UNIT-IV Classification

- Supervised learning (classification)
  - New data classified based on training set
- Unsupervised learning (clustering)
  - establishing classes on clusters to identify unknown data

Classification vs Numeric prediction

- classification (discrete, Nominal data) to predict class for New data
- Numeric prediction (models continuously valued functions)
  → Predicts unknown or missing values. → Regression analysis is used

Applications:

- fraud detection
- medical diagnosis

Classification

↓

Two steps

1. **model construction**
   - describing a set of Predetermined classes [class-label attribute]
   - Training set used
   - model represented as rules, decision trees or formula

   ↓ Training data

   Classification Algorithm

   ↓

   Classifier (Model)

   → Predict If (Accuracy) is great

   → Classifier

   Trained

2. **model usage**
   - estimate accuracy [compare Predicted ↔ actual]

     ↓

     Accuracy rate = % of samples correctly classified

     If (Accuracy) is great then use model to classify new data

     ↓

     Testing data → Classifier → unseen data

     ↓

     Predict.

If test set is used to select models then it is called validation test

Issues with classification & prediction

- **Data preparation**
  - data cleaning
  - feature selection
  - data transformation (normalization)
- **Evaluating classification methods**
  - Prediction accuracy
  - Speed & scalability
    - time to construct
    - & use model
    - efficiency
  - Robustness
  - interpretability

Classification Techniques

- Decision Tree Based methods
- Rule based methods
- memory based Reasoning
- Neural Nets
- Bayseian Belief Nets & Naive Bayes
- SVM's

Decision Tree induction

- Basic algorithm:
  - Tree is constructed in Topdown · recursive divide & conquer manner
  - Starting, Training examples at root (all)
  - Attributes are categorical (if continuous valued, discreted in advance)
  - examples are partitioned recursively based on selected attributes
  - Test attributes are selected on basis of Statistical Measure (Information gain)
- Conditions for stopping partitioning:
  - All samples of a given node belong to same class
  - no remaining attributes. (Majority voting applied for classifying leaf)
  - no samples left

Entropy [A measure of uncertainity]

- high entropy → high uncertainity
- low entropy → low uncertainity

Conditional entropy

$$H(Y/X) = \sum_x P(x)\, H(Y/X=x)$$

Attribute selection measure : [Information gain ID3/C4.5]

successor of

- select attribute with highest information gain

→ expected information (entropy)

let D → Tuple , P → Probabability

C → class

$$Info(D) = \sum_{i=1}^{M} P_i \log_2(P_i)$$

→ Information need after splitting D into V partitions using A

$$Info_A(D) = \sum_{j=1}^{V} \frac{|D_j|}{|D|} \times Info(D_j)$$

→ Information gained by branching on attribute A :

$$gain(A) = Info(D) - Info_A(D)$$

Attribute selection : Information GAIN

eg:- class 'P' : buys-computer = "yes"
class 'N' : buys-computer = "NO"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(9/14) - 5/14 \log_2(5/14) = 0.940$$

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

• 5 samples out of 14 for age <=30
• 2 Yes & 3 NO.

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31..40	4	0	0
>40	3	2	0.971

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

schema

age	income	stud	credit rati..	b..

∴ similarly
Gain(income) = 0.029
Gain(student) = 0.151
Gain(credit-rating) = 0.048

Computing Information GAIN for continuous valued attributes

• let A → continuous valued attribute

Steps:-
1. sort A in increasing order
2. find mid $(a_i + a_{n-1})/2$ and select it as Best split point
3. split tuple 'D' into D_1 & D_2
$A(\leq midpoint) \rightarrow D_1$
$A(> midpoint) \rightarrow D_2$

