PATTERN RECOGNITION

&

NEURAL NETWORKING

BY TANUJIT CHAKRABORTY

(RESEARCH SCHOLAR, ISI KOLKATA)

Mail : tanujitisi@gmail.com

CONTENT **Page No.**

PATTERN RECOGNITION

— Tanujit Chakraborty
(RS, ISI Calcutta)

1. Pattern Recognition and Machine Learning — CM Bishop (Springer)
2. Pattern Classification — Duda • Hart • Stock.
3. Statistical Pattern Recognition – K Fuknaga.
4. P.A. Devijver, J. Kittler — PR – A Statistical Approach.
5. Book by Vapnik.

Machine Learning Techniques

Supervised Learning (Classification)
(Learn an input to output Map)
(Labelled Training data)

- Artificial Neural Networks
- Support Vector Machines
- Decision Trees (CART)
- Set of rules
- Bayesian Networks

Unsupervised Learning (Clustering)
(Discover patterns in the data)
(Unlabelled training data)

- Clustering: Cohesive grouping
- Association Rule Mining.

• Pattern Recognition:- Examples →
- Face Recognition Problem,
- Recognition of Speech and Speakers.
- Hand writing recognition.

• Face Recognition Problem:- Criminal database with 1lakh photographs and matching 1 new photograph with others. So, by looking at all photographs we can't do this. So, we need machine to do this. So, features/categories them like Male/Female, heights, colours, i.e., in CS language we are making a tree so that we can make identification faster. This is known as 'Face Recognition Problem'.

• Satelite image processing, we use Pattern Recognition.

• 'ROBOT' — recognising the difference between snake & a chalk.

• Medical imaging — Reading X-Ray by a machine.

Patterns are present and we have to recognise it property. Thinking of human being, making classification and putting this into a machine is the job of pattern recognition people. Like we say "the man is OK. Not good and Not bad". But computer only understands '0' and '1', 'Yes or No'. Human being has 'EYE' to see, 'EARS' to hear, 'NOSE' to understand to smell, 'TONGUE' to taste. These are all operated by 'BRAIN', and human beings are able to learn new things. We want to make a machine to do the same.

<u>Advantage of Machine</u> is that it is not emotional but human beings are, so we sometimes make wrong decisions.

- <u>Satelite Image Processing Problem:</u> — Classification of pixels in satelite images is the problem in hand. Indian Remote Sensing satelite (IRS) takes multispectral images, i.e., for each location on earth you are going to have four images at the same time. These images correspond to the wave lengths blue, green, red and infrared. Now our problem is, whatever pixels you have, you are supposed to classify each pixel to one of the land cover types (e.g. water region on earth, hilly region, building, etc). Questions arise How to do it? What is the use of doing this classification?

<u>Uses:</u> - The government may be interested in knowing how much of forest area may be getting diplited every year. If you ask human beings to do it, you may not get acurate estimate of this. You need to have a machine to do it. Pattern Recognition can be used to do these.

- We can take decisions without doing the job actually (like climbing up the hill, jumping into the river) but a '<u>ROBOT</u>' must be able to make the decision, distinguish between objects and make judgements. All these are a part of pattern recognition. We are supposed to recognise the patterns.

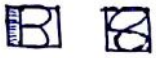
<u>Patter recognition</u> people wants to model human thinking process. Neural network tries to mimic the thinking of human being so that it does the classification. In Pattern Recognition subject, we want to achieve what human being can think logically can be done by a Comp.

Pattern Recognition

Measurement space → Feature space → Decision space

Main tasks: Feature selection and Supervised / Unsupervised Classification.

Example (Character Recognition Problem): Distinguish between B and 8.

Solution to this problem can be 

So, if the distance between the parallel line and B are same for each of points, then we can classify it as 'B', otherwise it will be '8'.

So, from measurement space, we went to feature space, then further proceeded to decision making. This is just a very simple method.

Supervised Classification:- (Classification)

Two cases:

1. Conditional probability density functions and priors probabilities are known.
2. Training sample points are given.

Some properties that could be possibly to be used to distinguish between the two types of fishes are:

- Length
- Lightness
- Width
- Number and shapes of fins
- Position of the mouth, etc...

} Features

Feature is a property (or characteristic) of an object which is used to compare or distinguish between (or classify) two objects.

Features must be invariant to: Translation, Rotation, Scale, Noise & Other Projective Transform. (Image case)

Desirable properties of Features:

- Must be distinct and unique for a given object / shape / signal.
- Computational cost must not be high
- Must have graceful degradation due to discontinuities and missing parts.

Feature Vector:-

- A single feature may not be useful always for classification.
- A set of features used for classification form a feature vector.

METHODS OF CLUSTERING AND CLASSIFICATION

CLUSTERING

- Representative Points
- Split & Merge
- Linage
- Self-Organising Map (SOM)
- Model Based
- Vector Quantization

CLASSIFICATION

- Bayes Decision Rule
- Linear Discriminant Analysis (LDA) (Fishers Criteria)
- K-Nearest Neighbour (K-NN)
- Feed Forward Neural Network (FFNN)
- Support Vector Machine (SVM)
- CART & Random Forest.

Classification Problem (Bayes Decision Rule) :- Let there be two classes.

Let p_i denote the conditional probability function for the ith class; $i=1,2$.

$p_1(\underset{\sim}{x})$, $p_2(\underset{\sim}{x})$ are the pdfs. $x \in \mathbb{R}^N$

Let P_i be the prior probability of the ith class ; $i=1,2$.

Then $P_1 + P_2 = 1$, Ω : set of all positive values of n-dimensional feature vector.

Also assume that $\Omega \subseteq \mathbb{R}^N$ for the sake of convenience.

And $0 \leq P_1, P_2 \leq 1$.

Partitioning Ω into Ω_1 and Ω_2 such that

(i) $\Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_1 \cap \Omega_2 = \phi$ and $\Omega_1 \neq \phi, \Omega_2 \neq \phi$.

