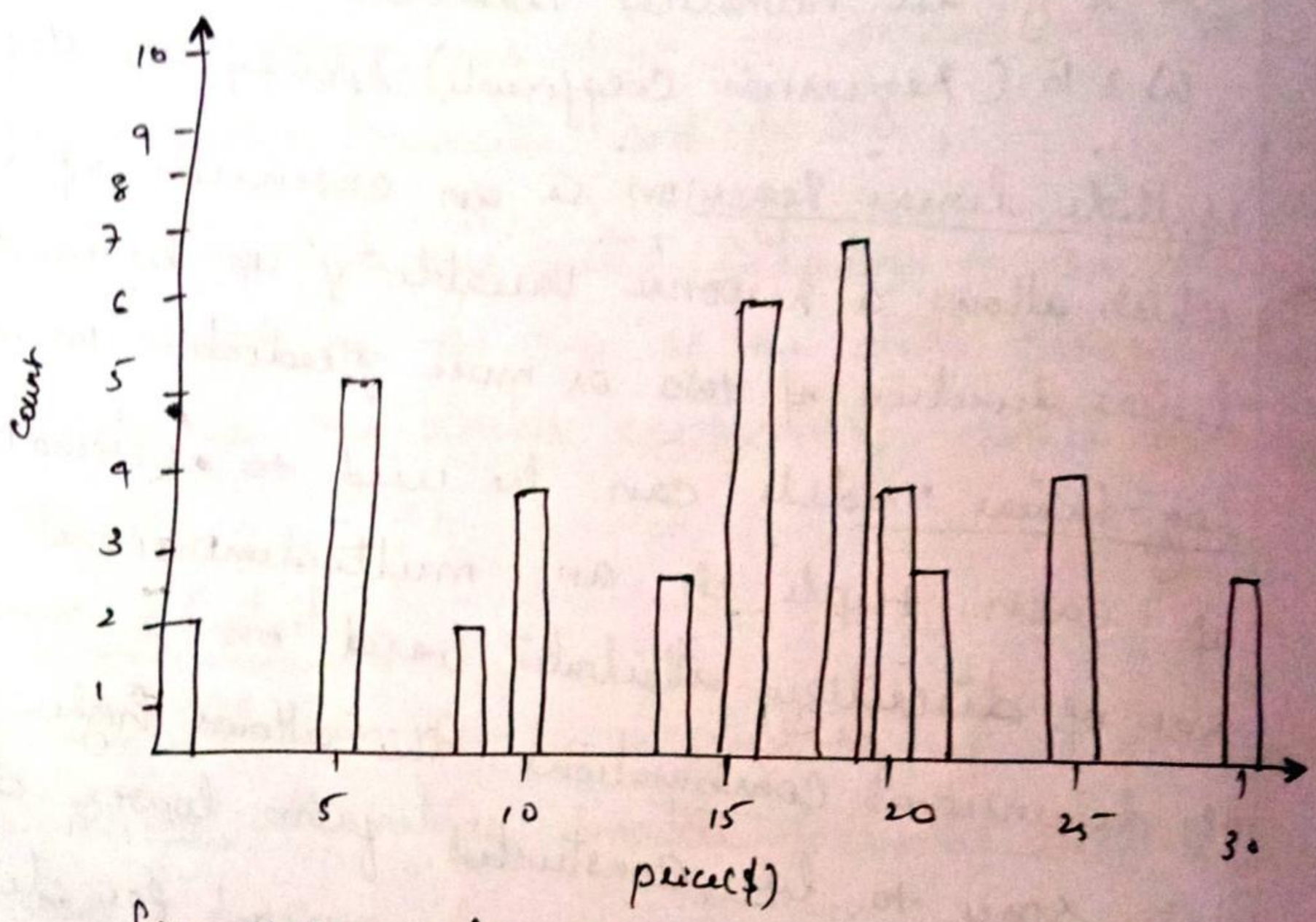
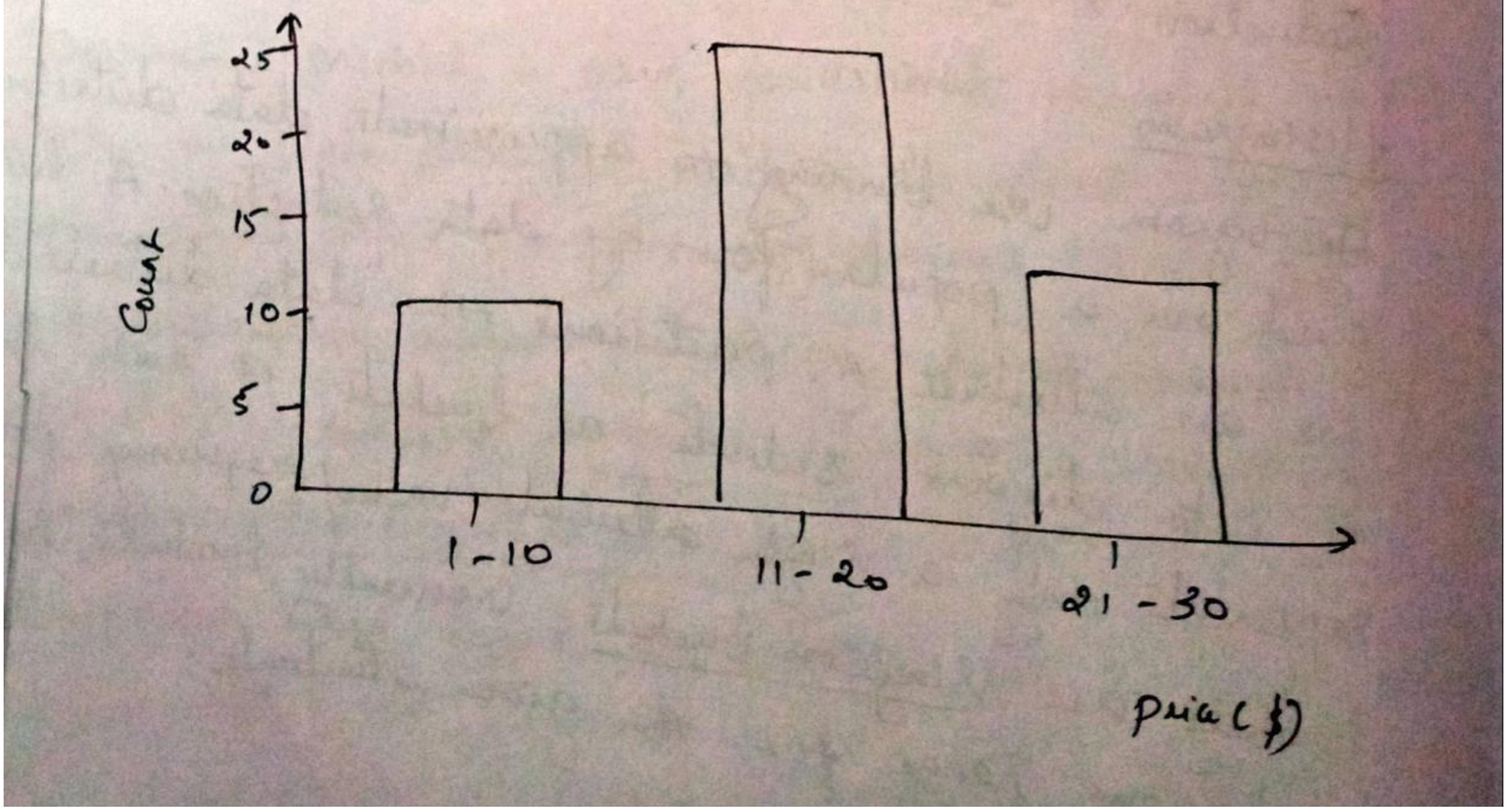


fig: Histogram for price using singleton-buckets - each bucket represents one price-value/frequency pair



Scanned by TapScanner

there are certain partitioning rule for the histogram

* Equal-width - In an equal-width histogram, the width of each bucket range is uniform

* Equal-frequency - In an equal-frequency histogram, the buckets are created so that, roughly the frequency of each bucket is constant.

* V-optimal - V-optimal histogram is the one with the least variance. Histogram variance a the weighted sum of original values that the bucket represents, where the bucket weight is equal to the number of values in the bucket.

* MaxDiff - In a MaxDiff histogram, a bucket boundary is established between each pair for pairs having the $\beta - 1$ largest differences, where β is the user-specified number of buckets.

Clustering

Clustering techniques consider data tuples as objects. They partition the objects into groups or clusters, so that objects within a cluster are similar to one another and dissimilar to objects in other clusters.

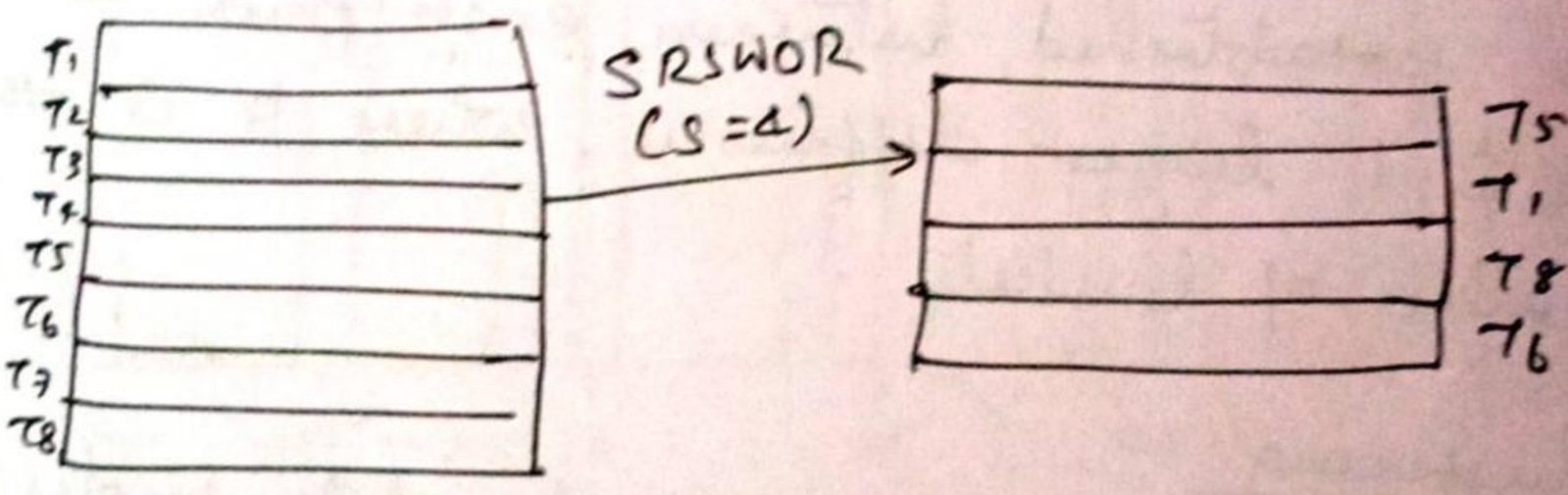
In data reduction, the cluster representations of the data are used to replace the actual data. The effectiveness of this technique depends on the nature of the data. It is much more effective for data that can be organized into distinct clusters than for smeared data.

Scanned by TapScanner

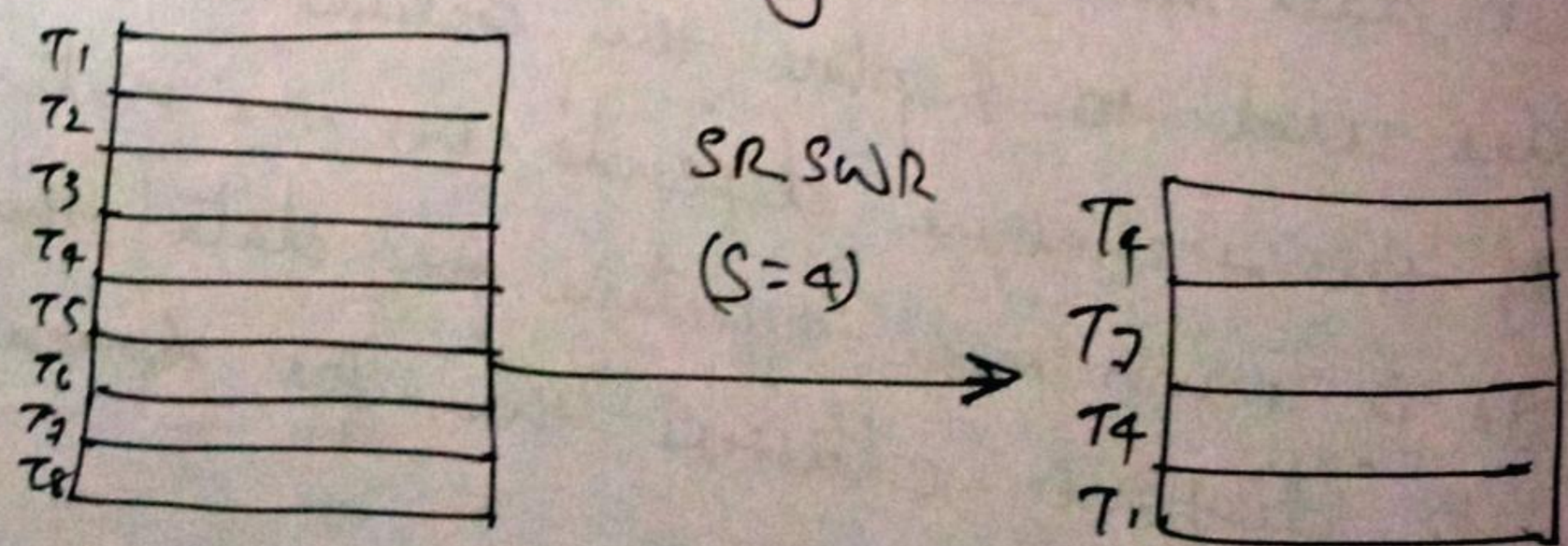
Sampling

Sampling can be used as data reduction technique because it allows a large data set to be represented by a much smaller random sample of the data.
Suppose that a large data set, D contains N tuples.
The different methods of samplings are given below.

① Simple Random Sample without Replacement of Size(s) - (SRSWOR)
This is created by drawing s of the N tuples from D(s<N), where the probability of drawing any tuple in D is 1/N. that is all tuples are equally likely to be sampled

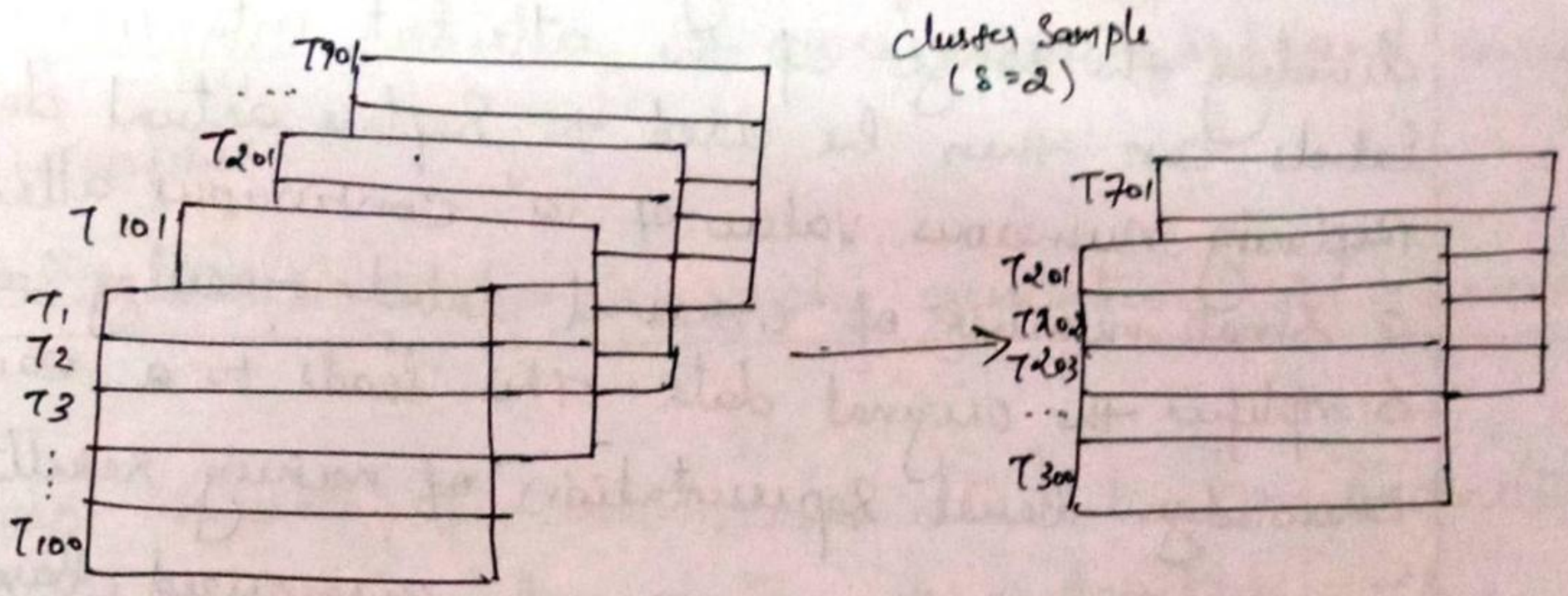


② Simple Random Sample With Replacement of Size s (SRSWR)
This is similar to SRSWOR, except that each time a tuple is drawn from D, it is recorded and then replaced, that is after a tuple is drawn, it is placed back in D so that it may be drawn again.

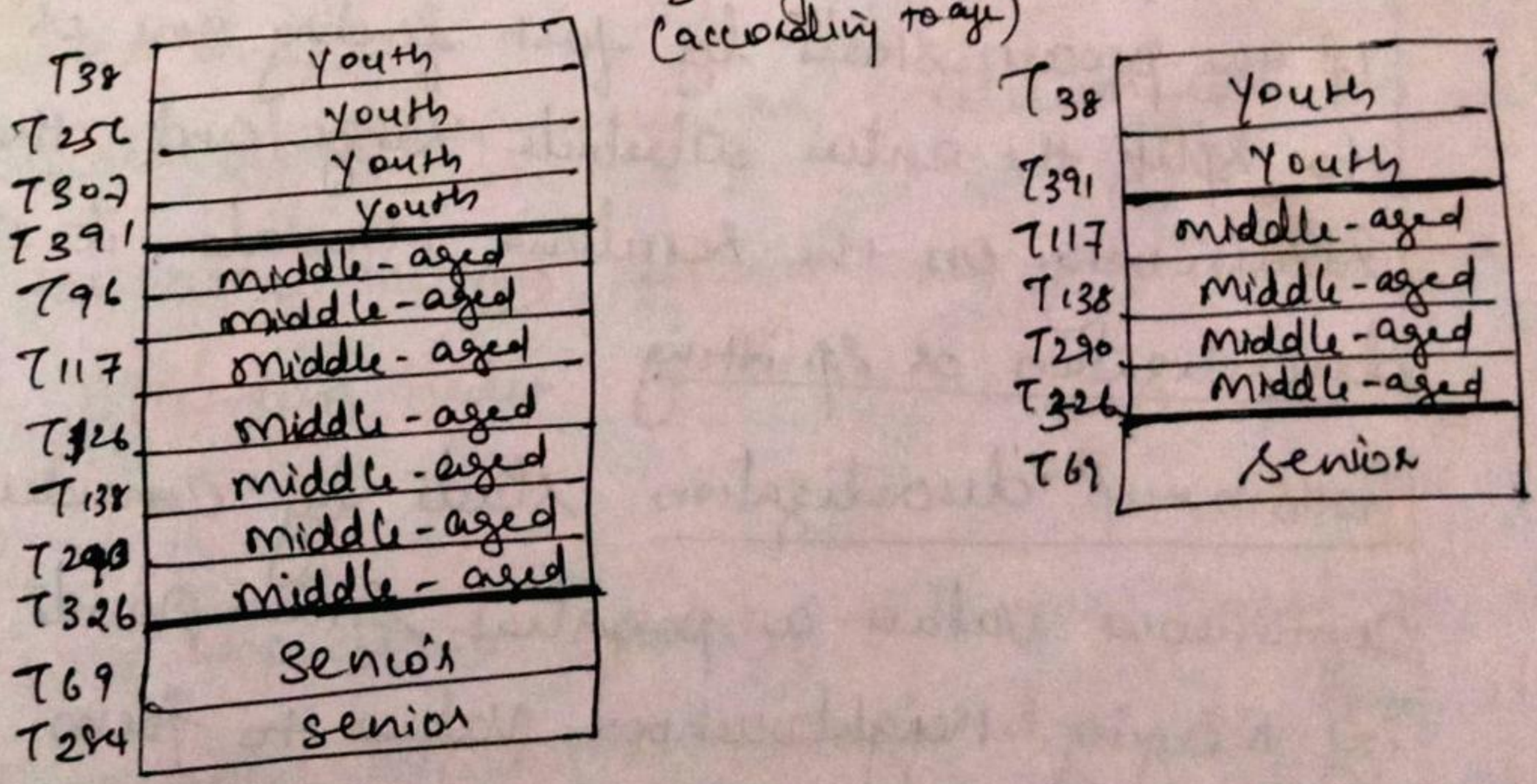


Scanned by TapScanner

⊛ Cluster Sample - If the tuples in D are grouped into M mutually disjoint clusters then an SRS of s clusters can be obtained where s < M.



⊛ Stratified Sample :- If D is divided into mutually disjoint parts called Strata, a stratified sample of D is generated by obtaining an SRS at each stratum. This ensures a representative sample, especially when the data are skewed.



Advantage of sampling for data reduction is that the cost of obtaining a sample is proportional to the size of the sample, s, as opposed to N, the data set size. Sampling complexity is proportional to the size of the sampling data.

Scanned by TapScanner

Data Discretization and Concept Hierarchy Generation

Data Discretization techniques can be used to reduce the numbers of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values. Replacing numerous values of a continuous attribute by a small number of interval labels thereby reduces and simplifies the original data. This leads to a easy-to use knowledge-level representation of mining results.

Discretization techniques can be categorized based on how the discretization is performed such as

i) whether it uses class information or which direction it proceeds (top-down vs bottom-up).

If the discretization process class information then it is Supervised discretization otherwise it is unsupervised

If the process starts by first finding one or a few point to split the entire attribute range and then repeats this recursively on the resulting intervals, it is called top-down discretization or Splitting

Bottom-up discretization starts by considering all of the continuous values as potential split-points, removes some by merging neighbourhood values to form intervals and then recursively applies this process to the resulting inter

A Concept hierarchy for a given numerical attributes defines a discretization of the attribute. Concept hierarchies can be used to reduce the data by collecting and

Scanned by TapScanner

Replacing low-level concepts (such as values for age) with higher level concepts (such as youth, middle-aged or senior). Although details is lost by such data generalization, the generalized data may be more meaningful and easier to interpret.

Discretization & concept hierarchy generation for numerical data

Concept hierachy for numerical attribute can be constructed automatically based on data discretization. The following methods are used

i) Binning
ii) histogram analysis
iii) entropy based discretization
iv) χ^2 - merging
v) cluster analysis
vi) Discretization by intuitive partitioning

Each of the above method assumes that the values are sorted in ascending order.

Binning

Binning is a top-down splitting technique based on a specified number of bins. Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user specified number of bins as well as the presence of outliers

Histogram Analysis

Histogram analysis is an unsupervised discretization technique because it does not use class information. Histograms partition the values of an attribute, A into disjoint ranges called buckets.

Equal Width histogram - the values are partitioned into equal size partitions or range.

Equal frequency histogram - the values are partitioned into ideally, each partition contains the same number of data tuples.

Histogram analysis algm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy. ~~with the procedure terminating once~~

Entropy based discretization

Entropy is one of the most commonly use discretization measures. Entropy based discretization is a supervised, top-down splitting technique. It explores class distribution information in its calculation and determination of split-points.

Let D consist of data tuples defined by a set of attribute and ~~to~~ a class label attribute. The class label attribute provides the class information per tuple. The basic method for entropy-based discretization of an attribute A within the set is as follows.

Scanned by TapScanner

1. Each value of A can be considered as a potential interval boundary or split-point to partition the range of A. ie a split-point for A can partition the tuples in D into two subsets satisfying the conditions A ≤ split point & A > split-point.

2. Entropy-based discretization uses information regarding the class labels of tuples.

Expected information Requirement is given by

$$Info_A(D) = \frac{|D_1|}{|D|} Entropy(D_1) + \frac{|D_2|}{|D|} Entropy(D_2)$$

where D_1 & D_2 are the tuples in D satisfying the condition A < split-point & A > split-point. $|D|$ is the no: of tuples in D

the entropy function can be calculated as D,

$$Entropy(D_1) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where p_i is the probability of class C_i in D_1, determined by dividing the number of tuples of classes C_i in D_1 by $|D_1|$ the total no: of tuples in D,

3. The process of determining a split-point is recursively applied to each partition obtained until some stopping criteria is met

Scanned by TapScanner

Interval merging by χ^2 analysis

Discretization with χ^2 employs a bottom-up approach by finding the best neighbouring intervals and then merging these to form larger intervals. this method is Supervised in that it uses class information. If two adjacent intervals have a very similar distribution of classes then the intervals can be merged otherwise they Remain seperated.

Cluster Analysis

Cluster analysis is a popular data discretization method. A clustering algm can be applied to discretize a numerical attribute A, by partitioning the values of A into clusters or groups. clustering produce high-quality discretization. clustering can be used to generate a concept hierarchy for A by following either a top-down splitting strategy or a bottom-up merging strategy where each cluster forms a node of the concept hierarchy.

Discretization by intuitive partitioning

Discretization by intuitive partitioning methods are useful in the generation of numerical hierarchies, A The 3-4-5 Rule is used to segment numerical data into Relatively uniform natural seeming intervals. In general, the Rule partitions a given Range of data

Scanned by TapScanner

into 3, 4 or 5 relatively equal-width intervals. Recursively and level by level, based on the value range at the most significant digit. The rule is as follows -

(a) If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit then partition range into 3 intervals (3-equal width intervals for 3, 6 & 9 & 3 intervals in the grouping of 2-3-2 for 7)

(b) If it covers 2, 4, or 8 distinct values at the most significant digit, then partitioning the range into 4 equal width intervals

(c) If it covers 1, 5 or 10 distinct values at the most significant digit then partitioning the range into 5 equal width intervals.

Concept Hierarchy Generation for categorical Data

Categorical data are discrete data. Categorical attributes have a finite number of distinct values, with no ordering among the values. Example include geographic location, job category and item type. There are several methods for the generation of concept hierarchies for categorical data.

Specification of a partial ordering of attributes explicitly at the schema level by users or experts :-

A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level. eg: A dimension location of a

dataWarehouse may contain following group of attributes street, city, state or country. A hierarchy can be defined by specifying the total ordering among these attributes at the schema level. Such as street < city < state < country.

Specification of a portion of a hierarchy by explicit data grouping:-

In large database it is unrealistic to define an entire concept hierarchy by explicit value enumeration. Eg: State and country form a hierarchy at schema level, a user could define some intermediate level manually such as "{Alberta, Saskatchewan, Manitoba} ⊂ prairies-canada" & "{British columbia, prairies-canada} ⊂ western-canada"

Specification of a set of attributes, but not of their partial ordering:-

A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly state their partial ordering. The system can then try to automatically generate the attribute ordering so as to construct a meaningful concept hierarchy.

Specification of only a partial set of attributes:-

Sometimes the user may be careless when defining a hierarchy or have only a vague idea about what should be included in a hierarchy. Consequently the

Scanned by TapScanner

user may have included only a small subset of the relevant attribute in the hierarchy specification. eg:- Instead of including all of the hierarchy relevant attributes for location, the user may have specified only street and city.

Scanned by TapScanner

Question - Bank

1. Explain various stages in knowledge discovery process With neat diagram.

2. Describe about data integration

Scanned by TapScanner

How does a classification work ?

