It is a type of classification techniques.

Rule based classification

It is a type of classification technique in which we create rule based from the data instead of creating ~~training~~ model. and then classify the records by using a collection of If - then rules. which is generate by using data.

If condition then conclusion

Rules in the form of.

$R = (R_1 \vee R_2 \vee \cdots \vee R_k)$

Syntax

Rule: (condition) $\longrightarrow$ y

where.

condition is a conjunctions of attributes. → applied on attributes of the data.

and y is the class/label. (i.e in which attributes or data's goes)

In this classify the data/records on the basis of collection of if then rules. and the rules is created by the condition is applied on the feature/attributes of data. and according the statisfaction of condition class is defined.

condition is conjunction of attributes. ($F_1 \wedge F_2 \wedge F_3$) etc.

and y is the class/label in which sample goes.

Rule : (Condition) $\longrightarrow$ (y)
① ②

① LHS / pre condition / rule antecedent. (something happened or existed before)

② RHS: rule consequent. (resultant, effect)

Condition is form of conjuction of different attributes while rule is form of disjunction of different rules.

Condition: $(C_1 \wedge C_2 \wedge C_3 \cdots \wedge C_n)$.

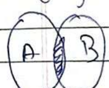
Rule: $(R_1 \vee R_2 \vee R_3 \cdots \vee R_k)$

Ex

Simple rule example ⟹ Rule: $\underbrace{(X = 10)}_{\text{Condition 1}} \wedge \underbrace{(Y = 9)}_{\text{Condition 2}} \rightarrow A$ → class.

Complex rule example ⟹ Rule: $\underbrace{(X=10)}_{\text{Condition 1}} \wedge \overbrace{(\underbrace{Y=9}_{\text{sub condition}} \wedge B) = C}^{\text{Condition 2}} \rightarrow D$. Rule 1

conjuction



$A \wedge B$

$(\wedge, \&, AND)$

T	T	T
F	F	F
F	T	F
T	F	F

disjunction

$A \vee B$

$(\vee, |, OR)$

T	T	T
F	F	F
F	T	T
T	F	T

In this we define the rule first we define the class (i.e how many classes are given)

Working → Rule → record X if the conditions of rule covers is fulfilled (i.e attributes of the record satisfy the condition).

If rules n (rule) > 0 → triggered or fired
records exist in it.

Assesment of rules.

quality of rules check
→ Coverage of rule.
→ Accuracy of rule.
Nature on original set of training data

(a) Coverage → Fraction of records that statisfy the condition antecedent of a particular rule.

$$\frac{\text{no. of records} \in \text{condition}}{\text{total records}} \times 100$$

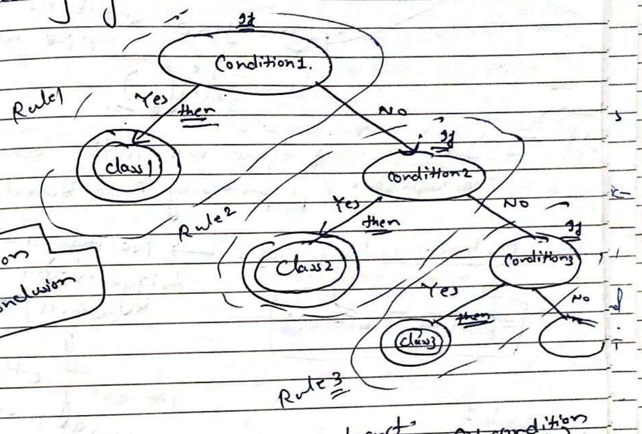
(b) Accuracy → Fraction of records that statisfy both the antecedent and consequent of a particular rule, with anterdent condition.

$$\frac{\text{no. of records} \in \text{Rule}}{\text{no. of records} \in \text{condition}} \times 100$$

Qns 1 How we create rule in the Rule based classification.

Qns 2 Is any attributes goes into more than one classes in rule based classification or not?

Qns 3 How we said classification is good, using coverage and accuracy of rule.



Dialogue
If condition
Then conclusion

Rule Antencedent → LHS part
→ pre condition (if condition part).
→ Something happened or existing before.
→ In this it is may be collection of one or more than one, different attributes as condition with conjuction / AND operator.

Rule Consequent → RHS part.
→ Resultant, Conclusion,
→ Class prediction (i.e consist attributes goes in which class).

Drawback of Rule based classifiers.

① # Multilabel classification is exist in Rule based classification. So it is issue to which class, rule, attribute is gone.

② If a attributes having no class (i.e no rule statisfy).

To Overcome this issue we notice some characterstics / properites of Rule based classification

(i) Mutually exclusive rules → No two rules are triggered by the same record / attributes.

$P(A \cap B) = 0$

$P(A \cup B) = P(A) + P(B)$

⇓

i.e every attributes / record covered by at most one rule.

Drawback 1.

If $P(A \cap B) \neq 0$ (i.e mutually exclusive not exist).

Problem

⇓

Multilabel classification occurs.

⇓

Solution 2 ways

(a) Ordered set of rules.

(b) Assiging votes for each classes depending on their weights.

⇓

Unordered set of rules.

(ii). Exhaustive Rules → In this each record is covered by atleast one rule.

$P(A) \neq 0$

Drawback 2

Problem

If $P(A) = 0$ (i.e not exhaustive rule exist)

⇓

Record not go in any class (i.e. no rule be statisfy).

Solution

Make default rule class also

Default classes ⇓ Concept Default rule concept

Used the concept of default class.

⇓

In this we can assign those records which is not statisfy / covered by any rules.

Switch case programming

Creation of rule making

We make a rule in such way that each attributes goes into any class and each attribute covered by at most one rule.

Mutually exclusive $P(A \cap B) = 0$

Exhaustive $P(A), P(B) \neq 0$

How we create rule in Rule based classification technique

① By using Decision Tree we can create rule. (Indirect Method)

② By cheking the number of class in the records and make rule which contain maximum records. (Direct Method)

Rules can be simplified or modified

Means we combine ~~the~~ remove the attributes from the existing rule, then mutually exclusive and ~~ex~~ non exhaustive problem occurs.

So we also simplified the rule also.

(a) Ordered ~~Rule~~ Set Rule → Concept of ordered / priority comes in this.

Ranking ~~ordering~~ the rules.

Rank the rules → We set ordered or give the priority to the rules (known as Decision list).

High priority rule is taken first.

Approach ordering → Rule-based ordering. → Class-based Ordering.

(i) Rule-based ordering → Ranked the rules on the based of quality.

