→ Decision Tree :-

(1) A decision tree has a structure that consists of a root node, internal node, branches & terminal or leaf node

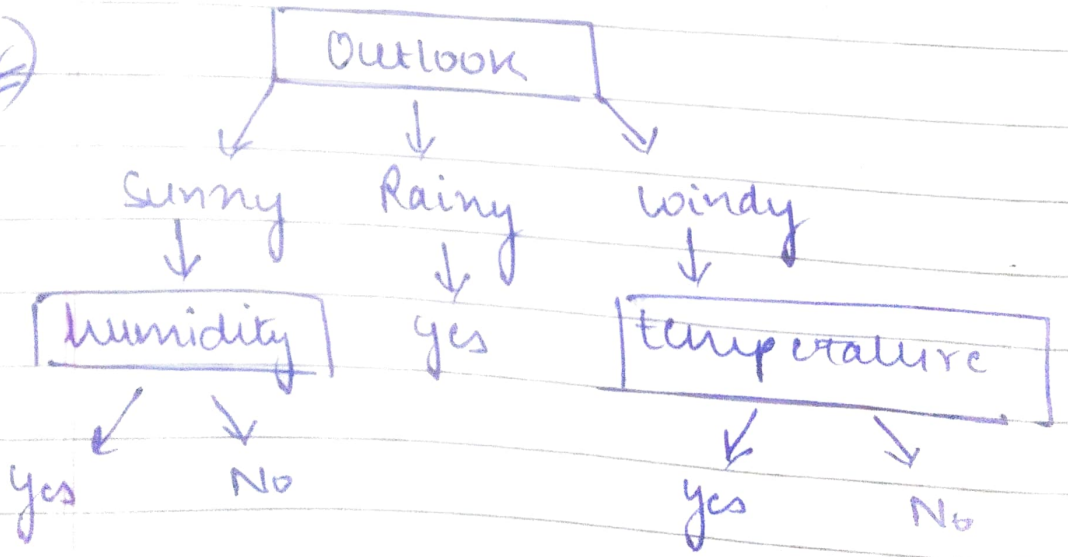
(2) Decision tree is a non linear representation. It is a classifier in the form of a tree.

(3) It is also known as tree structure classifier.

(4) It classifies instances by sorting them from root node to some leaf node.

(5) A decision tree is a graphical representation of all the possible solutions to a decision based on certain conditions.

Eg



→ Appropriate problems for Decision tree

ITEM

1) instances may be represented by ~~value~~ attribute value pairs
2) the target function has discrete output values
3) the training data may contain errors
4) the training data may contain missing attribute values

Which attribute is best classifier?
A statistical value called information gain measures ~~how well a given attribute seperates~~

b. Information Gain -
IG is the decrease in entropy after a dataset is split on the basis of an attribute.
Constructing decision tree is all about finding the attribute that returns highest information gain.

Gain = Entropy(s) - I(average attribute)

learning → removing unwanted data
impurity → degree of randomness tell how random is our data

2022/6/6

→ Entropy - (→ metric which measures the impurity) 1st step to solve desic tree

Entropy is the measure for uncertainity, impurity and information content.

It is an average optimal no of bits to encode information about certainity or or uncertainity in a given ~~(example~~ or dataset.

$$\text{Entropy} = \frac{-p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\left(\log_2\left(\frac{n}{p+n}\right)\right)$$

Algorithms

Cart Algo
uses gini index

ID3 (Iterative Dichotomiser 3)
uses entropy and Information gain to calculate homogenity of the sample

→ Strengths of Decision Tree :- SAND.

1) DT are able to generate understandable rules
2) Simple Calculation →
They perform calculation without requiring much computation
3) Attribute Types -
DT are able to handle both continuous & categorical values
4) Not include all attributes → process clean which fields are ...