Data classification is a two-step process. In the first step, a classifier is built describing a predetermined set of data classes or concepts. This is a learning phase or (training phase), where a classification algorithm builds the classifier by analyzing or learning from a training set made up of database tuples and their associated class labels. Each tuple that constitutes the training set is referred as sample or object or class.

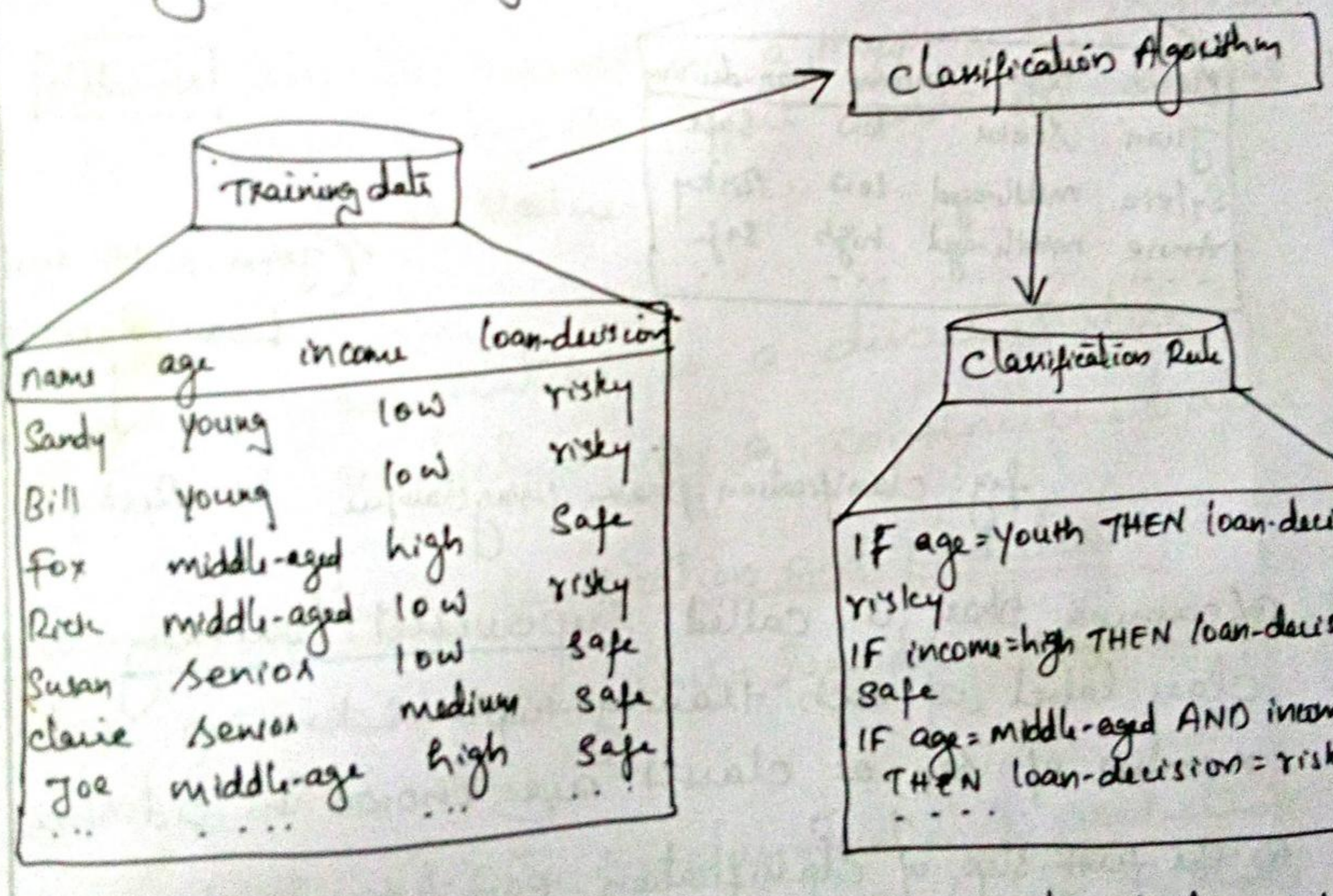


fig: Learing phase: Training data are analyzed by a classification Algorithm. Here the class label attribute is loan-decision and the learned model or classifier is represented in the form of classification Rules.

Scanned by TapScanner

(2) In the second step, the classifier is used for classification. Here the data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples, if the accuracy is considered acceptable

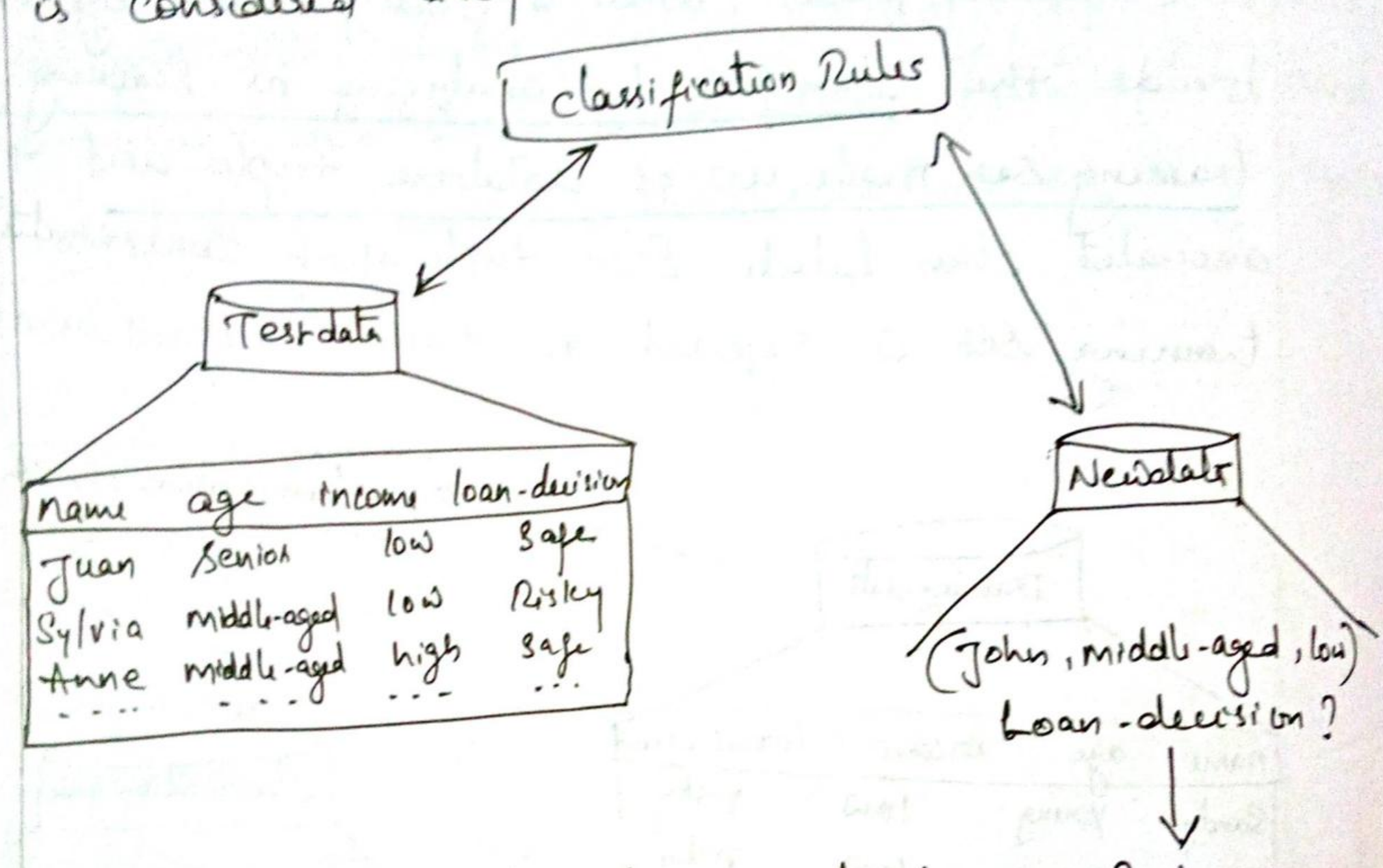


fig: classification phase using classifier

- Learning phase is called supervised learning because class label of each training tuple is known and the number of set of classes are known in advance
- The first step of classification can be viewed as a function $y = f(x)$ that can predict the associated class label 'y' for the given tuple 'x'
- The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier

Scanned by TapScanner

Prediction

Data prediction is a two step process, similar to that of data classification. For prediction there is no 'class labed attribute' because the attribute for which values are being predicted is continuous-valued (ordered) rather than categorical (discrete value and unordered). The attribute can be referred as predicted attribute.

Eg: A prediction model want to predict the expenditures in dollars of potential customers on computer based on the given income & occupation.

prediction can be viewed as a mapping or function $y = f(x)$ where 'x' is the input and the output 'y' is a continuous value.

Note:-
Output of classification is a discrete value
output of prediction is a continuous value.

Issues Regarding classification and prediction

There are mainly two issues regarding the classification and prediction. They are.

1. preparing the Data for classification & prediction
2. Comparing classification and prediction Methods

Preparing the Data for classification and prediction:-

- Data cleaning:- This refers to the preprocessing of data inorder to remove or reduce noise and the treatment of missing values.

Scanned by TapScanner

* 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. Eg: a strong correlation b/w attributes A_1 & A_2 suggest that one of the two could be removed from further analysis.
Attribute Subset Selection can be used in these cases to find a reduced set of attributes (ie removal of irrelevant attributes).

* Data Transformation and Reduction :- The data can be transformed by any of the following methods.

→ Normalization - The data is transformed using normalization. Normalization involves scaling all values for given attributes in order to make them fall within a small specified range. Nomalization is used when in the learning step, the neural networks or the methods involves measurements are used

→ Generalization - The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

Note:-
Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis and clustering.

Comparing classification and prediction Methods:-

* Accuracy - Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and

Scanned by TapScanner

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

* Speed :- This refers to the computational cost in generating and using the classifier or predictor
* Robustness - It refers to the ability of classifier or predictor to make correct predictions from given noisy data
* Scalability - Scalability refers to the ability to construct the classifier or predictor efficiently. given large amount of data
* Interpretability - It refers to what extend the classifier or predictor understands.

Classification by Decision Tree Induction

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 and each leaf node & holds a class label. The topmost node in a tree is a root node. Example :

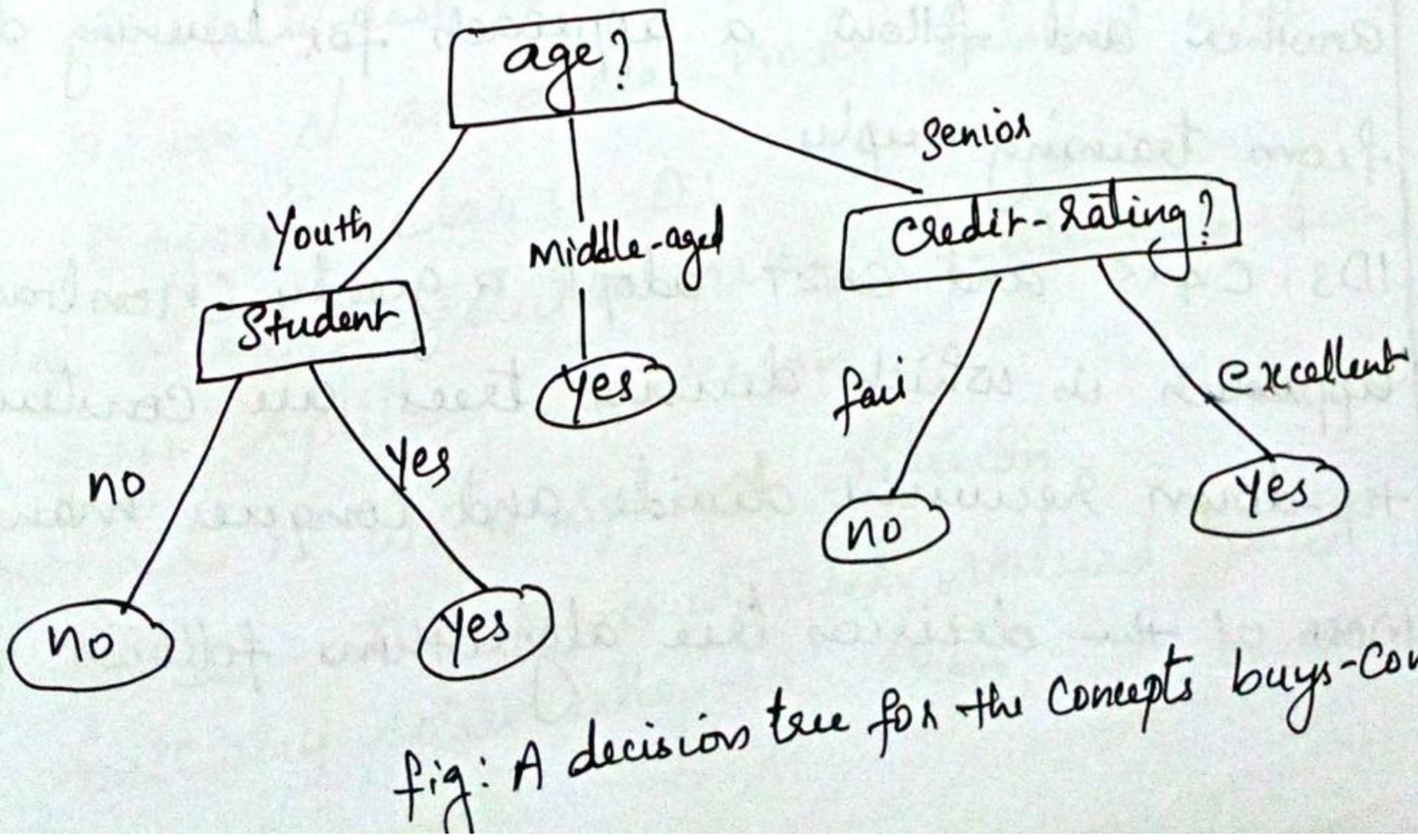


fig: A decision tree for the concepts buys-computer

Scanned by TapScanner

The above decision tree represents the concept buys-computer that is it predicts whether a customer at Allelectronics is likely to purchase a computer.

How are decision tree used for classification?

Given a tuple x, for which the associated class label is unknown the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple.

Advantage of Decision Tree

→ The construction of Decision tree classifier does not require any domain knowledge or parameter setting & is appropriate for exploratory knowledge discovery.

→ Decision trees can handles high dimensional datas.

Decision Tree Induction

Researcher in machine learning developed a decision tree algorithm known as ID3 (Iterative Dichotomiser) and is expanded earlier by C4.5 (Successor of ID3) and later by CART.

ID3 and CART were invented independently of one another and follow a approach for learning decision tree from training tuples.

ID3, C4.5 and CART adopt a greedy (nonbacktracking) approach in which decision trees are constructed in a top-down recursive divide and conquer manner

Most of the decision tree algorithms follows top-down

Scanned by TapScanner

approach which starts with a training set of tuples. and their associated class labels. the training set is recursively partitioned into smaller subsets as the tree is being built. A basic decision tree algorithm is summarized as follows.

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 criteria consists of a splitting attribute & either a split point or splitting point.

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;
(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

Scanned by TapScanner

(9) attribute_list ← attribute_list - splitting attribute
(10) for each outcome j of splitting criterion
(11) let Dj be the set of data tuples in D satisfying outcome j
(12) if Dj is empty then
(13) attach a leaf labeled with the majority class in D to node N;
(~~14~~) else ~~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;
end for
(15) Return N;

* The algorithm is called with 3 parameters D, attribute_list and Attribute_Selection method. Where D is the data partition, ~~attribute~~ attribute_list is a list of attributes describing the tuple and the attribute_selection method specifies procedure for selecting the best attributes that partition the tuple according to the class.

* The tree starts as a single Node N, representing the training tuple in D.

* If the tuple in D are all of same class, then node N becomes the leaf and is labeled with that class.

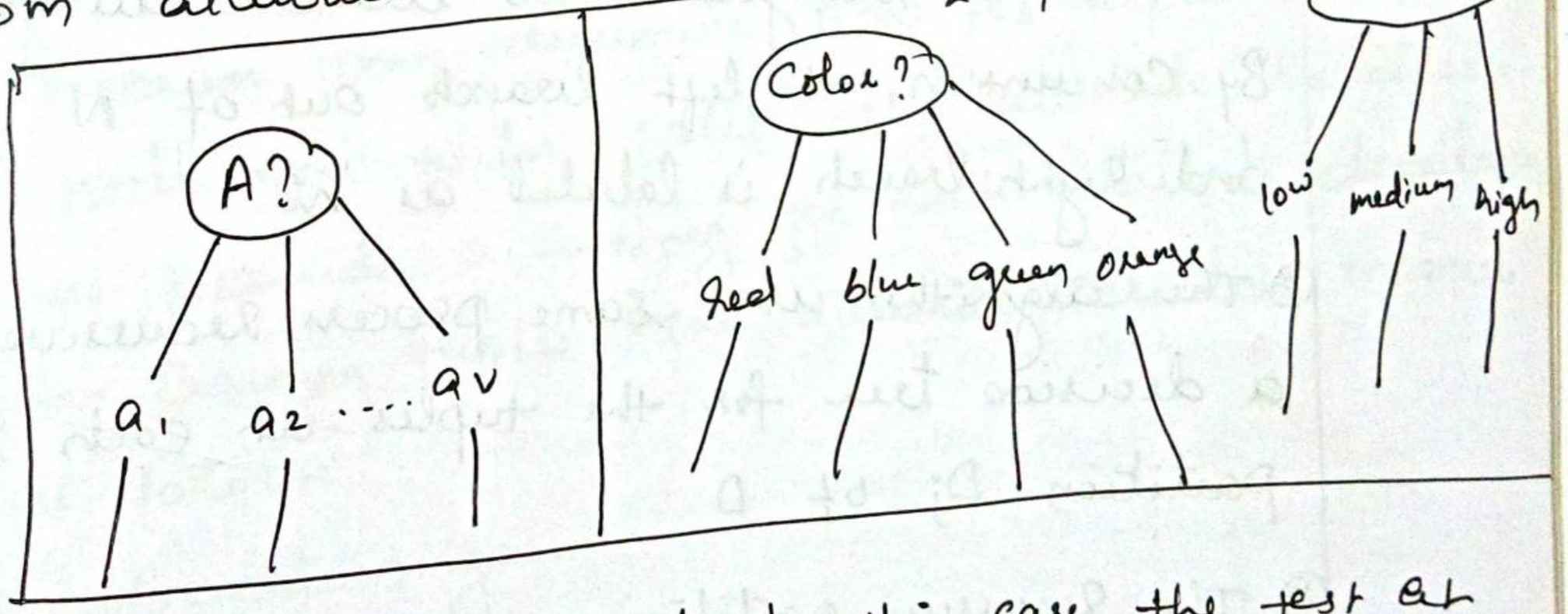
* Otherwise the algorithm calls attribute_selection method to determine the splitting criterion. Splitting criterion indicates splitting attribute that can be either split-point or splitting subset.

Scanned by TapScanner

* Splitting attributes has distinct values $\{a_1, a_2, .. a_v\}$ based on the training set. The value can be of 3 types.

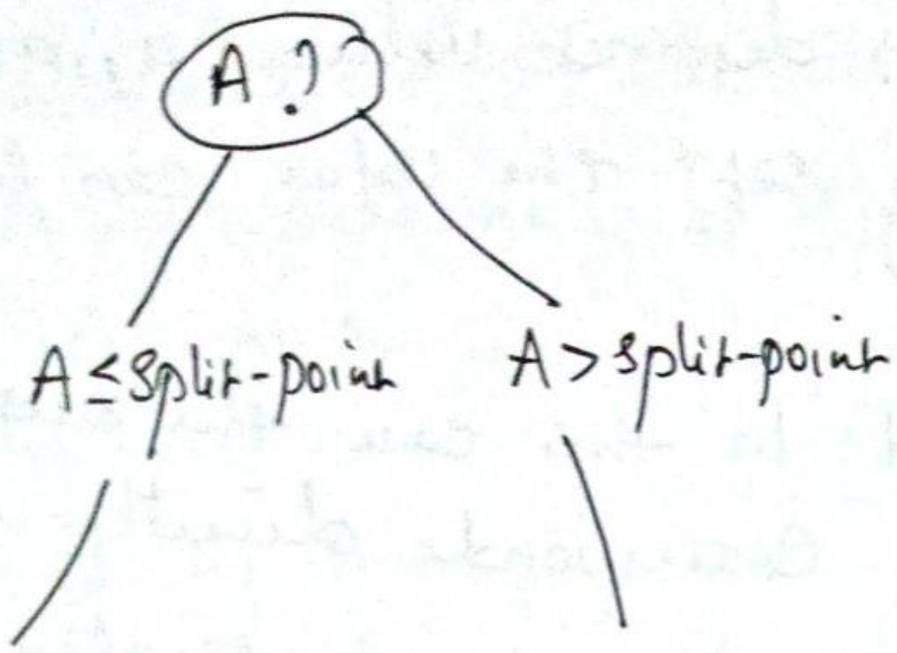
1. A is discrete valued: In this case the outcomes of the test at Node N corresponds directly to the known values of A. A branch is created for each known value a_j, of A and labeled with that value. Partition D_j is the subset of class-labeled tuples in D having value a_j of A. Because all of the tuples in the given partition have the same value for A, then A need not be considered in any future partitioning of the tuple. It is removed from attribute-list

Example

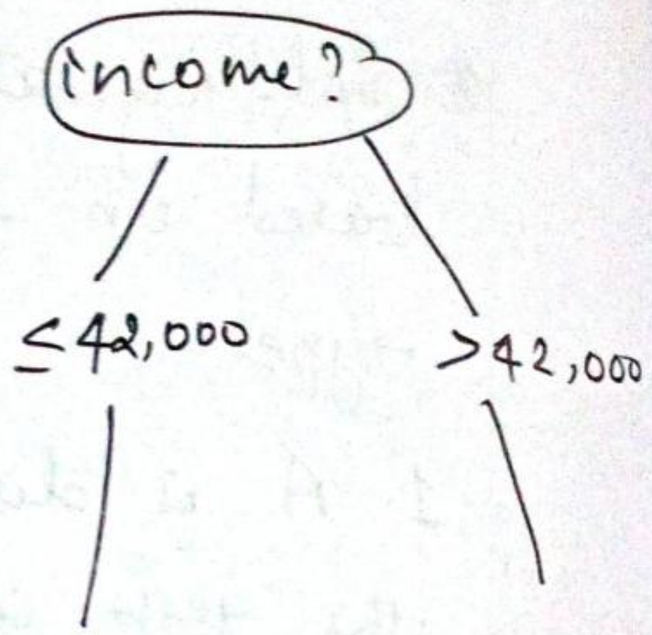


2. A is continuous-valued: In this case the test at node N has 2 possible outcome corresponding to the condition $A \leq$ split point and $A >$ split point. where split-point is returned by attribute-selection method as a part of splitting criterion. Two branches are grown from N and labeled according to the outcome

Scanned by TapScanner



Example.



3. A is discrete-valued and binary tree must be produced The test at node N is of the form "$A \in S_A$?". S_A is the splitting subset for A, returned by Attribute-Selection-method as a part of splitting criterion. It is a subset of the known values of A. If a given tuple has value a_j of A and if $a_j \in S_A$ then the test at node N is satisfied. Two branches are grown from N. By convention, the left branch out of N is labeled 'yes' and right branch is labeled as 'no'

④ The algorithm uses same process recursively to form a decision tree for the tuples at each resulting partition D_j of D

⑤ The recursive partitioning stops only when any of the following condition is true

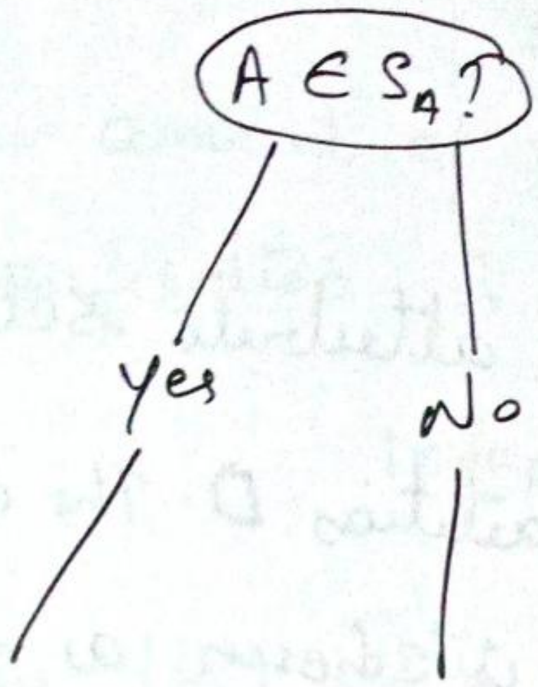
a) All the tuples in partition D belongs to same class

b) There are no remaining attributes in which the tuples may be further partitioned

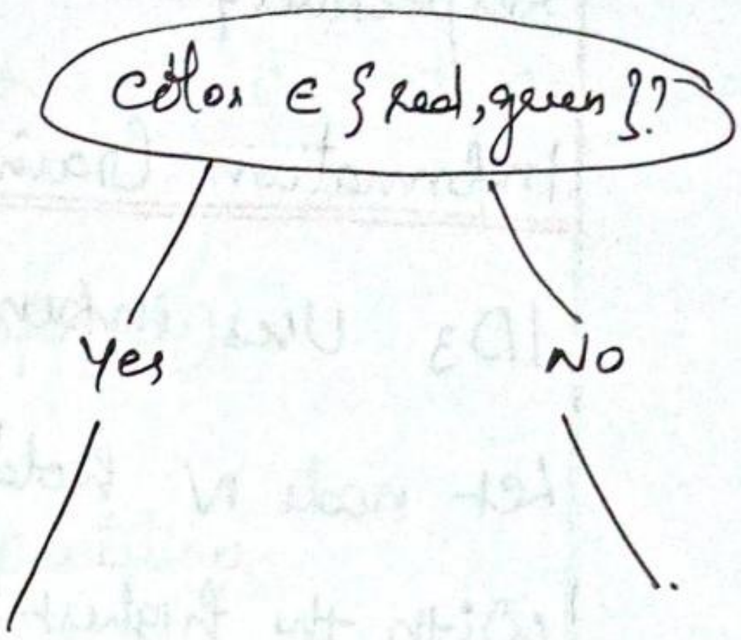
c) There are no tuples for a given branch, that is partition D_j is empty

Scanned by TapScanner

⊛ The resulting decision tree is finally returned



Example:



Attribute Selection Measure

An attribute selection measure is used for selecting the splitting criteria that seperates a given data partition D, of class labeled training tuples into individual class. Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split. The attribute selection measures provide a ranking for each attribute describing the given training tuples. The attribute selection measure are as follows.

a) Information gain
b) Gain ratio
c) Gini Index

~~Let~~ The notation used in above three methods are Let D be the data partition, a training set of class-labeled tuples. If the class label attribute has m distinct values defining m distinct class, C_i (for $i=1,..,m$) Let $C_{i,D}$ be the set of tuples of class C_i in D. Let $|D|$

Scanned by TapScanner

and $|C_i, D|$ denote the number of tuples in D and C_i, D respectively.

Information Gain - ID3

ID3 Uses information gain as its attribute selection measure. Let node N holds the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for Node N. This attribute minimizes the information needed to classify the tuple in the resulting partitions. ~~and~~

The expected information needed to classify a tuple in D is given by

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

where p_i is the probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. A log function to the base 2 is used, because the information is encoded in bits.