(ii) Ω_1 and Ω_2 are Borel Measurable sets.

- Decision rule for classification is given by $D(\Omega_1, \Omega_2)$ denotes:

$x \in \Omega_1$, $\Rightarrow$ the corresponding unit will be placed in class 1.

$x \in \Omega_2$ $\Rightarrow$ the corresponding unit will be placed in class 2.

Also, $D(\Omega_1, \Omega_2) \neq D(\Omega_2, \Omega_1)$.

$\mathcal{D} = \{ D(\Omega_1, \Omega_2) : \}$ denotes the set of all possible decisions.

Let us consider one decision rule, say, $D(\Omega_1, \Omega_2)$

Decision \ Reality	$\underset{\sim}{x}$ is from class 1	$\underset{\sim}{x}$ is from class 2
$\underset{\sim}{x}$ is put in class 1, $\underset{\sim}{x} \in \Omega_1$	✓	Error (*)
$\underset{\sim}{x}$ is put in class 2, $\underset{\sim}{x} \in \Omega_2$	Error (**)	✓

For 2 classes, we have $2^2 - 2$ errors.
For 3 classes, we have $3^2 - 3$ errors.
⋮
For k classes, we have $k^2 - k$ errors.

$$P(\text{Misclassification} / \text{Observation from class 1}) = \int_{\Omega_2} p_1(x)\, d\underset{\sim}{x} \quad (*)$$

$$P(\text{Misclassification} / \text{Observation from class 2}) = \int_{\Omega_1} p_2(\underset{\sim}{x})\, d\underset{\sim}{x} \quad (**)$$

Probability of misclassification for the rule $D(\Omega_1, \Omega_2)$ is given by,

$$\mathcal{E}(\Omega_1, \Omega_2) = P_1 \int_{\Omega_2} p_1(\underset{\sim}{x})\, d\underset{\sim}{x} + P_2 \int_{\Omega_1} p_2(\underset{\sim}{x})\, d\underset{\sim}{x}$$

We need to minimize $\mathcal{E}(\Omega_1, \Omega_2)$ over all possible D in $\mathcal{D}$.

To find $D(\Omega_1^0, \Omega_2^0)$ such that

$$\mathcal{E}(\Omega_1^0, \Omega_2^0) \le \mathcal{E}(\Omega_1, \Omega_2) \quad \forall\, D(\Omega_1, \Omega_2) \in \mathcal{D}$$

Then $D(\Omega_1^0, \Omega_2^0)$ is optimal decision rule.

$$\mathcal{E}(\Omega_1, \Omega_2) = \underbrace{P_1 \int_{\Omega_2} p_1(\underset{\sim}{x})\, d\underset{\sim}{x}} + P_2 \int_{\Omega_1} p_2(\underset{\sim}{x})\, d\underset{\sim}{x} + \underbrace{P_1 \int_{\Omega_1} p_1(\underset{\sim}{x})\, d\underset{\sim}{x}} - P_1 \int_{\Omega_1} p_1(\underset{\sim}{x})\, d\underset{\sim}{x}$$

$$= P_1 \int_{\Omega_1 \cup \Omega_2} p_1(\underset{\sim}{x})\, d\underset{\sim}{x} + \int_{\Omega_1} \left(P_2 p_2(\underset{\sim}{x}) - P_1 p_1(\underset{\sim}{x}) \right) d\underset{\sim}{x}$$

$$= P_1 + \int_{\Omega_1} \left(P_2 p_2(\underset{\sim}{x}) - P_1 p_1(\underset{\sim}{x}) \right) d\underset{\sim}{x} \quad \text{——— (1)}$$

Similarly, $$\mathcal{E}(\Omega_1, \Omega_2) = P_2 + \int_{\Omega_2} \left(P_1 p_1(\underset{\sim}{x}) - P_2 p_2(\underset{\sim}{x}) \right) d\underset{\sim}{x} \quad \text{——— (2)}$$

Adding ① and ②, we get,

$$2\,\varepsilon(\Omega_1, \Omega_2) = 1 + \int_{\Omega_2} \left(P_1 p_1(\underset{\sim}{x}) - P_2 p_2(\underset{\sim}{x})\right) d\underset{\sim}{x} + \int_{\Omega_1} \left(P_2 p_2(\underset{\sim}{x}) - P_1 p_1(\underset{\sim}{x})\right) d\underset{\sim}{x} \quad \text{③}$$

Like to minimize $2\,\varepsilon(\Omega_1, \Omega_2)$ over all possible (Ω_1, Ω_2), let us denote 'optimal' (Ω_1, Ω_2) as (Ω_1^o, Ω_2^o).

Let
$$A_1 = \left\{ \underset{\sim}{x} : P_1 p_1(\underset{\sim}{x}) - P_2 p_2(\underset{\sim}{x}) < 0 \right\} = \left\{ \underset{\sim}{x} : P_2 p_2(\underset{\sim}{x}) > P_1 p_1(\underset{\sim}{x}) \right\}$$
$$A_2 = \left\{ \underset{\sim}{x} : P_1 p_1(\underset{\sim}{x}) - P_2 p_2(\underset{\sim}{x}) = 0 \right\} = \left\{ \underset{\sim}{x} : P_1 p_1(\underset{\sim}{x}) = P_2 p_2(\underset{\sim}{x}) \right\}$$
$$A_3 = \left\{ \underset{\sim}{x} : P_2 p_2(\underset{\sim}{x}) - P_1 p_1(\underset{\sim}{x}) < 0 \right\} = \left\{ \underset{\sim}{x} : P_1 p_1(\underset{\sim}{x}) > P_2 p_2(\underset{\sim}{x}) \right\}$$

Optimal set, $\left\{ \begin{array}{l} \Omega_1^o = A_2 \cup A_3 \\ \Omega_2^o = A_1 \end{array} \right\}$ Without loss of generality.

This decision rule is known as Bayes Decision Rule. It minimizes the probability of misclassification.

