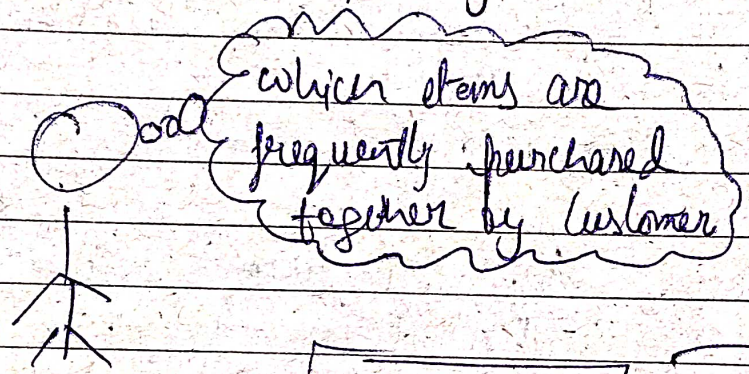
Chapt 6. Data Mining

Frequent Patterns → Freq patterns are patterns that appear frequently in a data set.

A typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying habits by finding associations b/w the different items that customers place in their "shopping baskets". The discovery of these association can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customer.



market based analysis

item that are frequently purchased together can be placed in proximity to further encourage the combined sale of such items.

Placing hardware & software at opposite ends of the store may entice customers who purchase such items to pick up other items along the way.

Date:
Page No.:

Association Rules

Customers who purchase computers also tend to buy antivirus software at the same time

Computer ⇒ antivirus_Software [Support = 2%, Confidence = 60%]

Rule Support and Confidence are two measures of rule interestingness.

Support → Reflect the usefulness

Confidence → Certainty of discovered rule

* A support of 2% means that 2% of all the transaction under analysis show that computer and antivirus software are purchased together.

* A Confidence of 60% means that 60% of the customers who purchased a computer also bought the software

Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold.

These thresholds can be a set by users or domain experts

Date :
Page No. :

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

$$\text{Confidence}(A \to B) = P(B|A)$$

Rules that satisfy both minimum support threshold and a minimum confidence threshold are called strong

The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also called frequency, support count, count

Note → $\text{Support}(A \to B) = P(A \cup B)$ → relative support

Occurrence frequency → absolute support

If the relative support of an itemset I satisfies a minimum support threshold (i.e, the absolute support of item I satisfies a minimum support count threshold) then I is a frequent itemset

$$\text{Confidence}(A \to B) = P(B|A) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

$$= \frac{\text{Support Count}(A \cup B)}{\text{Support Count}(A)}$$

Date :
Page No. :

In general, association rule mining can be viewed as a two step process :

1) Find all frequent itemsets : By definition, each of these itemset will occur at least as frequently as a predetermined minimum support count, min_sup

2) Generate strong association rules from the frequent itemsets : By definition, these rules must satisfy minimum support and minimum confidence

Because the second step is much less costly than the first, the overall performance of mining association rules is determined by the first step.

A Major challenge in mining frequent itemsets from a large data set is the fact that such mining often generates a huge number of itemsets satisfying the min_sup especially when min_sup is low.

For ex 100 digits of itemset frequent

$^{100}C_1 = 100$ frequent 1-itemset $\{a_1\}, \{a_2\}, \ldots$

$^{100}C_2 =$ frequent 2-itemset $\{a_1, a_2\}, \{a_1, a_3\} \ldots$

Total $\Rightarrow {}^{100}C_1 + {}^{100}C_2 + {}^{100}C_3 + \ldots {}^{100}C_{100} = 2^{100} - 1$

$\approx 1.27 \times 10^{30}$

This is too huge number of item set for any computer to compute or store.

Date :
Page No. :

To overcome this difficulty, we introduce the concept of closed frequent itemset and maximal frequent itemset

① Closed frequent itemset :→ An itemset is closed if none of its immediate ~~supersets~~ superset has same support as that of itemset.

or

"An itemset X is closed in dataset D if there exists no proper super-itemset Y such that Y has the same support count as X in D"

② Maximal frequent itemset :→ An itemset is maximal if none of its immediate ~~supersets~~ superset is frequent.

"An itemset X in dataset D if X is frequent, and there exist no super itemset Y such that X ⊂ Y and Y is frequent in D"

Itemset	
A	(3)
B	(4)
C	(5)
D	(4)

Min_Sup = 3

Item	Count	
A, B	3	
A, C	3	
A, D	2	→ not frequent
B, C	4	
B, D	3	
C, D	4	

Date:
Page No.:

A(3) → AB(3), AC(3), AD(2)

A (count) is not greater that its immediate superset. A is not closed

In A's immediate supersets, items are present with min-sup-count = 3

A is not maximal

Apriori Property

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

To improve the efficiency of level wise generation of frequent itemsets, an important property called the Apriori property

→ All nonempty subsets of a frequent itemset must also be frequent.

Antimonotonicity → If a set cannot pass a test, all its supersets will fail the same test as well.

min_support = 2

Date: / /
Page No.:

Ex

TID	Item
T100	1, 2, 5
T2	2, 4
T3	2, 3
T4	1, 2, 4
T5	1, 3
T6	2, 3
T7	1, 3
T8	1, 2, 3, 5
T9	1, 2, 3

C1

Itemset	Sup Count
1	6
2	7
3	6
4	2
5	2

L1

Itemset	Sup Count
1	6
2	7
3	6
4	2
5	2

C2

Itemset	Sup_count
{1, 2}	4
{1, 3}	4
{1, 4}	1 ✗
{1, 5}	2
{2, 3}	4
{2, 4}	2
{2, 5}	2
{3, 4}	0 ✗
{3, 5}	1 ✗
{4, 5}	0 ✗

L2

Itemset	Sup_count
{1, 2}	4
{1, 3}	4
{1, 5}	2
{2, 3}	4
{2, 4}	2
{2, 5}	2

C3 / L3

Itemset	Sup_count
{1, 2, 3}	2
{1, 2, 5}	2

Date:
Page No.

		Support	Confidence	
1,2,3 =	$1 \wedge 2 \rightarrow 3$	2	2/4	50%
	$1 \wedge 3 \rightarrow 2$	2	2/4	50%
	$2 \wedge 3 \rightarrow 1$	2	2/4	50%
1,2,5 =	$1 \wedge 2 \rightarrow 5$	2	2/4	50%
	$1 \wedge 5 \rightarrow 2$	2	2/2	100%
	$2 \wedge 5 \rightarrow 1$	2	2/2	100%
$P(A \rightarrow B) =$	$3 \rightarrow 1 \wedge 2$	2	2/6	33.3
Supp(A)	$2 \rightarrow 1 \wedge 3$	2	2/7	28.5
=	$1 \rightarrow 2 \wedge 3$	2	2/6	33.3
	$5 \rightarrow 1 \wedge 2$	2	2/2	100
	$2 \rightarrow 1 \wedge 5$	2	2/7	28.5
	$1 \rightarrow 1 \wedge 5$	2	2/6	33.3

Disadvantages

Requires Many Database Scans
Assumes Transaction Database is Memory resident

Improving the Efficiency of Apriori

1) Hash Based Technique →

TID	Items
T1	1,2,5
T2	2,4
T3	2,3
T4	1,2,4
T5	1,3
T6	2,3
T7	1,3
T8	1,2,3,5
T9	1,2,3