Info(D) is the average amount of information needed to identify the class label of a tuple in D. It is also known as entropy of D

Suppose we have to partition the tuples in D on some attribute A having V distinct values $\{a_1, a_2 .. a_v\}$ from the training data. If A is discrete values these values correspond directly to the 'V' outcomes of a test on A. Attribute A can be used to split D into 'v' partitions or subsets $\{D_1, D_2 - -- D_v\}$ where D_j contains tuples in D that

have outcome a_j of A. These partitions would correspond to the branches grown from Node N.

The amount of information need to assess at an exact classification is measured by

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

where $\frac{|D_j|}{|D|}$ acts as weight of j^{th} partition.

$Info_A(D)$ is the expected information required to classify the tuple from D based on the partition by A.

If expected information is smaller then purity (all tuples belong to same class) increases.

Information gain can be defined as the difference between the original information required and the new requirement original information - based on just proportion of classes. new requirement - obtained after partitioning on A

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

Example:-

The following table represents a training set D, of class labeled tuples randomly selected from Allelectronics Customer database.

The class label attribute, buys-computer, has two distinct values {ie m=2} namely {yes, no}. Two distinct classes are there C_1 = yes & C_2 = no.

Scanned by TapScanner

A Root node N is created for the tuples in D. To find the splitting criterion for these tuples, we must compute information gain of each attribute

Table 1:

RID	Age	income	Student	credit-Rating	class: Buys-comp
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	Medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

First, we can compute the expected information needed to classify a tuple in D.

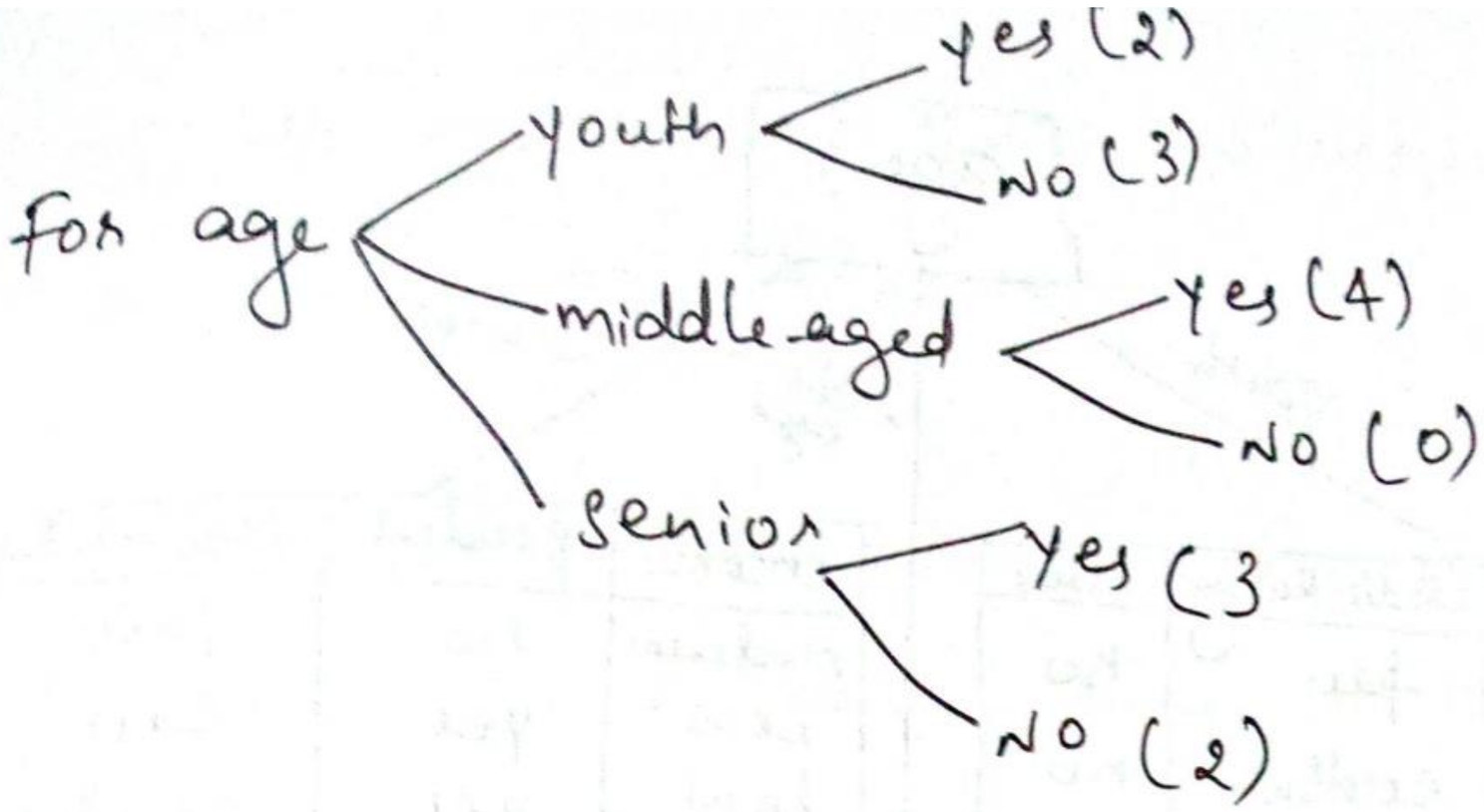
~~info (Age) tells start with age attribute~~

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

~~For the age attribute~~

Next we need to compute the expected information requirement for each attribute. Lets start with age

Scanned by TapScanner



$$\therefore \text{Info}_{(age)}^{(D)} = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) \right)$$

$$+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\left(\frac{4}{4}\right) - \frac{0}{4}\log_2\left(\frac{0}{4}\right) \right)$$

$$+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) \right)$$

$$= \underline{0.694 \text{ bits}}$$

Gain information for age attribute is

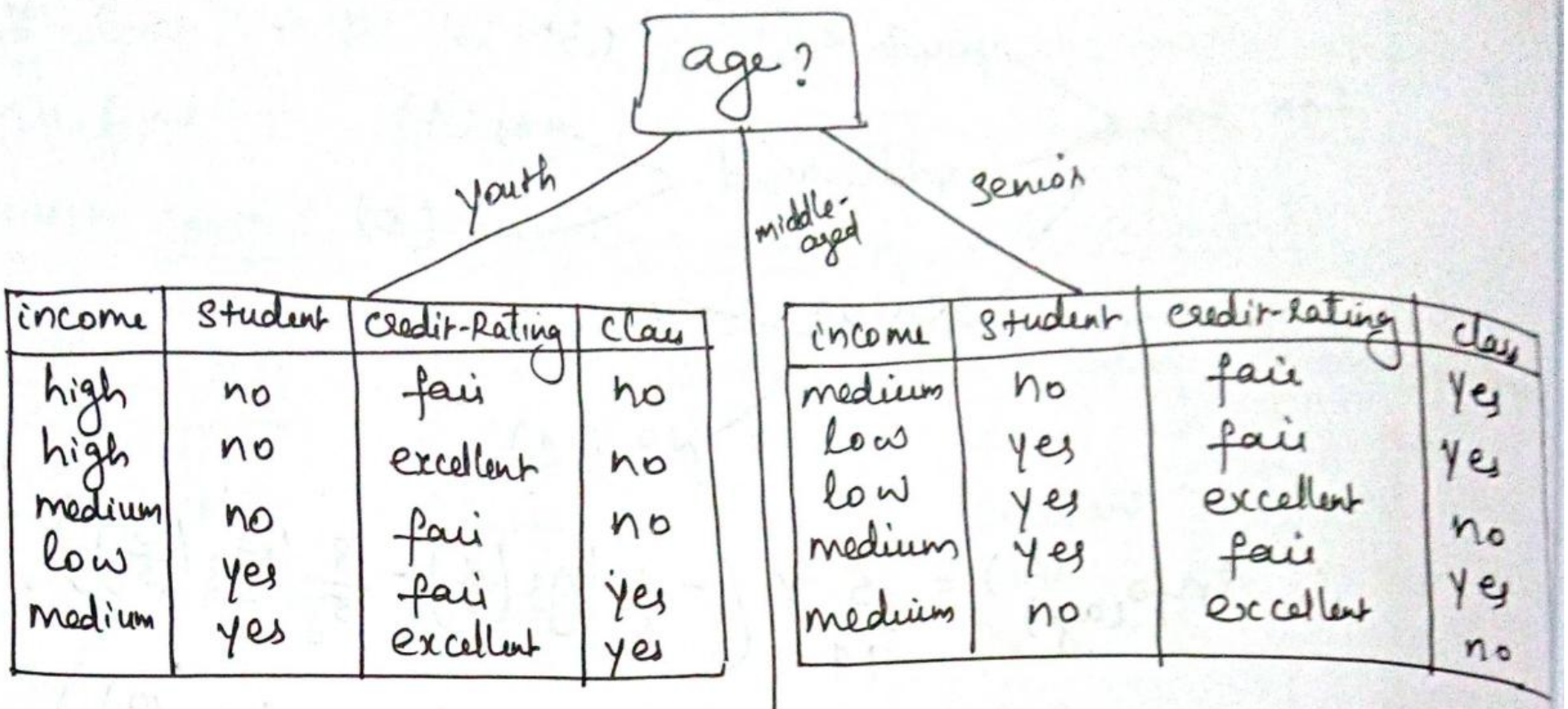
$$\text{Gain}(age) = \text{Info}(D) - \text{Info}_{(age)}^{(D)} = 0.940 - 0.694 = \underline{0.246 \text{bits}}$$

III^rly we can compute Gain(income) = 0.029 bits
Gain (student) = 0.151 bits
Gain (credit-rating) = 0.048 bits.

of the above attribute age is having highest information gain value so it is selected as splitting attribute.

The decision tree after the first partition is as given below.

Scanned by TapScanner



income	student	credit-rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit-rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Income	student	Credit-Rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

To compute the information gain of an attribute that is Continuous-Valued;

If the attribute A is continuous-valued, we must determine the best split-point for A, which is a threshold on A.

First sort the values of A in increasing order. The midpoint between each pair of adjacent values is considered as a possible split-point. If there are 'v' values for A, then v-1 possible splits are evaluated. The midpoint between each pair of adjacent value is considered as a possible split-point. ie If there are 'v' values then 'v-1' possible splits are evaluated.

Scanned by TapScanner

eg: the midpoint between the value a_i and a_{i+1} of A is $\frac{a_i + a_{i+1}}{2}$

If the values of A are sorted, then determining of best split-point for A requires only one pass through the values. For each split-point for A, $Info_{(A)}(D)$ is evaluated. The point with the minimum expected information requirement for A is selected as the split-point for A. D_1 is the set of tuples satisfied $A \leq$ split-point & D_2 is the set of the tuples satisfies $A >$ split-point.

Gain Ratio (C4.5)

The information gain measure is applied to tests which has many outcomes. That is it prefer to select attributes having a large number of values.

C4.5 a successor of ID3, uses an extension to information gain known as gain ratio. It applies a kind of normalization to information gain using split-information value defined with Info(D) as

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

this value represents the potential information generated by splitting the training data set, D into 'v' partitions corresponding to the v outcomes of a test on attribute A

For each outcome, it considers the number of tuples having

that outcome with respect to the total number of tuples in D. The gain ratio is defined as

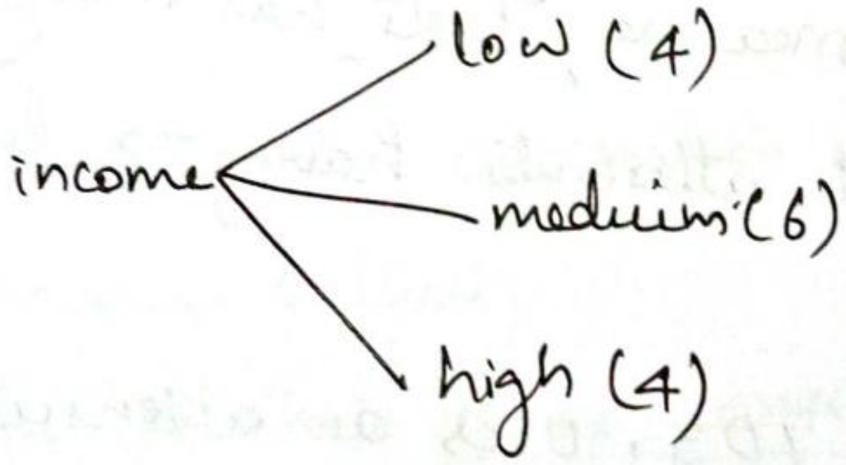
$$\text{Gain Ratio}(A) = \frac{\text{Gain}(A)}{\text{Split Info}(A)}$$

* The attribute with the maximum gain ratio is selected as the splitting attribute

Example:-

Computation of gain ratio for the attribute income from the above given Table 1:

A test on income splits the data of table into three partitions, namely low, medium and high.



To compute the gain ratio of income, we first evaluate $\text{SplitInfo}_{(A)}(D)$ ie

$$\text{SplitInfo}_A(D) = \frac{-4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right)$$

$$= \cancel{0.926 \text{ bits}} \quad 1.5586 \text{ bits}$$

From table 1, we already obtain the information Gain(income) = 0.029.

$$\therefore \text{Gain Ratio(income)} = \frac{0.029}{\cancel{0.926}\ 1.5586} = \cancel{0.0186.} \quad 0.0186$$

Scanned by TapScanner

Gini Index (CART) → classification & Regression Tree

Gini Index is used in CART.

Gini index measures the impurity of D, a data partition or set of training tuples as

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

where p_i is the probability that a tuple in D belongs to class C_i and is estimated by $\frac{|C_i, D|}{|D|}$. The sum is computed over m classes.

Gini Index peforms a binary split for each attribute.

Consider the case where A is a discrete-valued attribute having v distinct values $\{a_1, a_2, \ldots a_v\}$ occuring in D.

To determine the best binary split on A, we examine all the possible subsets that can be formed using known values of A. Each subset, S_A can be considered as a binary test for attributes A of the form $A \in S_A$

Given a tuple, this test is satisfied if the value of A for the tuple is among the values listed in S_A. If A has v possible values, then there are 2^v possible subsets.

Eg: If income is having 3 possible values, like {low, medium, high} then the possible subsets are

① {low, medium, high}
② {low, medium}
③ {low, high}
④ {medium, high}
⑤ {low}
⑥ {medium}
⑦ {high}
⑧ { }

Scanned by TapScanner

⊗ power set & empty set is excluding since they do not represent a split.

∴ there are $2^v - 2$ possible ways to form two partitions of the data D, based on a binary split on A.

when binary split is considered, a weighted sum of the impurity of each resulting partition is computed

For eg :-

If a binary split on A partitions D into D_1 and D_2, the gini index of D is given by

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

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

For continuous valued attribute, each possible split-point must be consider (midpoint b/w each pair of sorted adjacent values are considered). The point giving minimum Gini Index for a given attribute is taken as the split-point of that attribute.

The reduction in impurity that would be incurred by binary split on a discrete-Continuous-valued attribute A is

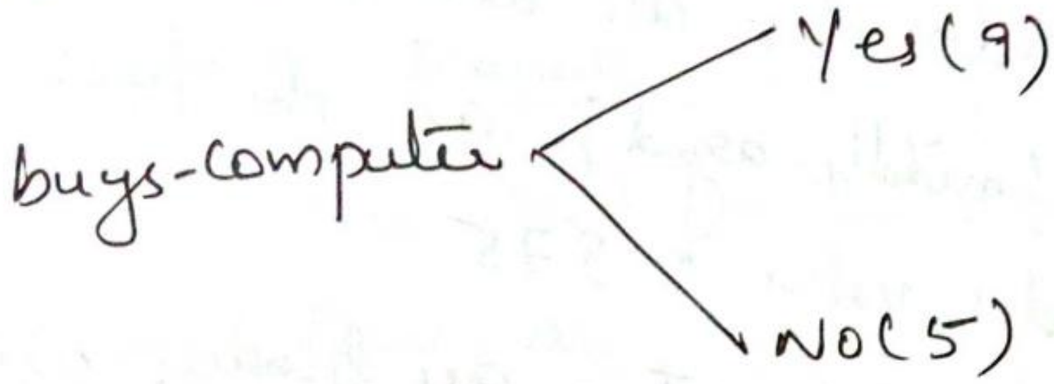
$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

Scanned by TapScanner

the attribute that maximizes the reduction in impurity (minimum Gini Index attribute) is selected as the splitting attribute.

Eg:- Induction of a decision tree using Gini Index.

Let D be the training data for the above given Table 1. where there are 9 tuples belongs to the class buys computer = yes & 5 belongs to the class no



Gini index to compute the impurity of D. equation is

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

To find the splitting criteria for the tuples in D, we need to compute the gini index for each attribute Lets start with the attribute income and considers the splitting subsets {low, medium}. 9+ Results 10 tuples in D_1 satisfying the condition income $\in$ {low, medium} The remaining tuples belong to D_2.

The Gini index value computed based on this partitioning is $Gini_{income \in \{low, medium\}}(D)$

$$= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right)$$

$$= 0.450$$

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

Scanned by TapScanner

III^rd Gini Index Values for the Splits on remaining subset.

Gini index {low, high} & Gini index {medium} = 0.315

Gini index {medium, high} & Gini index {low} = 0.300

The best binary split for attribute income is on {medium, high} or {low} since it have small gini index value.

Evaluating the Gini index for all attribute we obtain {youth, Senior} or {middle-aged} as the best split for age with Gini index value 0.375

attribut {Student} & credit-rating are binary with Gini index values 0.367 & 0.429

The attribute income and splitting subset {medium, high} therefore give the minimum gini index overall, with a reduction in impurity of 0.459 - 0.300 = 0.159

The binary split income ∈ {medium, high} results in the maximum reduction in impurity of the tuples in D and is returned as the splitting criterion Node N is labeled with the criterion, two branches are grown from it and the tuples are partitioned accordingly.

Other proposed attribute selection measures are :-

① CHAID - a decision tree algorithm that is popular in marketing, uses an attribute selection method that is based on statistical χ^2 test for independence

Scanned by TapScanner

④ C-SEP – In certain cases better performance than Gini Index & Information gain

⑤ G-Statistic – close approximation to χ^2 distribution

⑥ MDL (Minimum Description length) – least biased to wards multivalued attributes.

Tree pruning

Tree pruning method address the problem of overfitting the data in the tree. This method use statistical measure to remove least reliable branches from tree.

Advantages of pruning

① Pruned tree tend to be smaller and less complex

② Pruned tree are usually faster and better at correctly classifying independent test data than unpruned tree.

There are two common approaches to tree pruning

① prepruning

② postpruning

In the prepruning approach, a tree is pruned by halting its construction early (eg: by deciding not to further split or partition the subset of training tuples at a given node). Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset tuples or the probability distribution of those tuples.

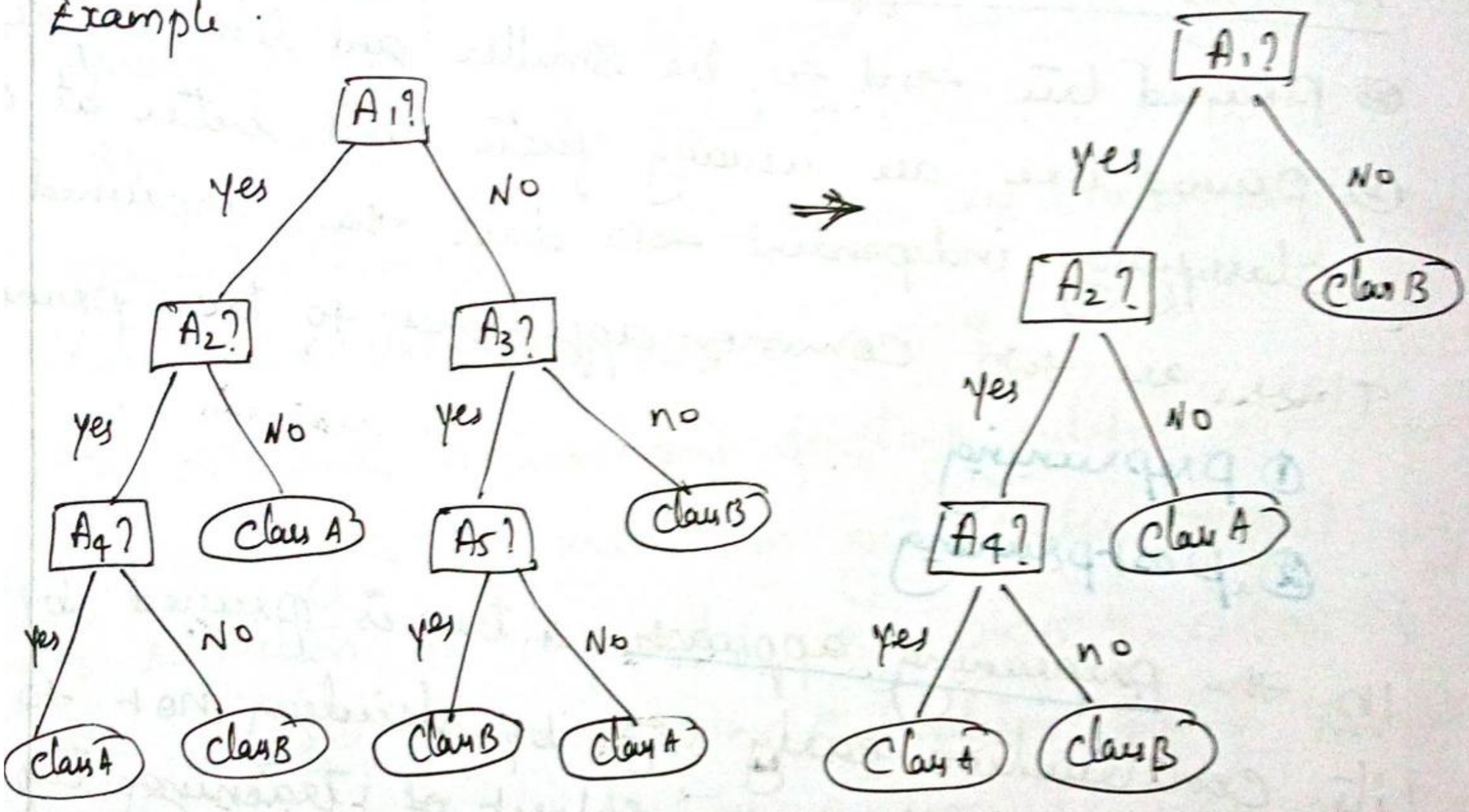
If partitioning the tuples at a node would result in a split that falls below a prespecified threshold, then further partitioning of the given subset is halted. There are difficulties in choosing an appropriate threshold,

Scanned by TapScanner

High thresholds could result in oversimplified tree, whereas low thresholds could result in very little simplification

In postpruning approach, subtrees are removes from a fully grown tree. A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most frequent class among the subtree being replaced.

Example.



In the above eg: the subtree at A_3 node is having class B as common class. So is pruned tree node A_3 is replaced with the leaf node class B.

Example of postpruning approach: Cost complexity pruning Algorithm

This approach considers the cost complexity of a tree to be a function of the number of leaves in the tree and the error rate of the tree.

Scanned by TapScanner

(where the error Rate is the percentage of tuples misclassified by the tree). It starts from the bottom of tree. for each internal node N, it Computes the cost Complexity of the subtree at N ~~If it were to be pruned (ie replaced by a leaf node)~~ and the cost complexity of the subtree at N if it were to be pruned (ie replaced by a leaf node). the two values are Compared. If ~~pru~~ pruning the Subtree at node N would result in a smaller cost Complexity, then the subtree is pruned. otherwise it is kept.

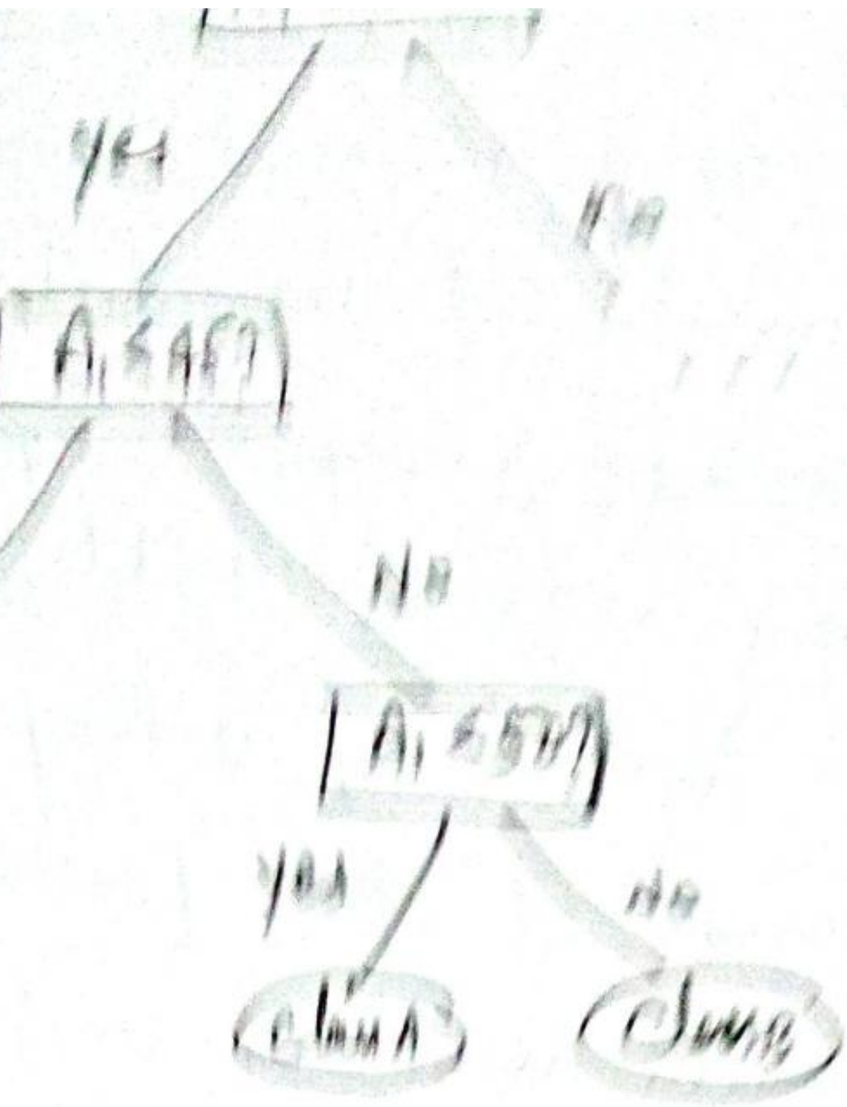
C4.5 uses a method called pessimistic pruning which is similar to the cost complexity method in that it also uses error Rate estimates to make decision Regarding Subtree pruning.

Instead of using error Rate the tree can be pruned based on the number of bits Required to encode them. The best pruned tree is one that minimizes the number of encoding bits. This method adopts the Minimum Description length (MDL) principle.

Postpruning Requires more computation than prepruning and leads to a Reliable tree.

