Chapt - 8

Date :
Page No. :

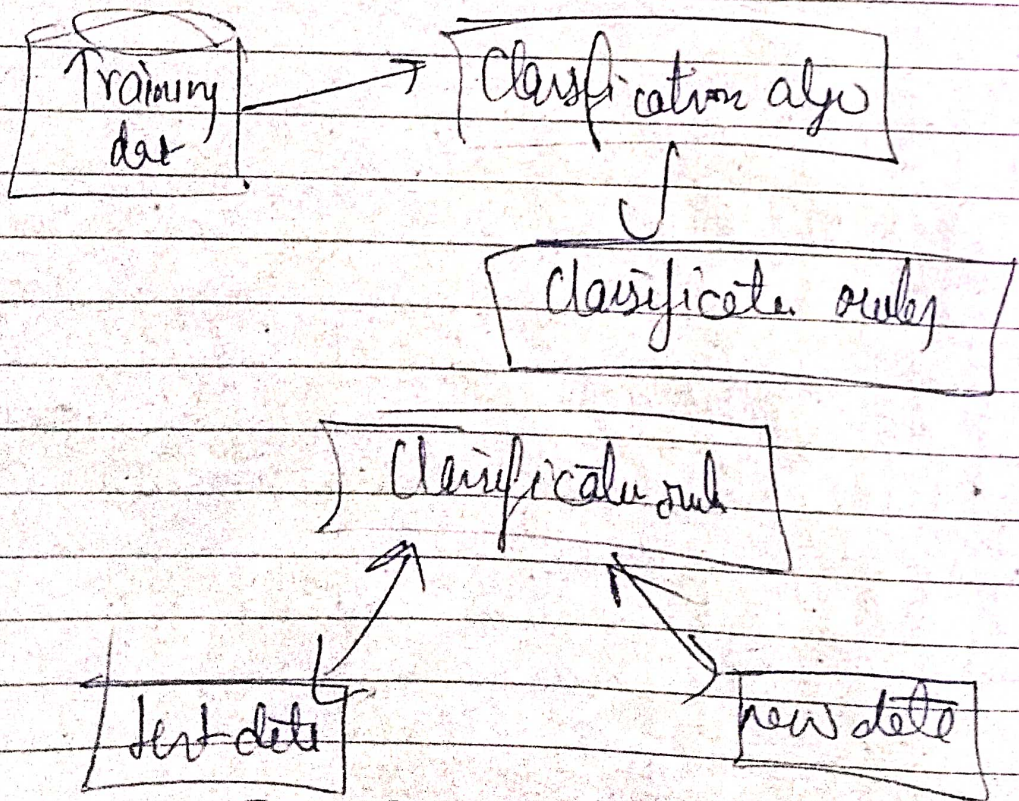
Classification

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

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

a) Learning → Training data are analyzed by a classificatn 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

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

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

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

$\frac{|D_j|}{|D|}$ 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

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

Entropy (Rainy) $\{2,0\} = 0$

$I_n(\text{Weather}) = E(\text{entire}) - \frac{2}{5}\times 0 - \frac{1}{5}\times 0 - \frac{2}{5}\times 0$

$I_n(\text{Weather}) = 0.969$

(iii) Temp

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

$= 3.321\,(0.159 + 0.117)$

0.9165

$E(\text{Mild})\{1,0\} = 0$

$E(\text{Cool})\{1,0\} = 0$

$I_n(\text{temp}) = 0.969 - \frac{3}{5}\times 0.9165 = 0.419$

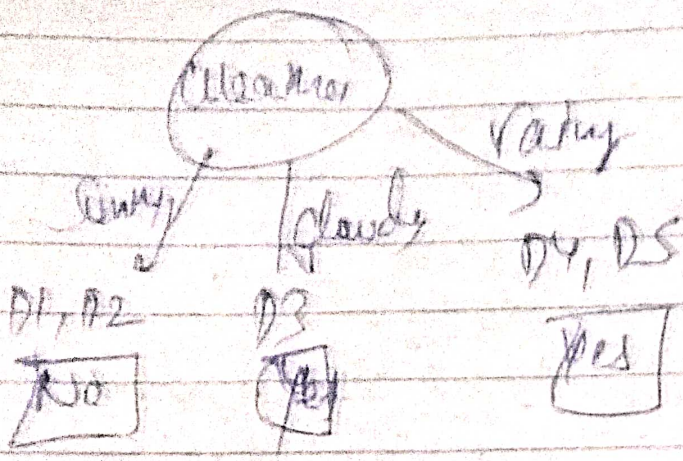
(iv) Humidity

$E(\text{High})\{1,-3\} = 0.9165$

$E(\text{Normal})\{2,0\} = 0$

$I_n = 0.419$

Maximum Weight = 0.91



Gini Index

Impurity of Data Set

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

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

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

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

Ex

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

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

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

Gini (Wearing)

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

Gini (Temp)

Gini (Hot) $\{1, -2\} = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{5}\right)^2 = 1 - 0.111 - 0.444$
$= 0.445$

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

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

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

Gini (Wearing) < Gini (Temp)

↓

less impurity

Wearing

Days	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

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

~~Interaction~~ Age

$$E(\text{Youth})\langle 2,-3\rangle = 0.97$$
$$E(\text{Middle})\langle 4,-0\rangle = 0$$
$$E(\text{Senior})\langle 3,-2\rangle = 0.97$$

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

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 (Yes/total)	
Age	Youth → NO	2/5	Total
	Middle → Yes	0/4	4/14
	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	
Medium → Yes	2/6	5/14
Low → Yes	1/4	

Student

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

Rules	Error	
Yes → Yes	1/7	4/14
NO → NO	3/7	

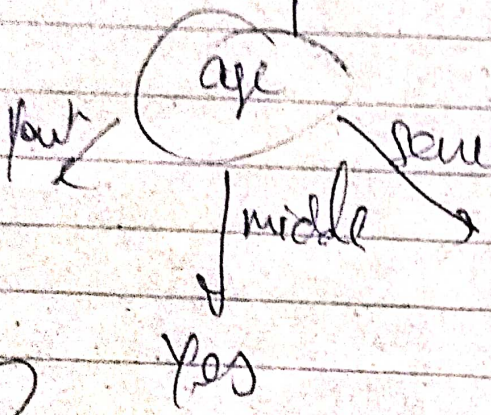
Credit

Fair	8	Yes	6
		No	2
Excellent	6	Yes	3
		No	3

Rules	Error
Fair → Yes	2/8
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	0/4	0/4
	No → no	0/3	

Same



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 tuples or the probability distribution of those tuples

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 a split.

