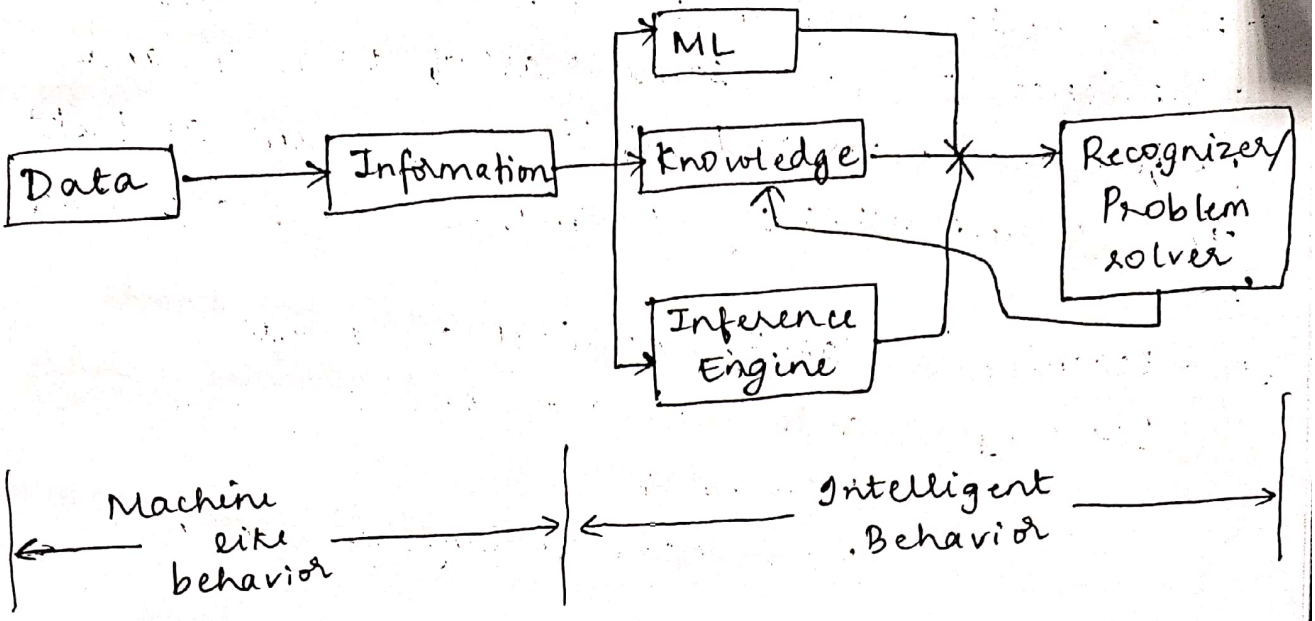
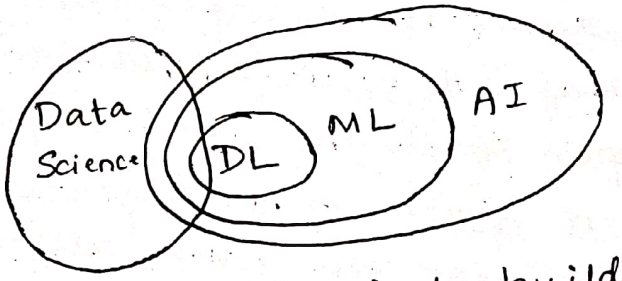
# Machine Learning Component




→ Goal of Machine Learning is to build computer systems that can learn from their experience and adopt to their environments.

→ Machine Learning is an imp aspect or component of Intelligence.

→ Intelligence is the ability to learn.

→ Well posed problem- A problem that has a unique solution that changes continuously with the initial conditions.

→ Ill-posed problem- They are typically the subject of ML methods and AI, including statistical learning. These methods do not aim to find the perfect solution; rather, they aim to find the best possible solution and/or the solution with the least errors.

## Machine Learning

→ A computer program is said to learn from experience E wrt some class of tasks T and performance measure P, if its performance at tas in T, as measured by P, improves with experie E.

ex:- Learning to classify chemical compounds
Learning to drive an autonomous vehicle
" —" — play bridge
" —" — parse natural language sentences

## Concept Learning

→ An approach of learning.
→ Inferring a Boolean-valued function from training examples of its input and output.
→ concept space - is the representation of the problem wrt the given attributes. It is defined by all combinations of values for every instance x.
→ Concept learning ← General hypothesis
Specific hypothesis

G = {'?', '?', '?', ...., '?'}
↳ No. of attributes

S = {'ϕ', 'ϕ', 'ϕ', ..., 'ϕ'}
↳ No. of attributes.

## Find-S Algorithm

→ It's the most specific hypothesis
→ Consider only +ve examples.

Algorithm -

1. Initialize h to most specific hypothesis
h = {'ϕ', 'ϕ' ..., 'ϕ'}
2. For each ~~attribute~~ +ve example:

For each attribute in the example::

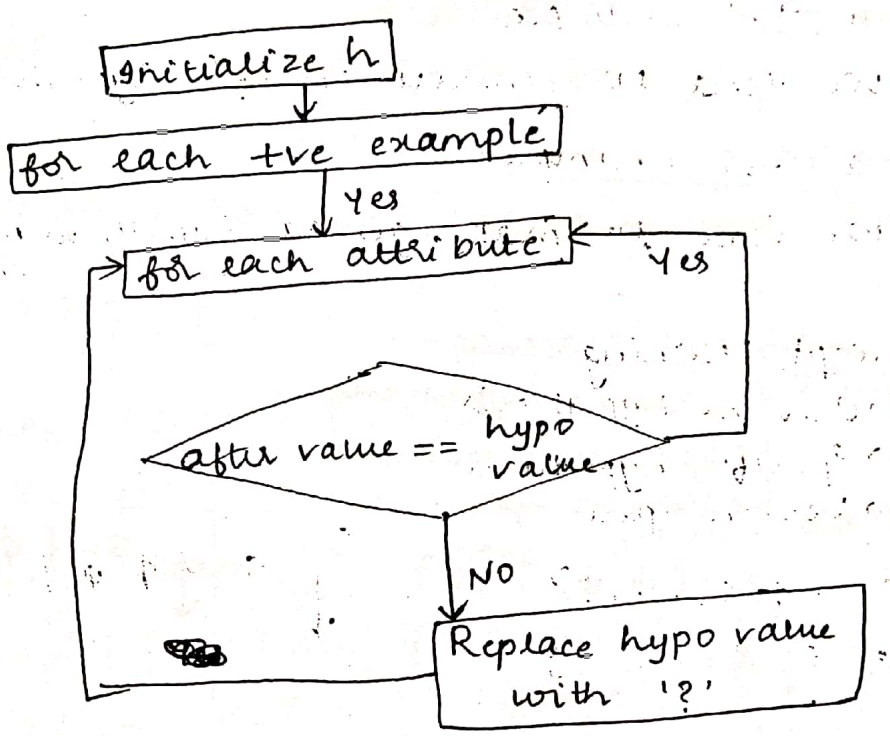
if attribute value = hypothesis value::

Do nothing

else:

Replace hypothesis value with more general constraint '?'.