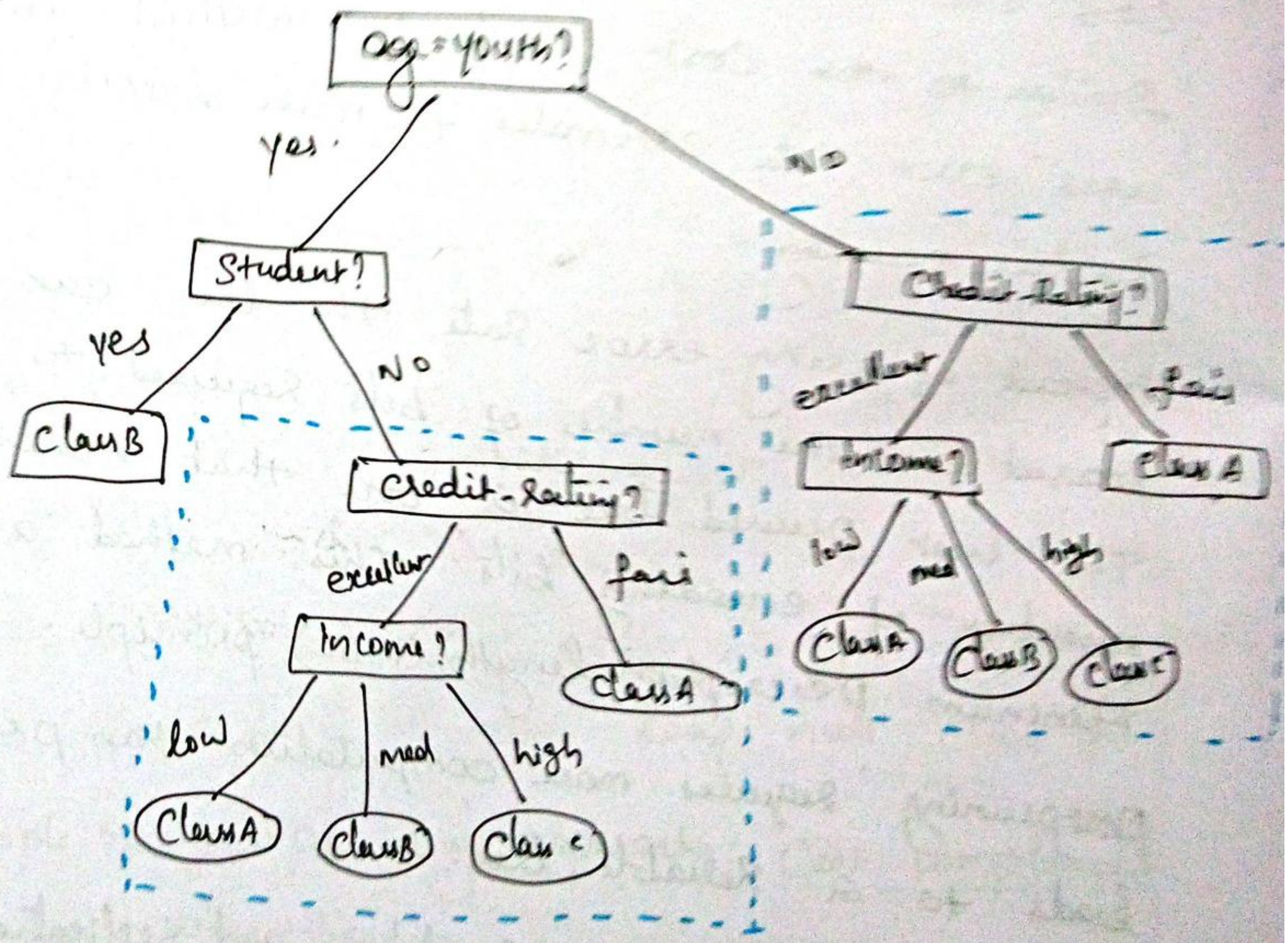
Decision tree Suffer from Repetition and Replication. Repetition occurs when an attribute is Repeatedly tested along a given branch of the tree

Scanned by TapScanner



In application duplicate subtree exist due to use of multivariate split can prevent the problem.

eg:



Scanned by TapScanner

Naive-Bayes Classifier

Bayesian classifier are statistical classifier. They can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Bayesian classification is based on Bayesian theorem and is described below.

Let X be the data tuple and is described on a set of n attributes. Let 'H' be the hypothesis such that the data tuple x belongs to a specified class C.

For classification problem, 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 of 'H' conditioned on X. It reflects the probability that the customer x will buys a computer or not with customer's information age and income.

$P(H)$ is the prior probability. This is the probability that any given customer will buy a computer, regardless of age, income or any other information if it is independent of x.

Bayes' theorem is given by.

$$P(H/x) = \frac{P(x/H)\ P(H)}{P(x)}$$

Scanned by TapScanner

Naive Bayesian classification

The Naive Bayesian classifier works as follows.

1. Let D be the training set of tuples and their associated class labels. Each tuple is represented by n attributes like $X = \{x_1, x_2, \ldots x_n\}$

2. Suppose there are m classes $C_1, C_2 \ldots C_m$. Given a tuple X, the classifier will predict that x belongs to the class having highest posterior probability conditioned on x. The Naive Bayesian classifier predicts the tuple x belongs to the class C_i iff -

$$P(C_i/X) > P(C_j/X) \text{ for } 1 \le j \le m \quad j \ne i$$

3. P(X) is constant for all classes, only $P(X/C_i)\,P(C_i)$ need to be maximized. If the class prior probability are not known, then it is assumed that the classes are equally likely that is $P(C_1) = P(C_2) = \ldots = P(C_m)$.
$P(C_i) = |C_{i,D}| / |D|$ where $C_{i,D}$ is the number of training tuples of class C_i in D.

4. To compute $P(X/C_i)$ of data set with many attributes the computation cost is expensive. To reduce the computation class conditional independence is made and is given by

$$P(X/C_i) = \prod_{k=1}^{n} P(x_k/C_i)$$

$$= P(x_1/C_i) \times P(x_2/C_i) \times \ldots \times P(x_n/C_i)$$

To compute $P(X/C_i)$ consider the following cases.

Scanned by TapScanner

a) (i) ... is discrete then $P(x_k/c_i)$ is the no: of tuples of class C_i in D having the value x_k for the attribute A_k divided by $|C_i,D|$ is the no: of tuples of C_i in D

b) If attribute value is continuous, then Guassian distribution with mean μ and standard deviation σ is defined by

$$g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

& $P(x_k/c_i) = g(x_k, \mu_{c_i}, \sigma_{c_i})$

5. To predict the class label of x, $P(x/c_i)\,P(c_i)$ is evaluated for each class c_i. The classifier predicts that the class tuple x is the class c_i if

$$P(x/c_i)\,P(c_i) > P(x/c_j)\,P(c_j) \quad \text{for } 1 \le j \le m, j \ne i$$

The predicted class label is the class c_i for which $P(x/c_i)P(c_i)$ is the maximum.

Example:- predict the class label using Naive Bayesian classification :-

To predict the class label of a tuple using naive Bayesian classification for the following given Table.

The data tuples are described by the attribute age, income, student and credit-rating.

The class label attribute, buys-computer has two distinct values {yes, no}.

Let C_1 corresponds to buys-computer = yes and C_2 corresponds to buys-computer = no

Scanned by TapScanner

RID	age	income	Student	Credit-rating	class: buys-computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

The tuple wish to classify using naive Bayesian classification is X = (age = youth, income = medium, student = yes, credit-rating = fair)

The value $P(X/C_i)/P(C_i)$ is to be maximize for $i = 1,2$

The prior probability of class can be computed based on training tuples.

$P(buys\text{-}computer = yes) = 9/14 = 0.643$

$P(buys\text{-}computer = no) = 5/14 = 0.357$

To compute $P(X/C_i)$ for $i = 1,2$ the following conditional probabilities are computed

$P(age = youth / buys\text{-}computer = yes) = 2/9 = 0.222$

$P(age = youth / buys\text{-}computer = no) = 3/5 = 0.60$

$P(income = medium / buys\text{-}computer = yes) = 4/9 = 0.444$

Scanned by TapScanner

p(income = medium | buys-computer = no) = 2/5 = 0.400
p(student = yes | buys-computer = yes) = 6/9 = 0.667
p(student = yes | buys - computer = no) = 1/5 = 0.200
p(credit-rating = fair | buys-computer = yes) = 6/9 = 0.667
p(credit-rating = fair | buys - computer = no) = 2/5 = 0.400

Using above probalities, we obtain

$P(X/buys\text{-}computer = yes) = P(age = youth / buys\text{-}computer = yes) \times$
$P(Income = medium / buys\text{-}computer = yes) \times$
$P(Student = yes / buys\text{-}computer = yes) \times$
$P(credit\text{-}rating = fair / buys\text{-}computer = yes)$
$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$

IIIly

$P(X/buys\text{-}computer = no) = 0.60 \times 0.40 \times 0.20 \times 0.40 = 0.019$

To find the class C_i which maximizes $P(X/C_i) P(C_i)$ we compute.

$P(X/buys\text{-}computer = yes) \times P(buys\text{-}computer = yes) = 0.044 \times 0.643$
$= 0.0282$

$P(X/buys\text{-}computer = no) \times P(buys\text{-}computer = no) = 0.019 \times 0.357$
$= 0.007$

Of the above 2 values $P(buys\text{-}computer = yes)$ have max value So the given tuple X comes under the class C_1 i.e buys-computer = yes according to naive baise classification.

Scanned by TapScanner

Question Bank.

1. Describe about Naive Bayes classifier with examples.
2. Explain the issues regarding prediction and classification
3. Explain how C4.5 differs from ID3 algorithm?
4. Write the different steps for drawing decision tree for the given data collection.
5. Write the significance of tree pruning in decision tree Algorithm.

Scanned by TapScanner

MOD-IV

Rule-Based classification

One straightforward method to perform classification is to generate if-then rules that cover all cases. A Decision tree is always used to generate rules, but they are not equivalent. There are differences between rules & tree.

- The tree has an order in which splitting performs. Rule has no such orders
- A tree is created based on looking at all classes. When generating rules, only one class must be examined at a time.

IF-THEN Rules for classification

Rules are good way of representing information or bits of knowledge. A Rule-based classifier uses a set of IF-THEN rules for classification. An if-then rule is an expression of the form

IF condition THEN conclusion.

eg: (R1) IF age = youth AND student = yes THEN buys-computer = yes

The 'IF' part of a rule is known as the rule antecedent or precondition. 'THEN' part is the rule consequent

In the antecedent part the condition consists of one or more attribute tests. and that are logically ANDed

The rule Consequent contain a class prediction.

If the condition in a rule antecedent holds true for a given tuple, then the rule antecedent is satisfied and the

Scanned with CamScanner

Rule covers the tuple.

A Rule R can be assessed by

i) coverage
ii) accuracy.

The coverage and accuracy of a Rule R can be defined as.

$$\text{Coverage}(R) = \frac{n_{covers}}{|D|}$$

where n_{covers} be the no: of tuples covered by Rule R.

$$\text{accuracy}(R) = \frac{n_{correct}}{n_{covers}}$$

where $n_{correct}$ be the no: of tuples correctly classified by R. and $|D|$ be the no: of tuple in D.

Example: Consider the Table of AllElectronics customer database.

For the above Rule R_1, there are 2 tuples out of 14.

Coverage $(R_1) = 2/14 = 14.28\%$.

accuracy $(R_1) = \frac{2}{2}$ (correctly satisfy buys-computer = yes) $= 100\%$

If a Rule is satisfied by X, then the Rule is said to be triggered.

Let X be $X = \{age = youth, income = medium, student = yes, credit-rating = fair\}$

The above tuple X satisfied R_1, which triggers the Rule.

If R_1 is the only Rule satisfied, then the Rule fires by returning the class prediction for X. The above tuple returns the class prediction buys-computer = yes

Scanned with CamScanner

If more than one Rule is triggered, conflict resolution strategy is to be used to figure out which Rule gets to fire and assign its class prediction to x. There are many possible strategies.

i) Size ordering
ii) Rule ordering

The Size ordering scheme assigns the highest priority to the triggering Rule that has the 'toughest' requirements. Toughest is measured by the Rule antecedent size. ie the triggering Rule with the most attribute tests is fired.

The Rule ordering scheme prioritizes the Rules. ~~Beyond~~ The ordering is based on class based or Rule based. With class ordering the classes are sorted in order of decreasing importance. ie Rules for the most frequent class come first, the Rules for the next prevalent class come next & so on.

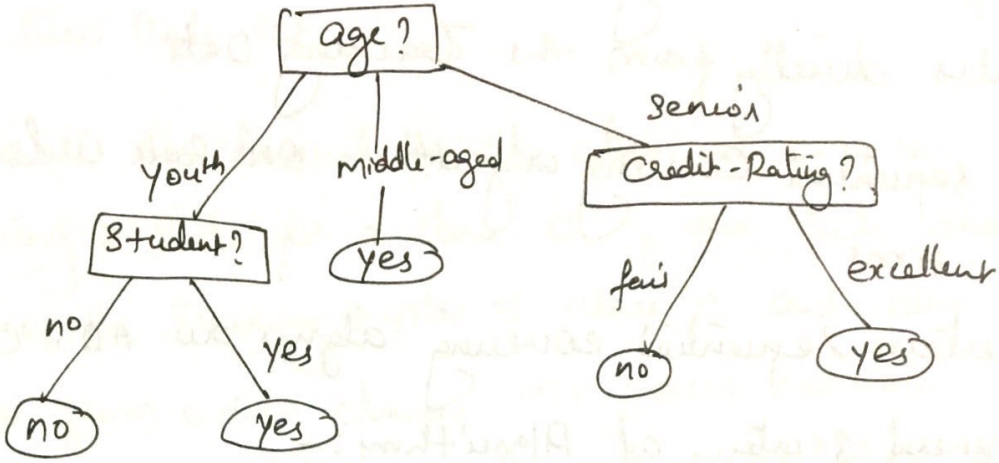
With Rule based ordering the rules are organized into one long priority list, according to some measure of Rule quality such as accuracy, coverage or size.

Rule Extraction from a Decision Tree

To extract Rules from the decision tree, one Rule is created for each path from the root to a leaf node. Each splitting criterion along a given path is logically ANDed to form the Rule antecedent. The leaf

Scanned with CamScanner

node holds the class prediction, forming the Rule Consequent.



For the above decision tree, the rules formed by tracing the path from root node to leaf node is as follows

R_1: If age = youth AND student = no THEN buys-computer = no

R_2: If age = youth AND student = yes THEN buys-computer = yes

R_3: If age = middle-aged THEN buys-computer = yes

R_4: If age = senior AND credit-rating = excellent THEN buys-computer = yes

R_5: If age = senior AND credit-rating = fair THEN buys-computer = no

Rules are mutually exclusive and exhaustive Senior rules are directly extracted from the tree.

Mutually exclusive - means that we cannot have rule conflicts because no two rules will be triggered for the same tuple.

Exhaustive - means there is one rule for each possible attribute-value combination so that this set of rules does not require a default rule

Scanned with CamScanner

Rule Induction Using a Sequential Covering Algorithm

Sequential Covering Algm is used to extract IF-THEN Rules directly from the Training Data

In sequential Covering algorithm one Rule is learned at a time.

Certain Sequential covering algms are AQ, CN_2, RipperK

General Strategy of Algorithms:-

i) Rules are learned one at a time

ii) Each time Rule is learned, the tuples covered by the Rule are removed, and the process Repeats on the Remaining tuples.

The basic Sequential Covering is as given below.

Input,

- D, a dataset class labeled tuples
- Att-Vals, the set of all attributes and their possible values

Output

A set of IF-THEN Rules

Method:

(1) Rule set = { } # initial set of Rules learned is empty

(2) for each class c do

(3) . Repeat

(4) Rule = Learn-One-Rule(D, Att-Vals, c);

(5) Remove tuples covered by Rule from D

(6) Until terminating condition

Scanned with CamScanner

(7) Rule-Set = Rule-set + Rule ; # add new Rule to Rule-se

(8) end for

(9) Return Rule-Set ;

The Rules are learned for one class at a time. When learning a Rule for a class C_i, the Rule should cover all of the training tuples of class C and none of the tuples from other classes. The process continues until the terminating condition is met, such as when there are no more tuples or the quality of a Rule returned is below a user-specified threshold.

How are Rules learned ?

Consider the training D, consists of loan application data. Attribute of each tuple include their age, income, education level, Residence, Credit Rating and the term of loan. The classifying attribute is loan-decision, which indicates whether a loan is 'accepted' or 'rejected'.

To learn a Rule for the class 'accept'. We start off the Rule with empty set i

IF THEN loan-decision = accept.

Then consider all possible attribute test that may be added to the Rule. It can be desired from the attribute-value pair (att, val). Choose the attribute which has

Scanned with CamScanner

high quality measure & improves the Rule quality based on the training samples.

Learn-one-Rule finds the attribute test income = high improves the current empty Rule and it becomes

IF income = high THEN loan-decision = accept.

During next iteration the attribute test ends up by selecting credit-rating = excellent and the Rule becomes

IF income = high AND credit-rating = excellent THEN loan-decision = accept.

This process repeat, and each iteration the quality of Rules increases thereby increases accuracy.

Rule Quality Measures

Learn-one Rule needs a measure to Rule quality.

One Rule quality measure based on Information gain is FOIL (First order Inductive learner) a sequential covering algorithm that learns first-order logic Rules.

FOIL-Gain is given by

$$\text{FOIL-Gain} = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right)$$

The above Rule have high accuracy and covers many positive tuples

Let pos(neg) be the number of positive negative tuples covered by R

Scanned with CamScanner

and pos' (neg') be the number of positive (negative) tuples covered by R'

Rule pruning

FOIL uses a simple method to prune a rule to avoid the overfit of data in Rule. It is given by

$$FOIL\text{-}prune(R) = \frac{pos - neg}{pos + neg'}$$

where pos and neg are the number of positive and negative tuples covered by R. This value will increase with the accuracy of R on a pruning set. ∴ If the FOIL-prune value is higher for the pruned version of R, then we prune R.

Classification by Backpropagation

Backpropagation is a neural network learning algorithm. The neural network is a set of connected input/output units in which each connection has a weight associated with it.

During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples. Neural network learning is also referred to as connectionist learning due to the connections between the units.

Scanned with CamScanner

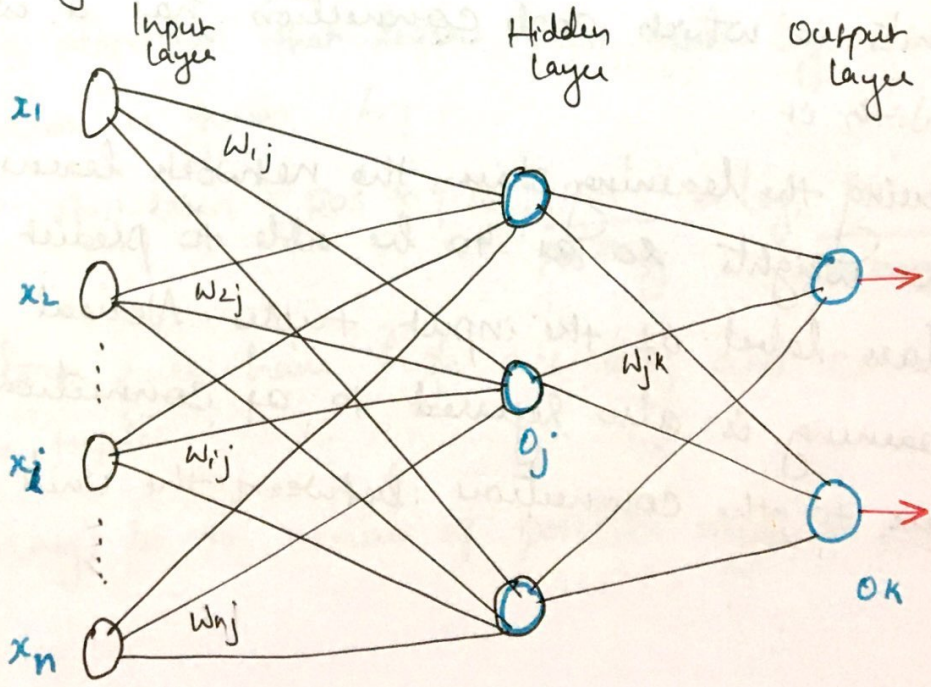
Advantages of neural n/w

- neural n/w have the ability to ~~classify~~ classify patterns on which they are not trained
- neural n/w have high tolerance towards noisy data.
- neural n/w can be used even we have a little knowledge of the relationship between attributes and classes.
- neural n/w are well suited for continuous valued inputs and outputs.

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 or more hidden layer and an output layer.



Scanned with CamScanner

In the above multilayer network,
Each layer is made up of units. The i/p of the n/w correspond to the attributes measured for each training tuple. The inputs are fed simultaneously into the units making up the input layer. This input pass through the input layer and are then weighted and fed simultaneously to the second layer known as hidden layer. The outputs of the hidden layer are inputs to units making up the output layer, which emits the networks' prediction for given tuples.
The units in the input layer is called input units
The units in the hidden layer & output unit is referred as neurodes. The above given n/w has one hidden layer and one output layer. So total of 2 output layer. So it is 2-layer neural n/w.
The n/w is called <u>feed-forward</u> because all the weights are forwarding in the connection, none of the weights cycles back to an input unit or to an o/p unit of previous layer.
It is <u>fully connected</u> in that each unit provides input to each unit in the next forward layer.

Each output unit takes an input, weighted sum of the o/p from the previous layer

Scanned with CamScanner

Back propagation

Backpropagation learns by iteratively processing a dataset of training tuples, comparing the network's prediction for each tuple with the actual known target value. The target value may be known class label of the training tuples. The modification is made in the backward direction that is from the output layer through each hidden layer down to the ~~input~~ first hidden layer hence the name backpropogation.

The Backpropagation Algorithm is summarized below.

Input:-

- D, a data set consisting of the training tuples and their associated target value
- l, learning rate
- network, a multilayer feed-forward network.

Output:

A trained neural network.

Method:

(1) Initialize all weights and bias in network;
(2) while terminating condition is not satisfied {
(3) for each training tuple x in D {
(4) // propagate the inputs forward:
(5) for each input layer unit j {
(6) $O_j = I_j$; # output of an input unit is its actual input value
(7) for each hidden or output layer unit j {

Scanned with CamScanner

(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; # compute the net input of unit j with respect to the previous layer i

(9) $O_j = \frac{1}{1+e^{-I_j}}$; # compute the o/p of each unit.
(output)

(10) # Backpropagate the errors:

(11) for each unit j in the output layer

(12) $Err_j = O_j(1-O_j)(T_j - O_j)$; # compute the error

(13) for each unit j in the hidden layers from the last to the first hidden layer

(14) $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; # compute the error with respect to the next higher layer, k

(15) for each weight w_{ij} in network {

(16) $\Delta w_{ij} = (l) Err_j O_i$ # weight increment

(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } # weight update

(18) for each bias θ_j in network {

(19) $\Delta \theta_j = (l) Err_j$; # bias increment

(20) $\theta_j = \theta_j + \Delta \theta_j$ # bias update

(21) } }

Initialize the Weight :-

The weights in the network are initialized to small random numbers ranging from -1.0 to 1.0 or -0.5 to 0.5. Each unit has a bias associated with it. The bias are also initialized to small random numbers.

Scanned with CamScanner

Each tuple X is processed by the following steps.

Propage the input forward :-

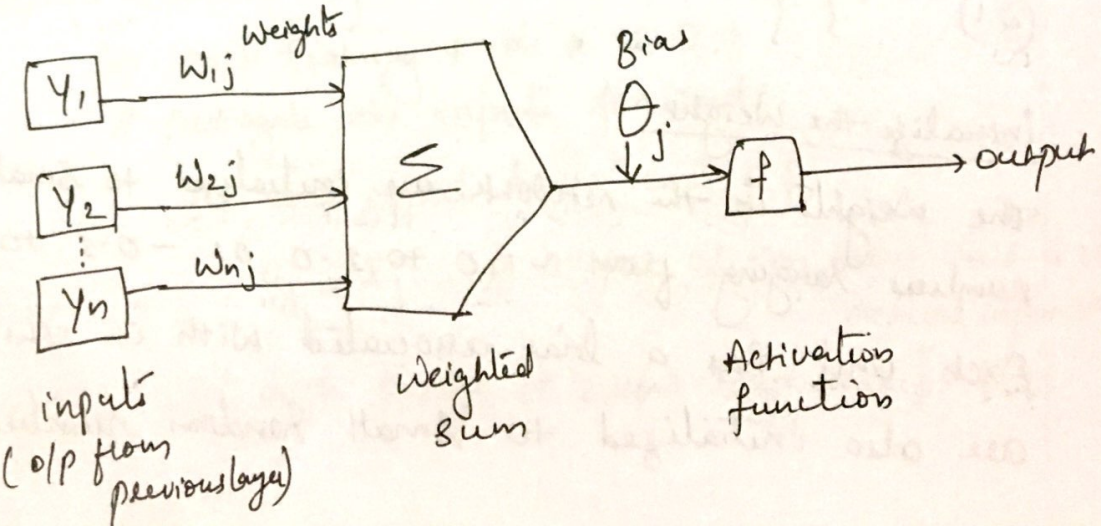
Fsit, the training tuple is fed to the input layer of the network. The inputs pass through the input units, unchanged. That is for an input unit j, its output O_j is equal to its input value I_j.

Next, the net input and output of each unit in the hidden and output layer are computed. Each connection has a weight. To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight and this is summed.

The net input I_j of the Unit (hidden/output layer) j is given by

$$I_j = \sum_i W_{ij} O_i + \theta_j$$

Where W_{ij} is the weight of the connection from Unit i in the previous layer to Unit j, O_i is the output of Unit i from the previous layer and θ_j is the bias (threshold value of that unit) of the Unit.



Scanned with CamScanner

An activation function f is applied to each unit the function used is sigmoid or logistic. For the given net input I_j to the unit j the output O_j of unit j is given by

$$O_j = \frac{1}{1+e^{-I_j}}$$

This function is referred as squashing function because it map large number of domains into small range 0 to 1

The output O_j is computed for each of the hidden layer and the output layer. which gives the network's prediction.

Backpropage the Error:-

The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction. for a unit j in the output layer the error Err_j is given by

$$Err_j = O_j(1-O_j)(T_j - O_j)$$ → output layer error

where O_j is the actual output of unit j and T_j is the the known target value of the given training Tuple.

To compute the error of a hidden layer unit j, the weighted sum of the errors of the units connected to unit j in the next layer is considered. The error of a hidden layer

Scanned with CamScanner

Unit j is given by.

$$Err_j = O_j (1 - O_j) \sum_k Err_k W_{jk}$$ → Error of hidden layer

Where W_{jk} is the weight of the connection from Unit j to Unit k in the next higher layer and Err_k is the error of Unit k (higher layer unit)

The weights and biases are updated to reflect the propagated errors. Weights are updated by the following equations.

$$\Delta W_{ij} = (l)\, Err_j\, O_i$$
$$W_{ij} = W_{ij} + \Delta W_{ij}$$

weight updation

where 'l' is the learning rate, a constant typically having value b/w 0.0 to 1.0.

Biases are updated by the following equations, where ΔO_j is the change in O_j

$$\Delta O_j = (l)\, Err_j$$
$$O_j = O_j + \Delta O_j$$

Bias updation

Case-updating :- updation of weight and bias is done after the presentation of each tuple

Epoch-updating :- updation of weight and bias is done after all of the tuples in the training set is presented.

