Actual	Predictions	
1	1	✓
0	0	✓
1	0	
1	1	✓
0	1	
0	1	
0	0	✓
1	1	✓
1	0	
0	1	

Accuracy = 50%

⊙ Accuracy depends on the content of the problem.

⊙ Accuracy is not good with imbalance data.

Spam message is say 0.1% of total data & it is predicting for correct predictions for all ;
Accuracy = 99.9% (No use)

⊙ Accuracy considers 1 → 0 & 0 → 1 predictions as same.

CONFUSION MATRIX

Actual \ Prediction	1	0
1	TP	FN
0	FP	TN

Type 1 error

Type 2 error

For data

Actual \ Predictions	1	0
1	3	2
0	3	2

Actual \ Predictions	Spam	Not Spam
Spam	100	170
Not Spam	FP 20	700

M_1

Actual \ Predictions	Spam	Not Spam
Spam	100	190
Not Spam	FP 0	700

M_2

Accuracy (M_1) =

Accuracy (M_2) = 80%

But we will take M_2

Reason : Non spam को Spam बताना बड़ी problem है।
(Imp. document can be lost)

$(FP)_{M_1} > (FP)_{M_2}$ ⇒ $(\text{type-1 error})_{M_1} > (\text{type-1 error})_{M_2}$

Precision :- What proportion of predicted positive is truly positive.

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

We can ~~choose~~ calculate precision for any class (0 or 1)
[Precision for each class can be calculated]

		Predictions: Detected Cancer	Predictions: No Cancer
Actual	Cancer	1000	200 FN
	No Cancer	800	8000

M_1

		Predictions: Detected Cancer	Predictions: No Cancer
Actual	Cancer	1000	500 FN
	No Cancer	500	8000

M_2

Accuracy (M_1) = Accuracy (M_2) type-2-error

& Recall (M_1) $= \frac{1000}{1000+200}$ > Recall (M_2) $= \frac{1000}{1000+500}$

Better to take M1 → दोनों same Data दे तो क्या करेगा]

Recall :- Proportion of actual positive is correctly classified.

$$\text{Recall} = \frac{TP}{TP+FN}$$

We can calculate recall of all individual cases.

Input image is cat or dog.
Then problem is whether to consider precision or recall.
Use F1 score.

F1 score = Harmonic mean of precision & recall. → brings F1 to lower value among two

$$\text{F1 Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

	M_1	M_2
Precision	80	60
Recall	80	100
F1 score ⇒	.80	75

~~F1 (M_1)~~ F1 score (M_1) > F2 score (M_2)

∴ M_1 is better than M_2

MULTICLASS CLASSIFICATION :-

		Predicted: Dog	Cat	Rabbit	
Actual	Dog	25	5	10	=40
	Cat	0	30	4	= 34
	Rabbit	4	10	20	= 34

Accuracy = $\frac{25+30+20}{40+34+34} \times 100$ = 69.4%

Precision (Dog) = $\frac{25}{25+0+4} \times 100$ = 86.20%

Precision (Cat) =

Precision (Rabbit) =

Macroprecision :-

Macroprecision : If dataset is balanced then calculate average of all classes precision.

Precision (model) = $\frac{P(dog)+P(cat)+P(Rabbit)}{3}$ =

If dataset is not balanced then use weighted precision (SK learn)

Precision (model) weighted = $\frac{40}{108} \times P(dog) + \frac{34}{108} P(cat) + \frac{34}{108} P(Rabbit)$

Recall (Dog) = $\frac{25}{25+5+10}$ = 62.5%

Same as precision (average & weighted).

F1 Score (Dog) = $\frac{2 * Precision(Dog) * Recall(Dog)}{Precision(Dog) + Recall(Dog)}$

same as precision (average & weighted)

19.02.2024 Monday

NAIVE BAYE'S CLASSIFIER

① Supervised learning algorithm used for classification.
② It is a part of generative learning algorithm.

Learning Algorithm

Generative → ① Hidden Markov algo. ② Naive Baye's
Discriminative learning Algo. → ① Logistic Regression ② Support Vector Machine ③ Neural N/W

There is a strict assumption.

Naive Baye's learn the features of class
ex. Car & Dog is there
To which it is more similar to ?
Things that differentiate two classes.

This classifier works on the principle of conditional probability.

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

Prob. of A given B already happened

Q) Suppose a coin is tossed 2 times. What is the prob. that both the toss is head given that atleast one of the toss is head.

Sol: $S = \{HH, HT, TH, TT\}$

$B = \{HH, HT, TH\}$

$A = \{HH\}$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1/4}{3/4} = \frac{1}{3}$$

Bayes Theorem :-

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \Rightarrow P(A \cap B) = P(A|B) \cdot P(B) \longrightarrow (I)$$

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

$$\Rightarrow P(A \cap B) = P(B|A) \cdot P(A) \longrightarrow (II)$$

From (I) & (II):

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

$$\Rightarrow \boxed{P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}}$$

Disadvantage: Every feature is independent.

$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$ → Marginal Prob.

likelihood → prior → probability

Likelihood Prob. :- It is the relative probability of the observation occurring for each other. "Posterior probability."

Marginal Prob. :- Prob. of observation

Prob. of the event after the evidence has been

Naive ~~Classifier~~ Bayes Algo.

Suppose we have a dataset of n features $(x_1, x_2, x_3, \ldots, x_n)$:

↳ y is o/p. variable.

x_1	x_2	x_3	....	y
classroom				C_1
				C_2
				:
				:

→ Find probability

Yes $P(C_1|X) = P(Yes|X) = \frac{P(X|C_1) * P(C_1)}{P(X)}$

No $P(C_2|X) = P(No|X) = \frac{P(X|C_2) * P(C_2)}{P(X)}$

DT does not play importance.

Output class = $\underset{C_K}{\arg\max} \{ P(C_K) \cdot P(X|C_i) \}$

argmax → $\arg\max (P(C_1) \cdot P(x_{k1}), P(C_2) \cdot P(x_{k2}))$

Org. is different or ← value given

DATE
PAGE

$P(x|C_k) = P(\langle x_1, x_2, x_3, \ldots x_n\rangle / C_k)$

~~Naive Assumption~~

Naive Assumption

No pair of ~~~~ features are dim.

$P\langle x_1, x_2 \ldots x_n | C_k : P(x_1|C_1) * P(x_2/C_2 - * \ldots$ ~~(xn)~~

~~F(xn)~~ / $F(X_n) A_n)$

↳ All assumpt

& indirectly we ? naive classify

$$\prod^{n} P(x_i|C_2)$$

$P(X|C_k) = P(x_1, x_2 \ldots x_n) / k$

~~P(C_1|x \ldots)~~

input output class = $\arg\max \{P(C_k) \prod_{i=1}^{n} (x_i|C_k)\}$

PLAY Tennis ← features → o/p

Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	~~Yes~~ No
7	Overcast	Cool	Normal	Strong	~~No~~ Yes
8	Sunny	Mild	~~Cool~~ High	Weak	~~Yes~~ No
9	Sunny	Cool	Normal	Weak	~~No~~ Yes
10	Rain	Mild	Normal	Weak	Yes.