Flowchart:-




Data Base:

Concept: Days on which person enjoys sport.

| Sky | Temp | Humidity | Wind | Water | Forecast | Enjoy |
|---|---|---|---|---|---|---|
| Sunny | warm | Normal | Strong | warm | Same | Yes |
| Sunny | warm | High | Strong | warm | Same | Yes |
| Rainy | cold | High | Strong | warm | change | No |
| Sunny | warm | High | Strong | Cool | Same | Yes |

Step 1:

$h_0 = \{'\phi', '\phi', '\phi', '\phi', '\phi', '\phi'\}$

$h_0 = \{'Sunny', 'warm', 'Normal', 'Strong', 'warm', 'Same'\}$

After comparison:-

$h_0 = \{'Sunny', 'warm', '?', 'Strong', 'warm', 'Same'\}$

→ 3rd record is not learnt cuz only +ve examples are considered.

## Candidate Elimination Algorithm

→ uses version space.
→ considers both +ve and -ve results
→ we have both specific and general hypothesis.

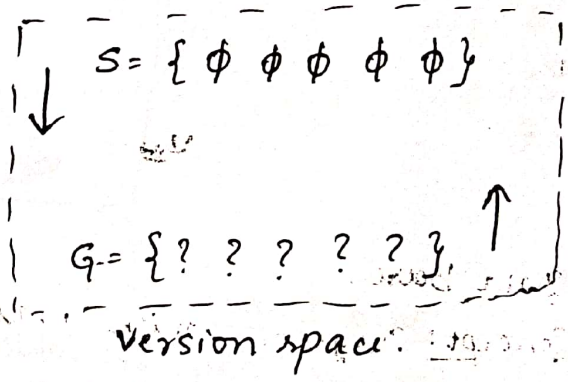
→ For a +ve example :-
We tend to generate specific hypothesis

→ For a -ve example :-
We tend to make general hypothesis more specific.

→ Concept learning -
↳ General hypotheris
↳ Specific hypothesis
↳ Version space.

$S = \{\phi, \phi, \ldots, \phi\}$

$G = \{?, ?, \ldots, ?\}$




Version space -

$S_4 = \{ < Sunny, warm, ?, Strong, ?, ? > \}$

<Sunny, ?, ?, Strong, ?, ?> <Sunny, warm, ?, ?, ?, ?> <?, warm, ? Strong, ?, ?>

$G_4 = \{ < Sunny, ?, ?, ?, ?, ? > < ? warm, ?, ?, ?, ? > \}$

Algorithm -

1. Initialize G and S as most general and specific hypothesis.
2. For each example, e :
   if e is +ve :
      Make specific hypothesis more general (Find-S)
   else :
      Make general hypothesis more specific.

Ex:

$S_0 = \{\phi\ \phi\ \phi\ \phi\ \phi\ \phi\}$ $G_0 = \{?\ ?\ ?\ ?\ ?\ ?\}$

① $S_1 = \{$'Sunny', 'warm', 'Normal', 'strong', 'warm', 'same'$\}$

$G_1 = \{?\ ?\ ?\ ?\ ?\ ?\}$

② $S_2 = \{$'Sunny', 'warm', '?', 'strong', 'warm', 'same'$\}$

$G_2 = \{?\ ?\ ?\ ?\ ?\ ?\}$

③ $S_3 = \{$'Sunny', 'warm', '?', 'strong', 'warm', 'same'$\}$

$G_3 = \{$'Sunny', ? ? ? ? ? > < ? 'warm' ? ? ? ? >
< ? ? ? ? 'same' ? > $\}$

④ $S_4 = \{$'Sunny', 'warm', '?', 'strong', ? , ? $\}$

$G_4 = \{<$Sunny ? ? ? ? > < ? warm ? ? ? ? > $\}$

## Feature Extraction and Selection

→ Feature Extraction aims to reduce the no. of features in a dataset by creating new features from the existing ones (and then discarding the original features). These new reduced set of features should then be able to summarise most of the info contained in the original set of the feature.

→ In ML, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, and generalization

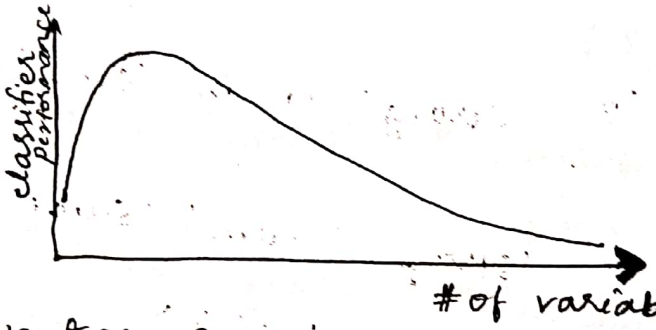
## Why Feature Extraction?

→ The process of feature extraction is useful when you need to reduce the no. of resources needed for processing without losing important relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis.

## Curse of Dimensionality:

→ No. of training examples is fixed.

→ The classifier's performance usually will degrade for a large no. of features.




advantage of Feature selection - Reduces complexity ↓, diversity ↑, Performance ↑

## Steps of Feature Selection -

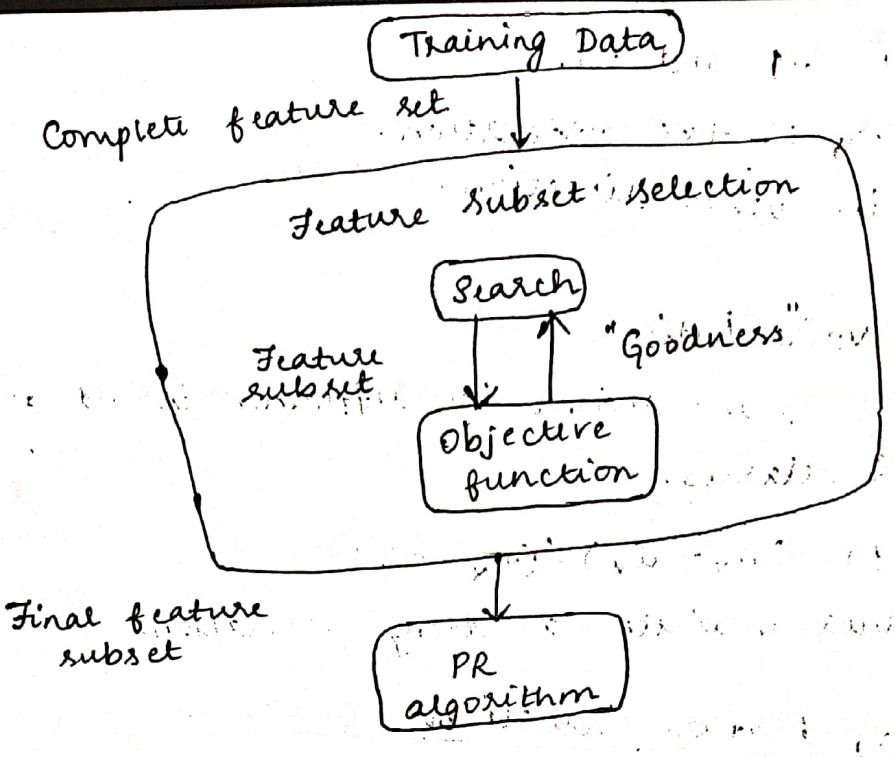
→ Feature selection is an optimization problem.

