Data Mining Notes

Contributor: Deepanshu

[KMV CS(H)]

Computer Science Notes

CHAPTER-1

1. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson

Data are raw facts which are not processed and mining when we try to get useful info. So in data mining we process our data and try to get maximum useful information from it. Data Mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data. It's the process of discovering useful information all useful patterns in a large data repositories that might otherwise remain unknown.

* APPLICATIONS :-

1) Customer Profiling :- Customer service records are collected from Call Centres, Web logs from e-Commerce Websites etc.
2) Targeted Marketing :- To find a suitable group of customers for their product and does marketing selectively.
3) Store layout
4) Fraud Detection :- To track an unusual transaction in an account.
5) Medicines :- It's used to address biological challanges such as protein structure prediction, modelling of biomedical pathways etc.

* INFORMATION RETRIEVAL

→ is looking up for individual records using a DBMS or finding particular webpages using a query.

Data mining techniques are used to enhance information retrieval systems.

* Types of Data :- (according to structure)

1) Structured :- Data that has a defined datatype or format. e.g - Word or excel files.
2) Semi-structured :- Textual data with a noticeable pattern. eg - Xml files.
3) Quasi-structured :- Textual data which requires effort, tools and time for formatting.
   e.g - Web server logs.
4) Unstructured :- This data has no structure and is stored as different types of file.
   e.g - pdf.

* The mined information must be :-

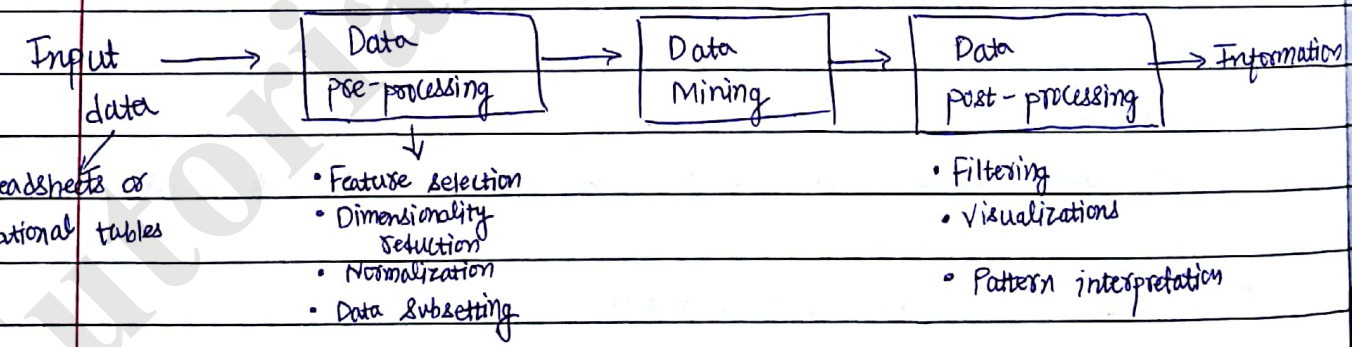
1) New :- The extracted information should give us new patterns or relationships among data entity. e.g - if we have a dataset regarding the people who smoke and if we get the result smoking causes cancer it's not something new. As it's not new so there is no purpose investing so much time, money and efforts

2) Useful :- The mined info from the data should be useful and relevant to us. for e.g - in the smoking dataset if we concludes about the longetivity of the person who smokes and who doesn't, it would prove to be more useful.

3) Correct :- The mined needs to be evaluated for it's correctness before we use it for any other purpose. for e.g - in the dataset of stock prices of all companies we predict company X → ↑ 20% but after sometime it actually got decreased by 10% so it's always neccessary to validate information before reaching to any conclusion.

* Data Mining and Knowledge Discovery

→ Knowledge discovery is the process of converting raw data into useful information which consists of transformation steps from data preprocessing to post processing of data mining results.



* Data Preprocessing

Feature selection :- is also known as attribute or variable selection is the process of selecting a subset of relevant features for use in model construction.

Dimensionality reduction :- includes feature selection and feature extraction.

Normalization :- to reduce data redundancy and improve data integrity.

Data Subsetting :- process of retrieving parts of large files.

Page No. Date

- Post Processing :-

Filtering patterns :- It's a design pattern that enables the developers to filter a set of objects using different criteria.

Visualizations -

Pattern interpretation :- evaluation and interpretation of pattern to decide what qualifies as knowledge.

Post processing ensures that only valid and useful results are incorporated into decision support systems.

Data Warehousing - Data from multiple sources and in multiple formats is stored together at one single location is called data warehousing.

Data Mining - Intelligent operations such as clustering, classification, association, regression are applied to extract patterns. It's the most important step in knowledge discovery.

R

is an opensource statistical environment modelled by Robert Gentleman and Ross A. Haker of statistics department of University of auckland in 1995. It's currently maintained by the R core development team consisting of international team of volunteer developers.

To see what packages are currently available ⟶ installed.packages().

You can install many more packages using :-

step1 - using the command install.packages ("")

step2 - loading using library (<packagename>)

once you have some extra packages installed they're not automatically ready for use and we must load them to make library of core routines available for use. The command to see which packages are currently loaded is search().

→ If u want to remove or unload a packages use command detach(package : packagename)

* R used as calculator

* Storing results of calculation

```
ans1 <- 3+9-2+3
ans1
[1] 13
```

Using the combined Command for making data

```
data1 = c(1,2,3,4).

data1
[1] 1 2 3 4
```

Entering text data :-

```
data2 = c("a", "b", "c", "d")

data2
[1] "a" "b" "c" "d"
```

```
data2 = c(1, 2, 3, "a")

data2
[1] "1" "2" "3" "a"
```

* Using Scan Command for making the data

When using the c Command typing all those commas to separate the values can be a bit tedious so we can use another Command scan() to do a similar job.

```
data5 = scan()
1: 2 3 4 5 ↵
5: 6 7 8 ↵
8: ↵
Read 7 items
> data5
[1] 2 3 4 5 6 7 8
```

```
> day2 = scan(what = 'character')
1: "a" "b" ↵
3: "c" ↵
4: ↵
Read 3 items
> day2
[1] "a" "b" "c"
```

```
data6 = scan(sep=',')
1: 1, 2, 3, 4 ↵
5: ↵
Read 4 items
```

Page No. Date

* Challenges faces by Traditional data analysis technique that motivated the development of data mining :-

1) Scalability :- As datasets with large sizes are becoming common, data mining algorithms are used which employs special strategies to handle exponential search problems also out of the H algorithms become necessary when processing a dataset that can't fit into main memory. e.g - climate data, hospital data. etc.

2) High dimensionality :- Now-a-days datasets have 100's and 1000's of attributes instead of a few handful attributes. for e.g - in bioinformatics the gene expression data involves 1000's of features, climate data contains measurements of temperature at various locations for an extended period of time so traditional data analysis technique don't work well for such high dimensional data.

3) Heterogeneous and Complex data :- Traditional methods deal with the datasets containing attributes of same types but recent years have seen emergence of more complex data objects. for e.g - DNA data with sequential and 3D structure, climate data which consists of time series measurements.

4) Data ownership and distribution :- Sometimes the data needed for analysis is geographically distributed among the resources belonging to multiple entities which requires the development of distributed data mining techniques.

* What is Supervised and Unsupervised learning ?

Unsupervised :- There are no predefined labels we just have the observations and the system clusters them based on their similarity this type of learning is called unsupervised. for e.g - there are mixed observations consisting of different animals and birds. so based on their similarities the system would be able to cluster them into 2 and only after clustering we get to know that one of the clusters is of the animals and second is of birds.

Supervised :- In this we have predefined labels and we have to classify our record in one of the predefined labels.

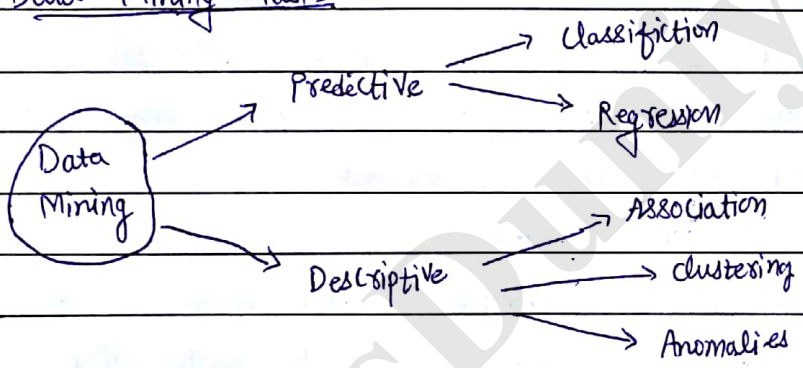
* Origins of Data Mining :-

Database Sys, Parallel Comp, Distributed Computing

Data mining is made up of :-

1) Sampling, estimation and hypothesis testing from statistics.
2) Search algorithms, modelling techniques and learning theories from AI.
3) Database system, parallel computing and distributed computing are needed to provide support for efficient storage, indexing and query processing.

* Data Mining Tasks



R Programming

UCI repository

Data folder → iris.csv

setwd("C:/Users/Desktop/Data") → set working directory

getwd() → get working director

read.csv("iris.csv")

read.csv("iris.csv", sep=" ", header= TRUE / FALSE)

access n-1 rows ↘ access all n rows

data1 = scan(file="abc.txt")

data1

dir("c/---")

data2 = scan(file.choose()) text file can be read

data2 = scan(file.choose(filename), sep=',', what='character' or 'char')

data3 = read.table(file.choose(), sep=',', header=FALSE)

→ comma would be removed

in this the header is FALSE by default.

for tab sep = '/t'

class1	class2
12	8
15	9
17	7
11	9
15	NA

Preprocessing

data	class
12	class1
15	class1
17	class1
11	class1
15	class1
8	class2
9	class2
7	class 2
9	class 2

* Data Mining Tasks Continued

* Predictive Tasks

To predict the value of a particular attribute based on the value of other attributes. the attribute to be predicted is known as targeted or dependent variable while the attribute used for making the prediction are known as explanatory or independent variables.

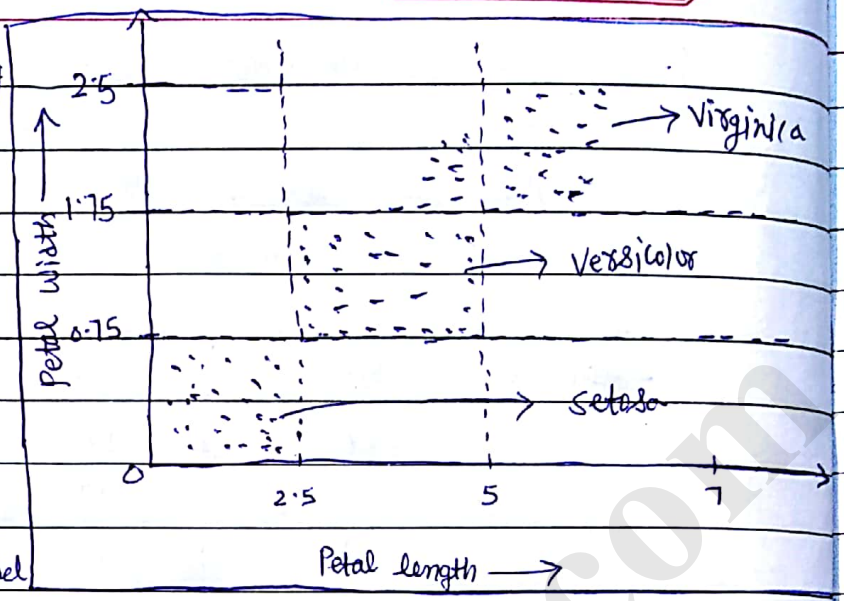
Predictive modelling refers to the task of building a model for target variable as a function of independent variables.

1) Classification:- It's used for discrete target variables. eg- iris data set is having 150 data items and 4 attributes sepal length, sepal width and petal length, petal width. The 3 species are setosa, versicolor and virginica. Petal width has values as low, medium, high same as petal length. Based on petal length and width following rules can be derived:-

Petal length	Petal width	
Low [0, 2.5]	Low [0, 0.75)	→ Setosa
Med [2.5, 5)	Med [0.75, 1.75)	→ Versicolor
High [5, ∞)	High [1.75, ∞)	→ Virginica

These rules help in classifying most of the flowers. Setosa species are well separated from versicolor and virginica but the latter two somewhat overlap with each other w.r.t. these attributes.

The goal is to create a model that minimizes the error between true and predicted values of target variable.



2) Regression:- is used for continuous target variable. for e.g - forecasting the future price of a stock.

* DESCRIPTIVE TASKS

Here the objective is to derive patterns that summarize the underlined relationships in data.

1) Association Analysis:- It's used to discover patterns that describe strongly associated features in an efficient manner. The discovered patterns are represented in the form of implication rules and feature subsets.

↳ Application :-

① Identifying webpages that are accessed together e.g- for buying an electronics item the lowest price of the product is displayed at all possible e-commerce sites, when we search for a research article on a particular topic all the related articles are shown one after other in search results.

② Finding groups of genes that have related functionality. It's used by scientists to improve the features of mutated plants.

③ Market basket analysis:- grocery stores apply association analysis to find the items that are frequently bought together by customers.

e.g - {Milk, butter} → {bread} whenever milk and butter is bought bread is also bought.

This type of rule can be used to identify cross selling opportunities among

related items.

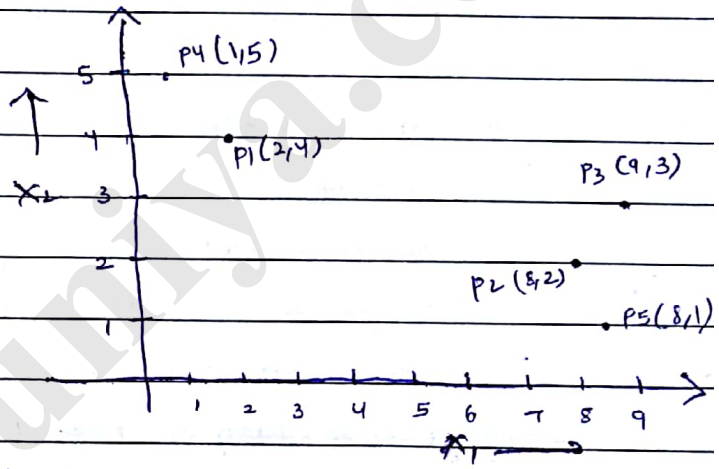
2) el

2) Cluster Analysis:- To find the groups of closely related information, observations that belong to the same cluster are more similar to each other than the observations that belong to other clusters.

e.g- it's used in compression of data, to find groups of related customers, collection of news articles can be grouped on the basis of repetitive topics.

Daily expenditure on food (x_1) and clothing (x_2) of 5 persons are shown:-

x_1	x_2	Person
2	4	P1
8	2	P2
9	3	P3
1	5	P4
8.5	1	P5



P1 and P4 in one cluster P2, P3 and P5 in another.

3) Anomaly detection:- To identify the observations called anomalies or outliers whose characteristics are significantly different from the rest of the data. Good anomaly detection algorithms must have a high detection rate and low false alarm rate (to call an observation an anomaly just beloz it's little different from rest of the observations).

Applications:-

a) detection of fraud:- a credit company builds a profile of every user by recording the legitimate transactions made by them along with some information like credit limit, name, income, transaction history. When a new transaction arrives it's compared against the profile of the user if the characteristics of the transaction are very different from the payment history (transactions) the transaction is flagged off as potentially fraudlent.

b) Network intrusion:- identifying strange patterns in network traffic that could signal hack.

c) system health monitoring

Prior to data processing we must ensure that either the outliers are removed or some adjustments are made before analysis of data.

R programming

ls() → lists all the name, Variable or items that are available.

* **Searching the patterns**

ls(pattern = 'b') → it will display all Variable names containing b

ls(pattern = '^b') → " " " " " starting with b

ls(pattern = 'b$') → " " " " " ending with b.

ls(pattern = '^[ba]') → " " " " " names starting either with b or a.

ls(pattern = 'b.c') → ------ of form bac, bbcl etc. with 1 letter between b and c.

* **Vector** as told

* **DataFrames**

dim(Variablename) lists no. of columns and rows in the data file.

colnames(Variable) = c(---) or names(data1) = c("sepall", "sepalw", "petall", "petal w") → names the columns

colnames(data1) or names(data1) to display name of column

* **Useful Summary of Dataset**

summary(data1), str(data1), glimpse(data1)
↳ for using it install dplyr package.

head(data1) → first 6 rows

tail(data1) → last 6 rows

head(data1, 15) → first 15 rows

> hist(data1$"petal-len")

> plot(data1$"petal-len", data1$"petal-width")

To select first two attributes of data1

data2 = data1[c(1,2)]

and to select only 50 rows

data2 = head(data2, 50)

CHAPTER - 2

DATA

Page No. Date

* Types of Data :- (Qualitative / Quantitative)

There are different types of data sets for e.g - the attributes used to describe the data objects can be either qualitative or quantitative

qualitative: colors, grades, skills, moods etc.

quantitative: age, marks, price, salary etc.

Types of data also determines the tools and techniques that can be used to analyze the data.

* Quality of Data

Data quality issues :-

1) Presence of noise or outliers 2) missing, inconsistent or duplicate data 3) data that's biased.

while most DM techniques can tolerate some level of imperfection, in data improving data quality improves the quality of resulting analysis.

* Processing steps to make the data more suitable for DM

Raw data must be processed :-

1) to improve data quality. 2) so that it fits better in a specified DM technique or tool.

e.g - length attribute may be transformed into an attribute with discrete values. e.g - short, medium or long

no. of attributes in a dataset is reduced so that techniques applied can be more effective.