- There can be many optimal decision rules. All of the giving the same value of the error probability. We can take any one of them. And every such decision rule is called as Bayes Decision Rules, $D(\Omega_1^o, \Omega_2^o)$.

<u>Generalization</u>:- – Let there be M classes. $(M > 2)$

– Class conditional probability functions $p_1(\underset{\sim}{x}), p_2(\underset{\sim}{x}), \ldots, p_M(\underset{\sim}{x})$; $\underset{\sim}{x} \in \mathbb{R}^N$

– Prior probabilities are given by $P_1, P_2, \ldots, P_M$.

$$0 < P_i < 1, \quad i = 1(1)M.$$
$$\sum_{i=1}^{M} P_i = 1.$$

– Bayes decision rule is put $\underset{\sim}{x}$ in class i if $P_i p_i(\underset{\sim}{x}) \geq P_j p_j(\underset{\sim}{x}) \;\forall j \neq i$

- <u>Example of Misclassification Problem</u>:- Banks use Cameras to monitor the persons coming with guns or not. It's difficult for a human being to watch the video footage for a long time. But if we put a machine to check it we may face two difficulties:
  1. The man is not carrying but Machine says carrying.
  2. The man is carrying but Machine says NOT carrying.

Two possible errors are there. But Bank Manager may compromise with 1st error. So, errors don't have equal weights. So, here we <u>can't apply Bayes Decision rule</u>.

Exercise:- Derive Bayes Decision Rule for $M = 3$.

Solution:- Num. of classes $= 3$.

Conditional probability functions will be p_1, p_2, p_3.

Prior probabilities will be P_1, P_2, P_3.

The optimal decision rule: $D(\Omega_1^o, \Omega_2^o, \Omega_3^o)$

If $\underset{\sim}{x} \in \Omega_1^o$, we put it in class 1.

$\underset{\sim}{x} \in \Omega_2^o$, we put it in class 2.

$\underset{\sim}{x} \in \Omega_3^o$, we put it in class 3.

$$\Omega_1^o = \left\{ \underset{\sim}{x} : P_1 p_1(\underset{\sim}{x}) \geq P_2 p_2(\underset{\sim}{x}) \;\&\; P_3 p_3(\underset{\sim}{x}) \right\}$$

$$\Omega_2^o = \left\{ \underset{\sim}{x} : P_2 p_2(\underset{\sim}{x}) \geq P_3 p_3(\underset{\sim}{x}),\; P_2 p_2(\underset{\sim}{x}) > P_1 p_1(\underset{\sim}{x}) \right\}$$

$$\Omega_3^o = \left\{ \underset{\sim}{x} : P_3 p_3(\underset{\sim}{x}) > P_1 p_1(\underset{\sim}{x}),\; P_3 p_3(\underset{\sim}{x}) > P_2 p_2(\underset{\sim}{x}) \right\}$$

Example 1:-

$$p_1(\underset{\sim}{x}) = \begin{cases} x & ; 0 < x < 1 \\ 2-x & ; 1 \le x \le 2 \\ 0 & \text{otherwise} \end{cases}$$ prior probability P.

$$p_2(\underset{\sim}{x}) = \begin{cases} x-1 & ; 1 < x < 2 \\ 3-x & ; 2 \le x < 3 \\ 0 & \text{otherwise} \end{cases}$$ prior probability $1-P$.

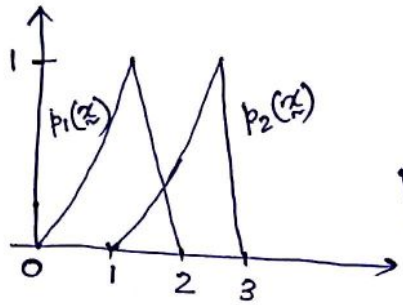
(i) Find Ω_1^0 and Ω_2^0 for the above case and find the probability of misclassification for the Bayes decision rule.

(ii) $\Omega_1 = (0,1]$ and $\Omega_2 = (1,3)$. Find the prob. of misclassification for the decision rule.

(iii) Show that it is (given in (ii)) $\geq$ the prob. of misclassification of Bayes Decision rule.

Solution:-

Graph of the functions:-



$\Omega = [0,3]$

[Triangular Distribution]

(i) $\Omega_1^0 = \{x : P p_1(\underset{\sim}{x}) \geq (1-P) p_2(\underset{\sim}{x})\}$

Case I:- $0 \le x < 1$

$\Rightarrow p_2(x) = 0$

$\Rightarrow P p_1(x) \geq (1-P) p_2(x) \Rightarrow [0,1) \subseteq \Omega_1^0$.

$\Rightarrow P p_1(x) \geq 0$

Case 2:- $2 \le x \le 3$

$\Rightarrow p_1(x) = 0$

$\Rightarrow (1-P) p_2(x) \geq P p_1(x) \Rightarrow [2,3] \subseteq \Omega_2^0$

Case 3:- $1 \le x < 2$

$P p_1(x) \geq (1-P) p_2(x)$

$\Rightarrow P(2-x) \geq (1-P)(x-1)$

$\Rightarrow 1+P \geq x$

$\Rightarrow x \le 1+P$

So, $\Omega_1^0 = [0, 1+P]$ and $\Omega_2^0 = (1+P, 3]$.