If partitioning the tuples 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-features etc

"Pre-Pruning is used on Big Data Set"

Post Pruning → remove subtrees 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 Data Set"

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

C4.5 → pessimistic pruning → error rate

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 entered

Likelihood

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

$P(H|D)$ → Posterior

$P(H)$ → Prior

$P(D)$ → 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 ?

$$P(\text{yes}) = \frac{9}{14} = 0.64 \qquad P(\text{No}) = \frac{5}{14} = 0.36$$

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

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

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

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

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

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

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.:

Ex: Fruit = {Yellow, Sweet, Long}

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

$$P(\text{Fruit} \mid \text{Banana}) = P(Y \mid B)\, P(S \mid B)\, P(L \mid B) = 1 \times 0.85 \times 0.09 = 0.15$$

$$P(\text{Fruit} \mid \text{Others}) = P(Y \mid \text{Others})\, P(S \mid \text{Others})\, P(L \mid \text{Others}) = 0.032$$

$\Rightarrow$ greatest $\rightarrow$ Fruit = Banana

Rule Based Classification

Using IF - then Rules

IF → Rule antecedent or Precondition

then → Rule Consequent

Rule Conflict

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

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

(ii) The 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

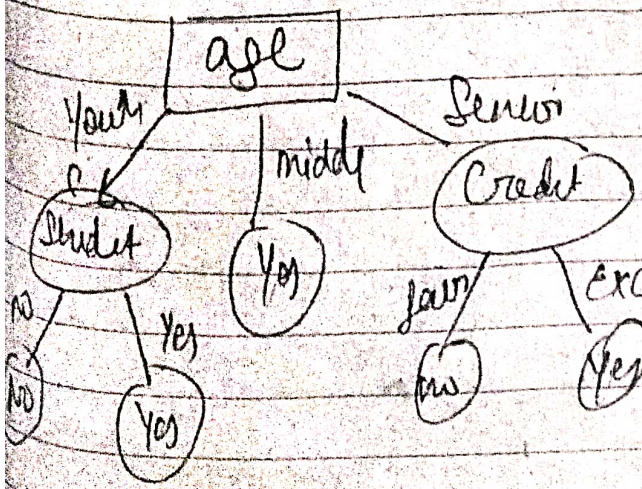
Rule_Set = {}
for each class do

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 Computer = no
R5: If age = Senior and credit = Excellent then buy Comp = Yes

Date:
Page No.:

Characteristics of Rules

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

(2) Exhaustive → There is one rule for each, and ~~every~~ tuple ~~can~~ ~~exist~~. 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

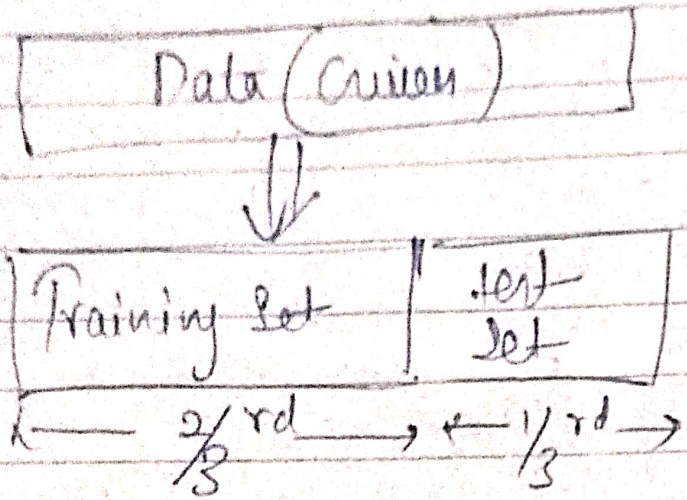
		Predicted		
		Yes	No	Row Total
Actual	Yes	TP	FN	P
	No	FP	TN	N
	Total	P'	N'	P + N

Date :
Page No.:

Measure	Formula
accuracy, recognition rate	$\frac{TP + TN}{P+N}$
error rate, misclassification 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 Score harmonic mean of precision & recall	$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{recall}}$
F_β, $\beta \rightarrow$ non negative	$\frac{(1+\beta^2) \times \text{Precision} \times \text{recall}}{\beta^2 \times \text{precision} \times \text{recall}}$

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 Sub Sampling

It is a Variation of holdout Method in which holdout method is repeated 'k' time.

The Overall accuracy → average of each accuracies

CROSS VALIDATION

K-FOLD Validation

Date:
Page No.:

Data ⇒ [D1 | D2 | D3 | D4 | D5 | ... | Dk]

i=1 [D1] [D2 | D3 | ... | Dk]

test data set — training data set

i=2 [D1 | D2 | D3 | ... | Dk]

test data set → training data set

and so on

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

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.

Stratified 10 fold Cross Validation → accuracy bcoz low bias and variance

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

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

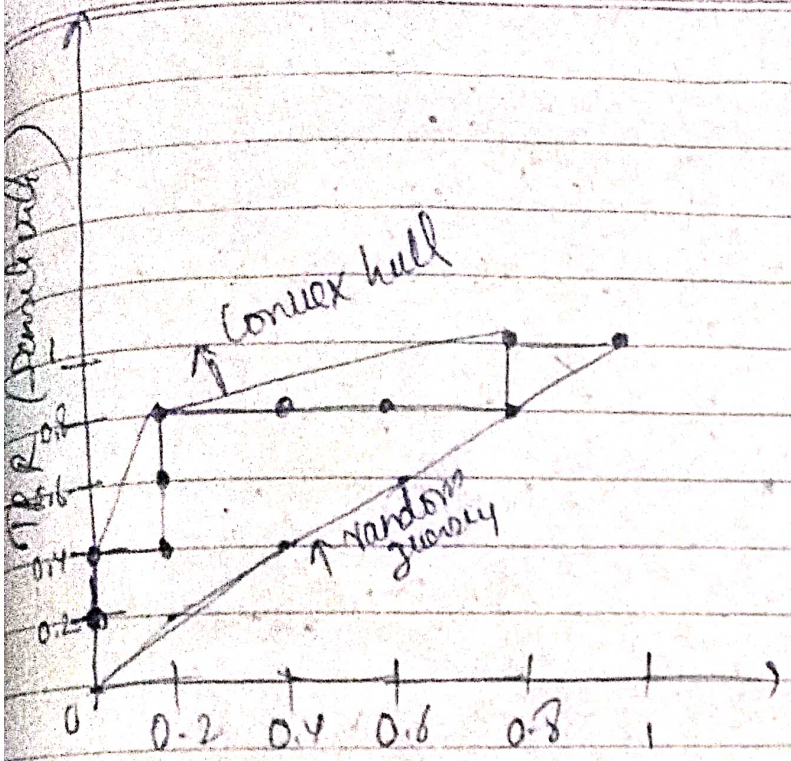
$FPR = \frac{FP}{FP+TN}$

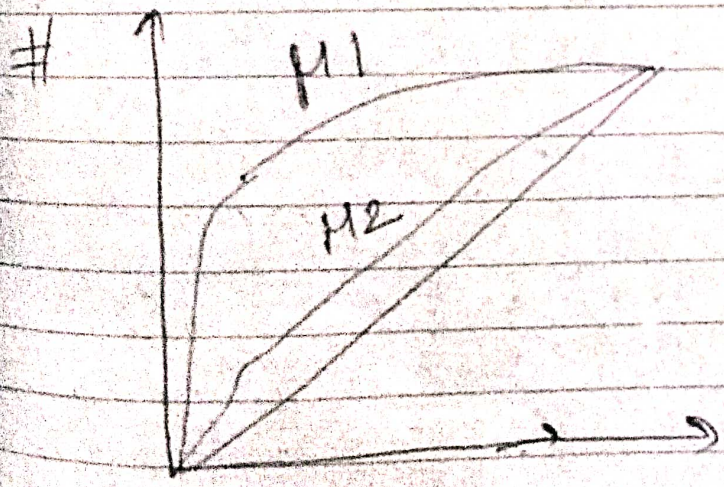
TP / P → True Positive P → P → Total Positive

