Unit 3

* Classification of Data

this chapter presents alternative techniques for building classification models from simple techniques such as rule based and nearest-neighbor classifier to more advanced techniques such as support vector machines and etc.

A) Rule-Based Classifier

a rule based classifier is a technique for classifying records using a collection of "if ... then ..." rules. the rules for the model are represented in a disjunctive normal form.

$$R = (r_1 \vee r_2 \vee \dots \vee r_k)$$

where, 'R' is known as the rule set and r_i's are the classification rules or disjuncts.

r_1: (Gives Birth = no) $\wedge$ (Aerial Creature = yes) $\rightarrow$ Birds
r_2: (Gives Birth = no) $\wedge$ (Aquatic Creature = yes) $\rightarrow$ Fishes
r_3: (Gives Birth = yes) $\wedge$ (Body Temp. = warm blooded) $\Rightarrow$ Mammals
r_4: (Gives Birth = no) $\wedge$ (Aerial Creature = no) $\rightarrow$ Reptiles.
r_5: ~~(Gives Birth =~~ (Aquatic Creature = semi) $\rightarrow$ Amphibians

Each classification rule can be expressed in the following way.

$$r_i : (\text{Condition}_i) \rightarrow y_i$$

the left-hand side of the rule is called the rule antecedent it contains a conjunction of attribute tests.

Page :
Date : / /

Condition$_i$ = $(A_1$ op $V_1) \wedge (A_2$ op $V_2) \wedge \cdots (A_k$ op $V_k)$

where (A_j, V_j) is an attribute - value pair and op is a logical operator chosen from the set $\{=, \neq, <, >, \leq, \geq\}$. Each attribute test $(A_j$ op $V_j)$ is known as a conjunct. The right-hand side of the rule is called the rule consequent, which contains the predicted class y_i.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has legs	Hibernates
hawk	warm blooded	feather	no	no	yes	yes	no
grizzly bear	warm blooded	fur	yes	no	no	yes	yes

r_1 covers the first vertebrate because its precondition is satisfied by the hawk's attributes. the rule does not cover the second vertebrate because grizzly bears give birth to their young and cannot fly, thus violating the precondition of r_1.

the quality of a classification rule can be evaluated using measures such as coverage and accuracy. Given a Data set D and a classification rule $r: A \rightarrow y$, the coverage of the rule is defined as the fraction of records in D that trigger the rule r. On the other hand, its accuracy is defined as the fraction of records triggered by r.

Coverage $(r) = \frac{|A|}{|D|}$

Accuracy $(r) = \frac{|A \cap y)}{|A|}$

where, $|A|$ is the number of records that satisfy the rule antecedent, $|A \cap y|$ is the number of records that satisfy the both the antecedent and consequent, and $|D|$ is the total number of records.

A1i) How a Rule-Based Classifier Works.

A rule-based classifier classifies a test record based on the rule triggered by the record.

Name	Body temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has legs	Hibernates
lemur	warm blooded	fur	yes	no	no	yes	yes
turtle	cold blooded	scales	no	semi	no	yes	no
dogfish shark	cold blooded	scales	yes	yes	no	no	no

- The first vertebrate, which is a lemur, is warm blooded and gives birth to its young. It triggers the rule r_3, and thus, is classified as a mammal.

- The second vertebrate, which is a turtle, triggers the rules r_4 and r_5 since the classes predicted by the rules are contradictory (reptiles versus amphibians), their conflicting classes must be resolved.

- None of the rules are applicable to a dogfish shark. In this case, we need to ensure that the classifier can still make a reliable prediction even though a test record is not covered by any rule.

* Mutually Exclusive Rules

The rules in a rule set R are mutually exclusive if no two rules in R are triggered by the same record.

This property ensures that every record is covered by at most one rule in R.

* Exhaustive Rules

A rule set R has exhaustive coverage if there is a rule for each combination of attribute values. This property ensures that every record is covered by at least one rule in R.

r_1 : (Body temperature = cold blooded) → Non-Mammals.

r_2 : (Body temperature = warm blooded) ∧ (Gives Birth = yes) → Mammals

r_3 : (Body temperature = warm blooded) ∧ (Gives Birth = no) → non-mammals.

If the rule set is not mutually exclusive then a record can be covered by several rules

* Ordered Rules

the rules in which a rule set are ordered in decreasing order of their priority, Ex (based on accuracy, coverage, total description length, or the order in which the rules are generated) An ordered rule set is also known as decision list

* Unordered Rules

this approach allows a test record to trigger multiple classification rules and considers the consequent of each rule as a vote for a particular class.

Unordered rules are less susceptible to errors caused by the wrong rule being selected to classify a test record.

Model building is also less expensive because the rules do not have to be kept in sorted order. Nevertheless, classifying a test record can be quite an expensive task because the attributes of the test record must be compared against the precondition of every rule in the rule set.

Explain rule ordering scheme.

* Comparison bet^n rule-Based & class-based Ordering Schemes.

Rule-Based Ordering	Class-Based Ordering
(Skin-Cover = feathers, Aerial Creature = yes) => Birds	(Skin cover = feathers, Aerial Creature = yes) => Birds
(Body temperature = warm blooded, Gives Birth = yes) => Mammals	(Body temperature = warm blooded, Gives Birth = no) => Birds
(Body temperature = warm blooded, Gives Birth = no) => Birds	(Body temperature = warm blooded, Gives Birth = yes) => Mammals
(Aquatic Creature = semi) => Amphibians	(Aquatic Creature = semi) => Amphibians
(Skin Cover = scales, Aquatic Creature = no => Reptiles	(Skin Cover = none) => Amphibians
(Skin Cover = scales, Aquatic Creature = yes) => Fishes	(Skin cover = scales, Aquatic Creature = no) => Reptiles
(Skin Cover = none) => Amphibians	(Skin cover = Scales, Aquatic Creature = yes) => Fishes

22-11-22 DM

Page No.
Date

* How to build a Rule based classifier

1) To build a rule based classifier we need to extract a set of rules that identifies a key relationship between the attributes of data and the class level. There are 2 broad classes of methods for extracting classification rules

i) direct method :- which extract classification rules directly from the data.

2) indirect method :- which extract classification rules from other classification models such as decision trees and neural networks.

1) Direct methods for Rule Extraction :-

* Characteristics of rule Base classifier

1) The expressiveness of a rule set is almost equivalent to that of a decision tree because a decision tree can be represented by a set of mutually exclusive and exhaustive rules. Both rule-based and decision tree classifiers create rectilinear partitions of the attribute space & assign a class to each partition.