Gain ratio for attribute selection

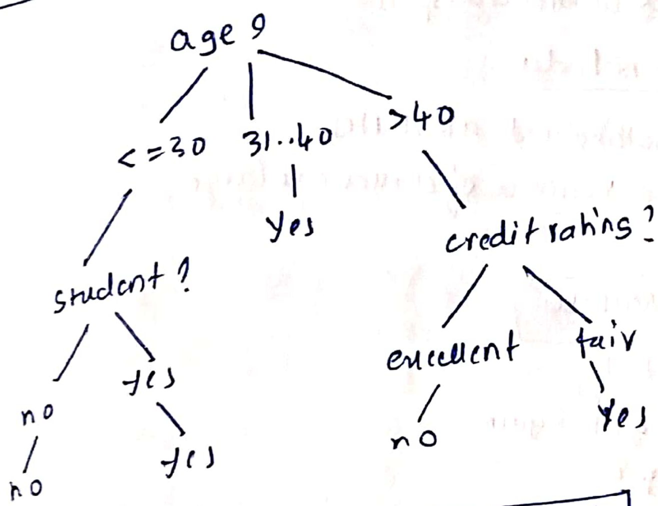
- Information gain measure is biased towards attributes.
- C4.5 (a successor of ID3) uses gain ratio to overcome

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$\text{GAIN Ratio} = \text{Gain}(A) / \text{SplitInfo}(A)$$

→ attribute with max gain ratio is selected as splitting attribute.

Decision tree - after Information gain



Gini Index (CART, IBM Intelligent Miner)

- If a data set D contains examples from n classes then gini index

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2 \qquad p_j = \text{relative frequency of class } j \text{ in } D$$

If D on A is split into D_1 & D_2

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- reduction in Impurity

$$\Delta gini(A) = gini(D) - gini_A(D)$$

eg: D → 9 tuples in buys_computer = "yes" & 5 in "no"

$$gini(D) = 1 - (9/14)^2 - (5/14)^2 = 0.459$$

Suppose D → D_1 & D_2
10 4
(low, medium) (medium+1, high)

$$= \frac{10}{14}\left(1-\left(\frac{7}{10}\right)^2-\left(\frac{3}{10}\right)^2\right)+\frac{4}{14}\left(1-\left(\frac{2}{4}\right)^2-\left(\frac{2}{4}\right)^2\right)$$

$$+\left(\frac{4}{14}\right) gini(D_2)$$

$$= 0.443$$

$$= Gini^{D}_{income \in \{high\}}$$

- $Gini_{(low, high)}$ is 0.458, $Gini_{(medium, high)}$ is 0.450
  - ∴ split the {low, medium} & {high}

<u>disadvantage of Gain Ratio</u>

- tends to prefer unbalanced splits

<u>disadvantage of Gini Index</u>:

- biased to multivalued attributes
- difficult when number of classes are large

Other popular Selection Measure

- <u>CHAID</u>: based on χ^2 test
- <u>C-SEP</u>: better than gini & gain
- <u>G-static</u>: closer to χ^2
- <u>MDL</u> (minimum description length) → Prefers simple solution
- <u>CART</u>: find multivariate split based on linear combination of attributes.

Overfitting & Tree pruning

- <u>Overfitting</u> :- A induced tree may overfit the training data
  - Too many branches, noise
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
- <u>Prepruning</u>: Halt tree construction early & remove branches falling below a goodness threshold
- <u>Post-pruning</u> :- Remove branches from fully grown trees

Enhancements to Basic Decision Tree Induction

- Allow for continuous valued attributes.
- Handle missing attribute values
- attribute construction.
  - create new attributes based on existing
  - reduces fragmentation, repetition & replication

classification in large databases → why decision trees so popular?

- can classify Millions of examples & hundreds of attributes using reasonable speed in decision trees.
- Scalable
- relatively faster running speed.
- convertible to simple rules.

Scalability framework for Rainforest

- Builds on AVC list AVC(Attribute, value, class_label)
- AVC-set : (of attribute X)
  → Projection of training data onto attribute X
- AVC-group (of a node N)
  → set of AVC sets. of projection attributes at node N

eg: AVC Set on income

income	Buys_computer	
	yes	NO
high	2	2
medium	4	2
low	3	1

BOAT (Bootstrapped Optimistic Algorithm for Tree construction)

- uses a statistical technique called bootstrapping to create several smaller samples (subsets), each fit in memory.
- each subset creates a new Tree, resulting in several trees

DB, an incremental alg.

