Data Mining & Warehousing

Mining: Process of extracting something you are interested in from a collection

Warehousing: How you organise your data so that data mining can better utilise that organized data in an efficient manner

After 1990s, we can better share & utilise the data

Stream data → real time data (Eg- stock market)

* DBMS vs Data Mining

In data mining some predictions are involved.

3, 15, 3, 19, 27 → marks in Course A

Using DBMS we can write SQL queries for performance, total marks etc. but not predict his ~~perfo~~ performance in another course next sem. But we can do that using Data Mining

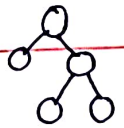
"interesting" (slide 7)

That satisfies your requirement

5, 15, 25, 10, 20 → data

5, 10, 15, 20, 25 marks in ↑ order → Inform"

↓ create some model

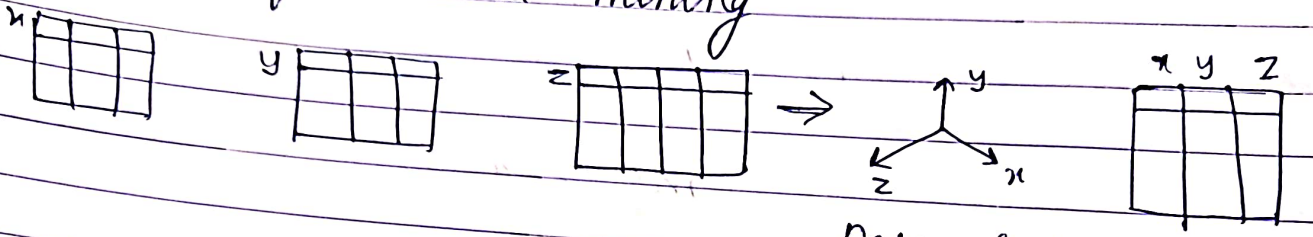


Knowledge → using accumulated inform"

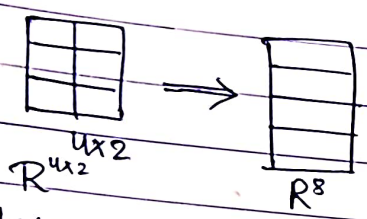
* KDD Process

Data available on diff servers may have a heterog. org". So we need to integrate that into homog. org" (data integ")

In data warehouse data is stored in a format suitable for data mining



Data integration



2D data 1D data

Web Mining → extracting data from web

Cube → Multidimensional data organisation (Homogeneous org", stored in warehouse)

Present" of result: Better way to represent the data so some conclusion can be drawn

I. * Preprocessing: Activity performed before execution of your algorithms, performed on i/p data

Lab

1. Preprocessing < cleaning / missing value

2. Feature selection

3. Algo

4. Result → Acc
→ Preci
→ Recall
→ F1 score

'arff (attribute relation file format)

3 section
1. @relation
2. @attribute
3. @ data

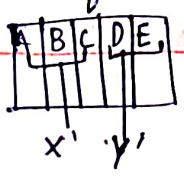
Visualization → ROC
→ AUC

—x—

Data is available in diff. dbs in heterog. form. So we need to transform it into a standard form, as per the requirement. Depends on algo, diff types of Data Integration is there

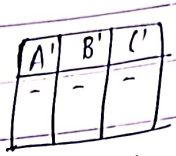
Normalization: 1-100 marks range ⟶ 1-10 range req. by algorithm. How you represent data values in a different range, which is required by the algorithm

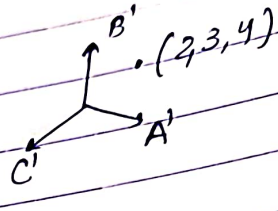
Feature Selection: Attribute in a table → Feature/variable/dimension (Linear algebra). ↑ features ↑ complexity. Redundancy in data is common. Ways to ↓ size of data:
→ Remove redundant / irrelevant features (Feature selec")
→ Feature extraction: Using initial features, design some new features more relevant for your application

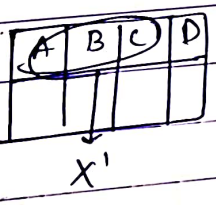


→ Issue:
F.S has the probability of information loss, unless you remove only irrelevant/redundant ones. But in reduction, "inform" loss should not be there → performance of model affected

Dimensionality reduction:
Dimension → Attribute
Due to complexity of algo., we need to ↓ dimension of dataset







3D "inform" transformed into 1D → D.R ← w/o into loss
After feature selection → transform" must be w/o "inform" loss

Mechanisms
1) PCA
2) LDL

II. * Data Mining
① Pattern discovery: Based on patterns we take business decisions (Eg- customer purchasing behaviour)

② Association & Correlation:
Transactional dataset

Transac"	item1	item2	item3
T_1	Bread	Milk	Soap
T_2	Bread	Milk	
T_3	Soap	Bread	
T_4	Bread	Milk	

Analysis → the customers buys bread along with milk, and in regular intervals.
∴ Bread & Milk are associated to each other

⇓

Transac"	Bread	Milk	Soap
T_1	1	1	1
T_2	1	1	0
T_3	1	0	1
T_4	1	1	0

75% of time he buys bread + milk
business prepare their inventories in diff. places based on this data.

Usual dataset → considering every purchased item as an attribute

Correlation - Milk → Bread

$-1 \leq 0 \leq 1$

↑ no corr. ↑ +ve correlation

Mechanism: PCC (Pearson correlation coeffient)

Numerical quantification of associa" b/w 2 attributes/variables

③ Classification: Classify samples in different categories

2 classes → Binary classification

>2 " → Multiclass "

T	B	M	S	class
				+ — valid transac"
				+
				− — invalid
				+

} Binary

④ Clustering:

Classific^n works on labelled data and
Clustering " " unlabelled

labelled → class attribute present in data
class label not known → clustering

⇒ Classific^n algos → Supervised machine learning algorithm (called)

You learn in presence of supervisor (check & rectify)
Learning w/ labelled dataset

⇒ Clustering → Unsupervised ML algo

⇒ Hybrid → Semi supervised ML algo

⑤ Outlier analysis:

T	B	M	height	class
			3.00	
			5.3	
			5.2	
			6.0	
			4.7	
			⋮	
			6.5	
			5.00	

Height permissable limit 0-7
on analysis you find 99% students have height b/w 0-6
Then 6.5 is treated as an outlier

Outlier: related to your data but are rare samples

Outlier Analysis is useful in Credit Card system.

A	B	C	D	class
-	-	-	-	valid
-	-	-	-	valid
-	-	-	-	valid
-	-	-	-	valid
-	-	-	-	invalid

Eg1 - 1-2% customers X paying bill on time → invalid customers

relevant customer ✓ (satisfied cond" for becoming bank's customer) but X paid bill on time → rare sample

*Outlier Analysis: How you identify (criteria) the rare samples (Identific" of rare by formulating some criteria)

Make business schemes → EMI offer, offers to pay bill on time

Eg 2- Identifying unsuccessful packet transmissions, what can be the causes etc.

III. Post processing:

Algo executed but result is unordered → need to represent in proper way

(1) Pattern evaluation:

Transac"	B	M	S
T_1	1	1	1
T_2	1	1	0
T_3	1	0	0
T_4	1	1	1

f	
3	Bread, Milk
2	Bread, Milk, Soap

} patterns

All patterns may not be useful

Unuseful patterns → space complexity ↑

P. Evaluation: Mechanism with which only useful patterns are shown

Eg- pattern useful if frequency > 75% → threshold/ criteria set to select relevant and useful patterns

② Pattern Selec"

③ " Interpretat"

④ " Visualization

* Dimensions of Data Mining

① Data to be mined: Many sources/ types

stream → video data
(pos"/coordinates) spatiotemporal → temperature, GPS
time-series → stock price
sequence → 1st computer then speakers then pen drive
→ DNA structure
graph → social networking
→ lattice structure

Diff. DBs as per type of data

② Knowledge to be mined:

Characterization: Summariz" of your data related to the target class. Eg- invalid in credit card system

Eg- Those who got A+ in Data Mining, did they also get A+ in DBMS → good students for data science

Discrimination: Using the summarisation, how we discriminate the class of with other classes

Descriptive vs Predictive Data Mining:

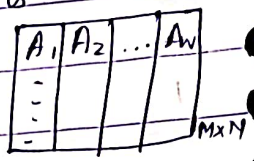
1) Descriptive: With your data your are formulating the descriptions (Eg- assoc" / correl" / existance of patterns) What method we use to describe our data (Eg- Pattern anal., cluster ana, asso, correl")

2) Predictive: On the basis of certain features we are predicting some other features. Eg- if one class label for a row is missing. Using some model we on the data we can predict the unknown class. So all ML algos → predictive)

Ch-2: Getting to know your data

Matrix : Arrangement of data in rows or columns

Data matrix : Data in numeric form in matrix



Document matrix

TF-IDF : Term Frequency - Inverse Document Frequency

Data — Structured
unstructured
semi structured

Algos we study are mainly based on structured data

* Important Characteristics of Structured Data

1) Dimensionality : No. of attributes in the database. Issue → How to tackle ↑ dimen.

Curse of dimen. problem: If you ↑ dimension of your database, complexity of your algo. ↑ → performance ↓
memory ↑ computation ↑ need for structure ↑

2) Sparsity :

Many terms can have freq. 1 or 0

T_1	T_2	T_3	T_4	T_5	T_6
1	0	0	0	0	0
0	✗	✗	✗	✗	1
0	0	0	0	0	0

In sparse data many useful analysis cannot be done. Eg- checking if 2 docu. are similar, here you will think its same since most words have 0 frequency

Sol" → remove ~~tuples~~ terms having frequency 0, then only you can get good result for similarity/dissimilarity

3) Resolution : Size of data
MxN (No. of tuples)

For good learning, sufficient samples should be there.

↓ samples → underfit

↑ " → overfit (also ↑ attributes)

Issue → optimum no. of samples

4) Distribution: Values corresponding to attributes

Eg. Gaussian dist" func"

Tendency of data, in which direc". its tilts to

Data Objects:

In database all tuples are data objects

Attributes:

qualitative (categorical)	Nominal (=, ≠)	
	Ordinal (<, >)	
quantitative (Numeric)	Interval (+, −)	
	Ratio (*, /)	

Binary → 2 values

Gender → Binary

Cancer + only few ppl have issue of cancer

Interval:
X True zero point
Multiplication factor exactly the same, reference pt, well defined → zero point

Discrete: Shirt size

Continuous: real values within in certain range

* Statistical

1) Mean (sample vs population)

High income (easily to identify) 100 → 10

med. inc. 300 500 → 10

low inc. 600 400 → 10 samples

+ve skewed data (mean biased towards +ve dirn)
considering pop^n X

Go w/ the sample not the pop^n

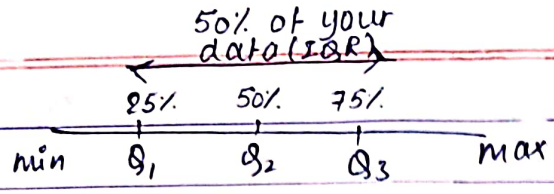
Tendency biased towards the outlier values.

10 12 15 20 21 (180) outlier

freq.

-ve skewed data (mean biased towards -ve dirn)

Sec how to check +ve/-ve skew?



Q_2 → median

IQR → denote behaviour of 50% of your data
Outlier = 1.5×IQR (standard way to identify outliers)

* Similarity & Dissimilarity (Proximity)
In your database if you want to know the similarity/dissim. of data objects in numerical form

	hair	weight	gender
obj1	black	32.5	1
obj2	red	16.2	0

Data Matrix: p attributes n data objects
Resolution → n×p

d(2,1) dissimilarity of 2nd object w/ 1st object
Sim(2,1) = 1 − d(2,1)

① Nominal attributes:

ML: $d(i,j) = \frac{p-m}{p}$ p → no. of attributes
m → matches across those attributes.

(Nominal) Employee Id
1
2
3
4
5

(Nominal) Hard. of material
V.H
S
V.S
S
H
V.S

similar: 1, 2, 4, 5 | 3

$d = \frac{2-0}{2} = 1$

d lies b/w [0,1]
d=1 → highly dissimilar

$d(6,3) = \frac{2-2}{2} = 0$

Symmetric binary attribute (Gender)
Asymm. (Cancer detcc")

SMC (Simple Matching Coefficient) $sim(i,j) = 1 - d(i,j)$ ((0,0) or (1,1) matching)

	A	B	C	D	E
1	1	0	0	0	1
2	0	0	1	0	0

1 0 → r frequency of 10

	1	0
1	q	r
0	s	t

× × → useless for dissimilarity

Symmetric binary attribute
(Eg-students giving exam in similar way→ 1,1 0,0)

Symmetric binary variables ⇒ $d(i,j) = \frac{r+s}{q+r+s+t}$ (1, 0 ones)

Asymmetric " " $d(i,j) = \frac{r+s}{q+r+s}$ (0,0 removed→negative result→x help in dissimilarity)

Similarity in asym. " " $sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$ (1,1)
(1,1) (1,0) (0,1) out of those doing some qns right
(0,0)→ wrong answer

Normalizing:

1) Z-score normalisation

X	X'
1	0
2	
3	
⋮	⋮
100	1

$$z = \frac{x - \mu}{\sigma}$$

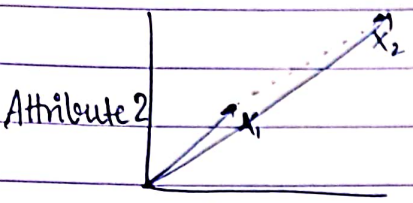
X	X'
1	-1
2	-0.5
3	0
4	0.5
5	1

$1 \to \frac{1-3}{2} = -1$

$2 \to \frac{2-3}{2} = \frac{-1}{2} = -0.5$

2) Absolute Z-score normalisation (Mean abs. deviation)

② Numeric attributes :



2 dimensions → cartesian system

point	att. 1	att. 2
x_1	1	2
x_2	3	5
x_3	2	0
x_4	4	5

point in vector space

vectors $4\hat{i} + 5\hat{j}$

For 2D → **Euclidean distance** $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

	x_1	x_2	x_3	x_4
x_1	0			
x_2	3.61	0		
x_3	5.1	5.1	0	
x_4	4.24	1	5.39	0

Minkowski distance : absolute diff. b/w 2 data obj. compared across p data objects

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

h = 1 Manhattan distance

h = 2 Euclidean ,,

h = ∞ Supremum ,,

↳ $\max_f |x_{if} - x_{if}|$ (max distance for a particular distance f)

$d(x_1, x_2)$

	①	②
x_1	1	2
x_2	3	5
	↓	↓
	2	③ ← max → supremum dist = 3

③ Ordinal attributes → Discrete

Qualitative, Categorical attribute

Order is imp.

(0,1) map T shirt size

$\frac{1-1}{4-1}=0$ ← 1 — S

2 — M

$\frac{3-1}{3}=0.6$ ← 3 — L

$\frac{4-1}{3}=1$ ← 4 — XL

0.333 ← 2 — M

0 ← 1 — S

can be converted to continuous (value taken within a range like 1-10)

* To transform discrete to ~~category~~ continuous:

Every category has a rank as per their order

m_f → no. of category values against that attribute

Rank

S → 1

M → 2

L → 3

XL → 4

$$z_{if} = \frac{rank - 1}{\text{no. of cat. vals} - 1}$$

Using the numeric measure we can calculate dissimilarity

④ Mixed Attributes

Weighted Average $d(i,j) = \frac{\sum_{f=1}^{p} \delta^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta^{(f)}}$ (weight)

0.2 0.5 0.2 0.1 weights

	ordinal X_1	nominal X_2	Binary X_3	Numeric X_4
1	[			]
2				
3				
4				
5				
6				

Let weights assigned to diff. attributes as per their importance in the app^n

Diff. func^{ns}/wt. comput^n mechanism can be defined

numerical
Normalize → $\frac{|x_{if} - x_{jf}|}{x_{max} - x_{min}}$

64 - 21 (or in final matrix ÷ by max value in matrix)

Nominal → prepare dissimilarity matrix $\begin{bmatrix} 0 & & \\ d(2,1) & & \\ d(3,1) & d(3,2) & \end{bmatrix}$

Ordinal → " " " "

Binary → " " " "

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

$$d(2,1) = \frac{0.2 \times d(2,1)_{ordinal} + 0.5 \times d(2,1)_{nom} + 0.2 \times d(2,1)_{bin} + 0.1 \times d(2,1)_{num}}{0.2 + 0.5 + 0.2 + 0.1}$$

Q.