The iteration of backpropagation algorithm stops when any of the following terminating conditions satisfies

i) All ΔW_{ij} in the previous epoch were so small

Scanned with CamScanner

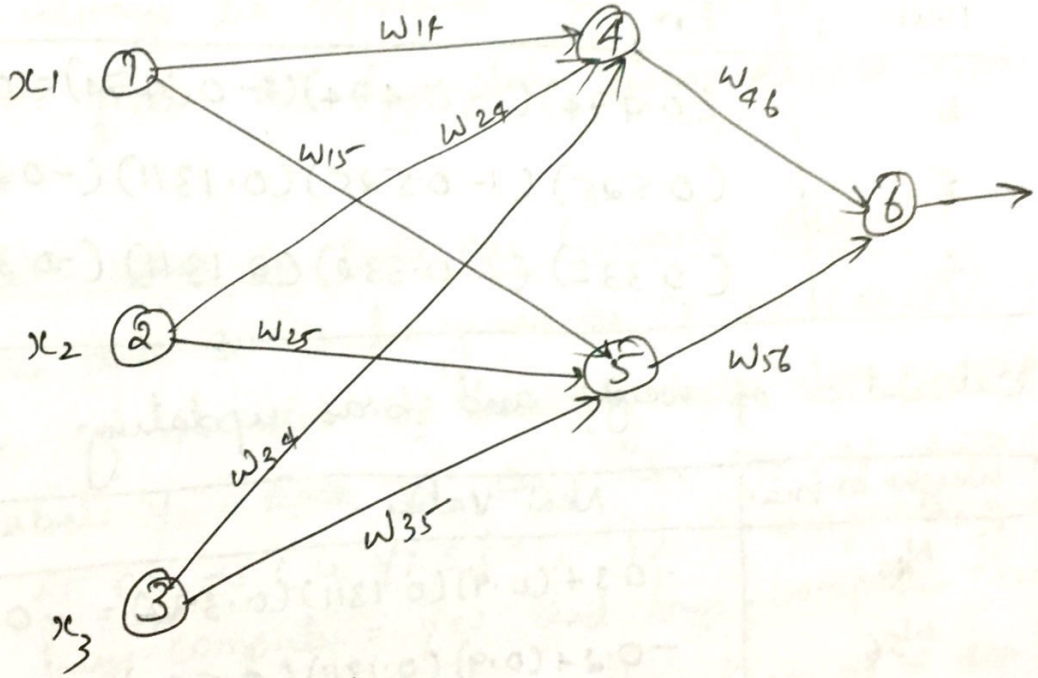
as to be below some specified threshold.

ii) The percentage of tuples misclassified in the previous epoch is below some threshold

iii) A prespecified number of epochs are expired

Example:-

Calculate the output of each unit using back propagation in which learning rate $l = 0.9$. The initial weight and bias value is given below with the first training tuple $X = (1, 0, 1)$ whose class label is 1.



Initial i/p, weight & bias.

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2

θ_4	θ_5	θ_6
-0.4	0.2	0.1

Scanned with CamScanner

Net input & output Calculation

Unit j	Net input, I_j	Output, O_j
4	$0.2+0-0.5-0.4=-0.7$	$1/(1+e^{0.7})=0.332$
5	$-0.3+0+0.2+0.2=0.1$	$1/(1+e^{-0.1})=0.525$
6	$(-0.3)(0.332)-(0.2)(0.525)+0.1=-0.105$	$1/(1+e^{0.105})=0.474$

Calculation of the error at each node.

Unit j	Err_j
6	$(0.474)(1-0.474)(1-0.474)=0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2)=-0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3)=-0.0087$

Calculation of weight and bias updating

Weight or bias	New value
W_{46}	$-0.3+(0.9)(0.1311)(0.332)=-0.261$
W_{56}	$-0.2+(0.9)(0.1311)(0.525)=-0.138$
W_{14}	$0.2+(0.9)(-0.0087)(1)=0.192$
W_{15}	$-0.3+(0.9)(-0.0065)(1)=-0.306$
W_{24}	$0.4+(0.9)(-0.0087)(0)=0.4$
W_{25}	$0.1+(0.9)(-0.0065)(0)=0.1$
W_{34}	$-0.5+(0.9)(-0.0087)(1)=-0.508$
W_{35}	$0.2+(0.9)(-0.0065)(1)=0.194$
θ_6	$0.1+(0.9)(0.1311)=0.218$
θ_5	$0.2+(0.9)(-0.0065)=0.198$
θ_4	$-0.4+(0.9)(-0.0087)=-0.408$

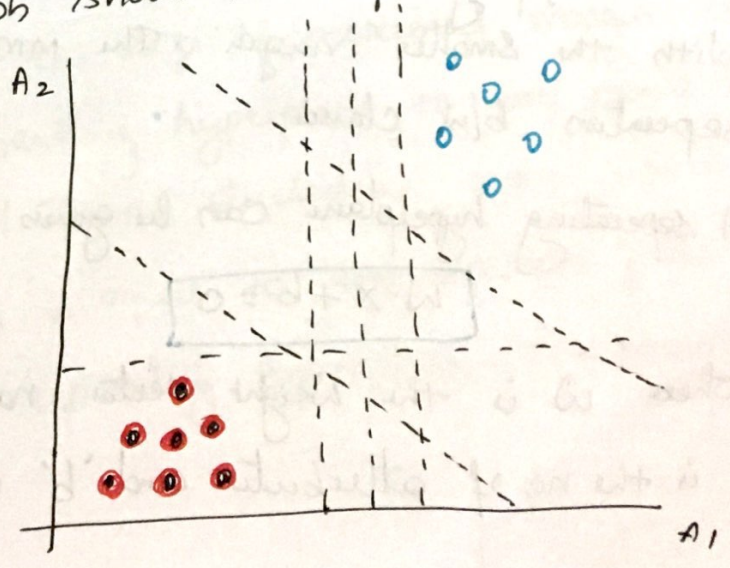
Scanned with CamScanner

Support Vector Machines

Support Vector Machines is a new method for the classification of both linear and nonlinear data. The algorithm of SVM works as follows.
It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal seperating hyperplane. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two clases always be seperated by a hyperplane
The SVM finds this hyperplane using Support vectors and margins.

The Case when the Data are Linearly Seperable

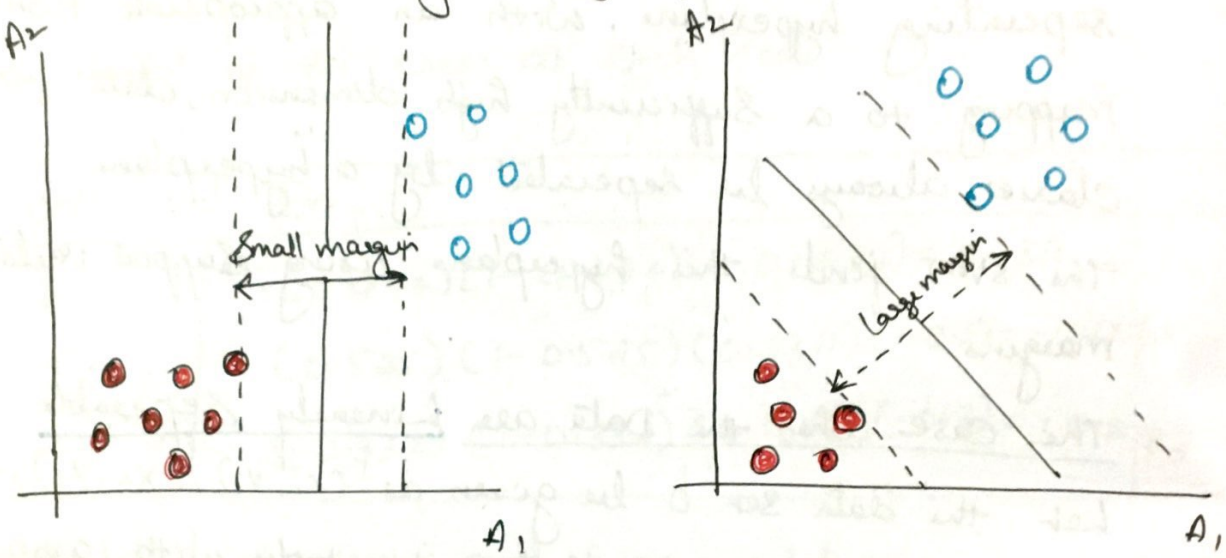
Let the data set D be given as $(X_1, Y_1), (X_2, Y_2), \ldots (X_{|D|}, Y_{|D|})$ where X_i is the set of training tuples with associated class labels y_i. Each y_i can take one of two values either +1 or -1 (ie $y_i \in \{+1, -1\}$) corresponding to the classes buys-computer = yes and buys-computer = no.
The following graph shows the seperation of a set of class



Scanned with CamScanner

A_1 and A_2 are two input attributes & the 2-D data are linearly seperable. There are infinite number of seperating lines that could be drawn. We want to find the best one with minimum classification error

A SVM found the best hyperplane by searching for the maximum marginal hyperplane (MMH)



○ class 1, $y = +1$ (buys-computer = yes)

● class 2, $y = -1$ (buys-computer = no)

The hyperplane with larger margin to be more accurate at classifying future data tuples than the hyperplane with the smaller margin. The MMH gives the largest seperation b/w classes.

A seperating hyperplane can be given as

$$W \cdot X + b = 0$$

where W is the weight vector, namely $W = \{w_1, w_2 .. w_n\}$ 'n' is the no: of attributes and 'b' is the bias.

Scanned with CamScanner

Consider a training tuples i.e $X=(x_1, x_2)$ and additional weight W_0 instead of bias b. Then the above equation can be written as

$$W_0 + W_1 x_1 + W_2 x_2 = 0$$

Any point that lies above the seperating hyperplane satisfies

$$W_0 + W_1 x_1 + W_2 x_2 > 0$$

Any point that lies below the seperating hyperplane satisfies

$$W_0 + W_1 x_1 + W_2 x_2 < 0$$

The weights can be adjusted so that the hyperplane defining the sides of the margin can be written as

$H_1 : W_0 + W_1 x_1 + W_2 x_2 \geq 1$ for $y_i = +1$ and

$H_2 : W_0 + W_1 x_1 + W_2 x_2 \leq -1$ for $y_i = -1$

Tuple fall on or above H_1 belongs to class +1 and tuples that fall on or below H_2 belongs to class -1

Combining the above two equation, we get.

$$y_i (W_0 + W_1 x_1 + W_2 x_2) \geq 1 \quad \forall i$$

Tuples that fall on hyperplane H_1 & H_2 are called Support vectors - they are equally close to MMH.

The formula for the size of maximal margin is the distance from separating hyperplane to any point on H_1 is $\frac{1}{\|W\|}$ where $\|W\|$ is the Euclidean norm of W is equal to $\sqrt{W \cdot W}$. this is equal to ~~any~~ distance from any point on H_2 on the seperating hyperplane. The maximal margin is $\frac{2}{\|W\|}$

Scanned with CamScanner

The Case when the Data are Linearly Inseperable

If the datas are not linearly seperable, then the straight line (hyperplane) cannot be drawn to seperate the classes the approach of linear SVM can be extended to create nonlinear SVMs for the classification of linearly inseperable data. Such nonlinear SVM are capable of finding nonlinear decision boundaries in input space.

Steps to convert nonlinear SVM to linear SVM.

i) Transform the original input data into a higher dimensional space using a nonlinear mapping.

ii) Search for a linear seperating hyperplane in the new space.

iii) By solving quadratic optimization peoblem using linear SVM formulations.

When searching for a linear SVM in the new higher dimensional space, the training tuples appears only in the form of dot product $\phi(x_i) \cdot \phi(x_j)$ where $\phi(x)$ is simply the nonlinear mapping function applied to transform the training tuples.

Instead of computing the dot product on the transformed data tuples, it turns out that it is mathematically equivalent to apply a kernal function $k(x_i, x_j)$ to the original input data ie

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

ie $\phi(x_i) \cdot \phi(x_j)$ can be replaced with $k(x_i, x_j)$

Scanned with CamScanner

After applying kernal functions we need to find a maximal seperating hyperplane.

Types of kernal functions used are

polynomial kernal of degree h: $k(x_i, x_j) = (x_i \cdot x_j + 1)^h$

Gaussian Radial basis function kernal: $k(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$

Sigmoid kernel : $k(x_i, x_j) = \tanh(kx_i \cdot x_j - \delta)$

Each of the above kernal functions results in different non linear classifier in input space.

Lazy Learners

The classification methods like Bayesian classification, rule-based, Backpropagation, Support vector machines are all examples of eager learners.

Eager Learners - will construct a generalization (classification) model with given set of training tuples before receiving new tuples to classify.

Lazy Learners - It simply stores the given training tuples and waits until it is given a test tuple. It will perform generalization only when it sees the test tuple and classify the tuple based on its similarity to the stored training tuples.

Lazy learners are expensive when classification or prediction is made. It requires efficient storage techniques

Scanned with CamScanner

K- Nearest Neighbour classifier

Nearest neighbour classifiers are based ~~by~~ on learning by analogy. ie by comparing given test tuple with training tuples that are similar to it. The training tuples are described by n attributes.

When given an unknown tuple, a k-nearest neighbour classifier searches the pattern space for the k-training tuples 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 $X_1 = (x_{11}, x_{12}, \ldots x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots x_{2n})$ is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

ie For each numeric attribute, we take the difference b/w the corresponding values of that attribute in tuple X_1 and in tuple X_2, Square this difference and accumulate it.

Value of each attribute is normalized and this helps to prevent attributes with initially large range from outweighting attributes with initially small range.

Scanned with CamScanner

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

$$v' = \frac{v - min_A}{max_A - min_A}$$

where min_A and max_A are the minimum and maximum value of the attribute A.

In k-nearest neighbour classification when k=1 the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space.

If attribute is not numeric :-

The simple method of classifying the tuples is by comparing the corresponding value of the attribute in tuple 'x_1' with that of tuple 'x_2'. If two are identical (ie x_1 & x_2 both are blue color) then the difference b/w is taken as 0. If both are different (x_1 blue & x_2 is Red) then difference b/w is considered as 1.

* Attribute values are missing :-

If the value of a given attribute A is missing in tuple x_1 or in tuple x_2. Suppose that each of the attribute have been mapped to the range [0,1]. If both values of x_1 & x_2 or one value of attribute A are missing then we take the difference as 1. If one value is missing and

Scanned with CamScanner

the other is present and normalized, then we can take the difference to be either $(0-v')$ or $(1-v')$, whichever is greater.

Determine the good value for k, the number of neighbours?

Starting with $k=1$, we use a test set to estimate the error rate of the classifier. This process can be repeated each time by incrementing k to allow for one more neighbour. The k value with the minimum error rate may be selected.

Case-Based Reasoning (CBR)

CBR classifier use a database of problem solutions to solve the new problems. CBR stores the tuples or cases for problem solving as complex symbolic descriptions. Business application of CBR include problem resolution for customer service help desk, Medical education etc

Prediction

Numeric prediction is the task of predicting continuous values for the given input. eg:- If we need to predict the salary of college graduate with 10 year of work experience or potential sale of a new product given its price.

Most widely approach used for numeric prediction is

Regression

Scanned with CamScanner

Regression analysis can be used to model the relationship b/w one or more independent or predictor variables and a dependent or response variable.

In datamining, predictor variables are the attributes of interest describing the tuples. In general, value of predictor variable are known. The response variable is what we want to predict

Regression analysis is good choice when all the predictor variables are continuous valued as well.

Linear Regression

Straight-line regression analysis involves a response variable y, and a single predictor variable x, It is the simplest form of regression and models y as a linear function of x, ie

$$y = b + wx$$

where the variance of y is assumed to be constant, and b and w are regression coefficient specifying the y-intercept and slope of the line. The regression coefficient, w and b can be considered as weight and the above equation can be rewrite as

$$y = w_0 + w_1 x_1$$

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

Let D be the training set consisting of values of

Scanned with CamScanner

predictor variable x, and their associated value for response variable y. The training set contains $|D|$ datapoints to form $(x_1, y_1), (x_2, y_2) \ldots (x_{|D|}, y_{|D|})$ The Regression coefficient can be estimated by eqn

$$w_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

where $\bar{x}$ is the mean value of $x_1, x_2, x_3 \ldots x_{|D|}$ and $\bar{y}$ is the mean value of $y_1, y_2 \ldots y_{|D|}$.

Example:-

The following table shows a set of paired data where x is the number of years of work experience of a college graduate and y is the corresponding salary of the graduate.

x years of experience	y salary.
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Scanned with CamScanner

$\bar{x} = 9.1$ and $\bar{y} = 55.4$ Sub $\bar{x}$ & $\bar{y}$ in equ

$$w_1 = \frac{(3-9.1)(30-55.4) + (8-9.1)(57-55.4) + \cdots + (16-9.1)(83-55.4)}{(3-9.1)^2 + (8-9.1)^2 + \cdots (16-9.1)^2}$$

$= 3.5$

$w_0 = 55.4 - 3.5 \times 9.1$

$= 55.4 - 31.85 = 23.6$

The equation for least square line is computed as

$$y = 23.6 + 3.5x.$$

With the above equation we can predict the salary of the graduate with x no: of experience years.

Multiple Linear Regression:

It is a extension of straight line regression so as to involve more than one predictor variable. It allows response variable y to be modeled as a linear function say, n predictor variables or attributes $A_1, A_2, \ldots A_n$ describing tuple X.

Our training data set, D contains data of the form (x_1, y_1) $(x_2, y_2) \ldots (x_{|D|}, y_{|D|})$ where x_i is n-dimensional training tuples with class labels y_i.

Eg: of multiple linear regression model based on two predictor attributes or variables A_1 and A_2 is

$$y = w_0 + w_1 x_1 + w_2 x_2$$

where x_1 and x_2 are the values of the attribute A_1 & A_2 in X. The method of least square is extended to solve w_0, w_1 & w_2.

Scanned with CamScanner

Nonlinear Regression

parabola or some other higher-order polynomial is more accurate model used for nonlinear data

Polynomial Regression is often of interest when there is just one predictor variable. It can be modeled by adding polynomial terms to the basic linear model By applying transformation to the variables, we can convert the nonlinear model into a linear one that can then be solved by method of least squares

Transformation of a polynomial regression model to a linear Regression model is given by.

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

To convert the above equation to linear form, we define new variables $x_1 = x$, $x_2 = x^2$, $x_3 = x^3$

The equation can be rewrite as

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

The above linear equation can be easily solved by method of least squares using software for regression analysis

polynomial Regression is a special type of multiple Regression. The addition of high-order terms like x^2, x^3 and so on which are simple functions of the single variable x can be considered equivalent to adding new independent variables.

Scanned with CamScanner

Accuracy and Error Measures

The estimation of accuracy is done by holdout, random sampling, k-fold cross-validation and bootstrap methods.

Classifier Accuracy Measures

Accuracy is measured on test set consisting of class labeled tuples that were not used to train the model. 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 pattern recognition, this is referred to as the overall recognition rate of the classifier. ie 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).

Confusion Matrix is a tool used for analyzing tuples of different class. If there is m classes then confusion matrix consist of atleast m x m size

True positiveness refer to the positive tuples that were correctly labeled by the classifier while true negative tuples are the negative tuples that were correctly labeled by the classifier.

False positives are the negative tuples that were incorrectly labeled (tuples of class buys-computer = no for which the classifier predicted buys-computer = yes)

Scanned with CamScanner

false-negatives are the positive tuples that were incorrectly labeled (tuples of class buys-computer = yes for which the classifier predicted buys-computer = no)

Sensitivity and Specificity are the two measures used to identify how well the classifier can recognize the classes of the given tuples. It is given by

$$\text{Sensitivity} = \frac{t\text{-pos}}{\text{pos}}$$

$$\text{Specificity} = \frac{t\text{-neg}}{\text{neg}}$$

$$\text{Precision} = \frac{t\text{-pos}}{(t\text{-pos} + f\text{-pos})}$$

where t-pos is the no: of true positives, pos is the number of positive tuples, t-neg is the number of true negatives, neg is the no: of negative tuples, f-pos is the no: of false positives (tuples that are incorrectly labeled).

The accuracy can be defined in terms of sensitivity and specificity as

$$\text{accuracy} = \text{Sensitivity}\,\frac{\text{pos}}{\text{pos+neg}} + \text{Specificity}\,\frac{\text{neg}}{\text{pos+neg}}$$

True positives, true negatives, false positives and false negatives are useful in assessing the cost and benefits.

Scanned with CamScanner

Predictor Error Measures

Let D^T be the test set of the form $(X_1, Y_1), (X_2, Y_2)$ $\ldots (X_d, Y_d)$, where X_i is the test tuple associated with known class label Y_i. and d is the no: of tuples in D^T It is difficult to predict the value Y_i for X_i is correct or not. Instead of checking correct or not we can check how the predicted value Y_i' is matched with actual value Y_i

Loss function measures the error between Y_i and the predicted value Y_i'

Absolute Error: $|Y_i - Y_i'|$

Squared Error: $(Y_i - Y_i')^2$

The test error rate or generalization error is the average loss over the test set. It is given by

Mean absolute error: $$\frac{\sum_{i=1}^{d} |Y_i - Y_i'|}{d}$$

Mean squared error: $$\frac{\sum_{i=1}^{d} (Y_i - Y_i')^2}{d}$$

The error can be Relative to the predicted mean value $\bar{Y}$ of Y from training data D.

Relative measure of error include.

Relative absolute error: $$\frac{\sum_{i=1}^{d} |Y_i - Y_i'|}{\sum_{i=1}^{d} |Y_i - \bar{Y}|}$$

Relative squared error: $$\frac{\sum_{i=1}^{d} (Y_i - Y_i')^2}{\sum_{i=1}^{d} (Y_i - \bar{Y}_i)^2}$$

where $\bar{Y}$ is the mean of Y_i i.e $\bar{Y} = \frac{\sum_{i=1}^{d} Y_i}{d}$

Scanned with CamScanner

MOD - V

Association Rules

Frequent patterns are the patterns that appear in the data frequently eg: a set of items such as milk and bread that appears frequently together in a transaction data set is a frequent item set.

Sequence pattern - A sequence such as buying a pc, then a digital camera and then a memory card, if it occurs frequently in a shopping history db.

Substructure can refer to different structural forms, such as subgraphs, subtrees or sublattices, which may be combined with itemset or subsequences. If a substructure occurs frequently it is called structured pattern.

Basic Concepts

Market Based Analysis:

Frequent itemset mining leads to the discovery of associations and correlation among items in large transactional or relational data set. The discovery of correlation relationship among huge amounts of business transaction records can help in many business decision making process, such as catalog design, cross-marketing and customer shopping behaviour analysis.

An typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying

Scanned with CamScanner

habits by finding associations between the different items that customers place in their shopping baskets.
Each item in the store has a Boolean variable representing presence or absence of that item in the basket. Each basket can be represented by boolean vector of values assigned to these variables.
The boolean vectors can be analyzed for buying patterns that reflect items that are frequently associated or purchased together. These patterns can be represented by association rules

Example:
The information that customers who purchase computers also tend to buy antivirus software at the same time is represented by association rule

$$\text{Computer} \Rightarrow \text{antivirus - software} \; [\text{support} = 2\% \; \text{confidence} = 60\%]$$

Association rule performance measures :-
→ confidence
→ support
→ Minimum support threshold
→ Minimum confidence threshold

A <u>support</u> of 2% for the above association rule means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.
A <u>confidence</u> of 60% means that 60% of the customers who purchased a computer also bought the software
Association Rules are considered only if they satisfy both a minimum

Scanned with CamScanner

Support threshold and a minimum confidence threshold. Such threshold can be set by users or domain experts

Frequent Itemsets, closed Itemsets and Association Rules

Let $I = \{I_1, I_2, \ldots I_m\}$ be the set of items. Let D be the set of the transactions, where each transaction T is a set of items such that $T \subseteq I$.

Each transaction is associated with an identifier called TID.

Let 'A' be the set of items.

A Transaction 'T' is said to contain A iff $A \subseteq T$

An association Rule is the implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \phi$

The Rule $A \Rightarrow B$ holds in the transaction set D with support 's', where s is the percentage of transactions in D that contains $A \cup B$ (ie both A and B). This is taken to the probability $P(A \cup B)$.

The Rule $A \Rightarrow B$ has confidence 'c' in the transaction set D, where 'c' is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability $P(B/A)$. ie

$$\text{Support}(A \Rightarrow B) = P(A \cup B)$$

$$\text{Confidence}(A \Rightarrow B) = P(B/A)$$

Rules that satisfy both a minimum Support threshold (min-sup) and minimum confidence threshold (min-conf) are called Strong.

Scanned with CamScanner

Support and Confidence values occur between 0 to 1.0.

An itemset that contain k items called k-itemset.

The occurance frequency of an itemset is the number of transaction that contain the itemset. This is called frequency or Support count

$$\text{Confidence}(A \Longrightarrow B) = P(B/A) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} = \frac{\text{Support-Count}(A \cup B)}{\text{Support-count}(A)}$$

Association Rule mining can be done in 2-step process

1. Find all frequent itemset: Each of these itemset will occur at least as frequently as a predetermined minimum Support Count, min-Sup
2. Generate Strong association Rules from the frequent itemsets: These Rules must satisfy min-Support & min-Confidence.

Challenge in mining Frequent itemset:-

Mining from a large dataset often generates a huge number of itemsets satisfying the minimum support (min-sup) threshold. If min-sup is set to low.

An itemset X is closed in a data set S if there exists no proper super itemset Y such that Y has the same support count as X in S.

An itemset X is closed frequent itemset in set S if X is both closed and frequent in S.

Scanned with CamScanner

An itemset X is a maximal frequent itemset in set S if X is frequent, and there exists no super itemset Y such that $X \subset Y$ & Y is frequent in S.

Classification of Association Rule Mining

Following are the different ways for the classification of Association Rule.

1. Based on the levels of abstraction involved in the rule set:
   - Single level association rules refer items or attributes at only one level
     buys(x, "computer") ⇒ buys(x, "Hp-printer")
   - Multilevel association rules reference items or attributes at different levels of abstraction
     buys(x, "laptop computer") ⇒ buys(x, "Hp-printer").
2. Based on the number of data dimensions involved in the rule:
   - Single dimensional association rule is an association rule in which items or attributes reference only one dimension
     buys(x, "computer") ⇒ buys(x, "antivirus software")
   - multidimensional Association Rule reference two or more dimensions
     age(x, "30--39") ∧ income(x, "42k--48k") ⇒ buys(x, "high resolution T.V")

Scanned with CamScanner

② Based on the types of values handled in the Rule

- Boolean association Rule involves association between the presence or absence of item

  buys(X, "SQLServer") ∧ (buys(X, "DM Book.") ⇒ buys(X, "DBMine

- Quantitative association Rule describes association between quantitative items or attribute

  age(X, "30..39") ∧ income(X, "42k...48k") → buys(X, "PC")

③ Based on the kinds of pattern to be mined:

- frequent itemset mining is the mining of frequent itemset from transactional or relational data sets.
- Sequential pattern mining searches for frequent subsequences in a sequence data set, where a sequence records an ordering of events.
- Structured pattern Mining searches for frequent substructures in a structured data set.

④ Based on various extensions to association mining:

- correlation, causality analysis

  Association does not necessarily imply correlation or causality
- Max patterns and closed itemsets
- constraints enforced

Frequent itemset Mining

Following are the different approaches used for the frequent itemset Mining

Scanned with CamScanner

1) Apriori Algorithm
2) Fp- Growth Algorithm.

Apriori Algorithm

Apriori Algorithm uses prior knowledge of frequent itemset properties.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1) itemsets.

First the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L_1. Next L_1 is used to find L_2 which is used to find L_3 and so on, until no more frequent k-itemsets can be found. The finding of each L_k requires one full scan of the database.

Apriori principle :-

Any subset of a frequent itemset must be frequent.

Apriori property follows a two step process :—

→ Join step : C_k is generated by joining L_{k-1} with itself.

→ Prune step : Any (k-1) itemset that is not frequent cannot be a subset of a frequent k-itemset.

Scanned with CamScanner

Algorithm Apriori :

Input :

- D, a database of transactions
- min_sup, the minimum support count threshold.

Output :

L, frequent itemsets in D.

Method :

1. L_1 = find_frequent_1_itemsets(D);
2. for ($k = 2$; $L_{k-1} \neq \phi$; $k+1$) {
3. C_k = apriori_gen(L_{k-1}).
4. for each transaction $t \in D$ { // scan D for counts
5. C_t = subset(C_k, t); // get the subset of t that are candidates
6. for each candidate $c \in C_t$
7. c.count++;
8. }
9. $L_k = \{ c \in C_k \mid c.count \geq min_sup \}$
10. }
11. return $L = L \cup L_k$

procedure apriori_gen(L_{k-1} ; frequent $(k-1)$ itemsets)

1. for each itemset $l_1 \in L_{k-1}$
2. for each itemset $l_2 \in L_{k-1}$
3. if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ then {
4. $c = l_1 \bowtie l_2$: // join step : generate candidate
5. if has_infrequent_subset(c, L_{k-1}) then

Scanned with CamScanner

6. delete c; // prune step: Remove unfruitful candidate
7. else add c to C_k;
8. }
9. Return C_k;

procedure has-infrequent-subset (c: candidate k-itemset;
L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
1) for each (k-1)-subset s of c
2) . if $s \notin L_{k-1}$ then
3) Return TRUE
4) Return FALSE.

