7 Jan 2024
Tuesday

Lecture - 1 :-

Data Analytics / Mining :-

Analysis + Algorithm
↳ which enables us to do analysis

* 3 components associated with it :-
  1) Association mining
  2) Classification
  3) clustering

1) Apriori algo
   ↳ then its several variation.
2) Frequent pattern (FP) growth algorithm
3) observations :- statistical add ons, metric sufficiency, clustering of ARs (Association rules)

1) Decision tree and its variations
2) Bayesian
3) Back propagation
4) SVM
5) Linear ~~re~~ regression

1) partition based
2) Hierarchical based
3) Density based

1) single, average and complete linkage algo.
2) BIRCH

1) CURE
2) DBSCAN
3) BFR

k-mean, PAM, clara, clarans.
↓
variations of k-mean & PAM, to enable larger data analysis

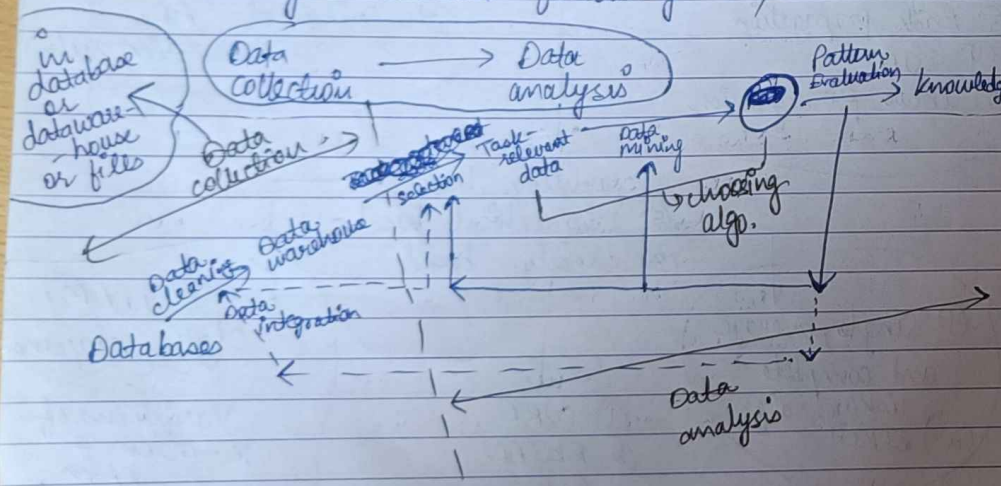
WOMK 30 DEC

① Introduction
↳ ① architecture
② DA engineering
③ steps
④ functionalities
⑤ Issues

(*) we are in era of data. we have data of different types: audio, vedio, text, numerical etc.
↳ How to handle and analyse such big data ??

Data Warehouse ?

Data analysis involves following steps :-



* Database and warehouse are two different things.

WOMK 30 DEC

* Data Mining — KDD process
↳ the core of knowledge discovery process.

→ If ek company ka database bnana h, then what u will ask from company :-

① what applications are req. for your company

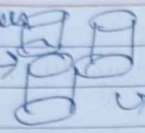
so,
database → application oriented data collection.
↳ so wo data to jo application mein help aaye.
eg. our college ki applications are :-
library, erp etc.

like college does not need our siblings details, as they are not useful for that application to be executed.

whereas
Data warehouse (time-variant) is subject-oriented and is always historical data.
↳ means like we are not in warehouse as we are still in clg and not history, jab clg se chle gye, then we are entered in warehouse.
→ means data in warehouse is fixed and will never change like my CGPA after going out from clg.
→ consists data from that date jab se clg bna
→ Database stores current data.

WOMK

Database



↳ why 3 cylinders?

↳ as database changes its formats with time, like pehle SQL, then kuch aur, kabhi koi column add kr diya hoga, kabhi kuch update etc.

warehouse → collection of entire data

↓

jiske liye saare format of database ko integrate krna pdhega

↓

for that need some interface programming, to reformat things so that they can talk to each other.

↓

database of different format

Analysis

↳ 1st thing :- get / extract task-relevant data out of entire huge data

→ Now how to extract this data from big data ???

↓

with help of firing queries

↓

which will definitely be a complex query.

Now, task-relevant data mil gya, ab ML, Deep learning algo use krenge.

↳ Now to use which algo ??

which algo to select ??

↳ most difficult thing to understand (?)

WOMK

→ algo to be used will depend on requirement of customer and not of developer.

→ when we apply algo

↳ we get patterns

↳ mostly 1000s of pattern milte hain, use depends on algo.

↓

then pattern evaluation comes into picture, ~~kaha~~ to figure out which pattern is good for customer.

↓

mostly 1, 2 nikaalte, not like 50 - 100.

↓

for this matrixes are used, and then we get 'knowledge'

↳ yehi toh hmaara main target hai!

22/2/2023 Wednesday

lec-2 :-

① 1st category of state :-

(*) Association Rule Mining :-

↳ tracks your pattern of purchase and interprets what and how you are purchasing.

Say :- Bread and butter has strong association.

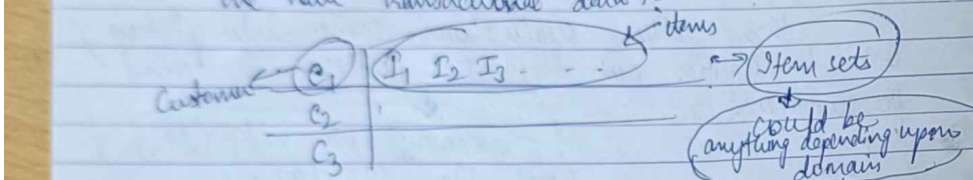
↳ How they come to know ?

↳ By looking into your basket and applying maths.

Say total 500 people, out of them 400 bought bread, (80%) and out of 400, 360 bought butter, so 90% confidence label based on purchase that bread and butter have strong association.