[$\because$ We know $0 \le P \le 1$]

Probability of misclassification $= P\int_{1+P}^{3} p_1(x)\,dx + (1-P)\int_{0}^{1+P} p_2(x)\,dx$

$$= P\int_{1+P}^{2}(2-x)\,dx + \left[\int_{1}^{1+P}(x-1)\,dx\right](1+P) = \frac{P(1-P)}{2}.$$

(ii) Prob. of misclassification $= P\int_{\Omega_2} p_1(x)\,dx + (1-P)\int_{\Omega_1} p_2(x)\,dx$

$$= P\int_{1.2}^{3} p_1(\underset{\sim}{x})\,d\underset{\sim}{x} + (1-P)\int_{0}^{1.2} p_2(\underset{\sim}{x})\,d\underset{\sim}{x}$$

$$= P\int_{1.2}^{2}(2-x)\,dx + \left[\int_{1}^{1.2}(x-1)\,dx\right](1-P)$$

$$= 0.32P + (1-P)(0.02) = 0.02 + 0.3P.$$

(iii) $0.02 + 0.3P \geq \frac{P(1-P)}{2}$

$\Leftrightarrow 0.04 + 0.6P \geq P - P^2$

$\Leftrightarrow P^2 - 0.4P + 0.04 \geq 0$

$\Leftrightarrow (P-0.2)^2 \geq 0$ [Q.E.D]

Example 2:- M classes
$P_1, P_2, \ldots, P_M$ are prior probabilities.

$$p_i(\underset{\sim}{x}) = \frac{1}{(\sqrt{2\pi})^N |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_i})'\Sigma_i^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_i})\right\}; \; i=1(1)M.$$

Case(i):- $P_1 = P_2 = \ldots = P_M = \frac{1}{M}$

$$\Sigma_1 = \Sigma_2 = \ldots = \Sigma_M = I.$$

$$P_i p_i(\underset{\sim}{x}) \geq P_j p_j(\underset{\sim}{x}) \quad \forall j \neq i$$

$$\Leftrightarrow p_i(\underset{\sim}{x}) \geq p_j(\underset{\sim}{x}) \quad \forall j \neq i$$

$$\Leftrightarrow e^{-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_i})'(\underset{\sim}{x}-\underset{\sim}{\mu_i})} \geq e^{-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_j})'(\underset{\sim}{x}-\underset{\sim}{\mu_j})} \quad \forall j \neq i$$

$$\Leftrightarrow (\underset{\sim}{x}-\underset{\sim}{\mu_i})'(\underset{\sim}{x}-\underset{\sim}{\mu_i}) \geq (\underset{\sim}{x}-\underset{\sim}{\mu_j})'(\underset{\sim}{x}-\underset{\sim}{\mu_j}) \quad \forall j \neq i$$

This classifier is known as '<u>MINIMUM DISTANCE CLASSIFIER</u>'.

Case (ii):- Two Normal distribution with equal covariance matrices:-

$M = 2,$

$\Sigma_1 = \Sigma_2 = \Sigma$

$P_1 p_1(\underset{\sim}{x}) \geq P_2 p_2(\underset{\sim}{x})$

$$\Leftrightarrow P_1 \frac{1}{(\sqrt{2\pi})^N |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_1})'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_1})\right\}$$
$$\geq P_2 \frac{1}{(\sqrt{2\pi})^N |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_2})'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_2})\right\}$$

$$\Leftrightarrow \log P_1 - \frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_1})'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_1}) \geq \log P_2 - \frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu_2})'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_2})$$

$$\Leftrightarrow \frac{1}{2}\left[(\underset{\sim}{x}-\underset{\sim}{\mu_2})'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_2}) - (\underset{\sim}{x}-\underset{\sim}{\mu_1})'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu_1})\right] \geq \log\frac{P_2}{P_1}.$$

$$\Leftrightarrow \frac{1}{2}\left[\underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x} - \underset{\sim}{x}\Sigma^{-1}\underset{\sim}{\mu_2} - \underset{\sim}{\mu_2}'\Sigma^{-1}\underset{\sim}{x} + \underset{\sim}{\mu_2}'\Sigma^{-1}\underset{\sim}{\mu_2} - \underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x} + \underset{\sim}{\mu_1}'\Sigma^{-1}\underset{\sim}{x} + \underset{\sim}{x}\Sigma^{-1}\underset{\sim}{\mu_1} - \underset{\sim}{\mu_1}'\Sigma^{-1}\underset{\sim}{\mu_1}\right] \geq \log\frac{P_2}{P_1}$$

$$\Leftrightarrow \frac{1}{2}\left[2\underset{\sim}{x}'\Sigma^{-1}(\underset{\sim}{\mu_1}-\underset{\sim}{\mu_2}) + \underset{\sim}{\mu_2}'\Sigma^{-1}\underset{\sim}{\mu_2} - \underset{\sim}{\mu_1}'\Sigma^{-1}\underset{\sim}{\mu_1}\right] \geq \log\frac{P_2}{P_1}.$$

$$\Leftrightarrow h(\underset{\sim}{x}) = \underbrace{(\underset{\sim}{\mu_1}-\underset{\sim}{\mu_2})\Sigma^{-1}}_{\text{constant}}\underset{\text{linear in } x}{\underset{\sim}{x}} + \underbrace{\frac{\underset{\sim}{\mu_2}'\Sigma^{-1}\underset{\sim}{\mu_2} - \underset{\sim}{\mu_1}'\Sigma^{-1}\underset{\sim}{\mu_1}}{2}}_{\text{constant}} \geq \underbrace{\log\frac{P_2}{P_1}}_{\text{constant}}.$$

So, $h(\underset{\sim}{x})$ is called <u>Linear Discriminant Function</u>.