Generating Association Rule from frequent itemset

Association Rule can be generated for the frequent itemset which satisfy both minimum-support and minimum-confidence

$$\text{Confidence}(A \Rightarrow B) = P(B/A) = \frac{\text{Support-Count}(A \cup B)}{\text{Support-count}(A)}$$

The conditional probability is expressed in terms of itemset support-count

Based on the above equation the association rule can be generated as follows.

⇒ For each frequent itemset l, generate all nonempty subset of l

Scanned with CamScanner

⇒ For every nonempty subset s of l, output the rule "$s \Rightarrow (l-s)$" if $\frac{\text{Support-count}(l)}{\text{Support-count}(s)} \geq$ min-conf,

where min-conf is the minimum confidence threshold.

Example ?

Consider the following ds in which |D| = 9 and minimum support is taken as 2. Use Apriori Algm for finding frequent itemsets in D

TID	List of item-IDs
T100	I_1, I_2, I_5
T200	I_2, I_4
T300	I_2, I_3
T400	I_1, I_2, I_4
T500	I_1, I_3
T600	I_2, I_3
T700	I_1, I_3
T800	I_1, I_2, I_3, I_5
T900	I_1, I_2, I_3

Step 1: Generation and pruning of candidate 1-itemset

→ Read the database to count the support of 1-itemsets

→ The set of frequent 1-itemsets, L_1 consists of the candidate 1-itemsets satisfying minimum-support

Scanned with CamScanner

Step 2: Generation and pruning of candidate 2-itemsets, C from L_1

→ To discover the set of frequent 2-itemsets, L_2 the algorithm uses the join L_1 on L_1 to generate a candidate set of 2-itemsets, C_2.

Therefore in the candidate generation step, we get.

(a) join : $C_2 = L_1 \bowtie L_1 = \{\{I_1\}, \{I_2\}, \{I_3\}, \{I_4\}, \{I_5\}\} \times \{\{I_1\}, \{I_2\}, \{I_3\}, \{I_4\}, \{I_5\}\}$

$= \{I_1, I_2\}, \{I_1, I_3\}, \{I_1, I_4\}, \{I_1, I_5\}, \{I_2, I_3\}, \{I_2, I_4\}, \{I_2, I_5\}, \{I_3, I_4\}, \{I_3, I_5\}, \{I_4, I_5\}\}$

b) Prune using the Apriori property:

Here all nonempty subsets of a frequent itemset is also frequent. Therefore

$C_2 = \{\{I_1, I_2\}, \{I_1, I_3\}, \{I_1, I_4\}, \{I_1, I_5\}, \{I_2, I_3\}, \{I_2, I_4\}, \{I_2, I_5\}, \{I_3, I_4\}, \{I_3, I_5\}, \{I_4, I_5\}\}$

after pruning

→ Next, the transactions in D are scanned and the support count of each candidate itemset in C_2 is accumulated

→ The set of frequent 2-itemsets, L_2 is then determined, consisting of those candidates 2-itemsets in C_2 having minimum support.

Scanned with CamScanner

Step 3 : Generation and pruning of candidate 3-itemsets, C_3, from L_2 using the Apriori property.

a) Join $C_3 = L_2 \bowtie L_2 = \{\{I_1, I_2\}, \{I_1, I_3\}, \{I_1, I_5\}, \{I_2, I_3\}, \{I_2, I_4\}, \{I_2, I_5\}\} \times \{\{I_1, I_2\}, \{I_1, I_3\}, \{I_1, I_5\}, \{I_2, I_3\}, \{I_2, I_4\}, \{I_2, I_5\}\}$

$= \{I_1, I_2, I_3\}, \{I_1, I_2, I_5\}, \{I_1, I_3, I_5\}, \{I_2, I_3, I_5\}, \{I_2, I_4, I_5\}$

b) prune using Apriori property. All nonempty subsets of a frequent itemset must also be frequent.

- The 2-item subset of $\{I_1, I_2, I_3\}$ are $\{I_1, I_2\}$, $\{I_1, I_3\}$ and $\{I_2, I_3\}$. All 2-item subsets of $\{I_1, I_2, I_3\}$ are members of L_2. Therefore the itemset $\{I_1, I_2, I_3\}$ are kept in C_3
- The 2-item subsets of $\{I_1, I_2, I_5\}$ are $\{I_1, I_2\}$, $\{I_1, I_5\}$ and $\{I_2, I_5\}$. All 2-item subsets of $\{I_1, I_2, I_5\}$ are members of L_2. $\therefore$ keep $\{I_1, I_2, I_5\}$ in C_3
- The 2-item subset $\{I_1, I_3, I_5\}$ are $\{I_1, I_3\}, \{I_3, I_5\}$, $\{I_1, I_5\}$ is not a member of L_2 and so it is not frequent. $\therefore$ Remove $\{I_1, I_3, I_5\}$ from C_3
- The 2-item subset $\{I_2, I_3, I_4\}$ are $\{I_2, I_3\}, \{I_3, I_4\}$, $\{I_2, I_4\}$ is not a member of L_2 and is not frequent. $\therefore$ Remove $\{I_2, I_3, I_5\}$ from C_3

Scanned with CamScanner

• The 2-item subset of $\{I_2, I_4, I_5\}$ are $\{I_2, I_4\}$, $\{I_2, I_5\}$ and $\{I_4, I_5\}$, $\{I_4, I_5\}$ is not a member of L_2 so it is not frequent. Therefore remove $\{I_2, I_4, I_5\}$ from C_3

(c) Therefore $C_3 = \{\{I_1, I_2, I_3\}, \{I_1, I_2, I_5\}\}$ after pruning

Step 4: Generation and pruning of candidate 4-itemsets, C_4

The algorithm uses $L_3 \bowtie L_3$ to generate a candidate set of 4-itemsets, C_4. Although the join results in $\{\{I_2, I_1, I_3, I_5\}$, this itemset is pruned because its subset $\{\{I_2, I_3, I_5\}\}$ is not frequent. Thus $C_4 = \phi$ and the algorithm terminates.

Scan D for count of each candidate →

C_1

itemset	Sup.count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

Compare candidate support Count with minimum support count →

L_1

itemset	Sup.count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

Generate C_2 Candidate from L_1

C_2

itemset
$\{I_1, I_2\}$
$\{I_1, I_3\}$
$\{I_1, I_4\}$
$\{I_1, I_5\}$
$\{I_2, I_3\}$
$\{I_2, I_4\}$
$\{I_2, I_5\}$
$\{I_3, I_4\}$
$\{I_3, I_5\}$
$\{I_4, I_5\}$

Scan D for Count of each Candidate →

C_2

itemset	Sup.count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_4\}$	1
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2
$\{I_3, I_4\}$	0
$\{I_3, I_5\}$	1
$\{I_4, I_5\}$	0

Scanned with CamScanner

Compare C_2 with min Sup count →

L_2

itemset	Sup. count
$\{I_1, I_2\}$	9
$\{I_1, I_3\}$	9
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2

$C_3 = L_2 \bowtie L_2$

C_3

itemset
$\{I_1, I_2, I_3\}$
$\{I_1, I_2, I_5\}$

C_3

itemset	Sup. count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

Compare C_3 with min-sup count

L_3

Itemset	Sup. count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

Generating Association Rules from Frequent Itemsets

Association Rule can be generated directly from the frequent itemsets in a database.

Strong Association Rules are those which satisfy both minimum Support and minimum Confidence. This can done by the following equation.

$$\text{Confidence}(A \Longrightarrow B) = P(B|A) = \frac{\text{Support-Count}(A \cup B)}{\text{Support-Count}(A)}$$

Scanned with CamScanner

The conditional probability is expressed in terms of itemset support-count where support-count (A∪B) is the no: of transactions containing the itemsets (A∪B) and support-count is the no: of transactions containing the itemset A. Based on this equation the association rules can be generated as follows

* for each frequent itemset l, generate all nonempty subsets of l
* For every nonempty subset s of l, output the rule $s \Rightarrow (l-s)$ if $\frac{\text{support-count}(l)}{\text{support-count}(s)} \geq$ min-confidence

Example :-

Consider the frequent itemset $l = \{I_1, I_2, I_5\}$ with min-confidence threshold is 70%. Write the association rules that can be generated from frequent itemset. The nonempty subsets of l are $\{I_1, I_2\}$, $\{I_1, I_5\}$, $\{I_2, I_5\}$, $\{I_1\}$, $\{I_2\}$ and $\{I_5\}$. The resulting association rule are

$I_1 \wedge I_2 \Rightarrow I_5$	Confidence = 2/4 = 50%
$I_1 \wedge I_5 \Rightarrow I_2$	Confidence = 2/2 = 100% ✓
$I_2 \wedge I_5 \Rightarrow I_1$	Confidence = 2/2 = 100% ✓
$I_1 \Rightarrow I_2 \wedge I_5$	Confidence = 2/6 = 33%
$I_2 \Rightarrow I_1 \wedge I_5$	Confidence = 2/7 = 29%
$I_5 \Rightarrow I_1 \wedge I_2$	Confidence = 2/2 = 100% ✓

Since the min-confidence is 100% only the associatin

Scanned with CamScanner

Rule $I_1 \wedge I_3 \rightarrow I_2$, $I_2 \wedge I_3 \Rightarrow I_1$ and $I_5 \rightarrow I_2 \wedge I_2$ are strong

Mining Frequent itemsets Without Candidate generation (Fp-Growth)

The Apriori Algm reduces the size of candidate sets, leading to good performance gain.

It has the following limitations

1. It may need to generate a huge number of candidate sets.
2. It may need to repeatedly scan the database and check a large-set of candidates by pattern matching

Frequent pattern growth or Fp-growth is a method that mines the complete set of frequent itemsets without candidate generation. It adopts a divide and conquer method.

First it compresses the database representing frequent items into a frequent pattern tree

Then it divides the compressed db into a conditional database.

Example:-

Consider the following data set and construct FP tree

Using Fp-growth Algorithm and create conditional pattern base.

Scanned with CamScanner

TID	List of items
T100	I_1, I_2, I_5
T200	I_2, I_4
T300	I_2, I_3
T400	I_1, I_2, I_4
T500	I_1, I_3
T600	I_2, I_3
T700	I_1, I_3
T800	I_1, I_2, I_3, I_5
T900	I_1, I_2, I_3

Step 1 : Scan the db D for supper count of each candidate.

item	Sup.count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

Step 2 : Sort the item in descending order

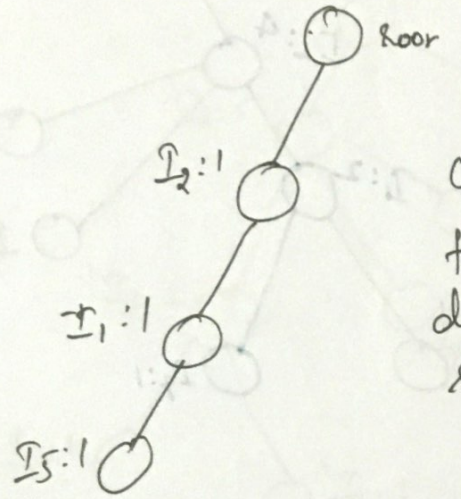
item	Sup.count
I_2	7
I_1	6
I_3	6
I_4	2
I_5	2

Scanned with CamScanner

Step 3: FP Tree Construction

1) Create the root of the tree as NULL
2) Scan database D
3) The item in each transaction processed in L-order i.e the order in descending support count and a branch is created for each transaction

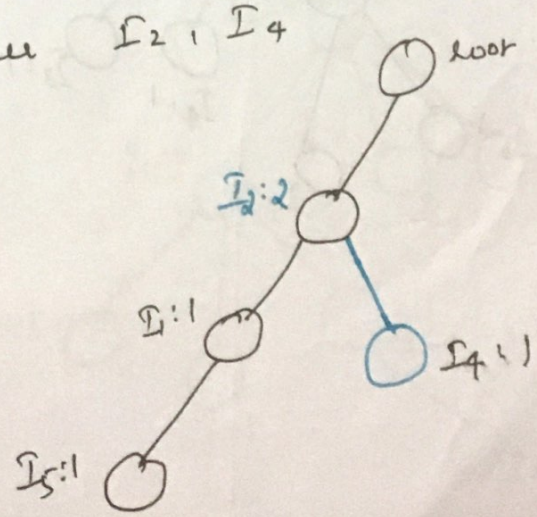
For the first transaction $T_{100} = I_1, I_2, I_5$ which contains 3 items I_2, I_1, I_5 in L-order. It leads to construction of first branch of tree with 3 node.



Giving support count as 1 for each item ordered in descending according to the support-count value

I_2 is linked to Root as I_1 & I_5 are linked as child of I_2.

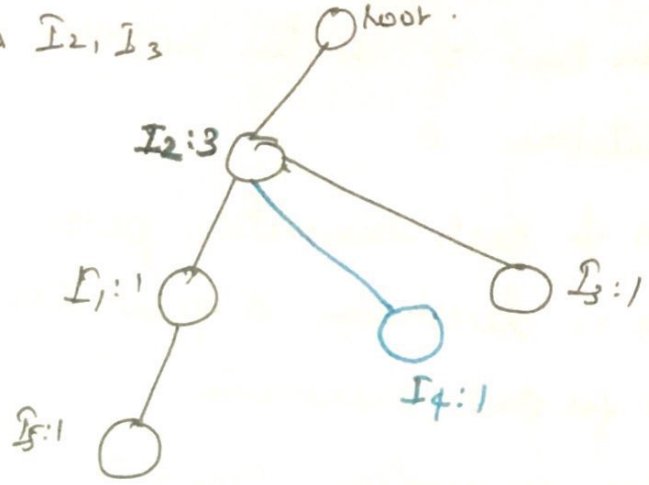
For $T_{200} = I_2, I_4$
In L-order I_2, I_4



Scanned with CamScanner

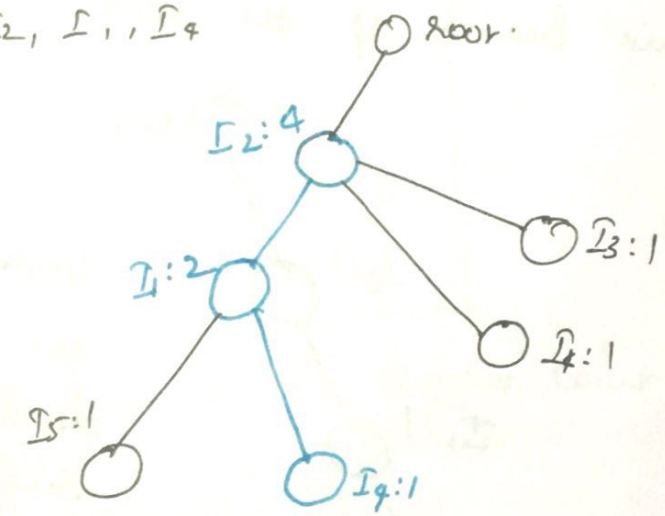
Scan T_{300} = I_2, I_3.

In L-order I_2, I_3



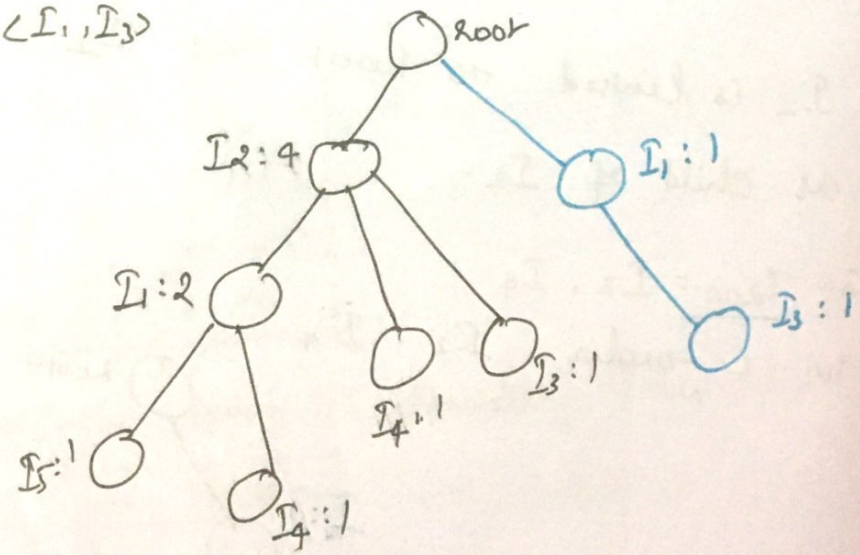
Scan T_{400} = I_1, I_2, I_4

L-order I_2, I_1, I_4



Scan T_{500} = I_1, I_3

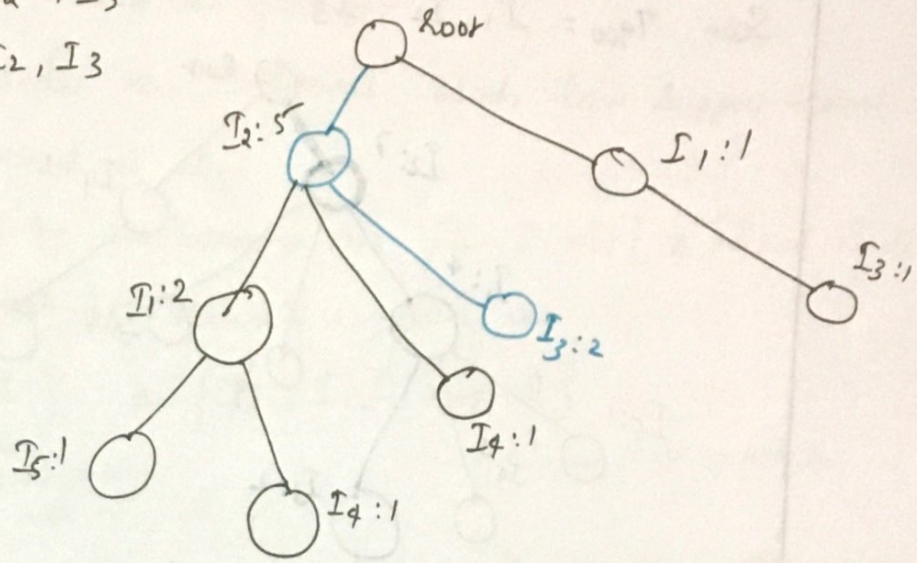
L-order $\langle I_1, I_3 \rangle$



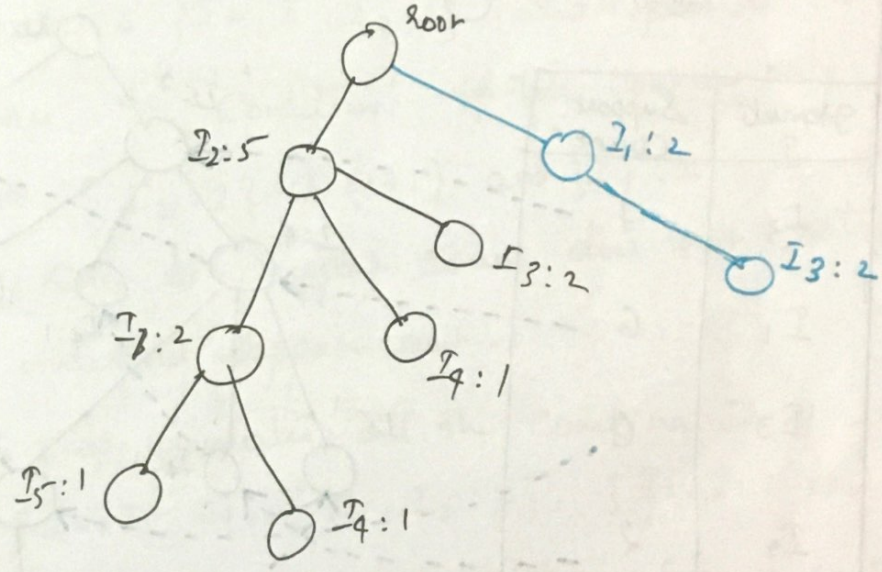
Scanned with CamScanner

Scan T_{600} = I_2 , I_3

L-order I_2, I_3

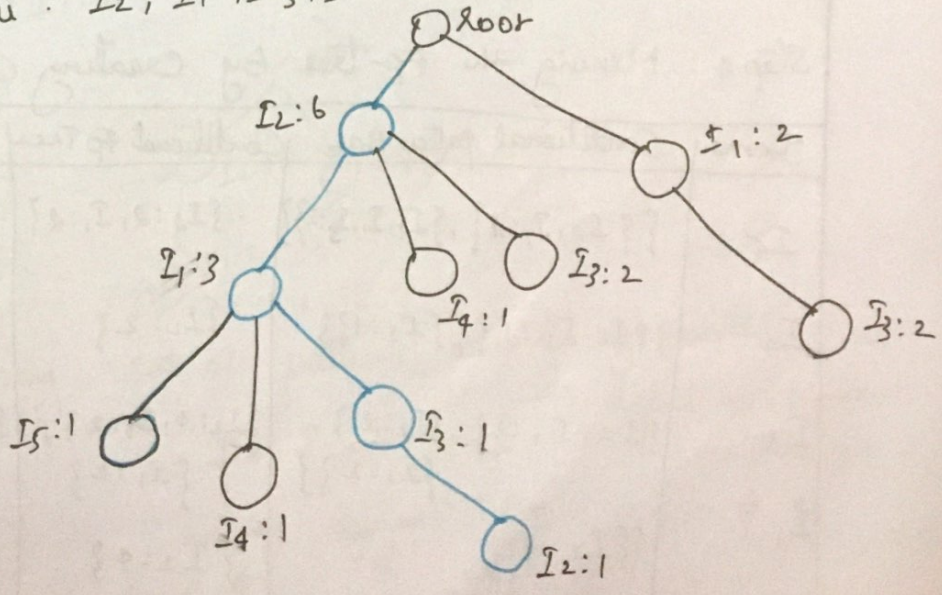


Scan T_{700} = I_1 , I_3



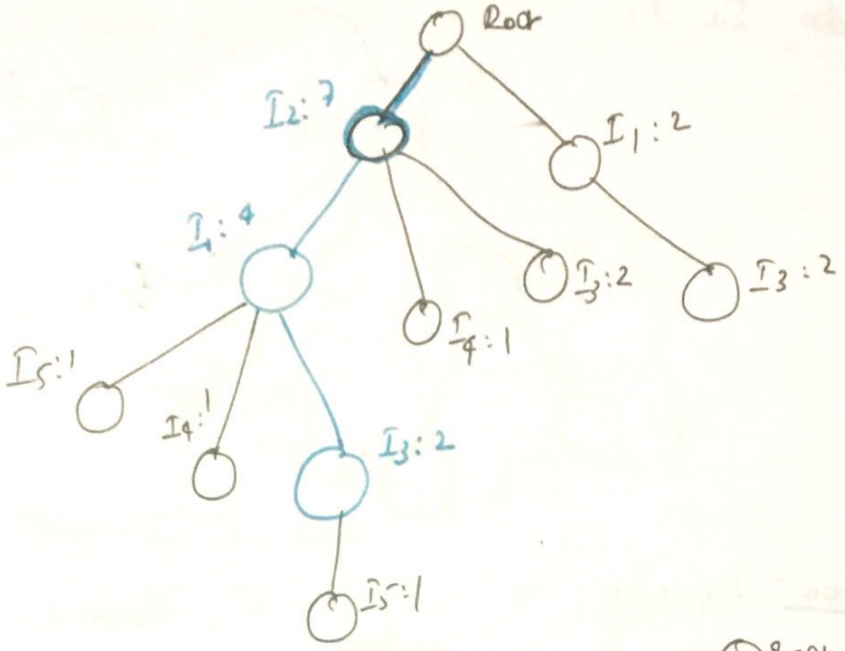
Scan T_{800} = I_1, I_2, I_3, I_5

L-order : I_2, I_1, I_3, I_5

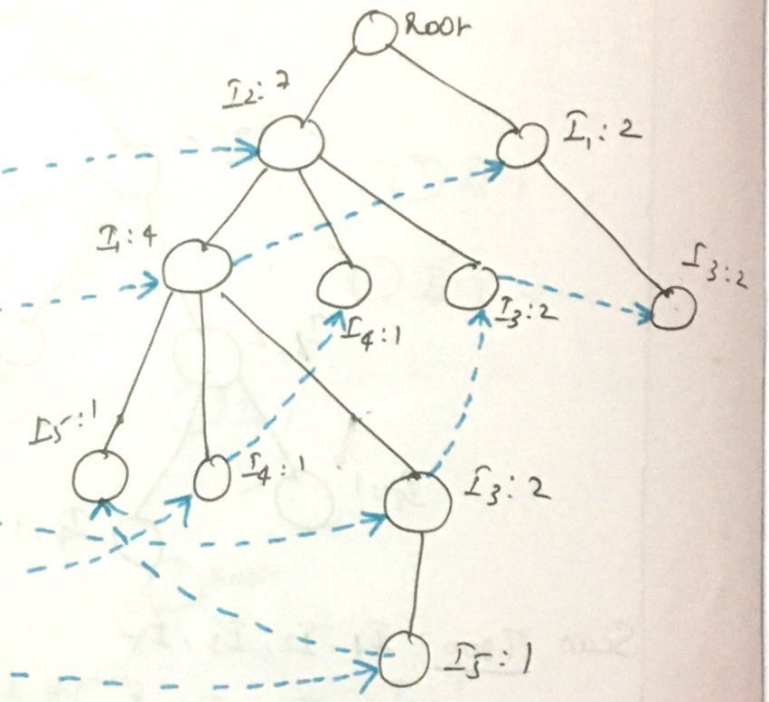


Scanned with CamScanner

Scan $T_{900} = I_1, I_2, I_3$



Itemset	Support Count
I_2	7
I_1	6
I_3	6
I_4	2
I_5	2



Step 4 : Mining the Fp-tree by creating conditional pattern base

Items	Conditional pattern Base	Conditional Fp Tree	Frequent pattern Generation
I_5	$\{\{I_2, I_1 : 1\}, \{I_2, I_1, I_3 : 1\}\}$	$\{I_2 : 2, I_1 : 2\}$	$\{I_2, I_5 : 2\}, \{I_1, I_5 : 2\}, \{I_2, I_1, I_5 : 2\}$
I_4	$\{\{I_2, I_1 : 1\}, \{I_2 : 1\}\}$	$\{I_2 : 2\}$	$\{I_2, I_4 : 2\}$
I_3	$\{\{I_2, I_1 : 2\}, \{I_2 : 2\}, \{I_1 : 2\}\}$	$\{I_2 : 4, I_1 : 2\}, \{I_1 : 2\}$	$\{I_2, I_3 : 4\}, \{I_1, I_3 : 4\}, \{I_2, I_1, I_3 : 2\}$
I_1	$\{\{I_2 : 4\}\}$	$\{I_2 : 4\}$	$\{I_2, I_1 : 4\}$

Scanned with CamScanner

The fp-tree is mined as follows.

Initially consider the itemset with low support-count.

So the first itemset is I_5

The paths formed by the branches are $\{I_2, I_1, I_5\}$ & $\{I_2, I_1, I_3, I_5\}$

The support-count of branch is given as

$\{I_2, I_1, I_5 : 1\}$ & $\{I_2, I_1, I_3, I_5 : 1\}$

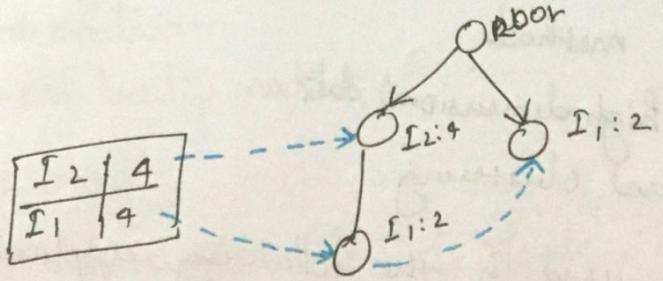
↓ Support count of node I_5

Support-count of node I_5

Considering I_5 as suffix. Its corresponding prefixes are $\{I_2, I_1 : 1\}$ & $\{I_2, I_1, I_3 : 1\}$, which form its conditional pattern base. Its conditional fp-tree contains only a single path. $\{I_2 : 2\}$ i.e (1+1) and $\{I_1 : 2\}$ I_3 is not included since its support count does not satisfy the minimum threshold support-count.