SNo	Test 1 (Nominal) 0.1	Test 2 (Ordinal) 0.5	Test 3 (Numeric) 0.4
1	code A	Fair	22
2	code B	Excellent	45
3	code A	good	64
4	code C	excellent	21

Find final dissimilarity matrix

Ans- ~~d(2,1)~~

Nominal

	1	2	3	4
1	0			
2	$\frac{1-0}{1} = 1$	0		
3	0	1	0	
4	1	1	1	0

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

~~Ordinal~~ Numeric

	1	2	3	4
1	0			
2	45-22=23	0		
3	42	19	0	
4	1	24	43	0

~~Nom~~

max → 43 min = 0

divide all by 43 → [0,1]

Ordinal		
Fair	1	$\frac{1-1}{3-1} = 0$
Good	2	$\frac{2-1}{2} = 0.5$
Excellent	3	$\frac{3-1}{2} = 1$

Ordinal	2	3	4
1	0		
2	1-0=1	0	
3	0.5	0.5	0
4	1	0	0.5	0

$d(2,1) = 0.1 \times 1 + 0.4 \times 23 + 0.5 \times 1$

$= 9.8$

$d(3,1) = 0.1 \times 0 + 0.4 \times 42 + 0.5 \times 0.5$

$= 17.05$

$d(3,2) = 0.1 \times 1 + 0.4 \times 19 + 0.5 \times 0.5$

$= 7.95$

$d(4,1) = 0.1 \times 1 + 0.4 \times 1 + 0.5 \times 1$

$= 1$

$d(4,2) = 0.1 \times 1 + 0.4 \times 24 + 0.5 \times 0$

$= 9.7$

$d(4,3) = 0.1 \times 1 + 0.4 \times 43 + 0.5 \times 0.5$

$= 17.55$

Write these 2 Qns in assignment

	1	2	3	4
1	0			
2	9.8	0		
3	17.05	7.95	0	
4	1	9.7	17.55	0

H.W

Q.

S.NO	(Binary) Test 4	(Binary) Test 5	(Nominal) Test 6
1	1	0	S
2	0	0	L
3	0	1	XL
4	1	1	L

Symmetric binary

Ans- Test 4 (Binary) & 5

	1	2	3	4
1	0			
2	0.5	0		
3	1	0.5	0	
4	0.5	1	0.5	0

$d(2,1) = \begin{matrix} & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{matrix} = \frac{1}{2}$

$d(3,1) = \begin{matrix} & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} = \frac{2}{2} = 1$

$d(3,2) = \begin{matrix} & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{matrix} = \frac{1}{2}$

Test 6

0			
1	0		
1	1	0	
1	1	1	0

Cosine Similarity

Terms in all the documents → attributes

Document	team	coach	hockey
document 1	5	0	3 → vector
document 2	3	0	2

Compare text documents by finding the cosine similarity

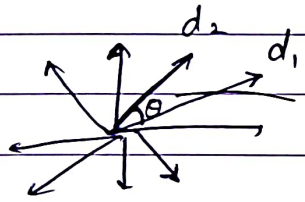
$X \to (2, 0, 3)$
$Y \to (1, 2, 3)$

x	y
1	4
2	5
3	6

Similarity (or diss.?) → $\cos(d_1, d_2) = \frac{(d_1 \cdot d_2)}{\|d_1\| \|d_2\|}$ → dot product

d_1 & d_2 → vectors

↳ norm (Length of vector)



$\cos\theta$ ↳ ∠ b/w document vectors

$\cos\theta = 0 \to$ similar

$\cos\theta = 90° \to$ ⊥r / orthogonal

$90° \to$ dissimilar

$\cos\theta = 0.42$

$\cos\theta = 0.30$ (more similar) ?

As $\theta \uparrow$, dissimilarity ↑

$d(i,j) = 1 - \cos(i,j)$

Chapter 3: Data Preprocessing

Objective: Prepare your data for requirement for data mining algorithm

Quality of data — Accuracy, Completeness, Consistency (Date field but its some other value), Timeliness (Specific time/deadline asso. w/ data) If some ppl X submit data by month end → data cannot be processed

$x = \frac{x_1 + x_2 + x_3}{3}$ ← generated

X_3 correlated to X_4, X_5, X_6 using them generate X_3

Interpretability (Format of data interpreted as per req)

Believability (How much trust on your data)

Major Tasks
- Data cleaning
- Data Integ^n
- " Reduc^n
- " Transform^n

① Data cleaning

a)

X_1	X_2	X_3	X_4
✓	✓	✓	✓
✓	✓	✓	–
✓	–	✓	–
✓	–	✓	–

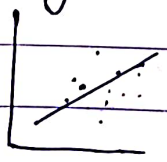
Handling Missing values

4 dimensional space (Multiple regression → linear for only 2)

1) Ignore missing values (good if dataset v. large & only few missing values) Eg- 1L total, 4 missing (X advised if moderate size)

2) Statistical method (mean ...)

3) Regression analysis



$y = 3x + 7$

	X_1	X_2
1		
2		
3		
	x	y

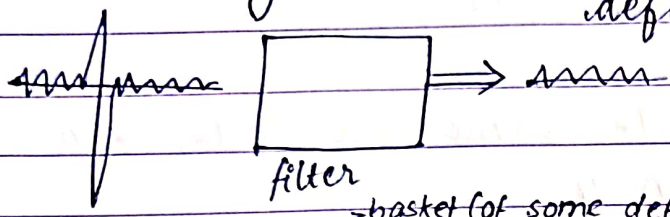
4) Predic^n mechanism (decision tree) etc. Eg- 1000 total 3 missing using 997 → train model 3 → ~~Miss~~ Predict

4, 8, 15, 21, 21, 24, 25, 28, 34

① Equal Partitioned Mean
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

② Bin Mean (replace all by mean)
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22

b) Smoothing noisy data
Some random values/variance in your data
How to handle?
Smoothing method: smooth your data within the defined limit



→ basket (of some defined size)

Eg - Binning method
4, 8, 15, 20, 21, 27, 32, 35, 91, 43, 47, 102
(102: outlier/noise)

→ sort
→ w = 4 (basket width)
Bin 1 = 4, 8, 15, 20
Bin 2 = 21, 27, 32, 35
Bin 3 = 91, 43, 47, 102

Binning by mean → calc. mean of each bin & replace all values in the bin by that

Mean(Bin 1) = 12
Bin 1 = 12, 12, 12, 12
Bin 2 = 28, 28, 28, 28
Bin 3 = 59, 59, 59, 59

Another is binning by max value (Take max in a bin)

(21), 23, 27, 32, 34, (36)
2 + 6 = 8
4 + 2 = 6
4?

Binning can be used as discretization process & data reduc^{n} (No. of bits req. is less)

Reason - Data integ" (Causes redundancy as well)

d) Resolve inconsistencies

In a field of certain dtype, there is value of another dtype

How to handle?

1) Numeric attribute → correlation coefficient [-1, 1] (Pearson)

2) Nominal attribute → Chi squared test

A.cust-id = B.cust-# both same attributes → on integ" there is redundancy

Chi-square test $\chi^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$ — Used to find redun. or inconsis. in data

corr. coeff.

$-1 \leq 0 \leq 1$

Redundant ← -0.42 ... 0.42 →

X	Y — Numeric

If no correlation (CC = 0 → then X & Y are independent)

corr = 1 → similar → redundant

CST → Assumes attributes are independent

hypothesis → if rejected →

× (has 2 nominal values)

		Play chess	✗ play chess
Y	Like sci fiction	250	200
	✗ like sci fic	50	1000

observed freq. of diff. nominal values

X	Y
PC	LSF
NPC	LSF
PC	NLSF

→ Recognize freq. of PC & NPC

→ Make contigency table

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

expected frequency

check

$e_{11} = \frac{A(A=PC) \times B(B=\text{Isb})}{n}$

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

Probab. of occurance of each → multiply both → consider independent

(m-1) x (n-1) no. of indep. var. wrt

Significance level 0.001 χ^2 value are less this much.
0.001%

507.93 (highly correlated → χ^2 15 upper limit)

Deg. of freedom	Significance	χ^2
15	0.001	15

Within 15 → independent
>15 → highly correlated

Q

X	Y
2	9
3	28
4	65
5	? 126
6	217

using Linear Reg. method, find missing value?
cos(x,y) = ? 0.926
corr(x,y) = ? 0.964
Write your observations about cos & corr & what they indicate

Ans- $\frac{28-9}{3-2} = \frac{y-65}{5-4}$

$19 = y - 65$

$y = 84$

$y = mx + c$

$y = 19x + c$

$65 = 19 \times 4 + c$

$65 = 76 + c$

$c = -11$

$y = 19x - 11$

$y =$

$\cos(x, y) = ?$

$$= \frac{2\times9 + 3\times28 + 4\times65 + 5\times126 + 6\times217}{\sqrt{2^2+3^2+4^2+5^2+6^2}\ \sqrt{9^2+28^2+65^2+126^2+217^2}}$$

$$= \frac{2294}{9.48 \times 260.87} = 0.927$$

$$\text{corr}(x,y) = \frac{\text{Covariance}(x,y)}{\text{std}(x)\cdot\text{std}(y)}$$

$$\text{Covariance}(x,y) = \frac{\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y})}{n-1}$$

$$\text{std}(x) = \sqrt{\frac{\sum_{k=1}^{n}(x_k-\bar{x})^2}{n-1}}$$

$\text{corr}(x,y) = ?$ $\quad \bar{x} = 4 \quad \bar{y} = 89$

$$\text{Covariance}(x,y) = \frac{(-4+2)(9-89)}{4} + \frac{(3-4)(28-89)}{4} + \frac{(4-4)(65-89)}{4} +$$

$$\frac{(5-4)(126-89)}{4} + \frac{(6-4)(217-89)}{4}$$

$$= \frac{(-2)\times(-80) + (-1)(-61) + 37 + 2\times128}{4} = \frac{160+61+37+256}{4}$$

$$= 128.5$$

$$Std(x) = \sqrt{\frac{(2-4)^2 + (3-4)^2 + (5-4)^2 + (6-4)^2}{4}}$$

$$= \sqrt{\frac{10}{4}} = 1.58$$

$$Std(y) = \sqrt{\frac{80^2 + 61^2 + 37^2 + 128^2 + 24^2}{4}}$$

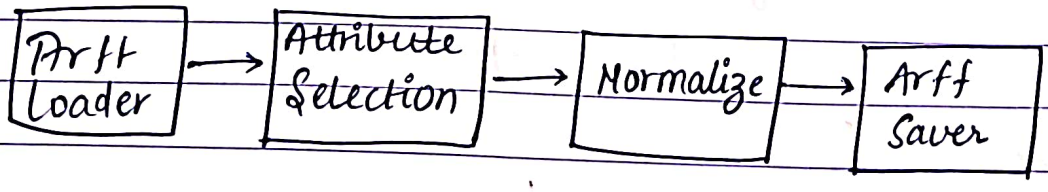
$$= \sqrt{\frac{6400 + 3721 + 1369 + 16384 + 576}{4}}$$

$$= 84.3$$

$$corr(x,y) = \frac{128.5}{1.58 \times 84.3} = 0.964$$

Lab

DataSources > Arff Loader (Drag & drop) > Configure > add file name
Filters > supervised > attribute > attribute selection



Arff loader icon > DataSet > Join w/ Attribute selec"
unsupervised > normalize > Drag & drop > join

DataSinks > ArffSaver > Drag, drop, join > Right click icon > configure > give directory to save file
▶ click → Run this flow (below pgm tab) > see log to see save successful

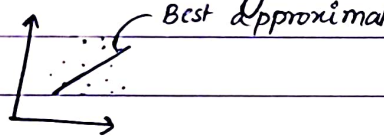
Q. Is -1 highly non correlated?

* Data Reduction

Attribute or sample level reduction

1) Dimensionality reduction (Attribute level reduc^n)
   - → Wavelet transforms
   - → PCA
   - → Feature subset selection, feature creation (10 features combined to create 1)

2) Numerosity reduction (Model level soln)



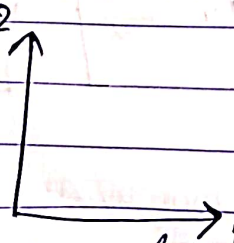
3) Data Compression
   - → Binary Compression

* PCA

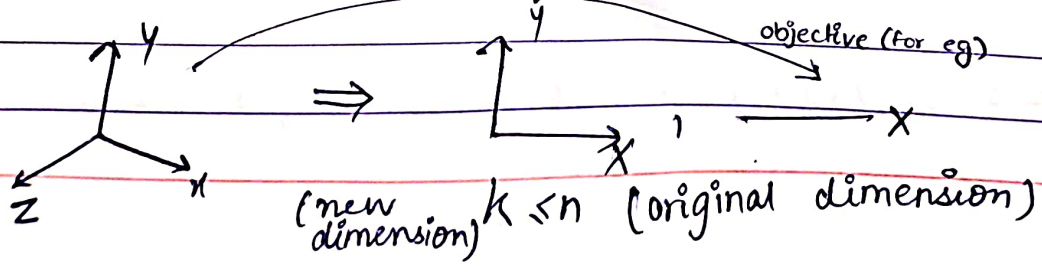
With the given data, we have to find principal components (10 attributes → 10 PC we can find)

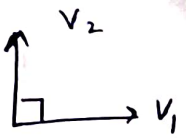
X_1	X_2
1 –	– 1
–	–
100 –	– 10

⟹ ≤ 2 dimension



2 variables can be transformed like this in terms of PCs





orthogonal → ∠ b/w vectors 90°
orthonormal → if length of these vectors is unit length & ∠ = 90°

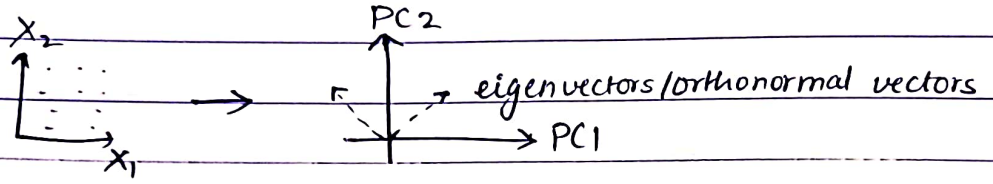
~~Ho~~

* Variance method

Variance (x) + Var(y) + var(z) = var(x)

total variance is same in original & transformed data → no data loss

① Normalize your data (to avoid bias) (Let 1-10 range)
② Compute k orthonormal vectors



③

$X_1 = a \cdot PC_1 + b \cdot PC_2$

④ Sort PCs in ↓ order of strength

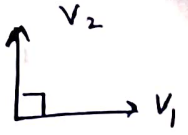
PC1, PC2 if you want to choose 1 → select PC1

PC1, PC2, PC3 data reduc^n to 2 → select first two

Eg-

X	y
132	80
130	82
136	78
140	80
134	84
130	82

A= $\bar{X}$ = 134 $\bar{y}$ = 81



orthogonal → ∠ b/w vectors 90°
orthonormal → if length of these vectors is unit length & ∠ = 90°

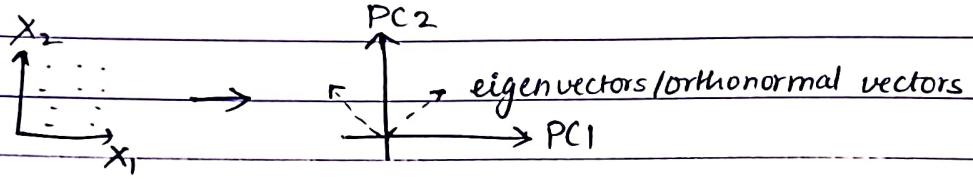
~~Ho~~

* Variance method

Variance (x) + var(y) + var(z) = var(x)

total variance is same in original & transformed data → no data loss

① Normalize your data (to avoid bias) (Let 1-10 range)
② Compute k orthonormal vectors