→ each customer is one transaction for retail shop.

we have Transactional data :-



↳ we do not need personal details of the customer.

→ Once transactional data hai, then association rule mining say, apply algo on this data and find patterns (also known as association rules).

like :- $X \rightarrow Y$.

X, Y maya be single item or multiple items.

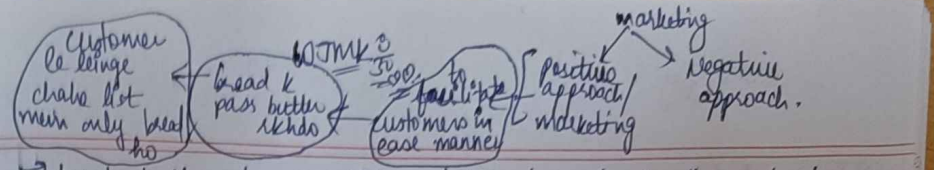
age (< 30) ∧ buy (laptop) → buy (earphone)

X Y

2 dimension rule
↳ age, buy

buy (bread) → buy (butter)
↳ single dimension

so, rule may be single dimension or multiple dimension.



↳ bread-butter ka confidence label obv. jyada than laptop-earphone, so confidence level varies with items, and yeh chlta hai,

agar confidence level of bread-butter = 90%
" " " " laptop-earphone = 20%,

then, is enough for retailers.

as, bread-butter jitna frequently koi buy krega, laptop-earphone nahi krega.

and when you go to buy laptop, tabi wo earphone bhi dikhate hain and your mind may change.

negative marketing approach
↳ butter and bread dono alag rkho,

say list has both, then bread leke, butter lene jaaoge, rste mein items dikhenge and you will start buying, chahe wo list mein nahi thi.

→ helps in inventory also.
or demand to manufacturer

association rules also (AR)
↳ also helps in managing warehouses. helps
↳ not only restricted to retail market, also comes in fraud detection (???)

Usually support and confidence of AR is user defined.

~~(1) 1st most important algo that comes in this category :-~~

· mostly retail market tells what is best and support and confidence for them or even analyst can also figure out at background because of knowledge he has.

WORK

(2) Types of AR :-

(1) Boolean / Quantitative ARs :-
↳ you look for absence or presence.

Bread → butter

age (X, "30 ... 39") & income (X, "42k ... 48k")
→ buys (X, projection TV)

There are 3 dimensions

(2) Single / multi-dimensional ARs.
(3) Single / multi-level ARs.

Bread → brown, white ... Butter → full cream butter ...

bread and butter bhi aage different types k hote hain, but not interested in that till now, but might be interested, then that comes under this type of ARs.
→ 3 may also be interested to know that how brown bread is associated with ~~bread~~ butter.

brown bread → butter (?)
Bread → full cream butter (?)

So, bread, butter are superset,

brown bread → butter ⇒ say (70%, 80%)
support confidence

WORK

⇒ even if (60%, 70%) hota, then it is also pattern for retailers.

Support and confidence :-

(1) A rule must have some minimum user-specified confidence and support.

(2) AR X ⇒ Y holds with support T, if T% of transactions in DB that support X also support Y.

so jab bhi confidence and support decide kate hain, then unko min." " " bolte hain, means agar uske neeche" " ", then we will reject.

I → set of all items
D → transactional database.

AR A ⇒ B has support 's' if 's' is %age of transactions in D that contain AUB (both A and B)

$$S(A \Rightarrow B) = P(A \cup B)$$ → bitne logon ne bread-butter khareeda.

AR A ⇒ B has confidence 'c' in D if c is % of transactions in D containing A that also contain B.

$$C(A \Rightarrow B) = P(B/A) = \frac{P(A \cup B)}{P(A)}$$

kitne logon ne bread k baad butter bhi khareeda.

Example:-

Transaction ID	purchased items
1	{1, 2, 3}
2	{1, 4}
3	{1, 3}
4	{2, 5, 6}

for min $s = 50\%$, min $c = 50\%$, what rules we should have.

A. Try:- $1 \rightarrow 2$

$s = 1/4$

$c = 1/3$

so not!

$1 \rightarrow 3$ ✓

$s = 2/4 = 50\%$

$c = 2/3 = 66.6\%$ so chlega ✓

(*) Algo for finding (FIs) (frequent itemsets):-

1. Apriori
2. sampling
3. partitioning
4. Hash based technique
5. Transaction reduction
6. etc.

Variation of apriori only, to solve some specific problem.

so, main algo → apriori hai.

Apriori algo:-

↳ has main 2 properties:-

(1) Apriority property.

(2) Anti-monotone property.

Say min $s = 50\%$

→ so item is said to be frequent, ~~then it~~ which appears in more than or equal to 50% of transaction.

↳ means it must satisfy min. support.

→ If {AB} is frequent itemset, both {A}, {B} should also be frequent itemset.

Anti monotone property

↳ If A set does not pass property, then its every superset will also not pass property.

⇒ So both properties are complementary to each other and both helps in analysis.

→ dusri property ketti again ~~for~~ chota set fail, big containing that chota, dekho hi nat.

Apriori algo example:-

Database (D)

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

say :- s = 50%

Step 1 list unique items
↳ 1, 2, 3, 4, 5

Step 2 :- Candidate set banao C_K (frequency wali bhi)

Scan D

Itemset	sup.
1	2
2	3
3	3
4	1 → this does not get qualified.
5	3

L_1 → frequent itemsets, large Itemsets.

Step 3 :-

Itemset	sup
1	2
2	3
3	3
5	3

Step 4 :- 4 * 4 (join kiya)

C_2 =

Itemset
1, 2
1, 3
1, 5
2, 3
2, 5
3, 5

Step 5.