* Analyzing the data in terms of it's relationships

→ Data analysis can be done by finding the relationship among the data objects and then performing the remaining analysis using these relationships. for e.g - We can compute the similarity or distance b/w pair of objects and then perform analysis (clustering, classification or anomaly detection) based on these similarities or distance

* TYPES OF DATA

A dataset is a collection of data objects (records, event, case, sample, observation, entity)

Attributes are basic characteristics of an object (also known as field, feature, dimension)

for e.g - in dataset of students each row corresponds to a student and each column describes some aspects of a student such as student id, name, address, Cgpa score etc.

Generally symbolically attributes have small number of possible values while numeric attributes have potentially unlimited no. of values. e.g - employee data set has attribute employeeage and employeeid. The age attribute is very much related to employee while the id attribute is only used to ensure that they are distinct employees so knowing the type of attribute is important in understanding whether the values of the attributes are consistent while underlined properties of the attributes.

Lab

iris_1 = data1

iris_1 $ class <- NULL (removes column class from iris_1.)

→ fix(iris_1) shows the changes made in iris_1

Check for missing values:-

Complete.cases(iris_1) shows with row is complete as TRUE or which is not as FALSE.

length(which(Complete.cases(iris_1)) shows no. of complete rows without NA

which(Complete.Cases(iris_1)) row no. of complete rows.

which(!Complete.cases(iris_1)) row no. of incomplete rows [NA rows].

Store rows with no NA:-

com <- which(Complete.cases(iris_1))

data3 <- iris_1[com,]

or

data4 <- na.omit(iris_1)

> data4

data5 = cut(data1$sepal_len, 3) divides into 3 levels

q = quantile(data1$petal_len, (0:3)/3) divides into 3 quantiles

* Converting one variable type to another

vec <- c(1,2,3,4,5,6)

class(vec) → "numeric" [tells data type of variable]

as.character(vec) → converts vec into character

Character > Numeric > Logical

Page No. Date

vec1 <- c("f", 2)
class(vec1) "character"

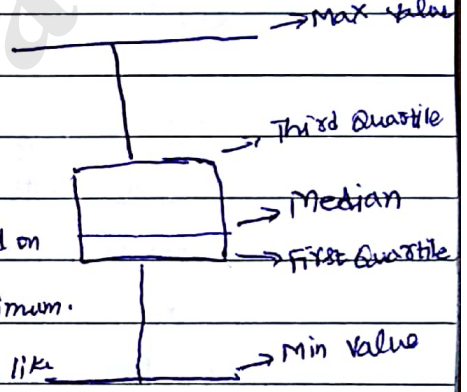
vec2 <- c(TRUE, 2)
vec2 [1] 1 2
class(vec2) "numeric"

vec2 <- c(TRUE, "f")
vec2 - "TRUE" "f"
class(vec2) "character"

Function to remove duplicate rows from data.
distinct(iris-1) // removes duplicate rows

Read about → dplyr, ggplot2, tidyr, tm, stringr.

OUTLIER DETECTION (boxplots)
boxplot(data1) → used to detect outliers and removing them
↳ part of preprocessing data



boxplot(data1, horizontal=T) → for horizontal boxplot.

A boxplot is a graphical representation of statistical data based on minimum, first quartile, median, third quartile and maximum. The term boxplot comes from the fact that the graph looks like a rectangle with lines extending from the top and bottom. because of the extending lines this type of graph is sometimes called box-and-whisker plot. The relative vertical spacing between the labels reflect the values of the variable in proportion. A boxplot can be placed on a coordinate plane resembling the cartesian system so that the 5 values arranged vertically one above the other run parallel to y axis.

Program:

```
r1 = which(!(0<=d$Age & d$Age<=150))
r1
length(r1)
```

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

Properties of numeric attributes

1. Distinctness
2. order
3. Addition / Subtraction
4. Multiplication / Division

on the basis of these properties we have 4 types of attributes

1. Nominal
2. ordinal
3. Interval
4. Ratio

(1) → The value of nominal attribute provide enough information to distinguish one object from another. E.g:- EmpId, Pincodes. It requires one to one mapping of values in case of transformation. If all employee id are rearranged it will not make any difference. e.g. Emp-id (1-100) → (1001-1100).

The observation that it supp

(i) Mode, entropy, x^2 test, etc.

↳ Attribute with more information gain increases predictibility which reduces entropy x^2 tests (cut-square-test) → is use to access the goodness of fit between a set of observed values and those expected theoretically.

(2) Ordinal attribute:- We can order the objects on the basis of information provided by the attributes. for e.g- length attribute will have the values small, medium and large, grade attributes, hardness of minerals good, better, best. While transforming their must be order preserving change of values.

new_value = f (old-value)

operations that it support → median, percentile, rank, Co-relation etc.

(3) Interval type of attributes

There exists a unit of measurement, the differences between the values are meaning ful.

$$T(°F) = \frac{9}{5} T°(C) + 32$$

- For transforming a new value the following conversion formula would be used
  new value = a * old-value + b
  where a, b are constants.
  °C and K are two different units of measurement which differ in their zero value and unit size.
  operations that these attributes support :- mean, standard deviation etc.

(4) Ratios

this type of attributes deal with numeric data. Here the transformation function used will be
new_value = a * old_value ,
where a is constant.
e.g - count, name, length, salary etc.
Here the attributes don't differ in these zero value. for eg - the statistical operation when performed on length attribute will give the same result whether length is in meter or foot.
operations :- Geometric mean, percent variation etc.

→ Nominal and ordinal attributes are collectively referred to as categorical or qualitative attributes. these attributes though represented by numbers must be treated like symbols.
They lack most of the properties of numbers.

→ Interval and ratio attributes are collectively referred to as quantitative or numeric attributes. They can be integer values or continuous.

Important properties of these attributes

1) Each attribute type possesses all the properties and operations of the attribute types above it. i.e. any operation that is valid for nominal, ordinal and interval attributes is also valid for ratio attribute.

Describing attributes on the basis of number of values

We have two types of attributes :-

1) Discrete :- It has a finite set of values. For e.g - categorical attributes such as zipcode, id no, count attributes. They generally have 2 values e.g - TRUE or FALSE

YES or NO, 0 or 1 etc. They are represented either by using boolean or integer variable.

2) Continuous:- It's values are real numbers and are represented by floating point variables. Eg - temperature, height, weight etc.

* Assymetric Attributes :-

e.g- Consider a dataset with student records where each attribute records whether or not a particular student took a certain subject at the university. 1 value will signify that student has taken that particular subject and 0 specify student hasn't taken that particular subject. As the students will take small no. of all available subjects most of the values in dataset would be 0 so it's more imp to focus only on non-zero values. these types of attr. are called assym. binary attributes and are particularly imp. for association analysis.

* Types of DataSet

General Characteristics of datasets :-

(1) Dimensionality :- is the no. of attributes that the objects in a dataset possess. Data with lesser no. of dimensions tends to be qualitatively better than the moderate or high dimensional data. The difficulties associated with analyzing high dimensional data is referred to as "curse of dimensionality".

(2) Sparsity :- In some datasets in asymmetric features most attributes of the object have values → 0 practically it helps in saving computation time and storage becoz only non 0 values need to be stored and manipulated.

(3) Resolution :- The properties of data are different at different resolutions. e.g- surface of earth seems very uneaven at a resolution of a few metres but is relatively smooth at the resolution of a few km's if the resolution is to find a pattern may not be visible or may be buried in noise. if resolution is too coarse the pattern may disappear.

* Types of Datasets (Generally Asked)

(1) Record Data :- Here dataset is collection of records each of which consists of a fixed set of data themes. It has further sub types :-

a) Transaction / Market Basket Data → columns are items and rows are transactions

b) Data Matrix

c) Sparse Data Matrix → e.g-

a) Transaction data – is a special type of record data where each record involves a set of items purchased by a customer. This type also called market basket data attributes may be binary indicating whether or not an item was purchased or can be discrete or continuous such as the no. of items purchased or the amount spent on those items.

b) Data Matrix :- In it the data objects is a collection of data having the same fixed set of numeric attributes a set of such objects can be interpreted as an M x N matrix where there are M rows and N columns. The data matrix is the standard data format for most statistical data.

c) Sparse data matrix :- special case of data matrix in which the attributes are of same type and are asymmetric i.e. only non-zero values are important. for e.g- document data where document can be represented as a vector and each word as the attribute. the value of each component is no. of items the corresponding word occurs in the document. It's also called document term matrix.

(2) GRAPH BASED DATA

It is convenient and powerful representation of data. It has two specific types:

a) the graph captures the relationships among the data objects
b) the data objects themselves are represented as graphs.

→ The relationships among objects convey important info. where the data objects are mapped to nodes on the graph while the relationships among the objects are captured by the links b/w the objects and link properties. for e.g- webpages on WWW contain both text and links to other pages

→ Data with objects as graphs :- If objects have structure i.e. the objects contain sub objects that have relationships then such objects are frequently represented as graphs for e.g- the structure of chemical compounds can be represented by graphs where the nods are atoms and the link b/w the nods

are chemical bonds.

↓

A graph representation makes it possible to determine which substructures occur frequently in a set of compounds and to ascertain whether the presence of any of these sub-structures is associated with the presence or absence of certain chemical properties.

Substructure Mining → is a branch of data mining that analyses such data.

③ ORDERED DATA

For some types of data the attributes have relationships that involve order in time or space

a) Sequential data :- It's also referred as temporal data where each record has a time associated with it. for e.g- in a retail transaction data the time of the transaction is also stored. This time information makes it possible to find the patterns regarding which products sale peaks during a particular season or festival.

b) Sequence data :- Consists of a dataset that is sequence of individual entities such as sequence of words or letters there are no timestamps involved. e.g- the genetic info. of plants and animals can be represented in the form of genes.

c) Time series data :- it's a special type of sequential data in which each record is a time series i.e. a series of measurement taken over time for e.g- ① financial dataset that contains the time series of the daily prices of various stocks. ② time series of the average monthly temperature of a particular city during the years 2006 - 2010. In temporal data there's temporal auto-coelation i.e. if two measurements are close in time then the values of those measurements are often very similar.

d) Spatial Data :- It's a data collected at different geographical locations for e.g- weather data In spatial data there's spatial autocorrelation i.e. objects that are physically close tend to be similar e.g- 2 points on the earth that are geographically close to each other have similar values for temperature and rainfall.

* HANDLING NON-RECORD DATA

Record Oriented Techniques can be applied to non-record data by extracting features from the data objects and using these features to create a record corresponding to each object. for e.g- given a set of common substructures each compound can be represented as a record with binary attributes that indicate whether a compound contains a specific substructure.

* DATA QUALITY

Data mining focuses on :- 1) the detection & correction of data quality problems. also called data cleaning

2) the use of algorithms that can tolerate poor data quality.

→ Measurement and data collection issues

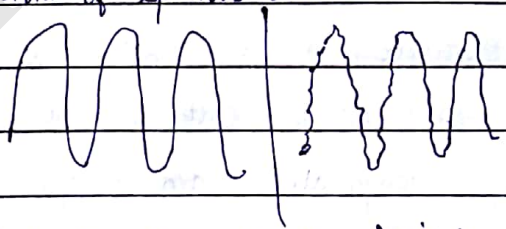
There may be problems due to human error, limitations of measuring devices or flaws in the data collection process, values or even entire data objects may be missing (erroneous) there may be spurious or duplicate objects, all the data may be present but there may be inconsistencies. for e.g- a person with height 2m has weight of only 2 kg

1) Measurement Error :- It occurs when the value recorded in the measurement process differs from the true value to some extent.

2) Data collection Error :- refers to errors such as omitting the data objects or attribute values or inappropriately including a data object.

→ Noise and Artifacts

Noise is the random component of a measurement error involving distortion of a value or addition of spurious objects



The elimination of noise is difficult in DM so our main focus is on making robust algorithms that produce acceptable results even when the noise is present

Data errors which are repeated for every record such as a mark (→ strange) in the same place on a set of photos such distortions of the data are often referred to

as artifacts.

→ Precision, Bias and accuracy

In statistics & experimental science the quality of measurement process is measured by precision and bias.

Precision:- The closeness of repeated measurements to one another. Precision is often measured by the standard deviation of a set of values

Bias:- A systematic variation of measurements from the quantity being measured. It is measures by taking difference b/w the mean of the set of values & known value of the quantity being measures.

Accuracy:- The closeness of measurement to the true value of the quantity being measured. Without understanding the accuracy of data & results an analyst can commit serious data analysis blunders.

eg. Standard deviation $\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$

e.g- Consider a sample of IQ scores 96, 104, 126, 134, 140

$$126 + \frac{-30 - 22 + 0 + 8 + 14}{5}$$

$$\frac{122}{-96}$$
$$6$$

$$126 + -6 = 120$$

$$\frac{(26)^2 + (18)^2 + 4^2 + 12^2 + 8^2}{4}$$

$$\sqrt{\frac{(34)^2 + (16)^2 + 6^2 + 14^2 + 20^2}{4}} = 19.13$$

5

Find SD of first 10 natural no.

$$\sqrt{\frac{(-4.5)^2 + (3.5)^2 + (2.5)^2 + (1.5)^2 + (0.5)^2 + (0.5)^2 + (1.5)^2 + (2.5)^2 + (3.5)^2 + (4.5)^2}{9}}$$

$$= 3.027$$

Suppose that we have a standard laboratory weight with mass of 1g and want to access the precision and bias of the laboratory scale we weigh the mass 5

times and values 1.015, 0.990, 1.013, 1.001, 0.986

$\bar{x} = 1.001$

1.000

0.001 → bias

= 0.0013095 [precision]

1.001
- 0.990
0.011

1.013
- 1.001
.12

1.001
- 0.986
015

0.00017/5

~ * OUTLIERS

are the data objects that have the characteristics which are different from most of the other data objects in the dataset. The values of an attribute are unusual w.r.t. the typical values of that attribute. Outliers are legitimate data objects or values that may sometimes be of interest for e.g. in fraud and network intrusion detection the goal is to find unusual objects or events from among a large no. of normal ones.

~ * Missing VALUES

It is not unusual for an object to have one or more missing attribute values

1 Reasons :-

1) The info. was not collected. for e.g. some people decline to give their age or weight.

2) Some attributes are not applicable for all the objects e.g. in library there's a student record which has the attributes student name, id, mobileno, course, booksissued, returndate and fine. If a student hasn't issued any book booksissued, fine and returndate columns are invalid to him.

There are several strategies for dealing with missing values :-

1) Eliminate the data objects or attributes. A simple and effective strategy is to eliminate the objects with missing values. If partially specified data object contains some info. and if many objects have missing values then reliable analysis can be impossible. If we eliminate attributes that have missing values then the eliminated attributes may be the ones that are critical to analysis.

2) Estimate missing values - sometimes missing data can be reliably estimated for eg. Consider a dataset that has many similar data points the attribute values of the points closest to the point with a missing value are often used to estimate

the missing value. If the attribute is continuous average attribute value of the nearest neighbours is used if it's categorical then most commonly occuring attribute value can be taken.

3) Ignore the missing values during analysis – Many DM approaches can be modified to ignore the missing values. Suppose the objects are being clustered & the similarity b/w pairs of data objects needs to be calculated so here the approx similarity can be calculated by using only those attributes that don't have those missing values. this degree of inaccuracy may not matter much provided the total no. of attributes are large and the no of missing values isn't very high.

(6) → Inconsistent Values

Data can contain inconsistent values. for e.g- Consider an address field where the pincode area specify is not present in the city. so maybe a digit was misread when the info. was scanned so it's important to detect and if possible correct such problems. Some inconsistencies are easy to detect. for e.g- a person's height and age shouldn't be -ve. Sometimes there are discrepancies in a dataset but that doesn't mean the data shouldn't be used only the analyst should consider the potential impact of such discrepancies on the data mining analysis.

(7) → Duplicate Data

A dataset may include data objects that are duplicates. for e.g- many people recieve duplicate mails bcoz they appear multiple times in a DB with slightly different names. To detect and eliminate such duplicates inconsistent values must be resolved also care needs to be taken to avoid accidentally combining the data objects that are similar but not duplicates. such as 2 distinct people with identical names. The term D-Duplication is often used to refer to the process of dealing with these issues.

Data Quality

→ * ISSUES RELATED TO APPLICATION OF COLLECTED DATA

(1) Timeliness:-

(2) Relevance

(3) Knowledge about data.

(1) Timeliness:- Some data starts to age as soon as it has been collected. for eg- the data of purchasing behaviour of customers or web browsing patterns represents reality for only a limited time. If the data is out of date then so are the models and patterns based on it. for e.g- stock market trends for the next day

(2) Relevance:- The available data must contain the info. necessary for application for e.g- the task of building a model that predicts accident rate for drivers if the info about age and gender of the driver is absent or omitted then it's likely that the model will have limited accuracy. for e.g- a survey was held to see how many people would prefer to have fast food in lunch instead of home cooked food that survey didn't include the individuals from age group 15-25. So the survey won't serve the purpose as we have excluded an important part of our sample.
Another problem is "sampling bias" which occurs when a sample doesn't contain different types of objects in proportion to their actual occurence in the population. Sampling bias will result in an erroneous analysis because the results of data analysis will reflect only the data that's present.

(3) Knowledge about the data:- The datasets are accompanied by documentation that describes different aspects of data like precision, types of attributes etc. The quality of the documentation can help or hinder the subsequent analysis. for e.g- if the documentation fails to tell us that missing values for a particular field are indicated with 9999 then our analysis of the data may be faulty

* DATA PREPROCESSING

1) **Aggregation:-** means combining of two or more objects into a single object. for eg- Consider a dataset consisting of transactions recording the daily sales of products in various store locations one way to aggregate transactions for this dataset is to replace all the transactions of a single store with a single store wide transaction. this reduces the 100's and 1000's of transactions to a single daily transaction and the no. of data objects is reduced to the

no. of stores. Q How an aggregate transaction is created i.e. how the values of each attribute are combined ?