③

$X_1 = a \cdot PC_1 + b \cdot PC_2$

④ Sort PCs in ↓ order of strength

PC1, PC2 · if you want to choose 1 → select PC1

PC1, PC2, PC3 data reduc^n to 2 → select first two

Eg-

X	y
132	80
130	82
136	78
140	80
134	84
130	82

A: $\bar{X} = 134$ $\bar{y} = 81$

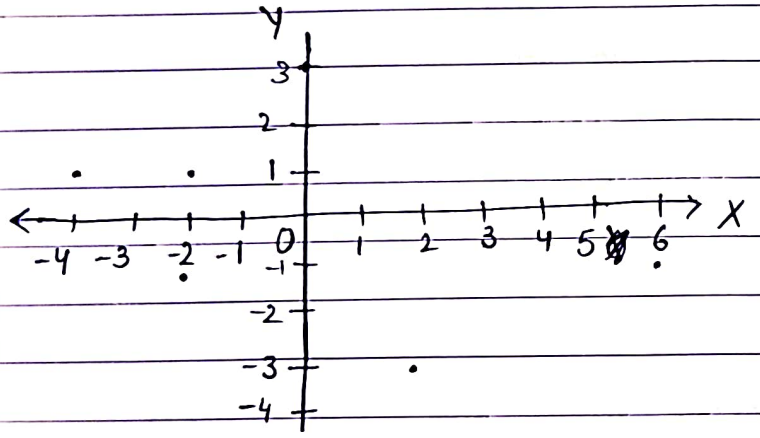
Variance = How much variation in your data wrt mean

Covariance = How 1 variable vary wrt another (If you vary x, what will be var^n in y)

① Normalize by mean

X	$X-\bar{X}$	$Y-\bar{Y}$	Y
	-2	-1	
	-4	1	
	2	-3	
	6	-1	
	0	3	
	-2	1	

centre data / normalised data



② Covariance matrix (:· variance method)

cov (X,X)

$$cov(A,B) = \sum_{i=1}^{n} \frac{(a_i - \bar{A})(b_i - \bar{B})}{n}$$ (n or n-1 both ok)

	X	Y
X	10.6	-2.6
Y	-2.6	3.67

$$cov(X,X) = \frac{(-2)\times(-2) + (-4)^2 + (2)^2 + 6^2 + (-2)^2}{6}$$

$$= \frac{4 + 16 + 4 + 36 + 4}{6} = 10.66$$

$$cov(X,Y) = \frac{(-2)\times(-1) + (-4) + (2)(-3) + 6(-1) + (-2)(1)}{6}$$

$$= \frac{2 - 4 - 6 - 6 - 2}{6} = -\frac{16}{6} = -2.66$$

$$cov(Y,Y) = \frac{(-1)^2 + 1 + (-3)^2 + (-1)^2 + 3^2 + 1}{6}$$

$$= \frac{1+1+9+1+9+1}{6}$$

$$= \frac{22}{6} = 3.66$$

$$A = \begin{bmatrix} 10.6 & -2.6 \\ -2.6 & 3.67 \end{bmatrix}$$

③ Find eigen values & eigenvectors

$$\det(A - \lambda I) = 0$$

A large matrix can be decomposed into submatrices, which are eigenvectors

$$[||||||] = [\ ] + [\ ] + [\ ]$$

Eigen vectors

Cond^n is that these eigen vectors should be orthonormal (or goral) to each other / independent to each other

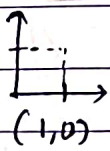
$$AI = \lambda I$$

$$AI = 0$$

$$A \cdot V = 0$$

orthogonal + unit vector = orthonormal

$$= a\begin{bmatrix} 1 \\ 0 \end{bmatrix} + b\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



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

det = 1

det of matrix → gives volume formulated by that object formed by the matrix data

$$A - \lambda I = \begin{bmatrix} 10.6-\lambda & -2.6 \\ -2.6 & 3.67-\lambda \end{bmatrix} = (10.6-\lambda)(3.67-\lambda) - (-2.6)^2$$

$$= 38.9 - 14.27\lambda + \lambda^2 - 6.76 = 0$$

$$\lambda^2 - 14.27\lambda + 32.14 = 0$$

$$\lambda = \frac{14.27 \pm \sqrt{14.27^2 - 4 \times 32.14}}{2}$$

$$= \frac{14.27 \pm \sqrt{75.07}}{2}$$

$$= \frac{22.93}{2}, \frac{5.61}{2}$$

$$= \underbrace{11.46}_{\lambda_1}, \underbrace{2.8}_{\lambda_2}$$

Eigenvector

$$A \cdot V = \lambda V$$

$$(A - \lambda I) \cdot V = 0 \quad \text{(orthogonal)}$$

Let $\lambda = \lambda_1$ (11.46)

$$\begin{bmatrix} 10.6-\lambda & -2.6 \\ -2.6 & 3.67-\lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 11.46 \begin{bmatrix} x \\ y \end{bmatrix}$$

should be 0 ?

$$\begin{bmatrix} 9.6 & -2.6 \\ -2.6 & 2.67 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 11.46 \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 9.6y - 2.6x \\ -2.6x + 2.67y \end{bmatrix} = \begin{bmatrix} 11.46x \\ 11.46y \end{bmatrix}$$

$$9.6y - 14.06x = 0 \Rightarrow y = 1.46x$$

$$-2.6x - 8.79y = 0$$

$$-0865x = 2.6y$$

$$-0.332x = y$$

$$-2.6x = 7.795y$$

$$-0.333x = y$$

$$\begin{bmatrix} 1 \\ -0.33 \end{bmatrix}$$

$$\begin{bmatrix} -0.865x & -2.6y \\ -2.6x & -7.795 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 11.46\begin{bmatrix} x \\ y \end{bmatrix}$$ → should be 0?

$$\begin{bmatrix} -0.865x - 2.6y \\ -2.6x - 7.795y \end{bmatrix} = \begin{bmatrix} 11.46x \\ 11.46y \end{bmatrix}$$

$$-0.865x - 2.6y = 11.46x$$

~~$y = -0.135x$~~

$$y = -4.73x$$

$x = t$ or $x = 1$

$y = -4.73$

$$\begin{bmatrix} t \\ -4.73t \end{bmatrix} = t\begin{bmatrix} 1 \\ -4.73 \end{bmatrix}$$

1st eigenvector $= \begin{bmatrix} 1 \\ -4.73 \end{bmatrix}$

$$\begin{bmatrix} 10.6-2.8 & -2.6 \\ -2.6 & 3.67-2.8 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 2.8\begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 7.8 & -2.6 \\ -2.6 & 0.87 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 2.8\begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} 7.8x - 2.6y \\ -2.6x + 0.87y \end{bmatrix} = \begin{bmatrix} 2.8x \\ 2.8y \end{bmatrix}$$

$7.8x = 2.6y$

$y = 3x$

$0.87y = 2.6x$

$y = 3x$

$\begin{pmatrix} 1 \\ 3 \end{pmatrix}$

$5x = 2.6y$

$y = 1.92x$

2nd eigenvector $= \begin{bmatrix} 1 \\ 1.92 \end{bmatrix}$

(3D space) $V \rightarrow W$ (2D space)

(2, 1, 4) (4, 3)

To transform you need a transform" matrix

Make eigenvector unit length

$$\frac{1}{\sqrt{1^2 + (-4.73)^2}}$$

$$= \frac{1}{\sqrt{23.37}} = \frac{1}{4.83} = 0.2$$

$$1^{st}\ EV = \begin{bmatrix} 0.2 \\ -0.97 \end{bmatrix}$$

$$2^{nd}\ EV = \begin{bmatrix} 0.46 \\ 0.88 \end{bmatrix}$$

$$\begin{bmatrix} 0.46 & 0.2 \\ 0.88 & -0.97 \end{bmatrix}$$

$$M_{6\times 2} \cdot \begin{bmatrix} 0.46 & 0.2 \\ 0.88 & -0.97 \end{bmatrix}_{2\times 2} = \begin{bmatrix} \cdot \end{bmatrix}_{6\times 2}$$

ash if orig matrix or this?

$$\begin{bmatrix} -2 & -1 \\ -4 & 1 \\ 2 & -3 \\ 6 & -1 \\ 0 & 3 \\ -2 & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 0.46 & 0.2 \\ 0.88 & -0.97 \end{bmatrix}}_{\text{Eigenvector}} = \begin{matrix} \text{PC1} \\ \text{PC2} \end{matrix} \begin{bmatrix} -1.8 & -1.37 \\ -0.96 & -1.77 \\ -1.72 & 3.31 \\ 1.88 & 2.17 \\ 2.64 & -2.91 \\ 0.04 & -1.37 \end{bmatrix}$$

Var = (PC1) Var = (PC2)

represent 98% var" in data so reduce 100 dimensions to only these 3

98% 2%

1, 2, 3 .... 100 $\rightarrow$ $P_{46}\ P_{32}\ P_{67} \sim P_1$

PC_1 PC_{100} sort

Most variance shifted to some PC values

Total var^n = 10.6 + 3.67 ≈ 14
(X,X) (Y,Y)

Now var^n shifted to some PCs

PCA → dimension reduc^n technique

Var^n moved in only a few directions (lesser dimension spaces) → represent overall data
That reduced dimension maximally represent your data

Ch-3

* Sampling
  (Technique to ↓ size of data
  → selecting some samples from the given data

Types

① Random sampling:
1L → select 10k randomly
Not good choice, maybe 10k belong to the same class

② Sampling w/o replacement (better training than ↓)
③ " w/ " (redundancy possible)
④ Stratified Sampling
As per data dist^n, select samples

250 ← {- - - - - - -
250 ← {- - - - -
250 ← {- - - - - -

Partition dataset & select from each
Skewed data (biased towards majority class)
↳ sampling → balance

→ Undersampling (Majority class instances are eliminated) ~~80k = + 20k =~~ 60k + removed
→ Oversampling (Regenerate minority instances w/o elim. maj) Generate 60k pseudo instances
→ Hybrid Sampling

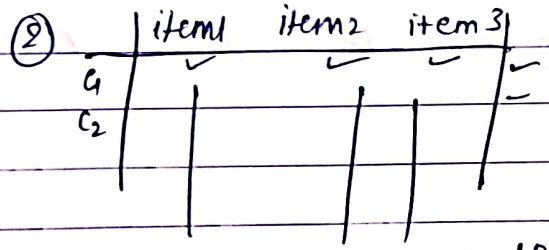
Ch-8

/Data Mining / Learning

Classification algorithms

① Supervised (2 step-process → training & trainging) labelled on basis of similarity they get
② Unsupervised (No training → categ. given data for training purpose)
③ Semi supervised (

②



item1 item2 item3 compare similarity of each value

① Feedback mech

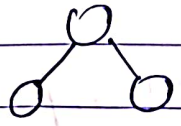
Learns from errors

Types:

If value of class attribute the discrete → classification
continuous → regression

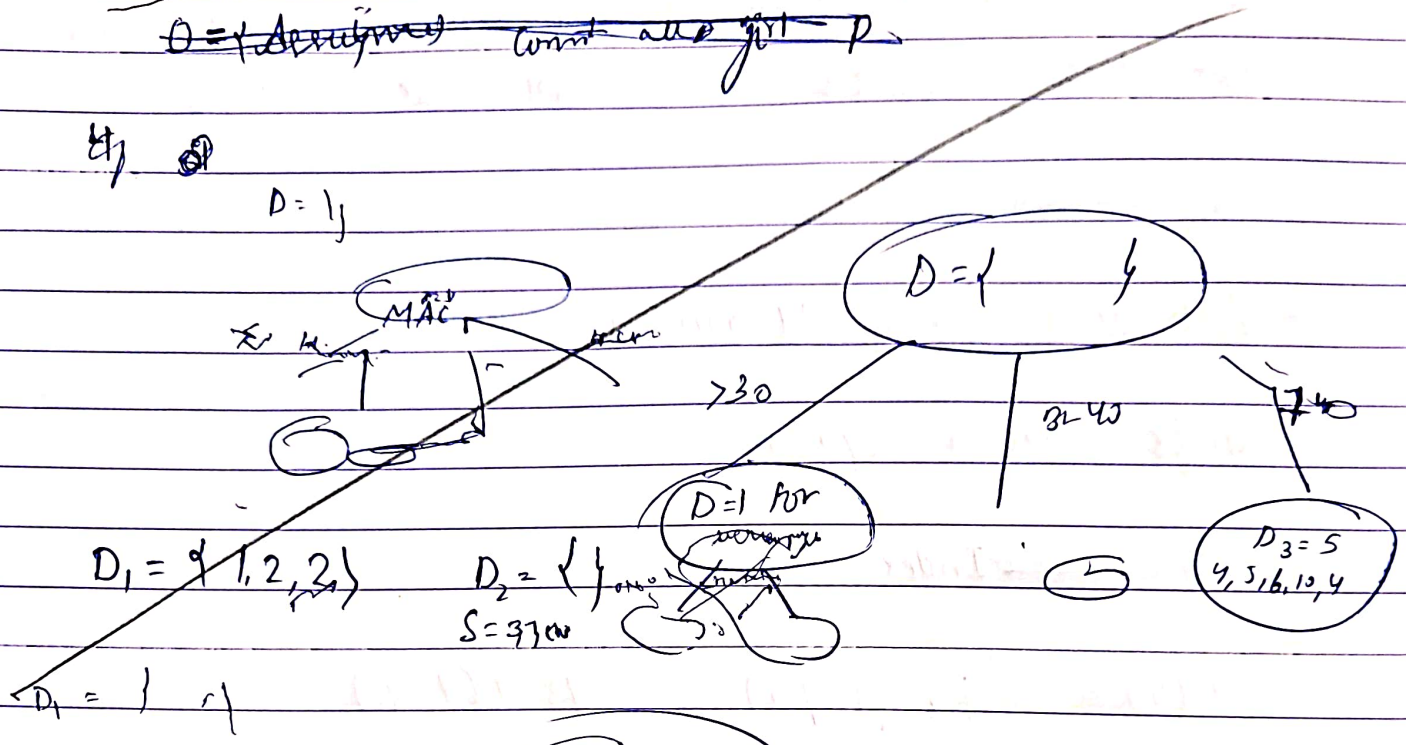
X_1	X_2	X_3	class
✓	✓	–	+
✓	–	✓	–
✓	✓	✓	?
✓	✓	✓	?

