CW 15/2/24

Data Science

It is the art of ~~learnin~~ turning raw data into meaningful insights. To do that we will use various tools. Eg- It is in i) agriculture
ii) marketing
iii) disease diagonsis
iv) Drug designing

• It is in everywhere nowdays.

eg
Crop yield (what will be the crop yield)
{ soil condition, weather condi, irrigation etc

Tools/steps/processes of Data sci (It consists of many steps)
↳ mining

i) Data collection
ii) Store Data
iii) Process Data (Data imputation is a step of pre processing the Data)
(Data removal is under pre processing the Data)

Data normalization — we try to convert all values in the same range. As machine learning model can't predict properly if there is a huge difference in ranges. so that we need data normalization.
[It is a data transformation process that aligns data values to a common scale or distribution of values so that]

General Expression Data

	S1	S2	S3	S4	.....	S100
G1	1100	1200	10			
G2	0-1	0-1	0-1			
⋮						
G1000						

Predictive model

iv) Describing data
v) model designing

Catagory of machine learning model

i) unsupervised learning [150 → clusters] → clustering
ii) supervised learning
iii) Reinforcement learning

as similar as possible
this are different
and then as similar as possible

- iris data – setosa, versicolor, virginica
  50 rows, 50 rows, 50 rows (total 150 rows)
  Petal length, sepal length & width & width

Unsupervised learning – we don't know which obj falls into which catagory. eg – In above eg data there will be clusters of 3 types of flowers but we can't distinguish that which cluster is of which flower.

Supervised learning – Here we basically train our model, & if there is a unknown obj in the model it ~~th~~ can correctly predict ~~that~~ in what catagory it falls into.

Reinforcement learning – Semisupervised learning. Here step by step we try to improve our ml model to get accurate result. If the model isn't giving more accurate result then penalize it otherwise give it reward (save it).
Eg – self driving

Iris data – The iris data are a data frame of 150 measurements of iris petal & sepal length & width, with 50 measurements for each species of 'setosa', 'versicolor' & 'virginica'.

Unsupervised learning – It is a type of m.l that learns from data without human supervision. Unlike supervised learning, unsupervised m.l models are given unlabeled data & allowed to discover patterns & insights without any explicit guidance / instruction.

Supervised learning – It is a category of m.l that uses labeled datasets to train algorithms to predict outcomes & recognize patterns. Unlike unsupervised l., s.l. algos are given labeled training to learn the relationship b/w the i/p & o/ps.

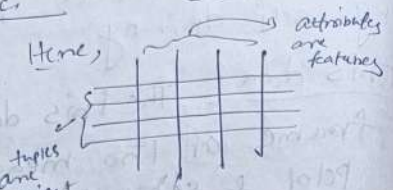
Reinforcement learning – It is a m.l. training method based on rewarding desired behaviours & punishing undesired ones. In general, a reinforcement learning agent – the entity being trained is able to perceive & interpret its environment, take actions & learn through trial & error.

Cla

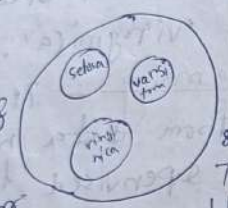
Data sci 20/2/24

1) Collect
2) Store
3) Process (Preprocessing)
4) Describe
5) model

Here,



1) supervised
2) unsupervised
   objective - To find & group the similar & dissimilar (similar group of datas is dissimilar from other groups) of datas



150 flowers split split
80 Training data set (120)
20 test d's (30)

[Ratio of splitting
70 - 30
Tr Test
80 - 20]

Classification algos - KNN, SVM, RF, DT, Logistic regression

21/2/24

We will train our model by using those classification algo & training data sets.

Then using the c. m ~~the~~ we will do show our model is working with the test data sets.

The c. algo will predic the species (class label) & we will compare the actual species with our predicted species. Based on that we'll judge how our classi. algo works (whether it is good for our model or not).

A classification algo is a supervised learning technique that uses training data to categorize new observations.

Regression - Here we try to predict numeric values instead of predecting class labels. It is a ml technique that identifies relationships b/w independent & dependent variables to predic an outcome. It inv training algo to identify patterns in the distribution of each data point.

Eg Boston housing ds

attributes

No. of rooms	length	bedrooms	width	Hight	Price

Price → target variable (y)

Independent variable (x) ← there is a relationship b/w this two → target variable (y)

We'll try to find out the mapping b/w y & x (or i/p variables)

$\boxed{y = f(x)}$

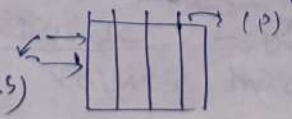
Our regression model will try to find out this particular func (i.e how these i/p variables varry the price) & when there will be a new house it will do the same. i.e it continiously ~~to~~ try to find out the numeric values.

Data Preprocessing (process of transforming raw data into a clean data set)

i) Remove noisy data (outliers) [eg. height 611, forget to give ., so then the height - 611 ft this is outliers]

ii) Data imputation (missing value imputation) [we try to estimate the values of the missing tuples, we can apply in various statistical computation]

iii) Dimentionality reduction (Feature extraction / Feature selection) (attributes)



When $P >> S$ the m.l model fails to give us the desired result. this problem is called curse of dimentionality. this can be solved by dimentionality reduction or F.E or F.S. [we will remove the very less imp attribute / feature] that will not give any impact to the target attributes. we will try to find out the less imp feature & will remove it, this is called feature selection (F.S). There are many algos for this. we'll try to approximate/ max no. of (the minimum features) attributes into n' no. of (another no. of features) attributes to get desired ~~attrib~~ result [5, 10, 000 ~ 100], this called feature execution (F.E). There is an algo for it that is called principal component analysis (PCA).

iv) Data transformation (data normalization)

In ds, diff attributes have diff ranges. If on this we apply the process it will not give us the desired result. So we have to take them all in same range.

It has 2 types - i) min-max

ii) Standard scaler (z-score)

i) try to transform the data in b/w (0,1)

ii) try to " in b/w (-1, 1)

i) min-max

x_i = original value

x_i' = updated value / transformed value

$$\boxed{x_i' = \frac{x_i - X_{min}}{X_{max} - X_{min}}}$$

[Eg - $\frac{F_i}{x_i}$, $x_i \leftarrow x_2$, $\vdots$, x_n } from here min value = X_{min}, max " = X_{max}]

ii) Standard scalar (z-score)

$$\boxed{x_i' = \frac{x_i - \mu}{\delta}}$$

[μ = mean, δ = Standard deviation]

formula for z-score normalization

we have to perform this step before proceeding to the other ~~sets~~ steps.

Types of (Data Attributes) Features

- Numeric (either int or floating point values)
  - Continuous
    - Internal scale
    - Ratio scale
  - Discrete
    - ordinal ~~original~~ attribute
    - Nominal attribute
- Categorical (it has certain string values with some contain range - Age (1-7, 7-20 ...), eye colour)
  - Coding scheme

Continuous - Height → 5.5, 5.51, 5.53 ... (Infinite no. of counts)

↓ range 5 - 6

the values are in a continuous range.

Discrete - whose values are only the finite & desirete no. of values in a range. Age → 1, 30, 2 ....

(finite no. of counts)

Interval - The intervals among the 2 attribute values of 2 diff. objects are meaningful.

Eg - ① temp (°C, °F) ; dates ② temp - 31°, 25°, 27° - - -

We can calculate the intervals b/w the temps. [Here ratio isn't meaningful]

Ratio - The - - - - - - - & the ratio is also meaningful. Eg - height, weight, growth of bacteria (growth is 1hr, growth in 2hr like that)

Nominal - A form of name / identification giving the info about it doesn't provide anything. Eg - ① I person name is Ram & another Sam. We can't say that Ram is better than Sam. ② Here we can't perform any m. & s. cal.

② EmpID
E0001
E0019 } Samely we can't say.

It is used just for distinguishing the attributes (obj)

Ordinal - Eg - ① Customer satisfaction

1
2
⋮
10 → Distinguishing as well as comparing

[If we say that 10 is heighest satisfaction, then 10 is far far better than 1]

② Grade
F
⋮
EX
O
A → Here, we know that O is better than C, EX is better than A

Here we can distinguish values & we can perform atleast subtraction (mathematical calculation)

Coding scheme - Instead of values we assign some ~~codeg~~ codes. Eg - Colour code.

Book → ① Data mining - By P. Tan
② Data mining - Han & Kamber (for pre processing mainly)

Clas ~~Fmt~~ ~~Gss~~ 5/8/24

Dissimilarity measures

$\boxed{\text{dissm} = 1 - \text{sim (similarity)}}$ → dissm. b/w 2 objs that has nominal attrib.

Dissimilarity for nominal attributes

$$\boxed{d(i,j) = \frac{p-m}{p}} \quad [m \in p]$$

i, j → (obj) both objs is & has p no. of objs

p → No. of ~~features~~ attributes

m = The no. attributes in which i & j matches in their nominal attribute values | No. of matches

Eg - ①

Object	test - 1 (nominal) (attribute/feature)
1	A
2	B
3	C
4	A

$d(i,i) = 0$ [∵ As $d(1,1) = 0$]

$d(1,2)$: $\frac{1-0}{1}$ → no match in test-1 values (not obj 1 & 2)

$$\begin{bmatrix} 0 & 1 & \# & \# \\ 1 & 0 & \# & \# \\ 1 & 1 & 0 & \# \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$d(1,1)$, $d(2,2)$, $d(3,3)$, $d(4,4)$

This not will be given as i/p during ML Proj

This 2 objs are same

$d(1,2) = \frac{1-0}{1}$

This 2 objs are to many dissimilar

→ (Symmetric matrix) It is dissimilarity matrix

→ [This is applicable for only nominal attribute]

Binary attribute – It is an attribute where there exist only 1 & 0. It is a nominal attribute that has only 2 states, such as 0 & 1 or true & false.

Dissimilarity b/w attri. when we are considering binary attri.

S-1	Gender (can be symmetric / Asymmetric)	Fever	Cough	test-1	test-2	test-3	test-4
name							
1 Jack	M	Y 1 1	N 0	P 1 1	N	N 0	N
2 Jim	M 1	Y 1 1	Y 1 1	N 0 0	N	N 0	N
3 mary	F 1	Y 1 1	N 0	P 1 1	N	P 1 1	N

let, Y = P = 1
N = N = 0

$$\begin{array}{c|c} i \backslash j \rightarrow & \\ 1 & \begin{bmatrix} a = 1 & b = 1 \\ c = 1 & d \end{bmatrix} \\ 0 & \end{array}$$

(fever & cough)

[a = i having 1 & j having 1
b = i " 1 & j " 0
c = i " 0 & j " 1
d = i " 0 & j " 0]

$$\boxed{d(i,j) = \frac{b+c}{a+b+c+d}} \longrightarrow \text{simple matching coefficient}$$

~~d(jack, jim) =~~

- $\boxed{\text{Jaccard Coefficient} = \frac{b+c}{a+b+c}}$

[d is not significant as d is i = 0, j = 0 so 0 is matched for 1 to 5 N/N value, that isn't significant]

$$d(\text{jack}, \text{jim}) = \frac{1+1}{1+1+1} = \frac{2}{3}$$ (dissim. b/w obj i & j)

$$d(\text{jack}, \text{mary}) = \frac{0+1}{2+0+1} = \frac{1}{3}$$

~~dis(~~
$$d(\text{jim}, \text{mary}) = \frac{1+2}{1+2+1} = \frac{3}{4}$$

7/3/24

Dissimilarity measures

Internal-scale attribute (''')

generalized distance measure,

minkowski

$x\ (x_1, x_2, x_3 \ldots x_p)$ — A_1, A_2, A_3 … A_p → attribute values [A_i]

$y\ (y_1, y_2, \ldots y_p)$ [Assume these are internal-scale attributes]

$$\boxed{d(x,y) = \sqrt[q]{|x_1 - y_1|^q + |x_2 - y_2|^q + |x_3 - y_3|^q + \ldots |x_p - y_p|^q}}$$

when, q = 1, [just the absolute diff. values]

$$\boxed{d(x,y) = |x_1 - y_1| + |x_2 - y_2| + \ldots |x_p - y_p|} \longrightarrow \text{manhattan distance}$$

when $q = 2$, $\longrightarrow$ Eucledian distance

$$\boxed{d(x,y) = \sqrt[2]{|x_1 - y_1|^2 + |x_2 - y_2|^2 + \cdots + |x_p - y_p|^2}}$$

values lies b/w

suppose, $A_1 \rightarrow 0-1$ (diff low) $\rightarrow$ not in same range, so that we need to apply normalization before proceeding

$A_2 \rightarrow 1-3$

$A_3 \rightarrow 0-10{,}000$ (diff high)

eg – lets, col 3 $\rightarrow$ 0–2 (diff. range) $\leftarrow$

500–3000 $\rightarrow$ 0–2500

$\therefore 2^2 + 2500^2 \rightarrow$ this particular value has no effect in calculating the proper distance. so we need normalization to put all the attributes in equal effect. if not, we will get wrong result & our ml. model/algo will fail.

ordinal attributes, (how to calculate the distance b/w 2 obj's for ordinal attributes)

$\{1, \ldots m_p\}$ $\rightarrow$ m = max rank

$$\boxed{r_i^{(f)} = \frac{r_i^{(f)} - 1}{m_i^{(f)} - 1}}$$

(Particular feature of a property)

f (feature no.)

eg lets

A5 Day

Mon $\rightarrow 1 (r_x^{(5)})$

wed $\rightarrow 3 (r_y^{(5)})$

sun $\rightarrow 7 (r_z^{(5)})$

x, y, z — row no. of obj

$r_x^5 = \frac{1-1}{7-1} = 0$, $r_y^5 = \frac{3-1}{7-1} = \frac{2}{6} = \frac{1}{3}$

$r_z^5 = \frac{7-1}{7-1} = 1$

$d(x,z) = |0-1| = 1$ [After calculating $r_i^{(f)}$, we will apply the generalized eqn] (interval-scale formula) $\rightarrow$ [input not $|x_i - y_i|$ only $|x - y|$]

Ratio scale

It expressed by $= \frac{a}{b}$ like that (or) in terms of A & B

ratio $= \frac{A}{B}$ like that

gene expression
G_1 5 $\rightarrow$ state 1
G_2 2 $\rightarrow$ state 2

- If the change in gene expression ≥ 2 then, it is meaningful.