Ans → Quantitative attributes such as price are aggregated by taking sum or average, qualitative attributes such as items can be summarized as the set of all the items that were sold at that location. We can also reduce the possible values for ~~days~~ dates from 365 days to 12 months. This type of aggregation is commonly used in OLAP (online Analytical Processing) systems. OLAP tools enable users to analyze multidimensional data from multiple perspectives.

Advantages of Aggregation :-

a) The smaller datasets resulting from data reduction require less memory and processing time which permits the use of more expensive data mining algorithms.

b) It provides a higher level view of data.

c) The behaviour of group of objects or attributes is now more stable than that of individual objects or attributes. for e.g- the average yearly precipitation has less variability than the average monthly precipitation for any given particular loc^n over a given period of time.

Disadvantages of aggregation is the potential loss of interesting details for e.g- which day of the weak has the highest sale for a particular product

2) Sampling

Sampling approach is used for selecting a subset of the data objects to be analyzed. A sample must be representative i.e. having the same property as original set of data. for e.g- mean of the sample data must be the same as the mean for the entire dataset. We must choose a sampling scheme that guarranteed a high probability of getting a representative sample.

Different sampling approaches :-

a) Simple Random sampling
   - With replacement
   - Without replacement

b) Stratified Sampling

c) Progressive Sampling

① Simple Random Sampling

There is an equal probability of selecting any particular item

It's of two types

a) Sampling without Replacement → as each item is selected it's removed from the set of all the objects that constitute the population.

b) " With " → The objects are not removed from the population as they are selected for the sampling in this method the same object can be picked more than once. In this the probability of selecting any object remains constant during the sampling process.

Problem in Simple Random Sampling :-

When the population consists of diff. types of objects with widely different no's of objects simple random sampling can fail to adequately represent those type of objects that are less frequent this can cause problems when the analysis requires proper representation of all object types. so in stratified sampling equal no. of objects are drawn from each group even though the groups are of different sizes. A slight variation can be done in this method such that the no. of objects drawn from each group is proportional to the size of that group.

② Stratified sampling →

~~② Stratified Sampling~~

Q. How to choose the correct sample size ?

Ans. It's necessary to choose correct sample size larger ss increases the probability that a sample will be representative but also eliminate the advantage of sampling with smaller sample sizes patterns may be missed or erroneous patterns (may be detected.

To determine proper ss it requires a methodological approch as follows:

③ ~~Progressive Sampling~~ Suppose we're given a set of data that consists of almost equal sized groups then we need to find atleast 1 representative point for each of the groups assuming the objects in each group are highly similar to each other

SS → Sample Size

Page No. Date

③ Progressive Sampling

The proper SS can be difficult to determine so adaptive or progressive sampling schemes are used these approaches start with a small sample and then increase the SS until a sample of sufficient size has been obtained. This technique eliminates the need to determine the correct SS initially but there must be a way to evaluate the sample to Judge if it's large enough.

In progressive sampling how will we come to know that we've got correct SS :-

2, 3, (5), 7, 9, (4), (6), 8., (7), 5, 4, 6. $\frac{66}{12} = 5.5$

Mean = 5.5

5, 4, 6, 7 → Mean = 5.5 so it's a representative sample

If We're trying to find a pattern in which we've total 3000 records so first we take 500 records as our sample and get a very faded pattern so we take some more samples say 300 so new SS = 800 now We get somewhat clear pattern so again we need to check whether it's correct SS ? We're not missing out an imp. part of our pattern so we further increase our SS by 200 now we're getting the same pattern as before but is more clear so We'll come to know that the SS = 800 was adequate to form a representative sample.

3) DIMENSIONALITY REDUCTION

Datasets can have a large no. of features for e.g - in document dataset

The advantages :-

a) Many DM algorithms work better if the dimensionality is lower hence dimensionality reduc" can eliminate irrelevant features and reduce noise.

b) A reduction of dimensionality can lead to a more understandable model as it may allow the data to be more easily visualised

c) The amount of time and memory required by the DM algorithm is reduced with reduction in dimensionality.

DR can be done in 2 ways :-

a) Creating new attributes that are combination of old attributes. e.g - the student marks in which we can combine all the marks column to 1 column CGPA.

b) Selecting new attributes that are a subset of old is known as featured

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

subset selection or feature selection.

Curse of Dimensionality :- (VVImp)

As dimensionality increases the data bcomes increasingly sparse in the space that it occupies as a result many clustering and classification algorithms have reduces classification accuracy and poor quality clusters.

4) FEATURED SUBSET SELECTION

This is generally useful if redundant and irrelevant features are present. Redundant features duplicate much or all of the info. contained in one or more attribute for e.g- age and date of birth attributes.

Irrelevant features contain almost no useful information for the DM tasks. for e.g- Student id's are irrelevant to the task of predicting student's marks.

The ideal approach to features selection is to try all possible subsets of features as input to DM algorithms and then take the subset that produces the best result. Since the no. of subsets involving n attributes in 2^n such an approach is impractical in most situations. Theare are 3 approaches to featured selection:-

a) Embedded Approach :- When the DM algorithm itself decides which attributes to use and which to ignore. e.g- Decision Tree classifiers

b) Filter Approach :- Features are selected before the DM algorithm is run using an approach that's independent of DM tasks. e.g- selection of the set of attributes whose pairwise correlation is as low as possible.

c) Wrapper Approach :- These methods use the target DM algorithm to find the best subset of attributes without enumerating all possible subsets. We may identify and run algorithm only on some of the best possible subsets.

5) FEATURE CREATION

It's possible to create a new set of attributes from the original attributes that captures the important information in a dataset much more ~~effectively~~ effectively. the no. of new attributes can (or must) be smaller than (no. of) that of original attributes. one of the methods for creating new attribute is FEATURE EXTRACTION. The creation

Page No. Date

of new set of features from the original raw data is known as feature EXTRACTION. It's application is found in image processing where each photograph is to be classified according to whether or not it contains a human face.

~~DISCRETIZATION~~

6) DISCRETIZATION AND BINARIZATION

Categ. Value	Integer value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
ok	2	0	1	0
good	3	0	1	1
great	4	1	0	0

Binarization →

x_1	x_2	x_3	x_4	x_5
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

(0, 110) → 10 split → (0-10) (10-20) - - - - (100-110)
A B G
Discretization

Some DM algorithms especially classification algorithms requires the data to be in the form of Categorical attributes. algorithms that find association patterns require that the data be in the form of binary attributes so it's often necessary to transform a continuous attribute into a categorical attribute called DISCRETIZATION and transforming both continuous and discrete attributes into one or more binary attributes called BINARIZATION.

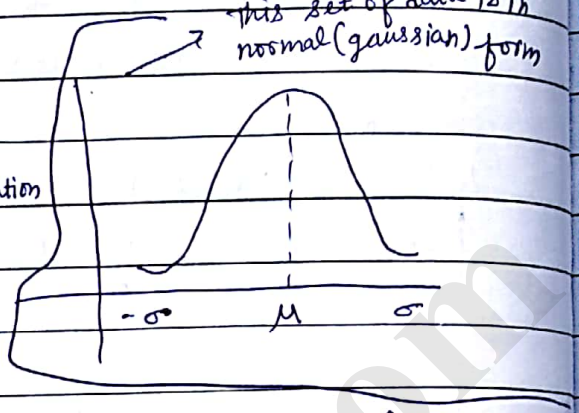
BINARIZATION → A simple technique to binarize a categorical attribute is if there are m categorical values then uniquely assign each original value to an integer in the interval $[0, m-1]$ if the attribute is ordinal then order must be maintained by the assignment.

In 1st table the transformation can cause complications such as creating unintended relationships among the transformed attributes. Here the attributes x_2 and x_3 are irrelevantly correlated. for association problems it's therefore necessary to introduce one binary attribute for each categorical value.

DISCRETIZATION OF CONTINUOUS ATTR. — the transformation of cont. attr. to categorical attribute involves 2 subtasks :-

a) Deciding how many categories to have b) how to map the values of cont. attr. to these categories. In the first step the values are divided into n intervals

by specifying n-1 split points. In the 2nd step all the values in one interval are mapped to the same categorical value.



7) Variable Transformation

A Variable transformation refers to a transformation that is applied to all the values of a variable.
In Variable transformation a simple mathematical function is applied to each value individually.
If x is a Variable then examples of transformation include :-

x^k, $\log(x)$, e^x, $\sqrt{x}$, $\frac{1}{x}$ etc.

Variable transformations especially (sqrt, log(x), $\frac{1}{x}$) are often used to transform the data that doesn't have the Gaussian or Normal distribution into the data that does.

↑ Speciality in Normal distribution :- [How normal distribution is different from others]

→ In normal distribution the data tends to be around a central value with no bias left or right. This distribution has bell shaped density curve described by the mean and standard deviation. the density curve is symmetrical about the mean.

e.g → of Variable transformation –

If the no. of data bytes that are transferred in a session ranges from 1 to 10^8 billion then it may be advantegeous to Compress by using $\log_{10}$ transformation

I → 10^8	II → 10^9	$\|8-9\| = 1$	So I & II are more similar in terms of no. of bytes transferred per session beoz they represent transfer of larger files.
III → 10	IV → 1000	$\|1-3\| = 2$	

Variable transfmn should be applied with caution since they can change the nature of data - for e.g the transformation $\frac{1}{x}$ reduces the magnitude of values

SD → Standard Deviation

that are 1 or larger but increases the magnitude of values b/w 0-1.

→ Normalization or Standardization → Their goal is to make an entire set of values have a particular property.

e.g → Consider comparing people on basis of two variables age and income. for any 2 people the difference in income will likely be much higher than the difference in age. If the diff. in the range of values of age and income are not taken into account the comparison b/w the people will be dominated by the differences in income so transformation is often necessary to avoid having a variable with large values dominate the result of calculn.

→ The mean and SD are strongly affected by outliers so mean generally replaced by median and SD by absolute sd.

* Dissimilarities b/w Data objects

→ Distances :- The euclidean distance b/w 2 points x & y in n-dimensional space is

$$d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$ Where n = no. of dimensions and x_k and y_k are the kth attributes of x & y.

The Euclidean distance is generalised by Minkowski distance matrix

$$d(x,y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$ where r is a parameter. Euclidean special case of Minkowski when r = 2

→ When r = 1 it's called Hamming (or Manhattan) Distance (or L_1 norm) which is the no. of bits that are different b/w two objects that have only binary attributes

→ When r = 2 it's Euclidean distance also called L_2 norm

→ When r = ∞ it's supremum distance $d(x,y) = \lim_{r \to \infty} \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$

↳ L_{max} or L_∞ distance

$$d(x,y) = \max_{k=1}^{n} |x_k - y_k|$$

The Euclidean, Manhattan and Supremum distances are defined for all values of n and specify different ways of combining the differences in each dimension into an overall distance.

18th Feb – DM Test [Ch 1,2]

If $d(x,y)$ is distance b/w 2 points x & y then following properties hold:-

(1) a) Positivity $d(x,y) \geq 0 \quad \forall x,y$

(2) b) $d(x,y) = 0$ if $x = y$

(2) Symmetry

$$d(x,y) = d(y,x) \quad \forall x,y$$

(3) Triangle Inequality

$$d(x,z) \leq d(x,y) + d(y,z) \quad \forall x,y,z$$

Measures that satisfy all the three properties are known as METRIC.

Point	x Coordinate	y Coordinate
P1	0	2
P2	2	0
P3	3	1
P4	5	1

$d(P_1, P_2) = |0-2| + |2-0| = 4$

$d($

L1 norm matrix

	P1	P2	P3	P4
P1	0	4	4	6
P2	4	0	2	4
P3	4	2	0	2
P4	6	4	2	0

5
5 | 26
25
1

(L2 norm) matrix

	P1	P2	P3	P4
P1	0	$\sqrt{8}=2.82$	$\sqrt{10}$	$\sqrt{26}$
P2	$\sqrt{8}=2.82$	0	$\sqrt{2}=1.41$	3.16
P3	$\sqrt{10}=3.14$	$\sqrt{2}=1.41$	0	2
P4	$\sqrt{26}$	3.16	2	0

(L∞ norm)

	P1	P2	P3	P4
P1	0	2	3	5
P2	2	0	1	3
P3	3	1	0	2
P4	5	3	2	0

Page No. Date.

Q. Calculate L_1 distance matrix, euclidean, supremum distance matrix for the following points

	x	y
P1	2	1
P2	3	4
P3	0	1
P4	2	4

Hamming

$r=1$

	P1	P2	P3	P4
P1	0	4	2	3
P2	4	0	6	1
P3	2	6	0	5
P4	3	1	5	0

1.41
3
4 2 3

Euclidian

$r=2$

	P1	P2	P3	P4
P1	0	$\sqrt{10}=3.16$	2	3
P2	$\sqrt{10}=3.16$	0	$3\sqrt{2}=4.24$	1
P3	2	$3\sqrt{2}=4.24$	0	$\sqrt{13}=3.605$
P4	3	1	$\sqrt{13}=3.605$	0

4+9

Supremum

$r=\infty$

	P1	P2	P3	P4
P1	0	3	2	3
P2	3	0	3	1
P3	2	3	0	3
P4	3	1	3	0

→ NON MATRIX DISSIMILARITIES

Set differences

$A = \{1,2,3,4\}$

$B = \{2,3,4\}$

$A-B = \{1\}$ $\quad d(A-B) \neq d(B-A)$

$B-A = \{\} = \phi$ $\quad d(A,B) \neq d(B,A)$

so A & B are not similar.

Test → 18th Feb

$d(A,B) = d(A-B) + d(B-A)$

$d(B,A) = d(A,B)$ now symmetry properties will hold

↳ 4 mark question to check whether set A & B are similar if they are not then make them similar.

Time

$$d(t_1,t_2) = \begin{cases} t_2 - t_1, & \text{if } t_1 \le t_2 \\ 24 + (t_2 - t_1), & \text{if } t_1 \ge t_2 \end{cases}$$

$d(1PM, 2PM) = 1$
$d(2PM, 1PM) = 23$ → non-symmetric

$d(t_1,t_2) = |t_1 - t_2|$
↳ now it became symmetric

* Similarity Between Data Objects

if $s(x,y)$ is similarity b/w the points x & y then the properties of Syring are as follows:-

① $s(x,y) = 1$ if $x = y$

② $s(x,y) = s(y,x)$, $\forall\, x \& y$

e.g→ Consider an experiment in which people are asked to ~~classify~~ classify a small set of characters as they flash on screen. Here each character is classified as itself or as another character. for e.g- "0" appeared 200 times and was classified as O 160 times and as ○ 40 times Suppose ○ appeared 200 tims and was classified as → small ○ 170 time but as O 30 times. Find out a similarity measures that is symmetric.

Ans $s(O,O) = (160, 40) = 160 + 40 = 200$

$s(○,○) = (170, 30) = 170 + 30 < 200$

New → similarity measure

$s'(x,y) = s'(y,x) = s(x,y) + s(y,x)$

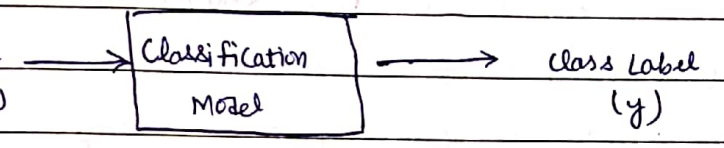
→X——X——X→

It's the end of test Syllabus.

CHAPTER 4

CLASSIFICATION

Classification is the task of assigning objects to one of the several predefined categories e.g- detecting spam e-mail messages based upon the message header and content. The i/p data for a classification task is a collection of records. Each record is characterized by a tuple (x,y) where x is the attribute set and y is a special attribute called as CLASS LABEL.



Consider a sample dataset Vertebrates data set

Name	Bodytemp	Gives Birth	Has Legs	Aerial Creature	Class Label
Human	Warm	Yes	Yes	No	Mammal
Frog	Cold	No	Yes	No	Amphibian
Bat	Warm	Yes	Yes	Yes	Mammal
Pigeon	Warm	No	Yes	Yes	Bird
:	:	:	:	:	:
:	:	:	:	:	:

A sample dataset used for classifying vertebrates into one of the following category :- (Mammal, bird, fish, reptile, amphibian) The attr. set includes the properties such as body temperature, ability to fly, method of reproduction etc. The class label must be a discrete attribute as it's a key characteristic that distinguishes classification from Regression. In Regression class label is continuous.

→ Classification is task of learning a target function f that maps each attribute set x to one of the predefined class label y. The target func^n is also known as classification model. A Classif^n model is useful for :-

1) descriptive modelling :- A CM can serve as an explanatory tool to distin-guish b/w objects of different classes.

2) predictive modelling :- this model can be used to predict class label of unknown records.

CM → classification model

predictive modelling →

A CM can be treated as a blackbox that automatically assigns a class label when presented with an attribute set of an unknown record

→ Classification techniques are most suited for predicting or describing the datasets with binary or nominal categories and are less effective for ordinal categories bcoz they don't consider the implicit order among the categories.

→ A classification technique is a systematic approach to building classification models from an input dataset. Some examples are decision tree classifiers, rule based classifiers, neural networks, support vector machines, naive Bayes classifiers. Each technique employs a learning algorithm to identify a model that best fits the relationship b/w the attribute set & class label of the input data and correctly predicts the class label of records.

→ For solving classification problems first a training set consisting of records whose class labels are known must be provided. The training set is used to build a CM which is subsequently applied to the test set, which consists of records with unknown labels evaluation of the performance of a classifier is based on the counts of test records predicted correctly or incorrectly by the model. the counts are tabulated in a table known as CONFUSION MATRIX. Each entry f_{ij} in the table denotes the no. of records from class i predicted to be class j.

