Classification
9/3/20

* classifn predicts categorical (discrete, unordered) labels
prediction model continuous

* Bank loans apply → risky / safe } classification

* Prediction model based on rainfall, year, season → crop yield
→ done by Regression - Linear

* classification is a two step process
→ Training (learning phase).
→ Testing.

* Model is learnt from training data
Hence referred as supervised learning.

* clustering is an unsupervised learning mech

* Prediction ⇒ $y = f(x)$ → i/p vector
→ continuous value o/p

! classifier ⇒ mapper between x and y

data cleaning - missing / noisy values.

Relevance Analysis (Data selection)

Attribute subset selection)

feature " "

distance based Methods

may overweigh attribute.

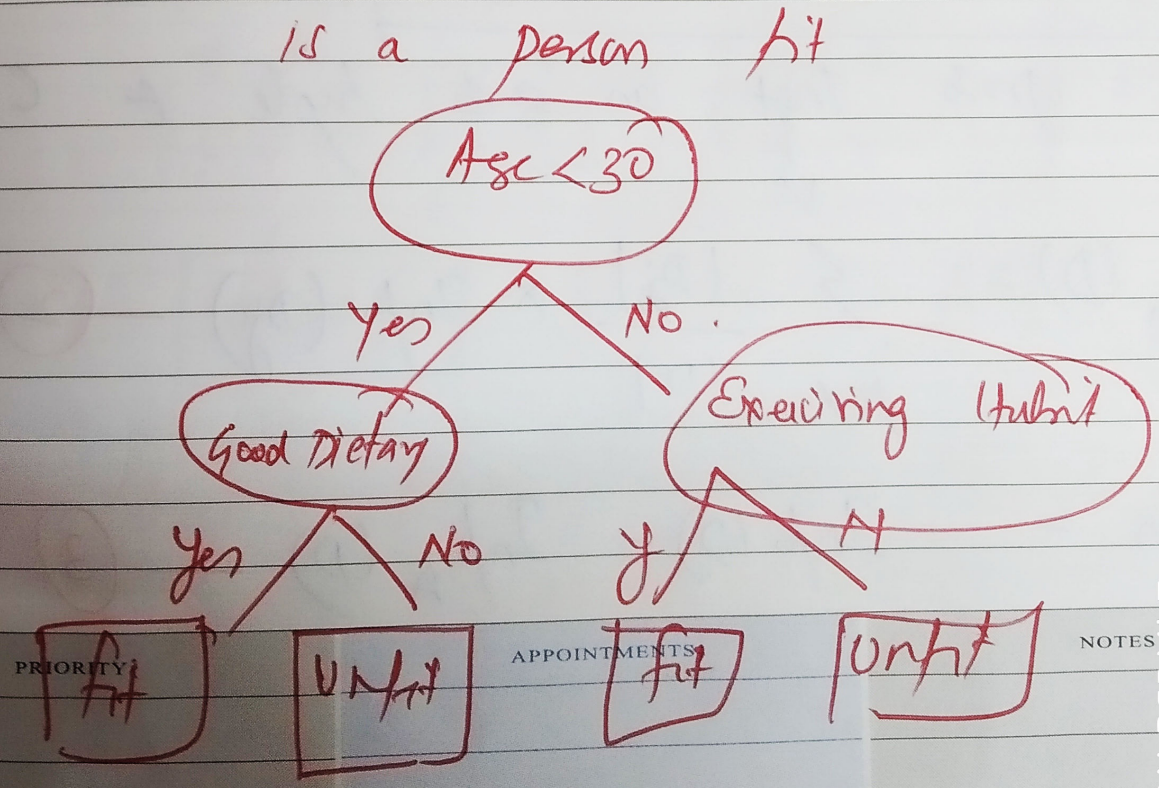
∴ Normalization within a range.

ics :-

- accuracy
- speed
- Robustness [in terms of noise]
- scalability
- Interpretability.

* Decision Tree Induction.

* learning of decision trees from xing data
* flow chart like structure ;
  * each internal node (Non leaf node) denotes an attribute or test ;
* Branch ⟶ test outcome
* leaf node (terminal node) → class label
* Topmost node → Root Node.