→ Step 1: Search the space of possible feature subsets.

→ Step 2: Pick the subset that is optimal or near-optimal with respect to some objective function.

→ Search strategies -
- Optimum
- Heuristic
- Randomized

→ Evaluation strategies
- Filter methods
- Wrapper methods




→ Evaluation of subsets

- supervised (wrapper method):
  - * Train using selected subset
  - * Estimate error on validation dataset
- Unsupervised (filter method):
  - * Look at i/p only
  - * Select the subset that has the most information.

Feature Selection

→ Univariate (looks at each feature independently of others).

- Pearson correlation coefficient
- F-score
- chi-square
- signal to noise ratio
- mutual information etc.

→ Univariate methods ranks features by importance.

Pearson correlation Coefficient

→ Measure the correlation b/n 2 variables.

→ Formula for Pearson correlation =

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}}$$

→ r is b/n +1 and -1.

→ +1 is perfect +ve correlation.

→ -1 is in other direction.

## Signal to Noise Ratio

→ Difference in means divided by difference in std. dev. b/n the 2 classes.

$$S2N(X,Y) = (\mu_x - \mu_y)/(\sigma_x - \sigma_y)$$

→ Large values indicate a strong correlation.

## Methods of Feature Selection

→ All the methods of feature selection involve searching for the best subset of features.

we have 2 approaches –

1. Foward approach
2. Backward approach

→ Generally, after removing a feature or adding a feature, a resulting set of features is evaluated and a decision is taken as to, whether to keep that feature or not.

→ The evaluation function is called the objective function J.

→ The evaluation usually employed is the classification accuracy obtained on a validation set of patterns

→ This can also be the classification error.

→ If E is the % error, then the objective function

$J = (100 - E)$.

## Feature Selection

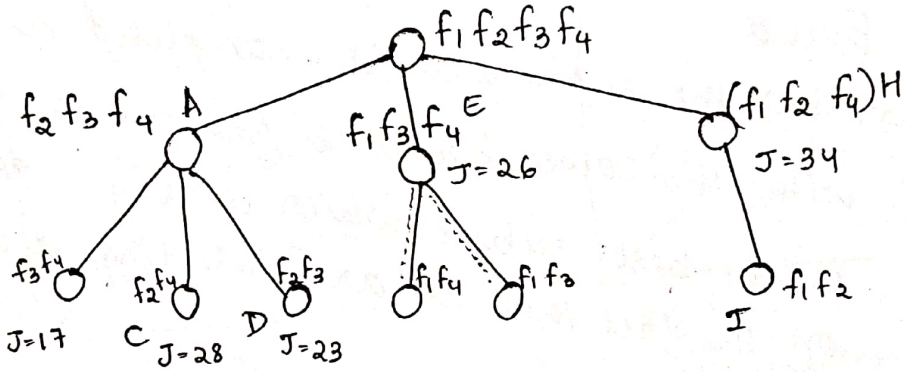
$F = \{x_1, x_2, x_3 \ldots x_N\}$

$F' \subset F = \{x_1, x_2 \ldots x_M\}$ $\quad M < N$

There are
→ irrelevant features
→ redundant feature
due to which performance comes down.
→ To address this and solve ~~of~~ the curse of dimensionality, feature selection is used.

## Branch and bound method of feature selection-

→ Consider a Training data of 4 feature use B & B method to select 2 features-




J = 100 - Error
J = Bound

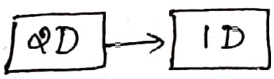
1. when the leftmost node B is reached which corresponds to the feature subset $f_3 f_4$, let the evaluation of node J be 17.
2. At this stage since this is a leaf node, the bound B = 17.
3. The next node generated is C having a j value of 28.
4. Since this is greater than the current value of B, the bound B is updated to 28 and the best subset so far is $f_2 f_4$.
5. The next node generated is D corresponding to the feature subset $f_2 f_3$.
6. This has a j value of 23 which is smaller than B, so B remains unchanged.
7. The next node generated is E which is found to have

8. Since the J value of this node is less tha this branch is not expanded any further.
9. This means that the branches F and G are generated.

→ The next node to be generated is H, which ha J value of 34.
→ Since 34 > B, this node is expanded to give I
→ The J value of I is 20 which is lower than I. bound.
→ Now the entire tree is completed and the nod with the lower bound B has the criterion fun is the best subset which is f4f4. This is base on the fact that we are selecting 2 out of 4 features.

## Feature Extraction

→ Transforming higher dimension data to lower dimension.
→ It transforms or projects a original set of features into a new space, which has smaller no. of dimensions.




- overlapping should not be present.
- variant should be extend and gap should be more large variation is good.
- Feature extraction refers to process of identifying certain.
- This is a pre-processing step of intelligent com.
- Before classification is carried its necessary to decide what attributes of training data are
- Feature chosen should be discrimination feature of

- It is imp ~~feature~~ stage of feature.

TD

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \longrightarrow Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$m < n$

$Y = W^T X$

$W^T$ is the orientation of separator / line / plane / hyperplane.

- We have 2 methods for feature extraction.

1) Fisher's union discriminant
2) Principal Component Analysis (PCA)

1)

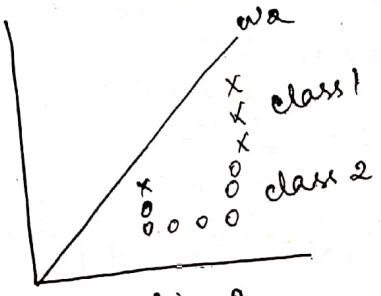



fig 2.
classes are separated
no overlap

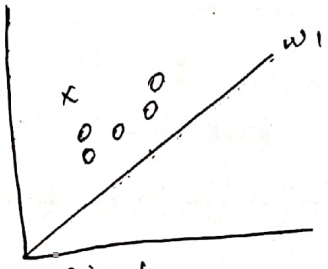



fig 1
class overlap.