Put $\underset{\sim}{x}$ in class 1 if $h(\underset{\sim}{x}) \geq \log \frac{P_2}{P_1}$.

$$E\left(h(\underset{\sim}{x}) / \underset{\sim}{x} \text{ is from class } 1\right) = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_1 + \frac{1}{2}\left(\underset{\sim}{\mu}_2' \Sigma^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1 \Sigma^{-1} \underset{\sim}{\mu}_1\right)$$

$$= \frac{1}{2}\left[\underset{\sim}{\mu}_1' \Sigma^{-1} \underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2' \Sigma^{-1} \underset{\sim}{\mu}_2 - 2\underset{\sim}{\mu}_2 \Sigma^{-1} \underset{\sim}{\mu}_1\right]$$

$$= \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)$$

$$= \frac{1}{2}\Delta^2, \text{ where } \Delta^2 = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)$$

= Mahalanobis Distance

$$Var\left(h(\underset{\sim}{x}) / \underset{\sim}{x} \text{ is from class } 1\right) = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2) = \Delta^2$$

$$= Var\left(h(\underset{\sim}{x}) / \underset{\sim}{x} \text{ is from class } 2\right).$$

$$E\left(h(\underset{\sim}{x}) / \underset{\sim}{x} \text{ is from class } 2\right) = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_2 + \frac{1}{2}\left(\underset{\sim}{\mu}_2' \Sigma^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1' \Sigma^{-1} \underset{\sim}{\mu}_1\right)$$

$$= \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2) = \frac{1}{2}\Delta^2$$

Probability of Misclassification

$$= Prob\left(h(\underset{\sim}{x}) \geq \log \frac{P_2}{P_1} / \underset{\sim}{x} \text{ is from class } 2\right) P_2 + Prob\left(h(\underset{\sim}{x}) < \log \frac{P_2}{P_1} / \underset{\sim}{x} \text{ is from } 1\right) P_1$$

$$= P_2 \int_{\log \frac{P_2}{P_1}}^{\infty} \frac{1}{\sqrt{2\pi}\,\Delta} e^{-\frac{1}{2}\left(\frac{y + \frac{1}{2}\Delta}{\Delta}\right)^2} dy + P_1 \int_{-\infty}^{\log \frac{P_2}{P_1}} \frac{1}{\sqrt{2\pi}\,\Delta} e^{-\frac{1}{2}\left(\frac{y - \frac{1}{2}\Delta}{\Delta}\right)^2} dy$$

$$= P_2\left(1 - \phi\left(\frac{t + \frac{1}{2}\Delta^2}{\Delta}\right)\right) + P_1 \phi\left(\frac{t - \frac{1}{2}\Delta^2}{\Delta}\right).$$

$t = \log \frac{P_2}{P_1}$ (let)

$$\phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy$$

Which classification method is good?

We divide the data sets into two parts: training set and test set. From the training set, we develop the classifier, like using K-nearest neighbour rule, bayes decision rule, multilayer perceptron, etc. Once the classifiers are developed, we use this classifier to classify points in the test set. Now whichever classifier is giving better performance on the test set assuming the performance of all these classifiers on the training set are same, that classifier is called better one.

How to use this method? Difficulties in application of Bayes Decision Rule?

To apply Bayes Decision Rule, we need to know the conditional probability density function of the given data set and prior probabilities. So, apart from being the best classifier, Bayes classifier is difficult to apply in practical scenario. In 1960, Perzen shown a way of estimation of pdfs which is commonly used. This is the best rule in the sense that it minimizes the probability of misclassification. So How do we apply this rule if we don't know the density functions of the training data set?

→ We estimate the density function or we assume some functional form and estimate the parameter and then use Bayes Decision Rule. We can't apply this rule to all the data sets other than knowing pdfs and prior probabilities.

So why one needs to know Bayes Decision Rule?

→ The answer is this is the best rule, if you develop a new classifier, you can find its performance by comparing your classifier with Bayes Decision Rule. You generate points artificially from known distributions, then we can apply Bayes Decision, and you apply your classifier that you have developed and check the prob. of misclassification in both cases and see its close to Bayes classifier or not. That's why this is the starting point to study Pattern Recognition. So, 1. It's the best classifier.

2. Due to application issue, we can't apply in some practical cases, there we use other classifier (SVM, Multilayer Perceptron), K-NN Classifier) and check its performance using Bayes classifier.

Assignment 1:- (i) Down 4 Kolkata images from Prof. CA Murthy's Webpage. Link: www.isical.ac.in/~murthy/ "Calcutta images" — gif. file.

(ii) Let $g_K(i,j)$ denote the gray value of $(i,j)^{th}$ part in the Kth image.
$K = 1, 2, 3, 4$; $0 \le i, j \le 511$; i, j are integers.

$$M_K = \max_{i,j} g_K(i,j), \quad m_K = \min_{i,j} g_K(i,j)$$

Let $$f_K(i,j) = \frac{g_K(i,j) - m_K}{M_K - m_K} \times 255 \; ; \; K = 1, 2, 3, 4.$$

(iii) Consider f_4. Take 50 locations (i.e., $(i_1, j_1), (i_2, j_2), \ldots, (i_{50}, j_{50})$) from river portion manually.
Similarly, take 100 locations from non-river portion manually.

(iv) Let, for a pixel location (i,j),
$$\underset{\sim}{x}'(i,j) = (g_1(i,j), g_2(i,j), g_3(i,j), g_4(i,j))$$

(v) Find mean and covariance matrix corresponding to the 50 four dimensional points of River water.
Similarly, find mean and covariance matrix corresponding to 100 four dimensional points of Non-river portion.

(vi) 1 – River
2 – Non River

Three cases:
Case (i) $P_1 = 0.3$, $P_2 = 0.7$
(ii) $P_1 = 0.5$, $P_2 = 0.5$
(iii) $P_1 = 0.7$, $P_2 = 0.3$

(vii) For each case, apply Bayes decision rule on each point (by assuming Normal distribution for each cases and by using the sample estimates for mean and covariance matrix) and produce a binary image.

K-Nearest Neighbour Decision Rule (Fix and Hodges)

Let (x_i, θ_i); $i = 1, 2, \ldots, n$ be given where $\underset{\sim}{x}_i \in \mathbb{R}^N$; $i = 1, 2, \ldots, n$; and θ_i denotes the label of x_i for each i, means the class from which the observation $\underset{\sim}{x}$ comes.