Adv: requires only two scans of DB, an incremental alg.

Bayesian classification why?

- A statistical classifier → does probabilistic prediction
- works on Bayes Theorem
- good performance relatively along with Neural Nets & Decision trees
- Incremental : improves with data
- standard : Bayesian methods are intractable & Provides decision making

Baye's theorem : Basics

- Total probability theorem:

$$P(B) = \sum_{i=1}^{M} P(B/A_i) P(A_i)$$

- Bayes theorem:

$$P(H/x) = \frac{P(X/H) P(H)}{P(X)}$$

$$= P(X/H) \times P(H)/P(X)$$

X ⇒ data sample
H → hypothesis that X belong to class C

→ Classification is to determine $P(H/x)$ [Posteriori probability]

$P(H)$ → initial probability
$P(X)$ ⇒ Probability that sample data is observed
$P(X/H)$ → likelyhood.

Theorem:-

Given training data X, posteriori probability of a hypothesis H, $P(H/x)$ follows Bayes theorem

$$P(H/x) = \frac{P(X/H) P(H)}{P(X)}$$

Simply ⇒ Posteriori = likelihood × Prior / evidence

→ let attributes $X = \{x_1, x_2, \ldots x_n\}$

→ classes $C_1, C_2 \ldots C_m$

→ classification --→ max Posteriori

$\therefore P(C_i/X)$

$$\therefore P(C_i/X) = \frac{P(X/C_i)\,P(C_i)}{P(X)}$$

as $P(X)$ is constant for all classes

$$\boxed{P(C_i/X) = P(X/C_i)\,P(C_i)}$$

Naive Bayes classifier

- If attributes are conditionally independent, then it greatly reduces the computational cost.

$$\therefore P(X/C_i) = \prod_{k=1}^{n} P(x_k/C_i) = P(x_1/C_i) \times \ldots \times P(x_n/C_i)$$

Dataset

age	income	student	credit-rating	buys-computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31..40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31..40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31..40	medium	no	excellent	yes
31..40	high	yes	fair	yes
>40	medium	no	excellent	no

eg:-

- $P_i(C)$:=

  P(buys_computer = "yes") = 9/14 = 0.643

  P(buys_computer = "NO") = 5/14 = 0.357

- Compute $P(X_i|C_i)$ for each class

  P(age <= 30 | buys_computer = "yes") = 2/9 = 0.222

  P(age <= 30 | buys_computer = "no") = 3/5 = 0.6

  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444

  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4

  P(student = "yes" | buys_computer = "yes") = 6/9 = 0.667

  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2

  P(credit = "fair" | buys_computer = "yes") = 6/9 = 0.667

  P(credit = "fair" | buys_computer = "no") = 2/5 = 0.4

- X = (age <= 30, income = medium, student = yes, credit = fair)

  $P(X|C_i)$: P(X | buys_computer = "yes")

  = 0.222 × 0.444 × 0.667 × 0.667

  = 0.044

  : P(X | buys_computer = "no")

  = 0.6 × 0.4 × 0.2 × 0.4 = 0.019

  $P(X/C_i) * P(C_i)$: P(X | buys_computer = "yes") * P(buys_computer = "yes")

  = 0.044 × 0.643 = 0.028

  P(X | buys_computer = "no") * P(buys_computer = "no") = 0.007

  ∴ X belongs to class ("buys_computer = yes")

Problem in Bayesian

- Naive Bayesian prediction requires each condition prob be non-zero, otherwise the predicted probability will be zero.

$$P(X/C_i) = \prod_{k=1}^{n} P(x_k/C_i)$$

- use laplacian correction

Advantages of Naive Bayes classifier

- easy to implement
- good results in most of cases

disadvantage

- If there is class conditional independence accuracy is lost

Rule-Based classification

- representing the knowledge in the form of If-then rules

R: If $\underbrace{\text{age=youth and student = yes}}_{\text{Rule antecedent / Precondition}}$ then $\underbrace{\text{buys-computer = yes}}_{\text{Condition}}$