* ID3 (Iterative Dichotomizer)

* C4.5 → CART

* SLIQ, SC } Two famous DT Clf.

Attribute Selection Measures

Info gain — Highest Chosen

— minimizes info reqd to classify

$$Info(D) = -\sum_{i=1}^{M} p_i \log(p_i) \quad \text{— (1)}$$

p_i = prob that an arb tuple $\in C_i$

$$Info_A(D) = \sum_{j=1}^{V} \frac{|D_j|}{|D|} \times Info(D_j) \quad \text{— (2)}$$

$$Gain = Info(D) - Info_A(D) \quad \text{— (3)}$$

PRIORITY APPOINTMENTS NOTES

* How much wud be gained by branching on A.

Example → Pg 299.

5 No samples and 9 yes samples

$$\text{Info}(D) = \frac{-9}{14}\log\left(\frac{9}{14}\right) - \frac{5}{14}\left(\log\frac{9}{14}\right) = 0.94 \text{ bits}$$

No
3 ✓ 2
Youth = 5; mid = 4; S = 5.

N	Youth	MA		S	
3	2	0	4	2	3

$$\text{Info}_{Age}(D) = \frac{5}{14}\left\{\frac{-2}{5}\log\frac{2}{5} \; \frac{-3}{5}\log\frac{2}{5}\right\}$$

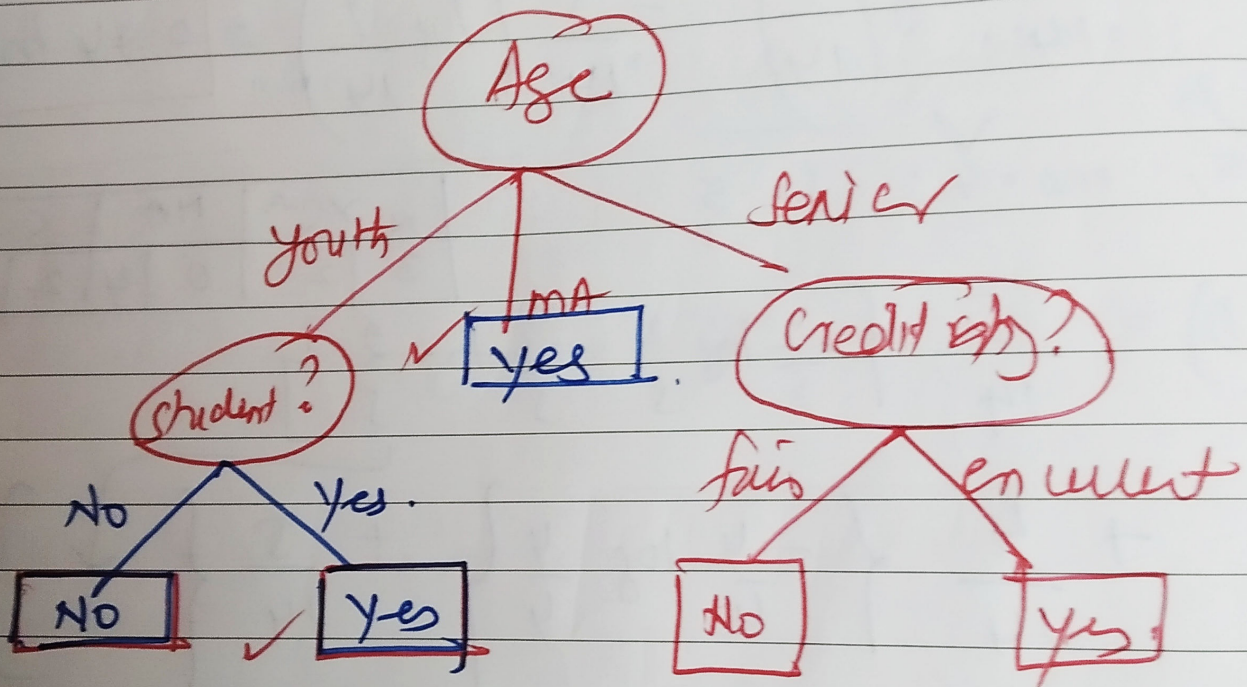
$$+ \frac{4}{14}\left\{-\frac{4}{4}\log\frac{4}{4}\right\} + \frac{5}{14}\left\{\rightarrow\right\}$$

$$= 0.694$$

$$\text{Gain}(Age) = 0.94 - 0.694 = 0.246$$

PRIORITY APPOINTMENTS NOTES

Similarly Gain (Age) = 0.246 (1)
(Income) = 0.029 (4)
(Student) = 0.151 (2)
(CR) = 0.048 (3)