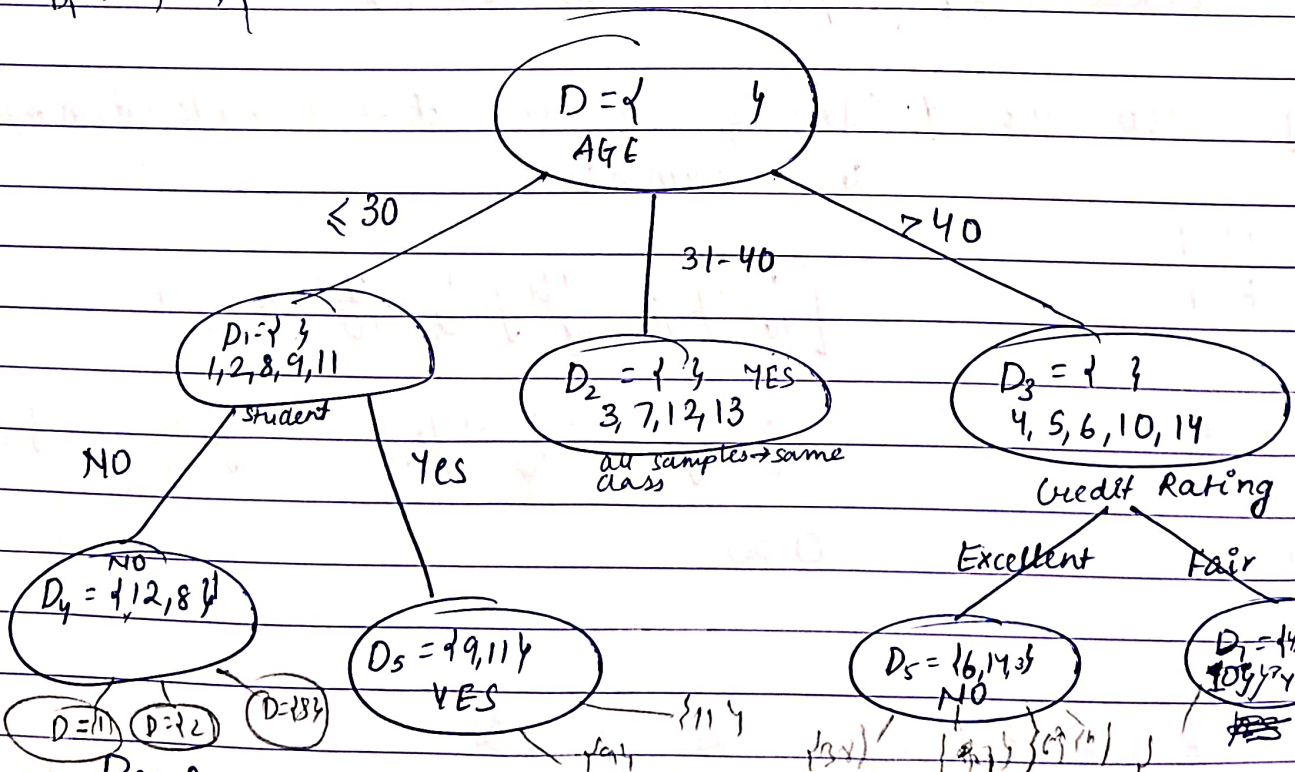
→ supervised



↓

sweeper wise eg.





Performance good if all attributes are discrete attribute (Ranking)

Selected attribute → selected randomly

If 12,8 other classes → either split further or use voting

see → Dec. Tree Madhumita

* Attribute Selec^n Measures

Information Gain
Gain Ratio
Gini ~~Ratio~~ Index

$$H(Y) = -\sum_{i=1}^{m} p_i \log(p_i) \qquad p_i = P(Y = y_i)$$

Eg- Hello world (amt of info in these 2 words → given by above formula)

H = 1
E = 1
L = 3
O = 2
W = 1
R = 1
D = 1
10

$$= -\left[\frac{1}{10}\log_2\frac{1}{10} + \frac{1}{10}\log_2\frac{1}{10} + \frac{3}{10}\log_2\frac{3}{10} + \frac{2}{10}\log_2\frac{2}{10} + \frac{1}{10}\log_2\frac{1}{10} + \frac{1}{10}\log_2\frac{1}{10} + \frac{1}{10}\log_2\frac{1}{10}\right]$$

$$= 0.30$$

k = No. of diff. class labels
m_i = No. of values in the i^{th} interval of a partition
m_{ij} = No. " " of class j in interval i

Entropy e_i of i^{th} interval

$$e_i = -\sum_{i=1}^{k} p_{ij} \log_2 p_{ij} \qquad p_{ij} = \frac{m_{ij}}{m_i}$$

Randomness / Impurity in data → Entropy

*Pure partition

$D = \{10, 7+, 3-\}$ (10 instances)

$D_1 = \{3-\}$ $D_2 = \{7+\}$

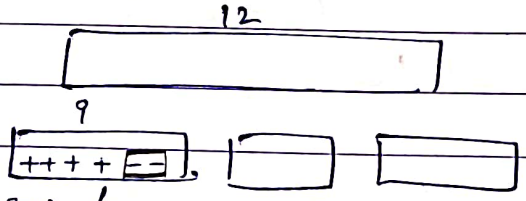
Objective of decision tree:
To classify sample in pure partition

Impurity involved

$D = \{10\}$

$D_1 = \{2-, 1+\}$ $D_2 = \{(-), 6+\}$ ← Amt of impurity / randomness

↳ got this partit^n due to selec^n of some attribute



+ve class -ve class

$$-\left[\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right]$$

$$[0.66 \times -0.59 + 0.33 \times -1.59]$$

$$= \cancel{0.66 \times 0.18}$$

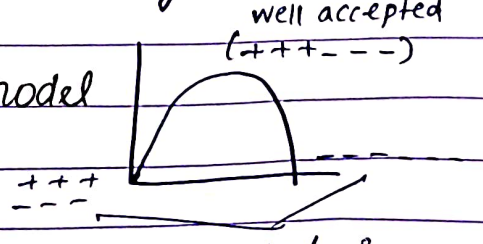
$$= -[-0.38 - 0.52]$$

$$= 0.9$$

Dup +++--- equal dist^n 50% probability → value of entropy is maximum (1)

++++++ 100% prob. → 0 entropy / impurity

→ Pure part^n X good for training model
→ Biased towards that class



X accepted in Data Mining → training will X be good

Attribute Selec^n Measure : Information Gain (ID3/C4.5)

Expected Inform^n (entropy)

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

Total info in original data (Amt of impurity in data before partition)

m → no. of class labels

Inform^n needed

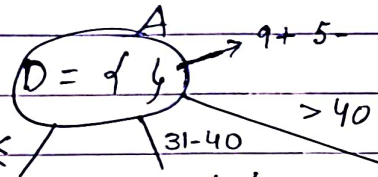
$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

" " remaining in splitted part

normalise impurity

Inform^n gained

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



$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

(Amt of impurity in given dataset)

Find impurity in respective part^n

$$D_1 = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

normalise to avoid biasing effect on diff. dist^{ns}

(0 4+ 0- → pure)

$$\frac{5}{14}[D_1] + \frac{4}{14}\left[-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right] +$$

$$\frac{5}{14}\left[-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right] = 0.694$$

(impurity ↓ gain ↑)

Gain(age) = Info(D) − $Info_{age}$(D) = 0.246 → max

Gain(income) = 0.029

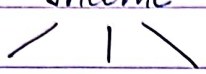
Gain(student) = 0.151

Gain(credit_rating) = 0.048

Decision Tree → good for discrete attributes

If continuous attribute in data,

Income
1000
2000
3000
2000
⋮
5000



75	70	85	60	120	100	95	90	125	220
No	No	No	Yes	Yes	Yes	No	No	No	No

60	70	75	85	90	95	100	120	125	220

① Sort in ↑ order

② Append the class label corr. to the row (above respective value)

③ Consider any value <60 & >220

	class
60	No
70	No

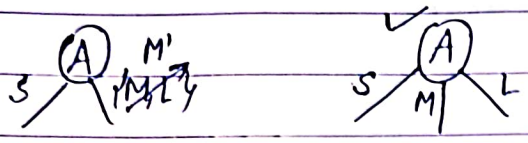
	55		60		70		75		85		90		95		100		120		125		220		230	
	<	>	<	>	<	>	<	>	<	>	<	>	<	>	<	>	<	>	<	>	<	>	<	>
Yes	3		0	3	0	3	0	3	1	2	2	1	3	0	3	0								
No	7		1	6	2	5	3	4	3	4	3	4	3	4	4	3								

No No No Y Y Y N N N N

You get dist" b/w diff intermed. pt.

Calculate gain → min value → select that as split point

Inform" gain measure is biased towards attributes w/ a large no. of values (2 ~~attributes~~ discrete values < 10 discrete values)



Normalise gain w/ cost

(pay cost for no. of splits)

$$GainRatio(A) = Gain(A) / SplitInfo(A)$$

(no. of part")

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

age → <=30 → $-\frac{5}{14}\log_2\frac{5}{14}$

$-\frac{4}{14}\log_2\frac{4}{14}$

$-\frac{5}{14}\log_2\frac{5}{14}$

} sum of all

$$Gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

+ + + − −

$$1 - \left[\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right]$$

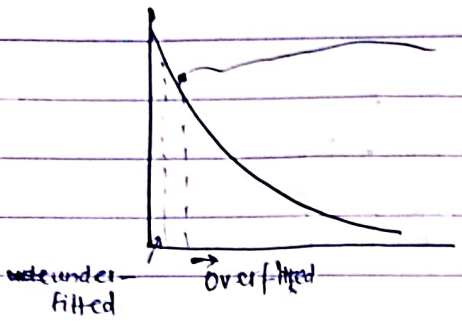
$$\Delta gini(A) = gini(D) - gini_A(D)$$

Consider smallest gini value for attribute selection

0 nodes → error ↑

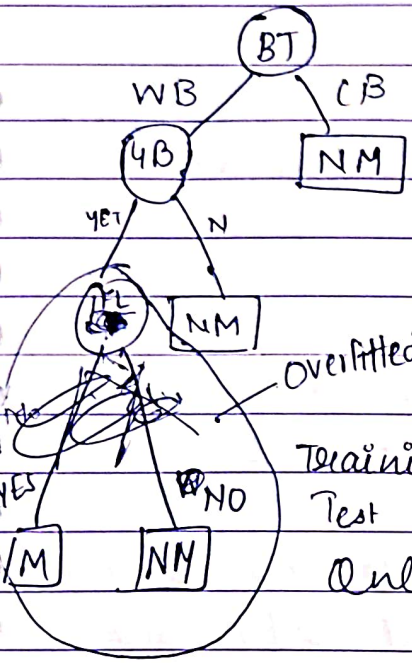
as you ↑ no. of node → Both errors ↓

After a pt if we ↑ no. of nodes, testing error



Both Train & Test error should be ↓ within a certain node

Underfitting → ↓ dataset

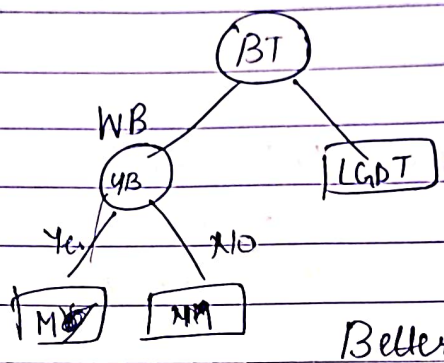


Overfitted

Training Error : ~~20%~~ 0 ~~30~~

Test Error : ~~10~~ 30%

Only splits master →



Better decision Tree

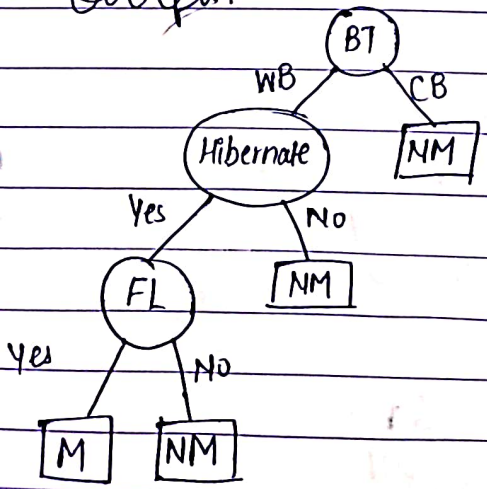
Training E. : 20%

Test Error : 10%

T. E ↓

not always you can ↓ ~~the~~ ~~any~~ root ~~to~~ bit w 10% error remains in Deci. Tree due to ~~test~~.)

Overfit.



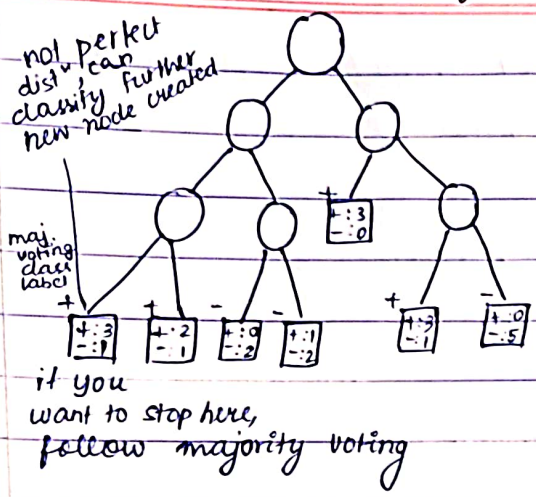
Training Error : 0

Test Error : 30% (3 samples wrongly classified)

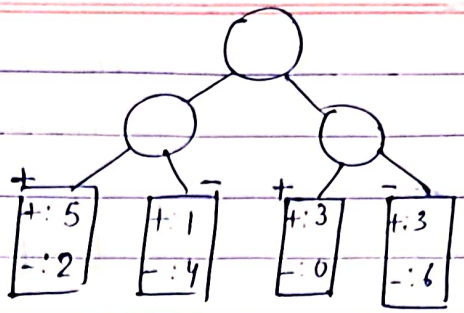
Reason for error : In training data with WB & Hibernates (no) only 1 class label → no. So training of model biased towards 1 class. No sample w/ WB & H(~~yes~~ no) & yes class

∴ Overfitting

Q. How to find amt of training & test error in the 2 decision trees?



if you want to stop here, follow majority voting



$$\text{Training Error} = \frac{2+1+0+3}{24} \times 100 = 25\%$$

Training Error $= 1+1+0+1+0+1+0$

$$= \frac{4}{24 \text{ (total samples)}} \times 100$$

$$= 16\ \%$$

+ : 3 - : 1 — +ve class label, 1 misclassified

This training error can be assumed to be equal to test error as we don't have test set

Misclassific" of sample is not considering complexity of model (no. of nodes). Include that for better evaluation

$$e_g(T) = \frac{\sum_{i=1}^{k} [e_i(T) + \Omega_i(T)]}{N(T)}$$

- k → no. of leaf node
- $e_i(T)$ → amt of error on respective leaf node
- $\Omega_i(T)$ → cost asso^d w/ that node creating
- $N(T)$ → total no. of samples

Overfitted DT may have many nodes so consider cost of creating new node in DT)

a) $\Omega = 0.5$

$$DT_1 = \frac{4 + 7 \times 0.5}{24} = 0.3125$$

(4 = misclassified; 7 = no. of nodes; 0.5 = penalty for each node)

$$DT_2 = \frac{6 + 4 \times 0.5}{24} = 0.333$$

$DT_1 < DT_2$

b) $\Omega = 1$

$$DT_1 = \frac{4 + 7 \times 1}{24} = 0.458$$

$$= \frac{6+4}{24} = 0.416$$

$DT_1 > DT_2$

↓ node ↓ error good

Ω can be used to control overfitting in your decision tree. (tuning parameter)

$\Omega \rightarrow$ penalty asso^{d} on each node

If you consider creation of new nodes & you are not ~~b~~ getting ~~it~~ benefit, then no need o.w more overfitting

Test. $X = (\text{Home Owner} = \text{No}, \text{Marital Status} = \text{Married}, \text{income} = 120) = ?$

A - $P(Y = \text{yes}) = \frac{3}{10}$ $\quad P(Y = \text{no}) = \frac{7}{10}$

$P(X \mid Y = \text{yes}) = P(\text{HO} = \text{No} \mid Y = \text{yes}) \times P(\text{MS} = \text{Married} \mid Y = \text{yes}) \times P(\text{income} = 120)$ ← continuous

$= \frac{3}{3} \times \frac{0}{3} \times P(\text{income} = 120) = 0$

$P(X \mid Y = \text{no}) = P(\text{HO} = \text{No} \mid Y = \text{no}) \times P(\text{MS} = \text{married} \mid Y = \text{no}) \times P(\text{income} = 120 \mid Y = \text{no})$

$= \frac{4}{7} \times \frac{4}{7} \times P(\text{income} = 120 \mid Y = \text{no})$

$P(\text{income} = 120 \mid Y = \text{yes}) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp \frac{-(120-\mu_{ij})^2}{2\sigma_j^2} = 0.0072$ (assume)