- methods which are manageable and dimension are low become highly manageable when dimension increase.
- This is an account of large computational time reqd.
- To overcome this problem we try to reduce dimensions.
- One method of dimensionality reduction is to map the end dimensional samples on to a line.
- But here the orientation of the line is very imp for class separation.
- This is explained in fig 1 and fig 2.
- In fig 1 w1 is the line on which the sample of 2 classes are mapped.
- It is clear from the fig that the mapped samples of w1, overlap with one another and there is no class separation.
- Fig 2 shows a line having a orientation w2 when the samples of 2 classes are mapped on w2, we find that samples of 2 classes are well separated.

will be to find good orientation such as
- samples in original space, a sample x can be
on to a line w into $y = w^T x$.

$$= [w_1 \; w_2 \ldots w_d] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

$w_1, w_2, w_3$ are
$x_1, x_2, x_3$ are components of x

- Now $w_1, w_2, \ldots, w_d$ must be found such that
is a good class separation.
- The fishel's linear discriminant function is
function $w^T x$ for which the
criterian function $J(w) = \frac{w^T S_B w}{w^T S_w w}$ is maximu

where $S_B$ = class scatter matrix

$$= (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$\mu_i = \sum x_i$ where $x_i \in$ ith class.

$S_w$ = within class scatter matrix

$= s_1 + s_2$ where $s_i = \sum (x - \mu_i)(x - \mu_i)^T$

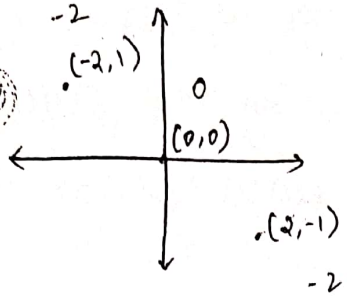
$x_i \in$ ith class

The solution of this is —

$$\boxed{w = S_w^{-1} (\mu_1 - \mu_2)}$$

# PCA - (Principal Component Analysis)

- One of the most imp & powerful dimensionality reduction technique.




$$\text{covariance} = \frac{(-2) + 0 + (-2)}{3} \quad \text{(product of cordinate)}$$

$$= \frac{-4}{3}$$

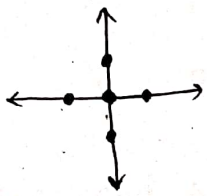
## Covariance matrix -

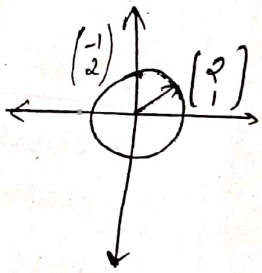
$$\Sigma = \begin{bmatrix} \text{var}(x) & \text{cov}(x,y) \\ \text{cov}(x,y) & \text{var}(y) \end{bmatrix}$$

Ex: $\begin{bmatrix} 9 & 4 \\ 4 & 3 \end{bmatrix}$

$(x,y) \rightarrow (9x+4y, 4x+3y)$

(0,0)
(1,0)
(0,1)
(-1,0)
(0,-1)





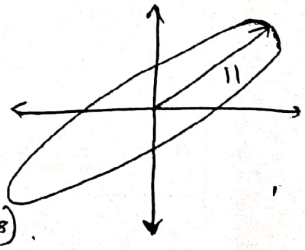

Eigen vectors (direction)

$\begin{pmatrix} 9 \\ 1 \end{pmatrix}$ $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$

Eigen values (magnitude)

11 1

$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances.




ultimately we end up with this line. It has the largest SS (distances).

→ PCA is a procedure by which no. of attributes are reduced.

→ The attempt here is to find a smaller no. of attributes which are uncorrelated.

→ The first principal component is the most important

possible.

→ The 2nd principle component comes next u

→ The no. of attributes are reduced by Proj
the data in the direction of max. variance ding

→ The method involves finding the eigen vecto
and the corresponding eigen values of the tors
co-variance matrix. ve

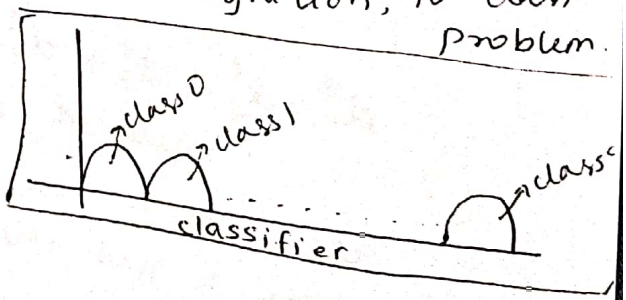
→ If the eigen vectors are ordered in descending
of the eigen values, the 1st eigen vector give
the direction of the largest variance of the data
By excluding the directions giving very low eige
values we can reduce the no. of attributes
being considered.

## What is classification?

- class
- classifier


## Problem domain

↳ linear problems —— 2 class problems → gender classifier
↳ non-linear problems —— n class problems.
↳ Handwritten numeric recognition, 10-class Problem.




Using training data we can classify sample P.

KNN, MN.

## Samples of the dataset

1 : (1, 1.25, 1)
2 : (1, 1, 1)
3 : (1.5, 0.75, 1)
4 : (2, 1, 1)
5 : (1, 3, 2)
6 : (1, 4, 2)
7 : (1, 5, 3.5, 2)
8 : (2, 3, 2)
9 : (4, 2, 3)
10 : (4.5, 1.5, 3)
11 : (5, 1, 3)
12 : (5, 2, 3)

| $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | class label |
|---|---|---|---|---|---|---|
| 1 | 4 | 3 | 6 | 4 | 7 | 1 |
| 4 | 7 | 5 | 7 | 4 | 2 | 2 |
| 6 | 9 | 7 | 5 | 3 | 1 | 3 |
| 7 | 4 | 6 | 2 | 8 | 6 | 1 |
| 4 | 7 | 5 | 8 | 2 | 6 | 2 |
| 5 | 3 | 7 | 9 | 5 | 3 | 3 |
| 8 | 1 | 9 | 4 | 2 | 8 | 3 |

## Metric and Non Metric measures

### Metric measure

1) Euclidean distance:

If we have two samples X and Y, the E.D will be

$$d(X,Y) = \sqrt{\sum_{k=1}^{d} (x_k - y_k)^2}$$

Ex: consider $X = (5, 3, 7, 5, 1)$, $Y = (6, 5, 2, 9, 5)$

$$\therefore d(X,Y) = \sqrt{(5-6)^2 + (3-5)^2 + (7-2)^2 + (5-9)^2 + (1-5)^2}$$

$$= \sqrt{1 + 4 + 25 + 16 + 16} = \sqrt{62} = 7.88$$

### Non-metric

1) k-median distance

2) Hausdorff distance (to measure distance b/n images)

(b/n vectors)

4) Edit distance (b/n strings)

Learning Algorithm

Nearest Neighbour classifier

1. If there are n samples $(x_1, x_2, \ldots, x_n)$ in the training data X and a test sample P.
2. If $X_k$ is the most similar sample to P from $X_1$ then the class of P is the class of $X_k$.
3. The similarity is usually measured by computing the distance from P to the training samples $X_1, X_2, \ldots, X_n$. If $d(P X_i)$ is the distance from P to $X_i$, then P is assigned the label of $X_k$ where $d(P X_k) = \min\{d(P X_i)\}$ where i is 1 to n.