$\rightarrow$ Here we have to transform the ratio scale attri. into interval scale attri. by 'log' transformation

$$\boxed{\log\left(\frac{a}{b}\right) = \log a - \log b}$$

- All dataset contains this 5 attributes.

Distance b/w a pair of obj whenever we have all those 5 types of attri. In our data sets (mixed dissimilarity)

Variables of mixed types

no. of features	A_1	A_2	A_3	—	A_p
no. of objects	x_i	$x_i^{(A_1)}$	$x_i^{(A_2)}$	$x_i^{(A_3)}$	—	$x_i^{(A_p)}$
	x_j	$x_j^{(A_1)}$	$x_j^{(A_2)}$	$x_j^{(A_3)}$	—	$x_j^{(A_p)}$

$$\boxed{d(x_i, x_j) = \frac{d_{A_1}(x_i, x_j) + d_{A_2}(x_i, x_j) + \cdots d_{A_p}(x_i, x_j)}{p \text{ (no. of attributes)}}}$$

Simplified formula (1),

$$d(x_i, x_j) = \frac{\sum_{k=1}^{p} d_{A_k}(x_i, x_j)}{p}$$

suppose we wish to assign weight, that is also applicable here,

$$d(x_i, x_j) = \frac{w_1 * d_{A_1}(x_i, x_j) + \dots w_p * d_{A_p}(x_i, x_j)}{w_1 + w_2 + \dots + w_p}$$

$$= \frac{\sum_{k=1}^{p} w_k \cdot d_{A_k}(x_i, x_j)}{\sum_{k=1}^{p} w_k}$$

weighted dissimilarity b/w a pair of objs b/w all different types of attributes.

mixed

- when $p >> s$ → no. of samples → m.l. algo will fail to give us the correct result, if this case arrives. This problem is called
  - p → no of features

To solve the problem we need ← curse of dimensionality

data reduction, one of such data reduction principle is principal component analysis (PCA)

p ← dim dataset (features)

d ← no. of samples

$d << p$

|| pow(same)

func → reduce dimentionality of a ds
- → Inherent obj
- → Relationship b/w attribute

etc.

Dim = no. of attri.

PCA

Steps

1. subtract the mean from each data dim. & find the mean adjusted data.

Suppose

2d			1d
x	y	trans-form →	$x - \mu_x$	$y - \mu_y$
x_1	y_1		$x_1 - \mu_x$	$y_1 - \mu_y$
x_2	y_2	original data	$x_2 - \mu_x$	$y_2 - \mu_y$
x_n	y_n		$x_n - \mu_x$	$y_n - \mu_y$

mean(x) substraction

mean adjusted data

2. Compute the covariance matrix 'A'.

(this is a type of mat that is used to represent the cov values b/w pairs of elems gives in a random vector)

covariance mat

p - dim

(no of features)

2d data

$$A = \begin{pmatrix} var(x) & cov(x,y) \\ cov(y,x) & var(y) \end{pmatrix}$$

$cov(x,y) = cov(y,x)$ always

$$var(x) = \frac{\sum (x_i - \mu_x)^2}{n-1}$$

for sample data

$$cov(x,y) = \frac{\sum (x_i - \mu_x)(y_i - \mu_y)}{n-1}$$

3. i) Calculate the eigen values & eigen vectors of the covariance matrix 'A'.

↳ It is applicable for square mat.

If n no. of eigen values & vectors are given, then n = dim of data set

ii) $A X = \lambda X$

$(A - \lambda) X = 0$

if consider,

$A - \lambda = 0$, we will get diff λ values

$\lambda_1, \lambda_2 \dots \lambda_p$

As the data is 2d we will get 2 eigen values — λ_1, λ_2

4. Arrange the eigen values in descending order from largest to smallest.

5. From eigen values we will find out the eigen vectors.

$AX = \lambda X$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \leftarrow (\lambda_1) \text{ eigen vector}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \leftarrow (\lambda_2) \text{ eigen vector}$$

As we have 2 eigen vals λ_1, λ_2 we have to solve this eqn for 2 times.

$AX = \lambda_1 X$ — (i)

$AX = \lambda_2 X$ — (ii)

we have to find out the value of x & y by using (i) & (ii)

$A = \begin{pmatrix} x \\ y \end{pmatrix}$ - 3 (n) mat.

6. We will transform the eigen vectors into unit length eigen vectors.

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} \frac{u_1}{\sqrt{u_1^2+u_2^2}} \\ \frac{u_2}{\sqrt{u_1^2+u_2^2}} \end{pmatrix}$$

7. derive the new dataset

$$\boxed{\text{new dataset} = \text{Row Feature vector} * \text{Row Data Adjacent}}$$
(dim − 2×n)

Feature vector = $\begin{pmatrix} u_1' & v_1' \\ u_2' & v_2' \\ \vdots & \vdots \\ u_p' & v_p' \end{pmatrix}$ (p×p − dim²) [p eigen values, p eigen vectors]

$\boxed{\text{fm } 2d, \begin{pmatrix} u_1' & v_1' \\ u_2' & v_2' \end{pmatrix}}$

Row Feature vector (p×p) = (Feature vector) T → transpose (p×p)

Row Data Adj = (mean Adj Data) T → transpose

2×n n×2

$\boxed{\begin{array}{l} \text{fm } 2d,\ 2\times 2 \\ \text{R.F.V.} = 2\times 2 \ \text{(Feature vector)} \\ \text{R.D.A} = 2\times 2 \end{array}}$

Eigen value of a matrix – It is a special set of scalar values that is associated with the set of linear eqns most probably in the matrix eqns.

Eigen vector of a matrix – It is a vector that is associated with a set of linear eqns. These are also known as latent vectors, proper vectors or characteristic vectors. (1)

Cw — PCA (Dimentionality reduction technique) — 14/3/24

let's we have 2D data & have to reduce the data into 1D data. ← this is sample dataset

x	y	$x-\bar{x}$	$y-\bar{y}$
2.5	2.4	0.69	0.49
0.5	0.7	−1.31	−1.21
2.2	2.9	0.39	0.99
1.9	2.2	0.09	0.29
3.1	3	1.29	1.09
2.3	2.7	0.49	0.79
2	1.6	0.19	−0.31
1	1.1	−0.81	−0.81
1.5	1.6	−0.31	−0.31
1.1	0.9	−0.71	−1.01

$\bar{x} = \frac{18.1}{10}$, $\bar{y} = \frac{19.1}{10}$ → mean adjusted data (1)

$\therefore \mu_x = \frac{18.1}{10} = 1.81$ $\mu_y = \frac{19.1}{10} = 1.91$

(2) A = covariance matrix $\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix} = \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.716 \end{pmatrix}$

$var(x) = \frac{\sum_i (x_i-\bar{x})^2}{n-1} = \frac{5.549}{9} = 0.6170$ $cov(x,y) = \frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{n-1} = \frac{5.539}{9} = 0.61544$

same

$var(y) = \frac{\sum_i (y_i-\bar{y})^2}{n-1} = \frac{6.449}{9} = 0.716$ $cov(y,x) = 0.7166$

(3) $Ax = \lambda x$

$\Rightarrow (A - \lambda I)x = 0$ $\quad$ I (Identity mat) $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $\quad$ [; value of $I = 1$]

$\Rightarrow \left\{ \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.716 \end{pmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\} x = 0$

$\Rightarrow \left\{ \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.716 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right\} x = 0$

$\Rightarrow \begin{pmatrix} 0.617 - \lambda & 0.615 \\ 0.615 & 0.716 - \lambda \end{pmatrix} x = 0$

to become this eqn. 0, $(A - \lambda)$ held to be 0

$\therefore \begin{pmatrix} 0.617 - \lambda & 0.615 \\ 0.615 & 0.716 - \lambda \end{pmatrix} = 0$

$\Rightarrow \begin{vmatrix} 0.617 - \lambda & 0.615 \\ 0.615 & 0.716 - \lambda \end{vmatrix} = 0$

$\Rightarrow (0.617 - \lambda)(0.716 - \lambda) - (0.615)^2 = 0$

$\Rightarrow 0.617 * 0.716 - 0.617\lambda - 0.716\lambda + \lambda^2 - 0.378 = 0$

$\Rightarrow 0.442 - 1.333\lambda + \lambda^2 - 0.378 = 0$

$\Rightarrow \lambda^2 - 1.33\lambda + 0.064 = 0$

$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ $\quad$ $\frac{1.33 \pm \sqrt{1.33^2 - 4 \cdot 1 \cdot 0.064}}{2 * 1}$

$= \frac{1.33 \pm \sqrt{1.5129}}{2}$

$= \frac{1.33 \pm 1.23}{2}$; $= 0.049, 1.283$

(4) $\therefore \lambda_1 = 1.28 \approx 1.283$, $\lambda_2 = 0.05 \approx 0.049$ (as we have to ~~sh~~ sort λ according to ~~the~~ the descending order $\therefore \lambda_1$ = highest, λ_2 = lowest like that)

(5) $Ax = \lambda x$ $\quad$ $\left[\because x = \begin{pmatrix} x \\ y \end{pmatrix}\right]$

$Ax = \lambda_1 x$

$\Rightarrow \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.716 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1.283 \begin{pmatrix} x \\ y \end{pmatrix}$

$\Rightarrow \begin{pmatrix} 0.617x + 0.615y \\ 0.615x + 0.716y \end{pmatrix} = \begin{pmatrix} 1.283x \\ 1.283y \end{pmatrix}$ $\Rightarrow$ can choose any 1 of them, will the same ans.

$\therefore 0.617x + 0.615y = 1.283x$

$\Rightarrow$ ~~0.617x~~ $0.615y = 1.283x - 0.617x$

$\Rightarrow 0.615y = 0.666x$

$\Rightarrow y = 1.083x$; $\Rightarrow x = 0.923y$

~~$0.615x + 0.716y = 1.283y$ — (1)~~

~~$\Rightarrow 0.615x + 0.716 * 1.083x = 1.283 * 1.083x$~~

~~$\Rightarrow 0.615x + 0.$~~

from eqn (1)

If, ~~$y = 1$~~ $x = 1$, $y = 1.083$

eigen vector corresponding to λ_1

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1.084 \end{pmatrix}$$ ← (unit length eigen vectors)

$$(6) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{1^2+1.084^2}} \\ \frac{1.084}{\sqrt{1^2+1.084^2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2.175} \\ \frac{1.084}{2.175} \end{pmatrix}$$

$$= \begin{pmatrix} 0.678 \\ 0.735 \end{pmatrix}$$

$$\boxed{\begin{array}{l} 0.615x + 0.716y = 1.283y \\ \Rightarrow 0.615x = \end{array}}$$

(5) $Ax = \lambda x$

$\Rightarrow Ax = \lambda_2 x$

$$\Rightarrow \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.716 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0.049 \begin{pmatrix} x \\ y \end{pmatrix}$$

$\Rightarrow 0.617x + 0.615y = 0.049x$

$\Rightarrow 0.567x = -0.615y$

$\Rightarrow y = -0.922x$ — (11)

If consider, $x = -1$, $y = 0.922$

Again, from eqn (11)

Eigen vector corresponding to λ_2

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0.922 \end{pmatrix}$$ ← (unit length vector)

$$(6) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \frac{-1}{\sqrt{(-1)^2+0.922^2}} \\ \frac{0.922}{\sqrt{(-1)^2+0.922^2}} \end{pmatrix} = \begin{pmatrix} -0.735 \\ 0.678 \end{pmatrix}$$

7) Transformed data on new dataset = Row feature vector * Row data adjacent

$$\begin{pmatrix} \text{Feature} \\ \text{vector} \end{pmatrix}^T \quad \begin{pmatrix} \text{mean adj dataset} \end{pmatrix}^T$$

$$\text{Feature vector} = \begin{pmatrix} u_1 & v_1 \\ u_2 & v_2 \end{pmatrix}$$

$$\text{Row feature vector} = \begin{pmatrix} 0.678 & -0.735 \\ 0.735 & 0.678 \end{pmatrix}^T = \begin{pmatrix} 0.678 & 0.735 \\ -0.735 & 0.678 \end{pmatrix}$$

$$\begin{pmatrix} u_1 & v_1 \\ u_2 & v_2 \end{pmatrix} = \begin{pmatrix} 0.678 & -0.735 \\ 0.735 & 0.678 \end{pmatrix}$$

This to eigen vectors are perpendicular or orthogonal to each other

$$\text{Mean adj data} = \begin{pmatrix} 0.69 & 0.49 \\ -1.31 & -1.21 \\ \vdots & \vdots \\ 0.71 & -1.01 \end{pmatrix}$$

$$\therefore \text{Row data adj} = \begin{pmatrix} 0.69 & -1.31 & 0.39 & 0.09 & 1.29 & \cdots & -0.71 \\ 0.49 & -1.21 & 0.99 & 0.29 & 1.09 & \cdots & -1.01 \end{pmatrix}$$

new dataset

Transform data = $\begin{pmatrix} 0.678 & 0.735 \\ -0.735 & 0.678 \end{pmatrix}$ *
(2×10) (2×2)

$$\begin{pmatrix} 0.69 & -1.31 & 0.39 & 0.09 & 1.29 & 0.49 & 0.19 & -0.81 & 0.31 & -0.71 \\ 0.49 & -1.21 & 0.99 & 0.29 & 1.09 & 0.79 & -0.31 & -0.81 & -0.31 & -1.01 \end{pmatrix}$$
(2×10)

$$\begin{pmatrix} 0.828 & -1.778 & 0.099 & 0.274 & 1.676 & 0.913 & -0.099 & -1.144 & -0.438 \\ -0.175 & 0.1429 & 0.3844 & -0.130 & -8.742 & 0.175 & 0.349 & 0.046 & 0.018 \\ & & & & & & & -1.223 & \\ & & & & & & & -0.163 & \end{pmatrix}$$

1st PC, 2nd PC

If we which to reduce the original dim 2 to 1, then we have to take only the 1st row.