TENN

NCEEN

NEENC

→ Weakness

① No continuous predictions -
less appropriate for tasks where we have to make continuous predictions

② ~~comp~~ can be computationally expensive to train

③ DT algo examines one attribute at a time for a node

④ Error prone

⑤ No sequence → problematic for time series data

→ Applications

Speech Recognition
Character Recognition
Traffic Risk Analysis
Remote Sensing
medical image Applications

→ Issues → HAHIA

① avoid overfitting the data → ID3 algo restrict to discrete values

② incorporating continuous valued attributes

③ alternative measures for selecting attributes

④ handling training examples with missing attribute values

⑤ handling attributes with different costs

2022/6/6

→ Instance Based Learning

Instance Based learning methods are conceptually straightforward approach

① Sometimes referred to as lazy learning
② In this are will not process the data to learn and develop a model
③ Instead are just store the example.
④ We will not immediately learn a model

) Comprises of 2 phases

ⓐ Testing Phase → when a new query as encountered, a set of similar related learning instances is retrieved from memory and used to classify or predict the target function value.

ⓑ Training Phase → we donot perform any processing on data. We simply store the data

↳ lazy learning (K- nearest neighbour)
↳ Radial Based functions (RBF)
↳ Case Based Reasoning (CBR)

Adv
① target function can be estimated locally and differently for each
② can use more complex symbolic representation

Disadv
① Cost of classifying new ins can be high
②

② storing train data requires less storage ③ storing for new instance is fast	may not have explicit model form ⑦ more storage scoring for new instance may be slow
Model Based learning	**Instance Based learning**
① store the model in suitable form ② Generalize the rules in form of model ③ Predict for unseen scoring instance using model	① no model to store no generalization before scoring ③ — — — — using training data
Can throw away training data after training	Training data must be kept
Train model from training data to estimate model parameters	Do not train data

→ KNN → Adv | disadv

This algorithm classifies a new instance by determining k vote similar instance & summarizing the output those k instances

If the target variable is discrete then it is a classification problem it selects the most common value among k instances by a majority vote.

If target ~~function~~ variable is continuous then it is a regression problem and hence the mean of k instances instance.

2022/6/6

Algo :-

(a) Continuous attribute -

Eulcdia distance blw 2 planes is with coor (x_1, y_1) & (x_2, y_2)

$$d = \sqrt{(x_2 - x)^2 + (y_2 - y_1)^2}$$

(b) Categorical attribute -

(1) If the value of 2 instances is same the distance $d = 0$ otherwise $d = 1$

(2) sort the distance in ascending order & select 1st k nearest training data instance

(3) Predict the class of test instance by majority vote or mean of k selected nearest neighbour

→ Input :-

Training data set = T

distance matrix = d

Test instance = t

nearest neighbour = k

O/p ⇒ predict class or category for test instance T

compute the distance blw test instance and every other instance in training dataset using a distance matrix

2022/06/6 21:05

weighted KNN nearest neighbour :-

① KNN algorithm has some limitations as its performance depends on
(i) Choosing nearest neighbour
(ii) Distance matrix
(iii) decision rule.

② The principle of weighted KNN is that K closest neighbours that are far away from the test instances. Here the weights are inversely proportional to distance

③ The selected K nearest neighbour can be assigned uniform way ~~that means~~

Algo :-

training dataset = T
distance matrix = d
weighted fun^n = w(i)
test instance t
no of K nearest neighbours K
~~Prediction → for test instance t~~

(a) Compute the inverse of each distance of K selected nearest instance
(b) find sum of inverse
(c) Compute the weight by dividing each inverse distance by sum
(d) add the weights of the same class
(e) Predict the class by choosing the class with maximum votes.

→ Locally Weighted Regression :-

Generalized form of KNN

~~(1) Local~~ → (1) Local → because the function is approximated based only on the data near the query point.

(2) Weighted → because the contribution of each training example is weighted by its distance from query point.

(3) Regression – because this term is widely used in statistical learning in the problem of approximating real valued functions

LWR uses nearly all distance weighted training example to form the local approximation function

LWR is also referred to as memory based method as it requires training data while prediction but uses only the training data distances locally around the point of interest.

T = training dataset

Hypothesis function H(x)

the predicted target ~~value~~ output is a linear function where

b_0 is the intercept &

β_1 is the coefficient of x

$$H(x) = \beta_0 + \beta_1 x \quad \text{---} (1)$$

Cost function is such that it minimizes the error difference b/w the predicted value $H(x)$ & true value or the actual value Y.