FP / N → False Positive N → P → Total Negative

Ex	Tuple #	Class	Prob	TP	FP	TPR = TP/5 (total P)	FPR = FP/5 (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	0.7	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

Date :
Page No. :



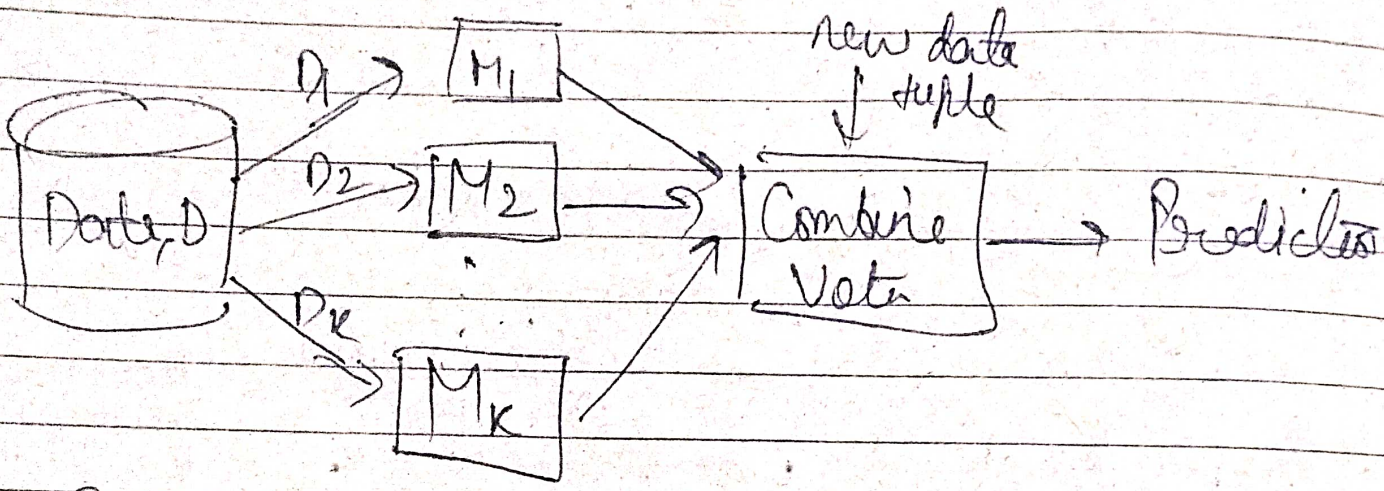


The ROC Curve is to the diagonal line, the less accurate the model is, Thus M1 is more accurate

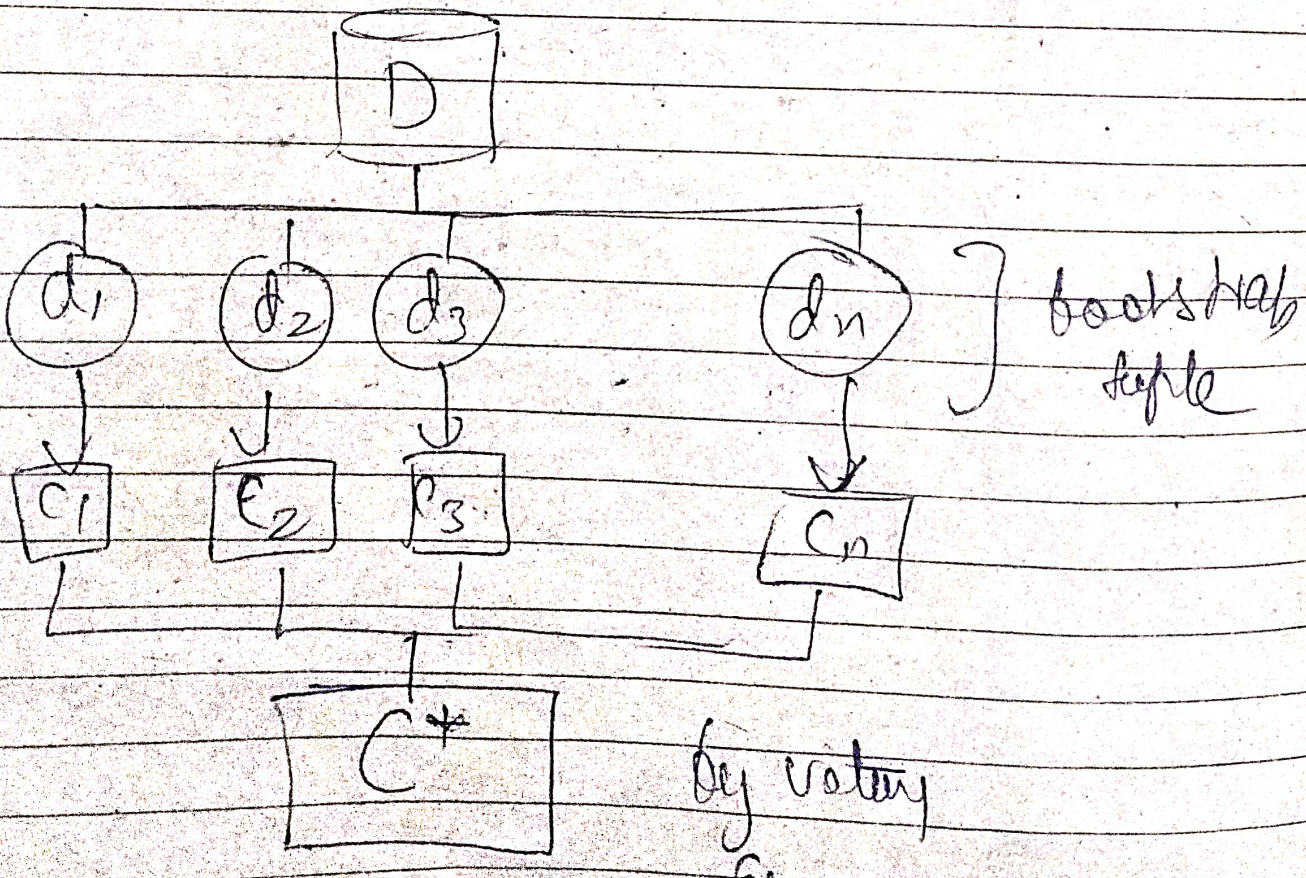
Date :
Page No. :

Ensemble Learning ~~in Doctrine Learning~~

It is a supervised learning technique used in machine learning to improve overall performance by combining the prediction from multiple models



Bagging → Bootstrap aggregation



Date :
Page No. :

Given set D, of d tuples, bagging works as follows. For iteration i (i=1, 2, ... K), a training set, D_i, of d tuples is sampled with replacement from original set of tuple D.