1st row is known as 1st principle Component

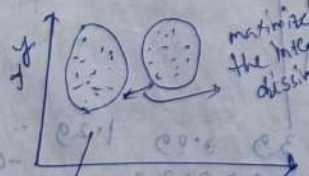
2nd " " " 2nd principle component

∴ Reduced data set = only the 1st row (1st PC)
(2×10)

$$= \begin{pmatrix} 0.828 & -1.778 & 0.099 & \cdots & -1.223 \\ \cancel{-0.175} & \cancel{0.1429} & \cancel{0.3844} & \cancel{\cdots} & \cancel{-0.163} \end{pmatrix}$$

ML 10/3/24

Clustering ← unsupervised learning



maximize the inter clusters dissimilarity

groups → clusters → modules

Obj: To find a set of group of objs, So that the objs in a cluster are similar amongst the same obj

Obj of ~~clustering~~ clustering

(minimize the intra-clusters distance / dissimilarity)

as possible & obs belonging to other clusters are as dissimilar as possible.

Clustering algos

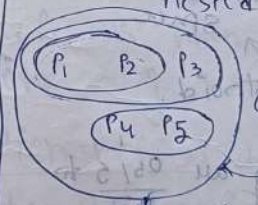
1) Partitional Clustering
2) Hierarchical "
3) Fuzzy Clustering
4) Density-based "

1) Partition C. - All objs are part of 1 cluster & all ~~obs~~ aren't non-overlapping clusters

→ K-means ~~Partition~~, K-medoid, Partition Around Medoid (PAM), C means

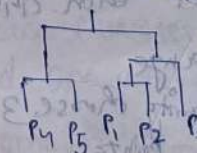
2) Hierarchical C. - It tries to generate a tree like structure. It has 2 catagories.

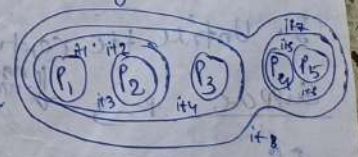
nested Clustering



Consider another obj then (All the obs are part of a single cluster)

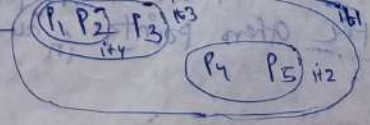
Its graphical representation is called dendogram



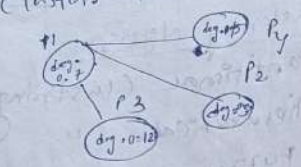


Agglomerative H.C →

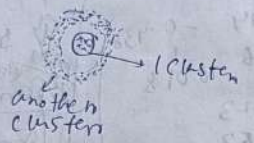
Divisive H.C →



3) Fuzzy C. - Each obj's are belonging to a cluster with a certain degree



4) Density based C. -



Helps us to identify the denser obj's in a cluster or can say denser clusters.

no. of centroids(k) = no. of clusters

K-means Clustering
Algo → no-of-clusters

central point in multidimensional space

1. select K points/ of/as initial centroid. /objects
2. Repeat
3. Form K-clusters by assigning all obj's to the closest centroid points
4. Recompute the centroid of each cluster
5. Untill the centroids don't change

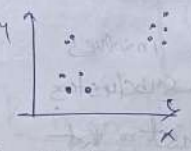
Suppose K=3 → Arbitrarily choose 3 points

[from all the centroids the distances are calculated to the other points in a cluster] ||

K depends of 2 no of obj's & no of centroid

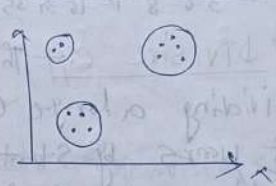
Clustering - It is a data set technique in m.l. that groups similar rows in a data set. After running a clustering technique a new column appears in the data set to indicate that the group each row of data fits into best. ||

original data



clustering

clustered data

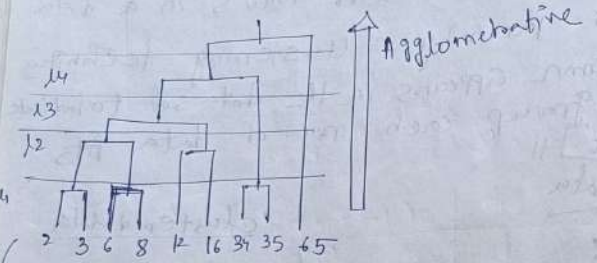


Partition clustering - [These methods partition the objects into K clusters & each partition forms 1 cluster. This method is used to optimize an objective criterion similarity func. such as when the distance is a major parameter eg K-means, CLARANS etc.] ||

Hierarchical Clu. - [It creates a tree of clusters. Hierarchical C. not surprisingly, is well suited to hierarchical data, such as taxonomies. It is new clusters are formed using the previously formed one. It is divided into 2 Catagories] ||

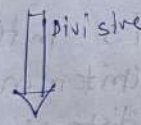
i) Agglomerative C. - Also known as AGNES (Agglomerative nesting), this is the most common type of hierarchical clustering. It groups objects into clusters based on their similarity. The algo starts by treating each obj as a single

-tion Cluster.]



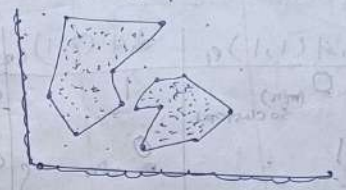
(ii) Divisive C:- This type of C. algo involves dividing a large cluster into smaller subclusters. It works by starting with a single cluster that contains all of the data points & then iteratively splitting the c. into 2.]

(+) same juse



Fuzzy C. - [It is a type of clustering algo in ml that allows a data point to belong to more than 1 cluster with different degrees of membership. Unlike traditional clustering algo, such as k-means/hierarchical clustering which assign each data point to single cluster, fuzzy clustering assigns a membership degree b/w 0 & 1 for each data point for each C.]

Density-based C. - [It considers the clusters as the dense region having some similarities & differences from the lower dense region of the space. These methods have good accuracy & the ability to merge 2 clusters. Eg DBSCAN (Density Based Spatial Clustering of Application with Noise), etc. OPTICS



]

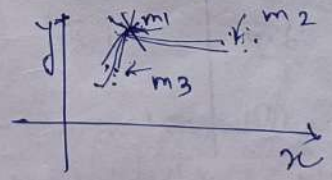
K-means clustering - [It is an unsupervised m.l. algo that divided data points into clusters based on their similarities. It iteratively assigns data points to the nearest cluster centroids & optimizes the centroids' positions.]

28/3/24

Clustering

K-means → objective is to find k no. of clusters / obj's. It will try to identify the groups of obj's

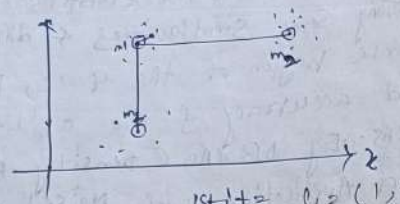
if k=3 → it randomly considers 3 obj's in the centroid of the cluster.



say (m_1, m_1)	distances $(m_1 - m_2)$	$(m_1 - m_3)$
2.5	15.7	25.9

↳ we'll assign this distance to the obj whose centroid is in m1.

↓ fresh picture



1st it = $c_1 = (1,1)$, $c_2 = (2,1)$

Eg-1

1st it

no. of groups/clus. $k = 2$

x	y	dist (1,1) c_1	dist (2,1) c_2	cl
1	1	0 (min) so clusters	1	cl1
2	1	1	0 min	cl2
4	3	$\sqrt{13}$	$\sqrt{8}$	cl2
5	4	5	$\sqrt{18}$	cl2

$\sqrt{(1-1)^2 + (1-1)^2}$

$\sqrt{4^2 + 3^2}$

$\sqrt{5^2 + 2^2}$

$\sqrt{(4-1)^2 + (3-1)^2}$

As there is only 1 point in cluster 1, so the centroid of this particular obj remains same (1,1). [for cluster 1] $c_1 = (1,1)$

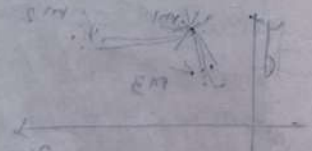
for cluster 2, there are 3 centroids so centroid of c12

2 1
4 3
5 4

$c_2 = (3.67, 2.67)$

mean $= \frac{11}{3}, \frac{8}{3}$

$= 3.67, 2.67$



2nd it

2nd it $c_1 = (1,1)$ $c_2 = (3.67, 2.67)$

x	y	c_1 dist (1,1)	c_2 dist (3.67, 2.67)	
1	1	0	3.15	cl1
2	1	1	2.36	cl1
4	3	$\sqrt{13}$	0.47	cl2
5	4	5	1.89	cl2

$\therefore c_1 = (1.5, 1)$

$c_2 = (4.5, 3.5)$ $c_1 = (1.5, 1)$ $c_2 = (3.67, 2.67)$

3rd it

x	y	c_1 dist (1.5, 1)	c_2 dist (4.5, 3.5)	
1	1	0.5	4.3	cl1
2	1	0.5	3.5	cl1
4	3	3.2	0.71	cl2
5	4	4.6	0.71	cl2

the centroids doesn't change, so no need to recect. So finally, there are 2 clusters 2 in cl1 & 2 in cl2.

Sum of square error (SSE) → Intra-cluster distance

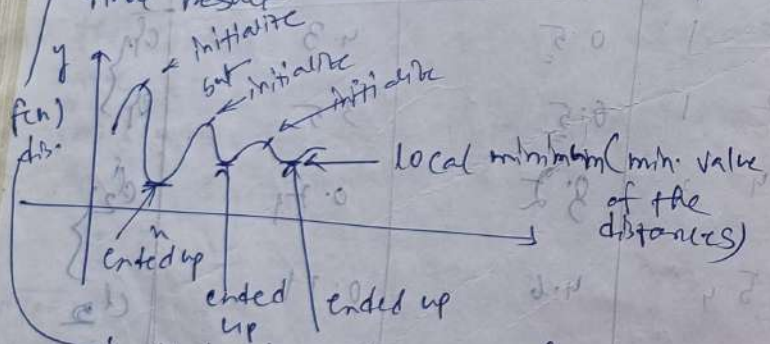
$$\boxed{SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d^2(x, m_i)}$$ → centroid for cluster i

or J

calculate dis. b/w each obj from a particular centroid & square it & sum it up & ~~cluster~~ this process will be followed for all of the clusters.

• K-means try to minimize this SSE.

As there are many datas, so if we vary ~~clust~~ centroids the result will also vary. so we have perform K means considering diff. centroids (choose of centroids) & the results depends upon that. The result that will give the min dis. will be the final solving method/ final result.



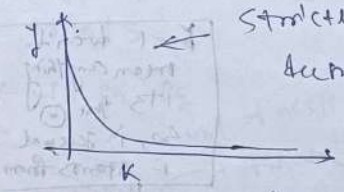
→ limitation of K-means (Once repeated running & choosing the best soln)

② K means only works for numerical datas / attributes.

Remedy K-medoid, K-mode

③ under limitations

If ↑K then J↓, while K=n then J=0



Strictly monotonic decreasing property

~~If~~ whenever J=0 for whatever the K is we can't say it's the best soln, so we must have to check that is J=0 for other centroids? In order to decide what is the best value we run the K-means (i.e. with diff value of K for ~~diff times~~ same times.

K=2 100 → min SSE

K=3 100 → min SSE

..... like that

K=10, K=100 → always find that min SSE is always the lowest one

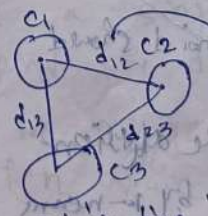
Remedy

Xie - Beni Index

$$\boxed{XB = \frac{\sum_{i=1}^{K} \sum_{x \in C_K} d^2(x, m_i)}{\max_{i \neq j}\left(n * \min(m_i, m_j)\right)}}$$

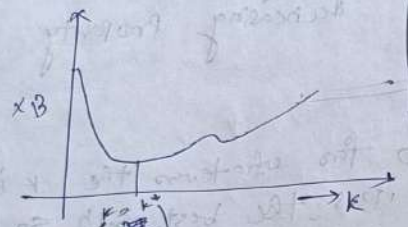
[m_i = min ~~intera~~ intra cluster distance]

→ min inter cluster distance among all pairs of clusters



Our objective is to min ~~XB~~ & the numerators ~~& n~~ (min the interclusters distance) &

maximize the inter cluster distance (Dino-meneton)



K=1 doesn't mean anything its just ⊙
So, in general K starts from 2
$K=\sqrt{n}$

val of xB is min here (choose that particular val of K to get the optimal no. of clusters)

best value of K in sol^n.

Under limitations

(a) Noise from the attribute values
those attribute values are diff. the very other attribute values.

(1,1) (2,3) (2,1), (3,4) (101,110)

the centroid is going to be affected by this noisy point

due to this point the centroid has been changed

In general, the centroid should lay here.

Remedy - we try to discard the outlier (noisy data) before solving it by K-means

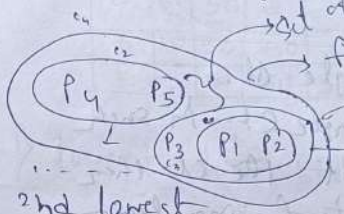
Ch 9

Hierarchical clustering

16/4/24

we use it to find the similarity & dissimilarity b/w objects.

P1 P2 P3 P4 P5



→ set of nested clusters
→ final cluster
→ Dendogram
→ dissim among P1 & P2 is 1st lowest

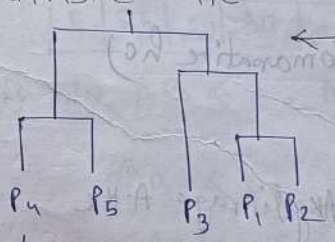
2nd lowest

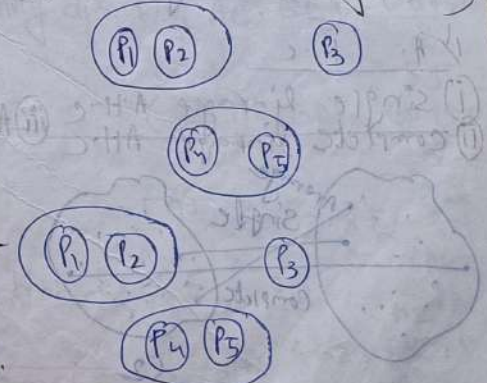
[C1 = nested C. inside the C3 C.
C3 = ... C4 ...]