- Assessment of Rule is done by Accuracy & coverage

n_{covers} = number of tuples covered by R

$n_{correct}$ = number of tuples correctly classified by R

$$\text{Coverage}(R) = \frac{n_{covers}}{|D|}$$

$|D| \rightarrow$ Training data set

$$\text{accuracy}(R) = \frac{n_{correct}}{n_{covers}}$$

- If more than one rule are triggered, we need conflict resolution.
  - size ordering
  - class based ordering : decreasing order of prevalence
  - Rule-based ordering (decision / Priority list).

Rules extraction from decision tree

- Rules are easier to understand
- one rule for each path from root to leaf
  - leaf holds class prediction
- Rules are mutually exclusive & exhaustive

Rule Induction : Sequential Covering method

- Sequential covering algorithm :-

  extracts rules directly from training data.

eg: Algorithms are : FOIL, AQ, CN2, RIPPER

[[Rules are learned sequentially, each for given class c, will cover many tuples of c]]

- Steps :-

  1. Rules are learned one at a time
  2. each time a rule is learned, the tuples covered by rules are removed
  3. Repeat the process until no more training examples left

```
while (enough target tuples left)
    generate a rule
    remove positive target tuples satisfying this rule
```

- Rule generation

```
while (true)
    find best predicate P
    If foil-gain (P) > threshold then add P to current rule
    else break.
```

How to learn one rule ?

- start with most general rule possible (condition = empty)

- add new attributes by adopting a greedy depth-first strategy
- Rule quality measures (in FOIL & RIPPER)
  - assess info-gain

$$FOIL\_Prune(R) = \frac{pos - neg}{pos + neg}$$

+ve examples

-ve examples

$$FOIL\_Gain(R) = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right)$$

Model evaluation & selection

- use validation test set of class-labelled tuples instead of Training set when assessing accuracy
- methods for estimating a classifier's efficiency.
  - Holdout method, random Subsampling [Partitioned into random independent subsets & cross-validated]
  - cross-validation → k-fold, k=10 (Popular) → randomly partition data into k mutually exclusive subsets of equal size
  - Bootstrap (.632 bootstrap) k = number of tuples (leave one out)
- Comparing classifiers :-
  - → Confidence intervals
  - → Cost-benefit analysis & ROC curves

Classifier evaluation metrics

- precision : % of tuples labelled by classifier as +ve are really +ve

$$Precision = \frac{TP}{TP + FP}$$

[exactness]

- Recall : % of what tuples did classifier labelled as +ve

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

Perfect score is 1.0

- F measure (F-score) : harmonic mean of precision & recall

$$F = \frac{2 \times Precision \times recall}{Precision + recall}$$

- F_β (weighted measure of Precision & Recall)

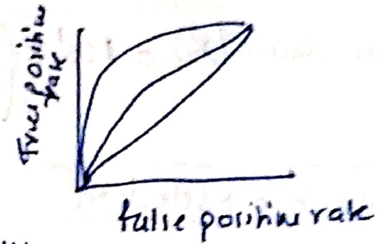
$$F_\beta = \frac{(1+\beta)^2 \times Precision \times Recall}{\beta^2 \times Precision + Recall}$$

Estimating confidence intervals : NULL hypothesis

- If we can reject NULL hypothesis (M1 & M2 are classifiers & are the same). then the difference between M_1 & M2 is statistically significant [choose lower error rate one]

→ t-test → Pairwise comparision

Model Selection: ROC curves



- Receiver operating characteristic curves for visual comparision of classifiers.
- A mode with perfect accuracy will have an area 1.0

