values of α's given by G.A.

Hyperplane

$$H = \alpha_1 a_1 + \alpha_2 a_2 + \dots + \alpha_n a_n + \alpha_{n+1}$$

I want to find a suitable hyperplane.

which is able to distinguish b/w objects of diff. decision classes

u_i, u_j where $d(u_1) \neq d(u_2)$

Info system

$$\text{Conflict}(IS) = \underbrace{\frac{1}{2}}_{\text{since } (u_i, u_j) = (u_j, u_i)} \left| \{u_i, u_j\} \right|$$

So I make a collection C

left hand side of Hyperplane

decision class

$$C_i^L(H) = \{ U \mid \underbrace{\sum \alpha_k a_k}_{H} < 0,\ U \in X_i \}$$

$$C_i^R(H) = \{ U \mid \sum \alpha_k a_k \geq 0,\ U \in X_i \}$$

Scanned by CamScanner

For every pair of decision class,
majority of one in $c_i^L(H)$ &
majority of other in $c_j^R(H)$

$$\text{award}(H) = \sum_{i \neq j} |C_i^L(H)| \cdot |C_j^R(H)|$$

has to be maximized to get the most suitable hyperplane

other ways: max minority
sum minority
entropy calc
Leave it

No.
200
100
300
150
20
30
1000

⟹ we finally get some rules.

we need to have a metric to check how good a soln it is
(we want most optimised soln)

1) Best possible soln (using least resources)
2) Define metrics & talk in terms of that
3) Have the ability to validate your results

$$\text{Significance } \gamma(C, D) = \frac{|POS_C|}{|U|}$$

set of attributes

antidecant

support of a rule : consequent → decision

$$\gamma_x(C, D) = \frac{|[x]_C \cap D(x)|}{|U|}$$

rule

Segment	Incoming calls	Outgoing calls	change	Churn
1	Med	Med	low	no
2	High	high	low	no
3	low	low	low	no
4	low	low	high	yes
5	Med	Med	low	yes
6	Med	low.	low	yes

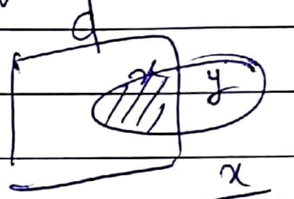
$\bigcup$

support

Decision Rule	Strength	Certainity	Coverage
1	$200/1000 = 0.2$	$200/420 = 0.48$	$\frac{200}{200+300+100} = 0.33$
2	0.1	1.00	0.17
3	0.3	1.00	0.5
4	0.15	1.00	0.38
5	0.22	0.52	0.55
6	0.03	1.00	0.07

probability that these features will lead to certain decision

Certainity of given rule :



$\frac{x}{x+y}$

$$Cer_x(C,D) = \frac{|[x]_C \cap D(x)|}{|[x]_C|}$$

for 1 :

$$\frac{(200+220) \cap (200)}{(200+220)}$$

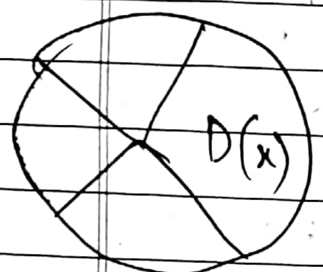
$$= 200/420$$

COVERAGE : To what extent this rule cover this decision class

$$Cov_x(C,D) = \frac{|[x]_C \cap D(x)|}{|D(x)|}$$

All this is going towards theory of total probability

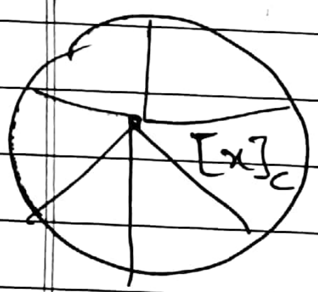
$$Cov_x(C,D) = \frac{|[x]_C \cap D(x)| \,/\, |U|}{|D(x)| \,/\, |U|}$$



$$= \frac{\gamma_x(C,D)}{\pi(D(x))}$$

$$Cer_x(C,D) = \frac{|[x]_C \cap D(x)| \,/\, |U|}{|[x]_C| \,/\, |U|}$$

$$= \frac{\gamma_x(C,D)}{\pi([x]_C)}$$



!!