2) Rule Base classifier are generally used to produce descriptive models that are easier to interpret but gives comparable performance to the decision tree classifier.

3) ~~The class~~ If the rule based classifier allows multiple rules to be triggered for a given record then a more complex decision boundary can be constructed.

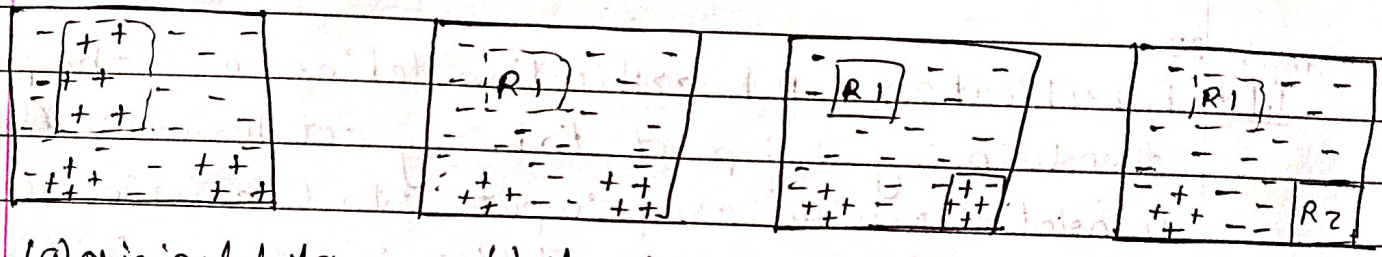
4) The class based ordering approach adapted by many rule-based classifiers is well suitated for handling data sets with imbalanced class distributions.

DM

How to build Rule base classifier

(I) Direct Method

The sequential covering algorithm is often used to extract rule directly from data rules are grown in a greedy fashion based on a certain evaluation measure The algorithm extracts the rules of one class at a time for data sets that contain more than two classes. For the vertibrates classification birds problem The sequential covering algorithm may generate rules for classifying birds first followed by the rules for classifying mammals amphibians, reptiles and finally fishes.



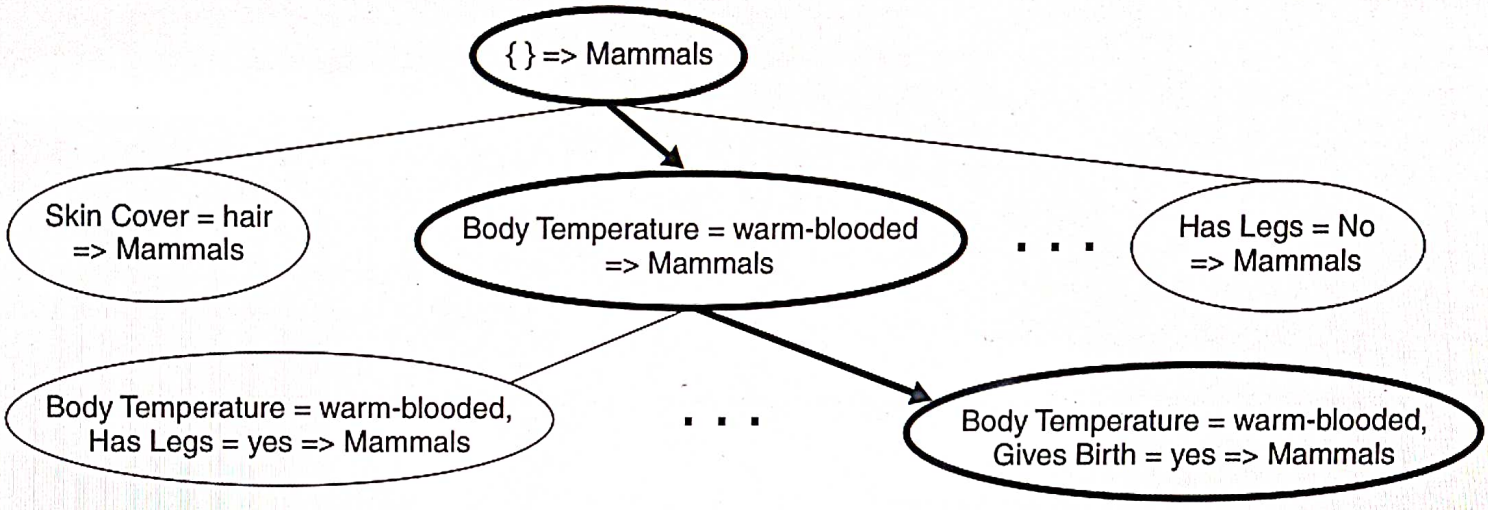
(a) original data b) step 1 c) step 2 d) step 3

Fig:- An example of the sequential covering algorithm

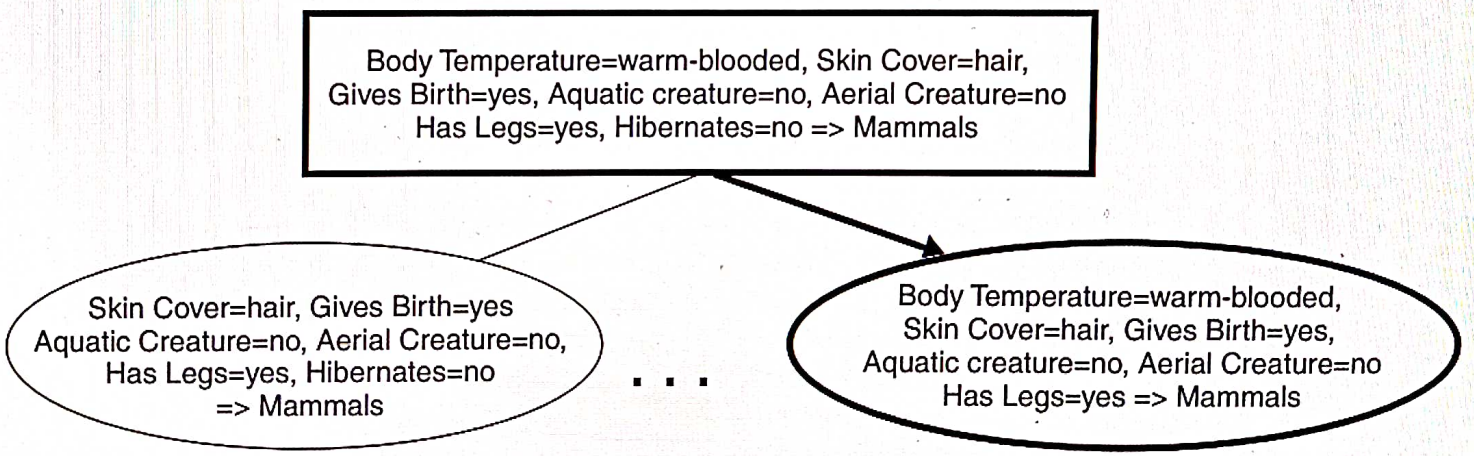
•) Learn-One-Rule Function :- The objective of the learn-One-Rule function is to extract a classification rules that covers many of the positive examples and non of the negative examples in the training set. The learn-one-Rule function addresses the exponential search problems by growing rules in the a greedy fashion.