Then $P \in X_k$.

k-Nearest Neighbor (Algorithm) classifier.

→ In k-nn an object is classified by a majority vote of the class of its neighbors. If k is 1, this becomes NN algo.

→ This algo may give a more correct classification for boundary samples than the NN algo.

→ The value of k has to be specified by user and the best choice depends on the data.

→ Larger values of k reduce the effect of noise on the classification.

→ The value of k can be ↑ when training data set is large in size

- The main disadvantage of knn algo is that it is very time consuming especially when training data

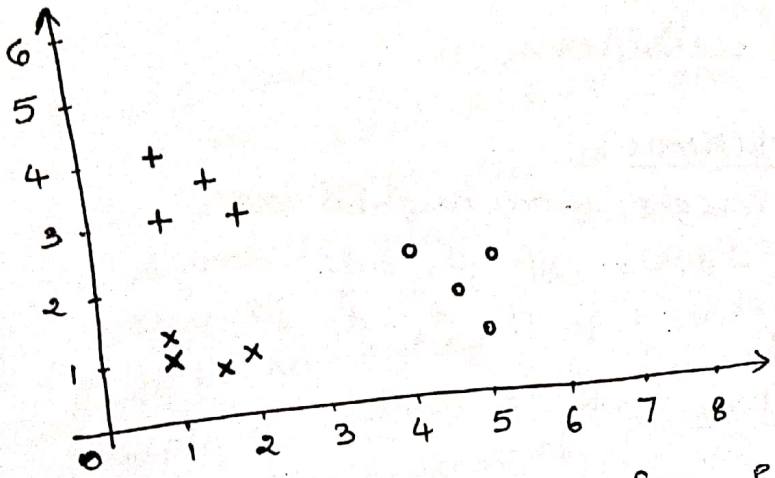
- To overcome this prob., a no. of algos have been proposed.

ex: Modified KNN classifier (MKNN), FKNN (Fuzzy)

Q) Plot the following training data on XY-plane find NN of P = (3,2) when K = 1, K = 3

(1,1,1)
(2,1,1)
(1,4,2)
(2,3,2)
(4.5, 1.5, 3)
(5,2,3)
(1,1.25,1)
(1.5,0.75,1)
(1,3,2)
(1.5,3.5,2)
(4,2,3)
(5,1,3)

x class 1
+ class 2
o class 3




Calculating 12 distances, i.e. from P(3,2,?) to all 12 points. Then find min.

$d_1 = \sqrt{(3-1)^2 + (2-1)^2} = \sqrt{5} = 2.236$

$d_2 = \sqrt{1+1} = \sqrt{2} = 1.414$

$d_3 = \sqrt{4+4} = \sqrt{8} = 2.828$

$d_4 = \sqrt{1+1} = \sqrt{2} = 1.414$

$d_5 = \sqrt{2.25+0.25} = 1.581$

$d_6 = \sqrt{4} = 2$

$d_7 = \sqrt{4+0.5625} \simeq 2.136$

$d_8 = 1.952$

$d_9 = \sqrt{4+1} = \sqrt{5} = 2.236$

$d_{10} = \sqrt{2.25+2.25} = 2.121$

$d_{11} = \sqrt{1} = 1$

$d_{12} = \sqrt{4+1} = \sqrt{5} = 2.236$

min = $d_{11} = 1$

∴ H belongs to class 3.

- The contribution of the neighbors to the classification is weighted according to its distance from the test sample.
- Hence, the nearest neighbor contributes more to the classification decision than the neighbors further away.

## Weighting Scheme

- The value of $w_i$ varies from 1 for the closest sample to 0 for the farthest sample among the k closest samples.
- This modification would mean that outlayers would not affect the classification as much as the KNN classifier.

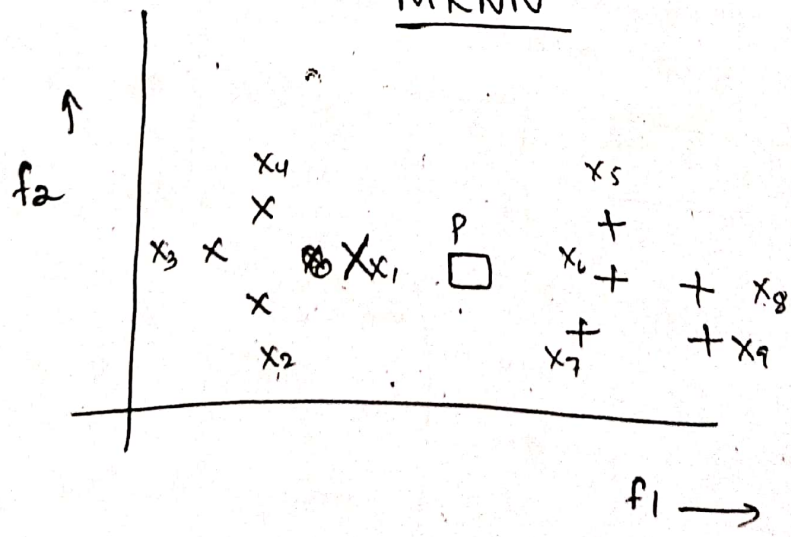
## Weighting scheme

Find the weight from neighbor as

$$w_i = \begin{cases} d_k - d_i & \text{if } d_k \neq d_i \\ 1 & \text{if } d_k = d_i \end{cases}$$

where $i = 1, 2, \ldots, k$

## MKNN



- Consider the 2 class problem shown in figure. There are 4 samples in class 1 marked as 'x' and there are 5 samples in class 2 marked as '+'. The test sample as P.
- Using the NN algo, the closest sample to P is $X_1$, whose class is 1.
  ∴ P will be assigned to class 1.