Expert advices, quality of attributes, Coverage, Accuracy

(ii) class-based ordering → Rule that belong to the same class appear together.

class wise ordering, i.e. comes together.

(i.e one class rule one side and other class rule other side).

(b) Unordered Set Rule → Concept of voting scheme occurs. Size ordering rule

High Voting count will be prefer.

In this we can assign votes for each classes depending ~~of the~~ on their weights / volume. (i.e no. of attributes count in the classes).

Implement Rules or Rule based classification Method → Direct method, Indirect Method

Direct method: No intermediate occur. Rules extract directly from data. algorithm (ex IR, RIPPER (N2))

Indirect Method: Extract rules from other classification models, (Decision tree, Neural Networks, Random forest etc) → (4.5 rules algorithms)

(a) Direct Method → Extracting rules directly from the data's (i.e no intermediate source occur)

eg. → RIPPER, CN2, 1R Holte's
Algorithms → 1R, Sequential Covering Algorithm.

(b) Indirect Method → Extracting rules from other classification models (eg decision tree, neural networks etc)

Algorithms → (C4.5 rules)

Input → (D, A)
output → If then Rules

(c) Sequential Covering Algorithm → Sequentially covering the data's by the classes

(i) Create Rule set where all rule be defined.
R = { };

(ii) for each class C do
Rule = (Learn One Rule) (D, A, C);

Learn the rule

1R

remove data's covered by Rule according to attribute (A)

Due to this there is drawback of that algorithm because it may not statisfied the properties of rule based classification

It is suitable for current data (i.e already know the prediction

Update rule set.
(R = R + Rule();)

until terminating condition → condition is set by users
→ if Accuracy < Required level Accuracy then stop.

(iii) end for()
(iv) Return the R.

→ It is most widely used algorithm in this classification
→ In this algorithm define rules cover the maximum data records of one class and no data records of other classes at one time

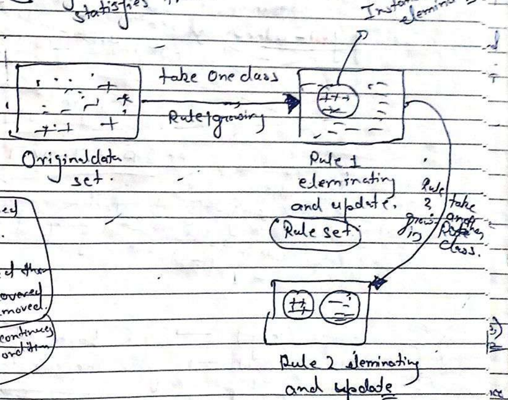
⇓

rules are grown sequentially (i.e class assign sequentially) at one time.

⇓

After rule is grown all data set which statisfies that rule are removed.

repeat the process until terminating condition not occur.



Rules are learned one at a time. For each time rules are learned then a records are covered by the rule is removed and the process continues until terminating condition not occured

Learn One Rule → Also called 1R algorithm.
→ Simplest algorithm rule.
→ It follows the general to specific. Search. (Beam search) approach.

Learn_One_Rule (D, Attributes, C)

1. Consider one class.
2. Pass through training data.
3. Find attribute that increases accuracy of current set. (It may be empty also)
4. Append the attribute into the rule

Learn sets of first-order rules → It contains attributes called first-order Horn clauses.

at most one +ve variable.

eg: $\boxed{A \lor \neg B \lor \neg R}$

Interpreted as program in the logic programming language (PROLOG), so it is called ILP (Inductive logic programming).

eg Parent (x, y) → y is parent of x.

grand parent: Ancestor (x, y) → y is ancestor of x.

Rules { If Parent (x, y) then Ancestor (x, y)

If Parent (x, z) ^ Ancestor (z, y) then Ancestor (x, y) }

Don't remove by rule (i.e remove attributes). ~~Application, ...~~ → Reason: Rule may perform well on the training data but less on test data. (over fitting case)

Rule purning → To cut somethings.

In this we remove / cut the irrelevant rule from the rule sets.

classifiers:- R_1, R_2, R_3

Rules eg R_1, R_2, R_3, R_4, R_5

cut.

⇓

Remove the rule which not affect the overall accuracy of the rule's $\boxed{\text{quality } (R_p > R)}$

C4.5 ~~and~~ algorithms. best example of this rule purning.

Measure other for rule quality

Likelyhood ratio

1. Information Gain, Entropy ~~also~~, Gain measure also used to check quality, measure accuracy.

① FOIL ⟹ $\text{Gain} = pos' \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right)$

It is an algorithm ~~...~~

Where.

pos → no. of positive record covered by a particular rule.

neg → no. of negative record coverd by a particular rule.

pos' → no. of positive record coverd by a rules excepts that particular rule.

neg' → No. of negative record covered by a rules excepts that particular rule.

② Likelyhood Test (Ratio) → $\text{Ratio} = 2 \sum_{i=1}^{n} f_i \log\left(\frac{f_i}{e_i}\right)$

where f_i = observed distribution things among classes.
e_i = expected distribution things among classes.

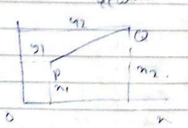
Real World application
→ Credit scoring.
→ Predictive Maintenance
→ Spam filtering.
→ Quality Control

Nearest Neighbour classifiers
→ classify based on similarity (distance)
→ It is also called K-nearest Neighbour classifiers (i.e k^{th} nearest neighbours of a particular data from data)
→ It is used for both types of supervised learning algorithm (regression and classification)
→ It is a type of distance algorithm.
→ It is basic algorithm for the classification

(Nearest Neighbour) → In this we find the distance between ~~the points and make a classes.~~ target point and all others point and make a class according to it.
Mainly used euclidean distance

find the most nearest point(s) records from each classes by distance

$$d_{(PQ)} = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$



Important to select ~~records~~ k nearest neighbour

k may be 1, 2, 3 - - - but mainly k be take 5.
→ defines no. of values/records near to that point.

major challenge to tune the value of k, but mainly value of k = 5 be prefer

if the value of k > 1 then it means a point have more than one neighbour is selected.
so, in that case we take ~~...~~ count no of nearest records in which class is maximum
⇓
(Majority voting case)

Ans.

Movie	ImDb rating	Duration	Genre
MI	8.0	160	Action
Gadar2	6.2	170	Action
Rocky & Rani	7.2	168	comedy
OMG2	8.0	155	Comedy

Barbie → 7.4 → 114 → ??