2 types of hierarchical clustering algo -

1) Agglomerative HC
2) Divisive HC

← Dendogram drawing (representation)



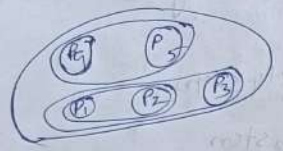


(not that, pto)

1) Agglomerative HC -

1) Considering each Cl. a single Cl.
2) Find the dis. b/w them.
3) we merge the Cl. having the lowest dis.
4) Then again find the dis. b/w new created merged Cl. (Step ① from) 5) Continue till getting a single cluster.

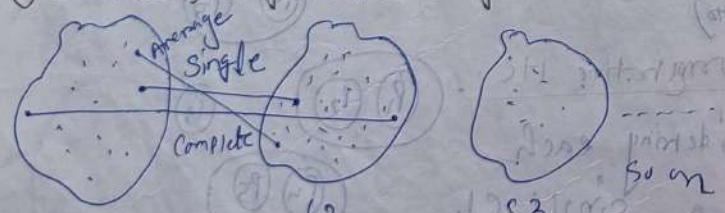
2) Divisive HC (not ~~HAC~~, PTO)



1) consider all cl. a single cl.
2) Try to split the single cl. in such a way that if we split the cl. then we'll obtain 2 cl., ~~it~~ having least dissimilarity.
3) In next iteration, step 2 will be done (we'll get 2 seperate obj) & so on until we seperate all the ~~seperate~~ objs belonging a same cl.
(Reverse way of aggiomarative hc)

1) A. H.C

(i) Single linrage A.H.C (iii) Avg linrage A.H.C
(ii) ~~complete linrage A.H.C~~



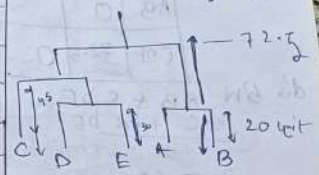
C1 — consider two (for time being)

Eg) Avg A.H.C

1) Dis mat

	A	B	C	D	E
A	0				
B	20	0			
C	60	50	0		
D	100	90	40	0	
E	90	80	50	30	0

Final dendogram =



1) lowest dis. b/w 2 objs = 20 (dis. b/w A & B)

⇓

	AB	C	D	E
AB	0			
C	55	0		
D	95	40	0	
E	85	50	30	0

dis b/w AB & C = Avg dis. b/w (AC, BC) = $\frac{60+50}{2}$ = 55

AB & D = (AD, BD) = $\frac{100+90}{2}$ = 95

AB & E = (AE, BE) = 85

2) lowest dis = dis b/w D & E = 30

⇓

	AB	C	DE
AB	0		
C	55	0	
DE	100	45	0

dis b/w AB & DE = Avg dis (AD, AE, BD, BE)
= $\frac{100+95+90+85}{2}$ = 90

dis b/w C & DE = Avg dis (CD, CE) = $\frac{40+50}{2}$ = 45

lowest dis = 45

3)

	AB	CDE
AB	0	
CDE	72.5	0

⟹

	ABCDE
ABCDE	0

↓ no need

dis b/w AB & CDE =

$$\frac{ABC + ABDE}{2}$$

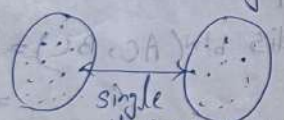
$$= \frac{55 + 90}{2}$$

$$= \frac{145}{2}$$

$$= 72.5 \text{ (lowest dis)}$$

① Single linkage agglomerative H.C

It's a bottom-up method that groups clusters by iteratively merging the 2 clusters with the closest pair of elements/objs that aren't already in the same cluster.

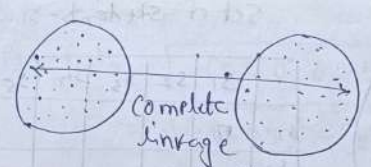


single linkage A.H.C

② Complete linkage agglomerative H.C

[It is a method of a.h.c. that starts with each element in its own cluster, & then merges clusters into larger clusters until all elements are in the same cluster.]

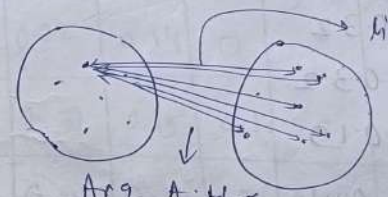
In complete linkage, we define the distance b/w 2 clusters to be the maximum distance b/w any single data point in the first cluster & any single data point in the 2nd cluster.



Complete linkage A.H.C

③ Average linkage Agglomerative H.C.

In Avg linkage a.h.c, the distance b/w 2 clusters is defined as the average distance b/w all data points in the first cluster & all datapoints in the 2nd cluster.



Avg A.H.C

(2)

18/4/24

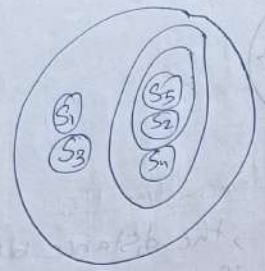
Clw pg23

Agglomerative H.C. Algo

Steps: A.H.C ∝ single linkage A.H.C

1) Compute the distance matrix
2) Let each data point be a cluster
3) Repeat
4) merge the 2 closests clusters
5) update the distance matrix
6) until only a single cluster remaining is obtained.

Set of students - $S_1 \ldots S_5$



	S_1	S_2	S_3	S_4	S_5
S_1	0				
S_2		0			
S_3			0		
S_4				0	
S_5					0

SN	x	y
P1	0.40	0.53
P2	0.22	0.32
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

$$d(P_1, P_2) = \sqrt{(0.40-0.22)^2 + (0.53-0.32)^2} = 0.23$$

$$d(P_1, P_3) = \sqrt{(0.05)^2 + (0.21)^2} = 0.22$$

$$d(P_1, P_4) = \sqrt{(0.14)^2 + (0.34)^2} = 0.37$$

$$d(P_1, P_5) = \sqrt{(0.32)^2 + (0.12)^2} = 0.34$$

$$d(P_1, P_6) = \sqrt{(0.40-0.45)^2 + (0.53-0.30)^2} = \sqrt{(-0.05)^2 + (0.23)^2} = 0.24$$

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0					
P_2	0.23	0				
P_3	0.22	0.14	0			
P_4	0.37	0.19	0.13	0		
P_5	0.34	0.14	0.23	0.23	0	
P_6	0.24	0.24	0.10	0.22	0.39	0

$P_1 - P_{3,6} = \min(0.22, 0.24)$

$d(P_1, P_{3,6}) = \min(d(P_1, P_3), d(P_1, P_6))$

$d(P_2, P_{3,6}) = \min(d(P_2, P_3), d(P_2, P_6))$

$d(P_4, P_{3,6}) = \min(d(P_4, P_3), d(P_4, P_6))$

$d(P_5, P_{3,6}) = \min(d(P_5, P_3), d(P_5, P_6))$

	P_1	P_2	$P_{3,6}$	P_4	P_5
P_1	0				
P_2	0.23	0			
$P_{3,6}$	0.22	0.14	0		
P_4	0.35	0.19	0.13	0	
P_5	0.34	0.14	0.23	0.23	0

$\Downarrow \longrightarrow d(P_1, P_{3,6,4}) = \min(d(P_1, P_3), d(P_1, P_6))$

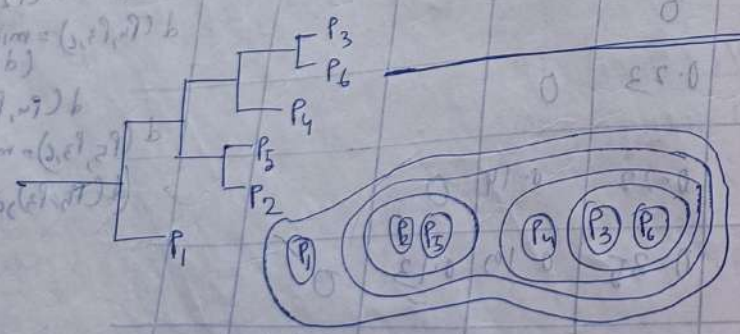
	P_1	P_2	$P_{3,6,4}$	P_5
P_1	0			
P_2	0.23	0		
$P_{3,6,4}$	0.22	0.14	0	
P_5	0.34	0.14	0.23	0

$\Downarrow$

	P_1	$P_{2,5}$	$P_{3,6,4}$
P_1	0		
$P_{2,5}$	0.23	0	
$P_{3,6,4}$	0.22	0.14	0

$\longrightarrow$

	P_1	$P_{3,1,4,2,5}$
P_1	0	
$P_{3,6,4,2,5}$	0.22	0



What is the disadvantages of K-means algo?

done (Pmw)

K-medoids algo (PAM variation).

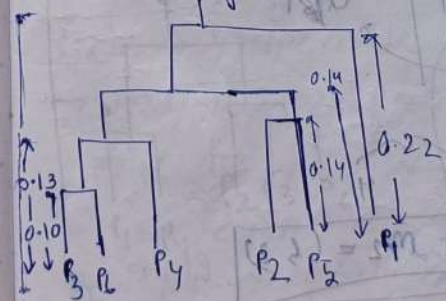
data normalization formula
$\bar{x} = \frac{x - \mu}{\sigma}$

Steps

1. Initialize: select K no. of the 'n' data points as the medoids.
2. Associate each data point to the closest medoid.
3. While the cost of the configuration decreases:
4. For each each medoid 'm' & for each non-medoid data point 'o':
5. Swapp 'm' & 'o', recompute the cost (sum of the distances of the points to their medoid).
6. If the total cost of the configuration increased in the previous step, undo the swapp.

K-medoids clustering - It is a partitioning technique that splits a data set into clusters. It's a variant of the K-means clustering algo.

dendogram



25/4/24

CN (2,6) (3,4)

PAM $|2-3| + |6-4| = 1+2 = 3$

$4+2 = 6$ → $\boxed{TC = \sum \{n-(n-k)^2\}}$

Partition Around medoids

	x	y	$m_1(3,4)$	$m_2(6,4)$	Range k=2
x_1	2	6	3	6	C_1 $x_2 = (3,4) \leftarrow m_1$
x_2	3	4	0	3	C_1 $x_6 = (6,4) \leftarrow m_2$
x_3	3	8	4	7	C_1
x_4	4	2	3	4	C_1
x_5	6	2	5	2	C_2
x_6	6	4	3	0	C_2
			10	2	

using this calc. it

$(x_1, y_1), (x_2, y_2)$

$\boxed{d = |x_1 - x_2| + |y_1 - y_2|}$ ← manhattan distance

Sum error, T.S.E = 10+2 = (12)

[TSE = total sum error]

m_1 (3,4) (SE)	m_2 (6,4) (SE)	C_1 Range	m_2 (4,2) Range	
3	8	C_1	6	C_1
0	5	C_1	3	C_1
4	9	C_1	7	C_1
3	2	C_2	0	C_2
5	0	C_2	2	C_2
3	2	C_2	4	C_2

Sum error dist. = 7 4

T.S.E = 7+4 = (11) $\boxed{\text{for } m_2 = (4,2)}$

S.E = 10, 2 $\boxed{\text{for } m_2 = (4,2)}$

∴ T.S.E = 10+2 = 12

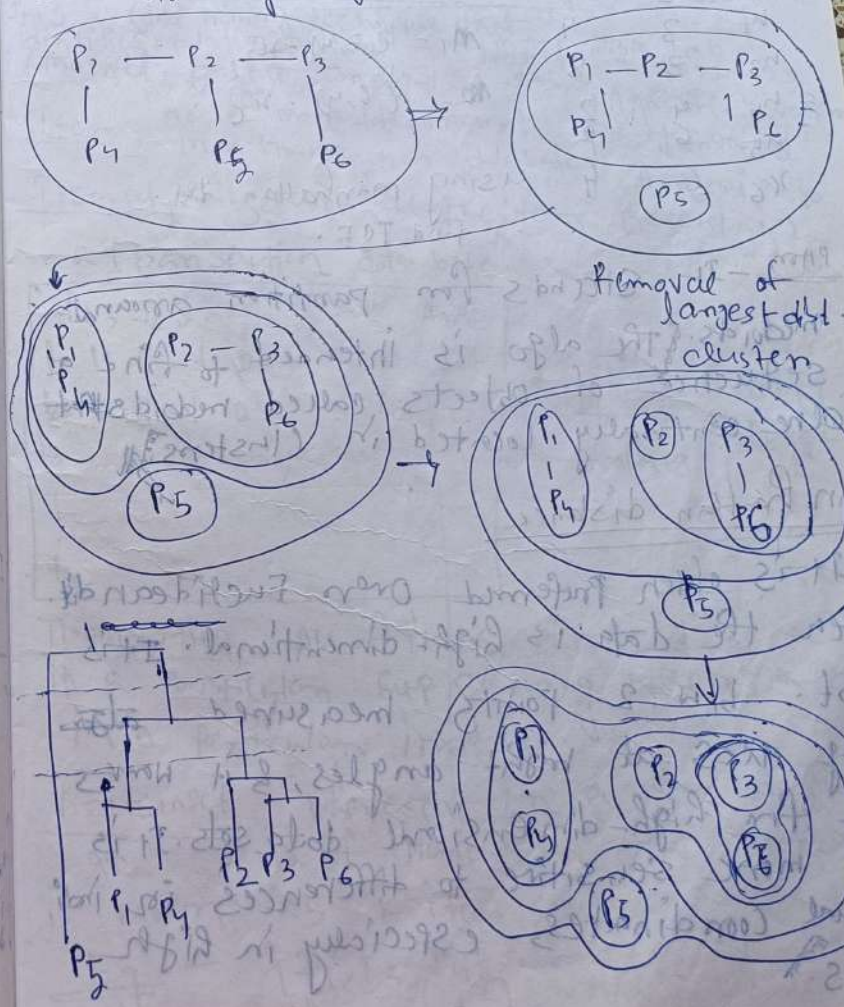
Divisive hierarchical clustering algo-

① Compute a MST for the proximity graph

② Repeat

③ Create a new cluster by removing the link corresponding to largest dist. (smaller similarity)

④ until only singleton cluster remains.