C_2 Itemset	sup
1, 2	1 → kicked out
1, 3	2
1, 5	1 → kicked out
(2, 3)	2
(2, 5)	3
(3, 5)	2

Step 6

L_2 =

Itemset	sup
1, 3	2
2, 3	2
2, 5	3
3, 5	2

→ Then do $L_2 * L_2$

Step 7

C_3 =

Itemset
2, 3, 5

Scan D → L_3

Itemset	sup
2, 3, 5	2

Step 8.

only one combination left, so algo stops, otherwise C_4 ... aage jaati, till ek combo reh jaaye!

here L_3 pe rukey, so :-

$L = L_1 \cup L_2 \cup L_3$ (Large itemsets).

* Single item can not generate sufficient rules.

$1 \rightarrow 3$] → are different, (bilkul different!)
$3 \rightarrow 1$

↳ dono ka confidence will be different.

~~So, hmmm 1, 3 deletion~~ ismein also may go for $\{2,3\} \rightarrow 5$ or $2 \rightarrow \{3,5\}$

and many other.

WORK

Now, so confidence bhi dekho!
So want min c = 90%,
So jinka 'c' >= 90% we rkho, and baaki discard kro!!!!

→ But this algo would be very-very complex, if transaction data is very-very large!
What is TC=? of this algo.

→ In each iteration, we are scaning data!

say D = 50 GB
and RAM = 8 GB.
↓
then each time, 7 blocks u have to retrieve,
so 7 times input-output, swap-in ⇄ swap out!

⇒ agar D ↑, I/O cost ↑.
that's why variations in apriori algo has been proposed.

Ex:- L = {2,3,5}
{2,3}, {2,5}, {3,5}, {2}, {3}, {5}

↳ RR are:-
2,3 ⇒ 5 → 100%
2,5 ⇒ 3 → 66%
3,5 ⇒ 2 → 100%
2 ⇒ 3,5 → 100%
3 ⇒ 2,5 → ??

WORK

WORK

28/1/25
TUESDAY

lec-3:-

When we talk about database or datawarehouse, the task-relevant data (TRD) is very important, not in files, bcoz. files can be simply called in the files.

→ and getting TRD is with help of complex queries which definitely must satisfy all constraints, as our analysis requires only specific data.

STEPS of a KDD process:-
(1) Learn the application domain:
— relevant prior knowledge and goals of the application
— Databases are actually created on the basis of application, bhut vaha reformat krna pdta jaata hai data, clean krke data ko.
(2) Creating a target data set
(3) Data cleaning and preprocessing
(4) Data reduction and transformation
(5) choosing fns of data mining
(6) choosing mining algo(s)
(7) Data mining
(8) Pattern ~~visuali~~ evaluation and knowledge presentation.
(9) Use of discovered knowledge.

very critical as it req. logical reasoning, no such hard and fast rule mostly,
as have to know what features are important and what are not.

→ As we get so many patterns mostly and scare zaroori nahi hote, so step (8) very-very important in which we filter patterns after evaluating them.
↳ 3 things to consider:-

(1) patter should be simple enough so that it can be interpreted by you.
(2) Novel i.e. new data
(3) Confidence
↓
Statistically | mathematically
(4) Durability

↳ yeh 4 cheezein zaroori hain, tabhi uss pattern ko consider krna.

koi ek bhi cheez nahi hai, then pattern of no use.

pattern ka future mein bhi use ho.

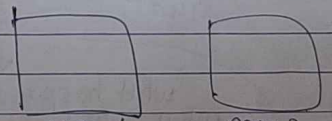
→ Preprocessing ⇒ most difficult task and intensive task out of all.
↳ firstly we will integrate datas into 1 but for it also we need middleware / interface programming as data can be in different format, and we want that they can interact with each other and integrate them, and even missing cheezein fill kro yaa ignore kro, in a data.

What kind of data, can I mine?
↳ any kind of data ☑
Yes tareeka bdlk jaayega, algo bdl jaayegi, preprocessing ka tareeka bdlk jaayega but mining ho skti hai.

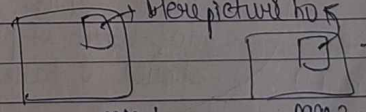
Spatial data
↳ locations are stored.
Time-series data
↳ Here time component is very-very important.
Heterogeneous data
↳ data stored in different locations and do not share same software or format,
WWW → semi structured

Xml → semi structured
↳ can make it structured as well

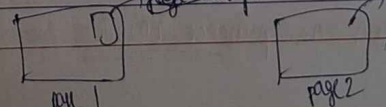
difference?



changes everything → unstructure



→ trying to structure it



→ semistructured, as some structured and some not

If a page has set template, then structured o/w not.

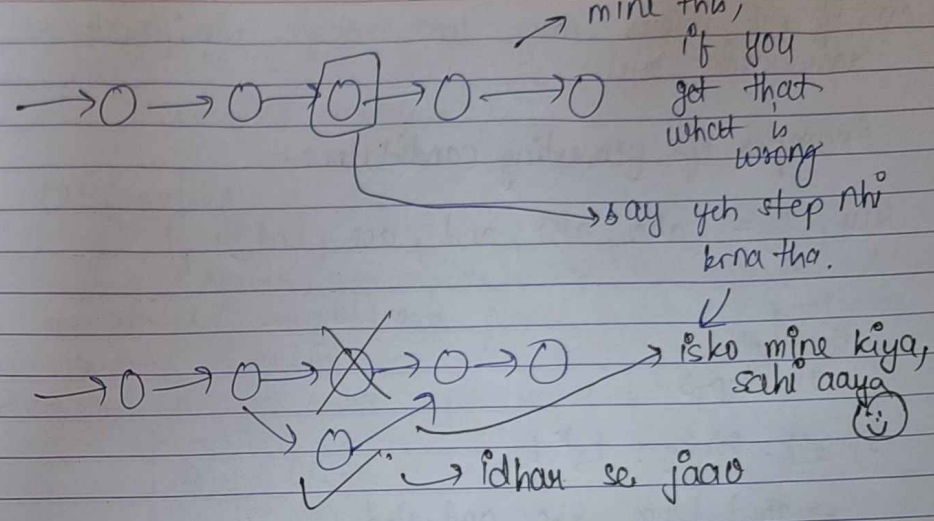
Data Mining
↳ confluence of multiple disciplines

Information Science
↳ isse we can calculate entropy etc.