$P(AI = 120 \mid Y = \text{no}) = 0.0072$

$= \frac{4}{7} \times \frac{4}{7} \times 0.0072 = 0.0023$

$P(Y = \text{No} \mid X) = P(X \mid Y = \text{No}) \times P(Y = \text{No})$

$= 0.0023 \times \frac{7}{10}$

$= 0.0016 \xrightarrow{10} \text{max}^m \rightarrow \text{No}$ ✓

Problem w/ Naive Bayes Classifier

If var are cond^{nal} indep then its a v. good classifier & v frequently used

Since it is a product of cond^{al} probab.,

If any cond^{al} probab becomes zero, then whole prob. becomes 0, so decision gets biased towards another

Test. $X = (\text{Home Owner} = \text{No}, \text{Marital Status} = \text{Married}, \text{income} = 120) = ?$

A - $P(Y = \text{yes}) = \frac{3}{10}$ $\quad P(Y = \text{no}) = \frac{7}{10}$

$P(X \mid Y = \text{yes}) = P(\text{HO} = \text{No} \mid Y = \text{yes}) \times P(\text{MS} = \text{Married} \mid Y = \text{yes}) \times P(\text{income} = 120)$ ← continuous

$= \frac{3}{3} \times \frac{0}{3} \times P(\text{income} = 120) = 0$

$P(X \mid Y = \text{no}) = P(\text{HO} = \text{No} \mid Y = \text{no}) \times P(\text{MS} = \text{married} \mid Y = \text{no}) \times P(\text{income} = 120 \mid Y = \text{no})$

$= \frac{4}{7} \times \frac{4}{7} \times P(\text{income} = 120 \mid Y = \text{no})$

$P(\text{income} = 120 \mid Y = \text{yes}) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp \frac{-(120-\mu_{ij})^2}{2\sigma_j^2} = 0.0072$ (assume)

$P(AI = 120 \mid Y = \text{no}) = 0.0072$

$= \frac{4}{7} \times \frac{4}{7} \times 0.0072 = 0.0023$

$P(Y = \text{No} \mid X) = P(X \mid Y = \text{No}) \times P(Y = \text{No})$

$= 0.0023 \times \frac{7}{10}$

$= 0.0016 \xrightarrow{10} \text{max}^m \rightarrow \text{No}$ ✓

Problem w/ Naive Bayes Classifier

If var are cond^{nal} indep then its a v. good classifier & v frequently used

Since it is a product of cond^{al} probab.,

If any cond^{al} probab becomes zero, then whole prob. becomes 0, so decision gets biased towards another

Joint Probability $P(x, y)$

Conditional " $P(x|y)$

$$P(x, y) = P(y|x)P(x) = P(x|y) \times P(y)$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$ Bayes Theorem

x				
X_1	X_2	X_3	X_4	C
S	–			+
M	–			+
S				–
M	–			+
L				–

Training Phase: $P(Y|X)$

Test Phase X' $P(Y'|X') = P(Y' = Yes|X') = 0.25$

$P(Y' = No|X') = 0.33$

Max value → that will be the prediction

$$P(Y = Yes|x) = \frac{P(x|Y = Yes) \times P(Y = Yes)}{P(x)}$$

$$P(Y = No|x) = \frac{P(x|Y = No) \times P(Y = No)}{P(x)}$$

Evaluate & compare

(denominator same so no need to compare)

$P(Y = Yes) = \frac{3}{5}$ $P(Y = No) = \frac{2}{5}$

reading skill arm length Age

↑ in arm length → ↑ reading skill when Age is varied

If Age fixed, reading & arm cond^{nal} Independent

$$P(x|Y = y) = \prod_{i=1}^{d} P(X_i|Y = y)$$

If var are cond. indep their prob = product of individual cond^{nal} prob

Joint Probability $P(x, y)$

Conditional " $P(x|y)$

$$P(x, y) = P(y|x)P(x) = P(x|y) \times P(y)$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$ Bayes Theorem

x				
X_1	X_2	X_3	X_4	C
S	–			+
M	–			+
S				–
M	–			+
L				–

Training Phase: $P(Y|X)$

Test Phase X' $P(Y'|X') = P(Y' = Yes|X') = 0.25$

$P(Y' = No|X') = 0.33$

Max value → that will be the prediction

$$P(Y = Yes|x) = \frac{P(x|Y = Yes) \times P(Y = Yes)}{P(x)}$$

$$P(Y = No|x) = \frac{P(x|Y = No) \times P(Y = No)}{P(x)}$$

Evaluate & compare

(denominator same so no need to compare)

$P(Y = Yes) = \frac{3}{5}$ $P(Y = No) = \frac{2}{5}$

reading skill arm length Age

↑ in arm length → ↑ reading skill when Age is varied

If Age fixed, reading & arm cond^{nal} Independent

$$P(x|Y = y) = \prod_{i=1}^{d} P(X_i|Y = y)$$

If var are cond. indep their prob = product of individual cond^{nal} prob

~~R = 2x~~ $f_+ = 50$ $f_- = 5$

$$R(R_1) = 2x\left[f_+ \log_2\left(\frac{f_+}{e_+}\right) + f_- \log_2\left(\frac{f_-}{e_-}\right)\right]$$

$$= 2x\left[50 \log_2\left(\frac{50}{34.375}\right) + 5 \log_2\left(\frac{5}{20.625}\right)\right]$$

$$= 2x\left[50 \log_2(1.45) + 5 \log_2(0.24)\right]$$

$$= 2 \times [50 \times 0.53 + 5 \times (-2.05)]$$

$$= 2 \times 16.25$$

$$= 32.5$$

$$R(R_2) = 2x\left[2x \log_2\left(\frac{2}{1.25}\right) + 0\right]$$

$$= 2x2x \log_2 1.6$$

$$= 4 \times 0.678$$

$$= 2.712$$

$R(R_1) = 32.5$ ✓ (higher one is better) $(R_1 > R_2)$ (R_1 better rule)

$R(R_2) = 2.712$

* Foil Gain

+ve covered by R_1

R_1 : $A \longrightarrow +$ $p_0 n_0$ { which one is better? what gain you get in

R_2 : $A \wedge B \longrightarrow +$ $p_1 n_1$ { the extended rule

$$Foil\_gain = P_1 \times \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0}\right)$$ (coverage x diff. in accu)

$\{\} \rightarrow +$ default rule

$D = +: 100$

$-: 60$

R_1 : $A \rightarrow +$ 50 5- 4 coverage of rule ss 60 + 5-

R_2 : $A \wedge B \rightarrow +$ 1

Find gain if > th then accept

~~R = 2x~~ $f_+ = 50$ $f_- = 5$

$$R(R_1) = 2x\left[f_+ \log_2\left(\frac{f_+}{e_+}\right) + f_- \log_2\left(\frac{f_-}{e_-}\right)\right]$$

$$= 2x\left[50 \log_2\left(\frac{50}{34.375}\right) + 5 \log_2\left(\frac{5}{20.625}\right)\right]$$

$$= 2x\left[50 \log_2(1.45) + 5 \log_2(0.24)\right]$$

$$= 2 \times [50 \times 0.53 + 5 \times (-2.05)]$$

$$= 2 \times 16.25$$

$$= 32.5$$

$$R(R_2) = 2x\left[2x \log_2\left(\frac{2}{1.25}\right) + 0\right]$$

$$= 2x2x \log_2 1.6$$

$$= 4 \times 0.678$$

$$= 2.712$$

$R(R_1) = 32.5$ ✓ (higher one is better) $(R_1 > R_2)$ (R_1 better rule)

$R(R_2) = 2.712$

* Foil Gain

+ve covered by R_1

R_1 : $A \longrightarrow +$ $p_0 n_0$ { which one is better? what gain you get in

R_2 : $A \wedge B \longrightarrow +$ $p_1 n_1$ { the extended rule

$$Foil\_gain = P_1 \times \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0}\right)$$ (coverage x diff. in accu)

$\{\} \rightarrow +$ default rule

$D = +: 100$

$-: 60$

R_1 : $A \rightarrow +$ 50 5- 4 coverage of rule ss 60 + 5-

R_2 : $A \wedge B \rightarrow +$ 1

Find gain if > th then accept

Issue:

How to know 1 rule is better of than the other

Training dataset

Eg- + 100

- 60

R_1 : 50 + 5 -

R_2 : 2 + 0 -

Accuracy = out of covered, how many have the correct class. (same class as the rule)

R_1

Let measurement be towards +ve class

Accuracy $(R_1) = \frac{50}{55} = 90\%$

$(R_2) = \frac{2}{2} = 100\%$

Problem w/ Accuracy → X good choice if data imbalanced

Likelihood Ratio statistic

$$R = 2 \sum_{i=1}^{k} f_i \log_2 \left(\frac{f_i}{e_i}\right)$$

(f_i: coverage of rule; $\frac{f_i}{e_i}$: accuracy of rule)

k = no. of classes

f_i = observed frequency of class i examples covered by the rule

e_i = expected freq^y of a rule that makes random predictions

R_1 $e^+ = \frac{55 \times 100}{160}$ ← total coverage/freq, Total +ve $\quad e^- = 55 \times \frac{60}{160}$ → 55 covered, 60 samples -ve

R_2 $e^+ = 2 \times \frac{100}{160}$ $\quad e^- = 2 \times \frac{60}{160}$

expected frequencies: probab belonging to +ve/-ve

Accuracy(R) = $\frac{1}{2}$ = 50%

↑ Coverage means acceptability of rule is ↑

R_2 ($A_1 = a_1$) ⇒ - (1,2,4)

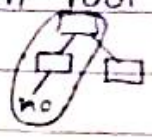
conflict w/ R_1 if $A_2 = a_1$ in prediction

① ↑ priority wrto rule w/ more attribute tests
② one w/ ↓ misclassification rule

To get rules:

① From dataset prepare a DT. for every path from root to leaf, write a rule for it

(Age = young) ∧ (student = no) ⇒ buys_computer = no



Issue: Rule set may have redundant / conflicting rules will be observed in rule set

Rule set must be mutually exclusive & exhaustive
(one sample covered by one rule) (some sample cannot be covered by >1 rule) (all samples in dataset must be covered by at least 1 rule)

② Sequential covering method

Rules are learnt 1 at a time

Write some rule for a class

$X \Rightarrow Y$

1,2 satisfies → remove

Threshold → Eg- accuracy should be >= 60%

$X \wedge Z \Rightarrow Y$ Acc: 40% X

$X \Rightarrow Y$ Acc = 80% ✓

Issue:

How to know 1 rule is better of than the other

Training dataset

Eg- + 100

- 60

R_1 : 50 + 5 -

R_2 : 2 + 0 -

Accuracy = out of covered, how many have the correct class. (same class as the rule)

R_1

Let measurement be towards +ve class

Accuracy $(R_1) = \frac{50}{55} = 90\%$

$(R_2) = \frac{2}{2} = 100\%$

Problem w/ Accuracy → X good choice if data imbalanced

Likelihood Ratio statistic

$$R = 2 \sum_{i=1}^{k} f_i \log_2 \left(\frac{f_i}{e_i}\right)$$

(f_i: coverage of rule; $\frac{f_i}{e_i}$: accuracy of rule)

k = no. of classes

f_i = observed frequency of class i examples covered by the rule

e_i = expected freq^y of a rule that makes random predictions

R_1 $e^+ = \frac{55 \times 100}{160}$ ← total coverage/freq, Total +ve $\quad e^- = 55 \times \frac{60}{160}$ → 55 covered, 60 samples -ve

R_2 $e^+ = 2 \times \frac{100}{160}$ $\quad e^- = 2 \times \frac{60}{160}$

expected frequencies: probab belonging to +ve/-ve

↓ node ↓ error good

Ω can be used to control overfitting in your decision tree. (tuning parameter)

$\Omega \rightarrow$ penalty asso^{d} on each node

If you consider creation of new nodes & you are not b getting ~~the~~ benefit, then no need ow more overfitting

* Overfitting & Tree Pruning

Lack of samples

Noise in data

2 ways to handle overfitting:

1) Prepruning (check cond^{n} of if while adding nodes to DT)
2) Postpruning (construct normally, then check cond^{n}, If accuracy ↑ after deleting nodes then do it)

* Rule based classifier

Prepare ↓ rule set from data set

	A_1	A_2	A_3	C
1	a_1	a_2	x_1	+
2	a_1	a_2	x_2	-
3	a_1	a_3	x_3	+
4	a_1	a_2	x_2	-

Rule set
R_1
R_2
R_3
} used to decide class label of test set

Test A: a_1 $A_2 = a_2$ $A_3 = a_3$ & C = ?

Rule: $X \Rightarrow Y$

Antecedant Consequent

R_4: $(A_1 = a_1) \wedge (A_2 = a_2) \Rightarrow +$

Coverage(R) = $\frac{|X|}{|D|}$ no of antecedents

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

coverage → how many instances your rule covers in dataset (1,2,4)

A_1 A_2

Coverage (R) = $\frac{|X|}{|D|} = \frac{2}{3} = 67\%$ (4 was added later)

accuracy → whatever the coverage, ~~what part~~ how many ha same class label as the rule

$\frac{n_c + mp}{n+m}$

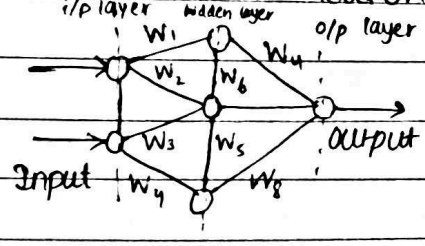
To handle it, M-estimation

$$P(x_i | y_j) = \frac{n_c + mp}{n+m}$$ cannot be 0

n: Total no. of instances from class y_j (yes class)
n_c: No. of training examples from class y_j that take on the value x_i
m: equivalent sample size (will be given in Qn)
p: user defined parameter (tuning param.)

*Neural Network Classifier

↳ Network of neurons



0.431 + (<→+)
↑
0.5 threshold
↓
0.701 -

Objective: Get optimum weights corr. to diff. links in your network.

Do updates in wt based on instances in your sample

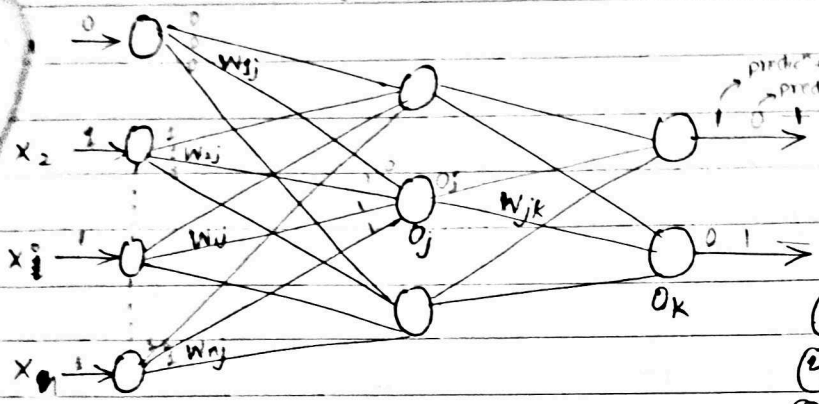
w_1 w_2 w_3 ...
1 → . . . . .
2 . . . .
3 . . . . .
.
n → . . . . . .

no. of nodes ↑ → complexity ↑

set of nodes
i/p layer: through which we receive i/p from dataset/env
o/p layer: " " " " o/p
hidden layer: b/w i/p & o/p layer

Multilayer Feedforward Neural Network

fully connected n/w

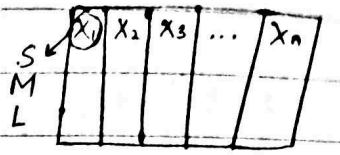


(1) Initialize all the weights and the bias in the network
(2) While terminating conditions is not satisfied {
(3) for each training tuple X in D {
(4) // Propogate the i/p forward
(5) for each i/p layer unit j {
(6) $O_j = I_j$ simply fwd the received i/p
(7) for each hidden or o/p layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$ // Compute the net i/p of unit j wrt the previous layer i
(9) $O_j = \frac{1}{1+e^{-I_j}}$
(10) // Backpropagating the error
(11) for each unit j in the o/p layer
(12) $Err_j = O_j(1-O_j)(T_j - O_j)$
(13) for each unit j in the hidden layer from the last to the first hidden layer
(14) $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$ // Compute the error wrt the next higher layer k
(15) for each weight w_{ij} in the n/w {
(16) $\Delta w_{ij} = (l) Err_j O_j$
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$
(18) for each bias θ_j in n/w {
(19) $\Delta \theta_j = (l) Err_j$
(20) $\theta_j = \theta_j + \Delta \theta_j$
} }

NN varies due to [illegible]
see → how to [illegible]

① As many features in dataset, consider that many nodes as I/P layer

② 0 S, 1 M, 2 L → X_1 → ○
X_2 → ○



Diff types of attributes

② X_1 → ○ (1)
X_1' → ○ (0)
X_1'' → ○ (0)

	X_1	X_1'	X_1''
S →	1	0	0
M →	0	1	0
L →	0	0	1

(or 0 0 0)

For diff values consider diff nodes

X_2 → ○

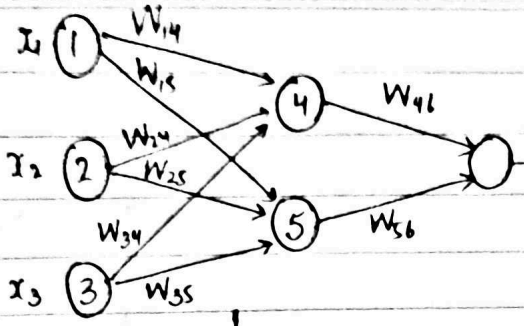
③ Continuous attribute

X
0 1 2 3 4 5 → ○
6 7 8 9 → ○
10 11 ⋮ 15 → ○

Node for each range

	X_1	X_2	X_3	...	X_n	C
1	0	1	0	-	1	0.531
2						
3						

0.421 → error

Q. 