Itemset	Support Count
1	6
2	7
3	6
4	2
5	2

$$h[x,y] = \left((\text{order of } x) \times 10 + (\text{order of } y) \right) \bmod 7$$

Itemset	Count	Hash
1,2	4	(1×10+2) mod 7 = 5
1,3	4	(1×10+3) mod 7 = 6
1,4	1	(1×10+4) % 7 = 0
1,5	2	(1×10+5) % 7 = 1
2,3	4	(2×10+3) % 7 = 2
2,4	2	(2×10+4) % 7 = 3
2,5	2	(2×10+5) % 7 = 4
3,4	0	(3×10+4) % 7 =
3,5	1	(3×10+5) % 7 = 0
4,5	0	

Hash Table H_2

bucket address	0	1	2	3	4	5	6
bucket count	1+1	2	4	2	2	4	4
bucket contents	{1,2} {3,5}	{1,5} {2,5}	{2,3} {1,5} {2,3} {2,3}	{2,4} {2,4}	{2,5} {4,5}	{1,2} {1,2} {1,2} {1,2}	{1,3} {1,3} {1,3} {1,3}

Transaction reduction → A transaction that does not contain any frequent k itemsets cannot contain any frequent (k+1)-itemset. Therefore, such a transaction can be removed from further consideration because subsequent database scans for j itemsets where j > k, will not need to consider.

Eg

T1	1,2,5
T2	2,3,4
T3	3,4
T4	1,2,3,4

min_sup = 2

Date:
Page No.:

	1	2	3	4	5
T1	1	1	0	0	1
T2	0	1	1	1	0
T3	0	0	1	1	0
T4	1	1	1	1	0

< 2 ✗

⇒

	1	2	3	4
T1	1	1	0	0
T2	0	1	1	1
T3	0	0	1	1
T4	1	1	1	1

④

	{1,2,3}	{1,3}	{1,4}	{2,3}	{2,4}	{3,4}
T1	1	0	0	0	0	0
T2	0	0	0	1	1	0
T3	0	0	0	0	0	1
T4	1	1	1	1	1	1

	{1,2}	{2,3}	{2,4}	{3,4}
T2	0	1	1	0
T4	1	1	1	1

⇓

	{1,2,3}	{1,2,4}	{2,3,4}
T2	0	0	1
T4	1	1	1

⇓

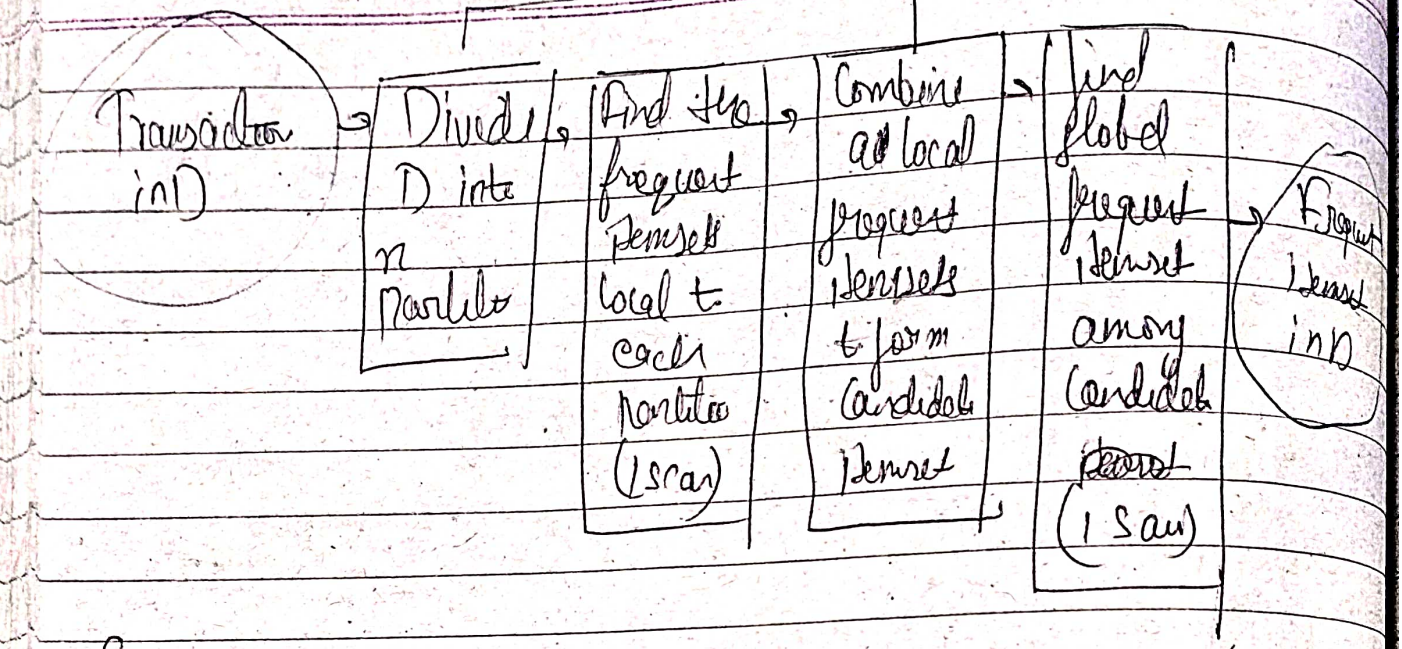
	{2,3,4}
T4	1

Partitioning

Any itemset that is Potentially Frequent in DB must be frequent in atleast one of the partitions of DB (2 DB Scans)

Phase I

Phase II



Eg

	1	2	3	4	5
T1	1	0	0	0	1
T2	0	1	0	1	0
T3	0	0	0	1	1
T4	0	1	1	0	0
T5	0	0	0	0	1
T6	0	1	1	1	0

Database is divided into 3 Partition

Each have 20% min-Sup

Trans	Itemset	First Scan. Sup = 20% min.sup = 1	Second Scan Sup = 20% min.sup = 2	Shortlist
T1	{1,5}	1→1, 2→1	1→1, 2→3	
T2	2,4	4→1, 5→1, {1,5}→1, {4,4}→1	3→2, 4→3	2→3
T3	4,5	4→1, 5→1, 2→1, 3→1	5→3,	3→2
T4	2,3	{4,5}→1, {2,3}→1	{1,5}→1	4→3
T5	5	5→1, 2→1, 3→1, 4→1	{4,5}→1	5→3
T6	2,3,4	{2,3}→1, {2,4}→1	{2,4}→2	{2,4}→2
			{4,5}→1	{2,3}→2
			{2,3}→2	
			{3,4}→1	
			{1,3,4}→1	

Date:
Page No.:

Dynamic Itemset Counting

It is an algorithm which reduces the number of passes made over the data while keeping the number of itemsets which are counted in any pass relatively low.

This technique can add new candidate itemset at any marked start point of the database during the scanning of the DB

Sampling

The Basic idea to pick up a random sample S of the given data D, and then search for frequent itemsets in S instead of D.

It may be possible to lose a global frequent itemset

This can be reduced by lowering the Min_Support

There is Trade off some degree of Accuracy against efficiency

FP Growth algo

Divide & Conquer strategy

Date :
Page No. :

Ex min Support = 30%.