① Proximity global
② Repeat
③ create
④ singleton

(*)

	x	y
x_1	2	6
x_2	3	4
x_3	3	8
x_4	4	2
x_5	6	2
x_6	6	4

let $k=2$

$M_1 = (3,4) = x_2$

$M_2 = (6,4) = x_6$

using manhattan dist.

Find TSE.

PAM - It stands for partition around medoids. The algo is intended to find a sequence of objects called medoids that are centrally located in clusters.

Manhattan distance

It is often preferred over Euclidean dist. when the data is high-dimentional. It is dist. b/w 2 points measured along axes at right angles, & it works well for high-dimensional data sets. It is also more sensitive to differences in individual coordinates, especially in high dims.

It is calculated by summing the dist b/w 2 points $x(x_1, x_2 --- x_n)$ & $y(y_1, y_2, y_3 --- y_n)$ in n-dimensional.

The formula ~~btw~~ of dist b/w 2 points (x_1, y_1) (x_2, y_2) is - $\boxed{|x_1 - x_2| + |y_1 - y_2|}$

Data mining

Association rule mining (ARM)

30.4.24

It's a data mining technique that identifies relationships b/w items in a large ds. (market basket analysis)

Invention data

IBM Lab 1992-93

Agarwal - Inventor

↓ Invented

Apriori algo

Item: No. of manipulated present in a transaction database

Itemset: Subset of all the items present in a transaction data base.

	I_1	I_2				I_m
t_1	1	0	1	0	--- 0	1
t_2	0	1	1	0	--- 0	0
t_3						
⋮						
t_n						

this is going to be a sparse matrix.

Transaction - All of items purchased by a custom in a particular ~~sup~~ shopping mall in a day / in a particular instanse.

K-itemset - whenever in a item set there are K no. of items then that itemset is known as K item set.

Eg - $\{I_1, I_3, I_4\}$

3-Itemset

$I \leftarrow$ represents all items

- If $X \in I$, then $X \subseteq I$ (subset)
- $X \rightarrow Y$ (X is proper subset of I) $X \cap Y = \phi$
  - $X \subset I$
  - $Y \subset I$ (proper subset)

X & Y will be some K-item sets.

meaning of this implication - whenever one purchase the set of items present in X, he/she also likely to purchase the items present in Y.

Our aim is to change the revenue by changing the purchase of the customer.

1. Support-Count (S) - It is the frequency of occurrence of an itemset/item in a dataset.

TID	Items
t_1	I_1, I_2, I_3
t_2	I_1, I_4
t_3	I_1, I_3
t_4	I_1, I_2, I_4
t_5	I_1, I_2, I_3, I_4, I_6
t_6	I_2, I_3, I_6
t_7	I_2, I_3, I_6

$I_1, \ldots, I_6$

$S(I_1) = 4$ — no. of times a particular item appears in a transaction

$S(I_5) = 1$

$S(I_1, I_2) = 3$ — no. of times this 2 items purchased together in t.d.b

2. Support (S) - It of a single / set of items is the fraction / ratio of the no. of times the sing. / set of items appeared(s) in the transaction divided by the no. of all transaction

$$S(I_1) = \frac{|I_1|}{n} = \frac{4}{7}$$

$$S(I_1, I_2) = \frac{|\{I_1, I_2\}|}{n} = \frac{3}{7}$$

$n \rightarrow$ total no. of transaction $|t|$

3. Support of an association rule

$X \rightarrow Y$

Support of an association rule $X \rightarrow Y$ is the ratio of no. of occ. items present in X & Y purchased together & total no. of transactions in the whole transaction d.b.

$$S(X \rightarrow Y) = \frac{|X \cup Y|}{n}$$

$$S(\{I_1, I_2\} \rightarrow \{I_3\}) = \frac{|\{I_1, I_2, I_3\}|}{n} = \frac{2}{7}$$

[It's the % of transactions in a d.s. that contain all the items in the rule's antecedent & consequent parts]

4. Confidence of an association rule (C) — [It's a measure of it's strength / reliability.]

$$C(X \rightarrow Y) = \frac{|X \cup Y|}{|X|}$$

$X \rightarrow Y$

Y ← Consequent

X ↓ antecedent

↳ Ratio of S of $(X \cup Y)$ & S of X.

5. Frequent Item set / Large item set → It is a set of items that appear together frequently in a d.s.

$S(\{I_1, I_2\}) = \frac{3}{7} = 43\%$

- min-Support 30%

$\{I_1, I_2\} \rightarrow I_1 \rightarrow I_2$ & $I_2 \rightarrow I_1$

(sup(item set) ≥ min-sup)

$C(I_1 \to I_2) = \frac{|I_1, I_2|}{|I_1|} = \frac{3}{4} = 0.75$

$C(I_2 \to I_1) = \frac{|I_2, I_1|}{|I_2|} = \frac{3}{5} = 0.6 \approx 60\%$

~~min sup~~ Consequent

Antecendent: It is an item found in data, & a consequent is an item that appears in combination with the antecedent.

{Bread, Egg} → {milk}
Antecedent consequent

Itemset = {Bread, Egg, milk}

Ch Association Rule mining 4/5/24

Purpose: i) To find out those sets of items that are sold in the super market & frequently bought by the customer.

ii) To reduce the production of X → Y

Such ~~fond~~ association rule some algos are invented.

Apriori Algorithm - 1st algo for it (1).

Steps - 1) Generating all possible frequent item sets. (most time consuming part)

2) Generating Association rules from sets.

In order to generate this frequent item sets (1) A. Produced some algo that wasn't that efficient. But some scientists generated some faster algo in replace of it.

Demonstration

Shorter / simpler version of Apriori algo (simplified)

Iterative algo (principle)

Based on which a this algo was proposed - Downward closure property

Eg - Any subset of a frequent item set are also frequent. Conversely also we can say that

Any superset of an infrequent item set are also infrequent.

Eg - i) {I_1; I_3; I_4} → Purchased together by a customer

If it's frequent item sets then, any subset of them are also frequent

ii) Infrequent item sets {I_4, I_7} → Aren't purchased together by the cus.

Then, {I_4, I_7, I_9} (or) {I_4, I_6, I_9} → Infrequent (superset)

- It is an iterative algo.

1) ~~Find~~ First of all find all 1-item sets which are frequent (1-item frequent item sets)

2) then find, all 2-item frequent item sets. ---

3) Like by, k-item frequent item sets.

4) In each iteration, K(say) → only considers item sets that ~~are~~ contains some (K-1) frequent item sets.

(eg), 5th item set → 5-item set → {I_1, ..., I_5} → consider only (5-1) = 4 no. of freq. items (not infreq. items) 4 no. of items

Algo 2nd

1) Find, freq item sets of size $k=1: F_1$

2) From $k=2$

3) C_k = candidate item sets of size k that could be frequent given F_{k-1}.

4) F_k = those item sets that are actually frequent $F_k \subseteq C_k$. For this we need to scan the database again.

Output: All possible freq. i.s. (F) =

$$F_1 \cup F_2 \cup \dots \cup F_k$$

2nd Eg – $I = I_1, I_2, \dots I_{10}$

F_1: $\langle I_1 \rangle \langle I_2 \rangle \langle I_5 \rangle \langle I_4 \rangle$

3) C_2: $\langle I_1, I_2 \rangle \longrightarrow F_2$ (If it is purchased for more than 5 times so, if not then this not be F_2) considered as

$\langle I_1, I_5 \rangle$ ×

$\langle I_1, I_7 \rangle \rightarrow F_2$

$\langle I_2, I_5 \rangle \rightarrow F_2$

$\langle I_5, I_7 \rangle \rightarrow F_2$

$\langle I_4, I_7 \rangle$ ×

$\langle I_2, I_4 \rangle$

F_2: $\langle I_1, I_2 \rangle, \langle I_1, I_7 \rangle, \langle I_2, I_5 \rangle, \langle I_2, I_7 \rangle$

4) $\langle I_2, I_7 \rangle \langle I_5, I_7 \rangle$ (let)

(From here, by scanning again we need to get (3))

1. When every a tran. d.s is given then sort them in ascending order acc. to their item no.
2. Arrange the ~~support~~ count ~~for~~ items in descending order acc. to their ~~item no.~~ support counts

While merging a pair of frequent item 3. sets of size $(k-1)$ to obtain an freq. itemset of size k, merge only those pair of itemsets whose item no.s are differ by only 1 item, specifically those itemsets which differs only by the last item.

NOW: F_2: $\langle I_1, I_2 \rangle \langle I_1, I_7 \rangle \langle I_2, I_5 \rangle \langle I_5, I_7 \rangle \langle I_2, I_7 \rangle$

(whenever there will be nine 2 diff. items then we'll combine)

- $\langle I_1, I_2, I_5 \rangle \longrightarrow$ not combine $\langle I_1, I_2 \rangle$ & $\langle I_2, I_5 \rangle$ as, $\langle I_1, I_5 \rangle$ is not freq. (merge it, but don't put it in F_3 but in C_3)
- $\langle I_1, I_2, I_7 \rangle \rightarrow$ will combine as, freq. item set.
- $\langle I_2, I_5, 7 \rangle \rightarrow$ will combine

Need to generate 3 items freq. item set

but

- Can't merge $\langle I_2, I_7 \rangle \langle I_2, I_5 \rangle \rightarrow$ [4 items]

$\therefore C_3$: $\langle I_1, I_2, I_7 \rangle, \langle I_1, I_2, I_5 \rangle, \langle I_2, I_5, I_7 \rangle$, $\langle I_1, I_5, I_7 \rangle$, ~~$\langle I_2, I_5, I_7 \rangle$~~

F_3: $\langle I_1, I_2, I_7 \rangle \langle I_2, I_5, I_7 \rangle$

If min_sup is not given then also we can calc. by cal. infreq. item sets.

$C_4 = \langle I_1, I_2, I_5, I_7 \rangle$ ×

F_4 = NULL (stop here) [as $\langle I_1, I_5, I_7 \rangle$ – isn't here, i.e infrequent]

Eg 1 2nd

TID	Items
t_1	1, 2, 4
t_2	2, 3, 5
t_3	1, 2, 3, 5
t_4	2, 5

$C_1 \rightarrow F_1 \rightarrow C_2 \rightarrow F_2 \rightarrow C_3 \rightarrow F_3 \rightarrow C_4 \rightarrow F_4 \rightarrow$

C_1

$<1>: 2$ (support/freq.) $<2>: 3$, $<3>: 3$, $<4>: 1$ ✗ (scanning)

$<5>: 3$

let, $\boxed{\text{min-support} = 2}$ (for ex it's)

F_1 (freq. item set): $<1>, <2>, <3>, <5>$ (scanned) — these items there are purchased 2 / more times

combine

C_2: $<1,2>$: 1, $<1,3>$: 2, $<1,5>$: 1 ✗, $<2,3>$: 2, $<2,5>$: 3, $<3,5>$: 2

(Scanning)

F_2: $<1,3>, <2,3>, <2,5>, <3,5>$

Combine

C_3: $<1,3,2>$: 1 ✗, $<1,3,5>$: 1 ✗, $<2,3,5>$: 2 [∵ $C_3 \rightarrow$ so, 3-items freq. item set]

(Scanning)

F_3: $<2,3,5>$

✗ → antito infrequent

As $<1,2>$ is infreq. so we need to generate it (By downward closure property)

Frequent pattern free growth algo — It is a popular method for frequent pattern mining in data mining. It is based on a d.s. — FP tree.

FP tree — It is a compact d.s that represents the data set in tree form. Each transaction is read & then mapped onto a path in the FP tree.

Div

Imp Frequent pattern growth ~~tree~~ algo 7/5/24

FP-tree growth algo — This is based on a d.s. — FP tree

Advs — we ~~need~~ scan the database (transaction db) only twice [main adv]

Advs

1st — we'll construct the FP tree.

2nd — then we will find the association rules from those frequent item sets.

Eg

t_1: I_1, I_2, I_5

t_2: I_2, I_4

t_3: I_2, I_3

t_4: I_1, I_2, I_4

t_5: I_1, I_3

t_6: I_2, I_3, I_6

t_7: I_1, I_3

t_8: I_1, I_2, I_3, I_5

t_9: I_1, I_2, I_3, I_7

[t = transaction]

[I = Items]

transaction data set → In ascending order acc. to Items are their item no.s.