Find updated wts & bias after 1st iteration

$\alpha = 0.9$

	x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
①	1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

Ans- $I_4 = \sum_i w_{ij} O_i + \theta_j$

$= w_{14} O_1 + w_{24} O_2 + w_{34} O_3 + \theta_4$

$= 0.2 \times 1 + 0.4 \times 0 + (-0.5) \times 1 + (-0.4)$

$= -0.7$

$O_4 = \frac{1}{1+e^{+0.7}} = 0.332$

$I_5 = w_{15} O_1 + w_{25} O_2 + w_{35} O_3 + \theta_5$

$= -0.3 + 0.1 \times 0 + 0.2 \times 1 + 0.2$

$= 0.1$

$O_5 = \frac{1}{1+e^{-0.1}} = 0.525$

$I_6 = w_{46} O_4 + w_{56} O_5 + \theta_6$

$= -0.3 \times 0.332 - 0.2 \times 0.525 + 0.1$

$= -0.105$ (rounded off)

$O_6 = \frac{1}{1+e^{0.105}} = 0.474$

$Err_6 = O_6 (1 - O_6)(T_6 - O_6)$

$= 0.474 (1 - 0.474)(1 - 0.474) = 0.1311$

$Err_5 = O_5 (1 - O_5)(Err_6 \, w_{56})$

$= 0.525 (1 - 0.525)(0.1311)(-0.2) = -0.0065$

$Err_4 = O_4 (1 - O_4)(Err_6 \, w_{46})$

$= 0.332 (1 - 0.332)(0.1311 \times (-0.3)) = -0.0087$

$\Delta W_{14} = (l)\, Err_4 O_4$
$= 0.9 \times (-0.0087)$
$W_{ij} = W_{ij} + \Delta W_{ij}$
$= 0.2 + 0.0078$
$\Delta \Theta_4 = (l) Err_4$
$\Theta_4 = \Theta_4 + \Delta \Theta_4$
$= -0.4 + \ldots$

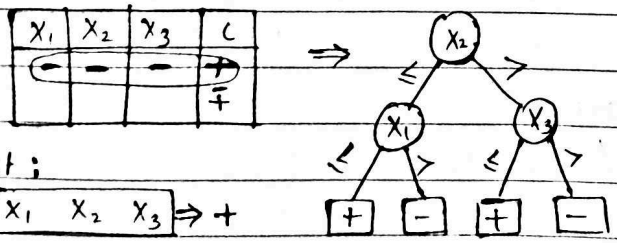
d,
, weight 1

ole in predic
e

* KNN Classifier

K Nearest Neighbour — Prepare model → test it on data → too much time in training

Classifiers
- Lazy learner/classifier (Bayesian, Decision Tree)
- Eager learner/classifier
  - reduced training time
  - Construc" of training model is not done until you have the test sample (postponed)



Objective: training sample must map on the model

Test:
$[X_1\ X_2\ X_3] \Rightarrow +$

(They try to find a sample matching the test sample in training sample then return that class (Comparison algo fast)

Drawback: If test sample doesn't match any training sample

Sol": Instead of finding actual match, find the

$\Delta W_{14} = (l)\, Err_4\, O_1$
$= 0.9 \times (-0.0087) \times (1) = 0.0078$

$W_{ij} = W_{ij} + \Delta W_{ij}$
$= 0.2 + 0.0078$

$\Delta \theta_4 = (l) Err_4$
$\theta_4 = \theta_4 + \Delta \theta_4$
$= -0.4 + \ldots$

* KNN Classifier

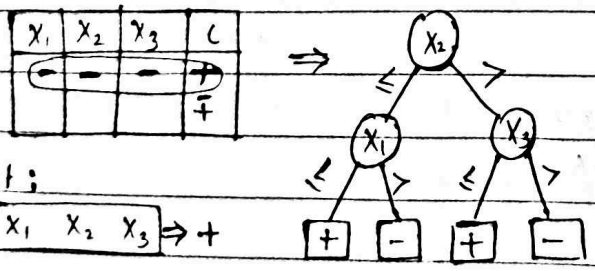
(K Nearest Neighbour — Prepare model → test it on data → too much time in training

Classifiers: Lazy learner/classifier (Bayesian, Decision Tree)

Eager learner/classifier

reduced training time

Construc" of training model is not done until you have the test sample (postponed)



Objective: training sample must map on the model

Test:

X_1 X_2 X_3 ⇒ +

(They try to find a sample matching the test sample in training sample then return that class (Comparison algo fast)

Drawback: If test sample doesn't match any training sample

Sol": Instead of finding actual match, find the

closest training sample w/ test sample

Test Similarity (sim measure as per type of data)

$T_1 \rightarrow 0.1 \rightarrow +$ $\quad K=3$ (3 nearest neighbours)
$T_2 \rightarrow 0.2 \rightarrow -$ 3 closest neighbours considered
$T_3 \rightarrow 0.3 \rightarrow +$
$T_4 \rightarrow 0.4$
$T_5 \rightarrow 0.5$

predict based on no. of nbrs you select

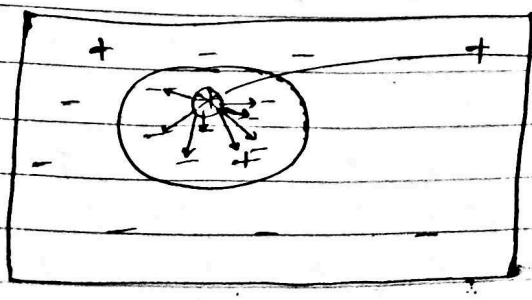
$T_1 \rightarrow +$
$T_2 \rightarrow -$ } majority voting $\Rightarrow +$
$T_3 \rightarrow +$

↳ class label of the training set

$K=2$ $\quad \rightarrow +$ } + or − anyone (or any tie breaking cond" if you define)
$\rightarrow -$ }

Problem: If you don't select K value wisely, its performance won't be good
K v. small : overfitting
K v. large : misclassification or underfitting (all samples r considered)

So many variants to select K optimally



Test sample
we follow majority, but here see many samples in circle are v. far & only some at close distance

Sometimes weight is defined,

$$w = \frac{1}{d(x, x')^2}$$ If distance is ↓, weight ↑

so closer sample is playing bigger role in predic"

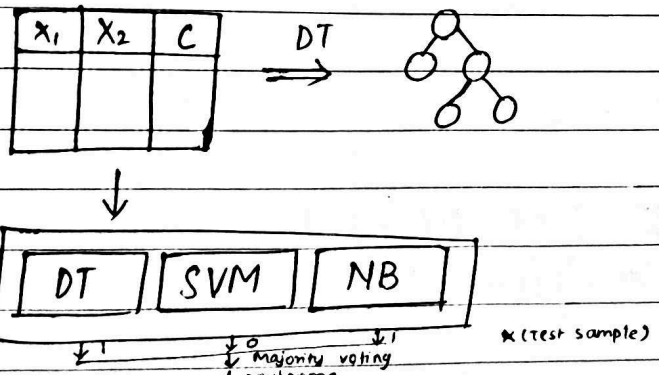
Majority voting → all samples playing equal role

$$= w \cdot \underbrace{1(x = y')}_{\text{output}}$$ → 1 or 0 predic"

Multiply w/ weight then predict

* Ensemble classifier

Instead of using a single classifier, we can use diff. classifier on the same layer

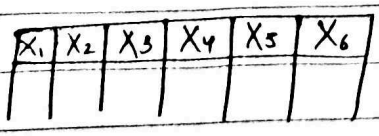


Ensembling: mixing diff. classifiers

Ensemble method gives good predic"s as compared to indiv. classifiers

Approaches to do Ensembling

1. Feature level ensembling
2. Data " "
3. Algorithm " "

1. 

Finding a subset of features used in 1 classifier

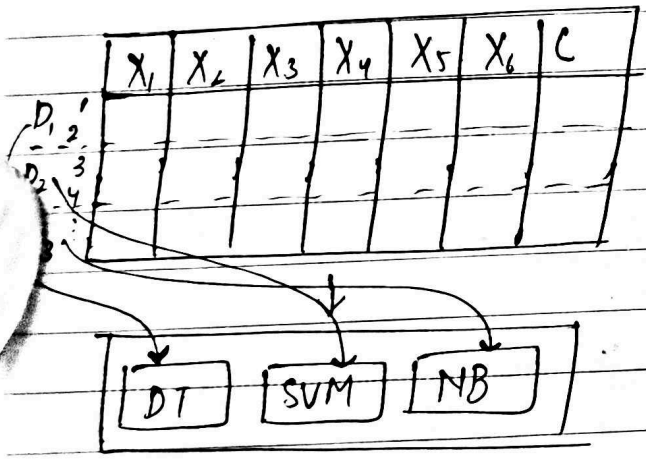
{X1 ... X6} → {X1 X2 X3} {X2 X3 X6} {X4 X5 X6}

↕ DT ↓ SVM ↓ NB

Take care what kind of data is good for a certain type of classifier:
DT good if discrete attributes
NB " " conditionally independent attributes

To get result:
1) Majority Voting
2) Weighted aggregation (wt. to features/ classifier)

② Data level Ensembling

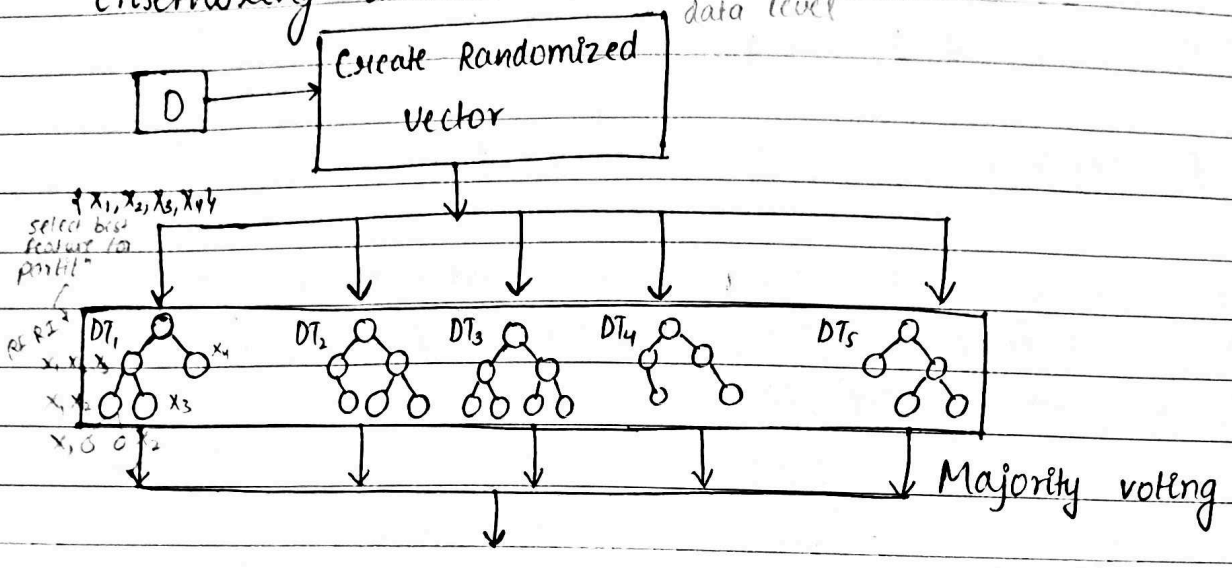


③ Algorithm level
DT SVM NB, DT SVM KNN, DT SVM Rule based
Diff combinations of classifiers

*Disadv of Ensembling: Complexity (Lot of classifiers, redundancy of features in feature level) for some app" it may not be affordable
Challenge → Lightweight ensembling method

1. Random Forest Classifier

Ensembling method



RF - RI^{input} (Random Forest with Random Input)

RF - RC^{comb^n} (" " " " Combination)

* Characteristics of Random Forest

Probability Distribution func" is used for all classifiers to create the Randomized vector

They use multiple decision trees as classifiers

Data & algo level ensembling

feature	class
X	Y
0.2	1
0.3	1
0.1	-1
0.4	1
0.6	1
0.5	-1
0.7	1
1	-1

⇓

sampling w/ replacement (method to create randomized vector)

X	Y	
0.2	1	→ D_1 used to train
0.2	1	DT_1 (have
0.3	1	their own
0.4	1	depth & no. of
0.6	1	nodes)
0.6	1	
0.7	1	
0.2	1	

$D_2 \rightarrow DT_2$ $D_3 \rightarrow DT_3$

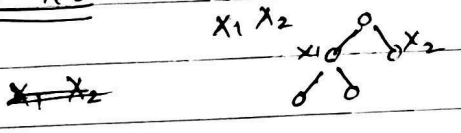
⇒ RF-RI

If dataset is large & has sufficient features → cond^n
Instead of looking for best partition in DT, they use random partition $\{X_1\ X_2\ X_3\ X_4\} \rightarrow \{X_1, X_2, X_3\}$ or $\{X_4\}$

Objective: Randomization w/ feature & data

If dataset has ↓ features → many features are common over diff. decision trees → performance ↓ → use RF-RC to avoid this

⇒ RF-RC



create new features using linear comb^n of features
$X_3 = C_1 X_1 + C_2 X_2$
use them to create diff depth DTs
avoids redundancy of features → helps in randomization

* Association Analysis/Pattern Mining

2, 3, 5, 9, 10, 12, 3, 5, (10, 13, 15) 18, 19, (10, 13, 15) 21, 20, (10, 13, 15)

Recommender system uses pattern mining to give sugges^n. Eg- Buy shirt → can buy pant

Transac^n: visit of customer in market & the item he purchased
Market-Basket Analysis: Record of items purchased & analysing it

Terminologies

1) item : any product customer purchases in mkt visit (Beer, Nuts etc.)
2) item set : {Beer} {Beer, Diaper} {Beer, Diaper, egg} (k items in a set)
   1-itemset, 2-itemset, 3-itemset, k-itemset
3) support count : No. of transac^n where an itemset is present
4) support
5) confidence
6) frequent itemset

S.C : no of transac^n supporting an itemset & what is its count

{Beer, Diaper} SC = 3

↳ it is a pattern → it appears in different transac^{ns}

↳ patterns in data are represented by an association rule

Beer, Nuts, Diaper
Beer, Coffee, Diaper
Beer, Diaper, Egg
Nuts, Eggs, Milk
Nuts, Coffee, Diaper, Eggs, Milk

↓

Beer ⟶ Diaper (or Diaper → Beer)
Nuts ⟶ Milk

Some patterns may be useful for the app^n some not

⇒ Measures to judge usefulness of an association rule — support, confidence

$$\text{Support} = \frac{\sigma(X \cap Y)}{N}$$

$\sigma(X \cap Y)$ — support count of $X \cap Y$

N — total transactions

Out of total transac^n in dataset how many have $X \cap Y$ (in a group, how many in favour)

$$\text{support}(\text{Beer} \rightarrow \text{Diaper}) = \frac{3}{5} = 60\%$$

$$\text{Confidence} = \frac{\sigma(X \cap Y)}{\sigma(X)}$$

$\sigma(X \cap Y)$ — support ct of $X \cap Y$

$\sigma(X)$ — support ct of X

Where Beer is present, diaper is also present

$$= \frac{3}{3} = 100\%$$

out of those supporting what is the level of confidence, how many follow

itemset is said to be frequent if its support > min defined support threshold

{Beer, Diaper} → 60% >= Min support (of app")
e.g- 40%

Objective: Identify frequent itemsets → make business decisions accordingly

{Beer, Diaper, Milk} → so many other itemsets can be formulated
sub patterns → sub itemsets
{Beer, Diaper} {diaper, milk} ....