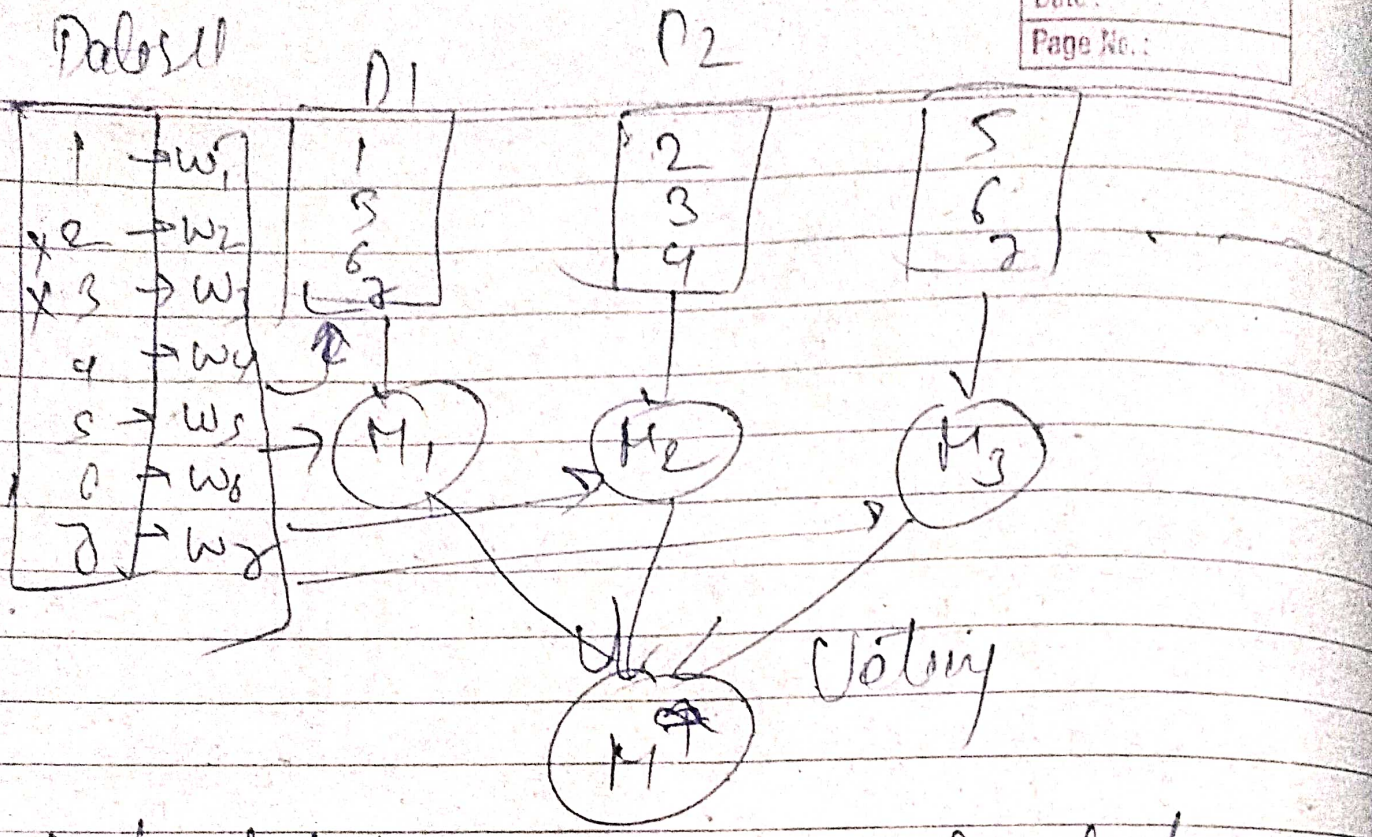
The bagged classifier often has significantly greater accuracy than a single classifier derived from D, the original training data.

It will not be considerably worse and is more robust to the effects of noisy data and overfitting.

The increased accuracy occurs because the composite model reduces the variance of the individual classifier

Boosting

Weights are also assigned to each training tuple. A series of K classifiers is iteratively learned. After a classifier, M_i, is learned, the weight are updated to allow the subsequent classifier M_{i+1}, to "pay more attention" to training tuple that are misclassified by M_i. The final boosted classifier M^*, combine the vote of each individual classifiers, where the weight of each classifier's vote is a function of its accuracy



misclassified tuple should give more priority to go to next subsequent dataset D_2

Q

True label	Predicted Probability	TP	FP	TPR	FPR
N → 1	0.8	0	1	0	1/2
P → 0	0.6	0	1	0	1/2
N 1	0.8	0	2	0	1
N 0	0.4	0	2	0	1
P 1	0.9	1	2	1/3	1

For 0.9 threshold

True	Predicted	threshold (0.9)	(0.7)	(0.3)
1	0.8	0	1	1
0	0.6	0	0	1
1	0.7	0	1	1
0	0.4	0	0	1
1	0.9	1	1	1

(i) Threshold 0.9

True	Predicted	TP	FP	TPR	FPR
1	0	0	0	0	0
0	0	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0
1	1	1	0	1/3	0

$$TPR = \frac{TP}{TP+FN} = \frac{1}{1+2} = \frac{1}{3}$$

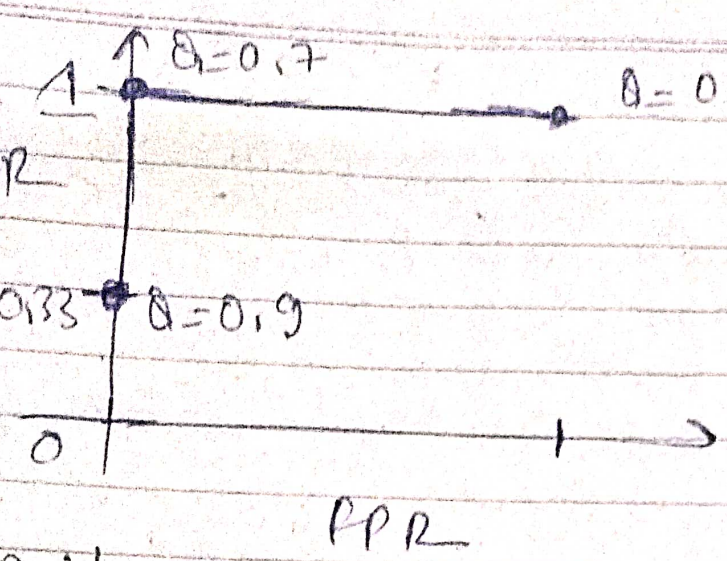
$$FPR = \frac{FP}{FP+TN} = \frac{0}{2} = 0$$

(ii) Threshold (0.7) $TPR = \frac{TP}{TP+FN} = \frac{3}{3} = 1$

$$FPR = \frac{FP}{P+TN} = 0$$

(iii) Threshold (0.3) $TPR = \frac{TP}{P} = \frac{3}{3} = 1$

$$FPR = \frac{FP}{N} = \frac{2}{2} = 1$$



Adaboost

Adaboost assigns each training tuple an equal weight of $\frac{1}{d}$. Generating K classifiers for the ensemble requires K rounds through the rest of the algo. In 'round i', the tuples from D are sampled to form a training set D_i of size d.