•) Rule Growing strategy: There are 2 common strategies for growing a classification rules (1) General to specific (2) Specific to general. Under the general-to-specific strategy, an initial rule $r: \{\} \rightarrow y$ is created, where the LHS is an empty set and and RHS contains the target class. The rule has poor quality because it covers all the info example in the training sets.

Diagram on page (214) Fig 5.3



(a) General-to-specific



(b) Specific-to-general

Figure 5.3. General-to-specific and specific-to-general rule-growing strategies.

DATE : __/__/____ PAGE No : ____

★ Rule Evaluation.

An evaluation matrix is needed to determine which conjuct should be added during the rule growing process.

★ Indirect method for rule extraction.

Every path from root node to the leaf node of a decision tree can be express as a classification rule the rule set is exosthe and contain mutually exclusive.

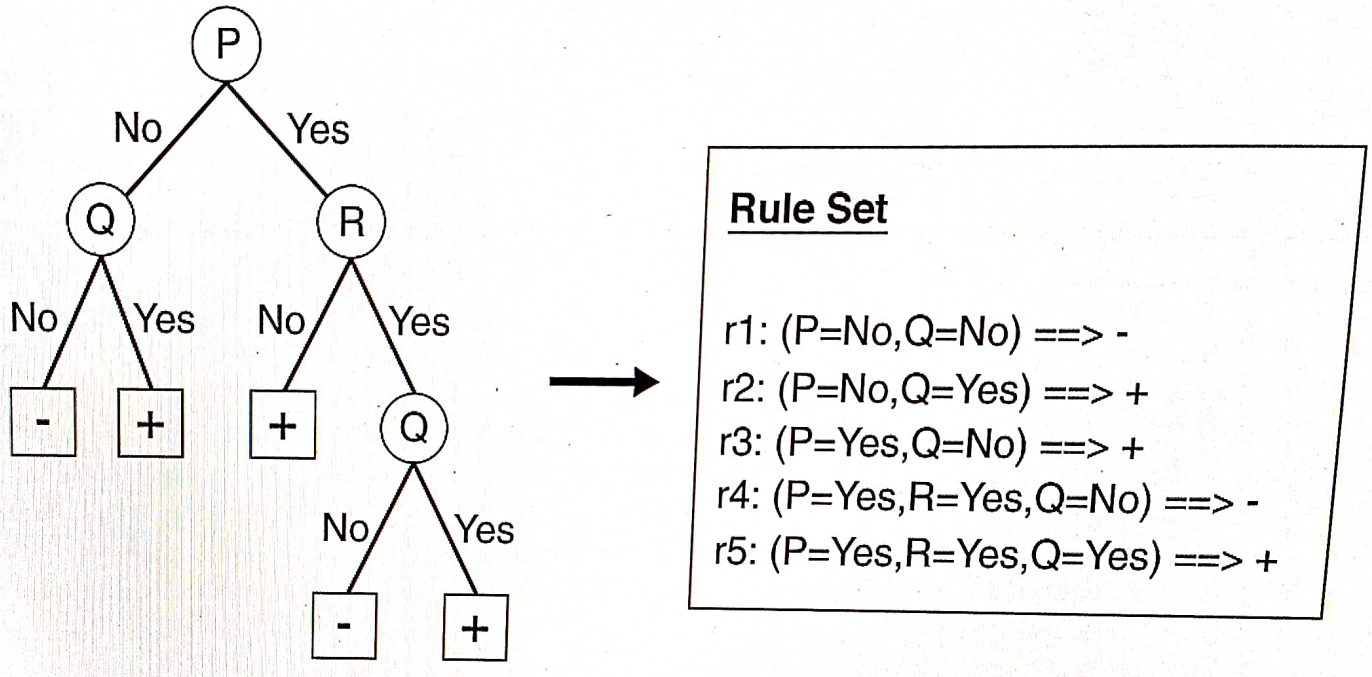


Figure 5.5. Converting a decision tree into classification rules.