Cost function →

$$J(\beta) = \frac{1}{2} \sum_{i=1}^{m} \left(H_\beta(x) - Y_i \right)^2 \quad \text{---} (2)$$

m = no of instances in the training dataset

Now the cost function is modified for locally weighted linear regression

Cost function is

$$J(\beta) = \frac{1}{2} \sum_{i=1}^{m} w_i \left(H_\beta(x) - Y_i \right)^2 \quad \text{---} (3)$$

w_i = weight associated with each instance x_i

→ Case Based Reasoning

Case Based Reasoning is a problem solving technique that matches a new instance with the previous solved instance and its solution is stored in the database

In design and implementation of any CBR application four Rs / steps are involved.

(i) Retrieve
(ii) Reuse
(iii) Revise
(iv) Retain / Stored

A CBR has 2 properties :-

(a) Lazy learning
(b) Classification is different for each instance

CBR only works on or is represented as symbols not values

2022/6/6 21:

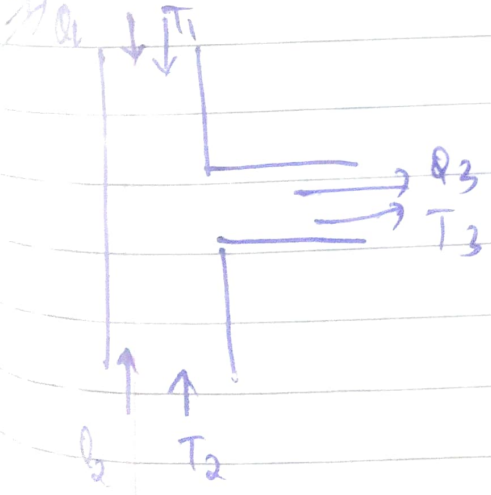
There are 3 components of CBR

1. Similarity functions or distance measure — SAS
2. Approximation or adjustment of instance
3. Symbolic Representation

To measure or implement CBR we use CADET system — FAPS

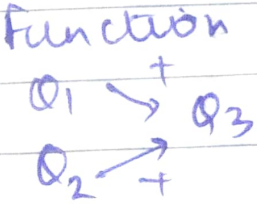
CADET system stands for case Based Design Tools. It has 75 predefined libraries. These libraries will be utilized to find the solution of new instance

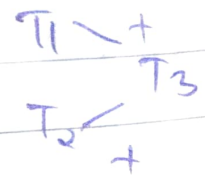
4) A stored case T junction pipe



Q = water flow
T = temperature

Function





Limitation

1. It has limited capabilities for combining & adapting multiple retrieved cases to (form the final design) and relly (heavily on the user for adaption stage of process)

2. Doesnot have the range of analysis algo that is needed to refine the abstract conceptual design into the final design.

→ ~~Comparison with KAN~~ :-

→ Application of CBR :-

1. Reasoning about new legal cases based on previous cases and their solution
2. conceptual design of mechanical devices based on stored library of previous designs
3. solving, planning & scheduling problems by receiving & combing portion of previous solutions to similar problems

2022/6/6 21:05

Model Based Learning -

Model based learning describes the assumptions about the problem domain in the form of model.

↳ supervised learning mostly used in classification

→ Support Vector Machine :-

① Support vectors are data points that lie closest to the decision surface or hyperplane

② They involve attention of hyperplanes (decision boundaries) that segregate data into classes

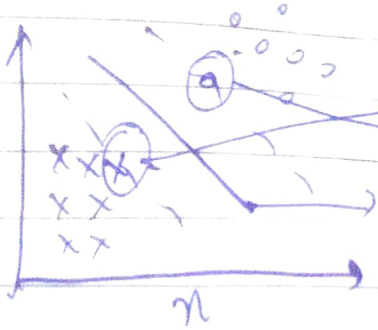
③ The aim of SVM is to produce a decision plane that defines the boundary b/w the classes to classify the data points

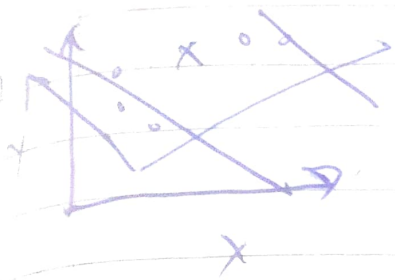
④ The decision boundary is drawn with following characteristics :-

① The distance should be maximized between the line & nearest data points
② Should avoid miscalculation of data

Margin → amount of space b/w 2 classes as defined by the hyperplane.