$$\text{error}(M_i) = \sum_{j=1}^{d} w_j$$

$$\text{Performance} = \frac{1}{2} \log_e \left(\frac{1 - \text{error}(M_i)}{\text{error}(M_i)} \right)$$

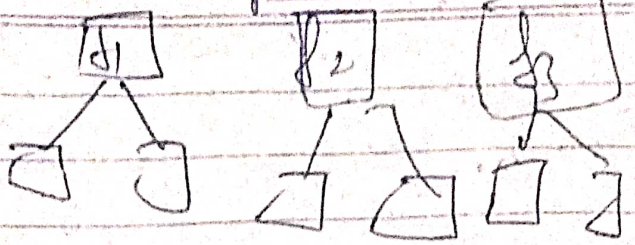
① Misclassified → weight ↑

$$\text{New weight} = \text{old weight} \times e^{+\text{Performance}}$$

② Classified → weight ↓

$$\text{new weight} = \text{old weight} \times e^{-\text{Perfor}}$$

Stump



f_1 f_2 f_3 o/p

Random forest

Random forest is a commonly used machine learning algo

A Random forest is an ensemble learning method where multiple decision tree are constructed and then they are merged to get a more accurate prediction

Random forest became popular because of its ease of use and flexibility in handling both classification and regression problem

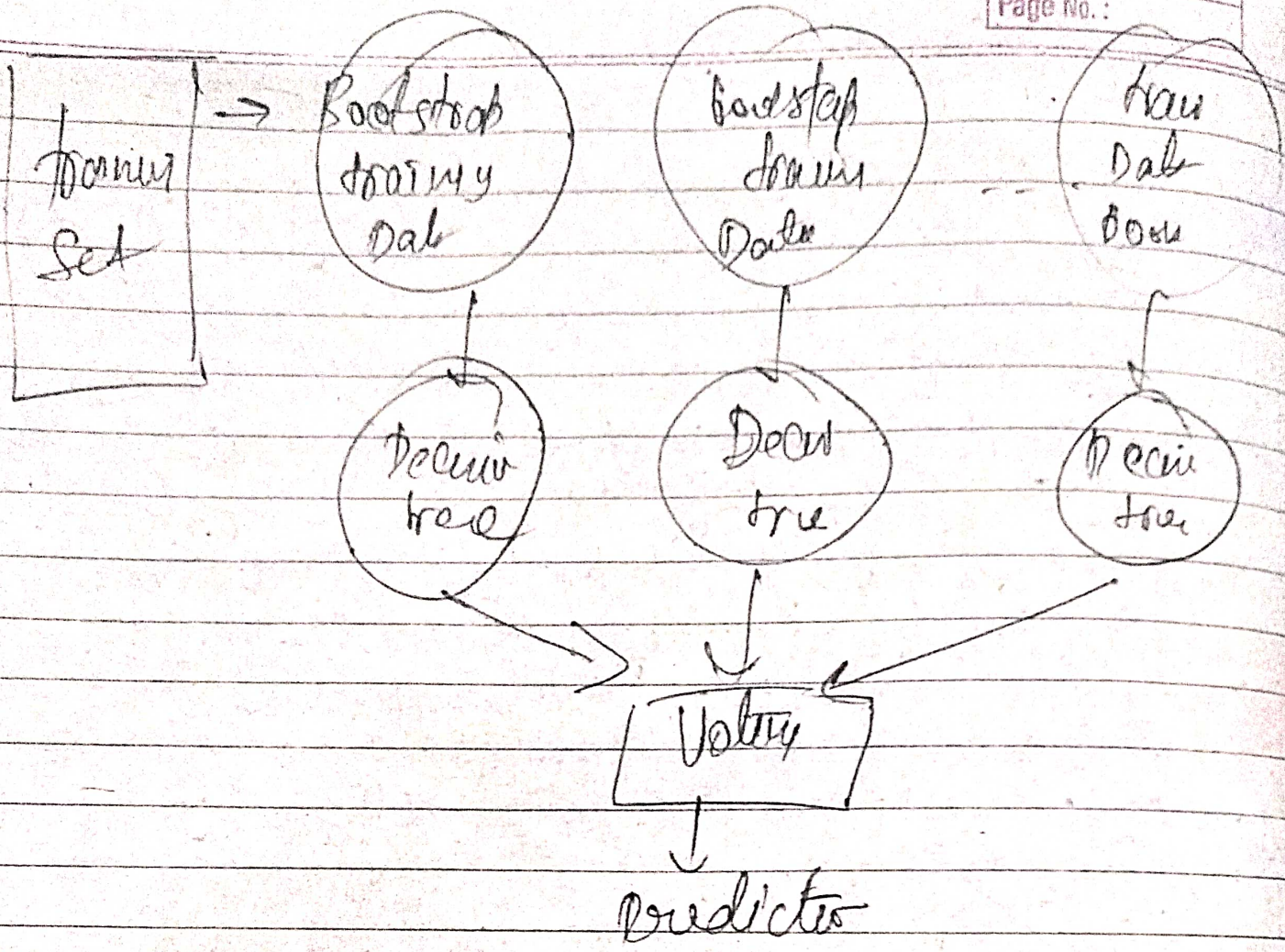
Steps

Build Random forest

a) If the number of examples in the training set is N take a sample of n example at Random - but with replacement from the original data

b) If there are 'M' input variables, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the generation of the various tree in forest

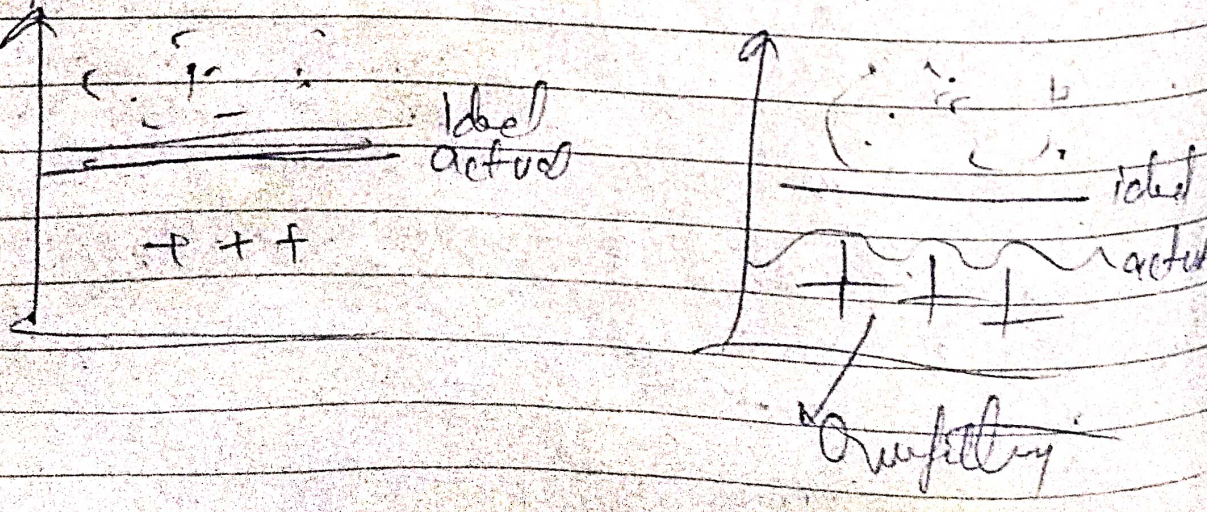
For new test data points, find the prediction of each decision tree, and assign the new data points to the category that wins the majority vote