Analysis → prescriptive
- descriptive → tells you "what happened"
- predictive → "what will happen"
- discovery → "pattern was already there, but we were not aware and hmne ab discovery kiya"

"what happened and what will happen ki gya pta, now tell what should I do? what steps should I take?"

→ aaj kal mostly prescriptive + descriptive analysis ki jaati hai.
combinely known as process mining (???)



(*) process mining aaj kal, jyada "system log" par ho raha hai,

Transaction Reduction :- we stop where candidate set can not be generated aage

↳ it says that in subsequent stages of apriori, if any transaction becomes insignificant, then we can ignore it, in all further steps. (that transaction)

means jitni bhi items ismein hain, that are no longer significant, means they are not appearing in large item set

→ How we can ignore them? what are advantages? what will be impact?
↳ support % formula will change,

so support both jaayega, toh again ignore nhi krenge, then we may miss out some prominent transactions or item sets. so ignore krna zaroori.

so, if not ignore, may loose some significant association rules.

Examples of generating candidates :-

say, L_3 = {abc, abd, acd, ace, bcd}

↑ large Item set at step-3

⊙ self-joining : $L_3 * L_3$

→ abcd from abc and abd
→ acde from acd and ace.

⊙ pruning
→ acde is removed bcoz ade is not in L_3.
↳ bcoz of anti monotone property.

So, C_4 = {abcd}

so no need to see database, evaluate support ... etc.
so pruning is done before scanning database and calculate support.

ARs from FIs :-

for each FI l, generate all non-empty subsets of l, and for each non-empty subset 's' of 'l', output the rule :-

$s \Rightarrow (l-s)$ if $\frac{\text{support\_count}(l)}{\text{support\_count}(s)} \geq$ min_conf

↓ minimum confidence.

Variations of Apriori :-

(1) Transaction reduction
↳ done.

(2) Sampling
↳ random transactions of the original database are selected (sampled) and placed in a much smaller sampled database.
→ our task is not reduce input/output (I/O).

D → Apriori → AR's
→ large I/O.

so instead take samples :-

S_1, S_2 → apply apriori on each sample
↳ calculate AR's

But these might does not represent all patterns,

so ek sample lo, apriori lgaao, AR nikaalo, then dusra sample lo, apply apriori, then AR milenge usko pehle wale mein add-on kro.

S_1 → apriori → PL (not LI) → potentially large items.

Then we talk about NBC() fn (Negative borderline function)

FI = $P^{PL} \cup$ NBC()
↓ frequent item sets.

returns itemsets that are not in PL but has all of their subsets in PL

Usually, while finding (FL) from sampled database, min support threshold is lowered.
↳ How much can be lowered? depends on application domain

$I = \{a, b, c, d\}$

$FL = \{\{a\}, \{c\}, \{d\}, \{c, d\}\}$ → after step 2.

$C_1 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{c, d\}\}$ ← after step 3.

Since a and c are frequent, so {a, c} also.
a and d — d —
so, NB() = {ac, ad} till now.

as, ac, ad, cd, a, c, d are frequent,
thus, {acd} also frequent.

so, NB() = {ac, ad, acd}
cd frequent hai but nahi hoga in NB(), as we already FL mein hai, after step-2.

(3) Partitioning
↳ instead of sampling, do (n) partitions:- $D_1, D_2, \ldots, D_n$.
Improve performance

Q. How to partition? what will size of each partition?
↳ each partition size must be less or equivalent to RAM
↳ so that each partition can be executed at once.

$D_1 \rightarrow$ apriori $\rightarrow FI_1$
$D_2 \rightarrow$ apriori $\rightarrow FI_2$
$\vdots$
$D_n \rightarrow$ apriori $\rightarrow FI_n$

$L = FI = FI_1 \cup FI_2 \ldots \cup FI_n$

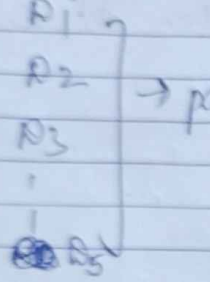
~~$L = L_1 \cup L_2 \ldots \cup$~~
$L = L^1 \cup L^2 \ldots \cup L^n$

~~$L = L^1 \cup L^2 \cup$~~

↓
then generate rules.
then what difference it makes?
↳ as still doing same thing only, accessing same and whole data.
will there be same time complexity as of apriori original?
↓
No! it will be less.

say D → apriori (2 GB = RAM)
↓
10 GB.
⇒ for each iteration of apriori, 5 input/output are req.
↓
min

Say 5 iteration
↳ then min 5X5 (=25) I/O req.



→ partition.

Now each iteration will take min 1 I/O.

↳ so overall 5 I/O.

(4) Hash based
↳ It uses hash function to generate LIs.

D → apriori → L_i

then hash use kiya,

$L_1 * L_2$ → C_2 → to prune C_2

So instead of scanning whole data base for L_2, we will do pruning first with help of L_1.

FP → frequent pattern approach / FP growth approach.
↳ In this just one iteration of database is req.

H/W → utility mining
↳ when we consider quantity of items in transaction database.

eg: how many bread and butter customer has taken.

4/2/25
Tuesday

lec-4 :-

FP growth algorithm
↳ frequent pattern
↳ which is actually tree data structure based algorithm.

in apriori → large I/O.
↳ so FP growth algo was revised,
↳ candidate generation
↳ $C_1 \ldots C_k$
↓
$L_1 \ldots L_k$

FP growth says that 'NO candidate generation req'.
↳ so this part will be skipped in FP, and that time of scanning database will get reduced.