Assuming min-sup : 2 ~~(How many times~~

$S(I) = |I| \rightarrow$ Support count (how many times an items ~~have has been~~ has been purchased)

Support count or Absolute support or frequency count

I_1: 6

I_2: 7

I_3: 6

I_4: 2

I_5: 2

I_6: 1 ✗ [as min-sup = 2]

I_7: 1 ✗ [— = 2]

2) Arrange the items according to their descending order of support count

$I_2 : 7$
$I_1 : 6$
$I_3 : 6$
$I_4 : 2$
$I_5 : 2$

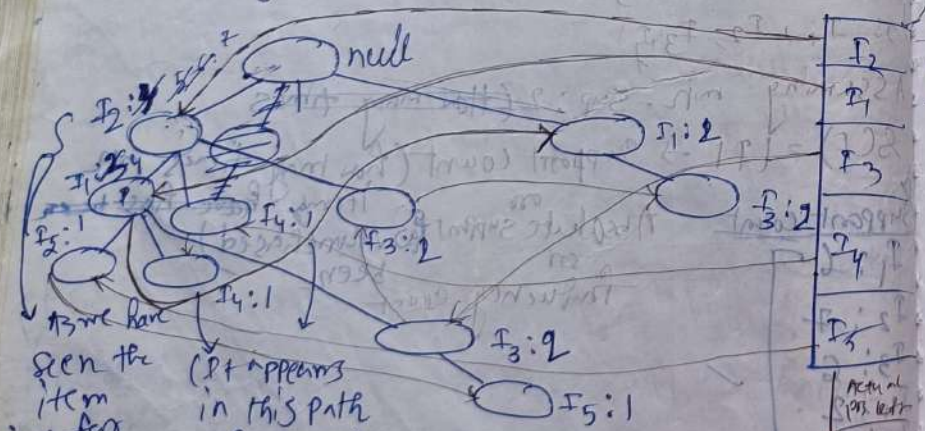
3) $t_1 : I_2, I_1, I_5$ — scan the items for once
$t_2 : I_2, I_4$
$t_3 : I_2, I_3$
$t_4 : I_2, I_1, I_4$
$t_5 : I_1, I_3$
$t_6 : I_2, I_3$
$t_7 : I_1, I_3$
$t_8 : I_2, I_1, I_3, I_5$
$t_9 : I_2, I_1, I_3$

We'll manage the items in tr. in descending order to acc. to their supp. count

Can also cross check the freq. patt. / itemsets from here

4) Build the FP tree

root node's label = null



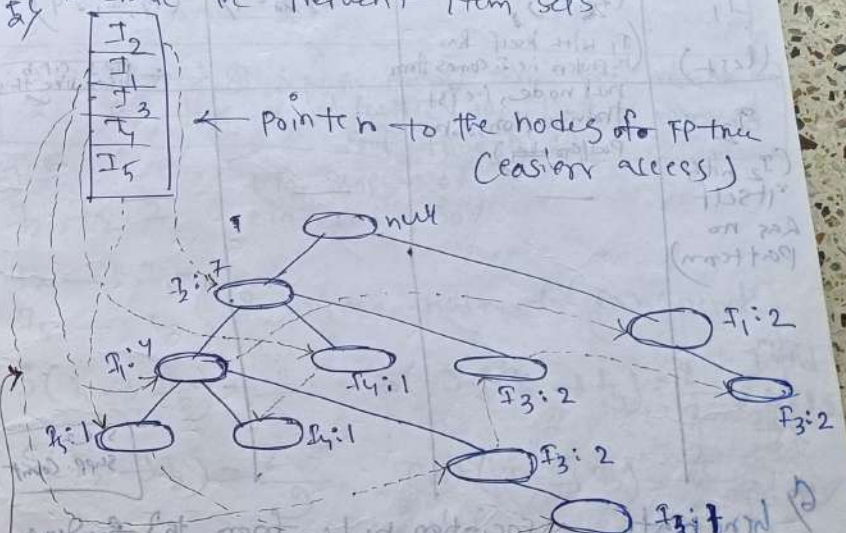
As we have seen the item just for once so 1 (count)

(It appears in this path for once) (So no increment)

FP-tree

We'll keep this d.s in RAM & will utilize this d.s for associative rule mining (as the size of this tree d.s is very ~~too~~ small). Besides we need only 1 scan for now (time saving) → main adv

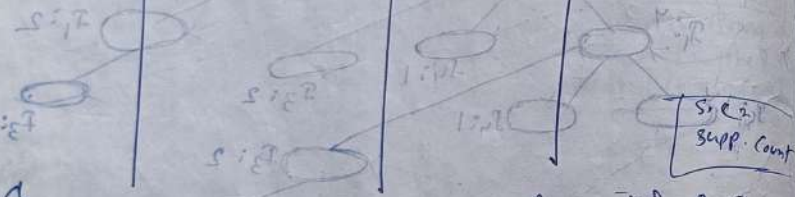
5) Generate the frequent item sets



We have to consider items acc. to their ascending order of sup c

Item	Conditional pattern base	condition FP tree	Frequent pattern itemsets
I_5	$(I_2, I_1 : 1)$ (I_2 follow by I_1) (Items appears before I_5 in FP tree) (1st appearence of I_5) $, (I_2, I_1, I_3 : 1)$	(As we have traversed only 1 path from I_2 to I_5, so 1)	(common part) $\langle I_2 : 2, I_1 : 2 \rangle$ (Combining this 2) (Count of I_3 1, & min-sup = 2 so, I_3 X)	$(I_2, I_5 : 2)$, $(I_1, I_5 : 2)$, $(I_1, I_2, I_5 : 2)$ (As (I_2, I_1) has been appeared in c.p.b for 2 (min-sup) times) [If, I_1 & I_2 never appeared for 2 times (i.e. <2) then they weren't incremented]

I_4	$(I_2, I_1 : 1)$, $(I_2 : 1)$	$\langle I_2 : 2 \rangle$ (common)	$(I_2, I_4 : 2)$
I_3	$(I_2 : 2)$, $(I_2, I_1 : 2)$, $(I_1 : 2)$	$\langle I_2 : 4, I_1 : 2 \rangle$, $\langle I_1 : 2 \rangle$ from (I_2, I_1 5 are diff. node so can't combine)	$(I_2, I_3 : 4)$, $(I_1, I_3 : 4)$, $(I_2, I_1, I_3 : 2)$
I_1 (last)	$(I_2 : 4)$, (I_1 with itself has no pattern i.e I_1 comes from root node, i.e 1st transaction, ∴ no pattern)	$\langle I_2 : 4 \rangle$	$(I_2, I_4 : 4)$ ← I_1 & I_4 then like this
I_2 (I_2 with itself has no pattern)	—	—	—



$S_i(i_2)$ supp. count

Generate association rule from this frequent pattern & keep only those ass. rul. that satisfy min-sup & min-conf threshold.

Eg (a) $(I_2, I_1, I_5 : 2)$

(consider) $\boxed{X \longrightarrow Y}$ (consider)

$(I_2, I_1) \longrightarrow I_5$

$\boxed{S(X \to Y) = \frac{|X \cup Y|}{|N|}}$ $\boxed{conf(X \to Y) = \frac{|X \cup Y|}{|X|}}$ $|N|$ = total no. of transactions

$S(I_2, I_1 \to I_5) = \frac{2}{9} \cong 0.22$, $conf(I_2, I_1 \to I_5) = \frac{2}{4} = 0.5$

min-supp = 20% (let)

min-conf = 50% (let)

In this case we can tell, this is a valid association rule, as it's satisfying min-supp & min-conf threshold.

If there is n no. of items, in t.d.b, then the no of valid association rule can be produced $= \boxed{2^n - 2}$

as, $X \longrightarrow Y$ ✓
$Y \longrightarrow X$ ✗ (As every ass. rul. can not feather the limit of min-supp & conf threshold.)

Eg-(b)

$I_2, I_5 : 2$ min-sup = 20%, min-conf = 50%

$I_2 \longrightarrow I_5$
$I_5 \longrightarrow I_2$ } 2 associ. rule. we can generate

$S(I_2 \to I_5) = \frac{2}{9} = 0.22$ $conf(I_2 \to I_5) = \frac{2}{7} = 0.285$ → ✗ (not valid)

$S(I_5 \to I_2) = \frac{2}{9} = 0.22$ $conf(I_5 \to I_2) = \frac{2}{2} = 1$ → ✓ (valid)

Present $\frac{9|5|24}{\text{in}}$

Use

- $X \longrightarrow Y$ (same as pre) ⟹ whenever items X are purchased items Y are highly likely to be purchased also (this is used by retailers to increase the revenue of the online retailers sites)

Insignificant rules - Lets understand through eg.

(already obtained) {aged} → {cataract} ✓ (valid rule)

× {aged, male} → {cataract} → insignificant as we already obtained {aged} → {cataract}. So this

is not giving, that much meaningful insight. So this is insignificant rule.

Redundant rules

Let's understand through eg

{aged, foot baller} → {catapult} → v=1/2

but might have been infant giving the meaning that most of the foot baller are catapult (But it need not to ~~be~~ happen).

So, it's a redundant rule.

	Cofee	$\overline{\text{Cofee}}$	Rowsum
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
colsum	90	10	100 (n) (total person)

D.S. of drinking behaviour of person (D.S = Dataset)

After obtaining the asso. ru., ~~say~~ 1 of the asso. rule is –

{Tea} → {cofee}

[for this, T∩C=ϕ always true]

$S(\text{Tea} \to \text{Cofee}) = \frac{|\text{Tea} \cup \text{Cofee}|}{N}$ (metric)

Quantitative measurement of collected datas.

$= \frac{95}{100} = 0.95$

$= \frac{80}{100} = 0.8$

$P(T) + P(C) - P(T \cap C)$

$= 20 + 90 - 95$

$= 95$

$C(\text{Tea} \to \text{Cofee}) = \frac{|T \cup C|}{|T|} = \frac{15}{20} = 0.75$ (metric)

then, calc. asso. ru. then intersection isn't there of

let, min-supp = 70%
min-Conf = 75%.

supp-count

when the ass. rule ~~have~~ has high confidence but, the ass. rule isn't appropriate ~~trait~~, then we need Lift ← Lift (metric) Imp

misleading

$$\text{Lift}(x \to y) = \frac{C(x \to y)}{S(y)}$$ (metric)

$$= \frac{S_c(x \cup y) / S_c(x)}{S_c(y) / n}$$

$$\text{Lift}(x \to y) = \frac{n + S_c(x \cup y)}{S_c(x) \cdot S_c(y)}$$

Imp

$$\text{Lift}(T \to C) = \frac{100 + 15}{20 + 90} < 1$$

→ As L<1, so it's invalid, i.e. the corresponding ass. rule invalid.

- It will be valid if it is >1, otherwise it's invalid. (Lift)

$$\text{Lift}(\overline{T} \to C) = \frac{100 + 75}{80 + 90} > 1$$ (the ass. rule is valid)

answer if the lift > 1 or not

$$C(\overline{T} \to C) = \frac{75}{80} = 0.90$$ (satisfying the min-Conf threshold)

- why ass. rule isn't appropriate in this case?

Here, {Tea} → {cofee} this association rule implies Tea means cofee, that isn't

approximate / it is misleading. And we can see though it is statisfying the min_conf & min_supp it isn't satisfying the lift metric. So it is invalid & logically it is invalid too. For eg, the next ass. rule. $T \to C$ (not tea means coffee) is both logically & mathematically valid, we can see that.

③

Supervised m.L. algo

→ classification algos → (purpose: To predict the class level of records)

→ regression → (whose class level is already known based on) (to generate continuous / numerical vals)

Classification algo

Findout the mapping func that can accurately predict the required datas based on the already available datas.

Eg — Already given some datas - Height, weight, gender of some students

Need to predict - Height, weight, gender of some student whose data isn't disclosed by the classification algo (ml algo).

$f(H, W, A) \to G$ (i.e, H, W, A of those are given, but not g)

generates — Catagorical levels / o/ps.

[class label info]

Regression (prov)

generates — Numerical values / continuous values. (prov)

• Diff. b/w supervised & unsupervised learning techniques.

Classification — It generates the class label of unknow data set. mapping func

To know whether our a/class. algo is properly predicting the unknow d.s. or not, we → split training d. / class d.s. into 2 parts — i) Training set ii) test set.

Let, 5000 students (H, W, A, G) ← Actual g info

80 : 20

$T_r \to$ 4000 students

$T_{st} \to$ 1000 students (H, W, A)

↳ But here we'll pass only the H, W, A not G info

↓ Gender

G_p (It is not actual g info it's predicted g info, so G_p) [known, G_A → Actual g info]

When we'll perform the training we'll use all this values in d.s.

T_r

$P(H, W, A) \to G_A$

→ m (mapping func)

test

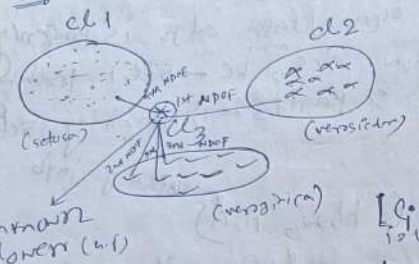
$m(H, W, A) \to G_p$ (predicted) — G_A (Actual)

Compare to see the accuracy of our mapping func / class. algo

KNN (K-nearest neighbour classification algo)

K = the no. of neighbour info that we use to build our classification model.

↳ [A supervised m.l. algo that uses a dist-ance-based approach to classify / predict the grouping of a data point]



[Cl = class]

let K = 5

[N.D.O.F = nearest dist of flower]

[$C_i = C_1, C_2, C_3$; $i = 1-3$] [Cl ≈ C]

As the, NDOF towards Cl_3 is the most, so we'll consider that unknown obj falls into Cl_3.

(demonstration)

- Cal. the dis b/w the u.f & all other flowers
- see, the ~~shortest dis b/w u.f & K no. of n.n~~ (nearest neighbours), here 5 n.n.
- Among the NDOF (n.n), we can see, {3, 3, 3, 3, 2} i.e the majority of the nearest flowers is belonging to Cl_3, so that u.f belongs to 3rd category / Cl_3.

$\{C_1 = \frac{1}{5} = 0.2,\ C_2 = \frac{1}{5} = 0.2,\ C_3 = \frac{3}{5} = 0.6\}$ → Heighest

for this cg

- If we have a tie among the no. of n.n b/w the classes we can increase the value of K & ~~see the same~~ check again (Way 1)

eg – {3, 2, 3, 2, 1}
d_1 d_2 d_3 d_4 d_5

[d = distance]

~~Way 2~~ $d_1 + d_3$ Compare $d_2 + d_3$

if $(d_1 + d_3) < (d_2 + d_3)$, then $Cl_3 \leftarrow$ u.K

if $(d_1 + d_3) > (d_2 + d_4)$, then $Cl_2 \leftarrow$ u.K

Date

KNN Algo

1) Store all the input data in the training set
2) For each object in the test set
3) search for K nearest neighbours to the input obj's / patterns using any distance matrix (Euclidean ~~etc~~ etc) (obj's)
4) For classification we'll compute the confidence for each classes as C_i / K where C_i is the no. of obj's among the K-nearest neighbours belonging to class label 'i'
5) The classification for the i/p obj is the class with the heighest confidence.