20.02.2024 Tuesday

$$\text{argmax}\left\{ P(C_1/\langle cond\rangle),\ P(C_2/\langle cond\rangle),\ P(C_3/\langle cond\rangle) \right\}$$

DATE
PAGE

11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	mild	High	Strong	No

9) < sunny, ~~Hot~~ Hot, Normal, Strong > for a day.
~~Will~~ Are we going to play tennis (B) or not?

Sol: Lookup tables: Learning process { conditional probability.

$$\text{Output class} : \underset{C_k}{\text{argmax}} \left\{ P(C_k) * \prod_{i=1}^{n} (x_i / C_k) \right\}$$

C_k → class

$$\text{argmax}\left\{ P(\text{Yes} / \langle \text{sunny, hot, Normal, strong} \rangle),\ P(\text{NO} / \langle \text{sunny, hot, normal, strong} \rangle) \right\}$$

Assuming that all the features are independent

$$P(\text{Yes}) * P(\text{sunny}/\text{Yes}) * P(\text{hot}/\text{Yes}) * P(\text{Normal}/\text{Yes}) * P(\text{Strong}/\text{Yes})$$

$$P(\text{NO}) * P(\text{sunny}/\text{NO}) * P(\text{hot}/\text{NO}) * P(\text{Normal}/\text{NO}) * P(\text{Strong}/\text{NO})$$

Prior probability — Conditional probability.

$P(\text{Yes}) = 9/14$ $\quad P(\text{NO}) = 5/14$

Likelihood Probability Tables — Lookup Tables.

Outlook

	Yes	NO
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

$$P(\text{outlook} = \text{sunny} / \text{yes}) = 2/9 = \frac{P(\text{sunny} \cap \text{yes})}{P(\text{yes})}$$

DATE
PAGE

Temp

	Yes	No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Wind

	Yes	No
Weak	6/9	2/5
Strong	~~9~~ 3/9	3/5

Humidity

	Yes	No
High	3/9	4/5
Normal	6/9	1/5

$$\arg\max \begin{cases} \frac{9}{14} \times \frac{2}{9} \times \frac{2}{9} \times \frac{6}{9} \times \frac{3}{9} = 0.007 \\ \frac{5}{14} \times \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{3}{5} = 0.01 \checkmark \end{cases}$$

Probability for 'No' is higher ∴ output = No.

Q) input <Overcast, Hot, Normal, Strong>

Sol: P(No | <input>) = 0   P(Overcast / No) = 0

But it is not logical.

Since dataset is small.

"Zero Probability Error"

Solution: Add atleast one hypothetical data entry

"Laplace Smoothing"

$$P(x_{ij}/c) = \frac{P(x_{ij} \text{ and } c) + \alpha}{N + \alpha K}$$

α = Smoothing parameter (hyper parameter).

N = No. of test examples with o/p = c.

K = No. of possible distinct values for feature j.
↳ Overcast   3 classes (Sunny, Overcast, Rainy)

$\alpha = 1$ ↳ New

P(Overcast / No) = $\frac{0+1}{5+1\times 3}$ कितने No: 0 | 1

$= \frac{1}{8}$

सब पर लगा दो ये formula |

or

जहाँ zero वहाँ change कर दो |

21.02.2024 Wednesday.

$P(\text{Sunny}/\text{No}) = \frac{3+1}{5+1\times3} = \frac{4}{8}$ → optional to change for some.

$P(\text{Rain}/\text{No}) = 3/8$

$P(\text{Overcast}/\text{No}) = \frac{1}{8}$

$\Sigma P_i = 1$ (Sum = 1)

$\therefore P(\text{No}) = 8/17$ $\quad P(\text{Yes}) = 9/17$ (Not compulsory for this change)

$P(\text{No}) * P(\text{O}/\text{No}) * P(\text{H}/\text{No}) * P(\text{N}/\text{No}) * P(\text{Str.}/\text{No})$

$\frac{8}{17} \times \frac{1}{8} \times \frac{2}{5} * \frac{1}{5} * \frac{3}{5} = 2.823 \times 10^{-3}$

$= 0.0028$

arg max

$P(\text{Yes}) * P(\text{O}/\text{Yes}) * P(\text{H}/\text{Yes}) * P(\text{N}/\text{Yes}) * P(\text{Str.}/\text{Yes})$

$\frac{9}{17} \times \frac{4}{9} \times \frac{2}{9} \times \frac{6}{9} \times \frac{3}{9} = 0.011$

α की higher value पर "underfit" हो जाएगा।
α की lower value पर "overfit" हो रहा था।

Advantages:-

1. Simple to implement & use.

Disadvantage:-

1. Zero probability error.
2. All situations can't be modelled.

~~Non linear~~

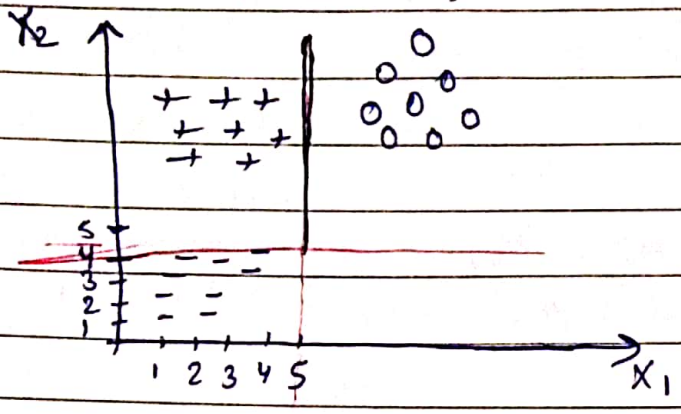
DECISION TREE

1. { ↳ classification
   { ↳ Regression
2. Non linear model
3. Dividing feature space on the basis of feature

"Decision trees works by dividing the data in regions based on "if-then" type of questions."

4. Graphically by asking many "if then" questions a decision tree

can divide up the feature space using little segments of vertical & horizontal lines.