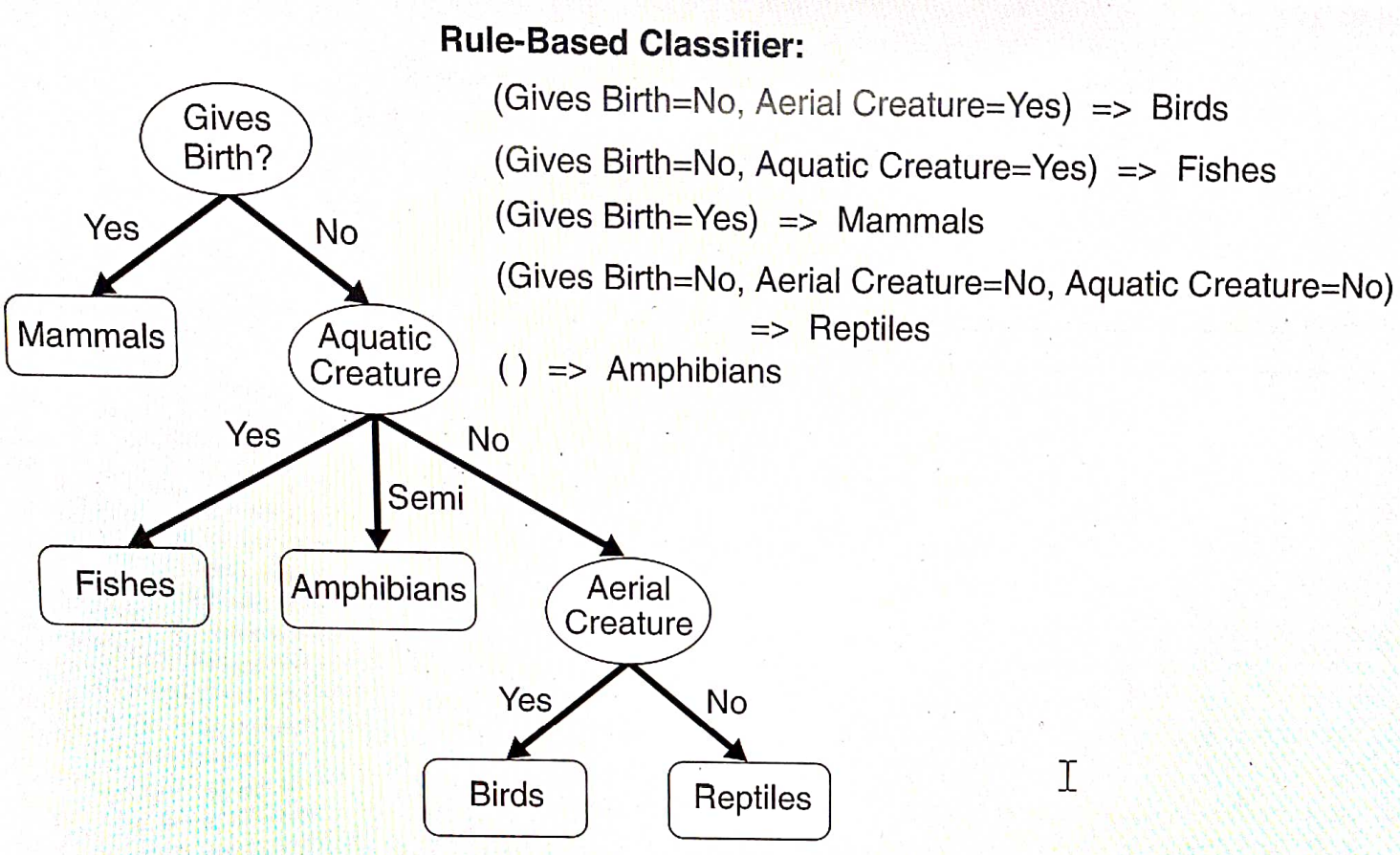


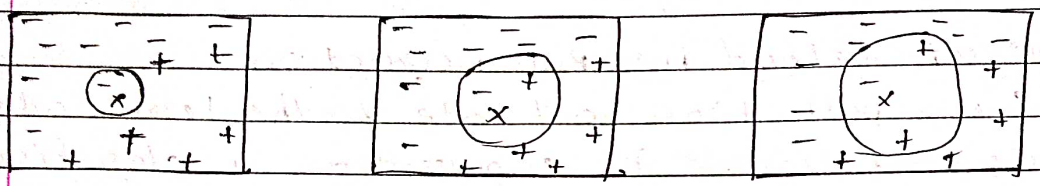
Figure 5.6. Classification rules extracted from a decision tree for the vertebrate classification problem.

•) Rule Generation :- classification rules are extracted for every path from the root to one of the leaf nodes in the decision tree. Given a classification rules $r : A \rightarrow y$,

•) Rule ordering :- After generating the rule set, class-based ordering scheme to order the extracted rules. Rules that predict the same class are grouped together into the same subset.

* Nearest - Neighbor classifiers : The classification framework involves a 2 step process (1) An inductive step for constructing classification model from data. (2) A deductive step for applying the model to test examples. Decision tree & rule-based classifiers are examples of eager learners.

because they are design to learn a model that maps the input attributes to the class label as soon as the training data will becomes available



a) 1-nearest neighbor b) 2-nearest neighbor c) 3-nearest neighbor

* Drawback of this approach is that some test record may not be classified because they do not match any training example. One way to make these approch more flexible is to find all the training examples that are relatively similar to the attributes of the test example. this examples which are known as nearest neighbors

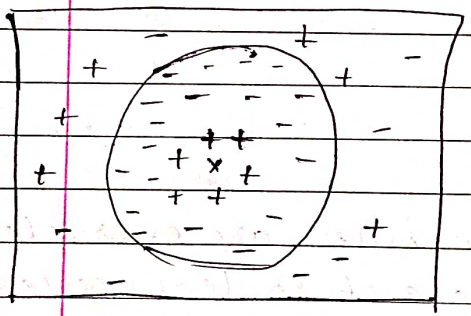


fig: K-nearest neighbor classification with large K.

If K is too small then the nearest neighbor classifier maybe suscepdible to overfitting because of noise in the training data. On the other hand if K is too large the nearest neighbor classifier may missclassify the test instance because ~~its~~ of list of nearest neighbor may include data points that are located far away from its neighborhood.

Page No.
Date

* Characteristic of Nearest -Neighbour Classifiers

1) Nearest Neighbor classification is a part of a more general technique known as instance based learning which uses specific training instance to make predictions without having to maintain an abstraction derive from data.
2) Lazy learner such as nearest ~~tea~~ neighbor classifier do not require model building
3) NNC ~~made~~ make there prediction based on the local information where as decision trees and rule base classifier attempts to find a global model.
4) The decision boundaries of the nearest neighbor classifier also have high variability because they depend on ~~decomposition~~ composition of training examples. Increasing the no. of nearest neighbor may reduce such variability.
5) NNC can produce wrong prediction unless the appropriate proximity measure and ~~data~~ pre processing steps are taken.

4] what is Bayesian classifier? Explain naive bayes classifier in detail.

Ans:- Bayesian classifier is a probabilistic approach to learning and inference based on a different view of what it means to learn from data, in which probability is used to represent uncertainty about the relationship being learnt.

* Bayes Theorem: Bayesian classifier uses bayes theorem to predict the occurence of any event. Bayesian classifier are the statistical classifier with the bayesian probability understandings. The theory expresses how a level of belief expressed as a probability.

Bayes theorem comes into existence after Thomas Bayes, who first utilized conditional probability to provide an algorithm as evidence to calculate limits as an unknown parameter.

Bayes theorem is expressed mathematically

$$P(x/y) = \frac{P(y/x) \cdot P(x)}{P(y)}$$

where x & y are events of $P(y) \neq 0$
$P(x)$ and $P(y)$ are the probabilities of occurring x and y.

eg: Consider a football game between two rival teams: Teams 0 and Team1. Suppose Team 0 wins 65% of the time and Team1 wins the remaining matches. Among the games won by Team 0, only 30% of them came from ~~Team~~ playing on Team1's football field. On the other hand, 75% of the victories for Team1 are attained while playing at home. If Team1 is to host the next match between the two teams, which team will most likely emerge as the winner.