(1) ~~Let we take~~
here we find the euclidean distance b/w barbie and all other's movie to check ~~the~~ for nearest process

euclidean distance

(Barbie – MI) → 46.00
(Barbie – Gadar2) → 56.01
(Barbie – Rocky & Rani) → 54.00
(Barbie → OMG2) → 44.00

Now, important thing is the value of k. (i.e how many nearest value, is taken) ~~for the classify each classes~~

if k = 1 (i.e near 1 nearest value among them)
So, Barbie → OMG 2 → having nearest value.
So it goes into comedy class.

if k = 3 (3 nearest value among them)

from this we get 2 comedy class and 1 a.t. class, So according to majority voting we take that value into comedy class.

Working — 1) In this we find the distance of that points with all the records and then we find the value of k (mainly it will be k = 5) after this we select the k nearest values / records among all the records occur on the basis of distance.

⇓

After this we check the voting no. of nearest records is maximum in which class

⇓

That record goes into that class.

In this find K nearest data points from that data point and check maximum data points in which class then that value goes into that point.

If distance is same and it is comes under nearest criteria, so that both record is consider.

~~Me~~ There is no specific method for find k value, it may be (trial and error method)

⇓

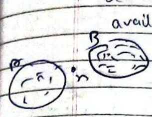
choose k value not high & not low (i.e select which gives the minimum error rate of prediction)

K(small) ⟶ Lead to be noisy and effect outliers

K(large) ⟶ find some difficulties

KNN also works for n dimensional features

KNN algorithm assumes the similarity between new data points with an available data point and put the new data point into the category that is most similar to the available categories.



⇓

KNN algorithm stores all the available data and classifies a new data point based on the similarity (most similar class)

1 — It is a non-parametric algorithm

↓

not make any assumption about the parameter

4 It is also a lazy learner algorithm.

it doesn't learn from the training data set immeditely.

⇓

At training phase it just store the dataset and when it gets new data, then it do perform action on dataset for classification.

It

Teacher's Signature

B.

\# Similarity between dataset by their discriptive statistics, distances ~~etc~~, Jaccard similarity coffcient.

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

distribution → It is a statistics function that shows the possible values for a variable, and how often they occur.

→ probability distribution

↓

function that gives the probability of occurance of different possible outcomes for an expirement.

Learning Algorithm → Lazy Learn
→ Eager Learn

(a) Lazy learn → It doesn't process training data unit it needs to make a prediction

⇓

It store the dataset in the training phase and when a new data set is comes then it comes into action and perform classification / prediction

⇓

wait & encounter type learing

prediction done in very lazy manner

Learn in testing phase

(b) Eager Learn → It is also known as non-lazy Algorithm.

Learn in training phase.

→ It process during the training phase and make a model for prediction

⇓

In this when training dataset is given then this type of algorithm build a model by that training dataset and use this model to make prediction during the testing phase

	Lazy	Eager
Traing speed	Fast (datastore)	Slow (model creation)
Testing speed	Learn (testing phase)	Learn in traing phase.
Pre-calculated Algorithm.	Absent	Present
Example	Bayesian Rule, KNN, Local regression, Instance-based Learning	Example. SVM, regression, Decision Tree.

(i) Select the value of K.

(ii) find the eclidian distance from new point with all the points

(iii) Select the points which is K nearest

(iv) Among these k neighbours, count no. of data point for each category
(v). Assign the new points to category for which, the number of neighbour is maximum

While in case of regression it return a real-valued prediction for a given data.

⇓

In this case classifier returns the average value of the real-valued labels associated with the k-nearest neighbours of the that data.

If variables is categorical, then distance be fixed as.

x_1 — red category
x_2 — red category $d = 0$

x_1 → blue category
x_2 → red category $d = 1$

Similarity Measures
→ Descriptive statistics
→ Correlation → linear dependency
→ Distance based similarity.
→ Jacobian Similarity. (measure dissimilarity)
→

Teacher's Signature ....

Measure of Similarity & Dissimilarity in Data Mining

Data → Collection of data objects and their attributes.

object → Real entity from which we make a conclusion.

attributes → It is a property or characteristic of an object.
→ Also known as variable, field, characteristic or features.

object → A ~~do~~ collection of attributes describe an object.
→ Also called record, entity, case, sample, instance etc.

Attributes Values → Are symbols or numbers assigned to an attributes

ex

Attributes	Attributes value
Height	→ feet
	→ meters.

ex ID & Age → integers.

Data similarity and dissimilarity both are important measures in data mining → helps in identifying patterns and trends in datasets, relationship in large datasets.

⇓

It is used to quantify the degree of proximity between two data points / objects. by mathematical function called similarity or dissimilarity ~~based~~ measure. — Both similarity & dissimilarity.

Teacher's Signature

Similarity → Used to determine how similar two datasets / objects / data points are.

⇓

Numerical measure how alike two data points are.

⇓

A mathematical function that quantifies the degree of similarity between two datapoints / two datasets / objects.

Data point 1 → Methods for similarity → Similarity
Data point 2

(Input) (Output)

↑ Score values ⟺ ↑ similarity ⟸ range [0, 1]
- 0 → highly dissimilar
- 1 → highly similar

or vice-versa

• → Methods
- → Cosine Similarity
- → Jaccard Similarity
- → Pearson correlation coefficient
- → Sørensen–Dice distance
- → Chi-square
- → Spearman

Proximity are a score (non-negative number) and may range from 0 (highly dissimilar) to some finite / infinite value (highly similar).

Types of proximity
- (1) Distance based
  - → Euclidean → SED
  - → Manhattan → Canberra
  - → Chebychev
  - → Minkowski
  - → Hamming
  - → Levenshtein
  - → Mahalanobis

(2) Cosine Similarity, (3) Jaccard Similarity, [illegible], (5) Pearson correlation coefficient, (6) Sørensen–Dice distance, [illegible], (8) [illegible], (9) Spearman (10) Chi-square.

• → It is generally used to identify duplicate records, more likely data points etc.

• → It is mainly used for clustering, identifying data-pre processing techniques, etc., classification, anomaly detection, outliers etc

→ properties of similarity

(i) $sim(A, B) = 1$ (highly similarity) if $A = B$.

(ii) $sim(A, B) = 0$ (highly dissimilarity) if $A \neq B$.

(iii) $0 \leq sim(A, B) \leq 1$

(iv) Symmetric in nature. (i.e $sim(A, B) = sim(B, A)$)

(v) $sim(A, B) > 0$ (for two distinct object)

Dissimilarity → opposite of similarity measure.
→ Used to determine how dissimilar two datasets / objects / data points are.