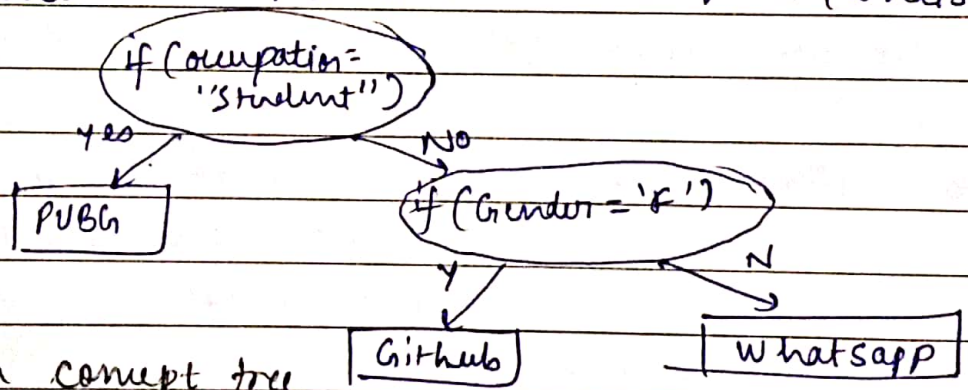

```
If (X2 <= 4)
    print ("- Class")
else if (X1<5)
    print ("+ Class")
```

Q)

Gender	Occupation	Suggestion
F	Student	PUBG
F	Programmer	Github
M	programmer	Whatsapp
F	programmer	Github
M	student	PUBG
M	student	PUBG

Sol:
```
if (occupation = "student")
    print ("PUBG")
else
    if (Gender = 'F')
        print ("Github")
    else
        print ("Whatsapp")
```

Sol: Control Flow Diagram



Decision Tree is a concept tree which summarizes the information contained in the training dataset in the form of a tree structure.

Structure of a Decision Tree :-

i) Root node → Top most node of tree is called root node

ii) Internal node / Decision node → Decision Node represents a test condition of an input feature & O/P of this test condition are branches come out from the decision node.

iii) Branches

iv) Leaf Nodes

⇒ Recursive greedy algorithm
↳ Decision Tree Algo.

⇒ Decision tree is representing the complete hypothesis test.

Max. branches from node = max no. of distinct values a feature can have.

Decision node can have maximum outdgree (branches) equal to possible values of features used to test in the node.

Termination Condition :-

Leaf node represents the outcome of the decision path.
Labels of the leaf nodes are the different classes a data point can belong to.

CART → Binary trees
↳ Categorical + Continuous data

ID3 → Only Non Binary trees
↳ only Categorical data

C4.5 → Improved ID3

Training process

(1) Tree is constructed in a top-down fashion. It starts from the root node.
(2) At the root we have entire dataset and we need to find the test split attribute.
(3) After split each child represents splitted dataset based on values of the features.
(4) This process is recursive and continued until we endup in the last level of the tree or finding a leaf tree.
↳ we can decide previously only that we need only 20 levels only.
or

3 major stopping criteria :-

(1) Leaf Node
(2) Node already decided (say 20 levels early)
(3) If data at any level less than say 5% of total data (say) then stop division → majority wins (in this case)

Else we will get overfitted tree.

< 5% → A = 60%, B = 40% → A Class.

Decision tree Induction Algorithms :-

① ID3 :-

This algorithm uses "Entropy" to measure the purity / impurity of the dataset at a node. (Any node)

$$\text{Entropy} = -\sum_{k} P(C_k) \cdot \log_2(P(C_k))$$

For 2 classes - $\text{Entropy} = -P(C_1) \cdot \log(P(C_1)) - P(C_2) \cdot \log(P(C_2))$

~~Single class → Entropy~~

For a 2 class problem, the min. entropy is 0 and max is 1

~~single~~ 100%, 0% 50%, 50%

For more than 2 classes the min entropy is 0 but max can be greater than 1. (विभाजन में इसका log Base 2 होता है)

Information Gain = Entropy before split - Entropy after split

(How much entropy is reduced on splitting on basis this.)

Entropy after split = weighted sum of entropies of children

$$\text{Weight of child} = \frac{\text{No. of examples in child node}}{\text{total No. of examples}}$$

Color	Diameter	Fruit
Green	3	apple
Red	3	apple
Red	1	grapes
Yellow	3	orange ~~grapes~~
Green	1	~~Grapes~~

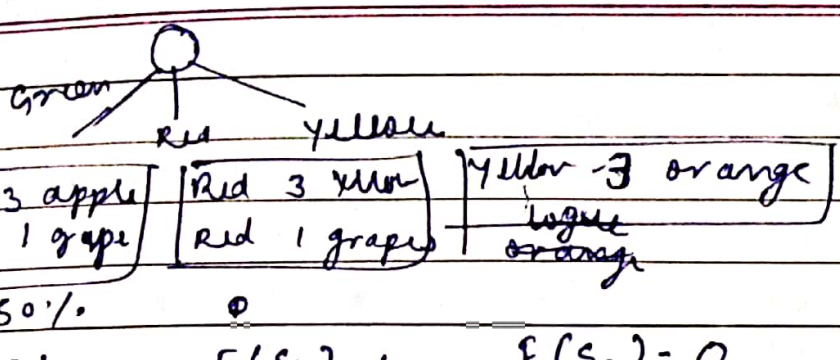
Sol:

$$E(S)_c = -P_k \log_2(P(C_k))$$

$$= -\sum_{i=A} P(C_A) \cdot \log_2(P(C_A))$$

$$= -\left\{ \frac{2}{5} \log_2\left(\frac{2}{5}\right) + \frac{2}{5} \log_2\left(\frac{2}{5}\right) + \frac{1}{5} \log_2\left(\frac{1}{5}\right) \right\}$$

$$= 1.52$$



50% 50%

$E(S_1) = 1$ $E(S_2) = 1$ $E(S_3) = 0$

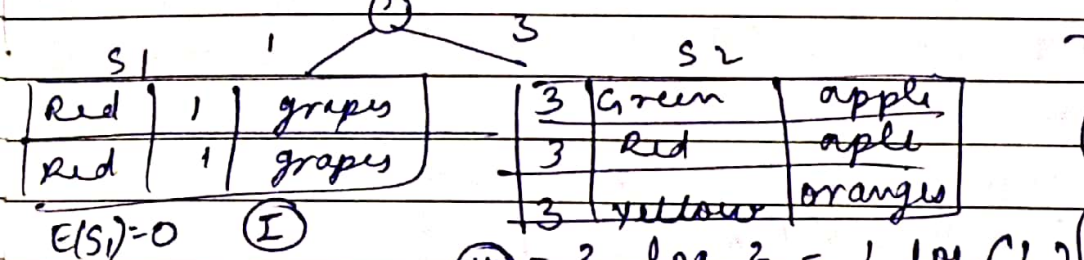
$\left(-\frac{1}{2}\log_2\frac{1}{2}\right)\times 2$

weighted sum of entropies =

$\frac{2}{5} \times E(S_1) + \frac{2}{5} \times E(S_2) + \frac{1}{5} \times E(S_3)$

$= 0.8$

Information Gain (color) $= 1.52 - 0.8 = 0.72$



$E(S_1) = 0$ (I)

(II) $-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\left(\frac{1}{3}\right)$