The single path generates all the combinations of frequent patterns: $\{I_2, I_5 : 2\}$, $\{I_1, I_5 : 2\}$, $\{I_2, I_1, I_5 : 2\}$.

For the item I_3, conditional pattern base is $\{\{I_2, I_1 : 2\}, \{I_2 : 2\}, \{I_1 : 2\}\}$. It has two branch.



The following set of patterns can be generated

$\{I_2, I_3 : 4\}$ → 2 + 2

$\{I_2, I_1, I_3 : 2\}$ → taking min sup-count out of 4 & 2 is 2

Supp-count = 4 | Support count = 2 | Support count = 4

$\{I_1, I_3 = 4\}$

$2 + 2$.

* FP growth method is efficient
* It is scalable for mining both long and short frequent patterns
* It is faster than Apriori Algm.

Cluster Analysis:

Clustering is a process of grouping the data into classes or clusters, so that objects within the cluster have high similarity in comparison to one another but are very dissimilar to objects of other clusters.

Clustering techniques are divided as follows.

→ partitioning Methods
→ Hierarchical Methods
→ Density based Methods.
→ Grid based methods
→ Model-based methods
→ Methods for high dimensional data
→ Constraint based clustering

Cluster analysis is used in the following applications

- Market Research
- pattern Recognition
- Data Analysis

Scanned with CamScanner

- Image processing
- Used to categorize the genes.
- Used to identify the area of similar land use in an earth observation

Clustering is also called data segmentation in some applications because clustering partitions a large data set into groups according to their similarity.
Clustering is also used for outlier detection
Clustering is an example for unsupervised learning.
Clustering is a form of learning by observation rather than learning by examples.

Types of Data in cluster Analysis

In clustering there are different types of data. Suppose that a data set to be clustered contains 'n' objects which may represent persons, houses, countries etc.
Clustering Algorithms typically operate on any of the following data structures.

→ Data matrix
→ Dissimilarity matrix

Data matrix

This matrix represents 'n' objects, such as persons with p variables such as age, height, weight, gender & so on. The structure is in the form of Relational Table or n by p matrix. It is called 2-mode matrix since rows & columns of data matrix is having different entities

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

Dissimilarity matrix

This matrix stores a collection of proximities that are available for all pairs of n objects. It is represented by n-by-n table. This is called one-mode matrix.

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

where $d(i,j)$ is the ~~dist~~ measure of dissimilarity between objects i and j.

Interval-Scaled Variables

Interval-scaled variables are continuous measurements of a roughly linear scale. eg:- Weight and height, latitude and longitude coordinates.

The measurement unit used can affect the cluster analysis Standardizing the measurement attempts to give all variables an equal weight.

The data of a variable can be standardized by

Scanned with CamScanner

Converting orginal measurements to unitless variables

Standarization for the measurement for a variable f can be peyormed as

1. Calculate the mean absolute deviation S_f:

$$S_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|\right)$$

where $x_{1f}, \cdots x_{nf}$ are n measurements of f, and m_f is the mean value of f, that is

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \cdots + x_{nf}\right)$$

2. Calculate the standarized measurement or z-score

$$Z_{if} = \frac{x_{if} - m_f}{S_f}$$

S_f is more robust to outlier

The dissimilarity between the objects described by interval scaled variables is typically computed based on the distance between each pair of objects. The popular distance measure is Euclidean distance, which is defined as

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

where $i = (x_{i1}, x_{i2}, \cdots, x_{in})$ and $j = (x_{j1}, x_{j2}, \cdots x_{jn})$ are the 2 n-dimensional data objects.

Another method to find dissimilarity is Manhattan distance which can be defined by

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

Both the Euclidean & Manhattan distance satisfy the following requirements of distance functions

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

Example: -

Let $x_1 = (1,2)$ and $x_2 = (3,5)$ represent two objects.

The Euclidean distance between the two is $\sqrt{2^2 + 3^2} = 3.61$

The manhattan distance between the two is $2+3=5$

Binary variable

A binary variable has only two state 0 or 1, where 0 means that the variable is absent and 1 means that it is present.

The dissimilarity between the two binary variables can be computed as ~~any one of the~~ following approach.

→ dissimilarity matrix

If all variables are thought of as same weight then we have 2x2 Contingency table as follows.

Scanned with CamScanner

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

where 'q' is the number of variables that equal 1 for both object i and j

'r' is the number of variables that equal 1 for object i but '0' for object j

's' is the number of variables that equal 0 for object i but equal 1 for object j

't' is the number of variables that equal 0 for both object i and j

The total number of variables is p, where $p = q+r+s+t$.

A binary variable is <u>Symmetric</u> if both of its states are equally valuable and carry the same weights, that is there is no preference on which outcome should be coded as 0 or 1. eg: gender: male or female.

<u>Symmetric binary dissimilarity</u> between the objects i and j are measured as

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

Scanned with CamScanner

The no: of matches 't' in the above dissimilarity is considered as unimportant and thus it is ignored in the computation.

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

The similarity between the objects i and j or sim(i,j) can be computed as

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

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

Example :-

Consider the following patient Record table which contains the attribute name, gender, fever, Cough, test-1, test-2, test-3 and test-4 where name is an object identifier, gender is a symmetric attribute and the remaining attributes are asymmetric binary.

For asymmetric attribute values let the values Y (yes) and P (positive) be set to 1 and the value N (no or negative) is set to 0.

The distance between the objects (patients) is computed based only on the asymmetric variables

name	gender	fever	Cough	test-1	test-2	test-3	test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	Y	N	N	N	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Scanned with CamScanner

The distance between each pair is given as

Mary

$$\text{Jack}\begin{bmatrix} 2\ (q) & 0\ (r) \\ 1\ (s) & 3\ (t) \end{bmatrix}$$

Jim

$$\text{Jack}\begin{bmatrix} 1\ (q) & 1\ (r) \\ 1\ (s) & 3\ (t) \end{bmatrix}$$

Jim

$$\text{Mary}\begin{bmatrix} 1\ (q) & 2\ (r) \\ 1\ (s) & 2\ (t) \end{bmatrix}$$

$$d(\text{Jack, Mary}) = \frac{r+s}{q+r+s} = \frac{0+1}{2+0+1} = \frac{1}{3} = \underline{0.33}$$

$$d(\text{Jack, Jim}) = \frac{1+1}{1+1+1} = \frac{2}{3} = \underline{0.67}$$

$$d(\text{Mary, Jim}) = \frac{2+1}{1+1+2} = \frac{3}{4} = \underline{0.75}$$

Categorical Variable

A categorical variable is a generalization of binary variable in that it can take on more than two states.

Let the number of states of a categorical variable be M. The states can be denoted by letters, symbols, or a set of integers such as 1, 2, ... M. Notice that such integers are used just for data handling and do not represent any specific ordering.

Dissimilarity between objects described by categorical variables can be ~~described~~ given as

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

Scanned with CamScanner

where 'm' is the number of matches (ie number of variable for which i and j are in the same state) and p is the total number of variables. Weight can be assigned to increase the effect of m or to assign greater weight to the matches in variables having a larger number of states.

Object identifier	test - 1 (categorical)
1	Code - A
2	Code - B
3	Code - C
4	code - A

In the above given sample data test-1 is the categorical data and the dissimilarity matrix is given below.

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

In the above given example p=1. So that d(i,j) evaluates to 0 if objects i and j match and 1 if the objects differ. Thus

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

Scanned with CamScanner

In the 2nd row Code B is compared with Code - A. It is different so the value is 1

In the 4th row 1st element Code - A is compared with Code - A so the value is 0.

Ordinal Variables

A discrete ordinal variable resembles a categorical variable, except that the M states of the ordinal value are ordered in a meaningful sequence. Ordinal variables are very useful for registering subjective assessments of qualities that cannot be measured objectively.

~~Suppose that an ordinal variable f has Mf states. These ordered states define the ranking 1, ..., Mf~~

Suppose that f is a variable from a set of ordinal variables describing 'n' objects. The dissimilarity computation with respect to f involves the following steps:

1. The value of f for the i^{th} object is x_{if}, and f has M_f ordered states, representing the ranking $1, \ldots M_f$. Replace each x_{if} by its corresponding rank $r_{if} \in \{1, \ldots M_f\}$

2. Since each ordinal variable can have a different number of states, it is often necessary to map the range of each variable onto $[0.0, 1.0]$ so that each variable has equal weight. This can be achieved by replacing the rank r_{if} of the i^{th} object in the f^{th} variable by

Scanned with CamScanner

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

3. Dissimilarity can then be computed using any of the distance measures described for interval-scaled variables using Z_{if} to represent the f value for the i^{th} object.

Example:- Consider the following sample data

Object identifier	test-2. (ordinal)	Rank	Normalize value
1	excellent	3	1.0
2	fair	1	0.0
3	good	2	0.5
4	excellent	3	1.0

There are 3 states for test-2 namely fair, good and excellent that is $M_f = 3$.

Step 1: Replace each value in test-2 by its rank.

fair = 1 (Rank)
good = 2
excellent = 3

The four object can be ranked as 3, 1, 2, 3.

Step 2:. Normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5 and rank 3 to 1.0.

Step 3: By using Euclidean distance the distance b/w the objects can be found

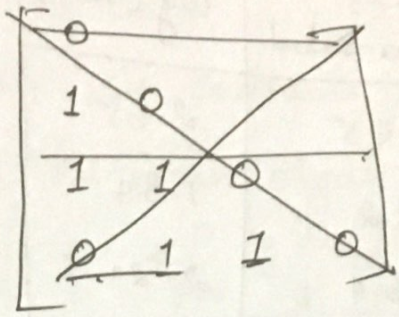
The matrix is given as below.

For the 2nd row $\sqrt{(0.0 - 1.0)^2} = 1$ (Euclidean distance)

Scanned with CamScanner

For the 3rd row $\sqrt{(0.5-1.0)^2} = \underline{\underline{0.5}}$

The matrix of dissimilarity is given by.



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

Ratio-scaled Variables

A ratio-scaled variable makes a positive measurement on a nonlinear scale, such as Ae^{Bt} or Ae^{-Bt} where A & B are positive constants

There are 3 methods to handle ratio-scaled variables for computing the dissimilarity between objects.

1) Treat ratio-scaled variables like interval-scaled variables.

2) Apply logarithmic transformation to a ratio-scaled variable f having value x_{if} for object i by using the formula $y_{if} = \log(x_{if})$. The y_{if} values can be treated as interval-valued.

3) Treat x_{if} as continuous ordinal data and treat their ranks as interval-valued.

Scanned with CamScanner

Example:-

Consider the following sample data.

Object identifier	test-3 Ratio-scaled	log (test-3)
1	445	2.65
2	22	1.34
3	164	2.21
4	1,210	3.08

Finding the Euclidean distance of each object, the dissimilarity matrix of the same data is given below.

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

$d(2,1) = \sqrt{(1.34 - 2.64)^2} = 1.31$

$d(3,1) = \sqrt{(2.21 - 2.65)^2} = 0.44$

$d(3,2) = \sqrt{(2.21 - 1.34)^2} = 0.87$

$d(4,1) = \sqrt{(3.08 - 2.65)^2} = 0.43$

$d(4,2) = \sqrt{(3.08 - 1.34)^2} = 1.74$

$d(4,3) = \sqrt{(3.08 - 2.21)^2} = 0.87$

Variables of Mixed Types

The dissimilarity between the objects of mixed variable types can be computed by two approach.

i) To group each kind of variables together, performing a seperate cluster analysis for each variable type.

ii) To process all variable types together, performing a single cluster analysis. One such techniques

Scanned with CamScanner

Combines the different variables into a single dissimilarity matrix bringing all of the meaningful variables onto a common scale of the interval [0.0, 1.0]

Suppose the data set contains 'p' variables of mixed type, the dissimilarity between objects i and j are defined as

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

where the indicator = $\delta_{ij}^{(f)} = 0$ if either

(i) x_{if} or x_{jf} is missing.

(ii) $x_{if} = x_{jf} = 0$ & f is symmetric binary

otherwise

$\delta_{ij}^{(f)} = 1$.

The contribution of variable f to the dissimilarity between i and j that is $d_{ij}^{(f)}$ is computed dependent on its type

→ If f is interval-based: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$ where 'h' runs over all nonmissing objects for variables f.

→ If f is binary or categorical: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise $d_{ij}^{(f)} = 1$

→ If f is ordinal: Compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$

treat z_{if} as interval scaled

Scanned with CamScanner

Example :- Consider the following given sample data.

Object identifier	test-1 (categorical)	test-2 (ordinal)	test-3 (Ratio-scaled)
1	Code-A	excellent	445
2	Code-B	fair	22
3	Code-C	good	164
4	Code-A	excellent	1,210

The procedure followed for categorical and ordinal as same as the above mentioned methods for processing the variables of mixed types.

So that we can use dissimilarity matrix obtained for test-1 & test-2.

For Ratio-scaled variable, logarithmetic transformation of values are already applied.

ii

Object identifier	test-3	log value
1	445	2.65
2	22	1.34
3	164	2.21
4	1210	3.08

$\max_h x_h = 3.08$

$\min_h x_h = 1.34$

Normalize the values in the dissimilarity matrix obtained by dividing each one by $(3.08 - 1.34) = 1.74$.

dissimilarity matrix ← Obtained by Ratio scaled variable test.

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

Scanned with CamScanner

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

The dissimilarity matrix for the three variables in our computation. ~~$d(i,j) = \sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{f}$~~

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

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

The resulting dissimilarity matrix ~~dis~~ for the data described by 3 variables of mixed type is

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

Vector objects

We can define the similarity function $s(x,y)$ to compare two vectors x and y is several ways. One popular way is to define the similarity function as a cosine measure as follows

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

Scanned with CamScanner

where x^t is a transposition of vector x, $\|x\|$ is the Euclidean norm of vector x, $\|y\|$ is the Euclidean norm of vector y and 's' is the cosine of the angle between vector x & y.

Example :-

consider two vector $x = (1,1,0,0)$ and $y = (0,1,1,0)$

The similarity between x and y is $s(x,y) = \frac{(0+1+0+0)}{\sqrt{2}\sqrt{2}} = 0.5$

A simple variation of the above measure is

$$s(x,y) = \frac{x^t \cdot y}{x^t \cdot x + y^t y - x^t \cdot y}$$

which is the ratio of the number of attributes shared by x and y to the number of attributes possessed by x or y. This function is known as Tanimoto Coefficient is frequently used in information retrieval.

Partitioning Methods

Let D be the data set of 'n' objects, and 'k' the number of clusters to form a partitioning algorithm organizes the objects into k partitions ($k \leq n$) where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion such as a dissimilarity function based on distance, so that the objects within a cluster are similar whereas the objects

Scanned with CamScanner

of different clusters are dissimilar in terms of the data set attributes

Commonly used partitioning Methods are

→ K-means
→ K-Medoids

Centroid Based Technique: The k-means Method

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

cluster similarity is measured with 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 cluster mean 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.

The square-error criterion is used, which is defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

Scanned with CamScanner

where E is the sum of the square error for all objects in the data set, p is the point in space representing a given object, and m_i is the mean of cluster C_i. For each object in each cluster, the distance from the object to its cluster center is squared and the distances are summed. This criterion tries to make the resulting k clusters as compact and as separate as possible. The k-means procedure is summerized as follows.

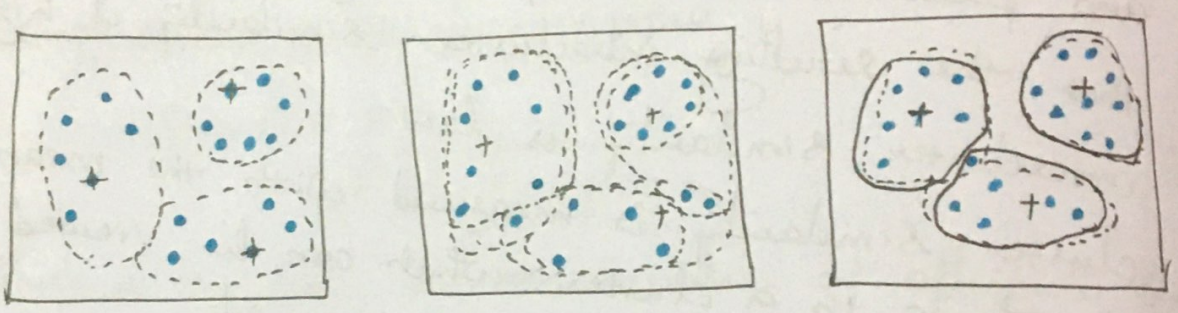


fig: k-means partitioning algorithm.

Algorithm k-means: The k-means algorithm for partitioning where each cluster's center is presented by the mean value of the objects in the cluster:

Input:
→ k: the number of clusters
→ D: a data set containing n objects.

Output: A set of k clusters.

Method:

(1) arbitrarily choose k objects from D as the initial cluster centers.

(2) Repeat

(3) Re assign each object to the cluster to which the object is the most similar based on the mean value of the objects in the manner.

Scanned with CamScanner

(4) update the clusters means, ie calculate the mean value of the objects for each cluster

(5) until no change;

Example:- Consider the following set of items with the value of $k = 2$. and the set $S = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$

$m_1 = 4$ & $m_2 = 12$

$K_1 = \{2, 3, 4\}$

mean of $K_1 = 3$

$K_1 = \{2, 3, 4, 10\}$

mean of $K_1 = 5$

$K_1 = \{2, 3, 4, 10, 11, 12\}$

mean of $K_1 = 7$

$K_1 = \{2, 3, 4, 10, 11, 12\}$

mean of $K_1 = 7$

$K_2 = \{10, 11, 20, 25, 30, 12\}$

mean of $K_2 = 18$.

$K_2 = \{11, 12, 20, 25, 30\}$

mean of $K_2 = 19.6 = 20$

$K_2 = \{20, 25, 30\}$

mean of $K_2 = 25$

$K_2 = \{20, 25, 30\}$

mean of $K_2 = 25$

If we got the same mean for 2 consecutive iteration steps then we should stop the iteration of k-means

K-Medoid Methods : Representative Object-Based Technique.

Limitation of k-means:

The k-means algorithm is sensitive to outliers because an object with an extremely large value may distort the distribution of data.

Scanned with CamScanner

In k-medoid method 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 objects 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 objects and its corresponding reference point. That is an absolute-error criterion is used defined as

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - O_j|$$

where E is the sum of the absolute error for all objects in the data set, p is the point in space representing a given object in cluster C_j and O_j is the representative object of C_j.

This algm iterates until each representative object is actually the medoid or most centrally located object of its cluster. This is the basis of the k-medoids method for grouping n objects into k-clusters.

In k-medoid method. The initial representative objects are chosen arbitrarily. The iterative process of replacing representative objects by nonrepresentative objects continues as long as the quality of the resulting clustering is

Scanned with CamScanner

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 nonrepresentative object, O_{random} is a good replacement for a current representative object O_j, the following 4 cases are examined for each of the nonrepresentative objects p

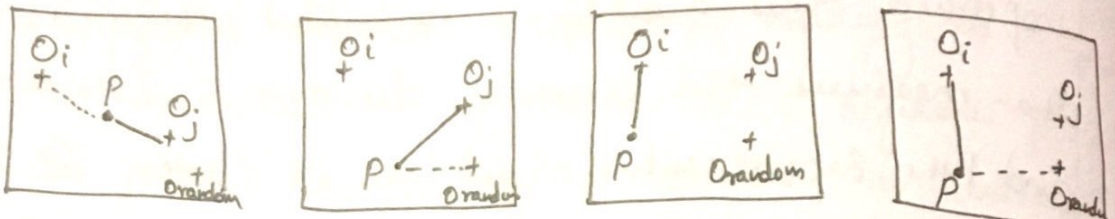
Case 1: p currently belongs to representative object, O_j. If O_j is replaced by O_{random} as a representative object and p is closest to one of the other representative objects O_i ($i \neq j$), then p is reassigned to O_i

Case 2: p currently belongs to representative object, O_j. If O_j is replaced by O_{random} as a representative object and p is closest to O_{random}, then p is reassigned to O_{random}

Case 3: p currently belongs to representative object O_i, ($i \neq j$) If O_j is replaced by O_{random} as a representative object & p is still closest to O_i, then assignment does not change.

Case 4: p currently belongs to representative object O_i, ($i \neq j$). If O_j is replaced by O_{random} as a representative object and p is closest to O_{random}, then p is reassigned to O_{random}.

Scanned with CamScanner



1. Reassigned to Oi 2: Reassigned to Orandom 3: No change 4: Reassigned to Orandom.

Each time a reassignment occurs, a difference in absolute error, E is contributed to the cost function. The cost function calculates the difference in absolute-error value if a current representative object is replaced by a nonrepresentative object. The total cost of swapping is the sum of costs incurred by all nonrepresentative objects. If the total cost is negative then Oj is replaced or swapped with Orandom since the actual absolute error E would be reduced. If the total cost is positive, the current representative object Oj is considered as acceptable.

PAM (partitioning Around Medoids) was one of the first k-medoid algorithm. It attempts to determine k-partitions for n objects. After the initial random selection of k representative objects, the algorithm repeatedly tries to make a better one object in each pair is considered a representative object and the other is not. The PAM algorithm is given below.

Algorithm: k-medoid

Input:

k: number of clusters

D: a data set containing n objects

Scanned with CamScanner

Output: A set of k clusters.

Method:

(1) arbitrarily choose k objects in D as the initial representative objects or seeds.

(2) Repeat

(3) assign each remaining object to the cluster with the nearest representative object.

(4) randomly select a nonrepresentative object, O_{random};

(5) Compute the total cost, S, of swapping representative object O_j with O_{random}

(6) if $S<0$ then swap O_j with O_{random} to form the new set of k representative objects.

(7) Until no change.

Example:-

Consider the following given dataset and group the data into two cluster.

	x	y
x_1	2	6
x_2	3	4
x_3	3	8
x_4	4	7
x_5	6	2
x_6	6	4
x_7	7	3
x_8	7	4
x_9	8	5
x_{10}	7	6

Consider randomly chosen data as $C_1 = (3,4)$ & $C_2 = (7,4)$

& find the difference of each data item with the randomly choosen data

Scanned with CamScanner

			C_1	C_2	C_1	C_2
x_1	2	6	$\|2-3\|+\|6-4\| = 3$	$\|2-7\|+\|6-4\| = 7$	(3)	7
(R) x_2	—	—				
x_3	3	8	$\|3-3\|+\|8-4\| = 4$	$\|3-7\|+\|8-4\| = 8$	(4)	8
x_4	4	7	$\|4-3\|+\|7-4\| = 4$	$\|4-7\|+\|7-4\| = 6$	(4)	6
x_5	6	2	$\|6-3\|+\|4-2\| = 5$	$\|6-7\|+\|2-4\| = 3$	5	(3)
x_6	6	4	$\|6-3\|+\|4-4\| = 3$	$\|6-7\|+\|4-4\| = 1$	3	(1)
x_7	7	3	$\|7-3\|+\|4-3\| = 5$	$\|7-7\|+\|3-4\| = 1$	5	(1)
(R) x_8	—	—				
x_9	8	5	$\|8-3\|+\|4-5\| = 6$	$\|8-7\|+\|5-4\| = 2$	6	(2)
x_{10}	7	6	$\|7-3\|+\|6-4\| = 6$	$\|7-7\|+\|6-4\| = 2$	6	(2)

Of the values of C_1 & C_2 take the minimum values of the two {(3,4) & (7,4)}

Total cost of two clusters = 3+4+4+3+1+1+2+2 = 20

Cluster 1 $C_1 = \{(2,6), (3,8), (4,7), (3,4)^{(R)}\}$

Cluster 2 $C_2 = \{(7,4)^{(R)}, (6,2), (6,4), (7,3), (8,5), (7,6)\}$

Change the random data item as (3,4) & (7,3)

x	y	C_1	C_2
2	6	(3)	$\|2-7\|+\|3-6\| = 8$
3	8	(4)	9
4	7	(4)	7
6	2	5	(2)
6	4	3	(2)
7	4	4	(1)
8	5	6	(3)
7	6	6	(3)

Scanned with CamScanner

$C_1 = (2,6), \overset{(R)}{(3,4)}, (3,8), (4,7)$

$C_2 = \overset{(R)}{(7,3)}, (6,2), (6,4), (7,4), (8,5), (7,6)$

Total cost = $3+4+4+2+2+1+3+3 = 22$.

previous cost = 20.

Current cost = 22.

Current cost - previous cost.

$22 - 20 = 2$ (positive cost).

So the iteration can stop and the cluster include the following data items are.

$C_1 = \{(3,4), (2,6), (3,8), (4,7)\}$

$C_2 = \{(7,3), (6,2), (6,4), (7,4), (8,5), (7,6)\}$

If we got negative cost difference then we have to choose random data item set again and find the cost.

Scanned with CamScanner

MOD - VI

Hierarchical clustering Methods.

A hierarchical clustering method works by grouping data objects into a tree of clusters. Hierarchical methods can be further classified into agglomerative or divisive depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion. The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed.

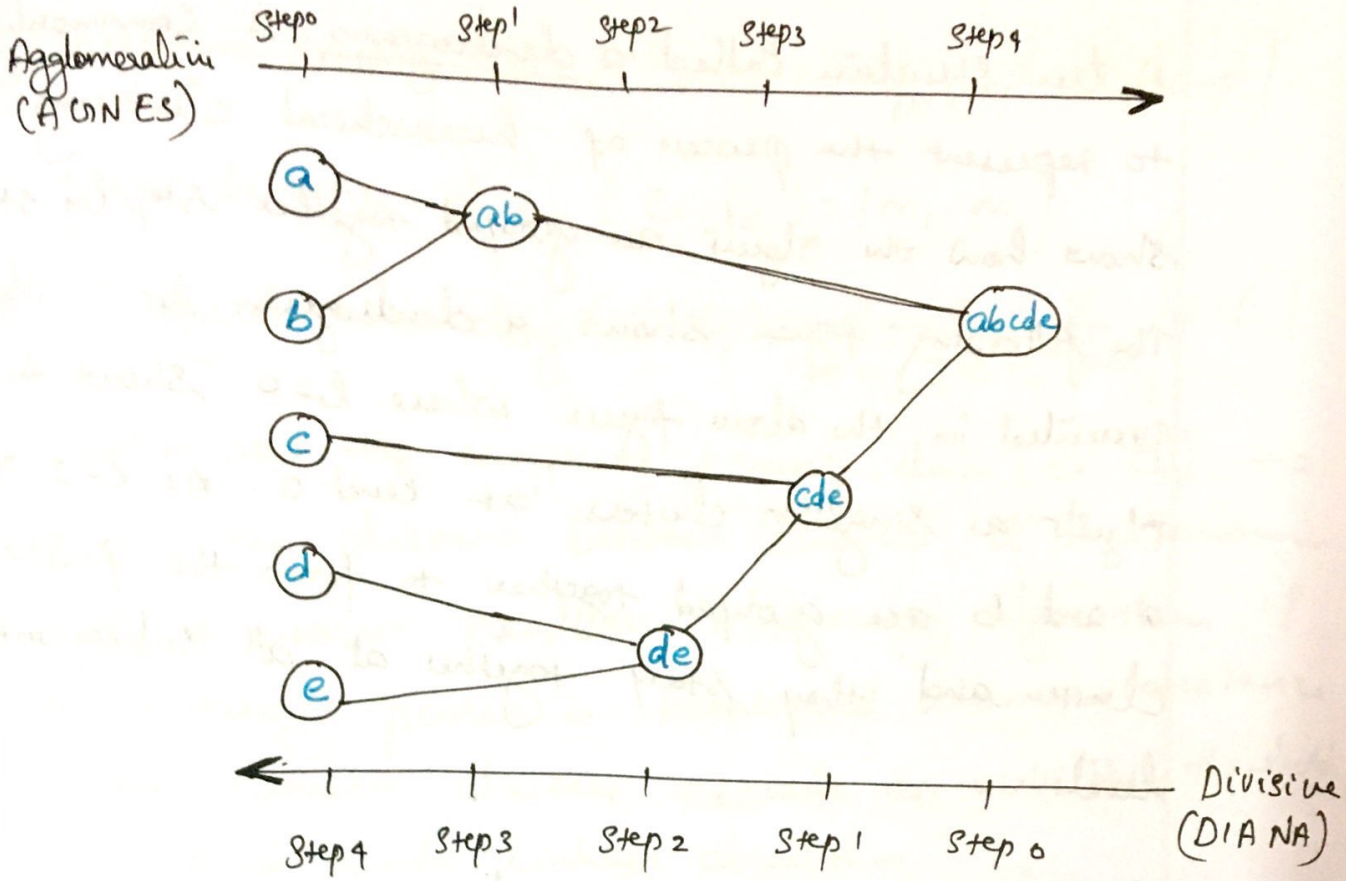
Agglomerative and Divisive Hierarchical clustering

There are two types of hierarchical clustering methods

① Agglomerative hierarchical clustering:
This bottom-up strategy starts by placing each object ~~into~~ 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.