⇓

numerical measure how dislike two datapoints are.

⇓

A mathematical function that quantifies the degree of dissimilarity between two data points / two datasets / objects

Data point 1 → Method for dissimalarity → Dissimalarity (output)

Data point 2

(Input)

range [0, 1] → { 0 → highly similar; 1 → highly dissimilar } or vice-versa

↑ values ⇒ ↑ dissimalrity

→ Methods → Distance based method / measure.

→ It is generally used to identify distanct records, identify clustering, classification, outliers, anomalies detection.

→ properties of dissimilarties

(i) dissim (A, B) = 0 (high dissimalrity) if A = B

(ii) dissim (A, B) = 1 (high dissimilarity) if $A \neq B$.

(iii) $0 \leq$ dissim (A, B) ≤ 1

(iv) Symmetric in nature (i.e dissim (A, B) = dissim (B, A))

(v) dissim (A, B) ≥ 0 (for two distanct object)

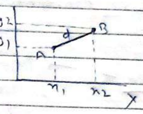
(ii) Distance measure → It is used for both similarity and (dissimilarity)

→ It is used to find distance between two data points

→ types → Manhattan, Euclidean, SED, Canberra, Minwokei, Hamming, Chebyschev, Mahalunobis, Levenshtein.

(a) Euclidean distance → Also called L2 norm

→ It is square root of the sum of the squared differences between their corresponding coordinates. (Smallest distance)



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

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$\Updownarrow$

$$d(X, Y) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2}$$

→ Commonly used for clustering, KNN classification, anomaly detection (outliers)

→ Highly sensitive on scale of features, outliers while dealing with high-dimensional data. (Not well dealing with high-dimensional data)

→ easy to visualize & interpretate.

(b) SED → It is squared of euclidean distance

$$d^2 = \sum_{j=1}^{n} (x_j - y_j)^2$$

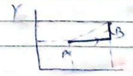
$\Updownarrow$

$$d = \sum_{j=1}^{n} (x_2 - x_1)^2 + (y_2 - y_1)^2$$

(c) Manhattan distance → Also called L1 Norm / City Block / taxicab distance)

→ It is sum of absoulte difference between their corresponding coordinates.

$$d = |x_2 - x_1| + |y_2 - y_1|$$



$$d(X, Y) = \sum_{j=1}^{n} |x_j - y_j|$$

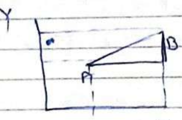
→ Mainly deal with high-dimensional dataset.
→ Commonly used in clustering, & KNN classification, anomality detection, outliers

→ Less sensitive in comparsion to eculidean distance.
→ Not suitable for non-linear relationship between features.

(d) Chebyshev distance → Also called $L\infty$ norm / maximum distance (metric)

→ Measure the maximum distance between the quardinates.

Used in stock prediction, cryptocurrences. etc

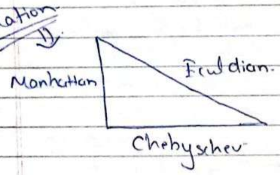


$d(X,Y) = \max(|X_j - Y_j|)$

⇓

$d = \max(|x_2 - x_1|, |Y_2 - Y_1|)$

Relation ⇓



in this min metric is not used because if $\min(0, 3) = 0$ but $A \neq B$ ⇓ $A = B$.

(e) Canberra distance → Weighted version of manhattan distance.

→ Commonly used for clustering (Fuzzy), classification, computer security, spam detection.

→ More robust to outliers.

$|x_2 - x_1| + |y_2 - y_1|$ ⟺ $d(X,Y) = \sum_{j=1}^{n} \frac{|x_j - y_j|}{|x_j + y_j|}$

(f) Minkowski distance → Also called L_p norm.
→ Generalised version of Eculiedan, Manhattan and Chebyshev distance.

→ here we take a hyperparameter be p. (degree of minkowski)

$$d(X,Y) = \left(\sum_{j=1}^{n} |X_j - Y_j|^p\right)^{1/p}$$

case of p:
- $p = 1$ → Manhattan
- $p = 2$ → Eculidean.
- $p = \infty$ → chebyshev distance.

($p = 0$??) asked by students

(g) Hamming distance → Used to check dissimilarity between two strings of equal length.

⇓ define the number of positions at which the corresponding symbols in two string are different.

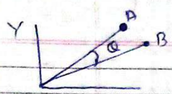
looks like manhattan in the context of categorical data.

ex
1 0 1 0 1 0
1 1 1 0 0 0
distance be (2).

→ Commonly used in error-correcting codes, cryptography, compare categorical or binary data.

(h) Cosine Similarity → ~~Measure~~ is used for both (similarity) and dissimilarity

→ Measure cosine of the angle b/w two non-zero vectors in high-dimensional space.



Measure two vectors are pointing in roughly the same direction or not.

$$\cos Q = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

range $[0, 1] \Rightarrow \begin{cases} 0 \rightarrow \text{highly dissimilar} \\ 1 \rightarrow \text{highly similar} \end{cases}$

$\boxed{\text{Cosine distance} = 1 - \text{Cosine similarity}}$

→ Commonly used in NLP, text analysis, information retrieval, text mining, document similarities etc.

(iii) Jaccard coefficient

→ Used to measure similarity and dissimilarity?

→ Also called Tanimoto distance.

→ Used to measure the similarity b/w two sets of data

Commonly used in clustering, text analysis, recommended system.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$\boxed{\text{Jaccard distance} = 1 - \text{Jaccard Coefficient}}$

range $[0, 1] = \begin{cases} 0 \rightarrow \text{highly dissimilar} \\ 1 \rightarrow \text{highly similar} \end{cases}$

(iv) Sorensen-Dice distance coefficient

→ Commonly used for text, image analysis.

→ Used to measure similarity and dissimilarity.

Dice similarity

→ Used to measure the similarity b/w two sets of data.

$$S(A, B) = 2 \times \frac{|A \cap B|}{|A| + |B|}$$

$\boxed{D = 1 - S(A, B)}$

→ not statisfies $\triangle$ inequalility condition

(v) Pearson correlation coefficient

→ Used to measure similarity and dissimilarity

→ Measure correlation b/w two continous variable.

→ Commonly used in feature selection, regression analysis.