$E(S_2) = 0.91$

Wt. sum of entropies $= \frac{2}{5} E(S_1) + \frac{3}{5} E(S_2)$

$= \frac{5}{5}$

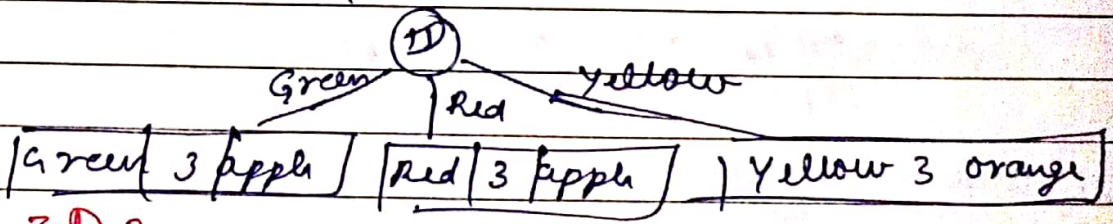
0.54

(III) IG (diameter) $= 1.52 - 0.54 = 0.98$

Clearly IG (diameter) > IG (color)

∴ Division on diameter

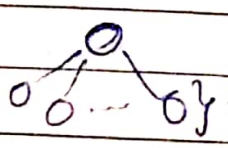
(I): Leaf Node
Single Target Class



Problem with ID3

(1) If scholar number is a feature then we will get all leaf nodes in single level. Highly ~~bias~~ biased for feature.

Attribute is having more number of ~~ins~~ different instances. It will dominate information gain (max. IG for such feature)



28.02.2024
Wednesday.

DATE
PAGE

C4.5 Algorithm

① It is an improvement over ID3

② ID3 is biased towards attributes with more diff. values. ex. Scholar number of student example.

③ To overcome this bias issue C4.5 to ~~not only~~ identify uses a purity measure 'Gain Ration" to identify the best split attribute.

④ In C4.5 algorithm, the Information Gain ~~measuring in~~ measure used in ID3 algorithm ~~for~~ is normalized by computing another factor. "T ~~is~~ is a dataset at node."

$$\text{Gain Ration}(A) = \frac{\text{Information Gain}(A)}{\text{split Info}(T, A)}$$

$$\text{Split Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \log_2 \frac{|A_i|}{|T|}$$

where, the attribute A have got distinct value $(a_1, a_2, a_3, \ldots a_v)$

a_1 Instances of a_1 In Entropy := instances of values.

$$= -\frac{|a_1|}{|T|} \log \frac{|a_1|}{|T|} - \frac{|a_1|}{|T|} \log \frac{|a_2|}{|T|} \ldots$$

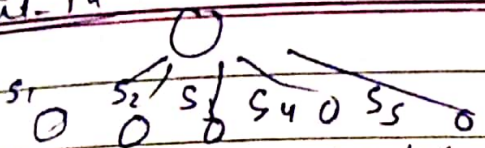
a)

Fruit_ID	Color	Diameter	Fruit
101	~~Green Grapes~~	3	apple
102	Red	3	apple
103	~~Yellow~~	1	grapes
104	Yellow	3	orange
105	Green	1	grapes

Sol: Entropy $E(S) = 1.52$

Calculate entropy, Information gain, split info, gain ratio for all attributes.

fruit_id



IG = 1.52 (Fruit_ids)

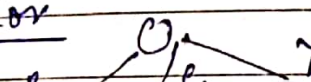
E(S1) = 0 - E(S2) = ... E(S5) = 0

Information Gain (Fruit_id) = $1.52 - \left(\frac{1}{5} \times 0 + \frac{1}{5} \times 0\right) = 1.52$

Split info (fruit_id) = $-\left\{\left(\frac{1}{5} * \log_2 \frac{1}{5}\right) \times 5\right\}$ $\log_2 \frac{1}{5} = \log 5 + \frac{1}{5} \log 5$ ≈ 2.3219

Gain ratio (fruit_id) = $\frac{1.52}{2.3219} = 0.656$

color



Split ratio = $-\left\{\frac{2}{5} \log_2 \frac{2}{5} + \frac{2}{5} \log_2 \frac{2}{5} + \frac{1}{5} \log_2 \frac{1}{5}\right\}$ $= 1.5$

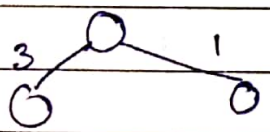
IG(color) = 0.72

E(color) = 0.8

Least class

Gain ratio = $\frac{0.72}{1.5} = 0.48$

Diameter

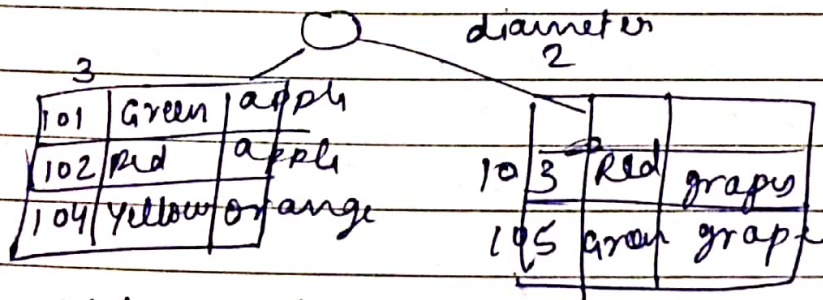


Split ratio = $-\left(\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5}\right)$ $= 0.97$

IG(diameter) = 0.98

Gain ratio = $\frac{0.97}{0.97} = 1$

Clearly, Gain ratio (diameter) is highest.



We were able to bypass fruit_id from selection which was the problem in ID3.

04.02.2024
Monday

Gini_Index ∈ [0, 0.5]
clean cut — most confusion

DATE
PAGE

CART Algorithm

Classification and Regression Tree
↳ standard deviation instead of entropy

① It constructs the tree as a binary tree by recursively splitting a node into 2 nodes.
(If we have 3 classes then combine 2 classes so that we have 2 children only)

② Gini Index

$$\text{Gini Index} = 1 - \sum_{i=1}^{k} P_i^2$$

P_i = Probability that a data instance belongs to class C_i

min value = 0
max value = 0.5

$$\text{Gini Index} = 1 - \sum_{i=1}^{k} [P(C_i)]^2$$

Entropy, same

③ Every attribute is considered as a binary attribute which split the nodes into 2 subsets S_1 & S_2

$$\text{Gini Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2)$$

Weighted Sum of Gini Index

④ ΔGINI is calculated for each attribute

~~ΔGINI(A) = Gini~~

$$\Delta\text{Gini}(A) = \text{Gini}(T) - \text{Gini}(T, A)$$

Attribute 'A' used to split the tree 'T'

If an attribute 'A' has 3 distinct values (a_1, a_2, a_3)
Then possible subsets are $\{\}, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_2, a_3\}, \{a_3, a_1\}, \{a_1, a_2, a_3\}$

X Y

combination of division		
$\{a_1\}$	$\{a_2, a_3\}$	→ Gini Index (T, A) → G_1
$\{a_2\}$	$\{a_1, a_3\}$	→ Gini Index (T, A) → G_2
$\{a_3\}$	$\{a_1, a_2\}$	→ Gini Index (T, A) → G_3
X	Y	

3 subsets are possible

{a1, a2} {a3, a4} For 4 {a1, a2} {a3, a4}

Select min({G1, G2, G3}) → to split the node.