Tid	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

min Support Count $= \frac{8 \times 30}{100} = 2.4 \approx 3$

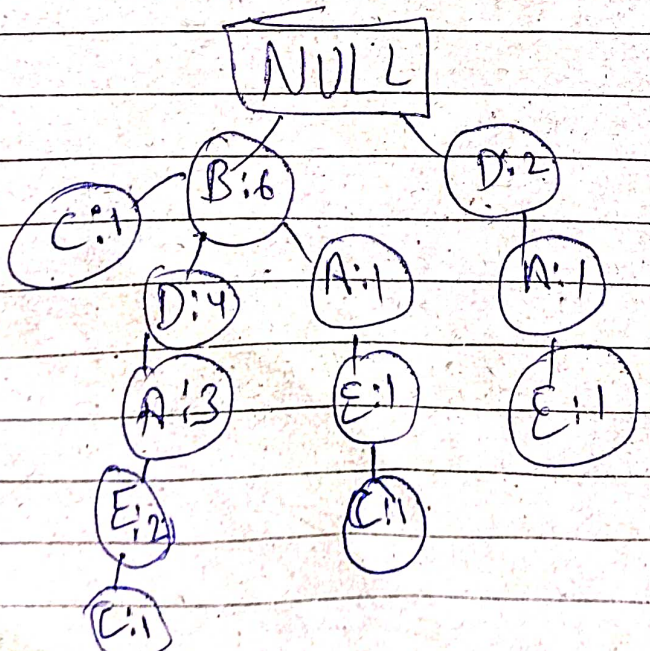
items	Sup Count	Priority
A	5	3
B	6	1st
C	3	5
D	6	2
E	4	4

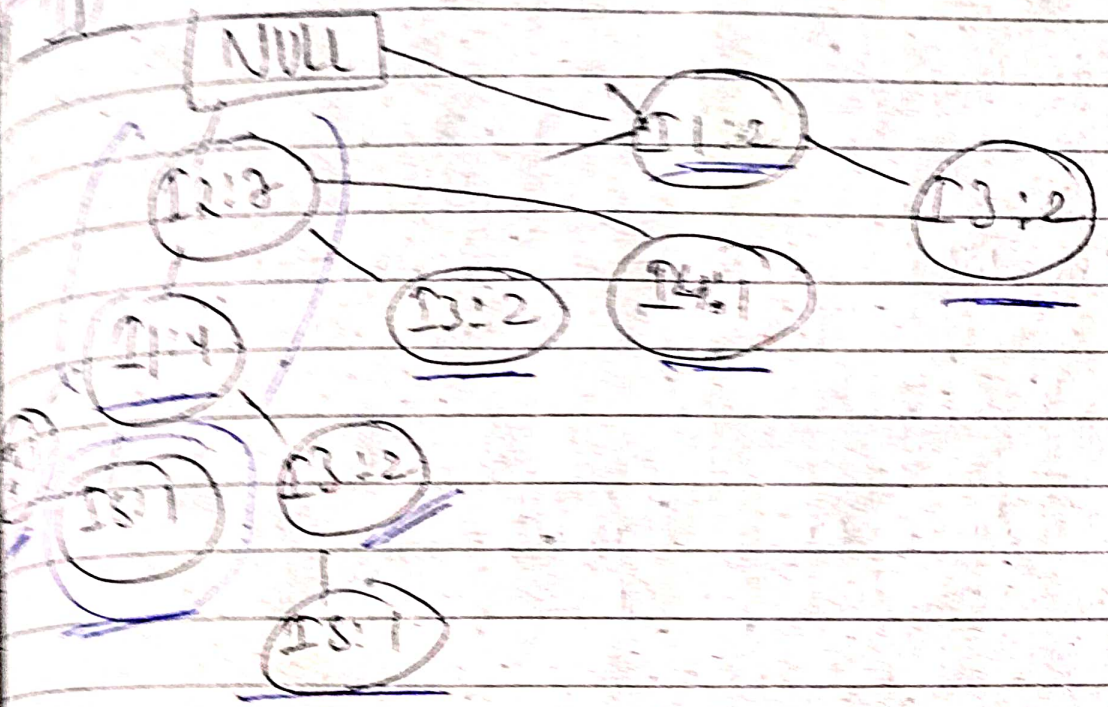
Tid	items	ordered items
1	E, A, D, B	B, D, A, E
2	D, A, C, E, B	B, D, A, E, C
3	C, A, B, E	B, A, E, C
4	B, A, D	B, D, A
5	D	D
6	D, B	B, D
7	A, D, E	D, A, E
8	B, C	B, C

B: 1, 2, 3, 4, 5, 6
D: 1, 2, 3, 4
A: 1, 2, 3
E: 1, 2
C: 1

A: 1
E: 1
C: 1

D: 1, 2
A: 1
E: 1
C: 1





Item	Conditional Pattern Base
I5	{I2, I1:1} {I2, I1, I3:1}
I4	{I2, I1:1} {I2:1}
I3	{I2, I1:2} {I2:2} {I1:2}
I1	{I2:4}

Conditional FP tree

Item	Frequent Pattern Generated
I5	{I2, I1:2} {I2, I3:2}
I4	{I2:2}
I3	{I2, I1:2} {I2:2} {I1:2}
I1	{I2:4}

Item	Frequent Pattern Generated
I5	{I2, I5:2} {I1, I5:2} {I2, I1, I5:2}
I4	{I2, I4:2}
I3	{I2, I3:4} {I1, I3:4} {I2, I1, I3:2}
I1	{I2, I1:4}

Date :
Page No. :

Q

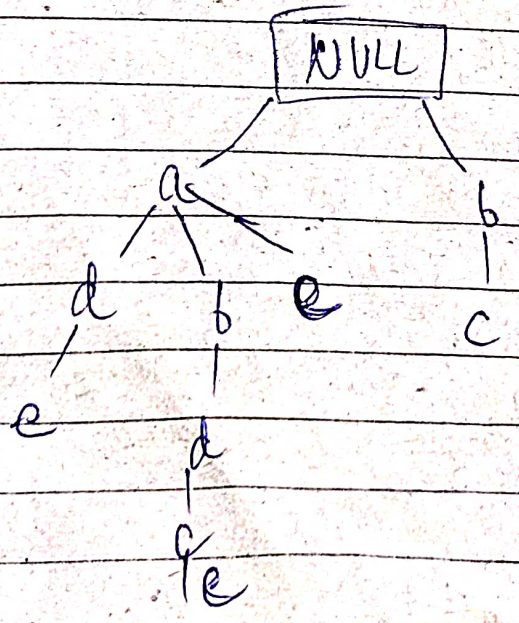
TID	itemset
1	{a, d, e}
2	{a, d, b, c}
3	{a, d, b}
4	{a, e}
5	{b, c}
6	{a, d, b, e, c}