Continuous Valued attributes:- Split Value

→ Sort A values in asc order

mid point b/w adj values = Sp.

a_i, a_{i+1} : $\left(\frac{a_i + a_{i+1}}{2}\right)$

January

DAY 25

measures are Given Below / Given Below

voting If termination node

is difficult to satisfy.

ning ——— Pre and post pruning

Refer a pruning example ppt

Bayesian classification

statistical classifiers.

predicts class membership probabilities.

Bayes Theorem → Basis.

↓

Naive Bayesian classifier

effect of an attrib value on C is independent of other attributes values.

turned on class conditional Independent

↓

dependence is acknowledged

Bayesian Belief Network

X → data tuple [evidence].

H → some hypothesis ⇒ X ∈ C.

for classification ~~P(e|~~ problems

P(H|X) → known is attribute distn (X)

test given

P(H|X) = posterior prob

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

prior prob ; any cust [buys/not]

PRIORITY | APPOINTMENTS | NOTES

$$P(C_i|x) = \frac{P(C_i) * P(x|C_i)}{P(x)}$$

$P(x)$ ↓ same for all classes

$\therefore P(C_i) * P(x|C_i)$ is to be maximized.

P(buys Comp = yes) = $\frac{9}{14}$ = 0.643

" No = $\frac{5}{14}$ = 0.357

P(age = youth | yes) = 2/9 = 0.222

" | No = 3/5 = 0.6 ✓ } * (1)

P(Income = medium | yes) = 4/9 = 0.444 }

" | No = 2/5 = 0.4 ✓ } * (2)

P(Student = yes | yes) = 6/9 = 0.667 }

| No = 1/5 = 0.200 ✓ } * (3)

$X \Rightarrow$ age = youth $\wedge$ income = medium $\wedge$ student = yes $\wedge$ credit rating = fair

$P(CR = \text{fair} \mid \text{yes}) = 6/9 = 0.667$ ✓

" No $= 2/5 = 0.4$ ✓ (9)

Comp au ✓ and ✓ prob.

$P(X \mid \text{Buys Comp} = \text{yes}) =$ au ✓ multiply.

$= .222 \times .444 \times .666) \times .667$

$= .044$ → (1)

$P(X \mid \text{Buys Comp} = \text{No}) =$ au ✓ ✓

$= .6 \times .4 \times 0.2 \times 0.4$

$= 0.019$ → (2)

$P(\text{Ci}) = P(\text{Buys} = \text{yes}) = 9/14 = .643$ → (3)

$5/14 = .357$ → (4)

∴ Rts of Mody ∴ $= .643 \times .044 = 0.028$ →

$.357 \times .019 = .007$

∴ Model predicts | buy comp = yes |
for the given X.

→

what if any one of yes / No probabilities = 0.

⇒ No age =

one zero prob might cancel all other attribute effect ⇒

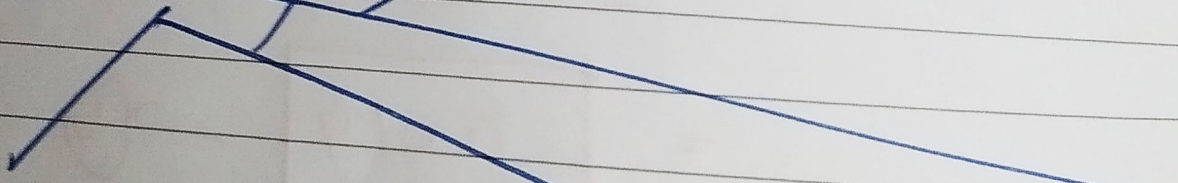
↓

Condition not to be allowed.

Laplacian Correction / Estimator

Ex:- 1000 tuples.

900 income = medium 10 income = high. 0 low.