- If KNN algo is used, after $X_1$, P is closest to $X_6$ and $X_7$. So if $k=3$, P will be assigned to class 2.
  It can be seen that the value of k is crucial to the classification.
- If k is 1, it reduces to the NN classifier.
- In this case $k=4$, the next closest sample could be $X_2$. If $k=5$, and $X_3$ is closer to P, then $X_5$. Then again due to majority vote, P will be assigned to class 1. This shows how imp the value of k is to the classification.
- If P is an outlier of 1 class but is closest to sample of another class by taking majority vote, the misclassification of P can be prevented.

Validation Set

original set ⟶ ⟵

| Training | Testing |
|---|---|

| Training | validation | Testing |
|---|---|---|

validation fails to use all the available data.

## Performance Evaluation of Learning Algorithms

→ There are 3 types of errors -

1. Absolute Error
2. Sum of squares error $= \frac{1}{n}\sum_{i=1}^{n}(hx-y)^2$

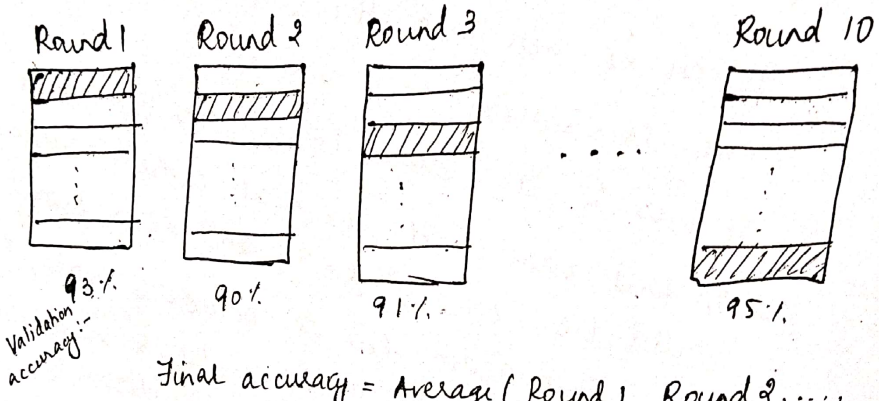
4) Confusion matrix.

1) Absolute Error

$$\text{Absolute error} = \frac{1}{n} \sum \|h(x) - y\|$$

K-fold cross validation

1. Split the data into k equal subsets
2. Perform k rounds of learning; on each roun
   - $1/k$ of the data is held out as a test se
   the remaining examples are used as t
   data.
3. Compute the average test set score of the k ;




Final accuracy = Average ( Round 1, Round 2, ..., Round 10)

validation set

training set

Confusion Matrix

- A confusion matrix is a summary of prediction results on a classification problem. (rail)
- The no. of correct and incorrect predictions are summarised with count values and broken down by each class. This is the key to the confusion matrix. (rail)
- The confusion matrix shows the ways in which your classification model is confused when it makes predictions. (rail)
- It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

| | Class 1 Predicted | Class 2 Predicted |
|---|---|---|
| class 1 Actual | TP | FN |
| class 2 Actual | FP | TN |

FN → samples are +ve but wrongly classified as -ve

FP → samples are -ve, but wrongly classified as +ve.

- Positive (P): Observation is +ve
- Negative (N): Observation is not +ve.
- True positive (TP): Observation is +ve, and is predicted to be +ve.
- False negative (FN): Observation is +ve, but is predicted -ve.
- True negative (TN): Observation is -ve, and is predicted to be -ve.
- False positive (FP): Observation is -ve, but is predicted +ve.

## Classification rate/Accuracy

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

- However there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

## Recall

Recall can be defined as the ratio of the total no. of correctly classified +ve examples divide to the total no. of +ve examples. High Recall indicates the class is correctly recognized (small no. of FN)

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

## Precision

- To get the value of precision we divide the total no. of correctly classified +ve examples by the total no. of predicted +ve examples. High precision indicates an example labeled as +ve is indeed +ve (small no. of FP).

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

- High recall, low precision: This means that most of the +ve examples are correctly recognized (low FN) but there are a lot of false +ves.
- Low recall, high precision: This shows that we miss a lot of +ve examples (high FN) but those we predict as +ve are indeed +ve (low FP).

## F-measure

- Since we have 2 measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.
- The F-measure will always be nearer to the small value of Precision or Recall.

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

**Ex:**

Confusion matrix :-

| n=165 | Predicted: No | Predicted: Yes | |
|---|---|---|---|
| Actual: No | Tn = 50 | FP = 10 | 60 |
| Actual: Yes | Fn = 5 | TP = 100 | 105 |
| | 55 | 110 | |

Accuracy = 0.90
Recall = 0.95
Precision = 0.91
F-measure = 0.92
} Answers.

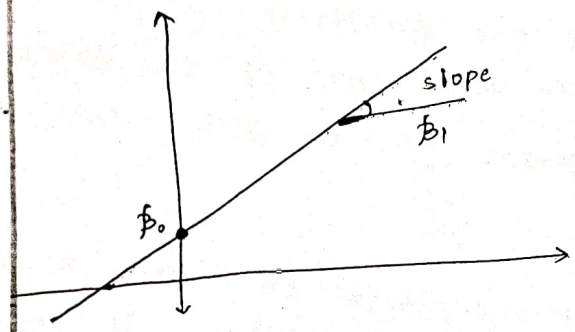
Relationship between variables is a linear function.

$Y = \beta_0 + \beta_1 x_1 + \epsilon$ → Random error.

$\beta_0$ → population Y-intercept

$\beta_1$ → population slope

$f : X \rightarrow Y$
↳ continuous value.
x is a single feature.




→ For the 2-d problem :- $Y = \beta_0 + \beta_1 X$
→ To find the values for the coefficients which minimize the objective function, we take the partial derivates of the objective function (SSE) wrt the coefficients. Set these to 0, and solve.

$$\beta_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2} \qquad \beta_0 = \frac{\sum y - \beta_1 \sum x}{n}$$

## MKNN

### Weighted KNN:

- The contribution of the neighbors to the classification is weighted according to its distance from the test sample P.
- Hence the nearest neighbor contributes more to the classification decision than the neighbors further

- One weighting scheme would be to give ea... neighbor a weight of $1/d$ where $d$ is the d... from P to the neighbor.

- Another weighting scheme binds the weight from the neighbor as $w_i = \begin{cases} \frac{d_k - d_i}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$

where $i = 1, 2, \ldots, k$

- Using MKNN the classification depends on the distances of the closest samples from the test sample. In the KNN algo, all the k samples will have equal importance.

- In MKNN the closest sample is given more significance than the farthest sample. The weightage given to the class of the first closest sample is more than for the 2nd closest sample and so on.