min_supp = 2

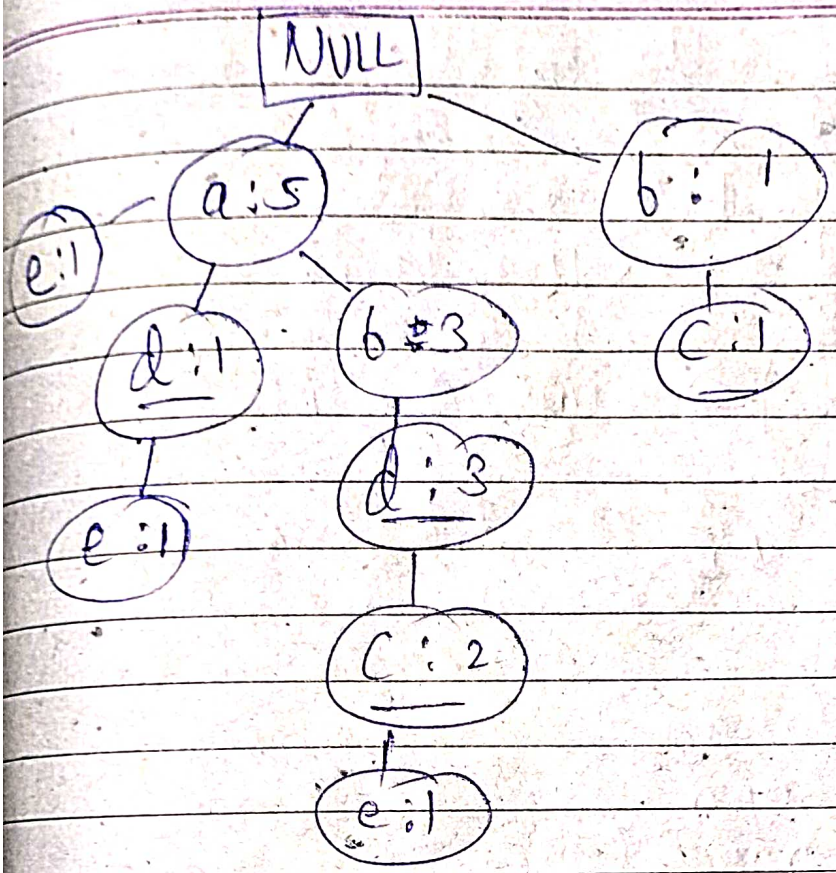
Item	Support Count	Priority
a	5	1
b	4	2
c	3	4
d	4	3
e	3	5

TID	itemset	ordered itemset
1	{a, d, e}	{a, d, e}
2	{a, d, b, c}	{a, b, d, c}
3	{a, d, b}	{a, b, d}
4	{a, e}	{a, e}
5	{b, c}	{b, c}
6	{a, d, b, e, c}	{a, b, d, c, e}

a:1, 2, 3, 4, 5
d:1
e:1
b:1, 2, 3
d:1, 2, 3
c:1, 2
e:1
b:1
c:1
e:1



Date :
Page No. :



Item	Conditional Pattern base	
e	{a, b, d, c : 1} {a, d : 1}, {a : 1}	all are in same branch
c	{a, b, d : 2} {b : 1}	
d	{a : 1} {a, b : 3}	all are in same branch
b	{a : 3}	

Item	Conditional FP tree	Freq. Pattern Generated
e	{a : 3, d : 2}	{a, e : 3} {d, e : 2} {a, d, e : 2}
c	{a, b, d : 2}	{a, c : 2} {b, c : 2} {d, c : 2}, {a, b, c : 2} {a, d, c : 2}, {b, d, c : 2}, {a, b, d, c : 2}
d	{a : 4, b : 3}	{a, d : 4} {b, d : 3}, {a, b, d : 3}
b	{a : 3}	{a, b : 3}

Horizontal format	Vertical Format
$T_1 : \{I_1, I_2, I_3$	$I_1 : \{T_{100}, T_{200}\}$
$T_2 : \{I_2, I_5\}$	$I_2 : \{T_{100}, T_{300}\}$

$buys(X, \text{"Computer games"}) \Rightarrow buys(X, \text{"Videos"})$

[Support = 40%, Confidence = 60%]

total transaction = 10000
Computer games = 6000
Videos = 7500

Computer game ∩ Video = 4000

$$\text{Support}(A \to B) = \frac{P(A \cup B)}{\text{①}} = \frac{4000}{10000} = 40\%$$

$$\text{Confidence}(A \to B) = \frac{P(A \cup B)}{P(A)} = \frac{4000}{6000} = \frac{2 \times 1000}{3} = 66.7\%$$

$$\frac{6000}{10000}$$

left

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

Date :
Page No. :

Lift $(A,B) < 1$ → negatively correlated

Lift $(A,B) > 1$ → positively correlated

Lift $(A,B) = 1$ → Independent

$$\text{Lift}(A,B) = \frac{P(B/A)}{P(B)} = \frac{\text{Confidence}(A \to B)}{\text{Sup}(B)}$$

χ^2 Correlation

	game	game	
Video	4000 (4500)	3500 (3000)	7500
Video	2000 (1500)	500 (1000)	2500
	6000	4000	10000

Expected $E_{ij} = \frac{\text{row}_i \times \text{col}_j}{\text{Total}}$

$$E_{11} = \frac{3500 \times 6000}{10000} = 4500$$

$$\chi^2 = \frac{\sum (O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(4500-4000)^2}{4500} + \frac{(3000-3500)^2}{3500}$$

$$+ \frac{(2000-1500)^2}{1500} + \frac{(500-1000)^2}{1000}$$

$$= 555.6$$

Because the χ^2 value is greater than 1, and the observed value of the slot (game, video) = 4000 which is less than expected value of 4500.

So buying game and buying video are negatively correlated.

Pattern Evaluation Measures

1) all_confidence (A, B)

$$= \frac{Sup(A \cup B)}{\max\{Sup(A), Sup(B)\}} = \min\{P(A/B), P(B/A)\}$$

2) max_conf (A, B) =

$$\max\{P(A/B), P(B/A)\}$$

$$= \max\left\{\frac{Sup(A \cup B)}{Sup(A)}, \frac{Sup(A \cup B)}{Sup(B)}\right\}$$

3) Kulczynki =

$$\frac{1}{2}\left(\frac{S(A \cup B)}{S(A)} + \frac{S(A \cup B)}{S(B)}\right) = \frac{1}{2}\left(P(A/B) + P(B/A)\right)$$

used to find out imbalance ratio robust

4) Cosine (A, B) $= \frac{S(A \cup B)}{\sqrt{S(A) \times S(B)}} = \sqrt{P(A/B) \times P(B/A)}$

Date :
Page No. :

Null Transaction → It is a transaction that does not contain any of the itemsets being examined.

Null Variant → If its value is free from influence of null transaction.