PLANNER

add 1 more to each income-value pair

∴ add 3 more records of high/med/low

∴ New low prob = $\frac{1}{1003}$ and $\frac{11}{1003}$

∴ med = $\frac{991}{1003}$.

This is accepted soln in literature

Back Propagation classification

→ Neural N/W learning Algorithm

→ psychologists + Neurobiologists created a Computational Analogue of Neurons

→ NN is a set of i/c i/p - o/p units with each connection having a weight

→ learning phase N/W learns by adjusting weights to predict class labels for inputs

→ also referred to as Connectionist learning

→ require long training times

→ No of parameters reqd — empirically

→ Poor interpretability

→ High tolerance to noise

→ New pattern classification

→ Ideal when i/p – o/p relship attributes is not known

→ they are well suited for Continuous valued inputs and outputs;

Character Recognition, NLP, etc applns

Inherently parallel — suited for parallel processing

Backpropagation is one such NN Algorithm

+

2020 February MON 03

6th Week • 034-332

MultiLayer feed fwd Neural Network

→ 1 i/p layer; 1/more hidden layer, 1 o/p layer

+ i/ps to n/w ⇒ attributes of tuple

→ i/p's pass thru i/p layer → weighted and fed to second layer of Neuron

i/p units ⇒ Hidden layer

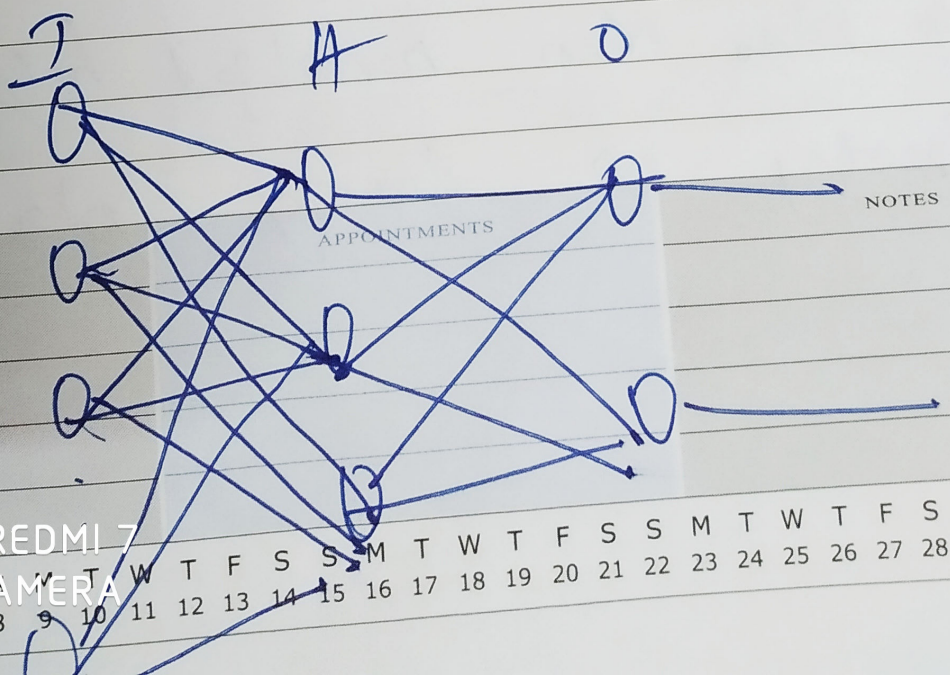
Comb[] iterative

last hidden layer o/p fed to o/p layer

that emits tuple's prediction class label

i/p layer units → i/p unit

#i/p " → Neurodes / op unit



PRIORITY APPOINTMENTS NOTES

SHOT ON REDMI 7
AI DUAL CAMERA

08.00 This is a 2 layer N/W [H + o/p]

09.00 → FF ∵ wts are not cycled back

10.00 to i/p or o/p unit of prev layer

11.00

→ FFN mostly treats as a Non linear

12 combination of inputs

13.00

(*) Network Topology :-

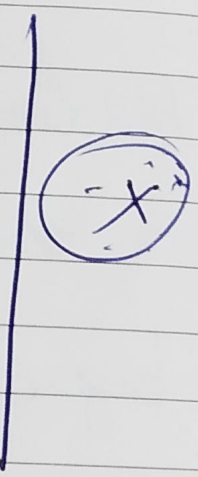
14.00

15.00 * No of units in ~~Hidden~~ i/p layer

16.00 * " " Hidden layers

17.00 * " "

* " " in o/p layer



18.00

→ i/p values when normalized helps in speeding up learning phase

→ Normalised to fall b/w (0 → 1)

Discrete valued attributes — encoded so that one unit for each domain value;

A has 3 possible values {a1, a2, a3}.