→ Determine strength of similarity between two variables.

$$\rho(x, Y) = \frac{Cov(x\, Y)}{\sigma_x \sigma_y}$$

range $[-1, 1]$ $\begin{cases} -1 \rightarrow \text{highly -ve correlated} \\ 0 \rightarrow \text{highly dissimilar} \\ +1 \rightarrow \text{highly correleated / highly similear} \end{cases}$

$\boxed{D = 1 - \rho(x Y)}$

1(vi) Spearman correlation coefficient → Used for both categorical or numerical attributes.
→ Used when both attributes are rank-ordered.

$\boxed{D = 1 - \rho(x,y)}$

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} (\hat{x}_i - \hat{y}_i)^2}{n(n^2-1)}$$

$\hat{x}_i$ → rank of x_i
$\hat{y}_i$ → rank of y_i
n → no. of pairs

1(vii) Chi-square → Used for both similarity and dissimilarities
→ Commonly used in computer vision. (feature analysis & feature extraction)
→ Used to measures similarity b/w 2 feature matrices.

data structure in measure of proximity
→ data matrix (store data objects) → 2-mode matrix (objects, Attributes)
→ dissimilarity matrix → used to stores dissimilarity values for pairs of objects
One-mode matrix (dissimilarities)

attributes

data matrix ⇒
$$\begin{bmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$
→ object

data object

n objects described by p attributes.

n×p (n data points with p dimension)

⇓ Object by attribute structure

⇕ transform before apply algorithm

Symmetric matrix

dissimilarity matrix ⇒
$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & & \\ \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & 0 \end{bmatrix}$$
⇒ $d(i,j) \geq 0$
$d(i,j) = d(j,i)$

where $d(i,j)$ → measure dissimilarity / difference b/w two data object / data points.

⇓ object by object structure

As we know that measure of similarity & dissimilarity is range of 0 to 1.

∴ $\boxed{Sim(A,B) = 1 - Dissim(A,B)}$

Data Type / Attribute Type
- Quantitative
  - Discrete
    - Binary
    - Nominal
    - Ordinal
  - Continous
    - Interval
    - Ratio
- Qualitative
  - Binary
  - Nominal
  - Ordinal

Quantitative → Numerical Value. (measurable)

Qualitative → Categorical Value. (descriptive / interpretable)

Discrete → Data typically show information for a particular event
→ cannot divided into smaller parts / interval

whole number, integers

Continous → Data shows trends in data over time
→ divided into smaller parts / interval.

Binary → Attribute with only 2 states / outcoms (0, 1)
→ Symmetric and Asymmetric

Symmetric ⇓ both outcomes equally important
ex Gender

Asymmetric ⇓ both outcomes equally not important
ex HIV

Nominal → Groups variables into categories / classes that not overlap
→ Not required rank

Ordinal → Group variable into categories.
→ ranking exist in this

Interval → In this each point is placed at equal distance from one another.
→ No true zero point

Ratio → Same property as Interval.
→ Have true zero points.

types with attributes values properties

- Nominal → distinctness $(=, \neq)$
- Ordinal → distinctness $(=, \neq / <>)$ / order
- Interval → distinctness / order / addition $(=, \neq / <> / +, -)$
- Ratio → All 4 $(=, \neq / <> / +, - / * /)$

Proximaty Measure → Quantitative Datatypes → Distance measure, correlation, chi-square.
Proximaty Measure → Qualitative Datatypes → chi-square, correlation, Jaccard coefficient, Sorensen-dice coefficient.

Proximity measure for nominal data? → Generalised of a binary ~~attributes~~

Let we take a nominal ~~attribute~~ in which number of (cases) attributes are M.

⇓

So dissimilarities is computed based on the ratio of mismatches

$$d(i,j) = \frac{p-m}{p}$$

where m be the number of matches (i.e. the number of attributes for which i and j are in the same state) where p is the total number of attributes describing the objects.

Disha Prasad ? Manya Singh

→ weights can be assigned to increase the effect of m. or to assign greater weight to the matches in attributes having a larger number of states.

Ans

Id	test-1
1	A
2	B
3	C
4	A

here case be 4 so 4×4 matrix is form.

$$d(i,j) = \begin{bmatrix} 0 & d(1,2) & d(1,3) & d(1,4) \\ d(2,1) & 0 & d(2,3) & d(2,4) \\ d(3,1) & d(3,2) & 0 & d(3,4) \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

here total number of attributes (for nominal) = 1

$\therefore \; p = 1$.

Now,

$$d(i,j) = \frac{p-m}{p} = \frac{1-m}{1} = 1-m. \quad \text{—(i)}$$

according to equation (i),

$$d(i,j) = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$ Ans

$$\text{Sim}(i,j) = 1 - d(i,j)$$

Proximity Measure for Binary Attributes → Binary → Symmetric, Asymmetric

→ It have two states (0, 1)

attributes → absent, present.

where ~~PQ~~ q, r, s, t are no of attributes according condition satisfies

		Object j		
		1	0	sum
Object i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

where $p = q+r+s+t$

(a) Symmetric Binary dissimilarity

$$d(i,j) = \frac{r+s}{p} = \frac{r+s}{r+s+t+q}$$

$$\text{sim}(i,j) = \frac{t+q}{p}$$

$$\frac{p-m}{p} = \frac{r+s+t+q-(t+q)}{r+s+t+q}$$

$$= \frac{r+s+\cancel{t}+\cancel{q}-\cancel{t}-\cancel{q}}{r+s+t+q}$$

$$= \frac{r+s}{r+s+t+q}$$ Proved.

(b) Assymmetric binary dissimilarity

negative matches ignored

$$d(i,j) = \frac{r+s}{q+r+s}$$

$$\text{Sim}(i,j) = \frac{q}{q+r+s} \qquad \frac{A \cap B}{|A \cup B|}$$ Jaccard coefficient

When both type of ~~symmetric~~ binary attribute case present in the same dataset then we used mixed attribute approaches.