Random Forest are comparable in accuracy to adaboost. Yet are more robust to errors and outliers

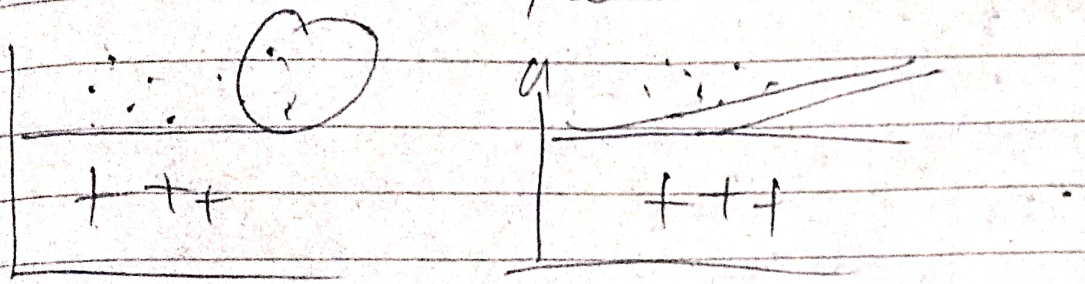
Imbalance data

(1) Oversampling → resampling the positive tuple so that the resulting training data contains an equal number of positive and negative tuple

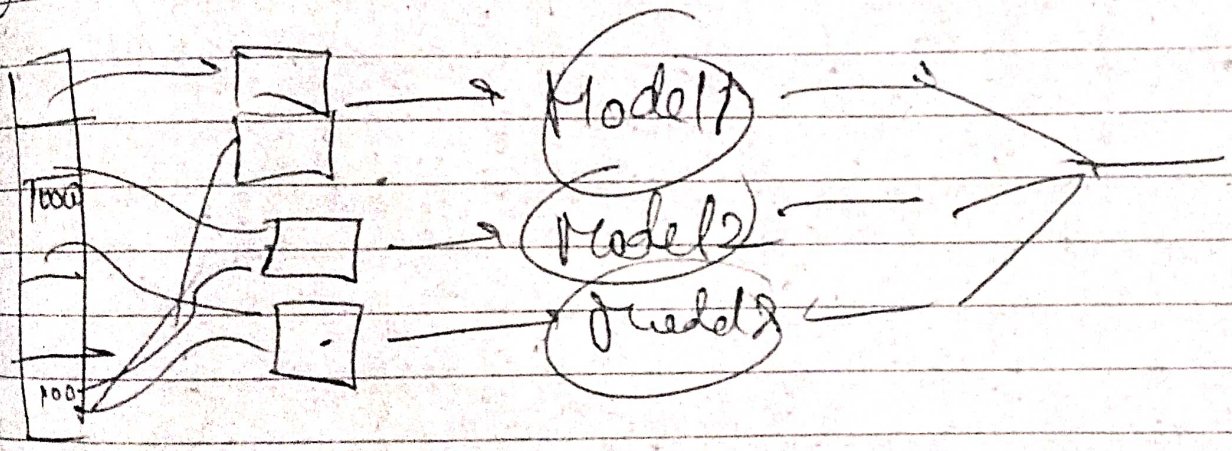


Date:
Page No.:

② Under Sampling → decreasing the number of negative tuple. It randomly eliminate tuple from the majority class until there are equal in numb



③ Ensemble



④ threshold moving

Chapter 10 data Mining

Clustering → Unsupervised learning because the class label information is not present.

Date :
Page No. :

The following are typical requirements of clustering →

1. Scalability
2. Ability to deal with different types of attributes
3. Discovery of clusters with arbitary shape
4. Requirements for domain knowledge to determine input parameter
5. deals with noisy data
6. Incremental clustering and insensitivity to input order
7. Capability of clustering high dimensionality data
8. Constraint based clustering
9. Interpretability and usability

The following are orthogonal aspects with which clustering methods can be compared

1. Partitioning criteria
2. Separation of cluster
3. Similarity measure
4. Clustering space

Basic Clustering Methods

1. Partitioning Method → Partitioning method Construct 'k' partitions of the data, where each partition represent cluster $k \leq n$.

→ K means clust
K medoids clust

Date :
Page No. :

Methods	General Characteristics
Partitioning method	→ Find mutually exclusive cluster of spherical shape → Distance based → May use mean or medoid to represent cluster center → Effective for small to medium size data sets
Hierarchical Methods	→ Clustering is a hierarchical decomposition → Cannot correct erroneous merges or splits → May incorporate other tech like micro clustering or consider object "linkage"
Density based Methods	→ Can find arbitrarily shaped clusters → Cluster are dense region of object in space that are separated by low-density region → Cluster density :- Each point have a minimum number of points within its "neighborhood" → May filter out outliers
Grid based	→ Use a multiresolution grid data str → Fast processing time

K means Clustering

Point	Centroid (2,3)	Centroid (6,5)	Cluster
(2,3)	0		1
(3,3)	1	3.6	1
(5,4)	3.16	1.4	2
(6,5)	4.4	0	2
(9,7)	$\sqrt{65} = 8.0$	$\sqrt{11} = 3.3$	2
(10,8)	$\sqrt{41} = 6.4$	5	2

new clusters

1 $\left(\frac{5}{2}, \frac{6}{2}\right)$ $\therefore (2.5, 3)$

2 → $\left(\frac{30}{4}, \frac{24}{4}\right)$ $(7.5, 6)$

Point	Cent 1 (2.5,3)	Cent 2 (7.5,6)	Old	new
(2,3)	0.5		1	1
(3,3)	0.5		1	1
(5,4)	7.25	10.25	2	1
(6,5)	12.25	3.25	2	2
(9,7)			2	2
(10,8)			2	2

new Cent

1 (3.33, 3.33) 2 (8.33, 6.66)

Date:
Page No.:

K Medoid

Partitioning around Medoids or the K-medoids algorithm is a partitional clustering algorithm which is slightly modified from the K means algo.