If $[x]_C$ would be the entire universe, cer_x would tell what would be the support

Strength of an equivalence class = Sum of certainties weighted with $\pi([x]_C)$

Strength of decision class = Sum of coverages weighted with $\pi(D(x))$

If you have many classifiers & strgth rules, we may eliminate rules

Date ____
Page ____

⇒ Now, we have a compact rule base.

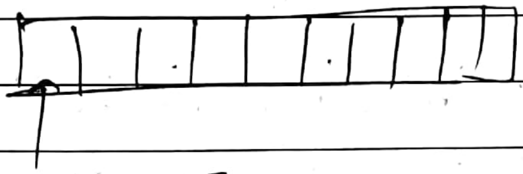
Now, we start classifying.

We have test objects and ~~we~~ what we do is:
We have database initially.
we keep 90% for train or 50%
10% for testing 50%

⇒ This is called cross-validation.

We go for n-fold cross validation

eg: 10-fold cross



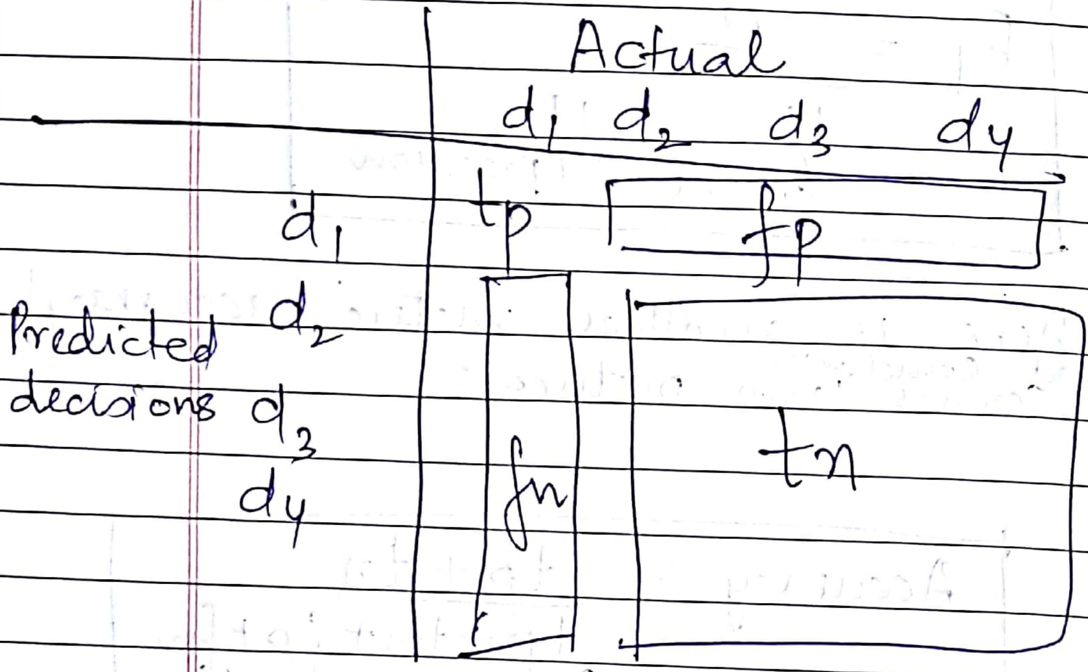
Iter^n 1: Seg1: Test
Rest: Train

Iter^n 2: Seg2: Test ⇒ we have 10 sets of rule
Rest: Train ~~sets~~ ~~soln~~

|

Iter^n 10 —

Confusion matrix



- we count the no. of cases where actual decision matches with predicted decision. ⟹ True positive

Predicted : True
Actual : No
] false positive

$$\boxed{\text{Recall} = \frac{tp}{tp+fn}}$$ → Also called sensitivity or true positive rate

$$\boxed{\text{Precision} = \frac{tp}{tp+fp}}$$ → also called as PPV (Positive predicted value)

generally as one ↑, other ↓.
⟹ we take harmonic mean of both

classmate

Date

Page

F1 score

$$\boxed{F_1 = \frac{2}{1/_{recall} + 1/_{precision}}}$$

There is another metric 'accurracy' that sometimes comes into picture :

$$\boxed{Accuracy = \frac{tp + tn}{\underbrace{tp + tn + fp + fn}_{\text{entire population}}}}$$

specificity

diagonistic odd rate

FP rate etc etc etc