Confusion Matrix

		Predicted	
		Class = 1	Class = 0
Actual	Class = 1	f_{11} (TP)	f_{10} (FN)
	Class = 0	f_{01} (FP)	f_{00} (TN)

TP → True Positive
FN → False Negative
TN → True Negative
FP → False Positive

To determine how well a classification model performs we may summarise the information with a single no. that would make it more convenient to compare the performance of different models. this can be done using performance metric.

$$\text{Accuracy} = \frac{f_{11} + f_{00}}{f_{00} + f_{01} + f_{10} + f_{11}} = \frac{\text{no. of correct predictions}}{\text{Total no. of predictions}}$$

(E) Error rate = $\frac{\text{no of wrong predictions}}{\text{Total no. of predictions}} = \frac{f_{01} + f_{10}}{f_{00} + f_{11} + f_{10} + f_{01}}$

6	4
2	8

$\text{Accuracy} = \frac{14}{20} = \frac{7}{10} = 0.7$

$\text{Error rate} = \frac{3}{10} = 0.3$ 9.09

Q Suppose there are 2 predicted classes Yes or NO. Classifier made total of 165 predictions out of which the classifier predicted yes → 110 times and NO → 55 times. In reality 105 patients in the sample had the disease and 60 doesn't. The classifier correctly predicted the no. of patients who had the disease as 100. Calculate accuracy and error rate

Ans

	Predicted: Yes	NO	
Actual Yes	100	5	105
NO	10	50	60
	110	55	

$\text{accuracy} = \frac{150}{165} = \frac{10}{11} = 0.909$

$\text{error} = \frac{15}{165} = \frac{1}{11} = 0.09$

Q Evaluating Spam classifier Consider a classical problem of predicting spam and non-spam email by using binary classification model. Classifier made 100 predictions are modelled Correctly classified 95 emails → 85 as non-spam and 10 as spam 5 emails were actually spam but predicted as non-spam. Calculate error rate and accuracy.

		Predicted: Spam	non-spam
Actual	Spam	5	5
	non-spam	0	85

$\text{accuracy} = \frac{95}{100} = 0.95$

$\text{error} = 0.05$

95

Q A binary classification model predicted it to be a cat 55 times dog 45 actual both 50. The model predicted the cat incorrectly 40 times. Find error and accuracy rate.

Accuracy = 0.25

Error rate = 0.75

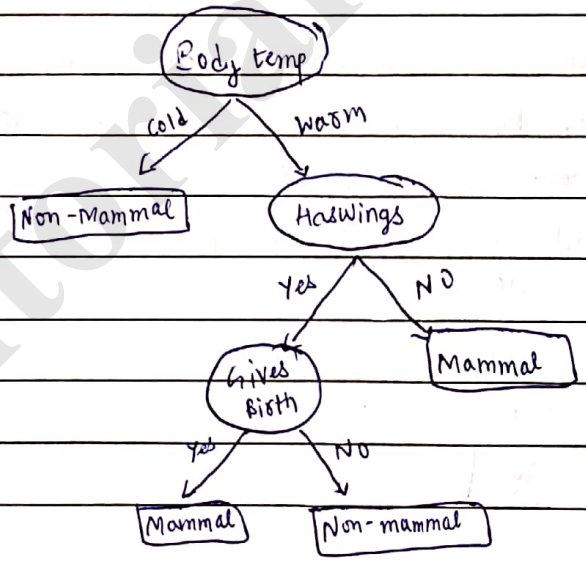
Actual	Predicted: Cat	Predicted: dog	
cat	15	35	50
dog	40	10	50
	55	45	

Q A binary classification model predicted fraud 7 times and not fraud 93 times in reality (6, 94). The model correctly predicted fraud as 4 times. Find accuracy and error rate.

	Fraud	non-Fraud	
Fraud	4	2	6
non-fraud	3	91	94
	7	93	

accuracy = $\frac{95}{100} = 0.95$

error rate = 0.05



→ Decision tree has 3 types of nodes :-

1) Root node which has no incoming edges and 0 or more outgoing edges.
2) Internal nodes each of which has exactly 1 incoming edge and 2 or more outgoing edges
3) Leaf or terminal nodes each of which has exactly 1 incoming edge and no outgoing edges.

Classifying a test record is straight forward once a decision tree has been constructed. starting from the root node we apply the test condition to the record and follow the appropriate branch based on outcome of the test. This will lead us either to another internal node for which a new test cond^n is applied or to a leaf node. The class label associated with the leaf node is then assigned with the leaf node.

Q HOW TO BUILD A DECISION TREE ?

Ans ~~Hunt Algorithm~~ There are exponentially many decision trees that can be constructed from a given set of attributes finding an optimal tree is computationally infeasible bec= of the exponential size of the search space. Efficient Algos have been developed that employ a greedy strategy to grow a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data Hunt Algorithm is the basis of many existing decision tree algorithms including ID3, CH.5 and CART

* HUNT'S ALGORITHM

→ Here a decision tree is grown in recursive fashion by partitioning the training records into successively purer subsets.

D_t → set of training records associated with a node t

$\{y_1, \ldots, y_c\}$ will be the class labels.

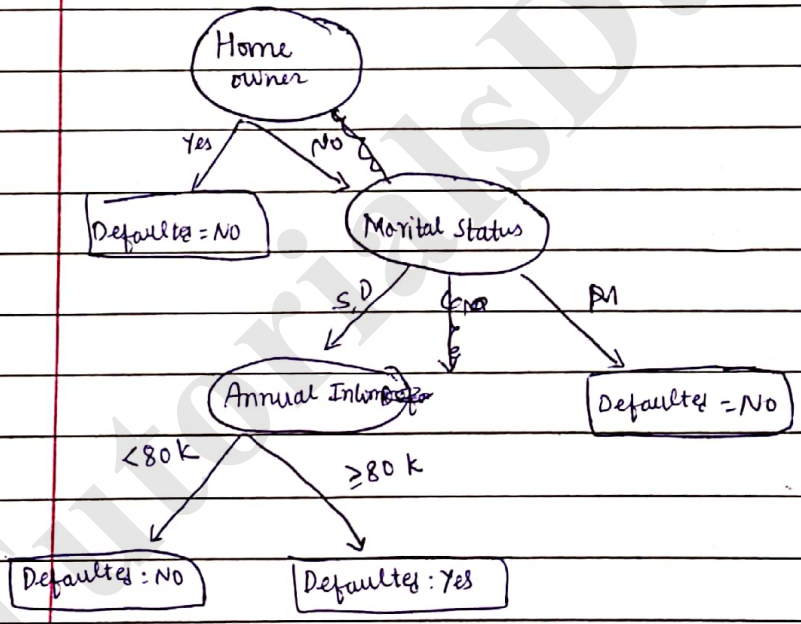
Step1 → If all records in D_t belong to the same class y_t then t is a leaf node labelled as y_t.

Step2 → If D_t contain records that belong to more than 1 class an attribute test cond^n is selected to partition the records into smaller subsets. A child node is

Created for each outcome of the test cond^n and records in D_t are distributed based on the outcomes. The algorithm is then recursively applied to each child node.

The problem of predicting whether a loan applicant will repay his or her loan obligation or becomes defaulter

Tid	Home owner	Marital Status	Annual Income	Defaulted Borrower
1	~~Yes~~	S	125 K	No
2	No	M	100 K	No
3	Yes	S	70 K	No
4	No	M	120 K	No
5	NO	D	95 K	Yes
6	No	M	60 K	No
7	Yes	D	220 K	No
8	No	S	85 K	Yes
9	No	M	75 K	No
10	NO	S	90 K	~~No~~ Yes



Hunt Algorithm will work if every combination of attr. values in present in training data and each combination has a unique class label but in most practical situations additional conditions are needed to handle following ~~that~~

1) If none of the training records have the combination of attribute values associated with some nodes then the node is declared a leaf node with the same

Class label as the majority class of training records associate with it's parent node.

2) If all the records associated with D_t have identical attribute values but different class labels then it's not possible to split records any further and the node is declared a leaf node with the same class label as the majority class.

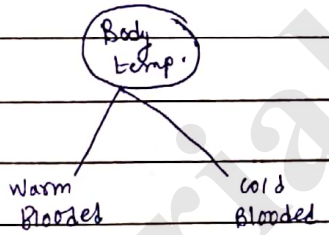
Design issues of decision tree Algo

1) How should the training records be split each recursive step must select an attr. test condition to divide the records into smaller subsets It must also define an objective measure for evaluates the goodness of each test condition.

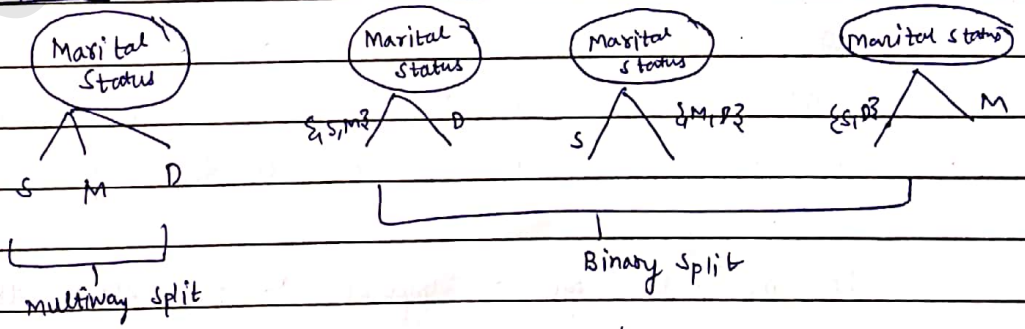
2) How should the splitting procedure stop? A possible strategy is to continue expanding a node until either ∀ the records ∈ same class or all the records have identical attribute values or other criteria may be imposed to terminate the tree growing procedure earlier.

* Methods for Expressing Attribute Test Conditions

1) BINARY ATTRIBUTES



2) NOMINAL ATTRIBUTES



Since a nominal attribute can have many values it's test condition can be expressed in 2 ways first is a multiway split in which the no. of outcomes depends on no. of distinct values for the corresponding attribute. ϕ

second is some decision tree algo. such as CART produce only binary splits

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

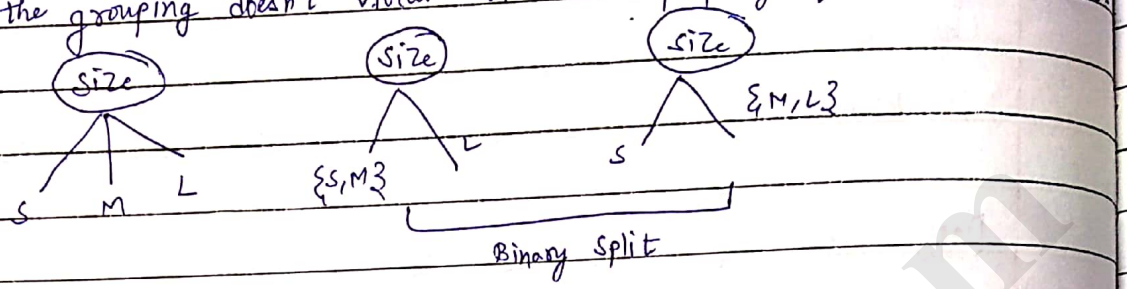
- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

3) ORDINAL ATTR.

They can also produce binary or multiway splits they can be grouped as long as the grouping doesn't violate the order property of attribute values.

(Multiway split)

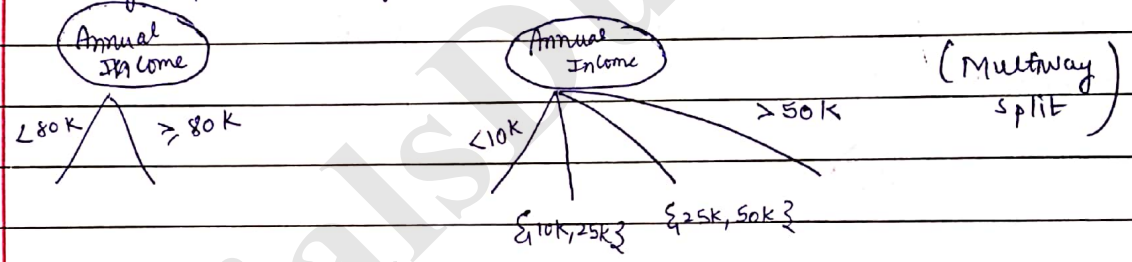


4) CONTINUOUS ATTRIBUTES

For contn. attr. the test condition can be expressed as a comparison test $A < V$ or $A \geq V$ with binary outcomes or a range query with outcomes of the form $V_i \leq A < V_{i+1}$, $i = 1 \dots k$.

For the binary case the decision tree algo. must consider all the possible split positions V and selects the one that produces the best partition. For multiway split the algo. must consider all possible ranges of continuous values.

(Binary split)



* MEASURES FOR SELECTING THE BEST SPLIT

Let $P(i|t)$ denote the fraction of records belonging to class i at a given node t. In a 2 class problem the class distribution at any node can be written as (p_0, p_1) where, $p_0 = 1 - p_1$. The measures developed for selecting the best split are based on degree of impurity of the child nodes.

The smaller the degree of impurity the more skewed the class distribution is. A node with class distribution $(0,1)$ has 0 impurity whereas a node with uniform C.D. $(0.5, 0.5)$ has the highest impurity.

$p(i|t) = \frac{\text{no. of records to class } i}{\text{total no. of records}}$

Different impurity measures :-

$C \rightarrow$ no. of classes

(1) Entropy $(t) = -\sum_{i=0}^{(c-1)} p(i|t) \log_2 p(i|t)$ $\quad [0 \log_2 0 = 0]$

(2) Gini $(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)^2]$

(3) Classification error $= 1 - \max_i^c [p(i|t)]$

Q

Node N_1	Count
class = 0	6
class = 1	6

Find Entropy

Entropy $= -\left[\frac{0}{6}\log_2\frac{0}{6} + 1\log_2 1\right] = 0$ (pure)

Gini $= 1 - \left[\frac{0^2}{6} + \frac{1^2}{6}\right] = 0$ (pure subset)

$p(i|t)_{max} = 1$

Classification error $= 1 - 1 = 0$ (pure subset)

Q

Node N_1	Count
Class = 0	1
class = 1	5

Entropy $= -\left[\frac{1}{6}\log_2\frac{1}{6} + \frac{5}{6}\log_2\frac{5}{6}\right]$

$= -\left[\frac{1}{6}[-\log_2 6] + \frac{5}{6}[\log_2 5 - \log_2 6]\right]$

Gini $= 1 - \left[\frac{1}{36} + \frac{25}{36}\right] = \frac{10}{36} = \frac{5}{18} = .278$

Classification error $= 1 - \frac{5}{6} = \frac{1}{6} = 0.167$

Q

N_3	Count
C = 0	3
C = 1	3

Entropy $= -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right]$

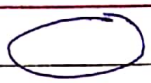
$= -\left[\frac{-1}{2} - \frac{1}{2}\right] = 1$

Gini $= 1 - \left[\frac{1}{4} + \frac{1}{4}\right] = \frac{1}{2} = 0.5$

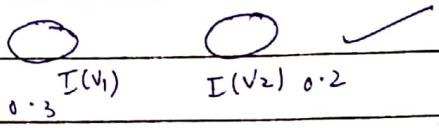
Classification error $= 1 - \frac{1}{2} = \frac{1}{2} = 0.5$

So Node N_1 is the best based on these calculations as it has lowest impurity value followed by N_2 and N_3.

max



min entropy or diff from parent



To determine how well a test condition performs we need to compare the degree of impurity of parent node before splitting with the degree of impurity of the child nodes after splitting. The larger the difference the better the test condition. This difference called Δ (gain).

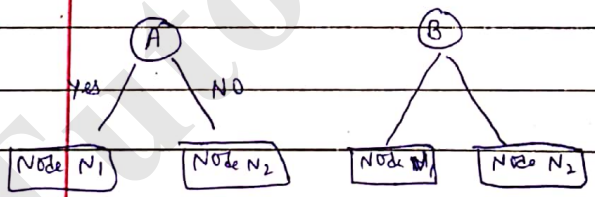
$$\Delta(\text{gain}) = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

N → total no. of records at parent node, k → no. of attributes values

$N(v_j)$ → no. of records associated with child node v_j. Decision tree algo. choose a test condition that maximizes the gain Δ. When entropy is used as impurity measure in the above formula the Δ is known as Δ_{info} or information gain.

* Splitting of binary attributes

	Parent
C_0	6
C_1	6



	N_1	N_2
C_0	4	2
C_1	3	3

	N_1	N_2
C_0	1	5
C_1	4	2

If attribute A is chosen

1) Gini index for N_1 = $1 - \left[\frac{16}{49} + \frac{9}{49}\right] = \frac{24}{49} = 0.489$

Gini index for N_2 = $1 - \left[\frac{4}{25} + \frac{9}{25}\right] = \frac{12}{25} = 0.48$

Weighted average of Gini index for attr. A = $\frac{7(0.489) + 5(0.48)}{12}$

= 0.486

attr A → Δ(gain) = 0.5 − 0.486 = I parent − 0.486

= 0.014

If attribute B is chosen

Gini index for N_1 = $1 - \left[\frac{1}{25} + \frac{16}{25}\right] = \frac{8}{25} = 0.32$

Gini index for N_2 = $1 - \left[\frac{25}{49} + \frac{4}{49}\right] = \frac{20}{49} = 0.408$

Weighted average of Gini index for attr. B = $\frac{5}{12}(0.32) + \frac{7}{12}(0.408)$

= 0.372

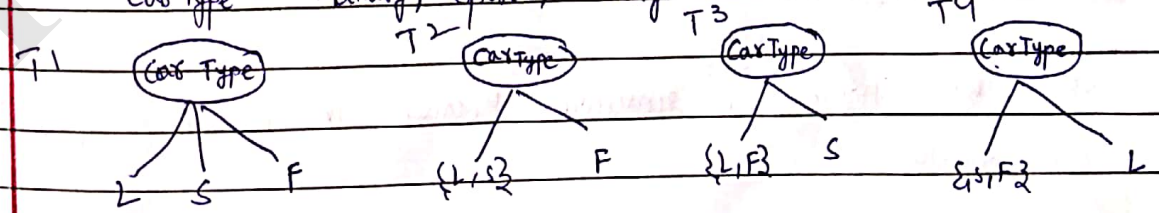
attr B → Δ(gain) = 0.5 − 0.372

= 0.128

So we will choose attribute B as next partitioning attribute.