- Let x be the random variable that represent team hosting the match
- Let y be random variable that represent winner of match

Ans: Probability Team 0 wins is $P(y=0) = 0.65$
Probability Team 1 wins is $P(y=1) = 1 - P(y=0)$
$= 0.35$

Probability Team 1 hosted the match it won is: $P(x=1 \mid y=1) = 0.75$
Probability Team 1 hosted the match won by Team 0 is $P(x=1 \mid y=0) = 0.3$

our objective is to find $P(y=1 \mid x=1)$, which is conditional probability that Team 1 wins the next match it will be hosting and compares it against $P(y=0 \mid x=1)$

Using Bayes theorem, we obtain

$$P(y=1 \mid x=1) = \frac{P(x=1 \mid y=1) \times P(y=1)}{P(x=1)}$$

$$= \frac{P(x=1 \mid y=1) * P(y=1)}{P(x=1)}$$

धैर्य कड़वा लगता है, पर उसका फल मीठा होता है।

$$= \frac{P(X=1|Y=1) \times P(Y=1)}{P(X=1, Y=1) + P(X=1, Y=0)}$$

$$= \frac{P(X=1|Y=1) \times P(Y=1)}{P(X=1|Y=1)P(Y=1) + P(X=1|Y=0)P(Y=0)}$$

$$= \frac{0.75 \times 0.35}{0.75 \times 0.35 + 0.3 \times 0.65}$$

$$= 0.5738$$

Here $P(Y=0|X=1) = 1 - P(Y=1|X=1)$

$$= 1 - 0.5738$$

$$= 0.4262$$

Since $P(Y=1|X=1) > P(Y=0|X=1)$

Team 1 has a little chance than Team 0 of winning the next match.

9
10

Naive Bayes classifier -

1) A naive Bayes classifier algorithm is a supervise learning algorithm which is based on bayers theorm and use for solving classification prob.

2) It is mainly used in tex classification that includes the high dramensional training data set. naive bayes classifier is one of the simple and most important classifier. which helps in building the fast machine learning model that can make quick prediction. it is the probabilistic classifier. which predicts on the basis of the probability of an object. some popular objects on Bayes by classifiers. and classifier antifier.

b) The Naive baied classifier Two words -

Naive - It is called Naive because it assums that because the acurance of the certain feature. is independent of the certain ~~features~~. orther feature.

Such as a fruit is identified on the basis of colour, shape, and taste they spherical and sweet fruit is reognised as an

Hence, this feature individually attributes.
So each feature individually treat it
is an apple without depend on each other
bay's it is called bayes theerm - because its
depends on the principal of

$$p(x|y=y) = \prod_{i=1}^{d} (p\, xi.|y=y)$$

where, each attributes set,

$x = \{x_1, y_2 - - - x_d\}$ consist of d attributes.

Conditional Independence - Let x, y, z denotes there sets of random variable.

The variables in x are said to be conditionally, independent of y given z.

$$p(x|y,z) = p(x|z)$$

Conditional Independence between,
x and y are also be attributes,

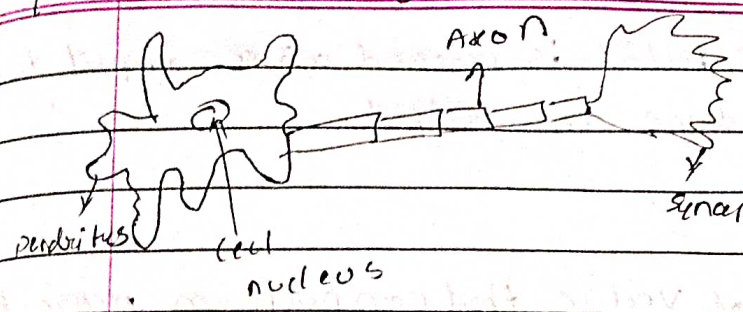
$$p(x,y/z) = \frac{p(x,y,z)}{p(z)}$$

$$= \frac{p(x,y,z)}{p(y,z)} \times \frac{p(y,z)}{p(z)}$$

$$= p(x|y,z) \times p(x|z)$$

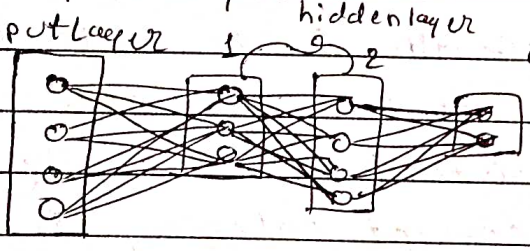
$$= p(x|z) \times p(y|z)$$

1-12-22 DM



BNN	ANN
Dendrites	Input
Cell nucleus	Nodes
Synapse	weights
Axon	Output

* ANN :- An ANN is the field of AI where it attempts to mimic the network of neurons, makes up the human brain so that the computers will have an option to understand things and make decisions in a Human like manner. ANN is designed by programming computers to behave simply like interconnected brain cells. ANN primarily consists of 3 layers 1) Input layer 2) Hidden layer and 3) Output Layers.



Architecture of ANN

1) Input layer :- It accepts input in several different formates provided by the programer.

2) Hidden layer :- The hidden layer presents between input & output layer it performs all the calculations to find hidden patterns and features

3) output layer :- The input goes through a series of transformation using the hidden layer which finally results in output that is conveyed using this layer.