Correlation χ^2 and lift are not Null Variant

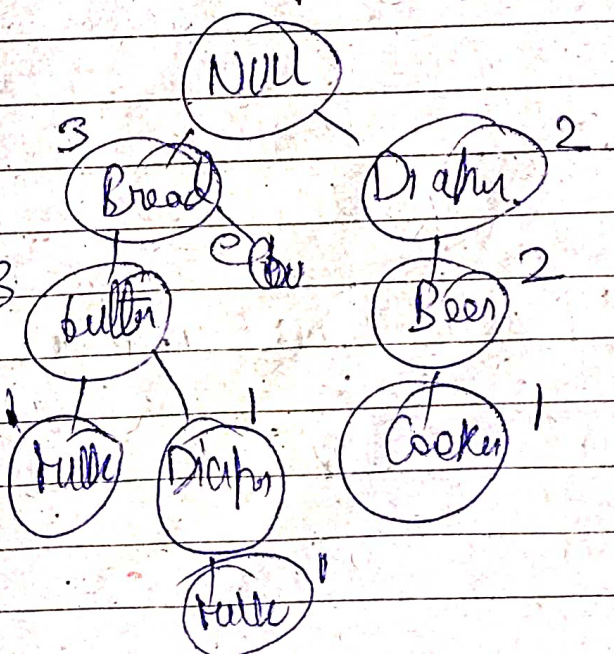
Imbalance Ratio

$$IR(A,B) = \frac{|Sup(A) - Sup(B)|}{Sup(A) + Sup(B) - Sup(A \cup B)}$$

TID	Item set
1	{Bread, butter, Milk}
2	{Bread, Butter}
3	{Beer, Cookies, Diapers}
4	{Milk, Diapers, Bread, Butter}
5	{Beer, Diapers}

Bread, butter, Milk
Bread, butter
Diapers, Beer, Cookies
Bread, butter, Diapers, Milk
Diapers, Beer

Item	Count	Records
Bread	3	1
butter	3	2
Milk	2	4
Beer	2	5
Cookies	1	6
Diapers	3	3



Chapt - 8

Page No.:

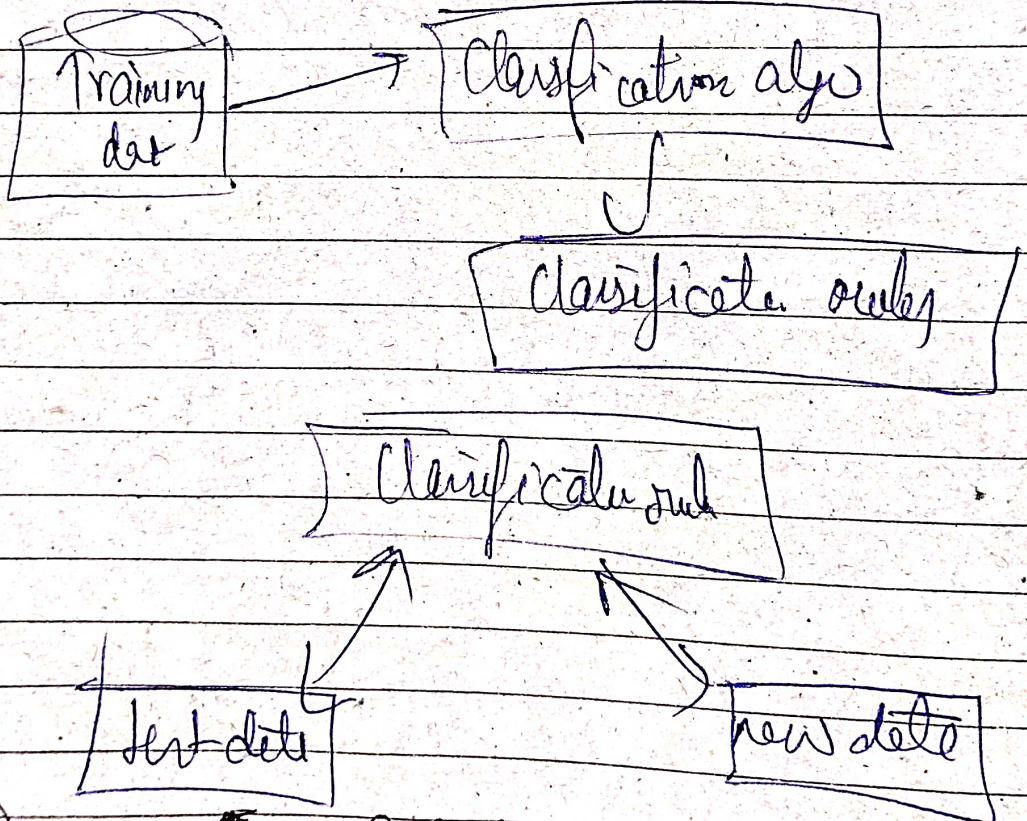
Classification

It is a form of data analysis that extracts model describing important data classes.

Data Classificatn → (i) learning step
(ii) classificatn step

a) Learning → Training data are analyzed by a classification algo.

b) Classification → Test data are used to estimate the accuracy of the classificatn rules



Decision Tree Induction:

It is the learning of decision tree from class-labeled training tuples.

Date :
Page No. :

Attribute Selection Measures

It is a heuristic for selecting the splitting criterion that "best" seperate a given data partition, D, of class-labeled training tuples into individual classes.

Information Gain

ID3 uses info Gain as its attribute selection measure.

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

Info(D) is known as Entropy

$$Entropy = -\sum_{i=1}^{m} \frac{P_i}{P} \log_2(P_i)$$

$$= \frac{-P}{P+N} \log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N} \log_2\left(\frac{N}{P+N}\right)$$

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

$\frac{|D_j|}{|D|}$ acts as the weight of jth partition

$$Gain(A) = Info(D) - Info\_Gain(D)$$

$$= Entropy - Info_{Gain}$$

tells how much would be gained by branching on A.

Date :
Page No. :

Entropy → Uncertainty associated with given info

Info Gain → Information Gained

ID3 algo

Day	Weather	Temp	Humidity	Play
D1	Sunny	Hot	High	N
D2	Sunny	Hot	high	N
D3	Cloudy	Hot	High	Y
D4	Rainy	Mild	Normal	Y
D5	Rainy	Cool	Normal	Y

(1) Entropy (entire) $\{3, 2\}$ = Y, N

$$= \frac{-P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

$$= \frac{-3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)$$

$$\frac{-P}{P+N}\log_{10}\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_{10}\left(\frac{N}{P+N}\right)$$

$$\log_{10}(2)$$

$$= 3.321\left(\frac{-3}{5}\log_{10}(0.6) - \frac{2}{5}\log_{10}(0.4)\right)$$

$$= 3.321(0.133 + 0.159)$$

$$= 0.969$$

Date :
Page No. :

(ii) Entropy (Weather) {Sunny} {0, -2} = 3.321 $\left(\frac{-0}{2}\log(0) - \frac{2}{2}\log(1)\right)$

= 0

Entropy (Cloudy) {1, 0} = $\frac{-1}{1}\log(1) - \frac{0}{1}\log(0)$ = 0

Entropy (Rainy) {2, 0} = 0

Info(weather) = Entropy(entire) − $\frac{2}{5}$ x 0 − $\frac{1}{5}$ x 0 − $\frac{2}{5}$ x 0

Info(weather) = 0.969

(iii) Temp

E(Hot) {1, -2} = 3.321 $\left\{\frac{-1}{3}\log\left(\frac{1}{3}\right) - \frac{2}{3}\log\left(\frac{2}{3}\right)\right\}$

= 3.321 (0.159 + 0.117)