FP
↳ divide and conquer methodology : decompose mining tasks into smaller ones.
→ requires 2 scans of transaction DB
→ 2 phase algo
↳ Phase I :- construct FP tree
↳ Phase II :- Seeing FP tree

FP-tree Construction :-
↳ Stem prefix tree
(→ Most frequent will always be prefered.)

→ FP Header table
→ dependent on ordering of items
→ sort items in decreasing order of support count
↳ so highest frequency wala ↑ professed.
→ Non FIs are ignored
→ Each Tr. is viewed as a list of FIs in descending order of support count.

Item	frequency ~~support~~	→ so → order →
1	5	1
2	3	3
3	4	2

Construct FP tree from Transaction DB.

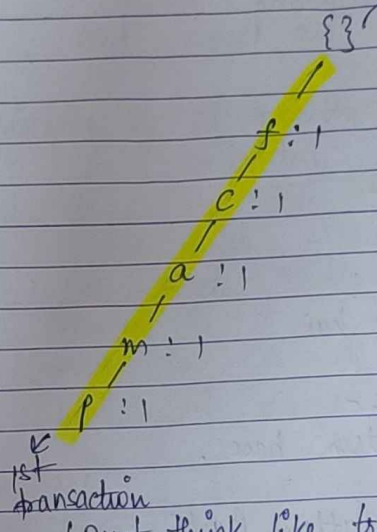
→ jinki frequency unko kaise mrzi order mein lo, no difference.

TID	Items	(ordered) frequent items
100	f, a, c, d, g, i, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m	f, c, a, m, p

as min. support = 0.5

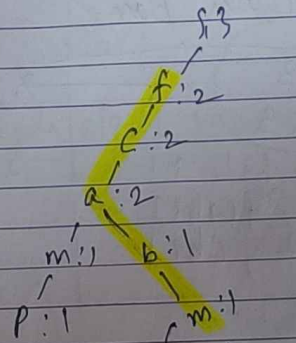
Header Table	
Item	frequency head
f	4
c	4
a	3
b	3
m	3
p	3

Now, we'll make tree :-



(just think like trie, maintaining frequency as well)

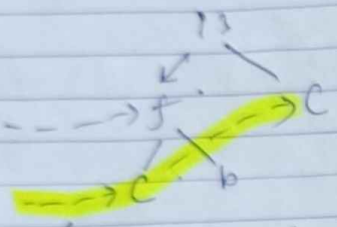
2nd transaction :- f, c, a, b, m



so on, complete the tree !!!

work in progress

⇢ → dotted arrows telling how transactions are participating



'c' esse jaa raha hai

* Major steps to mine FP tree :-
  ↳ conditional pattern base.

* From FP tree to Conditional Pattern Base.

Conditional Pattern Base.	
c	f:3
a	fc:3
b	fca:1, f:1, c:1
m	fca:2, fcab:1
p	fcam:2, cb:1

'a' appearing 3 times with 'fc'.

→ means 'c' appearing 3 times with 'f'.

these 2 pattern bases are highly frequent.

	confidence
f ⇒ c	3/4
c ⇒ f	1

⇒ can derive easily from pattern base

work in progress

→ So ab jo 'min- confidence' condition di hogi uss hisaab se dekh lenge, what to ignore and what not to 😊

Principles of FP growth :-
  ↳ pattern growth property.

Let α be frequent in DB, B be α's conditional pattern base, and β be itemset in B, then α∪β is a frequent itemset in DB iff β is frequent in B, o/w just prune the things.

"abcdef" is FP ⇔ "abcde" is FP and "f" is frequent in the set of transactions containing "abcde".

(and you can see it is very different from apriori).

Hash based algo (variation of apriori) :-
  ↳ primarily works for k=2

yaa, we can also use for k≥3, but mostly not used

↳ means C_2 tak jaana pdhega.

1st ly generate C_1, then using hash, we calculate C_2; no need to go to database for scanning again 😊

100	(1,3), (1,4), (3,4)
200	(2,3), (2,5), (3,5)
300	(1,2), (1,3), (1,5), (2,3), (2,5), (3,5)
400	(2,5)

$H(x,y) = \{(\text{order of } x) * 10 + (\text{order of } y)\} \bmod 7$

↳ Hash function

$(3)*10+5) \bmod 7 = 35 \bmod 7 = 0$

Hash table.

0	1	2	3	4	5	6
3,5	1,5	2,3	0	2,5	1,2	1,3
3,5	1,5	2,3	0	2,5	1,2	3,4
1,4	1,5	2,3	0	2,5	1,2	1,3
3	1	2	0	3	1	3
0	1	2	3	4	5	6

count

Bucket No.

↳ jo hash function se pta chl raha hai.

Yeh ignored

Yeh liya

so count ≥ 2 wale lo.

So,

bit vector | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

$L_1 * L_1 = \{\{1,2\}, \{1,3\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,5\}\}$

1 3 1 2 3 3

No. in the bucket with Item set.

So,

$C_2 = \{1,3\}, \{2,3\}, \{2,5\}, \{3,5\}$ ↳ Baaki 2 ignored.

WTMK 30 min.

WTMK 30 min.

WTMK 30 min.

support (????)

as min ~~conf~~ = 50%

4 transactions in total

↳ so >= 2 ✓ count kelo.

If Item sets are 'words' rather than 'no.', then hash function may not work.

Q. If prblm changes, how to figure out to the hash function ????

*Multiple-Level Association Rules:-

→ Items often ~~from~~ form hierarchy.

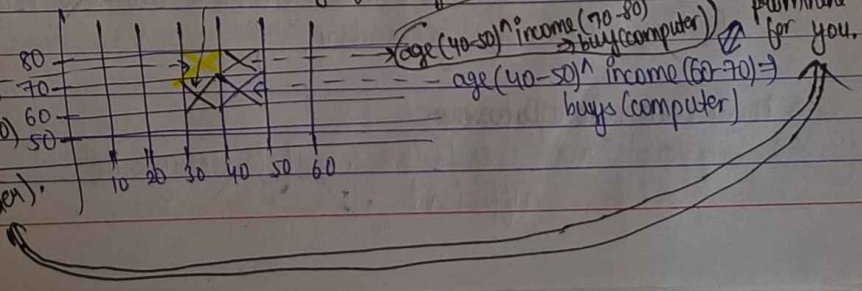
→ Items at lower level are expected to have lower support.