Eg ①

Dataset

(eg of binary classification)

S_1	CGPA	Assim	Proj m	Result (class label attribute)	
1	9.2	8.5	8	Pass	Training d.s (available d.s)
2	8	8.0	7	Pass	
3	8.5	8.1	8	pass	
4	6	4.5	5	Fail	
5	6.5	5.0	4	fail	
6	8.9	7.2	7	pass	
7	5.8	3.8	5	Fail	
8	8.9	9.1	9	pass	

A student has CGPA 6.1, assig = 4.0, Proj = 5, what is the result? with k=3 &

with k = 5

9.2, 8.5, 8

dist (euclidean) lowest 3 dist - C1/C2 conv here

one may be FAIL

1) $\sqrt{(9.2-6.1)^2+(8.5-4.0)^2+(8-5)^2}$

1. = 6.0728

8, 8.0, 7

$= \sqrt{(8-6.1)^2+(8-4)^2+(7-5)^2}$

2. = 4.86

8.5, 8.1, 8

$= \sqrt{(8.5-6.1)^2+(8.1-4)^2+(8-5)^2}$

3. = 5.62

6, 4.5, 5

$= \sqrt{(6-6.1)^2+(4.5-4)^2+(5-5)^2}$

4 = 0.51

6.5, 5, 4

$= \sqrt{(6.5-6.1)^2+(5-4)^2+(4-5)^2}$

5. = 1.47

8.2, 7.2, 7

$= \sqrt{(8.2-6.1)^2+(7.2-4)^2+(7-5)^2}$

6 = 4.312

5.8, 3.8, 5

$\sqrt{(5.8-6.1)^2+(3.8-4)^2+(5-5)^2}$

7 = 0.36

8.9, 5.1, 9

$\sqrt{(8.9-6.1)^2+(5.1-4)^2+(9-5)^2}$

8 = 7.06

2) k = 3, C_i, C_1, C_2 (i.e, i = 1, 2)

~~i = 3, i = 1, 2, 3~~

let, C_1 = Pass

C_2 = Fail

~~C1~~ = C_1 =

C_2 =

C_3 =

as k = 3, 3, nearest neighbours are –

= 0.36, 0.51, 1.47

= {Fail, Fail, Fail} → class label

∴ confidence, $C_1 = \frac{0}{3} = 0$

$C_2 = \frac{3}{3} = 1$ → Highest confidence

∴ (6.1, 4.0, 5) belongs to C_2 or Fail.

3) k = 5, ∴ i = 1, 2 ∴ C_i = C_1, C_2

let, C_1 = Pass, C_2 = fail

as k = 5, 5 nearest neighbours are –

= 0.36, 0.51, 1.47, 4.86, 5.62

= {Fail, Fail, Fail, Pass, Pass}

∴ $C_1 = \frac{2}{5} = 0.4$

$C_2 = \frac{3}{5} = 0.6$ → Highest confidence

∴ (6.1, 4.0, 5) belongs to C_2 or Fail.

Disadv of apriori algo & it's remedy?

Apriori algo is an iterative algo, it needs scan in each iteration, that is time consuming. Besides, it uses more memory to store information to utilize it. So, this algo is slow & takes more space in RAM. (Disadv)

The remedy of this disadv is, FP-tree growth algo. Here, we construct a FP-tree & keep this d.s (FP tree) in RAM & utilize it in association rule mining, as the size of this d.s (FP tree) is very small. Besides it needs only 2 scans allover, so, it is faster & takes less space in RAM (H.D.D). that saves time.

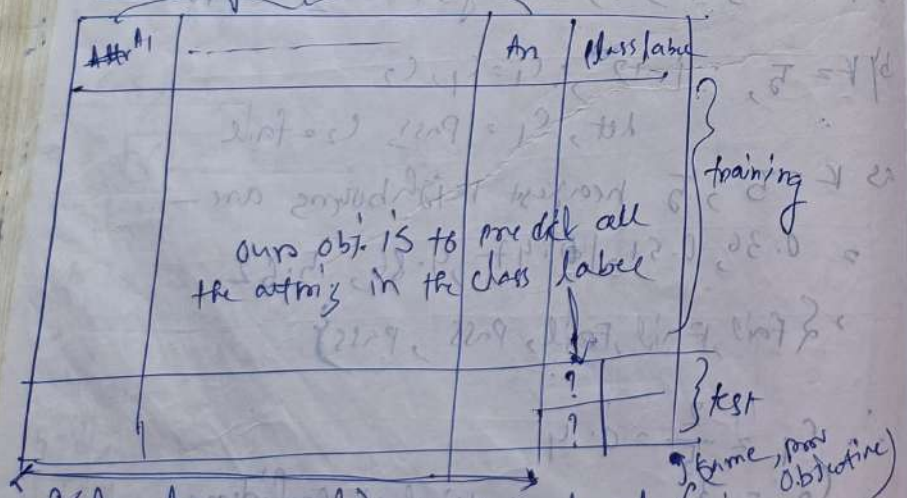
[Diff. b/w apriori & FP-tree growth algo]

CW

12/5/24

Decision tree algo/classifier (Imp) Supp

Training dataset



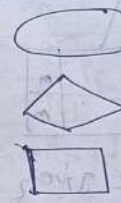
$f(A_1, A_2, \ldots A_n) \rightarrow$ class label = (Objective: same, prev objective)

- A fun that can properly/perfectly map the attributes to the class lable)
- For test d.s - we have 2 class label 1) predicted lebel, 2) actual label. (PTO)

~~After finding the mapping func~~

- Our obj. is to build a tree like struct from the given training d.s.

Decision tree's node's type



Root Node

Internal / Decision node

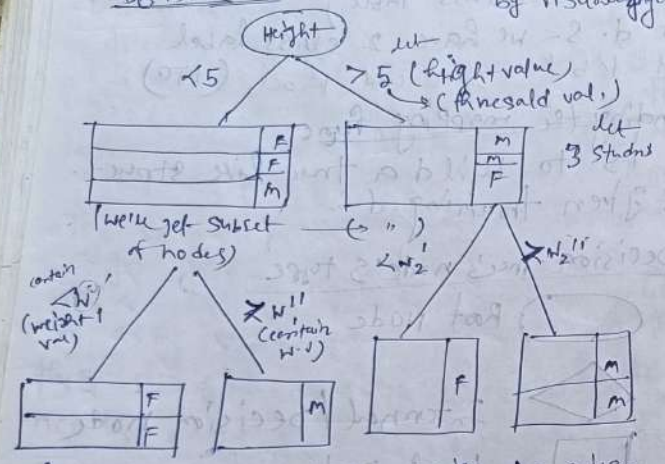
Leaf node

When ever we start to build the decision tree at each iterat. we try to generate internal node, each internal node will contain the subsets of training d.s. we split the subset of nodes into another subset of nodes until we get all the attributes of leaf nodes. (Homogeneous d.s)

Eg-

Height	weight	Gender
h1	w1	M
h2	w2	F
h3	w3	M
h4	w4	M
h4	w5	F
h6	w6	F

Decision Tree (A d.s that helps us to make decisions by visualising outcomes.)



When an unknow d.s. will be given (H, W) [unknown] hi & W, then we have follow the tree. [actually H & W will be filtered while passing the tree & we will get the gender]

Haunt's algo → 1st ~~disc~~ decision tree algo

Decision tree algos: (4.5, CART, ID3)

ID3 $\xrightarrow{\text{metric}}$ Entropy

CART $\xrightarrow{\text{metric}}$ Gini index

Entropy - degree of uncertainity associated with a variable.

Eg: Coin (H, T)

Here deg of uncertainity = 50% (50% we will get H / 50% we will get T)

throw a dice

dice, deg of uncertainity = $\frac{1}{6}$

$\frac{1}{6} < \frac{1}{2}$

deg of uncertainity of throwing dice

deg of uncertainity of flipping coin

- When we have higher uncertainity, the we are going to have higher entropy & vice versa.

$$\boxed{\text{Entropy} = -\sum_{i=1}^{m} p_i \cdot \log p_i}$$

p_i → probability of class labels

p_i = proportion of studs, who passed / failed the exam

$i = 2$, $m = 2$

Eg: 10 students (let)
Passed — 6
Failed = 4

~~outcomes~~ then there are 2 diff possible outcomes,

$$\text{entropy} = -\left(\frac{6}{10} \cdot \log_2 \frac{6}{10} + \frac{4}{10} \cdot \log_2 \frac{4}{10}\right)$$

Eg — ① (100, 0) = class label (let dataset (homogeneous d.s))

positive

none of the stud/ns belongs to neg class

lets no. of seconds = 100

$$\text{entropy} = -\left(\frac{100}{100} \log_2 \frac{100}{100} + \frac{0}{100} \log_2 \frac{0}{100}\right)$$

$$= 0$$

- Homogeneous d.s = entropy 0 or lowest

(All the class labels belong to a single category)

class label

Eg-2 Equal d.s
(50%, 50%)

$$\text{entropy} = \frac{50}{-}\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)$$

$$= 1$$

- class labels are equally distributed for a d.s - = entropy 1 or highest
- entropy value ranges b/w 0 & 1.

ID3 decision tree algo

- T → Training d.s
- A → set of all attributes in the d.s $A_1, A_2 \ldots A_n$
- m → no. of classes
- p_i → Probability that a data instance (d) / obj belongs to class C_i ($i = (1, \ldots m)$)
- $p_i = \boxed{\frac{|d_{c_i}|}{|T|}}$
- p_i → total no. of data instances belongs to class C_i.

$$\text{Entropy_Info}(T) \xrightarrow{\text{def}} \boxed{-\sum_{i=1}^{m} p_i \log_2 p_i} \quad (I)$$

$$\text{Entropy_Info}(T, A) = \boxed{\sum_{i=1}^{v} \frac{|A_i|}{|T|} * \text{Entropy_Info}(A_i)} \quad (II)$$

$\text{Entropy_Info}(A)$ for whole Tr. d.s based on A → def

where, v = total no. of distinct values for a particular attr

$|A_i|$ = No. of attr for a particular distinct values in the records.

Information Gain (A) based on A (attribute) partition

$$= \boxed{\text{Entropy_Info}(T) - \text{Entropy_Info}(A)} \quad (III)$$

(s the amt of reduction we'll have when we divide for the particular d.s

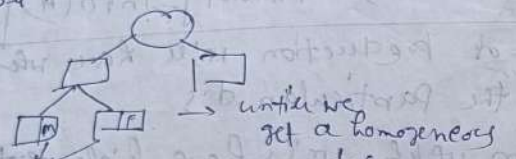
- Attribute for which we'll have higher entropy info gain → will be the root

ID3 algo (cont.)

1) Compute entropy info (T) using eqn (I) for the whole training d.s
2) Compute entropy_Info (A) & Info Gain (A) with eqn (II) & eqn (III) for each attribute A in the training d.s.
3) Choose the attribute for which entropy is minimum & therefore info gain is max. (entropy gain is min & so info gain is max & vice versa)
4) the best split attribute is placed as the root node.
5) the root node is branched into subtrees with each subtree as an outcome of the best cond/sat of the root attribute &

the training d.s is also split into subtrees / subsets

6. Recursively apply the same procedure for the subset of the training datasets with remaining attributes until a leaf node is derived or no more training instances are available in the subset.



Eg-① Smp

↓ training d.s

CGPA	Internship yes	Practical knowledge	Communication skill	Job offer (class label)
≥9	Yes	very good	Good	Yes
≥8	N	G	moderate (m)	Yes
≥8	N	Avg	Poor (P)	NO
<8	N	A	G	N
≥8	Y	G	M	Y
≥9	Y	G	P	N
≥9	N	V G	G	Y
≥8	Y	G	G	Y
≥8	Y	Avg	G	Y

Test d.s – for any unknown CGPA, I, P.K, C.S

1st it

1) Entropy Info(T) = $\boxed{-\sum_{i=1}^{m} p_i \cdot \log_2 p_i}$ [attri. of training set → Job offer]

Entropy-Info (Job offer) =

$= -\left(\frac{7}{10} \cdot \log_2 \frac{7}{10} + \frac{3}{10} \log_2 \frac{3}{10}\right)$ ← This is taken from an attri.

$= -\{(-0.3599) + (-0.5208)\}$

$= 0.8807$

$\boxed{\log_2 9 = \log_{10} 9 \times \log_2 10 = .323}$

2) Entropy-Info (T, A) =

i) $\boxed{\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy-Info}(A_i)}$ ← Based on attribute CGPA

~~Entropy-Info (T, (CGPA))~~

CGPA	Job offer = Y	Job offer = N	Total	Entropy (not needed)
1 ≥9	3	1	4	
2 ≥8	4	0	4	0
3 <8	0	2	2	0

homogeneous data

Entropy-Info (Job offer, (CGPA)) = $\frac{4}{10}\left[-\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right)\right]$

E-Info(CGPA) $+ \frac{4}{10}\left[-\left(\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right)\right] + \frac{2}{10}\left[-\left(\frac{0}{2}\log_2\frac{0}{2} + \frac{2}{2}\log_2\frac{2}{2}\right)\right]$

$= 0.3245$

Info Gain (CGPA) = 0.8807 − 0.3243
= 0.5564 (highest) ← Based on Interactiveness

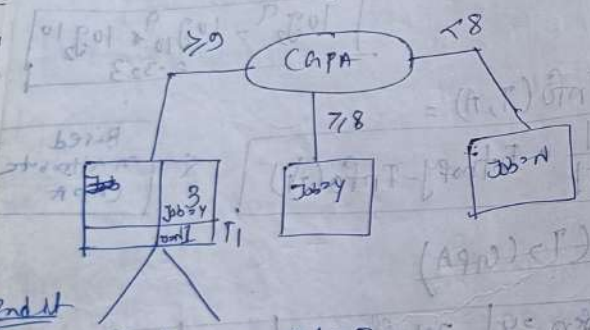
ii) Entropy Info (J, Interactiveness)
= 0.79

Info Gain (Interactiveness)
= 0.0311

As there are 2 values, it's so hard to create tree. If u understanding library tree then don't create

iii) I.G (P.K) = 0.2246 ← Based on P.K
iv) I.G (C.S) = 0.5203 ← Based on C.S

∴ root node = CGPA

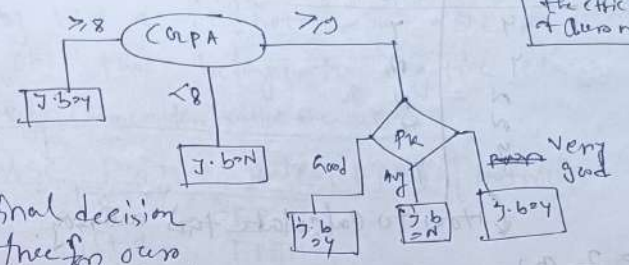


2nd lvl

i) E.I (T1) = 0.8108
ii) E.I (T1, Interactiveness) = 0.4997
iii) I.G (Interactiveness) = 0.8108 − 0.4997
= 0.0311

iii) I.G (T1, P.K) = 0.8108
iv) I.G (T1, C.S) = 0.8108 } common highest info. gain val.