In CART algorithm, we need to compute the best splitting attribute & the best split subset i in the chosen attribute.

Q)

Color	Diameter	Fruit
Green	3	Apple
Red	3	Apple
Red	1	Grapes
Yellow	3	Orange
Green	1	Grapes

Create Decision Tree using CART algorithm.

Sol: $Gini_{index}(T) = 1 - \sum_{i=1}^{3} P_i^2 = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2$ → $(P(Apple))^2 + P(Grapes)^2 + P(Orange)^2$

$= 1 - \frac{4}{25} - \frac{1}{25} - \frac{4}{25} = \frac{16}{25} = 0.64$

Consider Color :-

Color has 3 possible values,

So the combination of subsets for split.

G: Green
R: Red
Y: Yellow

	Gini
(i) {G}, {R, Y}	0.6
(ii) {R}, {G, Y}	0.6
(iii) {Y}, {G, R}	0.4 ✓

(i) {G} {R, Y}

{G} {R, Y}

Green 3 Apple
Green 1 Grapes
(S_1)

Red 3 Apple
Red 1 Grapes
Yellow 3 Orange
(S_2)

$Gini(S_1) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{2}{4} = 0.5$

$Gini(S_2) = 1 - \left(\frac{1}{3}\right)^2 \times 3 = \frac{2}{3}$

$Gini(T, (i)) = \frac{2}{5} \times \frac{1}{2} + \frac{3}{5} \times \frac{2}{3} = \frac{1}{5} + \frac{2}{5} = \frac{3}{5} = 0.6$

(ii) {R} {G, Y}

R 3 Apple
R 1 Grapes
(S_1)

G 3 Apple
Y 3 Orange
G 1 Grapes
(S_2)

$Gini(S_1) = \frac{1}{2} = 0.5$

$Gini(S_2) = 0.\frac{2}{3} = 0.67$

$Gini(T_2, (ii)) = \frac{2}{5} \times \frac{1}{2} + \frac{3}{5} \times \frac{2}{3} = 0.6$

Weighted Gini

(iii) {2} {0, 3}

4 3 orange (S_1)

G	3	Apple
R	3	Apple
R	1	Grapes
G	1	Grapes

(S_2)

$Gini(S_1) = 1 - 1^2 = 0$

$Gini(S_2) = 1 - \left(\frac{1}{2}\right)^2 \times 2 = \frac{1}{2} = 0.5$

$Gini(T, (iii)) = \frac{1}{5} \times 0 + \frac{4}{5} \times \frac{1}{2} = \frac{2}{5} = 0.4$

$\Delta Gini(color) = Gini(T) - Gini(T, color)$

$= 0.64 - 0.4$

$\Rightarrow \boxed{\Delta Gini(Color) = 0.24}$

$Giniinden(T, Color) = 0.4$ ← minimum value

Diameter

? {3}

G	Apple	Green	3	Apple
R	1 Grapes	Red	3	Apple
G	1 Grapes	Yellow	3	Orange

(S_1) (S_2)

$Gini(S_1) = 1 - 1^2 = 0$

$Gini(S_2) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2$

$= 1 - \frac{5}{9} = \frac{4}{9}$

$= 0.445$

weighted Gini → $Gini(T, Diameter) = \frac{2}{5} \times 0 + \frac{3}{5} \times 0.445$

$= 0.27$

$\Delta Gini(Diameter) = Gini(T) - Gini(T, Diameter)$

$= 0.64 - 0.27 = 0.37$

$\Rightarrow \boxed{\Delta Gini(Diameter) = 0.37}$

Clearly, $\boxed{\Delta Gini(Diameter) > \Delta Gini(color)}$

$\Rightarrow$ Diameter is used to divide the node.

05.03.2024 Tuesday

Some books, CART Algo. ⇒ No need for binary tree

Reason: ① Hypothesis (Maybe that works better in their case)

{1} 0 {3}

Red	1	Grapes
Green	1	Grapes

Leaf Node

G	3	Apple
R	3	Apple
Y	3	Orange

Same process ahead

{Y}, {G, R}.

Naive Baye's give equal weightage to every attribute but Decision tree does not give equal attributes.

SUPPORT VECTOR MACHINE (SVM)

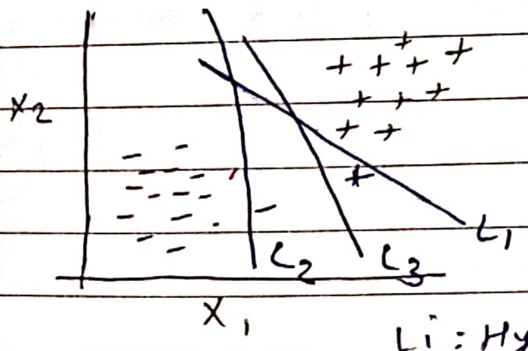
↳ Pairing dimension of data.

supervised learning algorithm.
- classification
- Regression

Linear → line can separate the data into 2 halves.

Robust to outliers

Work with non linear data as well.



L_3 unable to classify ⊕ blue as ⊕ instead classify it as ⊖

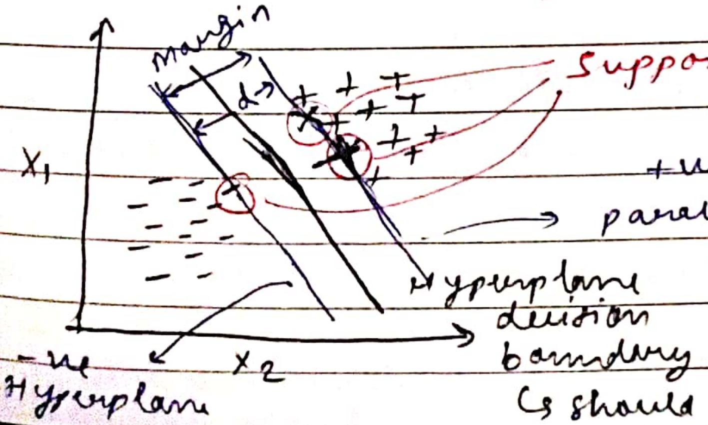
Similar problem with L_2 ⊖ blue as ⊖

[K nearest neighbour would have given correct answer]