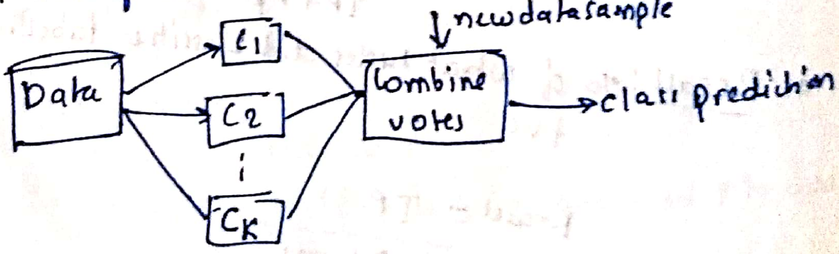
Issues affecting model selection

- Accuracy
- Speed
- Robustness
- scalability
- Interpretability

Techniques to improve classification Accuracy

ensemble methods

- use a combination of models to increase accuracy
- combine a series of k learned models M1, M2 - - Mk with aim of creating an improved model M*



Ensemble methods

- Bagging [diagnosis based on multiple doctors majority vote]
- Ada Boost / Boosting · [consult several doctors, based on combination of weighted diagnoses — weight assigned on previous diagnosis accuracy]
- ensemble [Random forest → Forest-RI (random input selection)
  → Forest-RC (random linear combination)

Handling different kinds of cases in classification

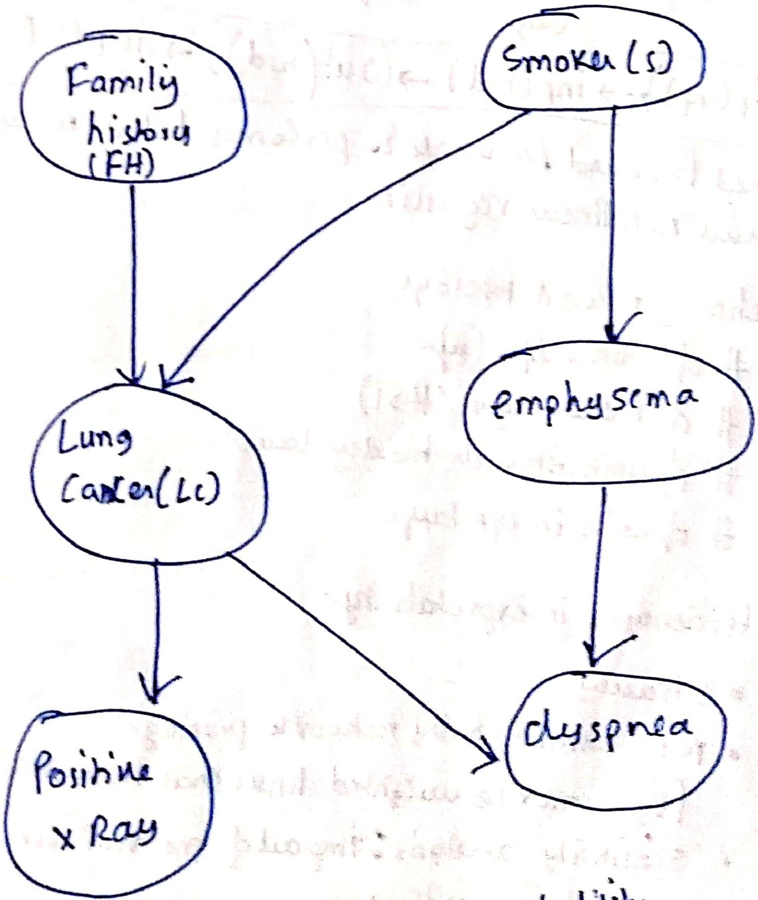
- Traditional methods [redacted] assumes a balanced distribution of data.

- Typical methods for imbalance data in 2-class classification
  - → oversampling :- re-sampling of data from positive class
  - → under sampling :- randomly eliminate tuples from negative class
  - → Threshold moving: moves the decision threshold so that rare-class tuples are easy to classify
  - → ensemble methods : ensemble multiple classifiers as discussed before.