Symmetric — Asymmetric

Ans

name.	Gender	fever	Cough	test.
Jack	M	Y	N	P
Jim	M	Y	Y	N
Mary	F	Y	N	P

for Asymmetric case.
take 1 for Y / P.
take 0 for N

Now,

$$d(i,j) = \begin{bmatrix} 0 & d(1,2) & d(1,3) \\ d(2,1) & 0 & d(2,3) \\ d(3,1) & d(3,2) & 0 \end{bmatrix}$$

$$d(2,1) = \frac{1+1}{1+1+1} = \frac{2}{3} = 0.667$$

$$d(3,1) = \frac{0+0}{1+0+0} = \frac{0}{0} = 0$$

$$d(3,2) = \frac{1+1}{1+1+1} = \frac{2}{3} = 0.667$$

Proximity Measures for ordinal attributes → Let M represent the number of possible states that an ordinal attribute can have.
rank $(1, 2, 3, 4, \ldots, M_f)$

where f is no. of attributes

(1) x_{if} value $\xrightarrow{\text{replace}}$ $r_{if} \in (1 \ldots M_f)$

(2) $$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

(3) use euclidean distance

Ans

Object	test-2 (ordinal)
1	excellent.
2	fair
3	good
4	excellent.

here state be 3
$\therefore M_f = 3$

replace the value of test-2 by its rank.
Suppose excellent (3), fair (1) and good (2)

Object	test-2	z_{if}
1	3	1
2	1	0
3	2	0.5
4	3	1

$$d_{ij} = \begin{bmatrix} 0 & d(1,2) & d(1,3) & d(1,4) \\ d(2,1) & 0 & d(2,3) & d(2,4) \\ d(3,1) & d(3,2) & 0 & d(3,4) \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1 & 0.5 & 0 \end{bmatrix}$$

Ans

Dissimilarity for Mixed type Attributes

Approaches
(1) each attributes of same type are group together and perform separately
(2) process all attribute types together, (performing a single analysis)

One such technique combines the different attributes into a single dissimilarity matrix, bringing all the meaningful attributes onto a common scale [0,1]

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

indicator

where, p attributes of mixed type

indicator = 0 if x_{if} or x_{jf} is missing

$x_{if} = x_{jf} = 0$

otherwise indicator = 1

Case 1 { f is numerical → $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$

where h runs over all non missing objects/data for attributes. }

max value

min value

Case 2 { f is nominal/binary → $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$

otherwise $d_{ij}^{(f)} = 1$ }

Case 3 { f is ordinal → compute the r_{if} and z_{if}

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Batch Learning → is a type of machine learning where the model is trained on the entire dataset at once.

Algorithm processes the entire training dataset, computes the gradients and updates the models parameter in a single step.

Algorithm uses the entire training dataset to update the model parameters.

Also called offline learning

(i.e. model is trained on all available data before any updates are made)

→ efficient computationally when the dataset is not to large and fits into memory.

→ Training on the entire dataset can lead to a stable convergence of the model parameters.

updates are made in the direction that minimizes the overall error across the entire dataset.

→ Since batch learning requires the entire dataset to be present, it is often used in offline scenarios where the model is trained periodically on a fixed dataset.

Incremental Learning

- Also called Online learning.
- In this model learns from new data incrementally over time.
  ||
  updating its knowledge without retraining on the entire dataset.
- More suitable than batch learning.
- It is particularly useful in situations where the data distribution changes over time and when new examples are continuously added to the dataset.
- It allows the model to adapt and learn from the new data without discarding the knowledge gained from previous data.
- Highly efficeent than batch learning for large datasets.
- Use for online recommendation system, fraud detection, NLP etc.

Batch-Incremental Learning

- Model is periodically updated with batches of new data.
- It is more computationally efficient than updating the model after every single example.

Transfer learning

- Training a model on a source task and then fine-tuning it on a target task.

Teacher's Signature

Difference b/w batch learning and Incremental learning

1. handling training data (training process)
2. Computational Requirements
3. Memory Usage.
4. Adaptability to change.
5. Application Scenarios.

(1) In batch learning model updated until it has been the entire dataset while in Incremental learning model is updated continously or sequentially as new data becomes aviable.

(2) In batch learning more computational resources required because in this learning through entire datasets while in incremental learning it is more efficient in comparison to batch learning

(3) ~~Memory Usage~~ In batch learning more memory required to storing the data while in incremental learning it need less memory.

(4) In batch learning as we know that ~~data~~ model update using entire dataset at once so it may be less adaptable to changes in the data distribution while in incremental learning it can adapt to changes more easily, as model is continuously updated with new information.

(5) Application
- Batch → Commonly used when the entire dataset is available at once and can be processed in a batch.
- Incremental → Used when new data arrives over time and the model needs to adapt to such ~~the~~ changes.

Bayes Minimum risk classification

→ It is a classification algorithms

→ It is a decision-making framework used in the field of statistical decision theory.

→ classify objects or events into different categories based on avaliable data. and minimizing the expected risk associated with missclassification

→ It is basically a classification technique that involves the use of the Bayes theorem which is used to find the conditional probablities

Bayes theorem. $P(A/B) = \frac{P(A \cap B)}{P(B)}$

by using the chain rule

$P(A \cap B) = P(A/B)\, P(B)$ — (i)

Similarly

$P(B/A) = \frac{P(A,B)}{P(A)}$

$P(A,B) = P(B/A)\, P(A)$ — (ii).

from (i) and (ii)

$P(A/B)\, P(B) = P(B/A) P(A)$

Bayes theorem. $\boxed{P(A/B) = P(B/A)\, P(A) / P(B)}$ — (iii)

→ Prior probability → Also known as before the training process

→ Represent how likely is each class is going to occur.

→ Conditional probability $P(x/A)$ → Also known as likelihood

→ probability of how likely a feature x occurs given that it belongs to the particular class

→ Decision Rule → Rule that assigns each observation or event to a specific category or class.

→ Rule is based on the principle of minimizing the expected risk associated with the classification decision.

→ Risk function → Measure of a cost or loss associated with making a particular classification decision when the true state of nature is known.

→ Risk of misclassification is often expressed as a numerical value (FP, FN).

→ Posterior Probability → Updated probability of a particular class after considering the avaliable data or evidence

\# It is a decision-making approach that takes into account the costs associated with different types of classification errors.

Decision / True State

eg Email → Spam → D1 (S)
→ Non Spam → D2 (NS)

Let
Cost of misclassifying spam as non-spam.
(false negative cost)
$C(S/D_2) = 5$

Let
Cost of misclassifying non-spam as spam.
(false positive)

Let
$C(NS/D_1) = 1$.

$P(S) = 0.2$ $P(NS) = 0.8$
as $P(S) + P(NS) = 1$

Bayesian minimum risk classification involves calculating the expected cost for each decision and choosing the decision with the minimum expected cost.

for D1.
$E(C(D1)) = P(S).C(S/D_1) + P(NS).C(NS/D_1)$
$= 0.2.0 + (0.8.1) = 0.8$

for D2
$E(C(D2)) = P(S).C(S/D_2) + P(NS).C(NS/D2)$
$= 0.2.5 + (0.8.0) = 1.0$

here $E(C(D1)) < E(C(D2))$

So, we classify an incoming email as spam to minimize the expected cost.