* Ex: If 5 nearest neighbors to P are $x_1, x_2, x_3, x_4$ and $x_5$, where $d(x_1, P)$ is 1 unit and $d(x_2, P) = 2$, $d(x_3, P) = 2.5$, $d(x_4, P) = 4$ and $d(x_5, P) = 5$. and if $x_1$ and $x_4$ belongs to class 1 and $x_2$, $x_3$ and $x_5$ belongs to class 2, then the weightage given to class 1 by $x_1$ will be $w_{11} = 1$ ($\because d_k = d_i$). The weight given to class 1 by $x_4$ will be $w_{14} = \frac{d_k - d_i}{d_k - d_1} = \frac{5-4}{5-1}$

$= \frac{1}{4} = 0.2$

Now, $w_1 = w_{11} + w_{14}$

$= 1 + 0.25 = 1.25$

The weight given to class 2 by $x_2$ will be

$w_{22} = \frac{5-2}{5-1} = \frac{3}{4} = 0.75$

The weight given to class 2 by $x_3$, $w_{23} = \frac{5-2.5}{5-1}$

$= \frac{2.5}{4} = 0.625$

The weight given to class 2 by $x_5$, $w_{25} = \frac{5-5}{5-1} = 0$

$\therefore w_2 = w_{22} + w_{23} + w_{25} = 1.375$

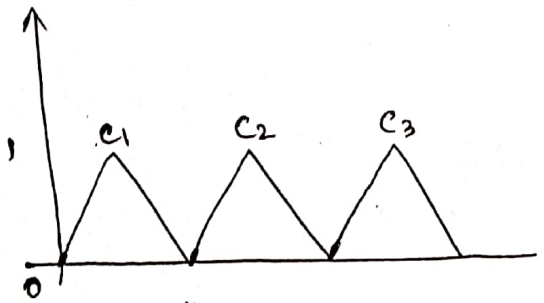
$\because w_2 > w_1$, P is classified as belonging to class 2.

KNN v/s MKNN :-

↓ Equal imp to all neighbors

→ more imp given to samples close to P.

Fuzzy KNN




Fuzzy membership value $[0, 1]$

As m ↑, the weightage of neighbors ↓

Let $A = \sum_{j=1}^{k} \mu_{ij} \left( \frac{1}{d(P, x_j)^{2/m-1}} \right)$

Let $B = \sum_{j=1}^{k} \left( \frac{1}{d(P, x_j)^{2/m-1}} \right)$

Then $\mu_i P = \frac{A}{B}$

$m = 2, 3, \ldots$

Q) Consider a test sample P which is at a distance of 1 unit from the closest point. This sample is of class 1. P is at a distance of 2 from the next closest point which belongs to class 2. P is at a distance of 3 from the next closest point which is of class 3. P is at a distance of 4 from the

point which is of class 2. Let m = 2.

⇒ The membership value of P to class 1 will be,

$$u_{1P} = \frac{\frac{1}{12} + \frac{1}{42}}{\left(\frac{1}{12} + \frac{1}{22} + \frac{1}{32} + \frac{1}{42} + \frac{1}{52}\right)} = 0.7259$$

$$u_{2P} = \frac{\frac{1}{22} + \frac{1}{52}}{\left(\frac{1}{12} + \frac{1}{22} + \frac{1}{32} + \frac{1}{42} + \frac{1}{52}\right)} = 0.19814$$

$$u_{3P} = \frac{\frac{1}{32}}{\left(\frac{1}{12} + \frac{1}{22} + \frac{1}{32} + \frac{1}{42} + \frac{1}{52}\right)} = 0.07591$$

(0.7259 is the greatest among the 3 values.
∴ P is assigned to class 1.)

$u_{4P} = 0$

$u_{5P} = 0$

Since the membership value wrt class 1 is the highest, P is assigned to class 1.

NN, KNN, MKNN, FKNN

→ They are all instance based learning algorithms.
→ Time and space complexity are issues, they are high.
→ No learning model here.

Efficient algorithms for nearest neighbor classifiers

→ While the NN algorithm gives good results and is robust, it is tedious to use when the training data is very large.
→ Finding the nearest neighbour to a test sample requires the computation of the similarity b/n the test sample and every training data.
→ If there are n samples with dimensionality d this will be O(nd).

→ The no. of training sample, if it's very large or if the dimensionality of samples is very large, this will be a very time consuming exercise.

→ Another disadvantage is that the entire training set should be stored and utilized to find the most similar sample at each iteration throughout the classification process.

## Efficient Algorithms

→ The solutions to these problem are as follows—

1. Some preprocessing is carried out on training set so that the process of finding out the nearest neighbor is easier. These algos are called efficient algos.
2. The no. of training samples is reduced. This is done so that it does not cause classification accuracy to come down too much. This is called prototype selection.
3. The dimensionality of the samples is reduced. The attributes which are less significant are removed and only the attributes which are significant are retained for classification. This process is called ~~feature~~ dimensionality reduction or feature selection.

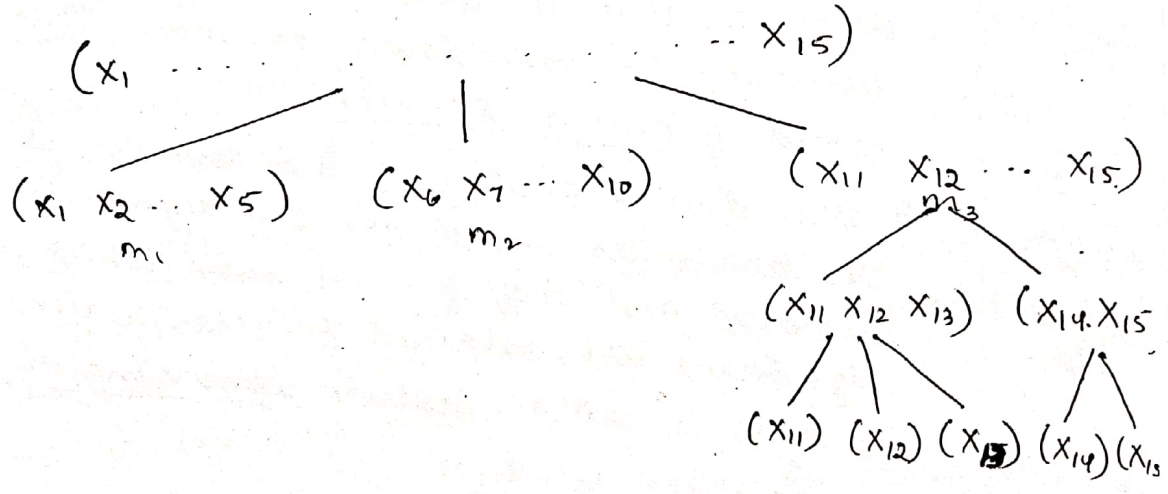
## Branch and Bound technique

- This is an efficient algorithm.
- The training data are clustered together so that points which are close together form a cluster.
- These first level clusters are again divided into clusters and this is done recursively till each cluster will have ~~only~~ one point.
- For every cluster j, the centre $m_j$ and the radius $r_j$ are calculated and stored.