→ 3 units for A

I_0	I_1	I_2
1	0	0

⇒ A = a1

→ NN can be used for classfn / prediction

for classfn; one o/p unit for 2 classes { 1 = Y, 0 = N }

> 2 classes ⇒ one unit / class is used

→ No Best design; Trial + Error Process

→ Weights value Important accuracy

06 THU February 2020

→ accuracy not acceptable ⇒
learn again any new weights for
learn with new N/w topology

BACK-PROPAGATION :-

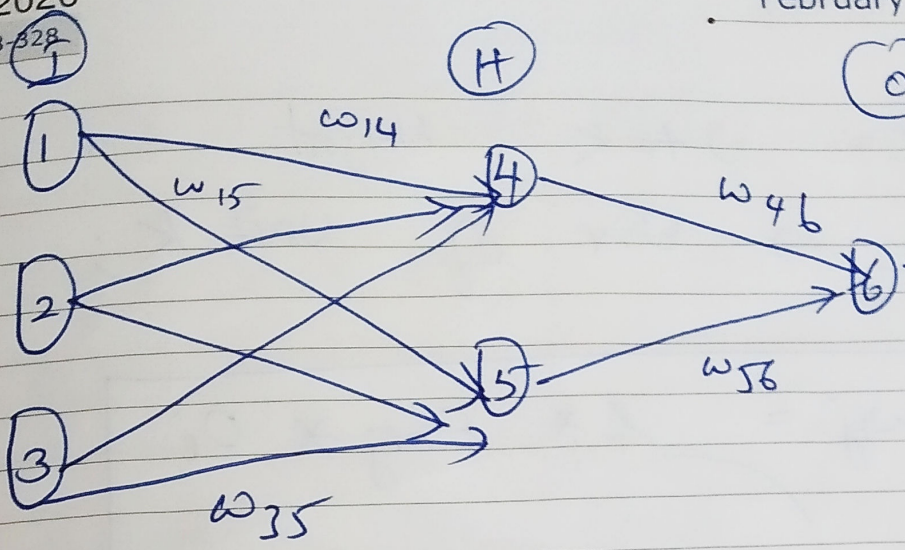
→ weights are adjusted to minimize the MSE b/w N/w's prediction — Actual Target Value

→ weight modifying in B/W flushing
o/p → hidden ---- to first layer

weights are initialized (-1 to 1) (-0.5 to 0.5)

PRIORITY | APPOINTMENTS | NOTES

W T F S S



Activation / o/p layer fn :-

$$O_j = \frac{1}{1+e^{-I_j}}$$

Squashing fn.
large i/p
to
small range 0 → 1

→ Non linear and Differentiable

$$Err_j \text{ (o/p layer)} :- \quad O_j(1-O_j)(T_j - O_j) \quad (1)$$

$$Err_j \text{ (Hidden)} :- \quad O_j(1-O_j)\sum_k Err_k w_{jk} \quad (2)$$

PRIORITY
APPOINTMENTS
NOTES

w_{jk} = j to k weight

Err_k → Err of unit k;

$$\Delta w_{ij} = l * Err_j * O_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

l → learning rate (0 to 1)

BPN uses Gradient Descent to adjust weights

l → avoids stuck @ local optimal.

learning rate = $1/t_s$ ⟶ No of iterations to

$$\Delta\theta_j = l * Err_j$$
$$\theta_j = \theta_j + \Delta\theta_j$$

one iteration = One epoch.