② Divisive hierarchical clustering:
This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions

Scanned with CamScanner



The above figure shows the application of AGNES & DIANA to a data set of 5 objects {a, b, c, d, e}.

Initially AGNES places each object into a cluster of its own. The clusters are then merged step-by-step according to some criterion.

In DIANA, all of the objects are used to form one initial cluster. The cluster is split according to some principle such as maximum Euclidean distance between the closest neighbouring objects in the cluster. The cluster splitting process repeats until, eventually each new cluster contains only a single object.

In AGNES & DIANA the user can specify the desired number of clusters as a termination condition

Scanned with CamScanner

A tree structure called a dendrograms is commonly used to represent the process of hierarchical clustering. It shows how the objects are grouped together step by step. The following figure shows a dendrogram for 5 objects presented in the above figure where $l=0$ shows the 5 objects as singleton clusters at level 0. At $l=1$ objects a and b are grouped together to form the first cluster and they stay together at all subsequent levels.

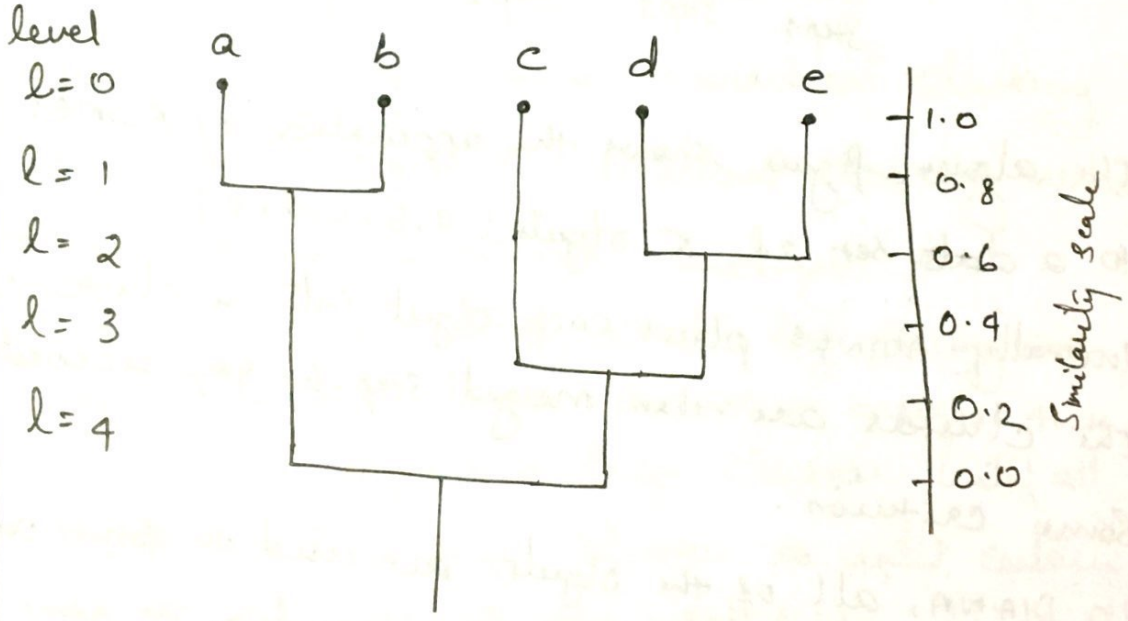


fig: Dendrogram representing objects $\{a, b, c, d, e\}$.

The distance between the clusters can be measures as follows. where $|p-p'|$ is the distance between two objects or points, p and p'; m_i is the mean for cluster, C_i and n_i is the number of objects in C_i

Minimum distance: $d_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p-p'|$

Scanned with CamScanner

Maximum distance : $d_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$

Mean distance : $d_{mean}(C_i, C_j) = |m_i - m_j|$

Average distance : $d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p \in C_j} |p - p'|$

Algorithm uses the minimum distance, $d_{min}(C_i, C_j)$, to measure the distance between clusters, it is sometimes called a nearest neighbour clustering algorithm. If the clustering process is terminated when the distance between nearest clusters exceeds an arbitrary threshold it is called single-linkage algorithm.

Agglomerative hierarchical clustering algm that uses the minimum distance measure is called minimal spanning tree algorithm.

When an algm use the maximum distance, $d_{max}(C_i, C_j)$ to measure the distance between clusters, it is sometimes called a farthest-neighbour clustering algorithm. If the clustering process is terminated when the maximum distance between nearest clusters exceeds an threshold it is called a complete linkage algm.

The use of mean or average distance is a compromise between the minimum and maximum distances and overcomes the outlier sensitivity problem.

Scanned with CamScanner

BIRCH: Balanced Iterative Reducing & Clustering Using Hierarchies

BIRCH is designed for clustering a large amount of numerical data by integration of hierarchical clustering and other clustering method such as iterative partitioning. It overcomes the two difficulties of agglomerative clustering methods:

1) Scalability
2) Inability to undo what was done in the previous step.

BIRCH introduces two concepts

→ Clustering feature
→ Clustering feature tree (CF tree)

The above 2 concept are used to summarize cluster representation. These structures help the clustering method achieve good speed and scalability in large db and also make it effective for incremental & dynamic clustering of incoming objects.

Given n d-dimensional data objects or points in a cluster the centroid x_0, radius R and diameter D of the cluster can be defined as

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_0)^2}{n}}$$

Scanned with CamScanner

$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2}{n(n-1)}}$$

where R is the average distance from member objects to the centroid and D is the average pairwise distance within a cluster. Both R and D reflect the tightness of the cluster around the centroid.

A clustering feature (CF) is a 3-dimensional vector summarizing information about clustering of objects.

Given n d-dimensional objects or points in a cluster $\{x_i\}$ then the CF of the cluster is defined as

$$CF = \langle n, LS, SS \rangle$$

where n is the number of points in the cluster, LS is the linear sum of n points (ie $\sum_{i=1}^{n} x_i$) and SS is the square sum of data points ie $\sum_{i=1}^{n} x_i^2$.

A clustering feature is essentially a summary of its statistics of the given cluster the zeroth, first and second moments of the cluster.

Clustering features are additive, ie let C_1 & C_2 be two disjoint clusters with clustering features CF_1 & CF_2. The clustering feature for the cluster formed by merging C_1 & C_2 is $CF_1 + CF_2$.

Clustering features are sufficient for calculating all the measurements that are needed for making clustering

Scanned with CamScanner

decisions in BIRCH.

Example:- Consider 3 points (2,5), (3,2) and (4,3) in a cluster C_1. The clustering feature of C_1 is

$$CF_1 = \langle 3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2) \rangle$$

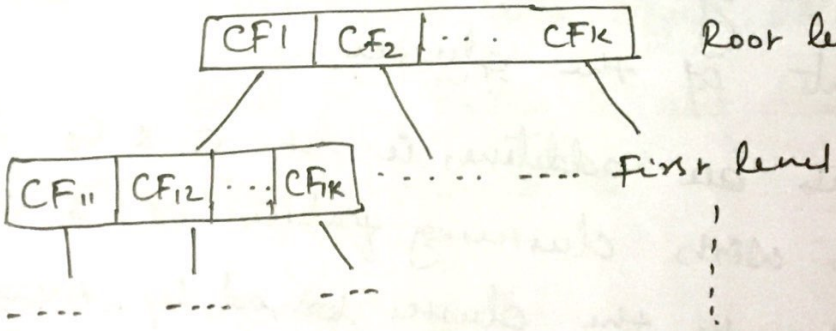
$$= \langle 3, (9,10), (417, 440) \rangle$$

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

$$CF_3 = \langle 3+3, (9+35, 10+36), (29+417, 38+440) \rangle$$

$$= \langle 6, (44, 46), (446, 478) \rangle$$

A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering. An example is shown in the following figure.



A nonleaf node stores sum of the CFs of their children and thus summarize clustering information about their children.

Scanned with CamScanner

A CF tree has two parameters

i) Branching factor
ii) Threshold, T

Branching factor specifies the maximum number of children per nonleaf node.

The threshold parameter specifies the maximum diameter of subclusters stored at the leaf nodes of the tree. These two parameters influence the size of the resulting tree.

BIRCH applies a multiphase clustering technique: a single scan of the data set yields a basic good clustering and one or more additional scans can be used to further improve the quality. The primary phases are

phase 1:

BIRCH scans the db to build an initial in memory CF-tree, which can be viewed as a multilevel compression of the data that tries to preserve the inherent clustering structure of the data

phase 2:

BIRCH applies a clustering algorithm to cluster the leaf nodes of the CF tree, which removes sparse clusters as outliers and group dense clusters into larger ones.

Scanned with CamScanner

For phase 1, CF tree is built dynamically as objects are inserted. Thus the method is incremental.

An object is inserted into the closest leaf entry. If the diameter of the subcluster stored in the leaf node after insertion is larger than the threshold value then the leaf node and possibly other nodes are split. This is similar to insertion and node split in B^+ trees.

After the insertion of the new object, the information about it is passed toward the root of the tree.

Once the CF tree is built, any clustering algorithm, such as partitioning algorithm can be used with the CF tree in phase 2.

Complexity of algorithm is $O(n)$ where 'n' is the no: of object to be clustered.

Advantages

- Scalability
- Good quality of clusters

Disadvantages

- CF tree can hold a limited no: of entries
- BIRCH do not perform well when clusters are not spherical in shape.

Scanned with CamScanner

Density Based Methods.

Density Based methods are developed to discover clusters with arbitrary shape, these typically regard clusters as dense regions of objects in the data space that are seperated by regions of low density.

Two Types of Density Based Methods are

→ DBSCAN
→ OPTICS

DBSCAN (Density Based Spatial clustering of Applications With Noise)

DBSCAN is a density Based clustering method based on connected region with sufficiently high density. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shapes in spatial db with noise. It defines a cluster as a maximal set of connected points.

Definitions used in DBSCAN:-

→ The neighborhood within a radius ε of a given object is called ε-neighbourhood of the object.

→ If the ε-neighbourhood of an object contains atleast a minimum number, Minpts of objects, then the object is called the core object.

→ Given a set of objects D, we say that an object p

is directly density reachable from object q if p is within the ε-neighborhood of q & q is a core object.

→ An object p is density reachable from object q with respect to ε & Minpts in a set of objects D, if there is a chain of objects $p_1, \ldots, p_n$, where $p_1 = q$ & $p_n = p$ such that p_{i+1} is directly density reachable from p_i with respect to ε and Minpts for $1 \leq i \leq n$, $p_i \in D$.

→ An object p is density connected to object q with respect to ε and Minpts in a set of objects D, if there is an object $o \in D$ such that both p & q are density-reachable from o with respect to ε and Minpts.

Density Reachability and density connectivity :-

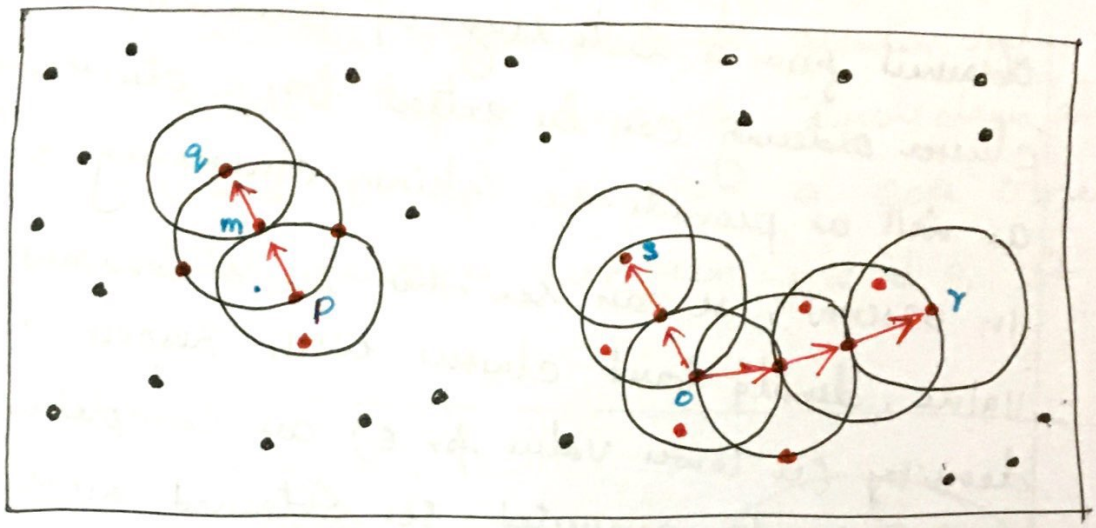
Consider the following figure for a given ε represented by the radius of the circle and let Minpts = 3. Based on the above definitions

- of the labeled points m, p, o and r are core objects because each is in an ε-neighborhood containing atleast 3 points
- q is directly density-reachable from m. m is directly density reachable from p and vice versa.
- q is (indirectly) density reachable from p because q is directly reachable from m and m is directly (density) reachable from p

Scanned with CamScanner

However p is not density reachable from q, because q is not a core object. Similarly r and s are density reachable from o and o is density reachable from r.

⊛ o, r and s are all density connected



A density based cluster is a set of density connected object that is maximal with respect to density reachability. Every object not contained in any cluster is considered to be noise.

How does DBSCAN find clusters?

DBSCAN searches for clusters by checking the ε-neighborhood of each point in the database. If the ε-neighborhood of a point p contains more than MinPts, a new cluster with p as a core object is created. DBSCAN then iteratively collects directly density reachable objects from these core objects which may involve the merge of a few density reachable clusters. The process terminates when no new points can be added to any cluster

Scanned with CamScanner

OPTICS: Ordering points to Identify the clustering Structure

OPTICS cluster analysis method computes an augmented cluster ordering for automatic and interactive cluster analysis. This ordering represents the density based clustering obtained from a wide range of parameter settings. The cluster ordering can be extract basic clustering information as well as provide the intrinsic clustering structure.

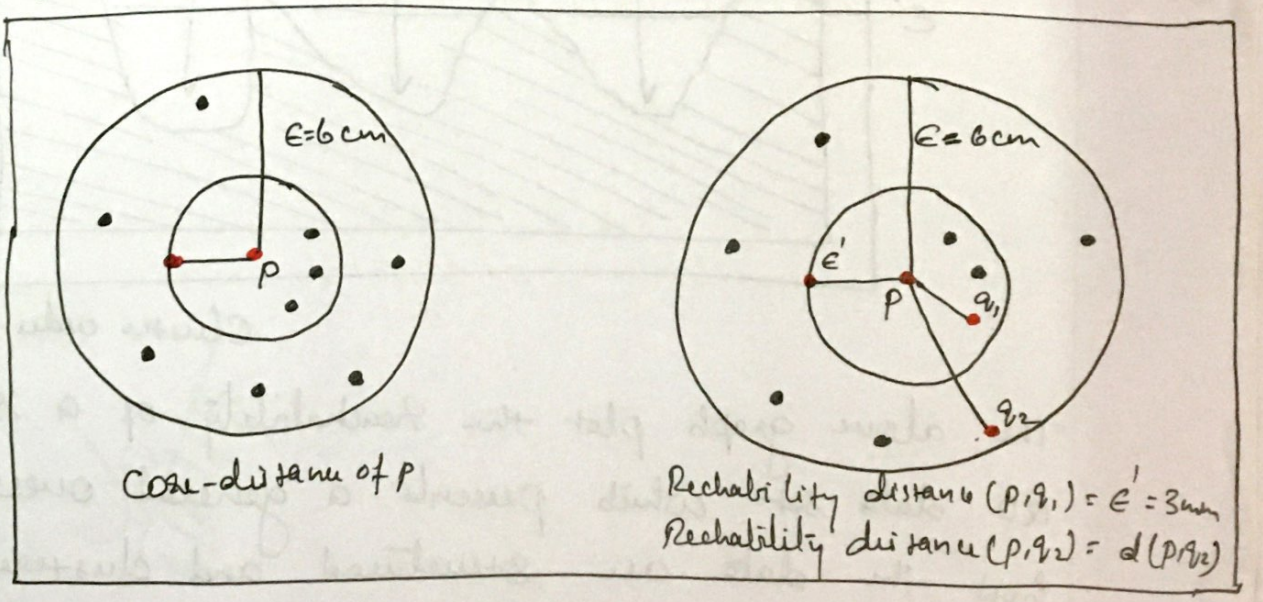
In DBSCAN, we can see that for a constant MinPts value, density based clusters with respect to a higher density (ie lower value for ϵ) are completely contained in a density connected set obtained with respect to lower density. The parameter ϵ is a distance ie neighbourhood radius. Therefore inorder to produce a set or ordering of density based clusters, we can extend the DBSCAN algorithm to process a set of distance parameter values at the same time. To construct the different clustering simultaneously, the objects should be processed in specific order. This order selects an object that is density reachable with respect to the lowest ϵ value so that clusters with higher density will be finished first.

Based on this idea, two values need to be stored for each object - Core distance & reachability distance.

Scanned with CamScanner

* The Core distance of an object p is the smallest ϵ' value that makes {p} a core object. If p is not a core object, the core distance of p is undefined

* The reachability distance of an object q with respect to another object p is the greater value of the core distance of p and the Euclidean distance between p and q. If p is not a core object, the reachability distance between p and q is undefined.



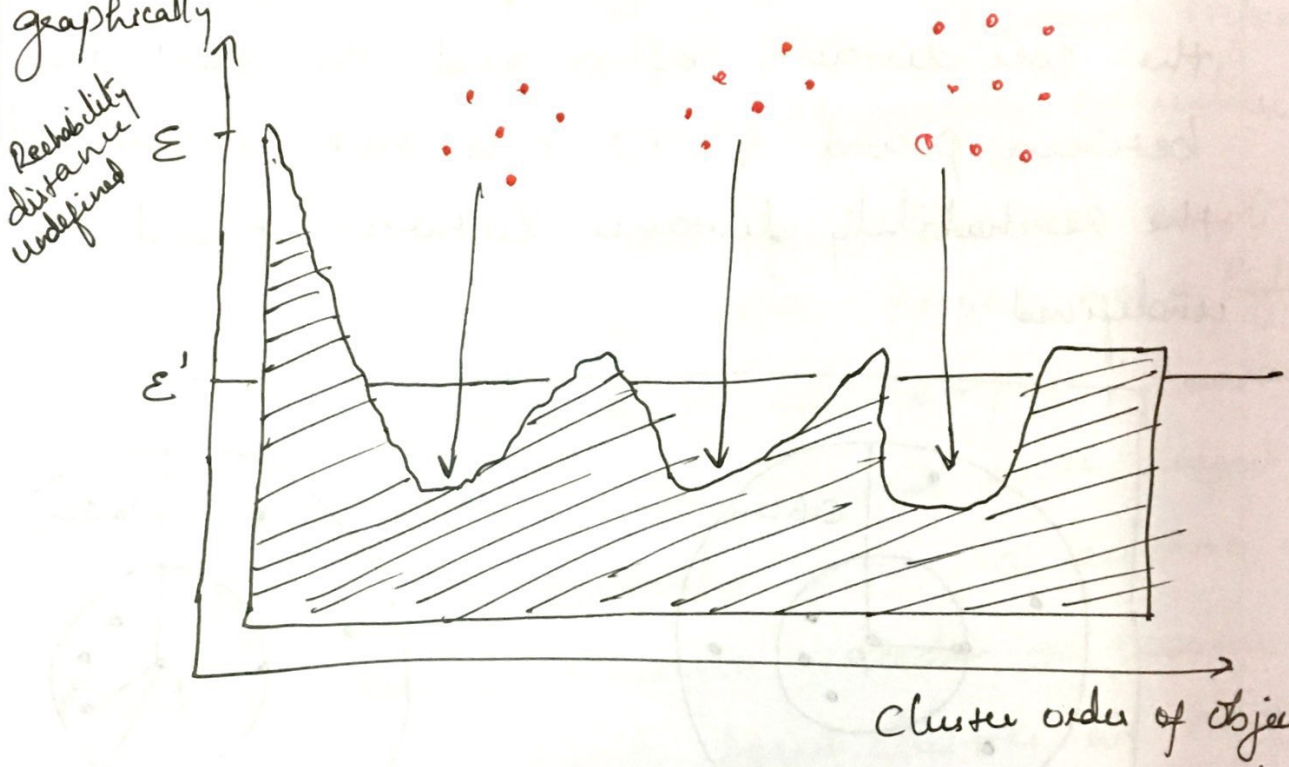
How these values are used?

The OPTICS algm creates an ordering of the objects in a database, additionally storing the core distance and a suitable reachability distance and ~~a suitable~~ for each object. An algorithm was proposed to extract clusters based on the ordering information produced by OPTICS Such information is sufficient for the extraction of all

Scanned with CamScanner

density based clustering with respect to any distance ε' that is smaller than the distance ε used in generating the order.

The cluster ordering of a dataset can be represented graphically



The above graph plot the reachability of a sample 2D data set which presents a general overview of how the data are structured and clustered. The data objects are plotted in cluster order together with their respective reachability distance. The three Gaussian bumps reflects three clusters in data set.

Complexity of OPTICS algm - $O(n \log n)$

Scanned with CamScanner

Data Mining tools

Data Mining is the set of techniques that utilize specific algorithms, statical analysis, artificial intelligence, and database systems to analyze data from different dimensions and perspectives.



Data Mining tools have the objective of discovering patterns/trends/groupings among large sets of data and transforming data into more refined information.

It is a framework, such as Rstudio or Tableau that allows you to perform different types of data mining analysis.

We can perform various algorithms such as clustering or classification on your data set and visualize the results itself. It is a framework that provides us better insights for our data and the phenomenon that data represent. Such a framework is called a data mining tool.

The Market for Data Mining tool is shining: as per the latest report from ReortLinker noted that the market would top **$1 billion** in sales by **2023**, up from **$ 591** million in **2018**

These are the most popular data mining tools:



Orange Data Mining:

Orange is a perfect machine learning and data mining software suite. It supports the visualization and is a software-based on components written in Python computing language and developed at the bioinformatics laboratory at the faculty of computer and information science, Ljubljana University, Slovenia.

As it is a software-based on components, the components of Orange are called "widgets." These widgets range from preprocessing and data visualization to the assessment of algorithms and predictive modeling.

Widgets deliver significant functionalities such as:

- Displaying data table and allowing to select features
- Data reading
- Training predictors and comparison of learning algorithms
- Data element visualization, etc.

Besides, Orange provides a more interactive and enjoyable atmosphere to dull analytical tools. It is quite exciting to operate.

Why Orange?

Data comes to orange is formatted quickly to the desired pattern, and moving the widgets can be easily transferred where needed. Orange is quite interesting to users. Orange allows its users to make smarter decisions in a short time by rapidly comparing and analyzing the data.It is a good open-source data visualization as well as evaluation that concerns beginners and professionals. Data mining can be performed via visual programming or Python scripting. Many analyses are feasible through its visual programming interface(drag and drop connected with widgets)and many visual tools tend to be supported such as bar charts, scatterplots, trees, dendrograms, and heat maps. A substantial amount of widgets(more than 100) tend to be supported.

The instrument has machine learning components, add-ons for bioinformatics and text mining, and it is packed with features for data analytics. This is also used as a python library.

Python scripts can keep running in a terminal window, an integrated environment like PyCharmand PythonWin, pr shells like iPython. Orange comprises of canvas interface onto which the user places widgets and creates a data analysis workflow. The widget proposes fundamental operations, For example, reading the data, showing a data table, selecting features, training predictors, comparing learning algorithms, visualizing data elements, etc. Orange operates on Windows, Mac OS X, and a variety of Linux operating systems. Orange comes with multiple regression and classification algorithms.

Orange can read documents in native and other data formats. Orange is dedicated to machine learning techniques for classification or supervised data mining. There are two types of objects used in classification: learner and classifiers. Learners consider class-leveled data and return a classifier. Regression methods are very similar to classification in Orange, and both are designed for supervised data mining and require class-level data. The learning of ensembles combines the predictions of individual models for precision gain. The model can either come from different training data or use different learners on the same sets of data.

Learners can also be diversified by altering their parameter sets. In orange, ensembles are simply wrappers around learners. They act like any other learner. Based on the data, they return models that can predict the results of any data instance.

SAS Data Mining:

SAS stands for Statistical Analysis System. It is a product of the SAS Institute created for analytics and data management. SAS can mine data, change it, manage information from various sources, and analyze statistics. It offers a graphical UI for non-technical users.

SAS data miner allows users to analyze big data and provide accurate insight for timely decision-making purposes. SAS has distributed memory processing architecture that is highly scalable. It is suitable for data mining, optimization, and text mining purposes.

DataMelt Data Mining:

DataMelt is a computation and visualization environment which offers an interactive structure for data analysis and visualization. It is primarily designed for students, engineers, and scientists. It is also known as DMelt.

DMelt is a multi-platform utility written in JAVA. It can run on any operating system which is compatible with JVM (Java Virtual Machine). It consists of Science and mathematics libraries.

- **Scientific libraries:**
  Scientific libraries are used for drawing the 2D/3D plots.
- **Mathematical libraries:**
  Mathematical libraries are used for random number generation, algorithms, curve fitting, etc.

DMelt can be used for the analysis of the large volume of data, data mining, and statistical analysis. It is extensively used in natural sciences, financial markets, and engineering.

Rattle:

Ratte is a data mining tool based on GUI. It uses the R stats programming language. Rattle exposes the statical power of R by offering significant data mining features. While rattle has a comprehensive and well-developed user interface, It has an integrated log code tab that produces duplicate code for any GUI operation.

The data set produced by Rattle can be viewed and edited. Rattle gives the other facility to review the code, use it for many purposes, and extend the code without any restriction.

Rapid Miner:

Rapid Miner is one of the most popular predictive analysis systems created by the company with the same name as the Rapid Miner. It is written in JAVA programming language. It offers an integrated environment for text mining, deep learning, machine learning, and predictive analysis.

The instrument can be used for a wide range of applications, including company applications, commercial applications, research, education, training, application development, machine learning.

Rapid Miner provides the server on-site as well as in public or private cloud infrastructure. It has a client/server model as its base. A rapid miner comes with template-based frameworks that enable fast delivery with few errors(which are commonly expected in the manual coding writing process)