= 0.9165

E(Mild) {1, 0} = 0

E(Cool) {1, 0} = 0

Info(temp) = 0.969 − $\frac{3}{5}$ x 0.9165 = 0.419

(iv) Humidity

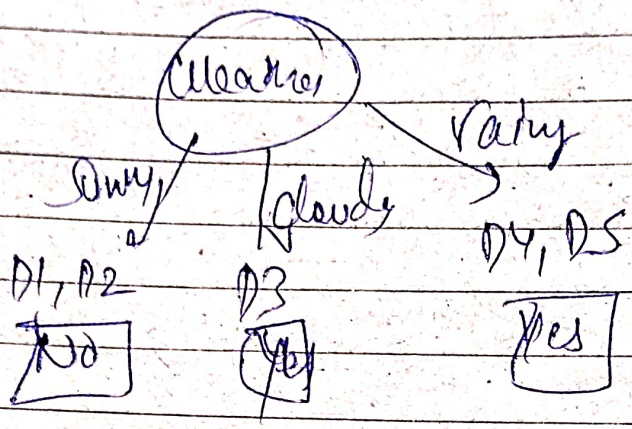
E(High) {1, -2} = 0.9165

E(Normal) {2, 0} = 0

Info = 0.419

Date :
Page No. :

Maximum Weight = 0.91



Gini Index

Impurity of Data Set

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

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

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

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

Ex

Day	Weather	Temp	Play
D1	Sunny	Hot	No
D2	Sunny	Hot	No
D3	Cloudy	Hot	Yes
D4	Rainy	Mild	Yes
D5	Rainy	Cool	Yes

Date:
Page No.:

Gini (Entire) $\{3, -2\} = 1 - \sum p_i^2$

$= 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48$

Gini (Weather)

Gini (Sunny) $\{0, -2\} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$

Gini (Cloudy) $\{1, 0\} = 0$

Gini (Rainy) $\{2, -0\} = 0$

Gini (Weather) $= \frac{2}{5} \times 0 + \frac{1}{5} \times 0 + \frac{2}{5} \times 0 = 0$

Gini (Temp)

Gini (Hot) $\{1, -2\} = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 1 - 0.111 - 0.444$

$= 0.445$

Gini (Mild) $\{1, 0\} = 0$

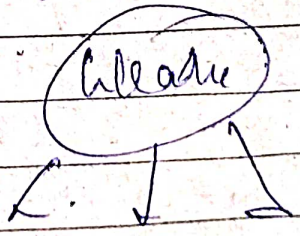
Gini (Cool) $\{1, 0\} = 0$

Gini (Temp) $= \frac{3}{5} \times 0.445 + \frac{1}{5} \times 0 + \frac{0 \times 1}{5} = 0.267$

Gini (Weather) $<$ Gini (Temp)

↓

less impurity



Date :
Page No. :

Day	Weather	Temp	age	income	Student	Credit	buy
D1	Sunny	Hot	Y	H	no	F	no
D2	Sunny	Hot	M	H	no	E	no
D3	Cloudy		M	H	no	F	y
D4	Rainy		S	M	no	F	y
D5	Rainy		S	L	Yes	F	y
D6	Rainy		S	L	Yes	E	N
D7	Cloudy		M	L	Yes	E	y
D8	Sunny		Y	M	no	F	N
D9	Sunny		Y	L	Yes	F	y
D10	Rain		S	M	Yes	F	y
D11	Sunny		Y	M	Yes	E	y
D12	Cloudy		M	M	no	E	y
D13	Cloudy		M	H	Yes	F	y
D14	Rainy		S	M	no	E	N

$$E(\text{Entire})\{9,-5\} = \frac{-9}{14}\log\left(\frac{9}{14}\right) - \frac{5}{14}\log\left(\frac{9}{14}\right) = 0.94$$

~~Weather~~ Age

$E(\text{Young})\{2,-3\} = 0.97$

$E(\text{Middle})\{4,-0\} = 0$

$E(\text{Senior})\{3,-2\} = 0.97$

$$IG = 0.94 - \frac{5}{14}\times 0.97 - \frac{4}{14}\times 0 - \frac{5}{14}\times 0.97 = 0.246$$

Date :
Page No. :

Cart Algo

Age

Youth	5	Yes	2
		NO	3
Middle	4	Yes	4
		NO	0
Senior	5	Yes	3
		NO	2

→ greater

Attribute	Rules	error (bes/total)	Total
Age	Youth → NO	2/5	4/14
	Middle → Yes	0/4	
	Senior → Yes	2/5	

Income

High	4	Yes	2
		NO	2
Medium	6	Yes	4
		NO	2
low	4	Yes	3
		NO	1

Rules	Error	Total
High → Yes/NO	2/4	5/14
Medium → Yes	2/6	
low → Yes	1/4	

Student

Yes	7	Yes	6
		NO	1
NO	7	Yes	3
		no	4

Rules	Error	
Yes → Yes	1/7	8/14
NO → no	3/7	

Date :
Page No. :

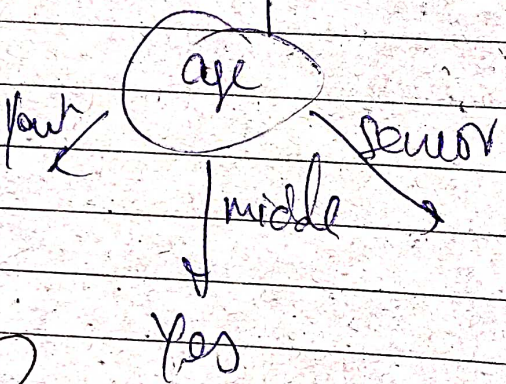
Credit

Fair	8	Yes	6
		no	2
Excellent	6	Yes	3
		no	3

Rules	Error	
Fair → Yes	2/8	5/14
Excell → Yes/no	3/6	

	Rules	error	Total
Age →	Youth → no	2/5	4/14
	Middle → Yes	0/4	
	Senior → Yes	2/5	
Student	Yes → Yes	0/7	4/4
	no → no	3/7	

Same direct ☺



Tree Pruning

To reduce Overfitting

Two approaches → (i) Pre pruning
(ii) Post pruning

Pre Pruning → halting its Construction early

Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset tuple or the probability distribution of those tuple

Date :
Page No.:

When constructing a tree, measures such as statistical significance, information gain, Gini index, and so on, can be used to assess the goodness of split.

If partitioning the tuple at a node would result in a split that falls below a prespecified threshold, then further partitioning is halted.

However choosing an appropriate threshold is very difficult

High threshold → over simplified trees

low threshold → very little simplification

Various measure → max-depth, max-feature etc

"Pre-Pruning is used on Big Dataset"

Post Pruning → remove subtree from a fully grown tree.

A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most frequent class among the subtree being replaced

"It is used on Small DataSet"

CART → Cost Complexity pruning algo → no of leaves error rate

C4.5 → pessimistic pruning → error rates

Date :
Page No. :

Bayesian Classifiers

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

→ Bayes' theorem
→ Naive Bayes Classifier

Bayes theorem

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

$$P(A|B) \times P(B) = P(B|A) P(A)$$

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}$$

H → hypothesis
D → Data tuple

Likelihood

$$P(H|D) = \frac{P(D|H) \times P(H)}{P(D)}$$

Posterior

Prior

Marginal

Date:
Page No.:

P(H) → Prior → Probability of hypothesis before giving evidence

Naive Bayes Classifier

Determine if weather = Sunny, Temp = Cool, Humidity = high, windy = strong ?