The ANN takes input and computes the weighted some of inputs and includes a bias this computation is represented in the form of a transformation. $\sum_{i=1}^{n} W_i * X_i + b$

W_i :- weighted function, b = bias

X_i :- Input function

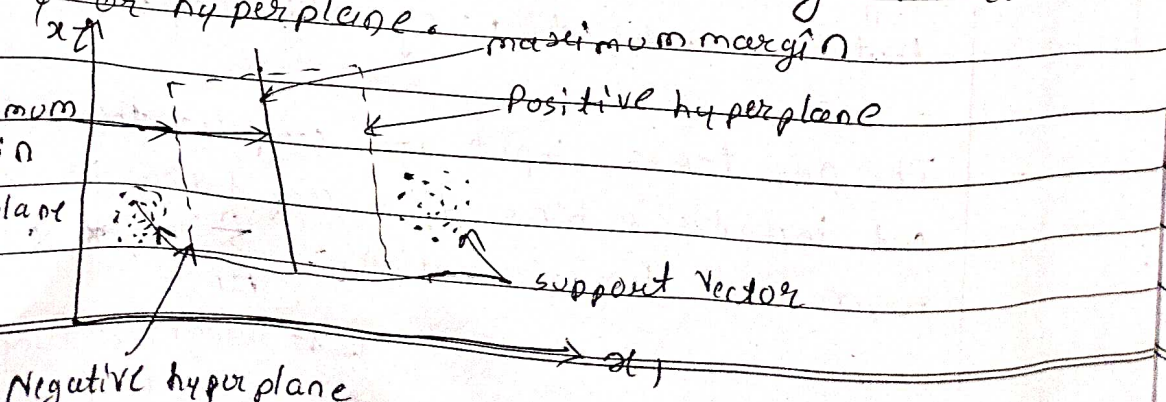
It determine weighted total is passed as an input to an activation fun to produce the output.

* Advantages of ANN

1) ANN have a numerical value that can perform more than 1 task simultaneously

* Support Vector Machine :-

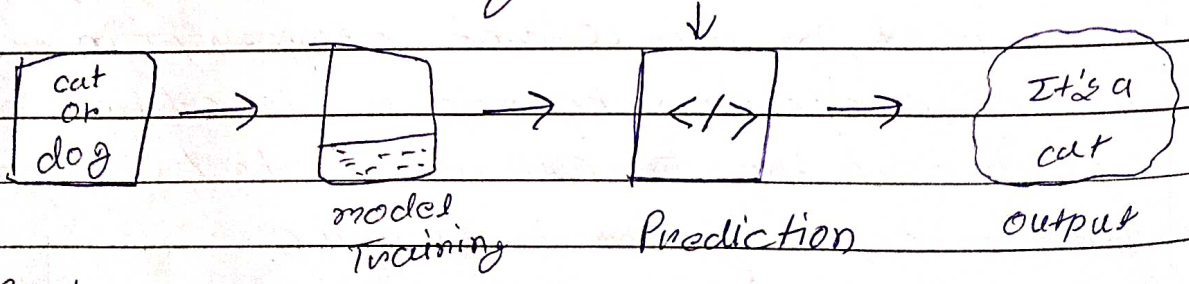
1) SVM or is one of the most popular supervised learning algorithm which is used for classification as well as regression problems.
2) Primarily it is used for classification problems in ML
3) The goal of SVM algorithm is to create the best line or decision boundary that can aggregate n dimensional space into classes so that we can easily put the new data point in the correct category in future.
4) This basted decision boundary is called a hyperplane
5) SVM chooses the extreme points / vector that helps in creating hyperplane.
6) This extrem cases are called as support vectors and hence algorithm is termed as SVM Support Vector algorithm
7) Consider the below diagram in which there are 2 different categories that are classified using a decision boundary or hyperplane.



Ex

SVM can be understood with the example that we have ~~using~~ used in KNN classifier. Suppose we see a strange cat that also have some features of dog so if we want a model that can actually identify whether it is a cat or dog so such a model can be created by using SVM algo.

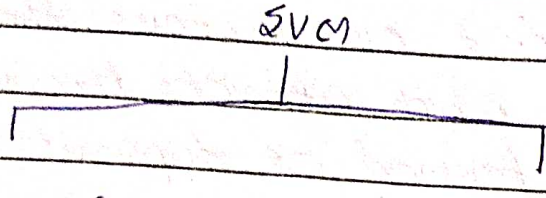
We were 1st train our model with lots of images of cats & dogs so that it can learn about the diffic different features of cat & dogs then we test it with this strenge creature so as support vector creates a decision boundry btn this two data (cat & dog) & choose extreme cases. (support vetors) it will see the extreme case of cat & dog on the basis of support vectors it will classify it with a cat Consider the below dig.



Past labelled data

- SVM algo can be is used in face detection, image classification and text categorisation.

Types of SVM.



linear SVM Non linear SVM

1) linear SVM

linear SVM is used for linearly separable data which means a data set can be classified into two class by using a single straight line. then such data is termed as linearly separable data & classified is used called as linear SVM classifier.

2) Non linear SVM

Non linear SVM is used for non-linearly separable data which means if a data set can not be classified by using a straight line then such data is termed as non-linear data & classifier use is called as non-linear SVM classifier.

Sub Algorithm

1) hyper plane.
2) Support vectors

1) hyperplane

there can be multiple lines / decision boundries to segregate the classes in N-dimensional space but we need to find out the best decision boundry that helps to classify the data points this best boundry is known as the hyperplane of SVM

2) Support Vector.

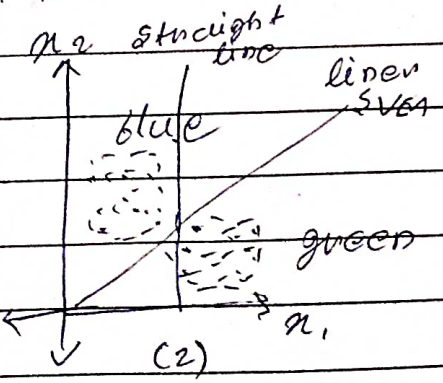
the data point or vectors that are closest to the hyperplane which affects the position of hyperplane are termed as Support Vector.

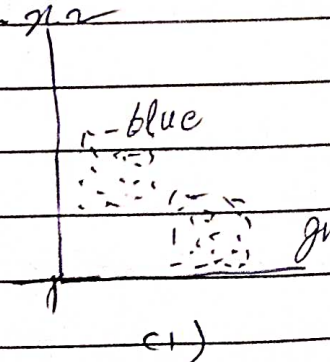
Since since this vectors support the hyperplane hence called Support vector.

Qu How does SVM works.