In K-means algo, they choose means as the centroids but in the K-medoids, data points are chosen to be the medoids

Ex

X	Y	$C_1=(3,4)$	$C_2=(7,4)$	cluster
2	6	3	7	1
3	4	0	4	1
3	8	4	8	1
4	7	4	6	1
6	2	5	3	2
6	4	3	1	2
7	3	5	1	2
7	4	4	0	2
8	5	6	2	2
7	6	6	2	2

Find distance Manhattan Dist = $|x_1-x_2|+|y_1-y_2|$

C1: (2,6), (3,4), (3,8), (4,7)

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

C2: (6,2) (6,4) (7,3) (7,4) (8,5) (7,6)

(6,2) (6,4) (7,3) (7,4) (7,1) (8,5)

Date :
Page No. :

$Cost(C, x) = \sum_i |C_i - x_i|$

Total Cost = { Cost((3,4),(2,6)) + Cost((7,4),(3,8)) +
Cost((9,4),(4,2)) + Cost((7,4),(6,2)) +
Cost((7,4),(6,4)) + Cost((7,4),(2,2)) +
Cost((7,4),(8,5)) + Cost((7,4),(2,6)) }

= 1+2+0+4+1+3+1+2+1+0+
0+1+1+1+0+2 = 20

Step 3 Again choose random new point other than 2 medoid

O: (7,3)

X	Y	C1 = (3,4)	O = (7,3)	cluster
2	6	3	9	C1
3	4	0	5	C1
3	8	4	9	C1
4	2	4	2	C1
6	2	5	2	O
6	4	3	2	O
7	3	5	0	O
7	4	4	1	O
8	5	6	3	O
7	6	6	3	O

C1: { (2,6), (3,4), (3,8), (4,2) }

O: { (6,2) (6,4) (7,3) (7,4) (8,5) (7,6) }

Date :
Page No. :

Cost = { Cost((3,4),(2,6)) + Cost((3,4),(3,7)) + Cost
((3,4)(4,7) + Cost((2,3)(6,2)) + Cost((2,3),(6,4))
+ Cost((2,3)(7,4)) + Cost((9,3)(8,5)) + Cost((7,2),(9,1))

= 1 + 2 + 0 + 4 + 1 + 3 + 1 + 1 + 1 + 2 + 0 + 1 + 1 + 2 +
0 + 3 = 22

Cost of Swapping

Current cost > Previous Cost
22 > 20

So no swapping is needed C2 with O

So cluster find as C1: { (2,6) (3,4) (7,3), (4,7)
C2: {(6,2)(6,4) (2,3) (2,4)
(8,5), (7,6)

" The K medoids is more robust than K means as the presence of noise and outlier because a medoid is less influenced by outlier or other extreme values than a mean "

but K medoids take $O(K(n-K)^2)$

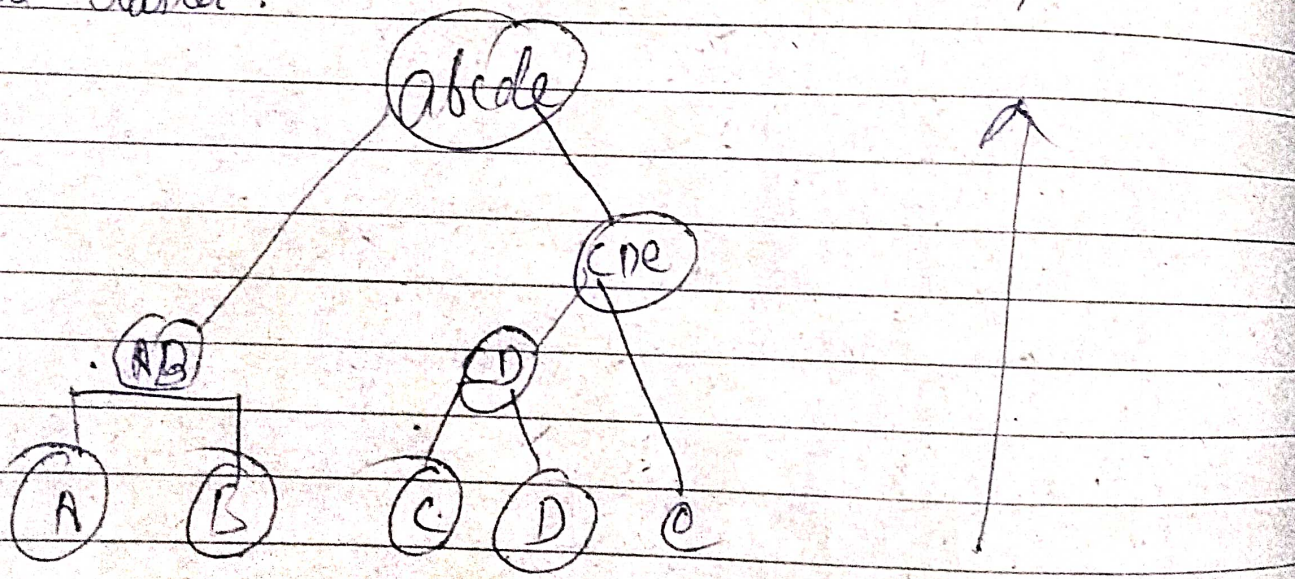
Date:
Page No.:

Hierarchical Methods

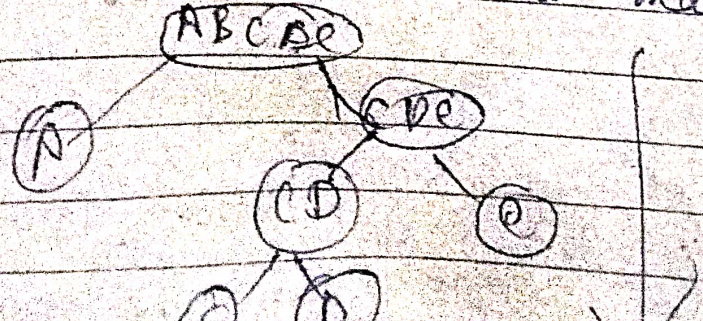
Works by grouping data objects into a hierarchy or "tree" of clusters.

Agglomerative Versus Divise

An agglomerative hierarchical method used bottom up strategy. It typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger cluster, until all the object are in a single cluster.