Discriminant function → Mathematical function used in statistical classification and pattern recognition.

Used for classification

purpose is to discriminate between different classes / categories based on input features.
⇓
Goal is to find a function that maximally separates the data points of different classes.

takes input features and assigns a class label to the data point based on the computed value.

$x \rightarrow$ Discriminant function $\rightarrow y$

learn direct mapping between feature vector x and label y.

Simplest discriminant function is very similar to the linear regression.

$y = w_0 + w_1x_1 + \ldots + w_mx_m$

where m feature.

$y = w_0 + w^Tx$

label, bais, weight vector, feature vector where y is a discrete quantity

$$y = w_0 + w^T x$$

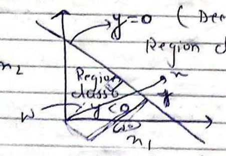
⇓

represents a hyperplane in $m-1$ dimensional space where m is the number of features.

features (m)	Discriminant function
1	$(1-1) = 0$ → Point.
2	$(2-1) = 1$ → line.
3	$(3-1) = 2$ → Plane.
4	$(4-1) = 3$ → Hyperplane in 3D space
⋮	⋮
m	$(m-1)$ → Hyperplane in $(m-1)-1$ space.

Binary class

eg for 2 feature ⇒



$y = 0$ (Decision boundary) / surface

Region class 1
$y > 0$

for w let we consider two points $x^{(A)}$ and $x^{(B)}$

$\therefore \quad y^{(A)} = w_0 + w^T x^{(A)} = 0 \quad \text{--- (i)}$

$y^{(B)} = w_0 + w^T x^{(B)} = 0 \quad \text{--- (ii)}$

from equation (i) and (ii) we get

$$y^{(A)} - y^{(B)} = w^T (x^{(A)} - x^{(B)}) = 0$$

orthogonal to every vector lying within the decision surface (it determines the orientation of the decision surface).

for w_0

$$w_0 + w^T x = 0$$

$$w^T x = -w_0$$

$$\frac{w^T x}{\|w\|} = \frac{-w_0}{\|w\|} \quad \Big\} \text{ normalize}$$

So, w_0 determine the location of decision surface.

Multiple classes

Discriminant functions build in two ways.

One-vs-rest ⇓ Build $k-1$ discriminant functions. Each discriminant function solves two classes classification problems.

eg (C_k vs not C_k)

One-vs-One ⇓ One discriminant functions per pair of classes

$\therefore$ total functions $= {}^{k}C_2 = \frac{k(k-1)}{2}$

Dimensionality Reduction

Reduction = (subtraction / minimization)

→ The no. of features/variables/columns present in a given dataset is known as dimensionality.

⇓

process to reduce these features is called dimensionality reduction.

Causes

As a feature is high there is chance of overfitting error / computational task more complicated, difficult visualization etc

⇓ (Curse of Dimensionality)

Is the process/way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that is provides similar information.

(feature extraction, feature selection)

→ Widely used in supervised learning problem.

Application of DR
→ Noise reduction
→ cluster Analysis.
→ Data Visualization.
→ bioinformatics.
→ Speech recognition.
→ Signal processing.

Mainly used for the fields that deal with high dimensional data.

Computer Vision

D R Techniques

- Feature Extraction
  ⇓
  PCA
  LDA.
  [illegible]
- Feature Selection.
- Dimensionality Reduction.
  - Components / Factors based
    ⇓
    (i) Factor Analysis
    (ii) PCA (Principle Component Analysis)
    (iii) ICA (Independent Component Analysis)
  - Projection Based.
    ⇓
    (i) ISOMAP
    (ii) t-SNE
    (iii) UMAP.
    data visualization mainly (NLP etc)

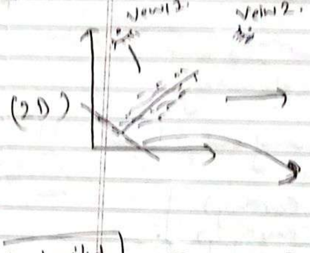
Benefits of applying DR
→ Less Computation training time.
→ Easily visualization
→ Reduce Overfitting case.
→ May remove multicollinearity if redundant features present

Disadvantages of DR
→ 1) Some of the data or feature may be lost.
→ In PCA, sometimes the principal components required to consider are unknown.

(1) Principal Component Analysis (PCA)

→ It is mainly used for overcoming overfitting.

→ In this we find the principle components.
⇓
apply views (different ways to understand the data's)



(1 D) → PC 1
(2 D) → PC 2

property 1: no. of PC ≤ no. of attributes (columns)

priority given → PC(1) > PC(2) > PC(3) > - - - - - - - PC(n)

property 2: Orthogonal property
⇓
Means two thing are independents (i.e. every principal component are independent to each other).

property 3: PC must be the linear combination of the original features.

Ans ①

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

here $\bar{X} = 1.81$
$\bar{Y} = 1.91$

Step 0 Describe dataset. (n, N) where n = no. of attribute, and N = no. of observation

Step 1 Calculate $\bar{X}$ and $\bar{Y}$ separately.

Step 2 Create covarrience matrix of ordered pair (n^2)
here 2 attribute,

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}_{2\times2}$$

where.

$$Cov(x,y) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})$$

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

$$\therefore C = \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

Step 3 find egienvalue by form $C - \lambda I = 0$
$\det(C - \lambda I = 0)$
Identity matrix

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{bmatrix} = 0$$

here $\lambda_2 > \lambda_1$

$$\lambda^2 - 1.333\lambda + 0.0630 = 0$$

$\therefore \lambda_1 = 0.0490$ and $\lambda_2 = 1.2840$.