(1) linear SVM

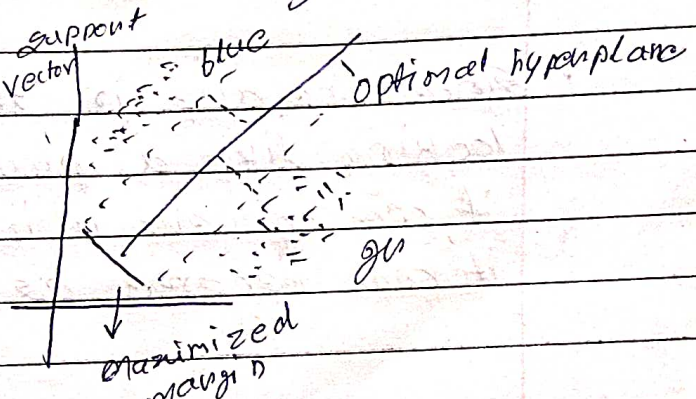
- the working of SVM algo can be understand by using an eg.
- Suppose we have a data set that has two tags (green & blue) & the data set has two feture x_1 & x_2.
- we want a classifier that can classify a pair x_1, x_2 of co-ordinates either green or blue





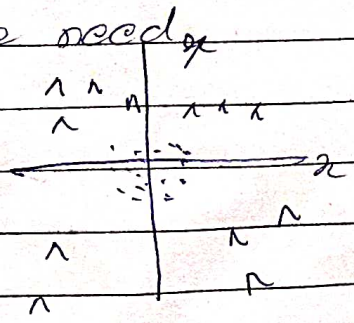
- So as it is 2D space so by ~~just~~ using a straight line we can easily separate these two classes but there can be multiple lines that can separate this classes.
- the SVM algo helps to find the best line or decision boundry these best boundry & region called as hyperplane.

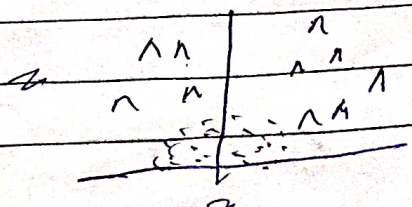
- SVM algo finds a closest points of the lines from both the classes this points are called support vector.
- The distance betn the vectors & hyperplane is called as margin
- The goal of SVM is to maximize this margin.
- the hyperplane with maximum margin is called the optimal hyperplane.



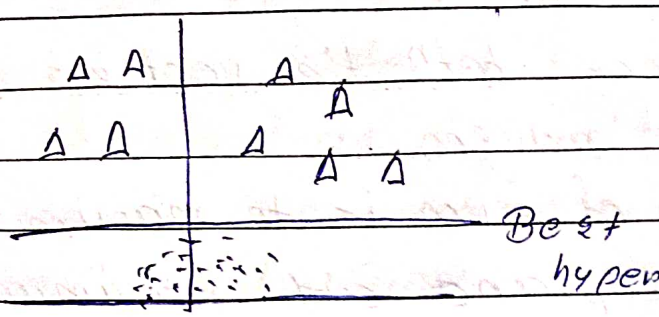
(2) Non-linear SVM

- If data is linearly arranged then we can separate it by using a straight line but for non-linear data we can not draw single straight line.
- Consider the below image.
- to separate this data points we need to add the one more dimension for linear data we have used two dimension x & y so for non linear data we will add third dimension z it can be calculated as $z = x^2 + y^2$
- By adding the third dimension the sample space will become as zero.

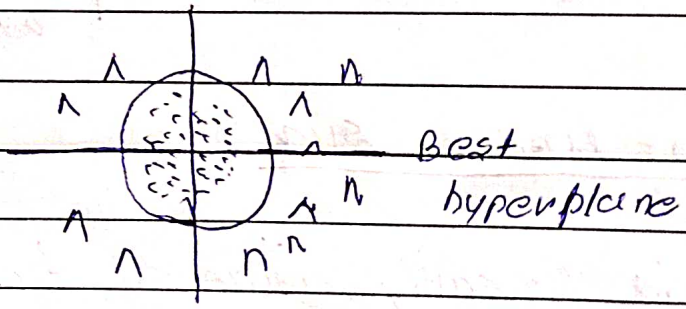




- So here ~~this~~ SVM will divide the data set into classes in the following way. Consider below img.



- Since we are in third ~~plane~~ space hence it is looking like a plane parallel to the x-axis if we convert it in 2D space with z = 1 then it will become as



hence we get a circumference of radius 1 in case of non-linear data.

14/12/22

★ Association Analysis:-

Association analysis is the task of finding interesting relationships in large datasets. These interesting relationship can take two forms.

1) Frequent Item sets.
2) Association Rules

DATE : __/__/__

PAGE No : _____

1) Frequent item ~~stet~~ sets.

→ Frequent item sets are a collection of items that frequently occur together.

2) Association Rules :-

→ Association rule mining finds interesting associations and relationships among large set of data items. This rule shows how frequently a item set occurs in a transaction. The association rule is very useful in analysing data sets. The data is collected using bar code scanner in super market.

Such databases consists of a large no. of transaction records which list all items bought by a customer on a single ~~pur~~ purchase.

☆ Applications of association rule learning.

1) Market basket analysis.
2) Bioinformatic
3) Medical diagnosis
4) Web mining
5) Scientific data analysis.
6) Protein Sequences

DATE : __/__/__ PAGE No : ____

* Terminology is related with association rule learning.

1) Item set.
In association rule learning a collection of zero or more items is known as item sets. If an item set consists of k-items, then it is known as k-item set.

Ex:- {pizza, burger, sweet}.
{bread, milk}.

2) Support
Support is the measure that tells us how frequent an item set is in a given dataset. It is calculated by dividing the total no. of occurences of an item set by the total no. of transactions.

formula: Support = $\frac{\text{no. of transaction containing}}{\text{total no. of transaction.}}$

3) Confidence :-
For a rule $X \xrightarrow{\text{associated}} Y$ confidence determines how frequently y has appeared in transaction that contain x

Confidence of (x, y) = $\frac{\text{support } x\,y}{\text{support of } x}$

4) Rule Generation :-
Whose objective is to extract all the high confidence rule from the frequent item set

DATE: __/__/__ PAGE No: ____

found in the previous ~~state~~ step. This rules are called strong rules.

* Association rule learning works on the concept of If and else statement such as If A then B.
Here If element is called anticident. and then segment is called consequent.
Association rule learning can be divided into 3 algo
✓ 1) Apriori Algorithm.
✓ 2) FP algo - (Frequency pattern).
x 3) Eclat algo (Equivalence class Transformation).

1) Apriori algorithm.
Apriori algo refers to the algo which is used to calculate the association rules between objects it means how 2 or more objects are related to one another In other words we can say the apriori algo is an association rule learning that analyse that people who bought product A also bought product B.

Ex: You must have noticed that the pizza shop seller makes a pizza soft drink and bread stick combo together He also offer a discount to their customer who buy this combos. Do you ever think

He thinks that customer who buy pizza also buy soft drinks and bread sticks however he

* The apriori algorithms refers to an algorithm that used in mining frequent product sets and relevant association rules. Generally the apriori algo. operates on a database containing a huge no. of transaction for example the items customers at a big bazar.