Bayesian Belief Networks / Bayesian networks

- bayesian belief networks, probabilistic networks allow class conditional independencies between subsets of variables.
- uses a directed acyclic graph. [redacted]

eg:



→ uses Conditional probability.

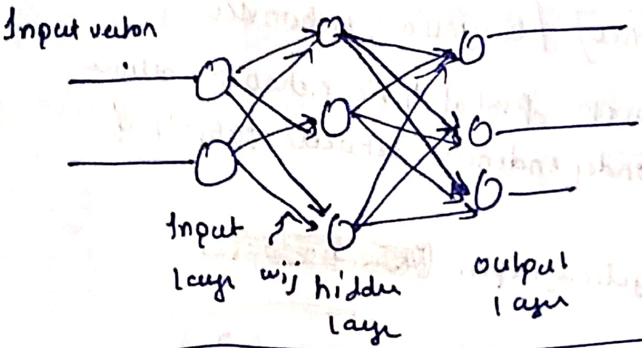
classification by backpropagation [connectionist learning]

- A nueral network learning algorithm
- A set of connected ip/op units where each connection has a weight associated with it.
- During learning phase, the network learns by adjusting weights to minimize mean squared error.
- ·Modifications are made in backward direction

Neural Network as a classifier

↳ Advantages:-
- high tolerance to Noisy data
- Ability to classify untrained patterns.
- Algorithms are parallel.

dis:- long training time



(Op(inp)) → inp(hid) → (Out(hid)) → inp(OP) → o/p

↳ Feed forward Network & performs better when used non linear regression

– defining network topology
  - # of units in i/p layer
  - # of hidden layers (if >1)
  - # of units in each hidden layer.
  - # of units in o/p layer

– efficiency & interpretability
  - efficiency
  - Rule extraction by network pruning [By removing weighted links that have least effect]
  - sensitivity analysis:- impact of one variable input on network output.

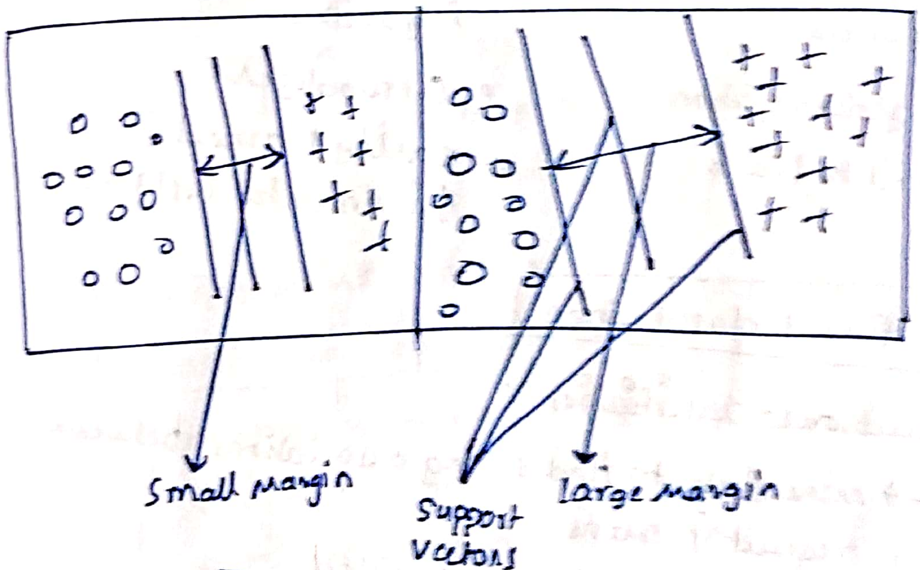
– discriminating classifiers
  - high accuracy v/s long training time