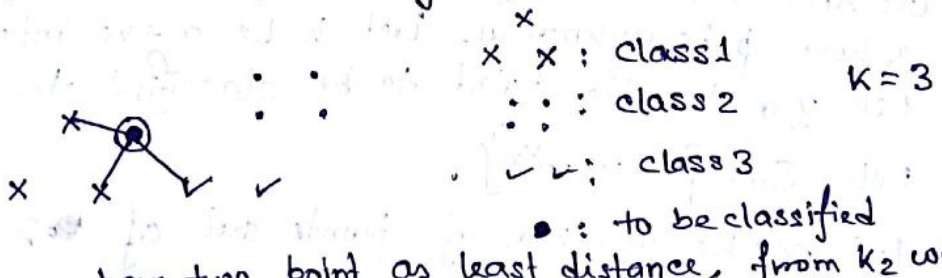
Let us assume that the number of classes is c, c is an integer and ≥ 2.

i.e., $\theta_i \in \{1, 2, \ldots, c\}$ $\forall$ i.

Let $\underset{\sim}{x}$ be a point for which the label is not known, i.e., $\underset{\sim}{x}$ be the point to be classified. We need to find the label of $\underset{\sim}{x}$.

Procedure:-

1. Let k be a positive integer.
2. Calculate $d(\underset{\sim}{x}, \underset{\sim}{x}_i)$ for all $i = 1(1)n$, where 'd' denotes the Euclidean distance.
3. Arrange the 'n' distances in non-decreasing order.
4. Take the first k distances.
5. Find those k points corresponding to those k-distances.
6. Let k_i denote the no. of points belonging to the ith class among the k points, $i = 1, 2, \ldots, c$. (k_i denote the no. of nearest neighbour)
7. Put $\underset{\sim}{x}$ in class i if $k_i > k_j$ $\forall$ $j \neq i$.



So, from k_1 we have two point as least distance, from k_2 we have one, from k_3 we have zero. So, we can put them in k_1.

Doubts:
1. How to choose the value k?
2. What happens when $k_i = k_j$?
3. How do I say some particular value of k is better than other values of k?
4. What is the theoretical justification of this rule?
5. Is is necessarily true for different values of k we will be getting same results? (NO: ANS)

Loftsgarden (1965), "Anals of Mathematical Statistics" (AMS)

K-nearest neighbour density estimation procedure

Let $X_1, X_2, \ldots, X_n$ be i.i.d. random vectors with common probability density function f, when f is unknown. Let X_i's take values in $\mathbb{R}^N$. Let x_0 be the point at which the pdf is to be estimated. Let K be an positive integer.

Let $S_n = \{\underset{\sim}{X_1}, \underset{\sim}{X_2}, \ldots, \underset{\sim}{X_n}\}$.

(i) Find K-nearest neighbours of x_0 in S_n.

(ii) Let the K-th nearest neighbour lie at distance r from x_0. Let V denote the volume of a sphere of radius r in $\mathbb{R}^N$.

(iii) Let $f_n(x_0) = \dfrac{K}{nV}$ = estimated density

If $\underset{\sim}{x_0}$ is a continuity point of f, then $f_n(x)$ is an assymptotically unbiased unbiased and consistent estimate of $f(x_0)$. When a) $K \to \infty$ as $n \to \infty$ &

b) $\frac{K}{n} \to 0$ as $n \to \infty$.

c) K is a function of n, like, $\sqrt{n}$.

Set up:- Let there be c classes with prior probs. $P_1, P_2, \ldots, P_c$. and density function of the ith class be p_i. Then the mixture prob. density function $p(\underset{\sim}{x}) = \sum_{i=1}^{c} P_i p_i(\underset{\sim}{x})$; $\underset{\sim}{x} \in \mathbb{R}^N$.

Let $X_1, \ldots, X_n$ be iid random vectors with common pdfs p, where p is unknown. Let K be a +ve integer.

Let $\underset{\sim}{x_0}$ be the point to be classified to one of the c classes

Let $S_n = \{X_1, \ldots, X_n\}$.

Let x_i be the no. of points out of x belongs to the class i, $i = 1, 2, \ldots, c$.

Thus $\hat{P}_i = \dfrac{n_i}{n}$; $i = 1, 2, \ldots, c$.

Finding:- 1. In K-NN Rule we assume the distances to be euclidean distance, what will happen when it won't be non-euclidean?

2. Estimate prior prob. by proportions estimate disity by Loftsgarden method; then if we apply Bayes Rule, we will get the K-Nearest Neighbour rule.

Finding K-nearest neighbours of x_0 in S_n :-

Let K_i of the nearest neighbours belong to class i, $i=1,2,\dots,c$.

Then $\sum_{i=1}^{c} K_i = K$.

Let the Kth nearest neighbour of x_0 be at a distance r from x_0. Let V denote the volume of a sphere of radius r in $\mathbb{R}^N$.

If x_0 is a continuity point of p then

$$\hat{p}(x_0) = \frac{K}{nV}.$$

Note that $\hat{p}_i(x_0) = \frac{K_i}{n_i V}$; $i=1,2,\dots,c$.

We still apply Bayes decision rule using the estimates $\hat{p}_i(x_0)$ and $\hat{P}_i$ $\forall i, j$.

Put x_0 in the ith class if

$$\hat{P}_i \hat{p}_i(x_0) > \hat{P}_j \hat{p}_j(x_0) \quad \forall j \neq i$$

$$\Leftrightarrow \frac{n_i}{n} \cdot \frac{K_i}{n_i V} > \frac{n_j K_j}{n \cdot n_j V} \quad \forall j \neq i$$

$$\Leftrightarrow K_i > K_j \ \forall j \neq i.$$

<u>Assignment 2:-</u> Apply K-NN rule for the satellite image dataset. for $K = 1, 3, 5, 7, 9, 11$.

<u>Remark:</u> (i) 1-NN decision rule is same as nearest neighbour decision rule. [When $K=1$]

(ii) It may be necessary to relate the size of the training data set if the size is large.

<u>Procedure for reducing the size of the training set for the nearest neighbour decision rule :—</u> (See Next Page)

<u>Finding:-</u> How to choose the value of K? (Way: Cross-validation (popularly used approach)) [See Fukunaga Book]

Let $S = \{x_1, \dots, x_n\}$ be the given training set.
Let θ_i denote the label of x_i $\forall\, i = 1, 2, \dots, n$; θ_i's are known.