Li: Hyperplane

Maximum margin hyperplane

$d_1 + d_2 = x$
$\frac{d_1 + d_2}{2}$ = $d_1\ d_2$

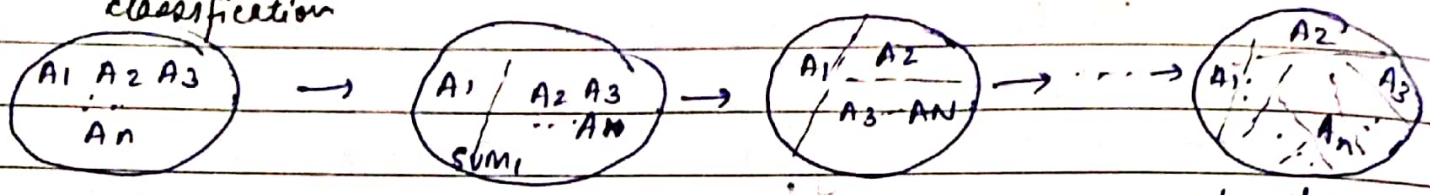


cost function is margin = d

↳ should be exactly in middle of both hyperplanes

dist^n bet^n line & vector
dist^n two || line
magnitude of a vector
difference of 2 vectors

SVM here is binary classification but can be extended for multiclass classification



SVM for n Class classification.

n class classification

The basic idea of SVM is distance betⁿ hyperplane & nearest datapoint should be maximum & hyperplane should avoid misclassification of data.

"Margin" is defined as the amount of space between the two classes as defined by the hyperplane.

The focus of SVM is to obtain "maximal margin classifier".

Types of SVM :-

① Hard Margin SVM → Linearly separable data (previous page ex.)

② Soft Margin SVM → Non-linearly separable data.

Support vectors :- Data points that fall on the supporting hyperplanes

① Hard Margin SVM as Binary classifier :-

A SVM implements a binary classifier this means there are only 2 classes say +1 and −1.

This idea can be extended to multiclass SVM.

Let us consider a dataset with n-dimensions

i/p : $x_1, x_2, x_3, \ldots, x_n$

o/p : $y = \{-1, +1\}$

Hyperplane is a n-dimensional generalization of a line, so equation of Hyperplane is $w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n + w_0 = 0$

$\Rightarrow \boxed{\bar{w} \cdot \bar{x} + w_0 = 0}$ ⭐⭐⭐

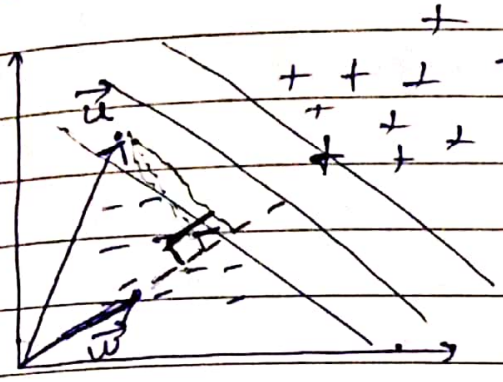
$5 \rightarrow 3x_1 + 4x_2 = 0$
$3 + 3x_1 + 4x_2 = 0$
Parallel lines

Assume
$w_0 + w_1x_1 + w_2x_2 = 1$ (+ve Hyperplane)
$w_0 + w_1x_1 + w_2x_2 = 0$ (Hyperplane)
$w_0 + w_1x_1 + w_2x_2 = -1$ (−ve Hyperplane)

26.03.2024
Wednesday

DATE
PAGE

SVM Problem formulation



vector $w = \vec{w}$

$\vec{w} \perp$ hyperplane

$\vec{w}, \vec{u} \geq c$ + ve

$\vec{w}.\vec{u} + w_0 \geq 0$ + ve

$\vec{w}$ & w_0 are not known.

Decision Rule

If $\bar{x}_i$ is +ve point then
$\bar{w}.\bar{x}_i + w_0 \geq 1$

if $\bar{x}_i$ is a -ve point then
$\bar{w}.\bar{x}_i + w_0 \leq -1$

→ Assumption

$h(x) = \bar{w}\bar{x} + w_0$

$\text{sign}(h(x))$

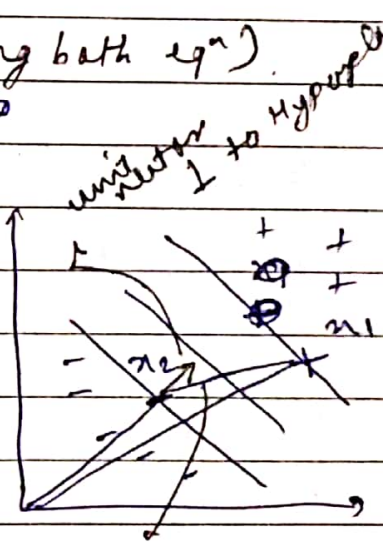
$y_i = \{+1, -1\}$

$\therefore \quad y_i(\bar{w}.\bar{x}_i + w_0) \geq 1$ (combining both eqn)

Constraints on traning data points

$\Rightarrow y_i(\bar{w}.\bar{x}_i + w_0) - 1 \geq 0 \longrightarrow$ (I)

Suppose x_1 is the +ve support vector
& x_2 is the -ve support vector



$\text{margin} = (\bar{x}_1 - \bar{x}_2) \cdot \frac{\bar{w}}{|w|}$

$\Rightarrow \text{margin} = \frac{\bar{x}_1\bar{w} + \bar{x}_2\bar{w}}{|w|} \hookrightarrow$ (II)

From (I):
$y_i(\bar{w}.\bar{x}_i + w_0) - 1 \geq 0$ → equality exists for support vectors

(I) & (II) $\boxed{y_i(\bar{w}\bar{x}_i + w_0) - 1 = 0}$

For x_1 & $y_1 = \{+1\}$ +ve SV (support vector)

$(\bar{w}\bar{x}_1 + w_0) = +1 \Rightarrow \bar{w}_1\bar{x}_1 = 1 - w_0 \hookrightarrow$ (III)

11.03.2024
Monday

DATE
PAGE

For x_2, $y_2 = -1$ ① - we SV

$-1 (\bar{w}_0 \bar{x}_2 + w_0) = 1 \Rightarrow = 0$

$\Rightarrow -\bar{w}\bar{x}_2 = +1 + w_0 \longrightarrow$ (IV)

put (III) & (IV) in (II);

$$\text{margin} = \frac{(1 - w_0) + (1 + w_0)}{|w|}$$

$$\Rightarrow \boxed{\text{margin} = \frac{2}{|w|}} \quad (\star\star\star)$$

↳ max margin $\Rightarrow$ $(|w|)$ minimize $\Rightarrow$ minimize $\frac{1}{2}(|w|)^2$.

Dual function ⇒ maximize ⇒ then original function minimizes.

Optimization Technique

Hard margin SVM as an Optimization problem.

$$\underset{w, w_0}{\text{Minimize}} \quad \frac{1}{2} |\bar{w}|^2$$

subject to the constraint

$$y_i (\bar{w} \cdot \bar{x}_i + w_0) - 1 \geq 0 \quad \forall x_i \in D$$

Its is an optimization problem with constraint.