$P(Yes) = \frac{9}{14} = 0.64$ $\quad P(NO) = \frac{5}{14} = 0.36$

Conditional Probability of each feature

Weather	Yes	NO
Sunny	Sunny & Yes / total Yes = 2/9	Sunny & NO / total NO = 3/5
cloudy	4/9	0/5
Rainy	3/9	2/5

Humidity	Yes	NO
high	3/9	4/5
Normal	6/9	1/5

Temp	Yes	NO
Hot	2/9	2/5
mild	4/9	2/5
Cool	3/9	1/5

Windy	Yes	NO
Strong	3/9	3/5
weak	6/9	2/5

$$V_{NB} = \underset{v_j \in \{Yes, No\}}{\operatorname{argmax}} \; P(V_j) \prod_i P(a_i | V_j)$$

$V_{NB}(Yes) = P(Yes)\, P(Sunny|Yes)\, P(Cool|Yes)\, P(High|Yes)\, P(Strong|Yes)$

$= \frac{9}{14} \times \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} = 0.053$

Date :
Page No. :

$$V_{NB}(no) = P(no)\, P(Sunny/no)\, P(Cool/no)\, P(high/no)\, P(Strong/no)$$

$$= \frac{5}{14} \times \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} = 0.026$$

To overCome Zero Probability Problem

Laplace Smoothing

$$P_k(x|y) = \frac{Count(x,y) + k}{Count(y) + k|X|}$$

$k \rightarrow$ represents smoothing parameter > 0

$|X| \rightarrow$ no of dimension / features $\rightarrow$ weather
- Sunny
- rainy
- cloudy

$\Rightarrow 3$

$$P_{k=1}(Outlook = Cloudy/no) = \frac{Count(Outlook = cloudy/no) + 1}{Count(no) + 1 \times 3}$$

$$= \frac{1}{8}$$

Date :
Page No. :

Eg Fruit = {Yellow, Sweet, Long}

Fruit	Yellow	Sweet	Long	Total
Orange	350	450	0	650
Banana	400	300	350	400
Others	50	100	150	150
Total	800	850	400	1200

$$P(\text{Yellow} \mid \text{Orange}) = \frac{P(\text{Orange} \mid \text{Yellow}) \times P(\text{Yellow})}{P(\text{Orange})}$$

$$= \frac{\frac{350}{800} \times \frac{800}{1200}}{\frac{650}{1200}} = 0.53$$

$$P(\text{Sweet} \mid \text{Orange}) = \frac{P(\text{Orange} \mid \text{Sweet}) \times P(\text{Sweet})}{P(\text{Orange})} = \frac{\frac{450}{850} \times \frac{850}{1200}}{\frac{650}{1200}}$$

$$= 0.69$$

$$P(\text{Long} \mid \text{Orange}) = \frac{P(\text{Orange} \mid \text{Long}) \times P(\text{Long})}{P(\text{Orange})} = \frac{0}{1200} = 0$$

$$P(\text{Fruit} \mid \text{Orange}) = P(Y \mid O)\, P(S \mid O) \times P(L \mid O) = 0$$

Similarly

$$P(\text{Fruit} \mid \text{Banana}) = P(Y \mid B)\, P(S \mid B)\, P(L \mid B) = 1 \times 0.75 \times 0.09 = 0.15$$

$$P(\text{Fruit} \mid \text{Others}) = P(Y \mid \text{Others})\, P(S \mid \text{Others})\, P(L \mid \text{Others}) = 0.022$$

⇒ greatest → Fruit = Banana

Date :
Page No. :

Rule Based Classification

Using IF - then Rules

- Rule antecedent or Precondition
- Rule Consequent

Rule Confliction

Conflict resolution strategy to figure out which rule gets to fire and assign its class prediction to X.

(i) Size Ordering → assigns the highest priority to the triggering rule that has the "toughest" requirements, where toughness is measured by the rule antecedent size

(ii) Rule Ordering →

a) Class based → Classes are sorted in order of decreasing "importance" such as by decreasing order of prevalence.

b) Rule Based → Organized into long priority list according to some measure of rule quality such as accuracy, coverage, size

Date :
Page No. :

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

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

Rule Induction Using a Sequential Covering algo

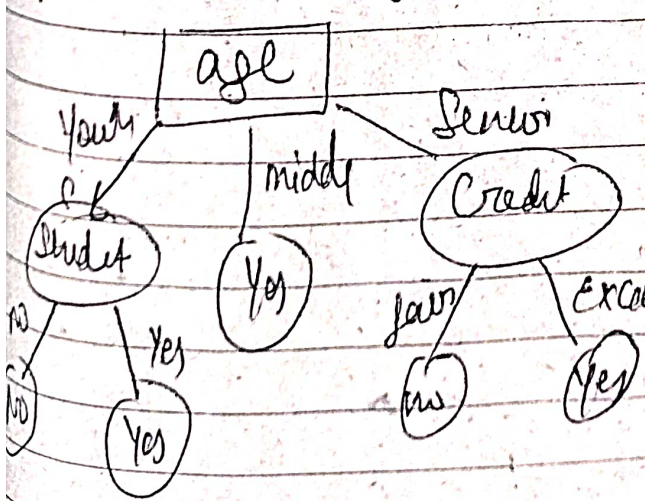
Rule_Set = {}
for each class c do

    Rule = Learn_One_Rule (D, att_value, c);
    remove tuples covered by Rule from D
    Rule_Set = Rule_Set + Rule;

    Until terminating condition

End for
return Rule_Set;

Rule Extraction from Decision Tree



R1: If age = youth and student = no
then buy Computer = no

R2: If age = youth and student = yes
then buy computer = yes

R3: If age = middle then buy
computer = yes

R4: If age = senior and Credit = fair
then buy Comput = no

R5: If age = senior and credit = Excellent then buy Comput = yes

Date :
Page No. :

Characteristics of Rules

(1) Mutually Exclusive → We cannot have rule conflict here because here the two rules will triggered for the same tuple.

(2) Exhaustive → There is one rule for each and every tuple. for each possible attribute value combination.

Rule Quality Measure

FOIL → First order inductive learner

(1) FOIL_Gain = $pos' \left(\log_2 \left(\frac{pos'}{pos' + neg'} \right) - \log_2 \left(\frac{pos}{pos + neg} \right) \right)$

(2) Likelihood_Ratio = $2 \sum_{j=1}^{m} f_i \log \left(\frac{f_i}{e_i} \right)$

Metrics for Evaluating Classifier Performance

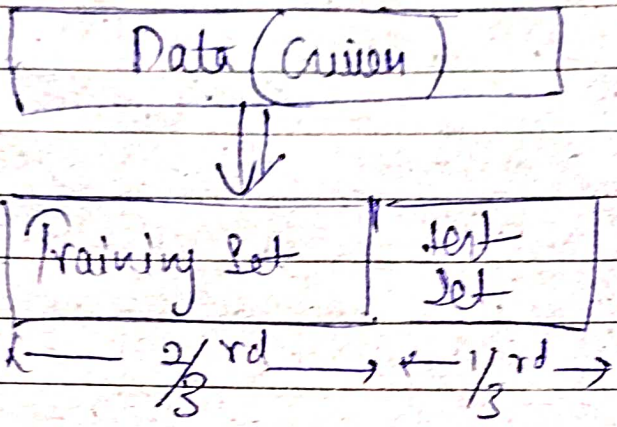
		Predicted		
		Yes	No	Total
Actual	Yes	TP	FN	P
	No	FP	TN	N
	Total	P'	N'	P+N