Step 4 Compute egienvector for each eigenvalue. By using this form. $C\nu = \lambda\nu$

$$C(\nu) = \lambda\nu$$

egienvector.

for $\lambda_1 \Rightarrow$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 0.0490 \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

solve this.

$$\begin{cases} 0.6165X_1 + 0.6154Y_1 = 0.0490X_1 \\ 0.6154X_1 + 0.7165Y_1 = 0.0490Y_1 \end{cases}$$

take any one of them.

$$X_1 = -1.0845Y_1$$

put $Y_1 = 1$ for temporary basis.

$$\therefore X_1 = -1.0845$$

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} -1.0845 \\ 1 \end{bmatrix}$$

Normalize egien vector

$$= \sqrt{(-1.0845)^2 + (1)^2}$$
$$= \sqrt{1.17614 + 1}$$
$$= \sqrt{2.17614}$$
$$= 1.47517$$

Now,

$$X_1 = \frac{-1.0845}{1.47517} = -0.7351$$

$$Y_1 = \frac{1}{1.47517} = 0.6778$$

for $\lambda_2 \Rightarrow$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = 1.2840 \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix}$$

$$0.6165X_2 + 0.6154Y_2 = 1.2840X_2$$
$$0.6154X_2 + 0.7165Y_2 = 1.2840Y_2$$

Teacher's Signature

$0.6678 X_2 = 0.6154 Y_2$
$0.6154 X_2 = 0.5675 Y_2$

take any one of them

$X_2 = 0.9215 Y_2$

put $Y_2 = 1$ take as temporary basis

$\therefore X_2 = 0.9215$

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0.9215 \\ 1 \end{bmatrix}$$

$= \sqrt{(0.9215)^2 + (1)^2}$
$= \sqrt{0.8491 + 1}$
$= 1.359$

Now,

$X_2 = \frac{0.9215}{1.359} = 0.6780$

$Y_2 = \frac{1}{1.359} = 0.7358$

- It is an unsupervised algorithm. (clustering?)
- Statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of (orthogonal) transformation.

mutually exclusive

$\vec{A} \cdot \vec{B} = 0$

New transformation feature is called Principal Components (PC).

- (Most widely used in EDA and predictive modelling)

supervised learning. like dummy variable).

draw strong patterns from the given dataset by reducing the variances.

Works by considering the variance of each attribute because the high attribute shows the good split between the classes and hence it reduces the dimensionality.

Orthogonal → Variables are not correlated to each other (i.e independent to each other)

Eigenvectors → A vector that is associated with a set of linear equations.

→ Also known as latent vector / characteristic vector.

→ Non-vector in which when a given matrix is multiplied it is equal to scalar multiple of that vector.

$A_{n \times n}$ v be non-zero vector

$\therefore Av = \lambda v$

λ → eigen value

Eigenvalue → A values are generally associated with eigenvectors in Linear algebra.

Dimensionality, # Correlation, # Orthogonal, # Covariance matrix.

Step 5 ~~Sorting the eigen vectors (according to the priority condition)~~

Step 6 ~~PC matrix = product~~ eigen vector^T Matrix = values

Application PCA → Computer Vision
→ image compression.
→ Finding hidden pattern in data has high dimension (Finance, data Mining, Psychology etc)

Example:

X	Y
4	11
8	4
13	5
7	14

here n (no. of features) = 2.
N = (no. of sample) = 4.

here

$\bar{X} = 8$ and $\bar{Y} = 8.5$

Now we calculate covariance matrix of ordered pair

$$C \begin{bmatrix} x,x & x,y \\ y,x & y,y \end{bmatrix} = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Now we calculate egien value, eigen vector and normalized eigen vector.

egien value ⇒ $\det(C - \lambda I) = 0$

$$\det\left(\begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$\det\left(\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}\right) = 0$$

$$\lambda^2 - 37\lambda + 201 = 0 \Rightarrow \left(\frac{-b \pm \sqrt{b^2-4ac}}{2a}\right)$$

$\therefore \lambda_1 = 30.38 \quad \lambda_2 = 6.6151$

here $\lambda_1 > \lambda_2$
PC1 PC2.

Now, egien vector for each λ (λ_1, λ_2)

$(Cv = \lambda_1 v)$ / $((C - \lambda I)v = 0$

$$\begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}\begin{bmatrix} x_1 \\ Y_1 \end{bmatrix} = 30.38 \begin{bmatrix} x_1 \\ Y_1 \end{bmatrix}$$

$14x_1 - 11Y_1 = 30.38 x_1$
$-11x_1 + 23Y_1 = 30.38 Y_1$

$(14 - \lambda)x_1 - 11Y_1 = 0$
$-11x_1 + (23-\lambda)Y_1 = 0$

take any one of them

$(14-\lambda)x_1 = 11 Y_1$

$$x_1 = \frac{11}{14-\lambda} Y_1$$

Similarly for λ_2 we do

$$\lambda_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} \quad \lambda_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

Now derive new dataset / PC.

PC1	PC2
P_{11}	P_{21}
P_{12}	P_{22}
P_{13}	P_{23}
P_{14}	P_{24}

here for PC_1

$$P_{11} = e_1^T \begin{bmatrix} (x_1 - \bar{x}) \\ (y_1 - \bar{y}) \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$$

$P_{11} = -4.3052$

Similarly for P_{12} P_{13} P_{14} all observations

Similarly for PC2

$P_{21} = e_2^T \begin{bmatrix} x_1 - \bar{x} \\ y_1 - \bar{y} \end{bmatrix}$

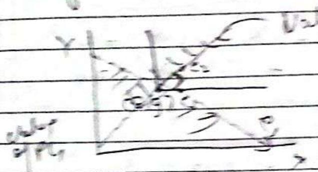
$P_{21} = \begin{bmatrix} 0.8303 & 0.5570 \end{bmatrix} \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$

$= -3.3212 + 1.3935$

$= -1.9277$

done for P_{22}, P_{23}, P_{24}

Now



② Factor Analysis → Also called latent variable models → find the common factor between the attributes hidden behind them.

Reduce large no. of attributes to a fewer number of factors ⇒ be a driving factor. // also called (common cause) (latent variable)

⇐ which columns are behaviour similarly comes in one group.

Equation be

factor analysis equation ⇒ attributes = α (factor/driving factor) + c_1

PCA equation ⇒ factor/driving factor $= \sum_{i=1}^{n} \alpha_n$ attributes where $\alpha_1 > \alpha_2 > \dots \alpha_n$

(Factor Analysis try to find hidden factors in the attributes while PCA try to explain most of data variance by few attributes)

Factor

At1 | At2 | At3 | At4 | at

eg

Columns / observed variables

Waiting time, cleanliness, staff behaviour → Service

Taste of food, Food freshness, Food temperature → Food quality

factor/driving factor

Latent Variable → Variables that are not directly observed but are inferred from other variables → factor/driving factor/common cause.

Latent Variable model → Mathematical models that aim to explain observed variables in of the latent variables

Assumption in FA

→ No outlier present in data.

→ Sample size > factor (3:5) factor : observation

→ Correlated variables (Bartlett test perform)

→ Normalised data

purpose FA
- Data Reduction
- Latent Variable Discovery.
- Dimensionality Reduction.

Types FA
- EFA (Exploratory Factor Analysis).
- CFA (Confirmatory Factor Analysis).