Transform to binary dataset ⇒
6 features →
Max^m length pattern is 6
(~~freq~~ itemset w/ longest length)
{Beer, Nuts ... Milk}
$2^6 - 1$ subsets / subpatterns

·	x_1 Beer	x_2 Nut	x_3 Diaper	x_4 Coffee	x_5 Egg	x_6 Milk
1	1	1	1	0	0	0
2	1	0	1	1	0	0
3	1	0	1	0	1	0
4	0	1	0	0	1	1
5	0	1	0	1	1	1

100 features → so many subpatterns possible → checking sup & conf. for all is difficult. Solutions:

Closed itemset / Closed pattern

min support = 2
{Beer, diaper} ⇒ Support count = 3 (freq. ✓)
{Beer, diaper, Milk} ↓ s.c = 0 ≠ 3
1) frequent
2) no superset has same support

Max-pattern

1) itemset frequent
2) no ~~superpatt~~ frequent superpattern / superset

Max^m frequent itemset (supcount max^m)

Downward closure property

Eg-

Tid	items
1	{a,b,d,e}
2	{b,c,d}
3	{a,b,d,e}
4	{a,c,d,e}
5	{b,c,d,e}
6	{b,d,e}
7	{c,d}
8	{a,b,c}
9	{a,d,e}
10	{b,d}

(a) Find frequent itemset with min. support = 2

(b) Find all association rules belonging to the frequent itemsets you obtained in a

(c) Use confidence based pruning to remove all association rules having threshold <80%

(d) Also draw a lattics based structure to prune the association rules as per confidence threshold of 30%

(e) Draw a hash-Tree structure for candidate 3-itemsets you obtained in Apriori generation step of (a) using the defined hash func^n H(P) = h(P) mod 3

A - (a)

{a}	5	{a,b} ✓	3	{a,b,c}	1	C_3	
{b}	7	{a,c}	2	{a,b,d} ✓	2	~~{c,d,e}~~ ⇒ {a,b,d,e}	2
{c}	5	{a,d} ✓	4	{a,b,e}	2	{a,b,d} {a,b,c,d}	0
{d}	9	{a,e} ✓	4	{a,b,c,d}		{a,b,e} ⇒ {b,c,d,e}	1 ⇒ {a,b,d,e}
{e}	6	{b,c}	3	{a,b,c,e}		{a,d,e}	
		{b,d} ✓	6	{a,b,d,e}		{b,c,d}	
		{b,e} ✓	4	{a,c,d}	1	{b,d,e}	
		{c,d} ✓	4	{a,c,e}	1		
		{c,e} ✓	2	{a,c,d,e}	4		
		{d,e} ✓	6	{a,d,e}	4		
				{b,c,d}	2		
				{b,c,e}	1		
				{b,c,d,e}			
				{b,d,e}	4		

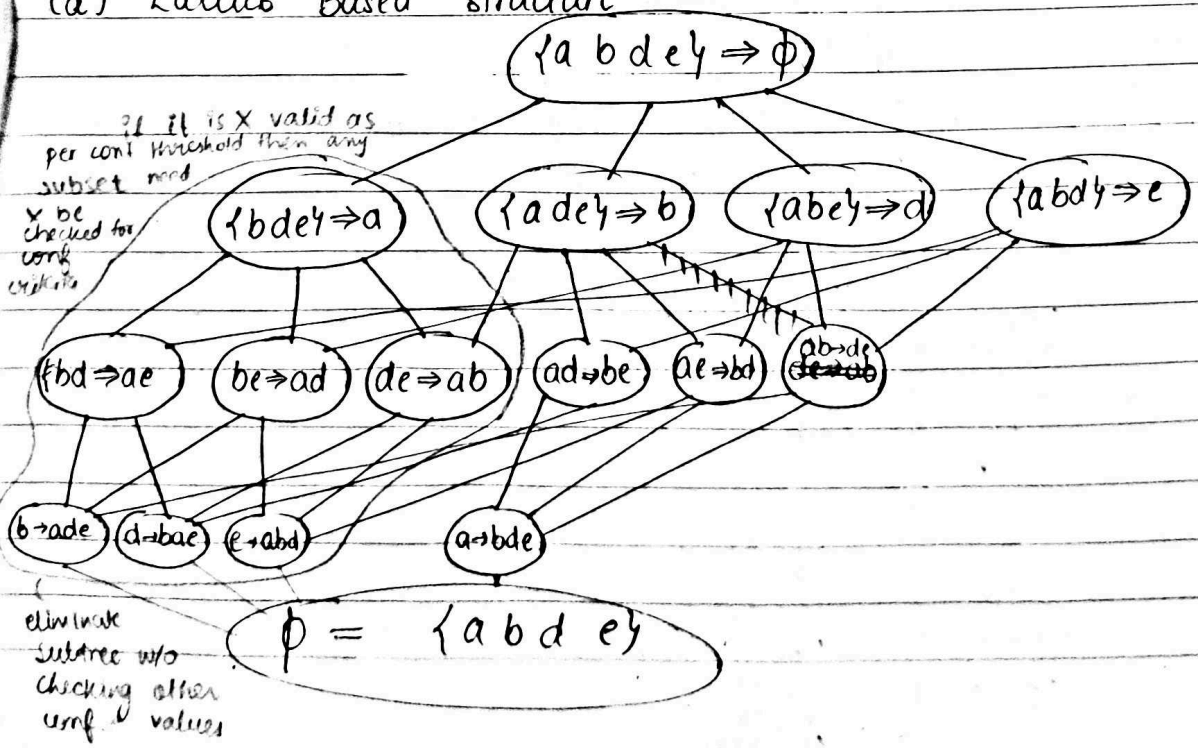
(c) {a, b, d, e}

length K = 4 $2^4 - 2 = 2^4 - 2 = 14$ rules

$\frac{\{abde\}}{\{ae\}} = \frac{2}{5} = 0.4$ ✓

a → bde	$\frac{2}{7} = 0.28$	
b → ade	$\frac{2}{9} = 0.22$	
d → abe	$\frac{2}{4} = 0.33$	✓
e → abd	$\frac{2}{3} = 0.67$	✓
ab → de	$\frac{2}{4} = 0.5$	✓
ad → be	$\frac{2}{4} = 0.5$	✓
ae → bd	$\frac{2}{6} = 0.33$	✓
bd → ae	$\frac{2}{4} = 0.5$	✓
be → ad	$\frac{2}{6} = 0.33$	✓
de → ab	$\frac{2}{2} = 1$	✓
abd → e	$\frac{2}{2} = 1$	✓
abe → d	$\frac{2}{4} = 0.5$	✓
bde → a	$\frac{2}{4} = 0.5$	✓
ade → b		

(d) Lattice Based structure



bde → a conf(R) = 0.5 = 60% > 30% ✓

If conf = 75% X (suppose)

$X \to Y - X$

$X' \subseteq X$

$X' \Rightarrow Y - X'$

} Prune subsets/subtree of bde → a

Lattice based procedure for pruning:

adc ⇒ b 0.5 ✓ abe → d ✓ abd → e ✓

bd → ae 0.33 ✓ ...

(e) rem = 1 1,4,7 rem = 1 2,5,8 3,6,9 rem = 0 divide as per remainder of mod

Raise major? candidate 3-itemsets

(1,2,3)
(4,8,2)
(1,5,3)
(2,5,6)
(2,3,6)
(2,4,5)
(5,2,6)
(3,8,9)
(1,5,6)
(2,4,6)
(2,3,5)
(1,3,8)
(2,9,5)

sorted order list?

(1,2,3) hash on 1st item 1%3=1 → left of HT

(4,8,2) 4%3=1 → left

hash on 2nd item

(1,2,3)
(4,8,2)
(1,5,3)
(5,2,6)
(1,5,6)
(1,3,8)

(2,5,6)
(5,2,6)
(2,4,6)
(2,9,5)
(2,3,5)
(2,4,5)
(2,3,6)

(3,2,6)
(3,8,9)
(3,2,6)
3,8,9

(1,2,3) (4,8,2) (1,5,3) (1,3,8) (2,4,6) (2,4,5) (2,5,6) (2,3,5) (2,9,5) (2,3,6)

Transaction How to check which buckets to check for transac?

1,2,3,5,6

1+ [2,3,5,6] 2 [3 5 6] 3 [5 6]

12 [3 5 6] 23 [5 6]

13 [5 6] 25 [6]

15 [6]

(1,2,3) (4,8,2) (1,5,3) (1,5,6) 1,3,8

2,4,6 2,4,5 2,5,6 2,3,5 2,9,5 2,3,6

3,8,9 3,8,9

Issues in Apriori algo:

improve w/ hash tree → ↓ no of comparisons

Trans id	Items
T_1	{M, O, N, K, E, Y}
T_2	{D, O, N, K, E, Y}
T_3	{M, A, K, E}
T_4	{M, U, C, K, Y}
T_5	{C, O, O, K, I, E}

Eg -

Find all freq itemsets belonging to this dataset

support = 3 (threshold)

Item	Count
{M}	3
{O}	3
{N}	2
{K}	5
{E}	4
{Y}	3
{D}	1
{A}	1
{U}	1
{C}	2
{I}	1

⇒ C_1: {M}, {O}, {K}, {E}, {Y}.

2-itemset

2-itemset	Count
{M, O}	1
{M, K}	3
{M, E}	2
{M, Y}	2
{O, K}	3
{O, E}	3
{O, Y}	2
{K, E}	4
{K, Y}	3
{E, Y}	2

⇒ C_2: {M, K}, {O, K}, {O, E}, {K, E}, {K, Y}

3-itemset	Count
{M, O, K}	1
{M, K, E}	2
{M, K, Y}	2
{O, K, E}	3
{O, K, Y}	2
{K, E, Y}	2

⇒ {O, K, E}

* FP Growth (Frequent Pattern Growth) (algorithm)

Trans id	items
T_1	I_1, I_2, I_5
T_2	I_2, I_4
T_3	I_2, I_3
T_4	I_1, I_2, I_4
T_5	I_1, I_3
T_6	I_2, I_3
T_7	I_1, I_3
T_8	I_1, I_2, I_3, I_5
T_9	I_1, I_2, I_3

minsup = 2

Partition into multiple subsets

for every subset we find the ... computing sup. values again & again

① Find freq. 1-itemsets

1-itemset	sup ct
{I_1}	6
{I_2}	7
{I_3}	6
{I_4}	2
{I_5}	2

② sort in desc

L-order list

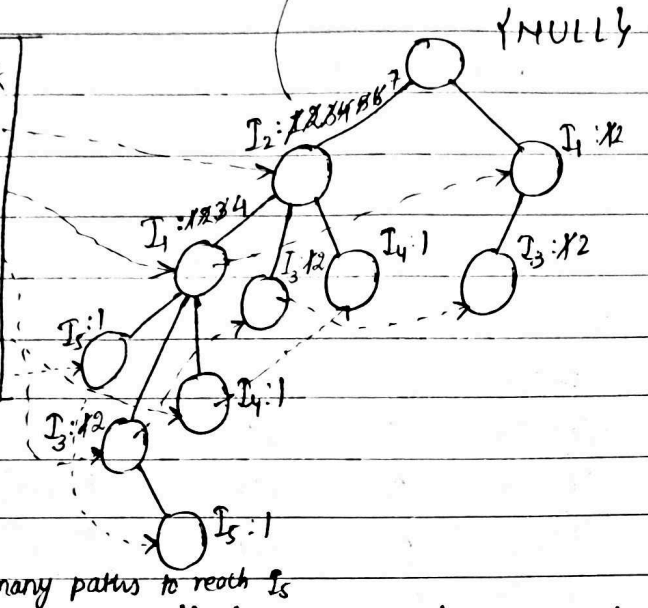
{I_2}	7
{I_1}	6
{I_3}	6
{I_4}	2
{I_5}	2

If Qn asks freq. 2-itemsets then stop Apriori algo there

Every transac^n in the dataset is considered as a branch in the FP tree. $T_1 \rightarrow I_1\ I_2\ I_5$ → as per L-order → $I_2\ I_1\ I_5$

comes 1st time in this tree so its support is 1

items	support	link
{I_2}	7	-
{I_1}	6	-
{I_3}	6	-
{I_4}	2	-
{I_5}	2	-



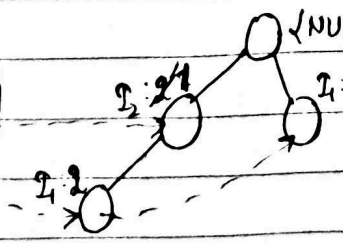
how many paths to reach I_5

items | support | link ⇒ Start w/ bottom of the list (min sup value)

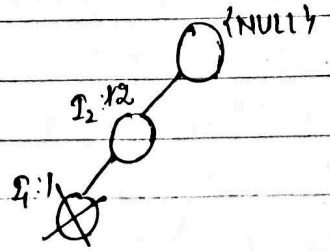
items	conditional pattern base	conditional FP-tree	Frequent Patterns
{I_5}	{I_2, I_1: 1}, {I_2, I_3: 1} (sup value of I_5)	{I_2: 2, I_1: 2}	{I_2, I_5: 2} concatenate I_5 item w/ those in cond^{nal} FP tree
			{I_1, I_5: 2}
			*{I_2, I_1, I_5: 2} ✓ freq 3 itemset
{I_4}	{I_2, I_1: 1}	{I_2: 2}	{I_2, I_4: 2}
	{I_2: 1} (I_4 value)		
* {I_3}	{I_2, I_1: 2}	{I_2: 4, I_1: 2}	{I_2, I_3: 4
	{I_2, I_1: 2}	{I_1: 2}	{I_1, I_3: 4 (I_3 support 4 in the table)
	{I_1: 2}		*{I_2, I_1, I_3: 2 (min (I_1, I_2) in the path min(4,2)=2) ✓

{I_3}

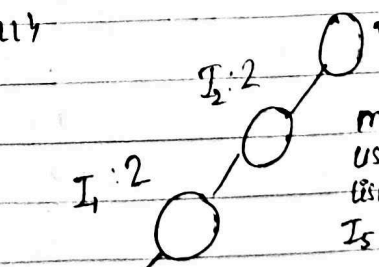
item	supp	link
I_2	4	-
I_1	4	-



{I_4}



{I_5}



make FP tree using L-order list again for I_5

remove infreq. one (minsup = 2)

	corr. to I_1 support in the tree path		
$\{I_1\}$	$\{I_2 : 4\}$	$\{I_2 : 4\}$	$\{I_2, I_1 : 4\}$
$\{I_2\}$			

Building FP tree ↓ complex than scanning whole dataset

* Strong Association Rule

N = 10,000 (Transac^{m})
Computer Games = 6000
Video = 7500
Computer Games and video = 4000.