→ Rules regarding itemsets at appropriate levels could be quite useful.

Observation:-

① Clustering of association rules is making life easy, as jyada rules we can not handle, so ek rule banao but then that rule must contain all other rules.

age(30-40) ^ income(60-70) ⇒ Buys(computer) ↔ this rule talking of this guid.



⇒ instead of these 4 rules, make single rule :-

age (30 - 50)^ income (60 - 80) ⇒ buys (computer)
↳ Best rule.

↳ and unhi rules ko cluster krna which has minimum confidence, wo rules jo cluster kr rahe ho must be genuine are.

(2) → mining (vertical) Transactional database (TD)
↳ also known as "long".

→ abhi tak jitne bhi pdhein hain TD, wo sare horizontal database the.

TID	Item
T_{10}	a, b
T_{11}	a, c, d

→ Horizontal DB

trying to figure out association b/w items

Items	TID
a	T_{10}, T_{11}
b	T_{10}
c	T_{11}
d	T_{11}

become ID

→ vertical DB
↳ why this? what advantages in practical?

trying to figure out association b/w customers. ↳ so that can handle similar customers

↳ this will tell how far T_{10}, T_{11} wale customers are related to each other.

⇒ Transaction is nothing but customer jab kuch buy krta hai, wo ek transaction bn jaata hai.

(3) → Support and confidence akele may not be sufficient ;

$A \Rightarrow B$

3rd metric can use is referred as lift
↳ tells correlation b/w A and B.

lift $(A \Rightarrow B)$

- > 1 → positively correlated
- < 1 → negatively "
- $= 1$ → no correlation / independent
  ↳ yeh nahi, as no association,

yeh chiye

H/W:
↳ one example having lift as well
↳ what is lift.

(4) → sometimes in place of 'lift', we may use

$(chi)^2 = (\chi^2)$

↳ H/W → isse also ek example and write what it is,

$$\chi^2 = \sum \left(\frac{(\text{observed} - \text{expected})^2}{\text{expected}} \right)$$

* Data preprocessing :-

Data ↳ can be text, image, audio, video.

process of converting data into meaningful data.

(1) why we do this?
↳ because our data dirty → incomplete, noisy, inconsistent and duplicate hota h,

and we want to clean it.

⊙ Steps of data preprocessing :-
- ↳ Data cleaning
- ↳ Data integration
- ↳ Data reduction
- ↳ Data Transformation

Data cleaning
↳ handles 2 tasks :-
(1) missing value
(2) noisy data

Data Integration
↳ merge the data from different sources.

Data reduction
↳ duplicacy Htaana.

Data Transformation
↳ normalise karna data ko.
↳ helps in data modeling.

(*) Techniques to remove noise :- → random error / unwanted data.
↳ like
- 1k
- 2k
- NA
- 3k

↳ noise.

(1) → Binning

fx :-
10, 2, 19, 18, 20, 18, 25, 28, 22

3rd step me hota hain

Types of Binning →
- smoothing by bin means
- medians
- boundary

Step 1 - smoothing by bin means

step 1 sort data :-
2, 10, 18, 18, 19, 20, 22, 25, 28

step 2. then divide data into BIN buckets say, bucket of 3 size each

2, 10, 18
18, 19, 20
22, 25, 28 → 3 buckets.

(*) say smoothing by bin mean :-

2, 10, 18 → mean = 10
18, 19, 20 → " " 19
22, 25, 28 → " " 25.

so sabko mean 'k' equal kardo →

10 10 10
19 19 19
25 25 25

(*) If by median :- odd = $\left(\frac{n+1}{2}\right)$ even = $\left(\frac{n}{2}\right)$

10
19 → aayega → so
25

10 10 10
19 19 19
25 25 25

WJMK

② say by boundary

min ↙ ↘ max.

2, 10, 18 → min = 2
max = 18

replace 10 by min as uske paas main hi hai

2,	2,	18
18,	18,	20
22,	22,	25

hmm → chhote hain,
dekha yeh kiske paas, as min and max dono

jyada

k paas, toh 'min' ko prefer krenge.

Normalization → as then big value dominate nahi kregi jab error nikaalenge yaa jab tree mein distance nikaalenge

→ krenge ek specific range mein laane k liye.

→ 2 techniques:-

(1) min - max normalization technique
(2) Z- ~~softmax~~ normalization "
~~exp~~ → square

House	Sq. foot	Bedrooms	price (in lakhs)
1	1200	3	50
2	1500	4	60
3	1000	2	40
4	1800	5	80

WJMK

old value.

$$X' = \frac{X - min}{max - min}$$

new value

for price ⇒
min = 1000
max = 1800

so for House 1 → $X' = \frac{1200-1000}{1800-1000}$

$= \frac{12}{18} = \frac{1}{4} = 0.25$ → [0,1] range mein aagya ☺

for House 2 → $X' = \frac{1500-1000}{1800-1000}$

$= \frac{5}{8} = 0.625$

for House 3 → $X' = 0$

for House 4 → $X' = \frac{1800-1000}{1800-1000} = 1$

for bedroom ⇒
min = 2
max = 5

for House 1 ⇒ X' = 0.33
—— 2 ⇒ X' = 0.6
—— 3 ⇒ X' = 0
—— 4 ⇒ X' = 1

for price ⇒
for House 1 ⇒ X' = 0.25
—— 2 ⇒ X' = 0.5
—— 3 ⇒ X' = 0
—— 4 ⇒ X' = 1

WTMK

Z-square → If values on same scale then use Z-square method.

↳ also known as 0-mean method.