* Components of apriori algorithms.
1) Support
2) Confidence
3) Lift.

1) Support :- Support refers to the default popularity of any product you find the support as a quotient of the division of the no. of transactions comprising that product by the total no. of transactions.

Out of 400 transactions

$$\text{Support}(x) = \frac{\text{Freq}(x)}{T}$$

Frequent Pattern Growth

This algo is an improvement to the apriori method. A frequent pattern is generated without the need for candidate generation.. (use a similarity between items to recommended item similar to what the user likes.).

FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree. This tree structure will maintain the association between the item set. The database is fragmented using one frequent item. This fragmented one part is called Pattern fragment. The item set of this fragmented patterns are analysed with this method the search for frequent item set is reduced comparitively

FP Tree

Frequent pattern tree is a tree like structure that is made with the initial item set of the database. The purpose of the FP tree is to mine the most frequent pattern each node of the FP tree represent of an item set. The root node represents null while the lowermost node represents the item sets. The association of the nodes with the lower nodes that is the item sets with the other item sets are maintain while forming the tree.

DATE : __/__/__ PAGE No : ____

→ FP algorithm steps.

Step 1) The first step is to scan the database finds the occurances of the item set in the database. This step is same as the first step of apriori. The count of one item set in the database is called support count

Step 2) The step is to construct the FP tree for this create a root of the tree. The root is represented by null

Step 3) The next step is to scan the database again an examine the transaction and find out the item set in it. Examine the first transaction and find out the item set in it. The item set with the max count is taken at the top. The next item set with lower count and so on. It means that the branch of the tree is constructed with transaction item set in descending order of count.

Step 4) The next transaction in database is examin the item sets are ordered in descending order of count. If any item set of this transaction is already present in another branch for example [in the first transaction then this transaction branch would share a common prefix to the rule].

DATE: __/__/__ PAGE No: ____

This means that the common item set is link to the new node of another item set in this transaction.

Step 5) Also the count of the item set is incremented as it occur in the transaction ~~add~~. Both the common node and new node is increased by 1 as they are created and linked according to transactions.

Step 6) The next step is to mine the created FP tree for this to lowest node is examine first along with the links of the lowest node. The lowest node represents the frequency pattern length 1 from this traverse the path in the FP tree. This path is called conditional pattern base (conditional pattern base is sub database consisting of prefix path in the FP tree occuring with the lowest node. ~~node~~ suffix

Step 7 Construct a conditional FP tree which is form by a count of item sets in the path. The item sets meeting with threshold support are considered in the FP tree.

Step 8. Frequent pattern are generated from the conditional FP tree.

DATE : __/__/__ PAGE No :

Support threshold 50%
Confidence 60%
Transaction list of items.

T_1	I1, I2, I3
T_2	I2, I3, I4
T_3	I4, I5
T_4	I1, I2, I4
T_5	I1, I2, I3, I5
T_6	I1, I2, I3, I4

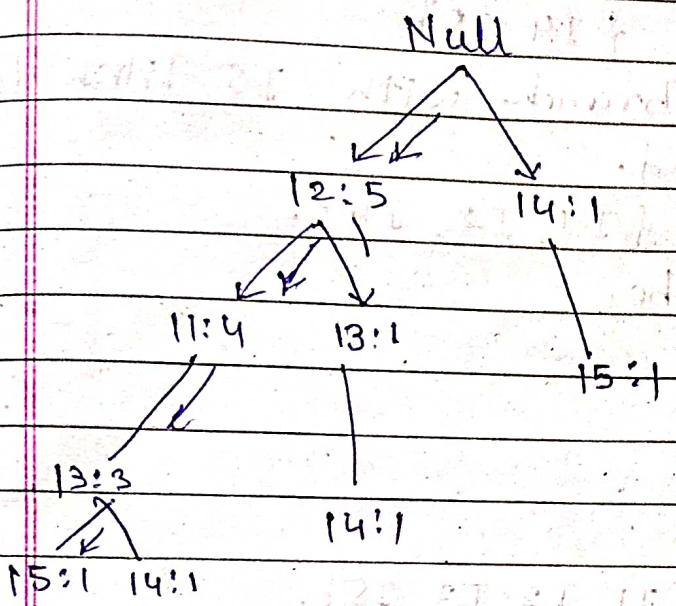
Support threshold = 50%.
= 0.5 × 6
= 3.
Minimum support = 3.

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

Arranging in descending order.

Item	Count	Order
I2	5	
I1	4	
I3	4	
I4	4	

DATE: __/__/__ PAGE No: _____



* Built FP tree :-

1) Considering the root node null
2) The first scan transaction T1, I1, I2, I3 contain 3 items {I1:1} {I2:1} {I3:1}
I2 is linked with child to root.
I1 is linked to I2 and I3 is linked to I1.
3) Transaction T2 {I2, I3, I4}.
where I2 is linked to root
I3 is linked to I2.
and I4 is linked to I3.
but this branch would share the I2 node as common as it is already used in T1
4) Increament the count of I2 by 1 and I3 is linked as a child to I2, I4 is linked as a child to I3. the count is
{I2:2} {I3:1} {I4:1}.

5) Transaction 3rd. {I4, I5}.
Similarly a new branch with I5 link to I4. as a child is created.

6) Transaction 4th {I1, I2, I4}.
The sequence will be. {I2, I1, I4} I2 is already linked to the root node hence it will be incremented by 1. As it is already linked with I2 in T1 {I2:3} {I1:2} {I4:1}.

7) Transaction 5th {I1, I2, I3, I5}.
The sequence will be {I2, I1, I3, I5} thus {I2:4} {I1:3} {I3:2} {I5:1}.

8) Transaction 6th {I1, I2, I3, I4}.
The sequence will be {I2, I1, I3, I4}. thus {I2:5} {I1:4} {I3:3} {I4:1}.

Advantages:

1) This algorithm needs to scan the database twice when compare to apriori which scan the transaction for each iteration

2) The pairing of items is not done in this algorithm making it faster

3) The database is stored in a compact version in memory

4) It is efficient and scalable for mining both long and short frequent pattern.

DATE : __/__/__ PAGE No : ____

Disadvantages :-

1) FP tree is more cumbersome and difficult to build than apriori.

2) It may be expensive.

3) The algorithm may not fit in the shared memory when the database is large.