Marginal distance → sum of shortest distance to the closest positive & negative points

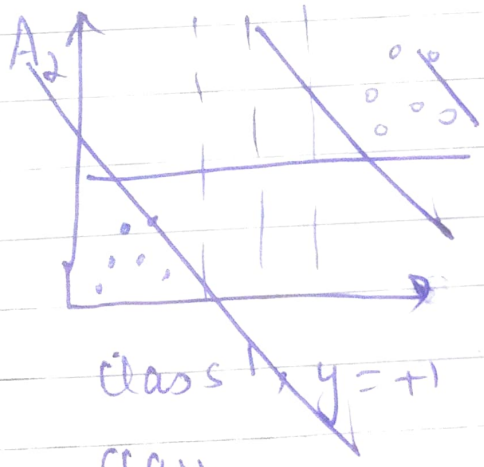


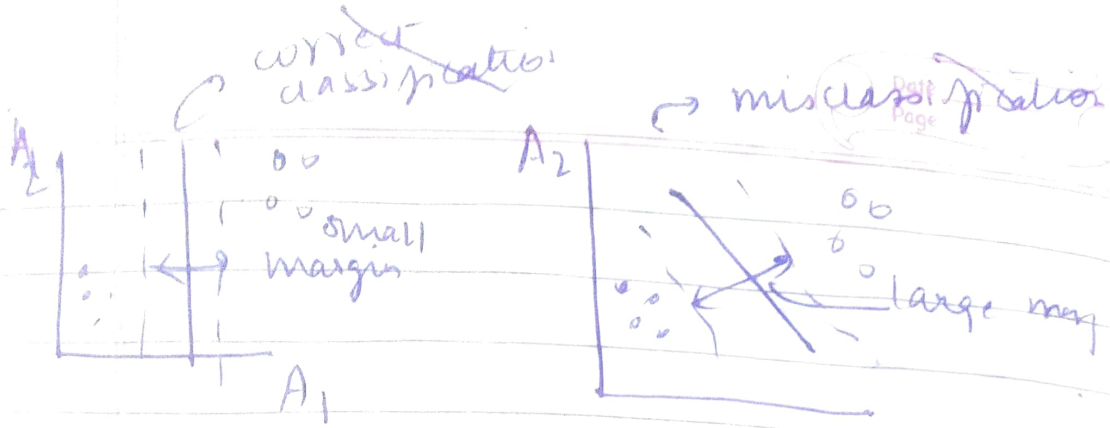


ⓐ SVM Linearly Separable

2D training data set are linearly separable. There are infinite no of separating hyperplanes or decision boundaries

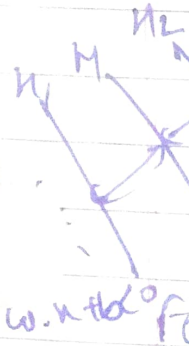
In this we have to find the "best" one that will have minimum classification error





the one with larger margin should have greater generalization accuracy

A seperating hyperplane can be written as



$$W \cdot X + b = 0$$

$W = \{w_1, w_2 \ldots w_n\}$ weight vector

w_n b's

$b \rightarrow$ scalar (bias)

For 2D

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

hyperplanes -

H_1: $w_0 + w_1 x_1 + w_2 x_2 \geq 1$ for $y_i = +1$

H_2: $w_0 + w_1 x_1 + w_2 x_2 \leq -1$

H_2: $wx+b \geq 0$ for $y = +1$

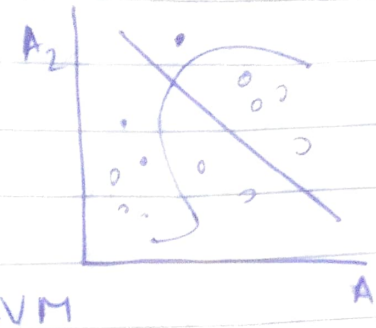
H: $wx+b < 0$ for $y = -1$

any training tuples that fall under hyperplanes H_1 & H_2 are support vectors.