Student	Height (in inches)
1	64
2	70
3	72
4	68
5	76

$$x' = \frac{x - \text{mean}(\mu)}{\text{Standard deviation}(\sigma)}$$

$$\sigma = \sqrt{\left(\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}\right)}$$

x_i → Term given in data
$\bar{x}$ → mean
n = total no. of terms

$$\mu = \frac{\sum x_i}{n} \quad (i \text{ from } 1 \text{ to } N)$$

and

$$\sigma = \sqrt{\frac{(64-70)^2 + (70-70)^2 + (72-70)^2 + (68-70)^2 + (76-70)^2}{5}}$$

$$= 4$$

then,

Student	Height
1	64 ⇒ x' =
2	70 ⇒ x' =
3	72 ⇒ x' =
4	68 ⇒ x' =
5	76 ⇒ x' =

WJMK

कालतंतु कारेचरिन्त एनर मरिष्कु,
निरमुक्तेर कालेत्वम अमरिष्कु॥

WJMK

Homework Solution :-

LIFT :-

- it is a simple correlation measure
- The occurence of itemset A is independent of the occurence of itemset B if $P(A \cup B) = P(A) \cdot P(B)$, otherwise both itemsets are dependent and correlated.

$$lift(A, B) = \frac{P(A \cap B)}{P(A) \cdot P(B)}$$

- If $lift(A, B) < 1$, occurence of A is negatively correlated with occurence of B, meaning that occurence of one likely leads to the absence of another one.
- If $lift(A, B) > 1$, then A and B are positively correlated, meaning occurence of one implies occurence of other.
- If $lift(A, B) = 1$, then A, B are independent and there is no correlation between them.

for example :- Total transactions analyzed = 10,000
transactions which include computer games = 6000
transactions which include vedios = 7500
transactions which include both = 4000
say, min-support = 30%
min-confidence = 60%

WMK

Then following association rule can be discovered :-

buys (X, "games") ⇒ buys (X, "vedios")

as,

support = $\frac{4000}{10,000} = 40\% \geq 30\%$

confidence = $\frac{4000}{6000} = 66\% \geq 60\%$.

but this is misleading because probability of purchasing videos is $75\% > 66\%$.

Infact games and vedios are negatively associated because purchase of one of these items decreases likelihood of purchasing the other.

So here confidence is deceiving and does not measure real strength (or lack of strength) of correlation and implication b/w A and B.

Hence, alternative required.

As,

$P(\{game\}) = 0.6$

$P(\{vedio\}) = 0.75$

$P(\{game, vedio\}) = 0.4$

$lift(game \Rightarrow vedio) = \frac{0.4}{0.6 \times 0.75} = 0.89 < 1$ → (negative correlation)

χ^2 :-

The second correlation measure is χ^2 measure.

WMK

$$\chi^2 = \sum \frac{(Observed - expected)^2}{expected}.$$

for example :-

Observed :-

	game	game	Σrow
vedio	4000	3500	7500
vedio	2000	500	2500
Σcol	6000	4000	10,000

Expected :-

	game	game	Σrow
vedio	4500	3000	7500
vedio	1500	1000	2500
Σcol	6000	4000	10,000

Now,

$$\chi^2 = \frac{(4000-4500)^2}{4500} + \frac{(3500-3000)^2}{3000} + \frac{(2000-1500)^2}{1500} + \frac{(500-1000)^2}{1000} = 555.6$$

Now, $\chi^2 > 1$ and observed value of slot (game, video) = 4000, < 4500 (expected), so game and vedio are negatively correlated,

⊙ so, χ^2 measure, test if two categorical variables are independent or correlated.

→ It checks whether observed values in a dataset are different from expected values, if there was no relationships.

High χ^2 → Items are highly correlated.
Low χ^2 → [illegible] means if χ^2 close to 0, observed values match the expected values, meaning no correlation.
If ~~observed~~. χ^2 large, >0, and observed value is less than expected value ⇒ negatively correlated.
If χ^2 large, >0 and Observed > expected ⇒ positively correlated.

⇒ Support tells us how frequently an itemset appears.
⇒ confidence tells us how strong rule is.

confidence. tells us how strong

25/2/25
Tuesday

Lec-5 :-

Classification :- Decision Tree :-

* Supervised Classification :-

Data
- ↳ Training
- ↳ Testing

learning
- ↳ Supervised
- ↳ Unsupervised

⇓

based on data, domain etc., we can decide the ratio.

→ Training data jyada rkhna hai to train model in nice manner, but kitne o/o, depends on total data we've.

WORK

agar

If total data = 1MB → then 80% lenge training mein, then very less.

But,

If total data = 1TB, then 80% is more than enough.

Decision Trees:- (DT)

- ↳ classification scheme
- ↳ represents a model of different classes
- ↳ generates tree and set of rules
- ↳ a node w/o children is leaf node, o/w internal node

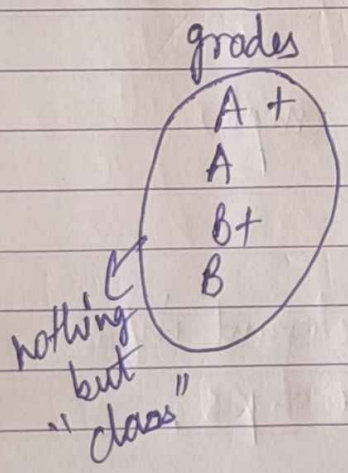
↓

each internal node has associated splitting predicate eg: binary predicates.

↳ eg: Age <= 20.

profession in { student, teacher }.

3* salary - 10000 > 0



* "sample DT" might be DT of ~~data~~ data, but not necessarily to be DT.

* Leaf node → classes / rule
* root + internal node ↔ feature / attributes
* edges → predicate / cond's

↳ How to decide which node is root? and kis feature ko konsa node dena hai???