(can choose any one of this attri. to split the d.s T1)



(final decision tree for our training d.s)

With test d.s we check the efficiency of our model/tree

Adv of decision tree

i) If any unknown test set is given we can traverse the tree & can tell / predict the result.

Eg: S.S, Y, G, M → Y (J.O (class label))
7.9, Y, V.G, G → N
9.2, N, A, G → N
} Prediction of the class label of the unknown students based upon known students data set / decision tree

Imp
Efficiency of classifications

Imp
Accuracy

test → class label
Actual label, Predicted label

Confusion matrix

		Predicted class label: YES	Predicted class label: NO
Actual class label	YES	True positive (a)	False negative (b)
	NO	False positive (c)	True negative (d)

← Job offers
→ falsly categorized as no/neg.
→ No of instances
→ Falsly predicted as yes

1 with test

$= 0.22 - 0.43$

$= 0.65$

$\underline{\underline{v = 2}}$

ii) E_I (i, Intern.)

is

E_I (Intern.) $= \frac{6}{10}\left[-\left(\frac{5}{6}\log_2\frac{5}{6} + \frac{1}{6}\log_2\frac{1}{6}\right)\right] + \frac{4}{10}\left[-\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right)\right]$

$= \frac{6}{10} \times (0.65) + \frac{4}{10} \times 1$

$= 0.79$

Intern	Y.O = Y	40 = N	Total
Yes	5	1	6
NO	2	2	4

(Homogeneous)

(1)

1 with test

$v = 3$ (7)

iii) $E_I(y, P.K)$

S_y

$$E_I(PK) = \frac{2}{10}\left[-\left(\frac{2}{2}\log_2\frac{2}{2} + \frac{0}{2}\log_2\frac{0}{2}\right)\right] + \frac{5}{10}\left[-\left(\frac{4}{5}\log_2\frac{4}{5} + \frac{1}{5}\log_2\frac{1}{5}\right)\right] + \frac{3}{10}\left[-\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right)\right]$$

$= 0 + 0.35 + 0.27$
$= 0.62$

P.K	y.o.y	y.o.N	total
V.G	2	0	2
G.	4	1 (Homogeneous deg)	5
Avg	1	2	3

$= -0.25$
-0.46
-0.62
-2.5

iv) $E_I(y, C.S)$ $\quad v = 3$

S_y

$$E_I(C.S) = \frac{5}{10}\left[-\left(\frac{4}{5}\log_2\frac{4}{5} + \frac{1}{5}\log_2\frac{1}{5}\right)\right] + 0 + 0$$

$= 0.357$ [∵ H.]

C.S	y.o.y	y.o.N	total
G	4	1	5
M.	3	0 (H.)	3
P	0	2 (H.)	2

-0.37

Now, we have to filter no ≥ 0 as it has subsets

Now $E_I((\text{hpa} \geq 0) = \left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right]$
$\quad \parallel$
$\quad T_1$

$= -0.81$

2) (i) ~~E_I (Intern) = $\frac{3}{4}$[~~ ~~$\frac{1}{4}$~~ ~~$v = 2$~~

$V = 2$

(3)

2/(i) $E_I(\text{Intern.}) = 0 + \frac{2}{4} + 1$

$= 0.5$

Intern	y.o.y	j.o.N	total
yes	2	0	2
NO	1	1	2

$V = 3$

(ii) $E_I(P.V) = 0 + 0 + 0$

$= 0$

P.V	y.o.y	j.o.N	total
V.G	2	0	2
G.	1	0	1
A.	0	1	1

$V = 3$

(iii) $E_I(C.S) = 0 + 0 + 0$

C.S	y.o.y	j.o.N	total
G.	2	0	2
M.	1	0	1
P	0	1	1

Eg:

Actual	Predicted
Y	Y
Y	N
N	N
N	Y
Y	Y

{ How to calculate this accuracy.

TP = 2 (a)
FN = 1 (b)
FP = 1 (c)
TN = 1 (d)

① $\boxed{\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN}}$

$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{2+1}{5} = \frac{3}{5} = 60\%$

Limitation of accuracy ~~major~~ measure

10,000

Actual class label	
9990	10
P	N

Predicted class label (P) → 10,000
N → 0

Imbalanced dataset

Fan imbalanced d.s. accuracy measure fails.

TP → 9990
FN → 0
FP → 10
TN → 0

$\text{Accuracy} = \frac{9990+0}{9990+10+0+0} = \frac{9990}{10,000} = 99.9\%$ →

That is very high accuracy, but Classifier is very bad as it is not being able to identify a obj that belongs to negative class.

Remedy (Some alternative measures)

② False positive rate (FPR) [metric]

$= \boxed{\frac{c}{c+d} = \frac{FP}{FP+TN}}$

③ Imp True positive rate (TPR) (r) [metric] → when even we have high accuracy

$= \boxed{\frac{a}{a+b} = \frac{TP}{TP+FN}}$

→ It is also known as sensitivity measure or recall measure

④ True negative rate (TNR)

$= \boxed{\frac{d}{c+d} = \frac{TN}{FP+TN}}$

→ It is also known as specificity

⑤ Imp Precision (p)

$\boxed{\frac{a}{a+c} = \frac{TP}{TP+FP}}$

⑥ F_1 - Score (F measure)

$= \boxed{\frac{2*r*p}{r+p}} = \frac{2a}{2a+b+c}$

$= \boxed{\frac{2*TP}{2*TP+FN+FP}}$

[r = recall
p = Precision]

Assignment ①

TP 35	FP 5
FN 10	TN 30

acc, recall, precision, F_1 score?

Sub 7

① given, a d.s, using KNN we have to findout predicted class label of a test instance.

② Given d.s, using ID3 algo, construct decision tree upto particular label.

③ Describe diff. types of metrics in modern ~~to measure~~ a (accuracy, F_1-score, etc (upto)) & compute given sums

clustering

④ Normalizations formulas

⑤ PCA – Explain the steps

⑥ PCA – Numerical

⑦ Data mining

How to compute supp. & conf of ass. ru.

⑧ Describe apriori algos

⑨ given, set of transaction, find frequent item sets (By using any 1 alg (mentioned)).

⑩ findout valid association rules from min supp, min conf, threshold.

$\frac{2}{total}$ $\frac{q*r+2}{q+r}$

$\frac{2*q}{q+r+q+r+2}$

Pyq 23

h-13

3) For the following 1 dimentional data calculate the cost function of k-means after 2 iterations [18, 5, 10, 20, 4, 8, 19].

let k = 2,

x	$c_1=5$	$c_2=19$	
18	13	1	cl_2
5	0	14	cl_1
10	5	9	cl_1
20	15	1	cl_2
4	1	15	cl_1
8	3	11	cl_1
19	14	0	cl_2

1st it

$c_1 = 5$, $c_2 = 19$ (using euclidean dist.)

c_1: 5, 10, 4, 8 — c_2: 18, 20, 19

$\frac{27}{4} = 6.75$

$\therefore c_1 = 6.75$ $c_2 = 19$

2nd it

$c_1 = 6.75$, $c_2 = 19$

	$c_1=6.75$	$c_2=19$	
18	11.25	1	cl_2
5	1.75	14	cl_1
10	3.25	9	cl_1
20	13.25	1	cl_2
4	2.75	15	cl_1
8	1.25	11	cl_1
19	12.25	0	cl_2

Clusters allotment same, means centroid remains same.

~~Clusters~~ doesn't change, so need to stop centroids

∴ SE for centroid 1 = (6.75) = 9 ~~22.75~~ [sum of squared distances of each obj's to its centroid]

SE (for (2,19)) = 2

∴ T.S.E (total sum error) / cost function =

9 ~~22.75~~ + 2 = ~~24.75~~ 11

h-c

1) For the 1 dim dataset in Pn 3 of h-B perform hei hierarchical clustering (single linkage) & draw the dendogram.

	x
S_1	18
S_2	5
S_3	10
S_4	20
S_5	4
S_6	8
S_7	19

dist matrix →

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
S_1	0						
S_2	13	0					
S_3	8	5	0				
S_4	2	1	10	0			
S_5	14	1	6	16	0		
S_6	10	3	2	12	4	0	
S_7	1	14	9	1	15	11	0

$d(S_1, S_2) = \sqrt{|18-5|^2}$

$= \sqrt{(13)^2}$ (can do manhattan dist too as 1d data)

$= 13$

[Although we can calc dist not by any dist. measure]

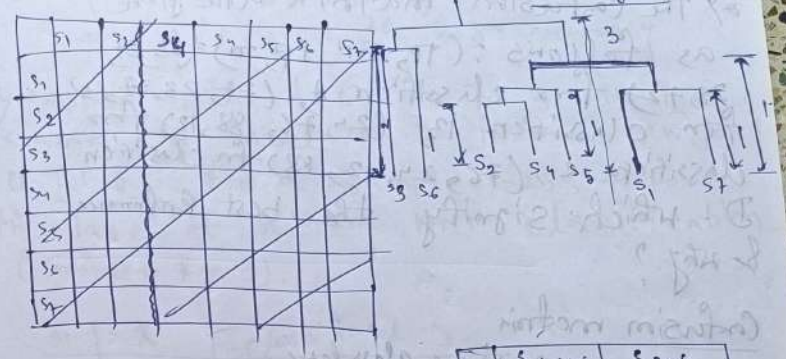
[nothing is going to be added with $(18-5)^2$ part so, there is no need of euclidean as it's an exact dist.]

hierarchical clustering (s.l.a.h.c)

⇓

A.h.c = s.l.a.h.c

dendogram



	S_1	$S_{2,4}$	S_3	S_5	S_6	S_7
S_1	0					
$S_{2,4}$	2	0				
S_3	8	5	0			
S_5	14	1	6	0		
S_6	10	3	2	4	0	
S_7	1	1	9	15	11	0

⇓

	$S_{1,7}$	$S_{2,4,5}$	S_3	S_6
$S_{1,7}$	0			
$S_{2,4,5}$	1	0		
S_3	8	5	0	
S_6	10	3	2	0

⇒

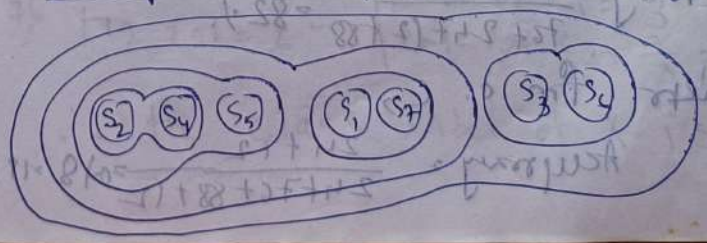
	$S_{1,7}$	$S_{2,4,5}$	S_3	S_6
$S_{1,7}$	0			
$S_{2,4,5}$	1	0		
S_3	8	5	0	
S_6	10	3	2	0

	$S_{1,7,2,4,5}$	$S_{3,6}$
$S_{1,7,2,4,5}$	0	
$S_{3,6}$	3	0

⇑

	$S_{1,7,2,4,5}$	S_3	S_6
$S_{1,7,2,4,5}$	0		
S_3	5	0	
S_6	3	2	0

⇑



5) The confusion matrix are given as follows: (TP, FN, FP, TN) = (63, 37, 28, 72) for classifier A, (77, 23, 77, 23) for classifier B, (24, 76, 88, 12) for classifier C, (76, 24, 12, 88) for classifier D. Which signify the best performance & why?

Confusion matrix

		Predicted class label	
		yes	NO
actual class label	yes	TP (a)	FN (b)
	no	FP (c)	TN (d)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{a+d}{a+b+c+d}$$

∴ for classifier A,

$$\text{Accuracy} = \frac{63 + 72}{63 + 37 + 28 + 72} = 0.675 = 67.5\%$$

for C. B,

$$\text{Accuracy} = \frac{77 + 23}{77 + 23 + 77 + 23} = \frac{(77+23)}{2(77+23)} = 0.5 = 50\%$$

for C. D,

$$\text{Accuracy} = \frac{76 + 88}{76 + 24 + 12 + 88} = 82\%$$

~~Classifier~~ for C. C,

$$\text{Accuracy} = \frac{24 + 12}{24 + 76 + 88 + 12} = 0.18 = 18\%$$

∴ classifier D signify the best performance as it gives the high accuracy.

6) Consider the d.s. in qn 3 in 9-B. Assume the label of 7 instances are: [0 1 0 1 1 0]. For a sample x=25, find out the label of x using K-NN classification (consider K = 3).

Sl	X	class label
1	18	0
2	5	1
3	10	1
4	20	0
5	4	1
6	8	1
7	19	0

X = 25

$\sqrt{(18-25)^2}$ euclid distance

1) $|18-25| = 7$
2) $|5-25| = 20$
3) $|10-25| = 15$
4) $|20-25| = 5$
5) $|4-25| = 21$
6) $|8-25| = 17$
7) $|19-25| = 6$

given, K = 3

Here i = 1, 2 ∴ $C_i = C_1, C_2$

let $C_1 = 1$, $C_2 = 0$

3 nearest neighbours are = 5, 6, 7

= {0, 0, 0} class label

Conf. of $C_1 = \frac{0}{3} = 0$

Conf. of $C_2 = \frac{3}{3} = 1$ (highest)

∴ the class label of X is C_2 or 0.