* Splitting of Nominal attributes

Car Type – Luxury, Sports, Family.



Gini index

	Car Type		
	F	S	L
C_0	1	8	1
C_1	3	0	7
	4	8	8

$F \rightarrow 1 - \left[\frac{1}{16} + \frac{9}{16}\right] = \frac{6}{16} = 0.375$

$S \rightarrow 1 - [1 + 0] = 0$

$L \rightarrow 1 - \left[\frac{1}{64} + \frac{49}{64}\right] = \frac{14}{64} = 0.21875$

Weighted average of Gini index T_1 $= \frac{4}{20}(0.375) + \frac{8}{20}(0) + \frac{8}{20}(0.21875)$

$= 0.163$

	Car Type		
	S	{S,L}	F
C_0		9	1
C_1	1	7	3

$\{S, L\} \rightarrow 1 - \left[\frac{81}{256} + \frac{49}{256}\right] = 0.492$

$\{F\} \rightarrow 1 - \left[\frac{1}{16} + \frac{9}{16}\right] = 0.375$

Weighted average T_2 $= \frac{16}{20}(0.492) + \frac{4}{20}(0.375) = 0.468$

	Car Type	
	{L,F}	S
C0	2	8
C_1	10	0

$\{L, F\} \rightarrow 1 - \left[\frac{4}{144} + \frac{100}{144}\right] = 0.278$

$S \rightarrow 1 - [1 + 0] = 0$

Weighted average T_3 $= \frac{12}{20}(0.278) + \frac{8}{20}(0)$

$= 0.167$

So we will choose T_1 structure because in multiway split the attributes are separated.

* Splitting of Continuous attributes

Annual Income

Class	NO	NO	NO	Yes	Yes	Yes	NO	NO	NO	NO
	60	70	75	85	90	95	100	120	125	220
	55	65	72							
	<=	>								
	0	3								
	0	7								

Annual Income → high prob. → high prob.

Class	No	NO	NO	Yes	Yes	Yes	NO	NO	NO	NO
	60	70	75	85	90	95	100	120	125	220
	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0	3 0
NO	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0	7 0
Gini												
	0.42	0.4	.375	0.342	0.4	0.4	0.3	0.343	0.375	0.4	0.42	

65 → 0

42 → $1-\left[\frac{5}{9}\right]=\frac{4}{9}=0.44$

$\frac{1}{10}(0)+\frac{9}{10}\left(\frac{4}{9}\right)$

72 → $1-\left[\frac{9}{64}+\frac{25}{64}\right]=\frac{30}{64}=0.469$

$\frac{8}{10}(0.469)=0.3$

80 → 0 $1-\left[\frac{9}{49}+\frac{16}{49}\right]=\frac{24}{49}=0489$

$\frac{7}{10}(0.489)$

87 → $1-\left[\frac{1}{16}+\frac{9}{16}\right]=0.375$

$\frac{2}{6}$ $1-\left[\frac{1}{9}+\frac{4}{9}\right]=\frac{4}{9}=0.44$

$\frac{3}{10}(0.375)+\frac{6}{10}(0.44)=0.376$

92 → $1-\left[\frac{4}{25}+\frac{9}{25}\right]=\frac{12}{25}=0.48$

$1-\left[\frac{1}{25}+\frac{16}{25}\right]=\frac{8}{25}=0.32$

$\frac{1}{2}(0.48)+\frac{1}{2}(0.32)$

$=0.4$

97 → 0.5

0.6

110 → $1-\left[\frac{9}{49}+\frac{16}{49}\right]=\frac{24}{49}=0.489$

$1-\left[\frac{9}{64}+\frac{25}{64}\right]=\frac{30}{64}=0.469$

$\frac{3}{9}+\frac{6}{9}$

$\frac{1}{3}+\frac{2}{3}$ 0.

$1-\left[\frac{1}{9}+\frac{4}{9}\right]=\frac{4}{9}=0.44$

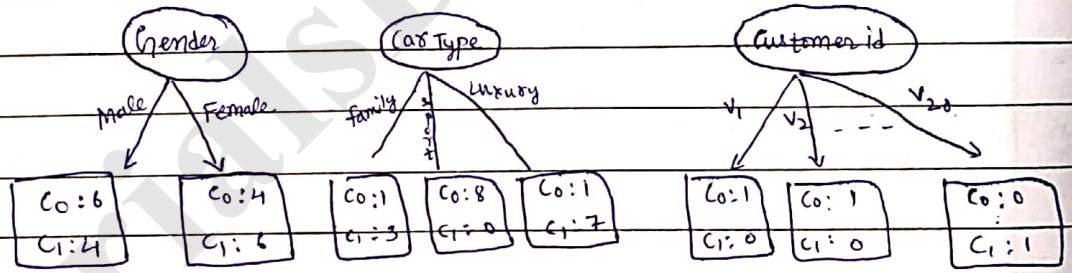
0.9

$1-\left[\frac{9}{100}+\frac{49}{100}\right]=$

A brute force method for finding v is to consider every value of the attribute as a candidate split position for each candidate v the dataset is scanned once to count the no. of records with annual income less than or greater than v. We then compute the Gini index for each candidate and choose the one that gives the lowest value. This approach is computationally expensive because it requires O(n) time to compute the Gini index at each candidate split pos^n since there are n candidates the overall complexity is $O(n^2)$. So to reduce the complexity the training records are sorted based on their annual income which requires O(n log n) time candidate split positions are identified by taking the mid points b/w two adjacent sorted values.

This technique can be further optimized by considered only the candidate split pos^n where located b/w 2 adjacent records with different class labels. This approach allows us to reduce the no. of candidate split positions from 11 to 2.

* GAIN RATIO



Here Customer id is not a predictive attribute because it's value is unique for each record if the no. of records associated with each partition is too small it will not enable us to make any reliable prediction. So there are 2 ways of overcoming this problem :-

1) We can restrict the test conditions to binary splits only. This strategy is used by CART decision tree algorithm.

2) To modify the splitting criteria to take into account the no. of outcomes produced by the attribute test condition. In C4.5 decision tree algo. a splitting criterion known as gain ratio is used to determine the goodness of a split.

$$\text{Gain ratio} = \frac{\Delta \text{info}}{\text{Split info}}$$

$$\text{Split info} = -\sum_{i=1}^{K} P(v_i) \log_2 P(v_i)$$

K → total no. of splits

If an attribute produces a large no. of splits it's split info will also be large which in turn reduces it's gain ratio.

~~If each attribute has same number of records~~

Ex Total no. of records are 21 There are 3 partitions 7 records each. Calculate split info.

Ans $$\text{Split info} = -3\left(\frac{7}{21}\log_2\left(\frac{7}{21}\right)\right)$$

$$= -\log_2\left(\frac{7}{21}\right) = +\log_2(3) = \frac{\log_{10}3}{\log_{10}2} = 1.585$$

		a_1	a_2	a_3	Target class
x	1	T	T	1.0	+
	2	T	T	6.0	+
x	3	T	F	5.0	−
x	4	F	F	4.0	+
	5	F	T	7.0	−
x	6	F	T	3.0	−
	7	F	F	8.0	−
	8	T	F	7.0	+
x	9	F	T	5.0	−

Q What are the information gains of a_1, a_2 and a_3 (Compute for every possible split) what is best split among a_1, a_2 and a_3 according to info. gain?

4:40 4:45 4:50 4:55

$$\text{Parent entropy} = -\left[\frac{4}{9}\log_2\frac{4}{9} + \frac{5}{9}\log_2\frac{5}{9}\right] = 0.991$$

a_1:

	T	F
+	3	1
−	1	4

$$\text{Entropy}^T = -\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right] = 0.811$$

$$\text{Entropy F} = -\left[\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right] = 0.721$$

$$\text{Weighted} = \frac{4}{9}(0.811) + \frac{5}{9}(0.721) = 0.761$$

0.991
−0.761
0.230

$$\Delta \text{info} = 0.230$$

a_2

	T	F
+	2	2
−	3	2

Entropy T $= -\left[\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right] = 0.970$

Entropy F $= -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1$

Weighted entropy $a_2 = \frac{5}{9}(0.970) + \frac{4}{9}(1) = 0.983$

Δ info $= 0.008$

K 2 3 4 5 6 7 8

0.5 1.5 2.5 3.5 4.5 5.5 $-\left[\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log 2\right]$

+ − +

1 3 4 5 6 7 8

	2		3.5		4.5		5.5		6.5		7.5		
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	
+	1	3	1	3	2	2	2	2	3	1	4	0	
−	0	5	1	4	1	4	3	2	3	2	4	1	
	0	0.954	1	0.985	0.918		0.983		0.972		0.888		
	0.848		0.988										

→ 0

$\frac{3}{8}\log\frac{3}{8} + \frac{5}{8}\log\frac{5}{8}$ $\quad$ $\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{2}{3}$

0.954

$\frac{8}{9}(0.954)$

Δ Entropy $= 0.848$

0.991
−0.848
0.143

Δ info $=$

So a_1 will be selected.

* Evaluating the Performance of a Classifier

There are 2 types of errors committed by a Classification model

1) Training Error or Resubstitution error or Apparent Error - It's the no. of misclassification errors committed on training records.

2) Generalization Error :- expected error of the model on previously unseen records or the test Set.

A good model must have low training error as well as low generalization error. A model that fits the training data too well can have a poor generalization error than a model with higher training error such a situation is known as model overfitting.

After building the decision tree a tree-pruning step can be performed to reduce the size of decision tree. To prevent a phenomenon known as overfitting. Pruning improves the generalization capability of decision tree.

It's often useful to measure the performance of the model on the test set because it provides an unbiased estimate of it's generalization error. However to do this the class labels of the test records must be known. Some of the methods commonly to evaluate performance of Classifier are :-

1) Holdout Method :- In this method the original data with labelled examples is partitioned into 2 disjoint sets called the training and testing set. A classification model is then induced from the training set and performance is evaluated on the test set. The proportion of data for training and testing is either 50-50 or $\frac{2}{3}$rd for training and 1/3 rd for testing. The accuracy of the Classifier is based on the accuracy of the induced model on the test ~~data~~ set.

Disadvantages :-

1) The induced model may not be as good as when all the labelled examples are used for training.
2) The model may be highly dependent on composition of training and test set.
3) The training and test sets are no longer independent of each other as they are subsets of the same data.

2) Random subsampling :- The holdout method can be repeated several ~~methods~~ times to improve the estimation of classifiers performance. This approach is

Known as random subsampling

acc_i → accuracy of model

$$acc_s = \frac{1}{k}\left(\sum_{i=1}^{k} \frac{acc_i}{k}\right)$$

Disadvant

1) It doesn't utilise as much data as possible for training.
2) It has no control over the no. of times each sample is used in training or testing set.

3) Cross Validation

In this approach each record is used the same no. of times for training and exactly one for testing

a) 2 Fold Cross Valid^n → in this we partition our data into 2 equal sizes subsets. We choose one of the subset for training and other for testing We then swap the role of subsets and the total error is obtained By suming up the error for both the runs.

b) K Fold Cross Validation → in this method the data is segmented into K equal size partitions during each run one of the partition is used for testing and others are used for training. this procedure is repeated K times so that each partition is used for testing at least once. The total error is obtained by summing up all the K errors.

c) Leave one out approach :- In this K = N (no. of records), in this each test set Contains only 1 record and it has the advantage of utilising as much data as possible for training. 1) It's Computationally to repeat the procedure N times. 2) Since each test set Contains only 1 record the Variance of the performance tends to be high.

4) Bootstrap

In bootstrap approach the training records are sampled with replacement ; on an average a bootstrap sample of size N Contains about 63.2% of the records in the original data.

Probability that a record is chosen out of n samples is $\frac{1}{n}$

Prob that a record is not chosen out of n samples = $1 - \frac{1}{n}$

Prob. that it's not chosen n times = $\left(1 - \frac{1}{N}\right)^n$

Prob that it's chosen at least once by a bootstrap sample = $1 - \left(1 - \frac{1}{N}\right)^n$

also called .632 bootstrap.

Records that are not included in the bootstrap sample become part of the test set. The model induced from the training set is then applied to the test set to obtain the estimate of accuracy of bootstrap sample (ε_i). The sampling procedure is then repeated b times to generate b bootstrap samples.

$$\text{Accuracy} = \frac{1}{b} \sum_{i=1}^{b} (0.632 \times \varepsilon_i + 0.368 \times acc_s)$$

$\varepsilon_i \rightarrow$ accuracy of test set

$acc_s \rightarrow$ accuracy obtained when we run model on training data.

CH-5

RULE BASED CLASSIFIER

* **Rule Based Classifier :→** It is a technique for classifying the records using a collection of if then rules the rules for the model are represented in disJunctive normal form.

$R = (r_1 \vee r_2 \vee \dots \vee r_k)$ Where R is the ruleset and r_i are the classification rules each classification rule can be expressed as.

$r_1 : (\text{Condition})_i \rightarrow y_i$

(GivesBirth = yes) ∧ (BodyTemp = WB) → Mammals

Antecedent — Consequent

$\text{Condition}_i = (A_1 \text{ op } V_1) \wedge (A_2 \text{ op } V_2) \dots (A_k \text{ op } V_k)$

A rule r covers a record x if preCondition of r matches the attributes of x. r is said to be fired or triggered whenever it covers a given record

Quality of a classification rule can be evaluated using coverage and accuracy

Coverage of a rule is defined as the fraction of records in D (dataset) that trigger the rule r

$$\text{Coverage} = \frac{|A|}{|D|}, \qquad \text{Accuracy} = \frac{|A \cap y|}{|A|}$$

|A| → Condition part same as rule

|D| → total no. records

Accuracy → defined as fraction of records triggered by rule r whose class labels are equal to y.

Name	B.T.	GivesBirth	Class Label
human	WB	Y	Mammal
python	CB	N	Reptiles
salmon	CB		

$|A| = 5$, $|Any| = 3$

$|D| = 16$

Coverage $= \frac{5}{16} = 0.3125$ Accu $= \frac{3}{5} = 0.6$

Q How a rule based classifier works ?

A rule based classifies classifies a test record based on the rule trigerred by the record

The two important properties of ruleset generated by a rule based classifier

1) Mutually Exclusive Rules – In this no 2 rules in R are trigerred by same record this property ensures that every record is covered by atmost one rule in R.

2) Exhaustive Rule :- In this every record is covered by at least one rule in R.

These properties ensure that every record is covered by exactly 1 rule.

If the ruleset isn't exhaustive then a default rule $r_d : () \rightarrow y_d$ must be added to cover the remaining cases. A default rule has an empty antecedent and y_d is the default class assigned to the training records not covered by existing rules.

→ If the ruleset isn't mutually exclusive then a record can be covered by several rules some of which may predict conflicting classes. To overcome this problem there are 2 ways :

1) Ordered Rule → An ordered ruleset also known as decision list when a test record is presented it's classified by the highest ranked rule that covers the record. this avoids the problem of having conflicting classes predicted by multiple classifier rules.

2) Unordered Rules → It allows a test record to trigger multiple classifier rules and consider the consequent of each rule as a vote for a particular class. The class label with the highest no. of votes is assigned to test record. It's advantages are :-

a) these are less susceptible to errors caused by the wrong rule being selected to classify a test record.

b) Model building is also less expensive because the rules don't have to be kept in sorted order.

Disadvantag

a) Classifying a test record can be an expensive task beca attr. of test record

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

must be compared against the precond^{n} of every rule in ruleset

b)

* Rule ordering schemes

1) Rule based ordering scheme :- This ensures that every test record is classified by the best rule covering it. the drawback of this scheme is that lower ranked rules are much harder to interpret.

2) Class based ordering scheme :- Here rules that belong to same class appear together in the ruleset. the relative ordering among the rules from the same class is not important. the problem is a high quality rule may be overlooked in favor of an inferior rule that predicts a higher rank class.

* How to Build a Rule Based Classifier

There are 2 methods for extracting classif^{n} rules :-

1) Direct Method - which extracts classif^{n} rules directly from data

2) Indirect method which extract classif^{n} rules from other classif^{n} models such as decision trees

Direct Method for Rule Extraction

The sequential covering algorithm is often used to extract rules directly from data in a greedy fashion based on a certain evaluation measure the algorithm extracts the rules one class at a time and the class selected first depends on no. of factors such as class priority or the cost of misclassifying records from a given class.

During rule extractn all the training records for class y are considered to be +ve example while these that belong to other classes are considered to be -ve example. So a rule is desirable if it covers most of the +ve examples and none of the -ve. Once such a rule is found the training records covered by the rule are eliminated and a new rule is added to the ruleset. This procedure is repeated until the stopping criteria is met (i.e. all records are covered).

NEAREST NEIGHBOUR CLASSIFIER

Some test records mayn't be classified beoz they don't match any training example so we find all the training examples that are relatively similar to the attributes of the test example these are known as nearest neighbours and can be used to determine the class label of the test example. A nearest neighbour classifier represent each e.g. as a data point in D dimensional space where D is the no. of attributes. The data point is classified based on the class label of it's neighbours. Where the neighbours having more than 1 label the datapoint is assigned majority class of it's nearest neighbour.

If K is chosen to be too small then the nearest neighbour classifier may be susceptible to overfitting beoz of noise in training data. If k is too large the nearest neighbour classifier may misclassify the test instance beoz it's list of nearest neighbours may include the points that are located far away from it's neighbourhood.