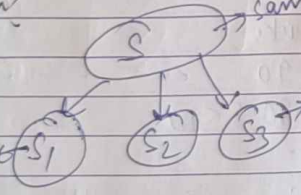
↳ saare features nhi rkhte hm DT, sirf wo rkhte hai which has high feature importance, means those which play significant role in decision making

→ and then we will make their priority list.

↳ This all is part of preprocessing.

* Agar distinct values kam hain, then shki ek-2 branch bna skte hain, o/w do grouping.

How to construct DT?
DT induction?



then isko is mrnkar aage bdta jayega... so on, till we get any relevant info and then tree stops.

Entropy:- (Info theory)
↳ measure of uncertainty associated with a random variable

$H(Y|X) = \sum_x p(x)\, H(Y|X=x)$
↳ Conditional Entropy

* Attribute Selection Measure
↳ Information Gain (ID3/C4.5)

→ Select attribute with highest info gain,

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

where,
p_i is probability that an arbitrary tuple in D belongs to C_i estimated by $\frac{|C_{i,D}|}{|D|}$.

(Expected info (entropy) needed to classify a tuple in D)

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

↑ Info needed to classify D after using A split to split D into v partitions

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

↑ Info gained by branching on attribute A.

* Gain for continuous valued attributes :-

→ firstly sort data / attributes in ascending order.

Gain Ratio for attribute selection (C4.5) :-

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

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

→ attribute with max gain ratio is selected as splitting attribute.

* what are special cases, where we can be in problem in ID3?

temp given	grouped
65	<=75
70	<=75
76	>75
78	>75
60	<=75

→ yehi kista ID3, but can't group key based attributes like your roll no, so then ID3 fails, and then only gain ratio kaam aa skti hai, i.e. C4.5.

* Gini Index (CART, IBM Intelligent Miner) :-

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

where p_j is relative frequency of class j in D and total there are 'n' classes.

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

* gini index is index of impurity
↳ so hm usko minimum se minimum krkne ki koshish krte hain →

→ biased to multivalued attributes.
→ has difficulty when # of classes is large
→ tends to favor tests that results in equal sized partitions.

Other attribute selection measures :-

→ CHAID, C-SEP, G-statistic, (MDL), CART, multivariate splits

(MDL) → quite popular

Which attribute selection measure is best?
↓
based on data, domain and features.

ID3 → when no noise, no key
so on!!!!

Overfitting the Data :- When DT is built, many of the branches may reflect anomalies in the training data due to noise or outliers.
→ we may grow the tree just deeply enough to perfectly classify the training data set.
→ This problem is known as "overfitting the data".

D

S_1 (Training) S_2 (Testing)

applied some algo

go DT

2 ways to avoid overfitting :-

(1) during DT construction itself
↳ pre-pruning

(2) after prepa bn gya, fir tree ko scan kro
↳ post-pruning

Problems of overfitting :- ?

Prepruning :-

↳ Usually we don't go for it, as we've to track thing - so on.

↳ tree is pruned by halting the tree construction i.e. by deciding not to further split or partition the subset of training samples at a given node.

→ Upon halting, a node becomes a leaf node.

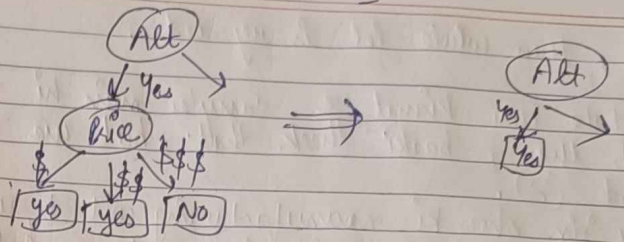
→ leaf may hold the most frequent class among the subset samples.

Post pruning :-

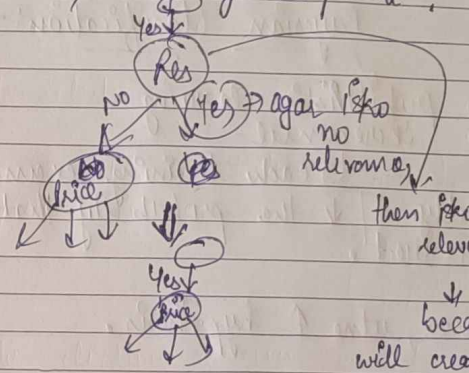
↳ more easy to implement
↳ as have entire tree with us, now scan and see whether we can prune something or not.

Two methods :-

(1) Subtree replacement replaces subtree with a single leaf node;



(2) Subtree raising :- moves subtree to a higher level in decision tree, subsuming its parent.



Bayesian Classification :-

Bayes Thm

↳ jab shoes/sandals beche jaate hain on discount even w/o any defect is based on bayes theorem

because companies works on lot, usmein se kuch samples nikaal ke discount pe becho, defect dekho, agar threshold niche chla gya

than decided, then whole lot is rejected.

↓

chale bari kismat se samples mein hi khraab tha, baaki theek that, but we rejected ☹.

→ issiliye jin shoes ki manufacturing cost ↓, unko bhi ↑ rate pare bechte hain, bcoz unhone, flats ko reject kiya hota hai! (bhut se)

Bayesian Classification:-

- probabilistic learning
- Incremental

↳ each training example can incrementally ↑/↓ the probability that a hypothesis is correct.

→ jitni ↑ classes, utne ↑ hypothesis.

- Estimating probabilities:-

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

- Naive Bayesian classification:-

↳ also called simple BC,

① class conditional Independence

↓

This assumption simplifies computations.

From bayes theorem:-

$$P(C_i|X) = \frac{P(X|C_i) \, P(C_i)}{P(X)}$$

* $P(X)$ is constant.

Only $P(X|C_i) \cdot P(C_i)$ needs to be maximized.

Naive Assumption:-

$$P(X|C_i) = \prod (P(x_k|C_i)) \text{ over } k=1 \text{ to } n,$$

→ This algo may fail also. (H/W)

(Naive Bayes classification) → what remedies to do, so that Naive Bayes classification can be applied.