Measure	Formula
accuracy, recognition rate	$\frac{TP + TN}{P+N}$
error rate, misclassified rate	$\frac{FP + FN}{P+N}$
sensitivity, true positive rate, Recall	$\frac{TP}{P = (TP + FN)}$
specificity, true negative rate,	$\frac{TN}{N = (FP + TN)}$
Precision	$\frac{TP}{TP + FP}$
F, F_1, F_1 Score harmonic mean of precision & recall	$\frac{2 \times Precision \times Recall}{Precision + Recall}$
F_β; $\beta \rightarrow$ non negative	$\frac{(1+\beta^2) \times Precision \times recall}{\beta^2 \times precision \times recall}$

Date :
Page No. :

Holdout Method

The given data are randomly partitioned into two independent sets, a training set and test set



Training set is used to derive the model

Test set is used to estimate the model's accuracy

The estimate is pessimistic because only a portion of the initial data is used to derive the model

Random Subsampling

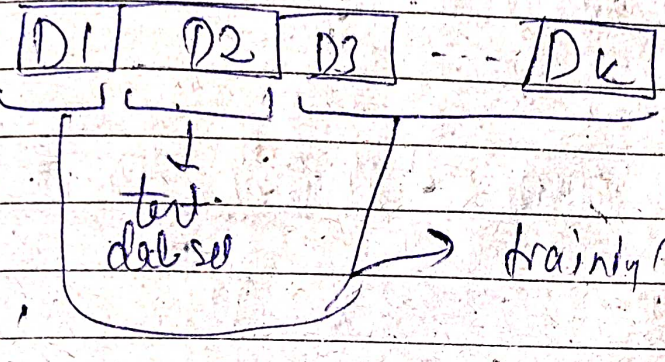
It is a variation of holdout method in which holdout method is repeated 'k' times.

The overall accuracy → average of each accuracies

CROSS VALIDATION

K-FOLD Validation

Date :
Page No. :



and so on

Accuracy = $\frac{\text{Overall no of Correct classifications for k iterations}}{\text{total no of tuples in initial data}}$

Leave-one-out

It is special case of K-fold Cross validation where K is set to the number of initial tuples.

Only one sample is "left out" at a time for the test set.

Stratified K-fold

The folds are stratified so that the class distribution of the tuples in each fold is approx the same as that in initial data.

Advantages of 10 fold Cross Validation → accuracy because low bias and variance

Date :
Page No. :

Bootstrap

The Bootstrap method samples the given training tuples uniformly with replacement. That is each time a tuple is selected, it is equally likely to be selected again and re-added to the training set

ROC Curve Receiver Operating Characteristic

TPR Vs FPR

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

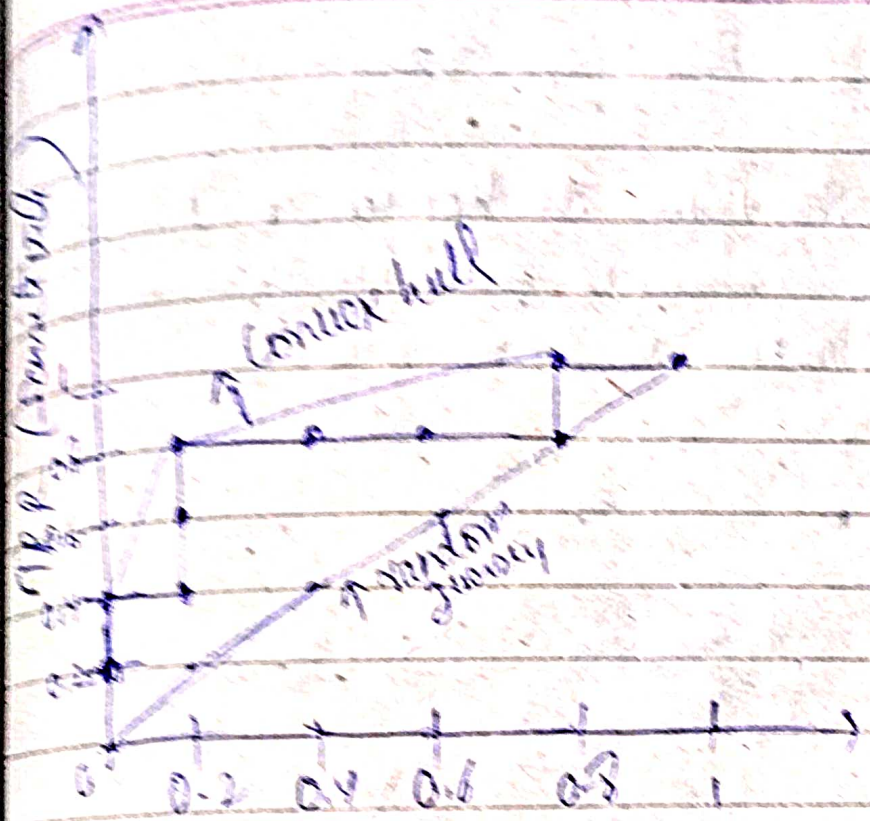
$\frac{TP}{P}$ → True Positive P → P → Total Positive

$\frac{FP}{N}$ → False Positive N → P → Total Negative

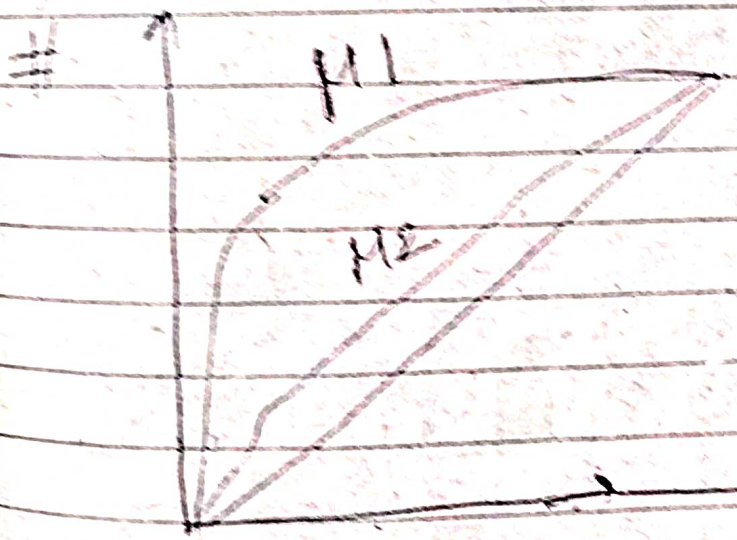
$FPR = \frac{FP}{FP+TN}$

Ex

Tuple #	Class	Prob	TP	FP	TPR = TP/P (Total P)	FPR = FP/N (Total N)
1	P→P	0.90	1	0	1/5 = 0.2	0
2	P→P	0.80	1+1=2	0	0.4	0
3	P→N	0.80	2	0+1=1	0.4	0.2
4	P→P	0.60	2+1=3	1	0.6	0.2
5	P→P	0.55	3+1=4	1	0.8	0.2
6	P→N	0.54	4	1+1=2	0.8	0.4
7	P→N	0.53	4	2+1=3	0.8	0.6
8	P→N	0.51	4	3+1=4	0.8	0.8
9	P→P	0.50	4+1=5	4	1	0.8
10	P→N	0.40	5	4+1=5	1	1



FPR (1 - Specificity)



The ROC curve is to the diagonal line, the less accurate the model is, Thus H1 is more accurate