The algorithm computes the distance b/w each test sample $z = (x', y')$ and all the training examples $(x, y) \in D$. We need to determine the nearest neighbour list D_z Such comput^n can be costly if the no. of training examples is large once the nearest neighbour list is obtained the test sample is classified based on the majority class of it's nearest neighbour

majority voting $y' = \underset{v}{\arg\max} \sum_{(x_i, y_i) \in D} I(v = y_i)$

~~In majority voting approach~~

$v \rightarrow$ class label $\quad y_i \rightarrow$ class label for nearest neighbour

$I \rightarrow$ indicator function that will be 1 if the argument is true and 0 otherwise

In majority voting approach each neighbour has same impact on classification which makes algo sensitive to choice of k so we may reduce the impact of k by assigning weights to the nearest neighbour according to it's distance

$$w_i = \frac{1}{d(x', x_i)^2}$$

As a result the training examples that are located far away will have a less impact on classification as compared to nearest ones

Majority voting $y' = \arg\max_{v} \sum_{(x_i, y_i) \in D} w_i \times I(v = y_i)$

Characteristics of nearest neighbour classifier

1) NNC is an instance based learning which uses specific training instances to make the predictions without having to maintain a model derived from data.
2) It's a lazy learner as it doesn't require model building. but classifying a test example can be quite expensive because we need to compute the proximity value individually b/w test and training examples.
Eager learners often spend most of their computing resources for model building once a model has been constructs classifying a test example is extremely fast
3) These classifiers make their predictions based on local information and are quite susceptible to noise.
4) These classifiers can produce wrong predictions if the scale of attributes isn't taken into account.

Practical Lab Commands

Q5 * Implementing decision tree Algo.

```
library(rpart)
s <- sample(150, 100)
iris_train <- iris[s,]
iris_test <- iris[-s,]
dtm <- rpart(Species ~ ., iris_train, method = "class")
rpart.plot(dtm)
lp <- predict(dtm, iris_test, type = "class")
lp <- iris_test[, 5]
lp <- as.factor(lp)
Confusion Matrix(as.factor(iris_test[,5], Species))
```

not considering species → other attributes

class → classification

Q. Consider a binary classification problem with following set of attributes and attribute values:-

Air_Cond → working, broken, Engine → Good or Bad, Mileage → high, med, low

Rust → yes or no

A rule based classifier produces the following ruleset

Mileage = High → Value = Low

Mileage = Low → Value = High

Air_Cond = working, Engine = good → Value = High

Air_Cond = Working, Engine = bad → Value = Low

Air_Cond = Broken → Value = Low

a) Are the rules mutually exclusive?

Ans No

Mileage	Air_Cond	Engine	Value
High	Working	Good	High/Low

b) Are the rules exhaustive?

Ans Yes

c) Is ordering needed for this set of rules?

Ans Yes, as it's not mutually exclusive.

d) Do u need a default class for ruleset?

Ans No, beoz it's an exhaustive ruleset.

Q. For K nearest

x	0.5	3.0	4.5	4.6	4.9	5.2	5.3	5.5	7.0	9.5
y	-	-	+	+	+	-	-	+	-	-

Consider 1D dataset as shown in the table

a) Classify the datapoint x = 5.0 according to 1, 3, 5 and 9 nearest neighbours (using majority vote)

Ans

1 → 4.9 +

3 → 4.9, 5.2, 5.3 −

5 → +

9 → −

b) Repeat the previous analysis using the distance weighted voting approach.

Ans

1 → +

3 → $\frac{1}{(0.1)^2}$ (+) $\frac{1}{(0.2)^2}$ (−) $\frac{1}{(0.3)^2}$ (−)

(100) 36.11

5 → $\frac{1}{(0.4)^2}$ + $\frac{1}{(0.5)^2}$ + 11 (+)

9 → $\frac{1}{(4.5)^2}$ (−) $\frac{1}{(3.0)^2}$ (−) $\frac{1}{(2)^2}$ (−) (+)

* BAYESIAN CLASSIFIER

In many applications the class label of a test record can't be predicted with certainity even though it's attribute set is identical to some of the training examples this situation may arise because of noisy data or presence of certain factors. Bayes classifier presents an approach for modelling probabilistic relationship b/w the attribute set and class variable.

$P(X=x)$ refers to probability that variable X will take on the value x.

Q. How we derive the bayes ~~problem~~ formula?

Ans $P(X,Y) = P(Y|X) \cdot P(X) = P(X|Y) \cdot P(Y)$

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$

Q. Consider a football game b/w two teams T0 and T1. Suppose T0 wins 65% of times $P(T0) = 0.65$ and T1 wins remaining matches $P(T1) = 0.35$ Among games won by T0 only 30% of them come from playing on T1's field on the other hand 75% of victories for T1 are obtained while playing at home. If T1 is to host next match. which team will most likely emerge as winner

Ans $P\left(\frac{T0}{H}\right) = 0.30$, $P\left(\frac{T1}{H}\right) = 0.75$

$P(T0) = 0.65$ $P(T1) = 0.35$

$\frac{1}{9\times10^{-2}}$

+ Classfriend +
Page No. Date

X → team hosting match Y → Winner of match

$P(Y=0) = 0.65$

$P(Y=1) = 0.35$

$P\left(\frac{X=1}{Y=0}\right) = 0.3$

$P(X=1/Y=1) = 0.75$

$P(Y=0/X=1) = \frac{P\left(\frac{X=1}{Y=0}\right)\cdot P(Y=0)}{P(X=1)}$

$= \frac{P(X=1/Y=1)\cdot P(Y=1)}{P(X=1, Y=1) + P(X=1, Y=0)}$

$= \frac{0.75 \cdot 0.35}{P(X=1/Y=1)\cdot P(Y=1) + P(X=0/Y=0)\, P(Y=0)}$

$= \frac{0.75 \cdot 0.35}{0.75 \cdot 0.35 + 0.3 \cdot 0.65}$

$\Rightarrow \frac{0.2625}{0.4575} = 0.57377$

$P(Y=0/X=1) = 1 - P(Y=1/X=1)$

Team 1 will win the match as $P(Y=1/X=1) > P(Y=0/X=1)$

* Using the Bayes Theorem for classification

. X → attribute set Y → class variable

If the class variable has non-deterministic relationship with attribute then we can treat X and Y as random variable and capture their relationship probabilistically using $P(Y/X)$ which is known as posterior probability for Y.

$P(X)$, $P(Y)$ → prior

During the training phase we need to learn the posterior probabilities $P(Y/X)$ for every combination of X & Y. based on the information gathered from the training data so that the test record X' can be classified by finding the class Y' that maximizes the posterior probability $P(Y'/X')$

$X = (\text{HomeOwner} = \text{No}, \text{MaritalStatus} = \text{Married}, \text{AnnualIncome} = \$120K)$

$P(\text{Yes}/X)$, $P(\text{No}/X)$

Here we'll be using Bayes theorem :-

$$P(Y/X) = \frac{P(X/Y) \cdot P(Y)}{P(X)} \qquad P(X) = \alpha$$

$P(Y = \text{yes}) = 0.3 = 3/10$

$P(Y = \text{no}) = 0.7 = 7/10$

Naive based classifier estimates the class conditional probability by assuming that the attributes are conditionally independent given the class label y.

$$P(X/Y = y) = \prod_{i=1}^{d} \left(P(x_i/Y = y) \right)$$

$$P(Y/X) = \frac{P(y) \cdot \prod_{i=1}^{d} P(x_i/Y = y)}{P(X)}$$

$P(X = \text{Homeowner/NO}/\text{Yes}) = \frac{3}{3} = 1$

$P(X = \text{Homeowner NO}/\text{NO}) = \frac{3}{7}$

$P(\text{Marital status Married}/\text{NO}) = \frac{4}{7}$

$= \frac{6}{7}$

* Estimating Conditional probability for Continuous attribute

1) We can discretise each continuous attribute and then replace the continuous attribute value with it's corresponding discrete interval.

$P(x_i/Y = y)$ is estimated by the fraction of training records belonging to class y that falls within interval X_i

If the no. of intervals is too large there would be very few training records in each interval and hence will not provide a reliable estimate

for $P(x_i/y)$. If no. of intervals is too small then some intervals may aggregate records from different classes.

2) A gaussian distribution is usually chosen to represent the class Conditional probability for continuous attributes.

$$P(X = x_i \mid Y = y_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{ij}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

$\sigma \rightarrow$ Standard deviation $\quad \mu \rightarrow$ mean

$$\bar{x} = \frac{125 + 100 + 70 + 120 + 60 + 220 + 75}{7} = \frac{770}{7} = 110$$

$$\sigma = \sqrt{\frac{225 + 100 + 1600 + 100 + 2500 + 12100 + 1225}{6}}$$

$$\sigma = 54.54 \qquad \sigma^2 = 2975$$

$$P(X = x_1 \mid Y = y_1) = \frac{1}{\sqrt{2\pi}\,(54.54)} e^{-\frac{(120-110)^2}{2 \times 2975}} = 0.0072$$

$$P(X/NO) = \frac{4}{7} \times \frac{4}{7} \times 0.0072 = 0.0024$$

$$P(Y = NO/X) = \frac{P(X/Y = NO) \cdot P(Y = NO)}{P(X)}$$

$$= \frac{0.0024 \times \frac{7}{10} \times 1}{\alpha}$$

$$P(Y = Yes/X) = 0$$

So X Classlabel = NO

* CLASS IMBALANCE PROBLEM

	Predicted	
	+	−
+	TP	FN
−	FP	TN

$$\text{Accuracy} = \frac{TP + TN}{\text{Total}}$$

$$\text{Error} = 1 - \text{Accuracy}$$

(True positive rate)

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

(γ)

$$TNR = \frac{TN}{TN + FP}$$

$$FNR = \frac{FN}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} = TPR$$

$$F_1 = \frac{2}{\frac{1}{\gamma} + \frac{1}{p}} \rightarrow \text{Harmonic mean} = \frac{2\gamma p}{p + \gamma}$$

$$\text{Weighted Accuracy} = \frac{W_1 TP + W_4 TN}{W_1 TP + W_2 FP + W_3 FN + W_4 TN}$$

$$F_\beta = \frac{(\beta^2 + 1) TP}{(\beta^2 + 1) TP + \beta^2 FP + FN}$$ → used to examine tradeoff b/w precision and recall

	W_1	W_2	W_3	W_4
Recall	1	0	1	0
Precision	1	0	0	0
F_β	$\beta^2 + 1$	β^2	1	0
Accuracy	1	1	1	1

CHAPTER - 6

ASSOCIATION ANALYSIS

In market basket transactions each row in the table corresponds to a transaction which contains a unique identifier transaction id and a set of items bought by a given customer. Retailers analyse the data to learn about the purchasing behaviour of the customers which can be used in marketing promotions to increase their profits and so on. It's also useful for discovering interesting relationships hidden in large datasets which can be represented in the form of association rules. 2 key issues must be taken care of when applying association analysis to market basket data

1) Discovering patterns from a large transaction dataset can be computationally expensive.
2) Some of the patterns may be spurious bcoz they happen simply by chance.

* BINARY REPRESENTATION

Market basket data can be represented in binary form where each item treated as binary variable whose value is 1 if the item is present in transc^n and 0 if not. thus an item is an asymetric attribute

Let $I = \{i_1, i_2, i_3 \ldots i_n\}$ be set of all the items in a market basket data and $T = \{t_1, t_2, \ldots t_m\}$ be the set of all transactions. Each transaction t_i contains a subset of items from itemset I.
If an itemset contains K items it's called K-itemset.

→ Transaction Width :- A TW is defined as no. of items present in a transaction. A transaction t_j is said to contain an itemset X if $X \subseteq t_j$

→ Support count :- refers to the no. of transactions that contain a particular itemset

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}|$$

→ Association Rule :- It's of the form $X \longrightarrow Y$ where X and Y are disjoint item sets $(X \cap Y = \phi)$. The strength of an association rule can be measured in :-

1) Support :- It determines how often a rule is applicable to a given dataset

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{N} \qquad c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Confidence determines how frequently the items we buy appear in the transaction that contains X.

{Milk, Bread} → {Eggs}

X → Y

TID	Milk	Bread	Eggs
1	1	0	0
2	0	1	1
3	1	1	1
4	1	1	1
5	1	1	0

Support = $\frac{3}{5} = 0.6$ Confidence = $\frac{2}{3} = 0.67$

A rule that has very low support may occur by chance also it may be uninteresting from a business perspective because it may not be profitable to promote the items that customers seldom buy together. Support is used to eliminate uninteresting rules.

Confidence measures the reliability of the inference made by rule. The higher the confidence more are the chances of Y being present in transactions that contain X.

* ASSOCIATION RULE DISCOVERY

Given a set of transactions T find all the rules having the support ≥ minsup and confidence ≥ Minconf where Minsup and Minconf are corresponding support and confidence thresholds.

Bruteforce approach for finding association rule is to compute the support and confidence for every possible rule. This approach is expensive because there are exponentially many rules that can be extracted from a dataset

$R = 3^d - 2^{d+1} + 1$

Total no. of rules (R) d (total no. of items)

$27 - 16 + 1$

⇒ 12

More than 80% of the rules are discarded after applying Minsup as 20% and Min Conf as 50%. To ~~avoid~~ reduce the no. of Computations it would be useful if we can prune the rules before having to compute their support and Confidence values. So a Common strategy adopted to decompose the problem into 2 major subtasks

1) Frequent itemset generation whose objective is to find all the frequent itemset that satisfy the Minsup threshold
2) Rule generation whose objective is to extract all the high Confidence strong rules from the frequent itemsets found in the previous step.

The 1st step is Computationally more expensive than 2nd.

* Frequent Itemset Generation

A dataset that Contains K items can generate $2^K - 1$. A brute force approach for finding the frequent itemsets is to determine the Support Count for every candidate item set in the lattice set. Such an approach can be very expensive because it requires $O(NMw)$ Comparisons ($N \rightarrow$ no. of transactions, $M \rightarrow$ total no of itemsets possible $(2^K - 1)$, $w \rightarrow$ max transactions width.

To reduce the computational Complexity 1) first way out is to reduce the no of Candidate itemsets it's an effective way to eliminate some of the Candidate itemsets without Counting their support values.

2) Reduce no. of Comparisons instead of matching each candidate itemset against each transaction we can reduce no. of Comparisons by using some advanced DS either to store Candidate itemsets or to Compress the dataset.

* Apriori Principle

→ If an itemset is frequent then all of its subsets must also be frequent.

If an itemset such as (a,b) is infrequent then all of its supersets must also be infrequent. It's based on a key property that the support for an itemset never exceeds the support of its subset. This property known as antimonotone property of the support measure.

e.g - If itemset (a,b) is infrequent then the entire subgraph Containing

Clearfriend
Page No. Date

Supersets of (a,b) in lattice structure can be pruned immediately.

Monotonicity Property:- Let I be a set of items and $J = 2^I$ (powerset of I) then a measure f is monotone if

$$\forall X, Y \in J : (X \subseteq Y) \rightarrow f(X) \leq f(Y)$$

Antimonotone Property:- $\rightarrow f(Y) \leq f(X)$

* Frequent itemset generation using Apriori algorithm

Candidate-1 itemsets (minsup = 3)

Item	Count	
Beer	3	
Bread	4	
Cola	2	x
Diapers	4	
Milk	4	
Eggs	1	x

Candidate-2 itemsets

Itemset	Count	
Beer, Bread	2	x
Beer, Diap	3	
Beer, Milk	2	x
Bread, Diap	3	
Bread, Milk	3	
Diap, Milk	3	

Candidate-3 itemsets

Itemset	Count	
{Beer, Diap, Bread}		x
{Bread, Milk, Diap}	3	
{Beer, Diap, Milk}		x
{Beer, Bread, Milk}		x

6 × 5 × 4

Beer, Bread
Beer, Diap
Beer, Milk
Bread, Diap
Bread, Milk
Diap, Milk

Using bruteforce = $^6C_1 + {}^6C_2 + {}^6C_3$

$= 6 + 15 + 20 = 41$

Using apriori = $^6C_1 + {}^4C_2 + {}^4C_3$ (support count for 1)

$= 6 + 6 + 4 = 16$ (13)

Let C_k denote the set of candidate-k itemsets and f_k denote set of frequent k-itemsets. The algorithm initially makes a single pass over the dataset to determine the support count of each item. Upon completion of this step the count of one items will be known. The algo. will iteratively generate new candidate-k itemsets using frequent k-1 itemsets found in the previous iteration. The algo. terminates when there are no new frequent itemsets generated.

The frequent itemset generation of Apriori algo. has 2 imp. characteristics

1) Candidate Generation:- At each iteration new candidate itemsets are generated from the frequent itemsets found in previous iteration

k_{max} → max size of itemset

2) Candidate Pruning :- In this support for each Candidate is counted and tested against the Minsup threshold. The total no. of iterations needed by the algo. is $k_{max} + 1$

Requirements for an Effective Candidate Generation Procedure

1) A candidate itemset is unnecessary if atleast 1 of it's subsets is infrequent (according to antimonotone property)
2) To ensure Completeness the set of Candidate itemsets must cover all the frequent itemsets.

$$\forall k : F_k \subseteq C_k$$

3) Generation of duplicate Candidates leads to wasted Computations and thus should be avoided for efficiency reasons.