Support Vector Machines

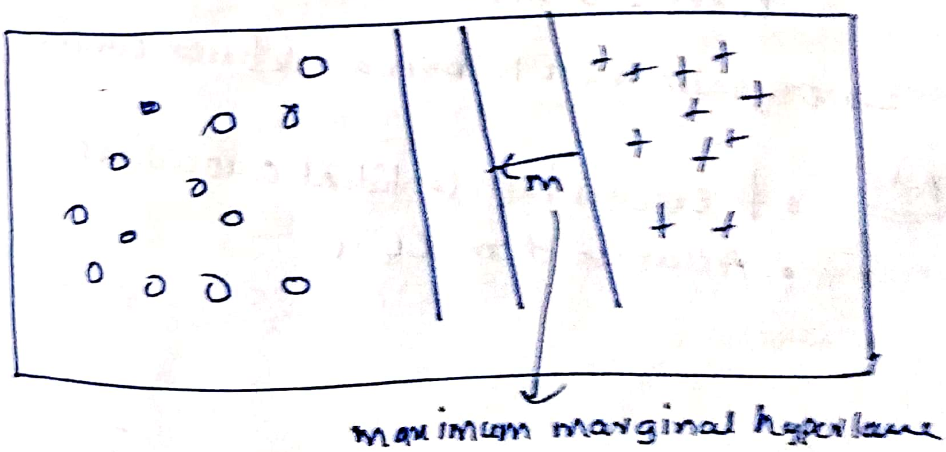
- A relatively new classification method for both linear & nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension.
- with the new dimension, it searches for linear optimal seperating hyperplane (decision boundary)
- with appropriate non linear mapping, data from two classes can always be seperated by a hyperplane.
- SVM finds this hyperplane using support vectors ("essentially training tuples") & margins (defined by support vectors)

Applications : handwritten digit recognition, object recognition etc.

[classification & numeric prediction]



→ When data is $\boxed{\text{linearly seperable}}$ === Concept



Why SVM effective on high dimensional data

- The complexity of trained classifier is charecterized by # of support vectors rather than dimensionality of data.
- with small # of support vectors, we can have good generalization

SVM Linearly Inseperable

- Transform the original input data into a higher dimensional space.
- search for linear seperating hyperlane in the new space.

SVM V/S Neural Nets

SVM	NN
deterministic Algorithm	Non-deterministic Algorithm
Nice generalization	Generalizes
Hard to learn	easily learned in Incremental fashion.

Pattren Based classification

- Association classification:
  - → mine data to find strong associations between frequent patterns
  - → Association rules are generated

$$P_1 \wedge P_2 \ldots \wedge P_i \rightarrow \text{"}A_{class} = C\text{"} \; (conf, sup)$$

  - → organise rules to form a rule based classifier

Adv:
- It explores high confident associations
- Accurate than C4.5

Typical Associative classification methods

- CBA (classification based on Associations)
- CMAR (classification based on Multiple Association Rules)
- CPAR (classification based on predictive Association Rules)

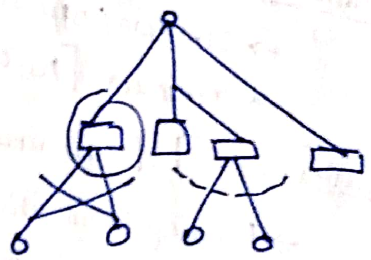
→ Discriminative Frequent Pattern - Based classification

- DDP Mine (Branch & Bound search)
- FPtree pruning

$sup(child) \leq sup(Parent)$

$Sup(b) \leq sup(a)$

Maximize $IG(C/b)$



Lazy v/s eager learning

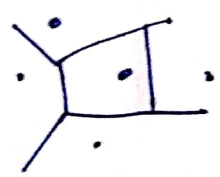
- lazy learning :- (instance-based learning)

  Simply stores data for minion processing & waits until it given a test tuple

- early learning : constructs a classifier before receiving data to classify.

→ Instance Based methods.

① K-nearest neighbour approach :-



- Instances represented as points in euclidean space

Vonori diagram is used

- nearest neighbour defined in terms of euclidean distance (x_1, x_2)
- Target function can be real or discrete valued
- for discrete valued, K-NN returns most common value among K training examples.

② locally weighted regression (uses local approximation)

③ case-based-reasoning (uses symbolic representations & inference)