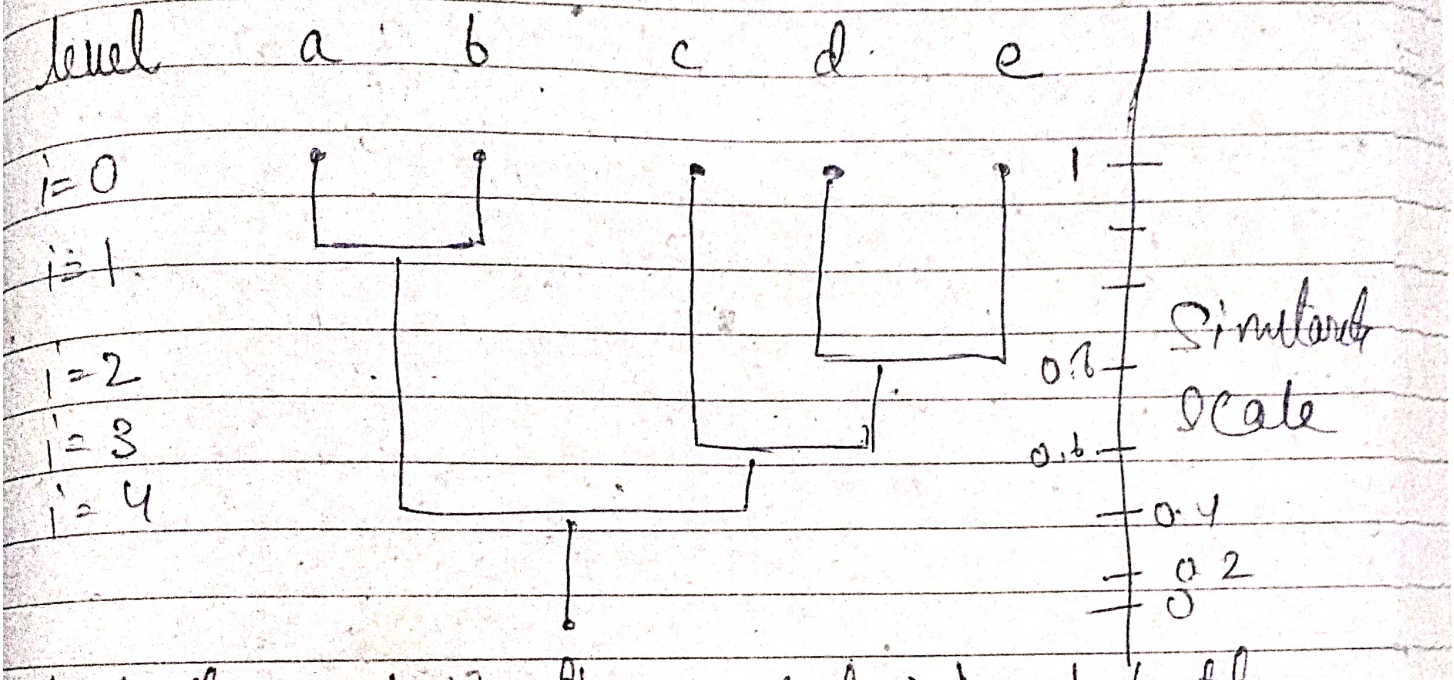


An divise method used a top down approach. It starts by placing all objects in one cluster, which is the hierarchy root. It then divides the root cluster into several smaller subcluster, and recursively partition those cluster into smaller ones.



Date :
Page No. :

Dendogram representation.



Challenges with divisive method → how to partition a large cluster into several smaller ones

Agglomerative

Single linkage → minimum

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2	8	0

↓ minimum

	P_1	P_2	P_3	$[P_4 P_5]$
P_1	0			
P_2	9	0		
P_3	3	7	0	
$[P_4 P_5]$	6	5	8	0

P_4 P_5

Cluster

Distance $(P_1, [P_3, P_4])$
$= \min(dist(P_1, P_3), dist(P_1, P_4))$
$= \min(3, 6) = 6$

Date :
Page No. :

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2	8	0

minimum

	P_1	P_2	$[P_3\ P_5]$	P_4
P_1	0			
P_2	9	0		
$[P_3\ P_5]$	3	7	0	
P_4	6	5	8	0

$$d(P_1, [P_3, P_5]) = \min(d(P_1, P_3), d(P_1, P_5)) = \min(3, 11) = 3$$

Single linkage

$$d(P_2, [P_3, P_5]) = \min(d(P_2, P_3), d(P_2, P_5)) = \min(7, 10) = 7$$

$$d(P_4, [P_3, P_5]) = \min(d(P_4, P_3), d(P_4, P_5)) = 8, 8 = 8$$

Now,

	P_1	P_2	$[P_3\ P_5]$	P_4
P_1	0			
P_2	9	0		
$[P_3\ P_5]$	3	7	0	
P_4	6	5	8	0

min

$[P_1, P_3 P_5]$ now clubb

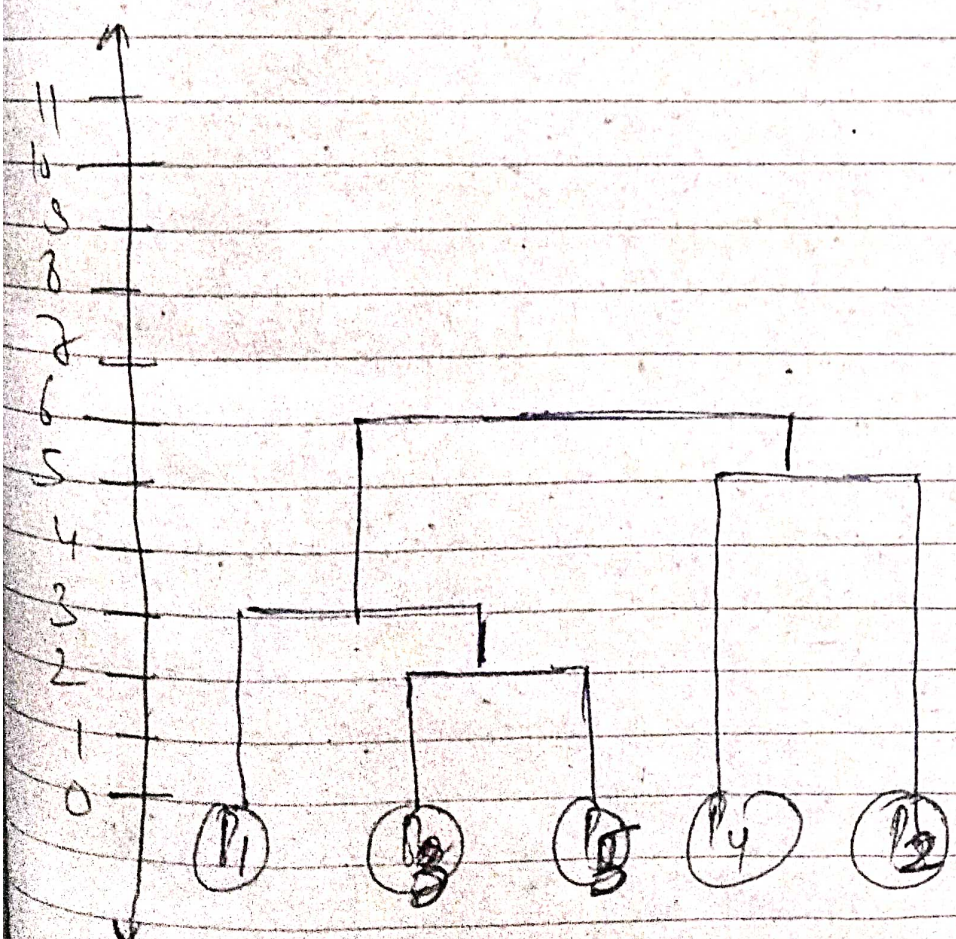
Date :
Page No. :

	P_1, P_3, P_5	P_2	P_4
P_1, P_3, P_5	0		
P_2	7	0	
P_4	6	5	0

min

$[P_2, P_4]$ new

	P_1, P_3, P_5	P_2, P_4
P_1, P_3, P_5	0	
P_2, P_4	6	0



Dendogram

Date :
Page No. :

Agglomerative where we minimum distance → minimum
- Prawn the
algo

Soft Clustering

On the other hand in Soft clustering each data point belongs to a cluster with a certain probability also known as membership value

FCM (Fuzzy C-mean clustering) algorithm is an Example of Soft Cluster