Lagrangian Multiplier are one way of solving the optimization problem with respect to equality constraints.

$$L(w, w_0, \alpha) = \frac{1}{2} |w|^2 - \sum_{i=1}^{m} \alpha_i [y_i (\bar{w} \cdot \bar{x}_i + w_0) - 1]$$

↳ Lagrangian multiplier.

Actual f" is minimization ∴ we will maximize

maximization

→ Find α & then find pairwise vector multiplication

α is Lagrangian Multiplier

maximize $L(w, w_0, \alpha)$

$$L = \frac{1}{2} ||\bar{w}||^2 - \sum_{i=1}^{m} \alpha_i y_i \bar{w}_i \bar{x}_i \bar{x}_i - \sum_{i=1}^{m} \alpha_i y_i w_0 + \sum_{i=1}^{m} \alpha_i \longrightarrow (I)$$

$$\frac{dL}{dw_0} = 0 + 0 + \sum_{i=1}^{m} \alpha_i y_i + 0 = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0 \longrightarrow (II)$$

$$\frac{dL}{d\bar{w}} = \bar{w} - \sum_{i=1}^{m} \alpha_i y_i \bar{x}_i - 0 + 0 = 0$$

$$\Rightarrow \bar{w} = \sum_{i=1}^{m} \alpha_i y_i \bar{x}_i \longrightarrow (III)$$

put the value of (II) in (I):

$$L(w, w_0, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^{m} \alpha_i y_i \bar{w}.\bar{x}_i + w_0 \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i$$

put (III):

$$\Rightarrow L(w, w_0, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^{m} \alpha_i y_i \bar{w}.\bar{x}_i + \sum_{i=1}^{m} \alpha_i$$

substitute (III):

$$\Rightarrow L(w, w_0, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^{m} \alpha_i y_i \left(\sum_{j=1}^{m} \alpha_j y_j \bar{x}_j\right) \bar{x}_i + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2}|w|^2 - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j + \sum_{i=1}^{m} \alpha_i$$

From (II):

$$= \frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i \bar{x}_i \sum_{j=1}^{m} \alpha_j y_j \bar{x}_j - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j + \sum_{i=1}^{m} \alpha_i$$

$\bar{w}.\bar{w}$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j + \sum_{i=1}^{m} \alpha_i$$

$$\Rightarrow \boxed{L(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j}$$

(I) (II) (III) (IV)

(V) $\frac{dL}{d\alpha_1} = 0, \frac{dL}{d\alpha_2} = 0, \frac{dL}{d\alpha_3} = 0, \ldots, \frac{dL}{d\alpha_m} = 0$

m equations, m unknown variables [= No. of data points]

can be solve for $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m$ (theoretically)

$\alpha_i > 0$ for support vectors
$\alpha = 0$ for not support vectors.

$$\max(L(\alpha)) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j$$

subject to $\alpha_i \geq 0 \quad \forall \; i = 1, 2, \ldots, m$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \quad \text{(from (II))}$$

From (III): $\bar{w} = \sum_{i=1}^{m} \alpha_i y_i x_i$

Equality for support vectors:- (found earlier)

$$y_i (\bar{w} \cdot \bar{x}_i + w_0) = 1$$

$\Rightarrow y_i y_i (\bar{w} \cdot \bar{x}_i + w_0) = y_i$ [Multiplied by y_i]

$\Rightarrow \bar{w} \cdot \bar{x}_i + w_0 = y_i$

$\Rightarrow w_0 = y_i - \bar{w} \bar{x}_i$

$y_i = \{+1, -1\}$
$y_i \cdot y_i = y_i^2$
$\therefore y_i^2 = \{+1\}$
$\Rightarrow y_i^2 = +1$

$\Rightarrow$ support vectors.

$$\Rightarrow w_0 = \frac{1}{S} \sum_{i=1}^{S} (y_i - \bar{w} \cdot \bar{x}_i)$$

Q) Three data point as support vectors

	x_1	x_2	y
1	1	2	-1
2	-1	2	-1
3	-1	-2	1

Sol:

$$L(\alpha) = \sum_{i=1}^{3} \alpha_i - \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j$$

$$= \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} \sum_{i=1}^{3} (\alpha_i \alpha_1 y_i y_1 \bar{x}_i \cdot \bar{x}_1 + \alpha_i \alpha_2 y_i y_2 \bar{x}_i \bar{x}_2 + \alpha_i \alpha_3 y_i y_3 \bar{x}_i \bar{x}_3)$$

m.1

$$= \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}[(\alpha_1 \alpha_1 y_1 y_1 \bar{x}_1 \cdot \bar{x}_1 + \alpha_2 \alpha_1 y_2 y_1 \bar{x}_2 \cdot \bar{x}_1 + \alpha_3 \alpha_1 y_3 y_1 \bar{x}_3 \bar{x}_1)$$
$$+ (\alpha_1 \alpha_2 y_1 y_2 \bar{x}_1 \bar{x}_2 + \alpha_2 \alpha_2 y_2 y_2 \bar{x}_2 \cdot \bar{x}_2 + \alpha_3 \alpha_2 y_3 y_2 \bar{x}_3 \bar{x}_2$$
$$+ \alpha_1 \alpha_3 y_1 y_3 \bar{x}_1 \bar{x}_3 + \alpha_2 \alpha_3 y_2 y_3 \bar{x}_2 \bar{x}_3 + \alpha_3 \alpha_3 y_3 y_3 \bar{x}_3 \bar{x}_3$$

$\hookrightarrow$ 9 terms

12-03-2024 Tuesday

M.2

$y_i x_i \quad y_j x_j$

Transpose $(y_i x_i) \cdot (y_i x_i)^T$

$y_i \cdot x_i \quad * \quad (y_i x_i)^T$

x_1 $\begin{bmatrix} -1 & -2 \\ 1 & -2 \\ -1 & -2 \end{bmatrix}$ * $\begin{matrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \begin{bmatrix} -1 & 1 & -1 \\ -2 & -2 & -2 \end{bmatrix} \end{matrix}$ m m

y	x_1	x_2
-1	1	2
-1	-1	2
1	-1	-2

$x_1 = \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j x_i x_j$

$x = a_1 x_1 + b_1 x_2$

$y_i x_i = (y_i a_i) x_1 + (y_i b_i) x_2$

i ↓ / j →	1	2	3
	$x_1 x_2$	$x_2 x_3$	
1	15	3	5
2	3	5	3
3	5	3	5

→ matrix Representation of complex calculation

$\sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j x_i x_j$

$L_D(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} (5\alpha_1\alpha_1 + 3\alpha_1\alpha_2 + 5\alpha_1\alpha_3 + 3\alpha_2\alpha_1 + 5\alpha_2\alpha_2 + 3\alpha_2\alpha_3 + 5\alpha_3\alpha_1 + 3\alpha_3\alpha_2 + 5\alpha_3\alpha_3)$