⑥ Non Linear SVM :

~~Kernel trick~~

This 2D figure show linearly unseperable SVM



Kernel Trick means replacing the dot product in the mapping function with a kernel function

Kernel function returns the inner product b/w 2 pts in a suitable feature space.

Types :-

① Linear Kernel

$$K(x, y) = (x^T y)$$

$$\boxed{K(x, y) = \phi(x)\,\phi(y)}$$

my are data points we want to classify

② Polynomial Kernel

$$\boxed{K(x, y) = (x^T y)^d}$$

d = degree of polynomial

$d = 2 \Rightarrow$ quadratic kernel

$d = 1 \Rightarrow$ more generalised form of linear kernel

③ RBF / Gaussian Kernel

It helps SVM classify in infinite dimensions which is not possible to visualize manually

It can map input space in infinite dimensional space.

$$K(x, x') = \frac{Exp(\|x - x'\|^2)}{2\sigma^2}$$

$$\gamma = \frac{1}{2\sigma^2}$$

→ Bayesian Belief Network

Bayesian Belief Network describes the probability distribution over the set of variables.

Let us consider ~~random~~ variables

Let us consider

$\langle y_1, y_2, \cdots, y_n \rangle$ ⇒ set of random variables

Each variable has a set of values say $V(y_i)$

Here we can define the joint space by cross product

$V(y_1) \times V(y_2) \times \cdots \times V(y_n)$

2022/07/19 21:0

probability distribution over the joint space is known as joint probability distribution

BBN describes joint probability distribution for set of variables

Application

used in diagnosis
Classification
Decision making
Prediction

Joint probability Distribution –

Joint probability distribution for set of values $\langle y_1, y_2 \ldots y_n \rangle$ to the tuples of network variables $(Y_1, Y_2 \ldots Y_n)$ can be computed by

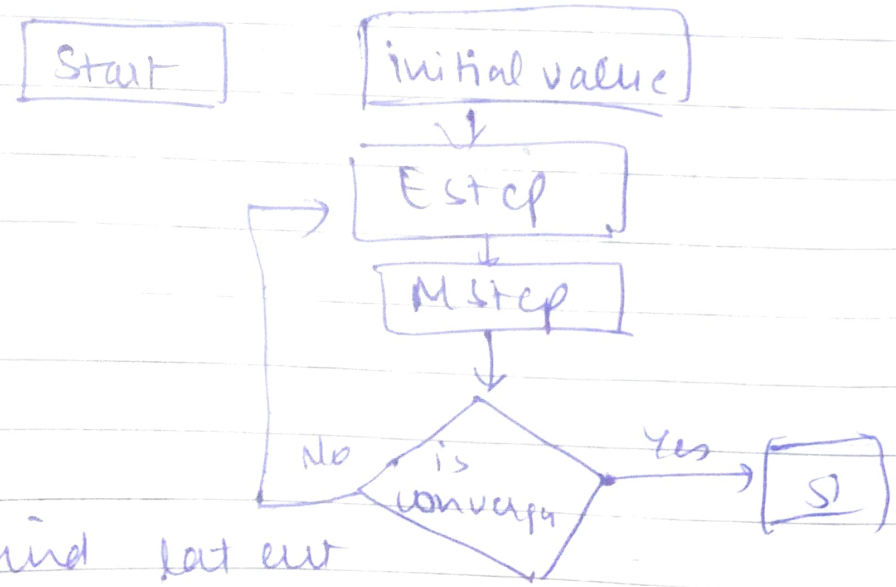
$$P(Y_1, Y_2, \ldots Y_n) \sum_{i=1}^{n} (Y_i \mid \text{Parent}(Y_i))$$

→ EM Algorithm –
Expectation Maximization Algorithm

It is used to find the latent variable.
It is basic for unsupervised clustering algorithm.

Use :-

① find missing data
② basic for unsupervised learning algo



③ find latent
④

Adv

1. guarantees that likelihood will increase each iteration
2. E step and M step are pretty easy & can often solve complex problems
3. Soln to M step often exists in closed form

Disadv

1. slow convergence
2. requires both probabilities
3. makes convergence to local optima only