STORE $= \phi$, GB1 $= \phi$, GB2 $= \phi$.

(i) Put $\underset{\sim}{x}_1$ in STORE

(ii) For $i = 2, 3, \dots, n$. classify $\underset{\sim}{x}_i$ using 1-NN rule where the training set is taken as STORE. If $\underset{\sim}{x}_i$ is misclassified, put $\underset{\sim}{x}_i$ in STORE. Otherwise put it in GB2.

(iii) For every point $\underset{\sim}{y}$ in GB2,

classify $\underset{\sim}{y}$ using 1-NN rule when the training set is STORE.

If $\underset{\sim}{y}$ is correctly classified, then put $\underset{\sim}{y}$ in GB1. Otherwise put $\underset{\sim}{y}$ in STORE.

(iv) If GB1 = GB2 then stop the algorithm, with output as STORE.

Otherwise Rename GB1 as GB2, GB1 $= \phi$ & go to (iii).

<u>Remark:-</u> (i) The final output depends on the ordering of points.

<u>Homework:-</u> Consider the procedure for K-NN rule.

<u>Naive Bayes Rule:-</u> The basic assumption made in Naive Bayes rule is that for each class, the probability density function corresponding to N random variables is the product of their marginal density functions.

The modified Bayes decision rule after including the above assumption is called Naive Bayes decision rule.

<u>Homework:-</u> Let there be c classes with prior probs $P_1, P_2, \dots, P_c$. Show that the prob. of misclassification for Bayes Decision rule is $\leq 1 - \text{Max}\{P_1, P_2, \dots, P_c\}$.

<u>Assignment 3:-</u> Apply Naive Bayes Rule on Satelite imagery for the three cases:

(i) $P_1 = 0.3$, $P_2 = 0.7$

(ii) $P_1 = P_2 = 0.5$

(iii) $P_1 = 0.7$, $P_2 = 0.3$.

Homeworks

1. Real life data sets, available in UCI Archive
2. Artificial data sets.

[It is always necessary to work on artificial data sets for understanding the limitations of the method]

[If new points (data) are added in real life data sets then the method we used can't be said to be useful for the new data sets sometimes]

$\underset{\sim}{\mu_1}' = (0,0)$, $\underset{\sim}{\mu_2}' = (0,1)$, $\underset{\sim}{\mu_3}' = (1,0)$

$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 2 & -1 \\ -1 & 1.5 \end{pmatrix}$, $\Sigma_3 = \begin{pmatrix} 2 & 1 \\ 1 & 1.5 \end{pmatrix}$

Case (i) $P_1 = 0.5 = P_2$
(ii) $P_1 = 0.4$, $P_2 = 0.6$ } $n = 100, 500, 1000, 2000, 5000$.

(A) Consider $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_1)$ under case (i). Generate n points from the mixture density function randomly.

(B) Consider the same under case (ii). Same Question.

(C) Consider $N(\mu_1, \Sigma_1)$, $N(\mu_2, \Sigma_2)$ under case (ii), generate n points from a Mixture density function randomly.

(D) Consider $N(\mu_1, \Sigma_1)$, $N(\mu_2, \Sigma_2)$ and $N(\mu_3, \Sigma_3)$ $P_1 = 0.3$, $P_2 = 0.4$, $P_3 = 0.3$. Generate n points from the mixture density function randomly.

[Note that when you are generating these points, you are not only knowing the points but also the class, level of the points]

1. Take the first $\frac{n}{2}$ points as training set, and rest as test sets,
2. For each training set and for each class; estimate mean, covariance matrix and prior probability.
3. Apply Bayes decision rule on each point in the test set by using the estimated values under the assumption of Normality, and find the misclassification rate.

[Misclassification rate: Take a point in the test set, if a point is put in class i and rule also saying same, then no misclassification

$$\text{Rate} = \frac{\text{No. of misclassified point}}{\text{Total number of points}}$$]

④. Apply Bayes decision rule on each point in the test set by using the actual parameter values under normality assumption, and find misclassification rate.

[As $n \uparrow$ the difference between misclassification rates in Q 3, 4 will decrease]
-: Standard Rule

(D) → [Hints: If you generate points between 0 and 1, and suppose it belongs to 0 – 0.3, then you take it from the first distribution, if 0.3 – 0.7, then you should generate a random vector from 2nd Distribution $(N(\mu_2, \Sigma_2))$; from 0.7 – 1, generate from $N(\mu_3, \Sigma_3)$.]

1. Real life Data set Problem:- Fisher's IRIS Data
(4 Dimensional Observations, 3 classes)

PROBABILITY DENSITY ESTIMATION

Whenever we have discussed classification or classifiers, many times we have assumed the form of the density either normal or some other density functions.

But in reality how to find density from given data set?

— There are various approaches, one such approach is fitting distribution and checking whether the fit is good or not. But that too is assuming a functional form of the distribution.

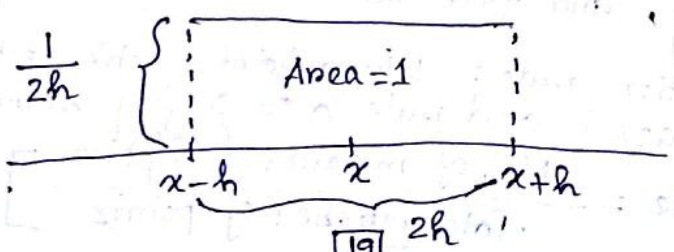
— Parzen, 1960, AMS Paper, "Parzen's density estimation".

Let $x_1, x_2, \ldots, x_n$ are observations drawn randomly from an unknown distribution.

$X_1, X_2, \ldots, X_n$ are i.i.d. random variable having same pdf f, where f is unknown.