Asso^{n} rules are representation of freq. itemsets
$\{X_1, X_2, X_3\}$
$X_1 \rightarrow X_2\ X_3$, $X_2 \rightarrow X_1\ X_3$

Computer Games → Video support = 40% Confidence = 66%

support = $\frac{\text{count}\{\text{computer} \cup \text{video}\}}{10000} = \frac{4000}{10000} = 40\%$ ✓

conf = $\frac{\text{count}\{\text{comp} \cup \text{video}\}}{6000} = \frac{4000}{6000} = 66\%$ ✓

satisfy both ↓ ∴ this rule is useful

x suffi for all app^{n} ↓ evaluate asso^{n} value as well

Association value

Probab. of video = $\frac{7500}{10000} = 75\%$

If cust. only purchase video prob. is 75%. If comp + video, conf. value is 66%. → some ↓ in %age. Negative asso^{n} in this rule of video w/ computer game

$(A \cdot B)$ conf $= \frac{P(A \cup B)}{P(A)}$

+ve asso^n → Strong assoc^n rule

lift value lift(A, B)

$$\text{lift}(A,B) = \frac{P(A \cup B)}{P(A)\,P(B)}$$

= 1 (then A, B are independent)
= < 1 (Negative associa^n)
= > 1 (Positive)

If A & B is indep. then $P(A) \times P(B) = P(A \cup B)$ or $A \cap B$?

Numerical measure of how much -ve or +ve association is there b/w 2 itemsets A & B

If lift(A,B) = 0.4 → this amt (Negative asso^n) of lifting req. to make it 1

Chi square test (Hypothesis is considered that the 2 itemsets are indep. if rejected → some asso^n is there

① Contingency table of observed values

	Computer Games	Comp. games	
Video	4000 e_{11}	3500 e_{12}	7500
Video	2000 e_{21}	500 e_{22}	2500
	6000	4000	10000

② Expected value

$e_{11} = \frac{6000 \times 7500}{10000} = 4500$

$e_{12} = \frac{4000 \times 7500}{10000} = 3000$

$e_{21} = \frac{6000 \times 2500}{10000} = 1500$

$e_{22} = \frac{4000 \times 2500}{10000} = 1000$

③ $\chi^2 = \frac{(\text{observed} - \text{expected})^2}{\text{Expected}}$

= 507 (suppose)

signif = (2-1)(2-1) (2×2 contingency table)

look in table → ?

+ve, -ve asso^n find → see how?

1 can be asso^d 1 cannot

Club diff items so that customer can buy them.

Eg- DVD is purchased along w/ Computer games or videos

N = 10k
comp. games = 6000
Video = 7500
Comp games & video = 4000
DVD = 2500
Comp games & DVD = 2000
Video & DVD = 3500
Comp games ⟶ Video
Comp games ⟶ DVD
Video ⟶ DVD
sup = 40% conf = 66%

Ans- ① $sup = \frac{\{C.G \cup DVD\}}{N} = \frac{2000}{10000} = 20\%$

$sup = \frac{\{V \cup DVD\}}{N} = \frac{3500}{10000} = 35\%$

② $conf = \frac{\{C.G \cup DVD\}}{6000} = \frac{2000}{6000} = 33\%$

$conf = \frac{\{C.G \cup V \cup DVD\}}{7500} = \frac{3500}{7500} = 46.66\%$

Both X satisfy
1 -ve 1 +ve → go w/ +ve
Both +ve → go w/ higher one } calc. lift

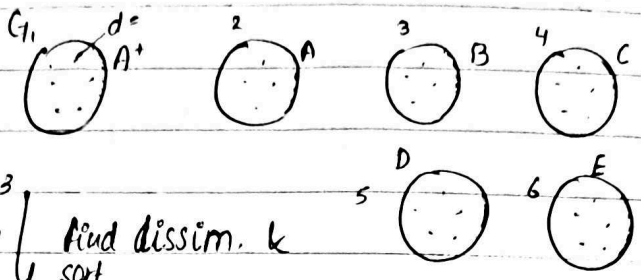
If they satisfy sup & conf criteria then calc. lift value to check +ve / -ve assoc" & choose b/w them

Dimensionality reduc" → Feature subset selec"
↘ Feature extraction (can use clustering)

$(X_1\ X_2\ X_3)$ $(X_4\ X_5\ X_6)$ Best group of features that can be clubbed
X_1 X_2

* Clustering

Manual labelling



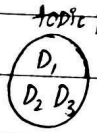
X feasible w/ large dataset

G_{13}
G_{14}
G_{15}
G_{16}
} find dissim. & sort

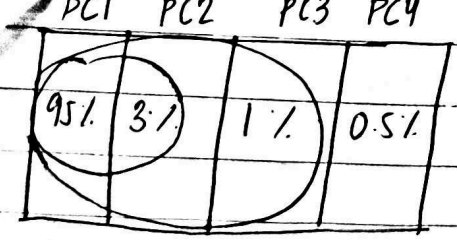
Objective: Finding group of data object such that data objects in same group are very similar & those in other groups are very dissimilar

Unsupervised method

Clustering → automatic classification



pal components can be grouped as clusters



Association analysis

	I_1	I_2	I_3	I_4	I_5
1					
2					
⋮					
9					

I_2 I_1 I_5 freq. itemsets

all instances having these items can be grouped

Compression → PCA lossy data compression

redundant attributes removed → lossless compression

KNN → Identify dist of 1 sample w/ other samples
better handled by clustering

Outlier detec" → sample very dissimilar w/ other groups

Intracluster var" sum over all clusters → should be minimal

Intracluster var" min, Inter cluster max"

Eg- Network data analysis, Credit card fraud detec"

M1: (1,3) (2,5) (4,6) } Intra cluster similarity = sum of similarity bw items of all clusters

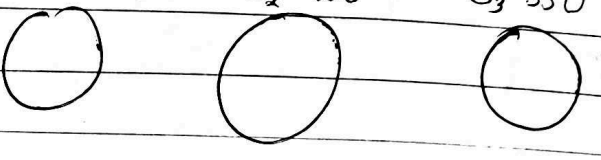
M2: (1,2) (3,4) (5,6)

	I_1	I_2	I_3	I_4	I_5
1					
2					
:					
1000					

(no. of clust.) (no. of instances)
k << N

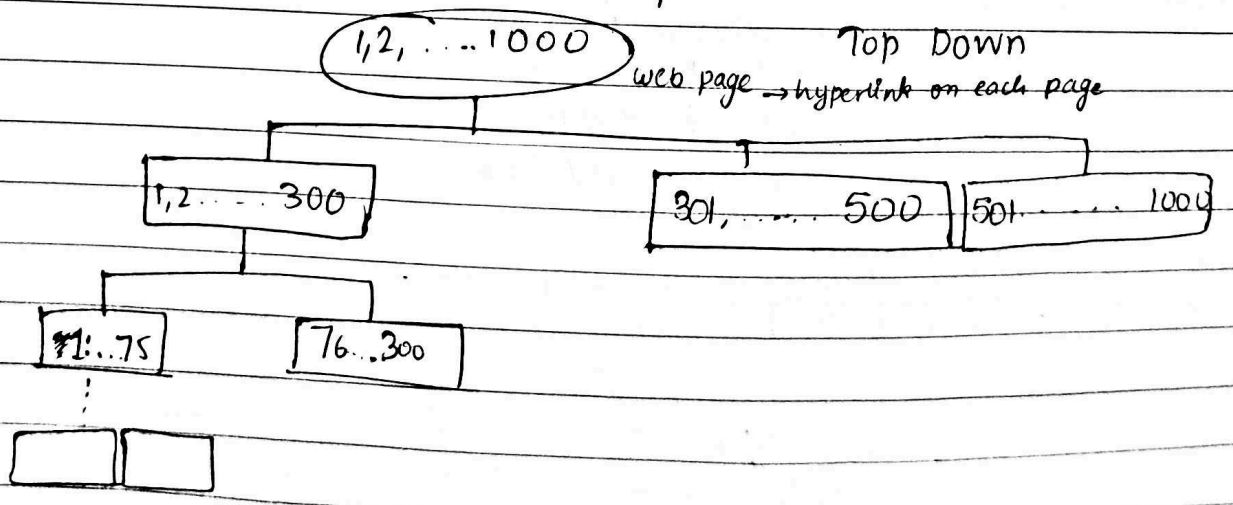
k <= 3 (3 clusters)

C_1 = 250 C_2 = 400 C_3 = 350

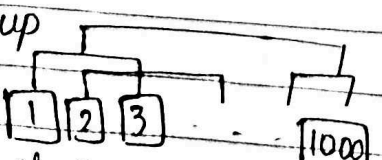


Partitioning approach:
How you partition data into various clusters

Hierarchical approach: ← Bottom up
← Top down



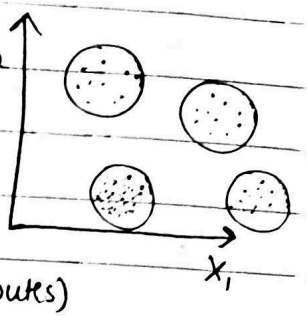
Bottom up



Similar → make hiera.

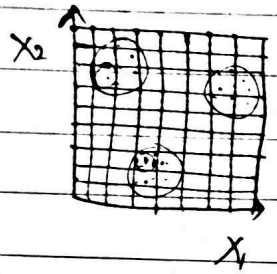
Density-based approach

No of data points within a region

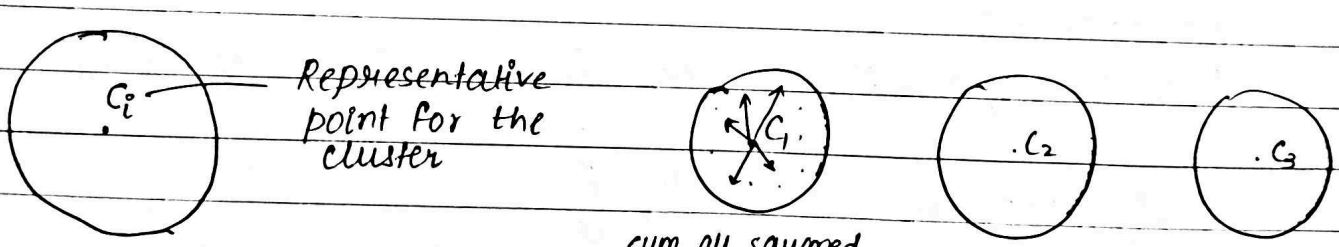


(X_1, X_2 attributes)

Grid based approach



respective samples in cells → aggregate func^{ns} defined



sum all squared distances (within cluster var^n)

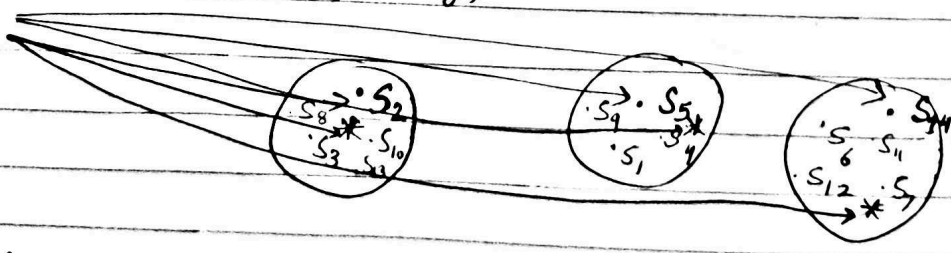
↳ Good clustering algo should have min value

① Randomly select 3 repres. samples

Assume all are numeric samples

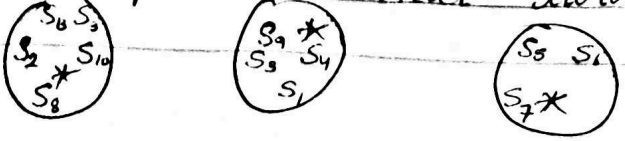
	X_1	X_2	X_3
1			
2	-	-	-
3	-	-	-
⋮			
20	-	-	-

② Distance of this pt wrt S_i → select as per min distance the cluster it belongs to (Euclidean distance / Similarity)



K = 3

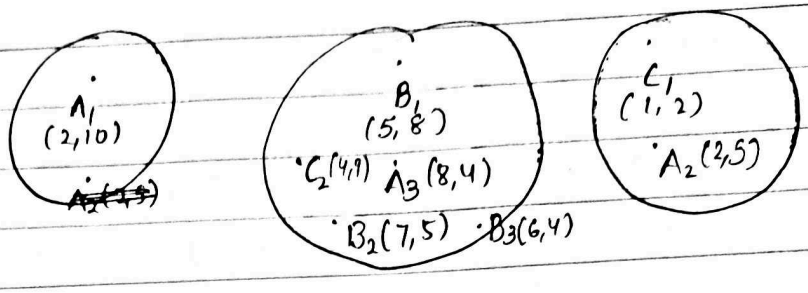
③ Update the representative sample → Mean value Considered as rep sample in next iteration



X Y

Eg- $A_1(2,10)$
$A_2(2,5)$
$A_3(8,4)$
$B_1(5,8)$
$B_2(7,5)$
$B_3(6,4)$
$C_1(1,2)$
$C_2(4,9)$

$K=3$

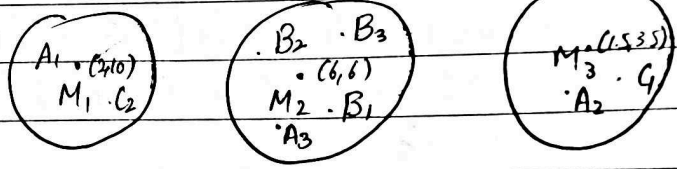


$d(A_2, A_1) = \sqrt{5^2} = 5$
$d(A_2, B_1) = \sqrt{9+9} = 3\sqrt{2}$
$d(A_2, C_1) = \sqrt{1+9} = \sqrt{10}$

Centroids

$M_2 = (6,6)$ $M_3 = (1.5, 3.5)$
$M_1 = (2,10)$

$d(A_3, A_1) = \sqrt{36+36} = 6\sqrt{2}$
$d(A_3, B_1) = \sqrt{9+16} = 5$
$d(A_3, C_1) = \sqrt{49+4} = \sqrt{53}$



$d(B_2, A_1) = \sqrt{25+25} = 5\sqrt{2}$
$d(B_2, B_1) = \sqrt{4+9} = \sqrt{13}$

$d(A_2, M_2) = \sqrt{16+1} = \sqrt{17}$
$d(A_2, M_3) = \sqrt{0.5^2 + 1.5^2} = \sqrt{\frac{1}{4} + \frac{9}{4}} = \sqrt{5}$

$d(A_3, M_2) = \sqrt{4+4} = \sqrt{8}$

Objective: $E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - c_i)^2$ ← point

Eg-X: 1, 2, 4, 6, 7, 8, 12, 28

1 feature → 8 samples

$k = 3$

{1, 2, 4}	{6, 7, 8}	{12, 28}	2 clusters sets
{1, 2, 4, 6}	{7, 8}	{12, 28}	
(3.25)	(7.5)	(20)	

$E_1 = (1-2.33)^2 + (2-2.33)^2 + (4-2.33)^2 + (6-7)^2 + (7-7)^2 + (8-7)^2 + (8)^2 + (8)^2 =$

$E_2 =$

C1 → Mean $\frac{7}{3} = 2.33$ $\frac{6+7+8}{3} = 7$ $\frac{12+28}{2} = 20$

$E_1 = 1.76 + 0.1089 + 1.67 + 1 + 1 + 128 = 133.53$ → total error

$E_2 = (3.25-1)^2 + (3.25-2)^2 + (4-3.25)^2 + (6-3.25)^2 + (7-7.5)^2 + (8-7.5)^2 + 128$

$= 5.0625 + 1.56 + 0.5625 + 7.5625 + 0.25 + 0.25 + 128$

$= 143.25$

$E_1 < E_2$ so better

Drawbacks:

① K Means clustering works fine for numerical data, not nominal attributes

Mean X defined for nominal attribute → variation is K-Mode clustering

② Very sensitive towards outliers (due to mean)