Brute Force method Complexity

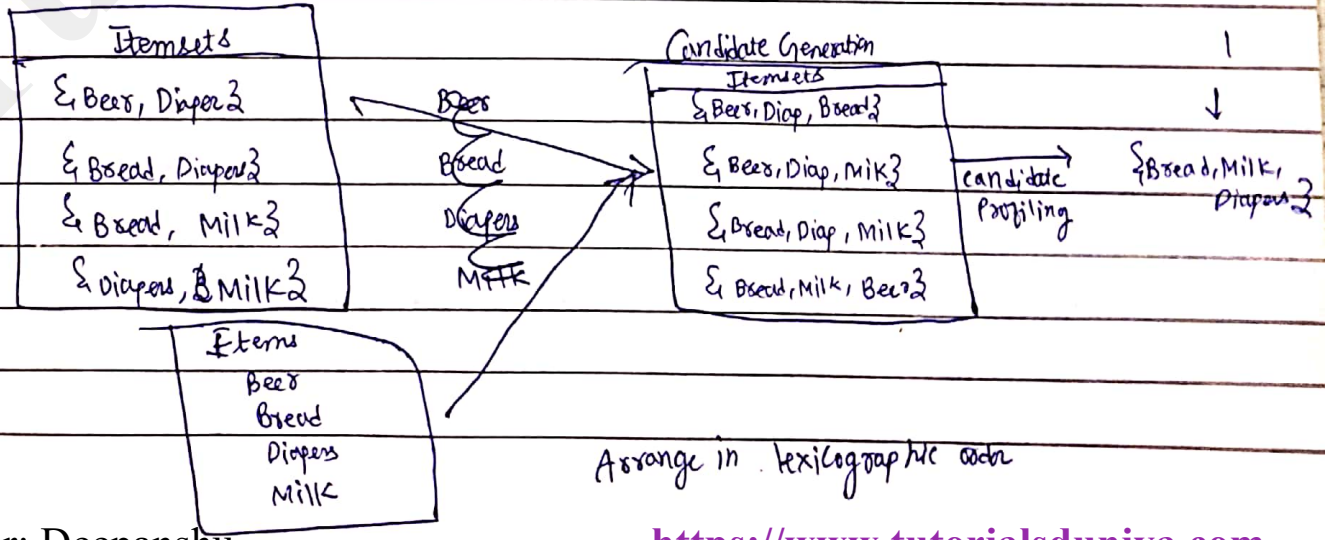
1) The no. of candidate itemsets generated at level k is $\binom{d}{k}$ where d → total no. of items
2) Candidate pruning requires O(k) time
3) Overall Complexity = $O\left(\sum_{k=1}^{d} k \cdot \binom{d}{k}\right) = O(d \cdot 2^{d-1})$

$F_{k-1} \times F_1$ method

It's an alernative method for candidate generation in which we combine frequent k-1 itemsets with other frequent items.

Complexity = $O\left(\sum_{k} k \cdot |F_{k-1}| \cdot |F_1|\right)$

This procedure is Complete beoz every frequent k itemset is composed of frequent k-1 and 1 itemset.



TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

Problem with this approach

(1) This approach doesn't prevent the same candidate itemset from being generated more than once. So the sol" to this is the items in each frequent itemset are kept sorted in their lexicographic order. Each frequent K-1 itemset is then combined with frequent items that are lexicographically larger than the items in the itemset.

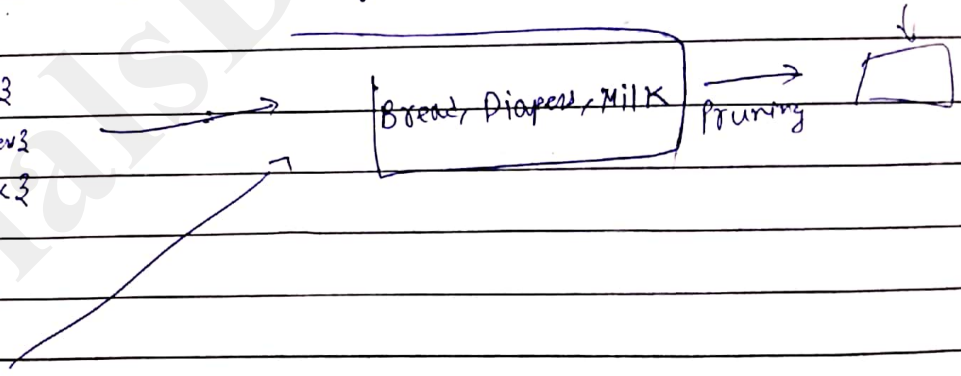
$F_{K-1} * F_{K-1}$ method

This ^ Candidate generation procedure merges a pair of frequent K-1 itemsets only if their first K-2 items are identical. A= {a1, a2, ... aK-1} B = {b1, b2, ... bK-1}

A= {a1 A and B be pair of frequent K-1 itemsets then A ≥ B will be merged if they satisfy following conditions

$a_j = b_i \ (i = 1, 2, \ldots K-2), \quad a_{K-1} \neq b_{K-1}$

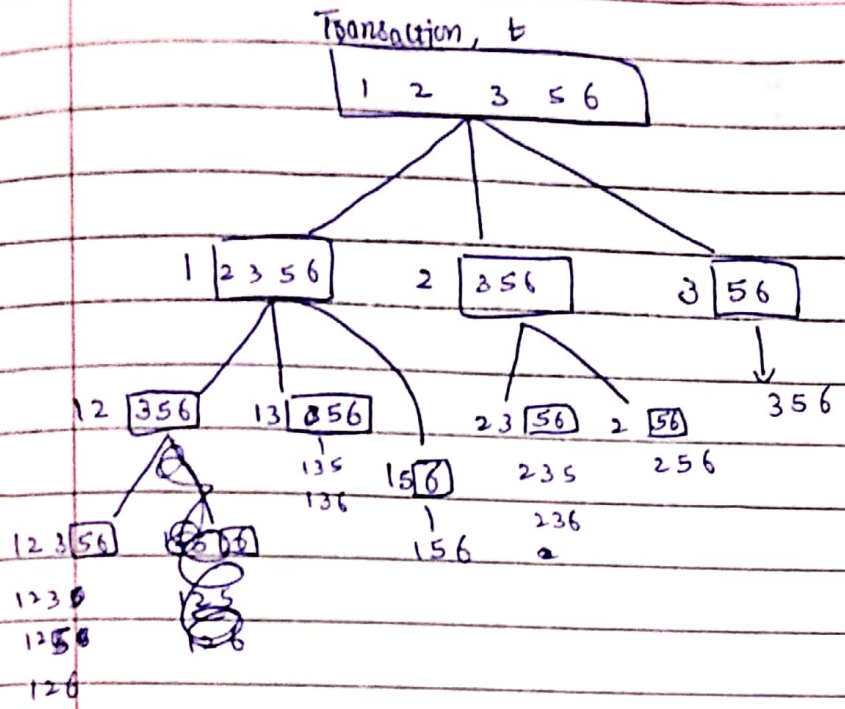
This method illustrates both the completeness of candidate generation procedure and advantages of using lexicographic ordering to prevent duplicate candidates. Also an additional candidate pruning step is required to ensure that remaining K-2 subsets of the candidate are frequent.



* Support Counting

One approach for doing this is to compare each transaction against every candidate itemset and to update the support counts of candidates contained in the transaction this approach is computationally expensive so we consider an alternative approach in this each itemset keeps its items in increasing lexicographic order and an itemset can be generated by specifying smallest item first followed by larger items.

$^nC_2 = n(n-1)$



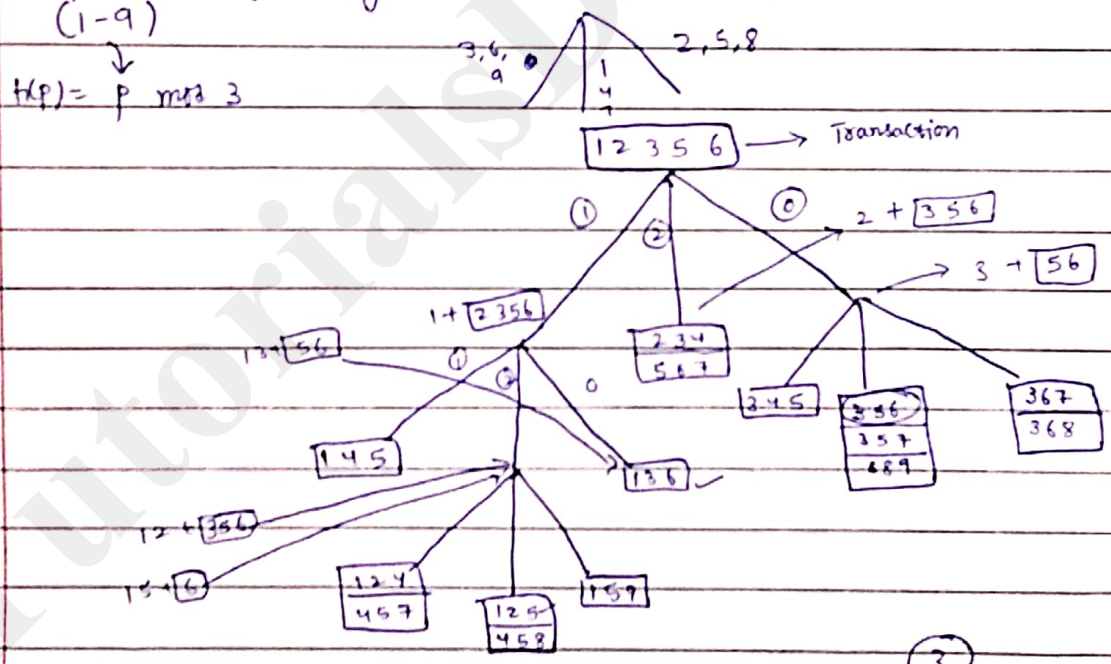
Here we have to determine whether each generated 3 itemset corresponds to an existing candidate itemset if it matches one of the candidates then support count of the corresponding candidate is incremented

* Support Counting using Hash Tree

(1-9)

↓

h(p) = p mod 3



3

3 → {356, 136, 125}

In the apriori algorithm candidate itemsets are partitioned into different buckets and stored in a hash tree during support counting itemsets contained in each transaction are also hashed into their appropriate buckets as a result each itemset in the transaction is matched only against candidate itemsets that belong to the same bucket.

Each internal node of the given has tree uses the following hash function

$h(p) = p \bmod 3$

Consider a transaction $T = \{1, 2, 3, 5, 6\}$ to update the support counts of the candidate itemsets the hash tree must be traversed in such a way that all the leaf nodes containing candidate 3 itemsets belonging to t must be visited at least once.

Item 1 is hashed to the left child of root node, Items $\{2, 5, 8\}$ are hashed to the middle while Items $3, 6, 9$ is hashed to the rightmost child.

This process continues until the leaf nodes of hash tree are reached.

The candidate itemsets stored at the visited leafnodes are compared against the transaction if the candidate is a subset of transaction it's support count is incremented

(CSNNA)

* The Computational Complexity of Apriori algorithm can be effected by following factors

1) Support threshold :- Lowering the support threshold results in more itemsets being declared as frequent thus affecting the computational complexity of algo.

2) No. of items :- As the no. of items increases more space will be needed to store the support count of items.

3) No. of transactions :- As the no. of transaction increases the apriori algo. makes repeated passes over the dataset.

4) Average transaction Width :- The max size of frequent itemsets tend to increase as the average transaction width increases. As a result more candidate itemsets must be examined during candidate generation and support counting. This will also increase the no. of hashed tree traversals performed during support counting.

Time Complexity

1) Generation of Frequent 1 itemset → $O(Nw)$ time

N → total no. of transactions
w → average transaction width

2) Candidate Generation → $\sum_{k=2}^{w} (k-2)|C_k| <$ cost of Merging $< \sum_{k=2}^{w} (k-2)|F_{k-1}|^2$

To generate Candidate k itemsets pairs of frequent k-1 itemsets are merged to determine whether they have at least k-2 items in common. Worst case scenario when we have to merge every pair of frequent k-1 itemsets

3) A hash tree is also constructed during the candidate generation to store candidate itemsets, so the cost for populating hash tree with max depth k

$$O\left(\sum_{k=2}^{w} k|C_k|\right)$$

4) The cost for looking up a candidate in hash tree is $O(k)$ and the candidate pruning step = $O\left(\sum_{k=2}^{w} k(k-2)|C_k|\right)$

5) Support Counting - Cost is $O\left(N \sum_{k} \binom{w}{k} \alpha_k\right)$

w → max transaction width α_k → cost for updating support count of a candidate k itemset in the hash tree

* RULE GENERATION

frequent k itemset rules = $2^k - 2$

$\phi \rightarrow$] $\times$

An association rule can be extracted by partitioning the itemset Y into 2 non-empty subsets X, Y-X such that $X \rightarrow Y-X$ satisfies the confidence threshold.

Let $X = \{1, 2, 3\}$ be a frequent itemset

$\{1,2\} \rightarrow \{3\}$ $\quad$ $\{1\} \rightarrow \{2,3\}$

$\{1,3\} \rightarrow \{2\}$ $\quad$ $\{2\} \rightarrow \{1,3\}$

$\{2,3\} \rightarrow \{1\}$ $\quad$ $\{3\} \rightarrow \{1,2\}$

Classified
Page No. Date

$\{1,2\} \rightarrow \{3\}$ = Confidence = $\frac{\sigma(\{1,2,3\})}{\sigma(\{1,2\})} = \frac{2}{3}$

Since the support count for both the itemsets were already found during frequent itemset generation there is no need to traverse to the entire dataset again.

Confidence doesn't have any monotone property but it holds the following theorem.

→ If a rule $X \rightarrow Y - X$ doesn't satisfy the confidence threshold then any rule $X' \rightarrow Y - X'$ where $X' \subseteq X$ must not satisfy the confidence threshold as well.

$X \rightarrow Y - X$ $\quad$ $s(X) = \frac{\sigma(Y)}{\sigma(X)}$

$X' \rightarrow Y - X'$ $\quad$ $s(X') = \frac{\sigma(Y)}{\sigma(X')}$

$\sigma(X') \geq \sigma(X)$

So $\boxed{s(X) \geq s(X')}$

Since $X' \subseteq X$ $\quad \sigma(X') \geq \sigma(X)$ will hold due to antimonotone property of support so $X' \rightarrow Y - X'$ can't have higher confidence than $X \rightarrow Y - X$.

Apriori algorithm uses a levelwise approach for generating association rules. Initially all the high confidence rules that have only 1 item in the rule consequent are extracted. Next candidate rules are generated by merging the consequent of 2 rules. If any node in the lattice has low confidence then according to the above theorem the entire subgraph spanned by the node can be pruned immediately.

Suppose the confidence for $\{b,c\} \rightarrow \{a\}$ is low. Then all rules containing a at right are pruned

$\{a\ b\}$
$\{a\ c\}$
$\{a\ b\ c\}$

The only difference in rule generation from candidate generation is that we don't make additional passes over dataset to compute the confidence of the candidate rules instead we use the support counts computed during frequent itemset generation.

Chapter-8

CLUSTER ANALYSIS

It divides data into groups or clusters that are meaningful and useful. We do clustering for 2 (BICB) purposes :-

1) Understanding - a) Biology b) Info. Retrieval c) Climate d) Business
2) Utilities - a) Summarization b) Compression c) Effectively finding nearest neighbours

Instead of applying the algo. to entire dataset it can be applied only to cluster prototypes if the objects are relatively close to the prototype of their clusters then we can use the prototype to reduce the no. of distance computations. Also the nearness of 2 clusters can be measured by finding the distance b/w their prototypes.

* Clustering :- The diff. ways to group a set of objects into a set of clusters. The greater the similarity within a group and greater the difference b/w the groups the better the clustering is.

* DIFFERENT TYPES OF CLUSTERING (From book)

1) Hierarchical vs Partitional
2) Exclusive vs overlapping vs Fuzzy
3) Complete vs Partial → in case of noise or outliers

In Fuzzy clustering every object belongs to every cluster with a membership weight that is b/w 0 and 1. There's an additional constraint that sum of the weights for each object must equal 1.

* DIFFERENT TYPES OF CLUSTERS (WPGDS)

1) Well separated → well separated boundary b/w them.
2) Prototype based → the data objects will be closest to prototype of their clusters as compared to the prototype of any other cluster. also known as centre based
3) Graph based → All the points in a cluster will be closer to each other as compared to points of any other cluster.
4) Density based → the clusters in which density at center is max and decreases as we move outwards
5) Shared property (Conceptual clusters) → All objects in a cluster have same property

Page No. Date

* K-means Clustering Algorithm (K-medoid)

(in case of centroid)

It's a prototype based clustering technique which defines a prototype in terms of the centroid (the mean of group of points) it's applied to the objects in a cont n-D space.

K medoid defines a prototype which is the most representative point for group of points while a centroid may or may't correspond to an actual data point a medoid must be an actual data point.

Basic K-Means Algorithm

We first choose K initial centroids where K is a user specified parameter (no. of clusters) Each point is then assigned to the closest centroid and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update the steps until no point changes the clusters or until the centroids remain the same.

Diagram from book

→ Assigning points to the closest centroid

To assign a point to the closest centroid Euclidian distance is most often used for the data points in Euclidean space the goal of clustering is expressed by an objective function that depends on the proximities of points to one another or to the cluster centroids for e.g- SSE (sum of squared Error) we need to minimize the square distance of each point to it's closest centroid we will prefer the set of clusters with the smallest SSE meaning that prototypes of this clustering are a better representation of the points in their cluster.

$x \rightarrow$ a data object $C_i \rightarrow$ ith cluster $c_i \rightarrow$ centroid of cluster C_i

$m_i \rightarrow$ no. of objects in ith cluster

$m \rightarrow$ total no. of objects in dataset $K \rightarrow$ no. of clusters

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2$$

$$c_i = \frac{1}{m} \sum_{x \in C_i} x$$

Page No. Date

→ Suppose we have 3,3D point the coordinates are (1,2,3), (3,4,5), (1,2,4)

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

If your dataset is document dataset similarity measure is

Total cohesion = $\sum_{i=1}^{K} \sum_{x \in C_i} \text{Cosine}(x, c_i)$

* Choosing initial Centroids

Choosing proper I.c. is the key step of K-Means algo a common approach is to choose the initial centroids randomly but the resulting clusters are often poor. So one solution is to perform multiple runs each with a diff. set of randomly chosen initial centroids and then select the set of clusters with a minimum SSE An optimal clustering will be obtained as long as 2 initial centroids fall anywhere in a pair of clusters since the centroids will redistribute themselves 1 to each cluster. other effective techniques employed for initialization are:-

1) Take a sample of points and cluster them using hierarchical clustering technique the centroids from the clusters formed in this technique are used as initial centroids it works well only if the sample size is small or no. of clusters are small.