## Classification Process

- To find the closest training sample to a test sample P, the following steps are followed.

1. For every cluster, calculate $a_j = (d(P,m) - r_j)$
2. All points in cluster j will have a distance from P which is more than $a_j$.
3. Choose the cluster k, among the clusters, which has the smallest $a_k$ and find the distances from P to the points in the cluster. Let the smallest distance be $d_{min}$.
4. Only the clusters j which have $a_j < d_{min}$ needs to be searched.
5. If $a_j \geq d_{min}$, then the cluster need not be searched.




$X_1 = (1\ 1\ 1\ 1)$ $X_2 = (1\ 2\ 1\ 1)$ $X_3 = (2\ 2\ 2\ 1)$

$X_4 = (2\ 2\ 2\ 1)$ $X_5 = (1\ 2\ 2\ 1)$ $X_6 = (1\ 2\ 5\ 2)$

$X_7 = (2\ 1\ 6\ 2)$ $X_8 = (2\ 2\ 5\ 2)$ $X_9 = (1\ 2\ 6\ 2)$

$X_{10} = (2\ 2\ 6\ 2)$ $X_{11} = (4\ 1\ 2\ 3)$ $X_{12} = (4\ 2\ 1\ 3)$

$X_{13} = (5\ 2\ 2\ 3)$ $X_{14} = (6\ 1\ 2\ 3)$ $X_{15} = (5\ 1\ 1\ 3)$

→ The radius of a cluster is the max. distance [illegible] and the centroid

$m_1 = \frac{1+1+2+2+1}{5} = \frac{7}{5} = 1.4$

~~$m_2 = \frac{1+2+2+1+2}{5} = \frac{8}{5} = 1.6$~~ $m_2 = \frac{1+2+2+2+2}{5} = 1.8$

$m_3 = \frac{1+1+2+2+2}{5} = \frac{8}{5} = 1.6$

Let us take point P = (4 4 3) and class label is unknown, as the test point for which we have to find the closest sample (nearest neighbor).

At the first level, we have to calculate $a_j$ for each cluster j. For cluster 1, the classification algorithm,

$d(P, m_1) = 3.94$. From this,

$a_1 = 3.94 - 0.98 = 2.96$

$a_2 = 3.18$ $a_3 = 1.73$

$a_3$ is smaller than $a_1$ and $a_2$.

The sub cluster

## Minimal Distance classifier :-

class 1

$x_1$ : (1.0, 1.0) $x_2$ : (1.0, 2.0)
$x_3$ : (1.5, 2.0) $x_4$ : (2.5, 1.5)
$x_5$ : (3.0, 2.0)

class 2

$x_6$ : (4.0, 2.0) $x_7$ : (5.0, 2.0)
$x_8$ : (5.0, 1.0) $x_9$ : (6.0, 2.0)
$x_{10}$ : (7.8, 1.0)

class 3

$x_{11}$ : (4.0, 4.0) $x_{12}$ : (5.0, 4.0)
$x_{13}$ : (4.0, 5.0) $x_{14}$ : (6.0, 5.5)
$x_{15}$ : (6.0, 5.0)

- Prototypes are samples which represent training data.
- The prototype set is a subset of training data or samples derived from the training data.
- One method is to find the centroid of each class and use it to represent all the samples in the class.
- This is called the minimal distance classifier.
- If $x_1, x_2, \ldots, x_n$ represent the n samples of a class, then the representative sample will be mean of centroid.

$$c = \frac{\sum_{i=1}^{n} x_i}{n} \qquad x_i = \{x_1, x_2, \ldots, x_n\}$$

$$c_1 = \left(\frac{1+1+1.5+2.5+3}{5}, \frac{1+2+2+1.5+2}{5}\right)$$

$$= \left(\frac{9}{5}, \frac{8.5}{5}\right) = (1.8, 1.7)$$

$$c_2 = \left(\frac{27.8}{5}, \frac{8}{5}\right) = (5.56, 1.6)$$

$$c_3 = \left(\frac{25}{5}, \frac{23.5}{5}\right) = (5, 4.7)$$

The point P's distance is calculated with these 3 centroids. Then mind the min dist. Computational

Let P be (3.5, 3.0)

$d(P\ C_1) = \sqrt{2.14}$

$d(P\ C_2) = 2.49$

$d(P\ C_3) = 2.26$

} 2.14 is the smallest.
∴ class label of point P is 1.

- Distance b/n classes should be maximum. Distance within class should be minimum. This is the best distribution.

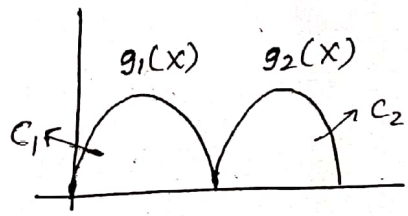
CNN (Condensed Nearest Neighbor)
Database should be prepared.

## Discriminant functions (DF) ※

→ It is a function which discriminates b/n classes, when the input vector to be classified is given.

→ input vector $X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

Its represented as —




→ $C_1$, $C_2$ are class labels of $g_1(x)$ and $g_2(x)$ respectively.

→ If $g_1(x) > g_2(x)$, then classify X to $C_1$, else classify X to $C_2$.

### Choosing of discriminant function

We have 2 methods —

Method 1:

- This is based on minimum error rate classification.
- We assign input sample X to the class $w_i$, if

$P(w_i/X) > P(w_j/X)\ \forall\ i \neq j$

∴ The DF can be taken as

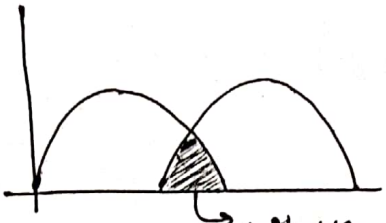
$$g_i(x) = P(w_i|x)$$

$$= \frac{P(x|w_i).P(w_i)}{\sum_{i=1}^{n} P(x|w_i)}$$ [Bayes Theorem]

In case of a 2 class classifier :-

$$g_1(x) = \frac{P(x|w_1)\,P(w_1)}{\sum_{i=1}^{2} P(x|w_i)} \qquad g_2(x) = \frac{P(x|w_2)\,P(w_2)}{\sum_{i=1}^{2} P(x|w_i)}$$

If $g_1(x) > g_2(x)$, then classify x to $w_1$, else to $w_2$.



when the classes are overlapping, then there is error. If they are not overlapping, then $P(\text{error}) = 0$.