Training stops when;

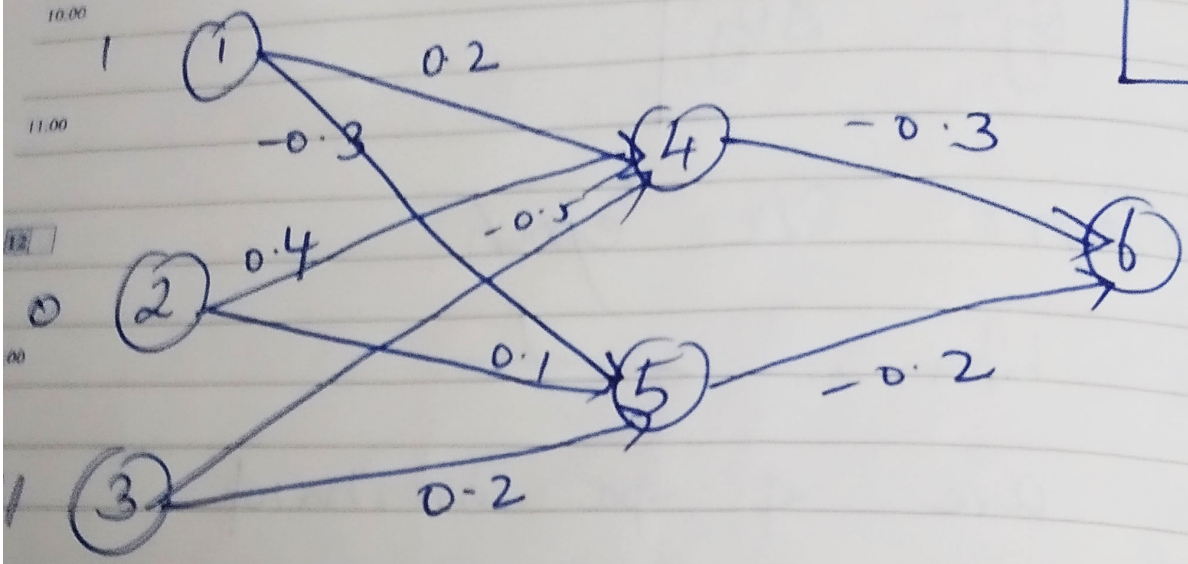
* all Δw_{ij} are < Spec. Predefined
* Percentage of misclassifications is < T.
* Pre specified no of epochs is reached

* Given D tuples, w weights; each epoch requires O(|D| × w) time.

Worst Case the # (epochs) can be exp.

Illustration.

$x = \{1, 0, 1\}$.



$\theta_4 = -0.4$
$\theta_5 = 0.2$
$\theta_6 = 0.1$

UNIT j	NET INPUT (I_j)	OUTPUT (O_j)
4	$1\times0.2 + 0 + 1\times-0.5 + -0.4 = -0.7$	$1/1+e^{-0.7} = .332$
5	$-0.3+0+0.2+0.2 = 0.1$	$= 0.525$
6	$-0.3(.332) - 0.2(.525) + 0.1 = -0.105$	$= 0.474$

(Target = 1)

2020 · February · WED 12

7th Week • 043-323

Calculation of Error @ each node :-

Unit j	Err j
6 (o/p) layer node	$.474 \times (1-.474)(1-.474) = 0.1311$
5	$.525\,(1-.525)\,(.1311)\,(-0.2) = -0.0065$
4	$.332\,(1-.332)\,(.1311)\,(-0.3) = -.0087$

Calculation of Weights + Bias Updates

$$w_{46}^{ij} = -0.3 + (0.9) \times (.1311) \times (.332) = -0.261$$

$$\theta_j = 0.1 + (0.9) \times (.1311) = 0.218$$

NOTES

APPOINTMENTS

February

13 T H U

* Teach GA classifier Quickly
2 to 3 classes max

* Performance Measure :-

✓ classification accuracy
↓
always Return Ratio of Cx,

✓ Sensitivity = $\frac{t\text{-}pos}{pos}$ → # of positive tp

✓ Specificity = $\frac{t\text{-}neg}{neg}$

✓ Precision = $\frac{t\text{-}pos}{t\text{-}pos + f\text{-}pos}$

accuracy = Sens. $\left(\frac{pos}{pos + neg}\right)$ + spec. $\left(\frac{neg}{pos + neg}\right)$

PRIORITY APPOINTMENTS NOTES