2) We start with a centroid of all the points then for each successive initial centroid we select the point that's farthest away from any of the initial centroids already selected Problem a) It can select outliers rather than the points in dense regions

b) It's expensive to compute the farthest point from the current set of initial centroids so the solution to these problems is

↳ Instead of applying this approach to all data points it's applied only to sample of points. Since outliers are rare they usually won't be selected as a part of random sample and also the points from very dense region are likely to be included in this sample.

Page No. Date

KNN (K- Nearest Neighbour)

```
K=13 ; #(sqrt(n))
normalize <- function(x) {
                return((x- mean(x))/sd(x))
}
library(class)          // class library for the knn
m1 <- knn(iris_train, iris_test, iris_train[,5], K=13)
Confusion Matrix(as.factor(iris_test[,5], m1)
```

→ * **Time and Space Complexity**

Space = $O((m+k)*n)$

Time Complexity = $O(I*K*m*n)$

I → no. of iterations
K → no. of centroids
m → no. of data objects
n → no. of attributes

1

→* **Handling Empty Clusters** :- If no points are allocated to a cluster during an assignment step then a strategy is needed to choose a replacement centroid

1) Eliminate the point that currently contributes most to the total squared error.
2) To choose a the replacement centroid from the cluster that has the highest SSE.

2

→ **Handling outliers** :- When the outliers are present the SSE will be higher so it is useful to discover outliers and eliminate them beforehand.

1) We can keep track of the SSE contributed by each point and eliminate the points with high contributions especially over multiple runs.
2) We may want to eliminate small clusters since they frequently represent the groups of outliers.

3 →* **Reducing the SSE with postprocessing**

1) one way to reduce the SSE is to find more clusters the strategy is to minimize the SSE contributed by individual of clusters since the total SSE is the sum of the SSE contributed by each cluster. We can change the total SSE by either splitting or merging the clusters.

Two strategies to decrease the total SSE by increasing the no. of clusters :-

a) Split a Cluster the Cluster with the largest SSE is usually chosen for splitting

b) Introduce a new cluster Centroid - 1) A point farthest from any cluster Center is chosen

2) We can choose randomly from all points or from the points with highest SSE

Two Strategies that decrease the no. of clusters while trying to minimize the increase in total SSE

1) Disperse a cluster :- We can remove the Centroid that corresponds to the cluster and reassign the points to other clusters.

2) Merge 2 clusters :- We can merge the 2 clusters that result in the smallest increase in total SSE. The techniques used for this is centroid method and Ward's Method.

3

4 → Updating the Centroids Incrementally

Instead of updating the cluster centroids after all the points have been assigned to a cluster the centroids can be updated incrementally after each assignment of a point to a cluster.

Advantages :-

a) This strategy guarantees that empty clusters are not produced because if a cluster ever has only one point that point will be reassigned to the same cluster.

b) Suppose we are given an arbitrary objective function to measure the goodness of the set of clusters when we process an individual point we can compute the value of objective function for each possible cluster assignment and then choose the one that optimizes the objective.

Disadvantages :-

a) The clusters produced may depend on the order in which the points are processed.

b) Also incremental updates are slightly more expensive.

Strengths and Weaknesses of K-means Algorithm

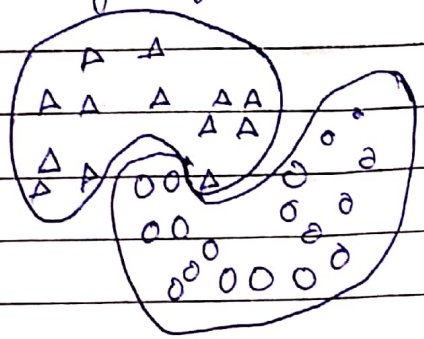
Strength :-

1) It's simple and can be used for a wide variety of data types

2) It's quite efficient

Weaknesses :-

1) It's not suitable for all types of data
2) It can't handle non globular clusters or clusters of different sizes and densities
3) It has trouble clustering the data that contain outliers.
4) It's generally restricted to data for which we can find a center point.



Non-globular clusters

Q) Given the following points 2, 4, 10, 12, 3, 20, 30, 11, 25 Given K = 3 and initial means $\mu_1 = 2$, $\mu_2 = 4$, $\mu_3 = 6$. Show the clusters obtained and the new means after each iteration using K Means algorithm.

Ans 1st iteration

d(2, μ1)
d(2, μ2)
d(2, μ3)

2 closest to μ_1 so it's assigned to cluster 1.
4 - - μ_2 - - - - 2.
10 - - - μ_3 - - - - - 3.
12 - - - - 3.
3 →

at the end of 1st iteration

$C1 = (2, 3)$ $= (2.5) = \mu_1$
$C2 = (4)$ $4 = \mu_2$
$C3 = (10, 12, 20, 11, 25, 30)$ $\mu_3 = 18$

In 2nd iteration

$C1 = (2, 3)$ $\mu_1 = 2.5$
$C2 = (4, 10, 11)$ $\mu_2 = \frac{25}{3} = 8.33$
$C3 = (12, 20, 25, 30)$ $\mu_3 = 21.75$

In 3rd iteration

$C_1 = (2, 3, 4)$ $\mu_1 = 3$
$C_2 = (4, 10, 11, 12)$ $\mu_2 = 11$
$C_3 = (20, 25, 30)$ $\mu_3 = 25$

10 marks

In 4th iteration

$c_1 = (2, 3, 4)$ $\mu_1 = 3$

$c_2 = (10, 11, 12)$ $\mu_2 = 11$

$c_3 = (20, 25, 30)$ $\mu_3 = 25$

4th iteration clusters same as 3rd so final clusters are

$C1 = (2, 3, 4)$ $C2 = (10, 11, 12)$ $C3 = (20, 25, 30)$

* ~~ENUMERATIVE~~ HIERARCHICAL ~~CLUSTERING~~

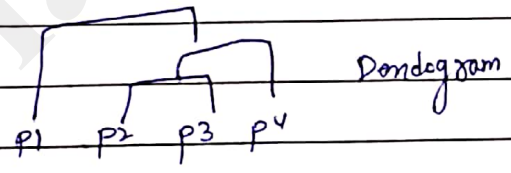
* AGGLOMERATIVE HIERARCHICAL CLUSTERING

There are two basic approaches for generating a hierarchical clustering:-

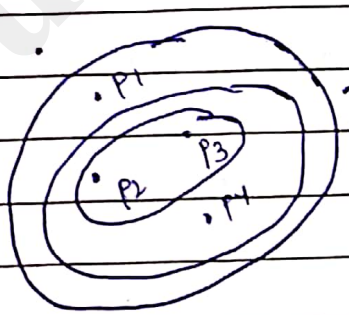
1) Agglomerative :- In this we start with the points as individual clusters and at each step merge the closest pair of clusters until one cluster remains.

2) Divigive:- In this we start with one all inclusive cluster and at each split a cluster until only singleton clusters of individual points remain

A hierarchical clustering is often represented graphically using a tree like diagram called dendogram which displays both the cluster-subcluster relationships and the order in which the clusters were merged.



Dendogram



Nested Cluster diagram

* Agglomerative Algo

1) Compute proximity matrix. $O(m^2)$
2) repeat
3) Merge the closest 2 clusters $O(m^2)$ — in sorted $O(m \log m)$ — $O(m^3)$
4) update the proximity matrix to reflect the proximity b/w new cluster and original $O(m)$
5) until only one cluster remains

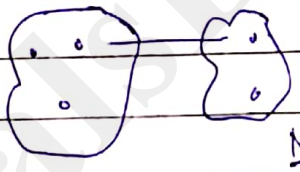
Time = $O(m^2) + O(m^3) = O(m^3)$

If this the distance from each cluster to all other clusters are stored as a sorted list it's possible to reduce the cost of finding 2 closet clusters.

So here Time = $O(m^2) + O(m^2 \log m) = O(m^2 \log m)$

Defining Proximity b/w the Clusters → the key operation is the computation of proximity b/w 2 clusters many agglomerative hierarchical clustering tech. such as min, max and group average are used.

MIN defines the cluster proximity as the proximity b/w the closet 2 points that are in different clusters.



MIN

MAX takes the proximity b/w the farthest 2 points in different clusters

MAX

GROUP AVERAGE define cluster proximity to be the average pairwise proximities of all pairs of points from different clusters. When using centroids the cluster proximity is commonly defined as proximity b/w cluster centroids

MAX → complete link

0.0324

	x	y
P1	0.4	0.63
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

	P1	P2	P3	P4	P5	P6
P1	0.0	0.24	0.22	0.37	0.34	0.23
P2	0.24	0.0	0.15	0.2	0.14	0.25
P3	0.22	0.15	0.0	0.15	0.28	0.11
P4	0.37	0.2	0.15	0.0	0.29	0.22
P5	0.34	0.14	0.28	0.29	0.0	0.39
P6	0.23	0.25	0.11	0.22	0.39	0.0

Single link (MIN) is good at handling non-elliptical shapes but is sensitive to noise and outliers. The height at which 2 clusters are merged in the dendogram reflects the distance of the 2 clusters (3,6), (2,5)

dist({3,6},1) = min(dist(3,1), dist(6,1))
= min(0.22, 0.23)
= 0.22

dist({3,6},2) = 0.15
dist({3,6},4) = 0.15 dist({3,6},5) = 0.28

{3,6,2}
dist({3,6,2},1) = 0.22 Next cluster {2,5}
dist({3,6,2},4) = 0.15 dist(2,5) = 0.14
dist({3,6,2},

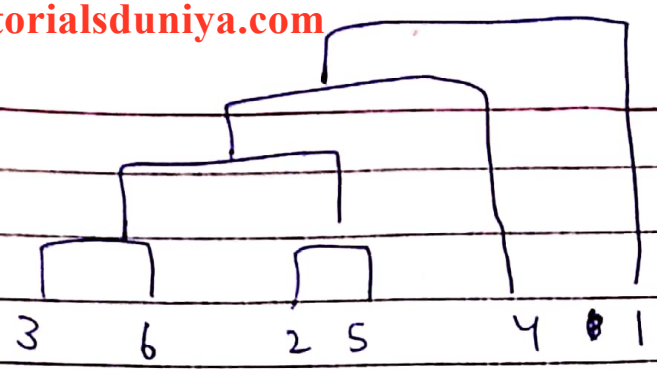
dist({2,5},1) = 0.24
dist({2,5},{3,6}) = 0.15
dist({2,5},4) = 0.2

Next cluster {(3,6),(2,5)}

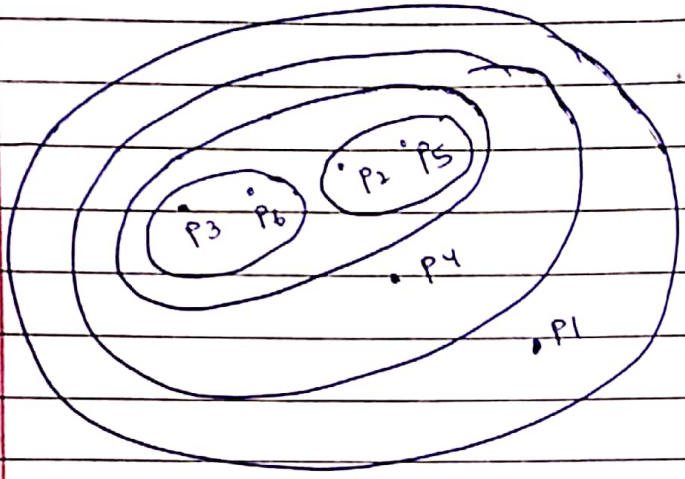
dist({(3,6),(2,5)},1) = 0.22
dist({(3,6),(2,5)},4) = 0.15

Next Cluster = {(3,6),(2,5),4}

and them {(3,6),(2,5),4,1}



dendogram



Bayesian

```
install.packages("e1071")
library(e1071)
model <- naiveBayes(Species ~ ., data = iris_train)
pred <- predict(model, iris_test)
View(pred)
confusionMatrix(pred, iris_test$Species)
```

K-Means

```
iris1 <- iris[, 1:4]
library(cluster)
results <- kmeans(iris1, 3)
results
table(
```

(3,4), (

Complete Link (MAX)

① {3,6} ② {2,5}

③ dist({3,6}, 1) = max(dist(3,1), dist(6,1)) = max(0.22, 0.23)
= 0.23

dist({3,6}, {2,5}) = max(0.15, 0.28, 0.25, 0.39) = 0.39

dist({3,6}, 4) = max(0.15, 0.22) = 0.22

③ {4, {3,6}, 4}

{{3,6}{4}, {2,5}} = 0.39

{{3,6}, {4}, {1}} = 0.37

~~{{3,6}, 4}~~ {{2,5}, 1} = 0.34

④ {{2,5}, 1}

⑤ {{3,6}, 4, {2,5}, 1}

(LAM)

* KEY ISSUES IN HIERARCHICAL CLUSTERING

1) Lack of a global objective function :- Agglomerative H.C. can't be viewed as globally optimising an objective function. It uses various criteria to decide locally at each step which clusters should be merged. These approaches don't have problem in choosing the initial points.

2) Ability to handle different cluster sizes :- one aspect of AHC is how to treat the relative sizes of the pairs of clusters that are merged there are 2 approaches a) weighted - which treats all the clusters equally i.e. treating the clusters of unequal size equally gives different weights to the points in different clusters

b) Unweighted - which takes the no. of points in each clusters into account i.e. it gives points in the different clusters the same weight.

3) Merging decisions are final :- AHC algorithms tend to make good local decisions about combining the 2 clusters since they use the info about the pairwise similarity of all points once a decision is made to merge 2 clusters it can't be undone at a later time which prevents it from becoming a global

optimz" criteria some techs attempt to overcome the limitation by using a partitional clustering technique such as K Means to create many small clusters and then perform HC using these small clusters as starting point.

Strengths (Advantages)

1) These algos are generally used as they form the clusters using a hierarchy and it's considered a ~~better~~ very good clustering approach.
2) These algos can produce better quality clusters

Disadvantages

1) HC algos are expensive in terms of their computational and storage requirement
2) The fact that all merges are final can be troublesome for noisy high dimensional data.

DBSCAN CLUSTERING ALGO (Density Based Scan)

It locates the regions of high density that are separated from one another by regions of low density. DBSCAN is a density based algo and is based on centre based approach in this approach density is estimated for a particular point in the dataset by counting the no. of points within a specified radius EPS of that point. this will also include the point itself. the method is simple to implement but the density of any point will depend on the specified radius. The point can be classified as :-

a) Core point :- It's present at interior of a dense region. A point called Core point if the no. of points within a given neighbourhood specified by the distance parameter EPS is exceeds a certain threshold min points (It's also a user specified param).

b) Border point :- It's located on edge of a dense region. It's not a Core point but falls within the neighbours of a Core point or Core points.

c) Noise point :- is present in a sparsely occupied region. A noise point is any point that's neither a Core point nor border point.

Any 2 Core points that are close enough within distance EPS of one another are put in the same cluster.

Any border point that is in the neighbourhood of a Core point is put in the

Same Cluster as core point.
Noise points are discarded.
Algo from book

* Time Complexity = $O(m \times$ time to find pts in Eps neighbourhood$)$
  worst case = $O(m^2)$

To reduce the time complexity we can use a special type of data structure called kd trees that allow efficient retrieval of points within a given distance of a specified point.

kd trees — K-dimensional tree Just like binary search tree where data in each node is a K-dimensional point in space. It's used for organizing points in K-dimensional space.

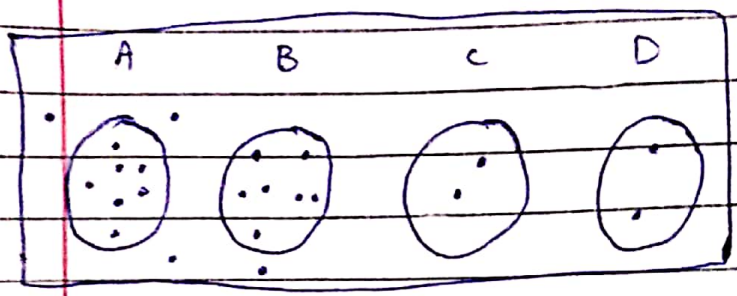
Time complexity now = $O(m \log m)$

* Space requirement = $O(m)$ because we only the points, the cluster label and identification of each point as core, border or noise.

For the points that are not in the cluster the K-distance will be large.
K-distance is distance b/w current point and kth point.
If we compute the K distance for all the points (data) for some k sort them in increasing order and then plot the sorted value we'll see a sharp change at a value of k distance that corresponds to a suitable value of EPS.
If we select this distance as EPS parameter and take the value of k as min points parameter then points for which the K distance is less than EPS will be labelled as core points while others are labelled as noise or border points.

& and Minpoints are same

* CLUSTERS OF VARYING DENSITY



The noise around a pair of denser clusters A and B has the same density as the clusters C and D if Eps threshold is low enough that DBSCAN finds C and D as clusters then the noise points surrounding A and B will also become part of a cluster. If Eps threshold is high enough such that the noise points surrounding A and B are marked as noise then points in clusters C and D will also be treated as noise

Advantages of DBSCAN

1) It's relatively resisting to noise and can handle the clusters of different shapes and sizes.
2) It can find many clusters that couldn't be found using KMeans.

DisadVantages :-

1) DBSCAN algo doesn't work when clusters have varying densities.
2) It's not suitable for high dimensional data bcoz it's difficult to define the density for such points.
3) It's quite expensive as the computation of nearest neighbours requires computing all the pairwise proximities.

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well