$\Rightarrow L_D(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} [5\alpha_1^2 + 3\alpha_1\alpha_2 + 5\alpha_1\alpha_3 + 3\alpha_2\alpha_1 + 5\alpha_2^2 + 3\alpha_2\alpha_3 + 5\alpha_3\alpha_1 + 3\alpha_3\alpha_2 + 5\alpha_3^2]$

$\frac{dL}{d\alpha_1} = 1 - \frac{1}{2} (10\alpha_1 + 3\alpha_2 + 5\alpha_3 + 3\alpha_2 + 5\alpha_3) = 0$

$= 1 - \frac{1}{2} (10\alpha_1 + 6\alpha_2 + 10\alpha_3) = 0$

$\Rightarrow 5\alpha_1 + 3\alpha_2 + 5\alpha_3 = 1 \longrightarrow (I)$

$\frac{dL}{d\alpha_2} = 1 - \frac{1}{2} (3\alpha_1 + 3\alpha_1 + 10\alpha_2 + 3\alpha_3 + 3\alpha_3) = 0$

$= 3\alpha_1 + 5\alpha_2 + 3\alpha_3 = 1 \longrightarrow (II)$

$\frac{dL}{d\alpha_3} = 1 - \frac{1}{2} (5\alpha_1 + 3\alpha_2 + 5\alpha_1 + 3\alpha_2 + 10\alpha_3) = 0$

$5\alpha_1 + 3\alpha_2 + 5\alpha_3 = 1 \longrightarrow (III)$

$\Rightarrow$ (I) − (II):

(I) × 3 − (II) × 5

$15\alpha_1 + 9\alpha_2 + 15\alpha_3 = 3$

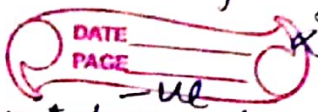
$15\alpha_1 + 25\alpha_2 + 15\alpha_3 = 5$

$16\alpha_2 = 2 \Rightarrow \boxed{\alpha_2 = \frac{1}{8}}$

$5\alpha_1 + 5\alpha_3 = 1 - \frac{3}{8} = \frac{5}{8}$

$\Rightarrow \boxed{\alpha_1 + \alpha_3 = \frac{1}{8}}$

⊙ kpt optimization
⊙ duall satisfactory cond^n $\alpha \geq 0$



$\alpha_1 + \alpha_3 = \frac{1}{8}$ $\quad$ $\boxed{\alpha_2 = \frac{1}{8}}$ → -ve hyperplane

α_2 & α_1 are related to -ve hyperplane

∴ we can make $\boxed{\alpha_1 = 0}$

∴ $\boxed{\alpha_3 = \frac{1}{8}}$ (+ve hyperplane)

$$\bar{w} = \sum_{i=1}^{3} \alpha_i y_i \bar{x}_i = 0 + \frac{1}{8}(-1)$$

$$= \alpha_1 y_1 \bar{x}_1 + \alpha_2 y_2 \bar{x}_2 + \alpha_3 y_3 \bar{x}_3$$

$$= 0 + \frac{1}{8}(-1)[-1 \quad 2] + \frac{1}{8}(1)[-1 \quad 2]$$ ↳ vector

$$= \frac{1}{8}[1 \quad -2] + \frac{1}{8}[-1 \quad -2]$$

$$= \frac{1}{8}(0 \quad -4) = [0 \quad -\frac{1}{2}]$$

w_0 $\quad$ w_2

$\therefore \boxed{\bar{w} = [0 \quad -\frac{1}{2}]}$ Ans.

$\alpha_1 = 0$ ∴ y_1 का मान नहीं लेंगे।

y_2 & y_3 ही होगा।

$$w_0 = \frac{1}{S} \sum_{i=1}^{S} (y_i - \bar{w}\bar{x}_i)$$ → evaluate it for support vector 1 & 2.

$$= \frac{1}{2}\left(y_2 - \bar{w}\bar{x}_2 + y_3 - \bar{w}\bar{x}_3\right)$$

$$= \frac{1}{2}\left(-1 - [0 \quad -\frac{1}{2}]\cdot[-1 \quad 2] + 1 - [0 \quad -\frac{1}{2}]\cdot[-1 \quad -2]\right)$$

$$= \frac{1}{2}(-1 + 0 + 1 + 1 - 1) = 0 \Rightarrow \boxed{w_0 = 0}$$

$\bar{w} = [0 \quad -\frac{1}{2}]$
$\quad\quad w_1 \quad w_2$

$w_1 x_1 + w_2 x_2 + w_0 = 0$

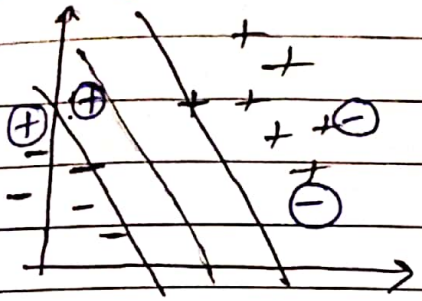
$\Rightarrow 0 x_1 + -\frac{1}{2} x_2 + 0 = 0$

$\Rightarrow \boxed{x_2 = 0}$ Final eq^n

Hard margin SVM → rigid to outliers

Flexibility is required → Soft margin SVM

Soft Margin SVM



All +ve should be on one side & all -ve sides on other

Not rigid on this cond-

⊙ Hard margin SVM is very rigid for 2 constraints of SVM:-
(i) Decision boundary should be as far as from the nearest data point.
(ii) Avoid misclassification.

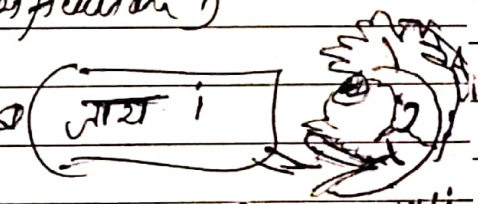
⊙ But the real world problems are complex and often datasets are affected by noise & outliers.
[data will not be completely linearly separable]

⊙ But dataset is almost linearly separable, (Relaxation in margin. Balance b/w margin & misclassification)

$$\min_{w_1, w_0} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

$\frac{1}{2}||w||^2$: distance error

$C\sum_{i=1}^{n} \xi_i$: classification error



Parameterization.

$\xi_i \Rightarrow$ Slack variable

subject to constraint $\xi_i \geq 0 \ \forall x_i$

$\xi_i = 0$: correctly classified

$\xi_i > 0$: misclassification

$y_i(\bar{w}.\bar{x}_i + w_0) \geq 1 - \xi_i$

hyperparameter → you can tune it

C ↑ more ⇒ more important to misclassification error

C ↓ less ⇒ distance error important ⇒ Margin error important

Lagrangian dual of this is similar to Hard margin SVM only change $\alpha >= 0$ & $\alpha < C$