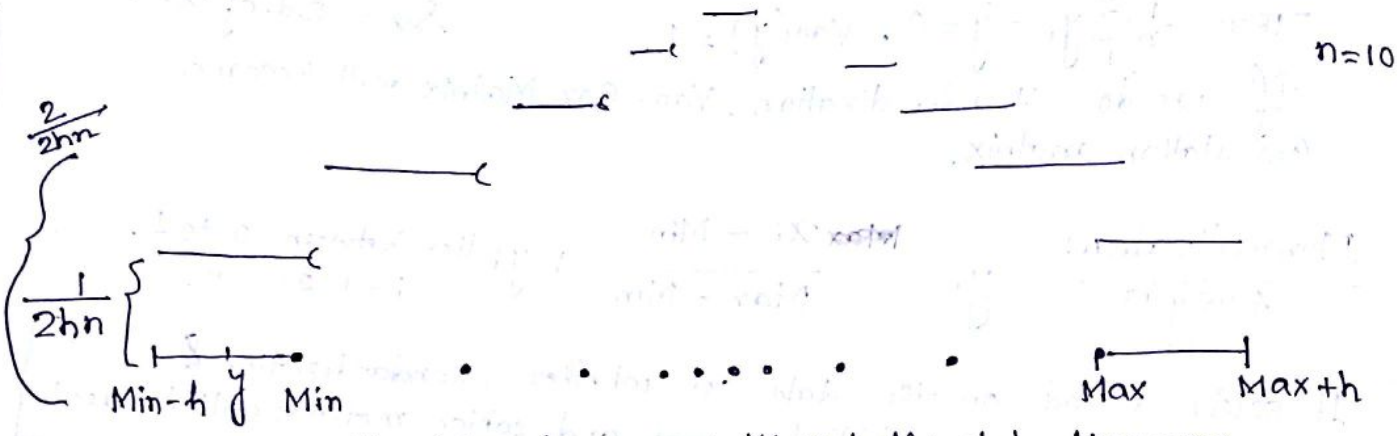
Let's assume $h > 0$;

Define $g_x(y) = \begin{cases} \frac{1}{2h} & ; y \in (x-h, x+h) \ \forall x \in \mathbb{R} \\ 0 & ; \text{OW} \end{cases}$



Define estimate of the function f,

$$\hat{f}_n(y) = \frac{1}{n}\sum_{i=1}^{n} g_{x_i}(y)$$



Depending on the data set, we will get the step diagram.

Q. How to choose h?

$\rightarrow$ $h = h(n)$; Properties are:

(i) $h_n \to 0$ as $n \to \infty$

(ii) $nh_n \to \infty$ as $n \to \infty$

(iii) $nh_n^2 \to \infty$ as $n \to \infty$

Example of such $h(n)$s are:

$\frac{1}{n^{1/3}}$, $\frac{1}{\log n}$.

Check:- $\frac{1}{\log n} \to 0$ as $n \to \infty$

$\frac{n}{\log n} \to \infty$ as $n \to \infty$

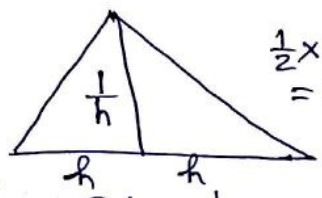
Now, note that $\hat{f}_n(y)$ is asymptotically unbiased, i.e., $E(\hat{f}_n(y)) \simeq f(y)$

This method is known as Parzen's Window Techniaue.

Q. If we don't consider uniform distribution at the beginning then what will happen?

Ans:- Parzen has shown this for a class of functions, like, triangular, normal, etc.

He called it as "Kernel Density Estimation"



$\frac{1}{2} \times 2h \times h = 1$.

For Triangular.

For normal: $x = \mu$, $h = \sigma$.

- Whichever kernel function you will take, it will be unbiased and consistent ($MSE \to 0$)
- More Properties of kernel function is given in Fuknaga book.
- Generalisation was done by CACOULOUS, AMS (1963).

$$g_{\underset{\sim}{x}}(\underset{\sim}{y}) = \begin{cases} \dfrac{1}{2^m h_{1n} h_{2n} \cdots h_{mn}} & ; \text{if } y_i \in (x_i - h_{in}, x_i + h_{in}) \ \forall i = 1(1)n. \\ 0 & ; \text{ow} \end{cases}$$

Standardization:- A variable takes values $x_1, x_2, \ldots, x_n$.

Then a way of standardizing it as $y_i = \frac{x_i - \bar{x}}{s_x}$; $i = 1, 2, \ldots, n$.

where, $\bar{x}$ = mean of x_i

s_x = s.d. of x_i.

Then $\frac{1}{n}\sum_{1}^{n} y_i = \bar{y} = 0$, $Var(y) = 1$.

If we do Standardization, Var-Cov. Matrix will become Correlation matrix.

Normalization:- A way is $y_i = \frac{x_i - Min}{Max - Min}$; y_i lies between 0 to 1. $i = 1, 2, \ldots n$.

[It solely depends on the data set whether Standardization & Normalization will be applied or not and which methods will be used.]

Another way $y_i = 2\left[\frac{x_i - Min}{Max - Min}\right] - 1$; $i = 1(1)n$; $-1 \le y_i \le 1$.

[Standardization, Normali-zation are related to feature selection problem]

METRIC SPACE:- $X \neq \phi$

$d : X \times X \to [0, \infty)$ $\quad$ $\times$: cross product

d is said to be a METRIC on X if (i) $d(x,y) = d(y,x)$ $\forall$ $x, y \in X$.

(ii) $d(x,x) = 0$ $\forall$ $x \in X$

$d(x,y) = 0 \Leftrightarrow x = y$

(iii) $d(x,y) + d(y,z) \geq d(x,z)$ $\forall$ $x, y, z \in X$.

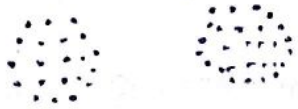
- The symbol 'd' is used as distance.
- (iii) Triangular Inequality [Sum of two sides > third side '=' holds when 3 sides are on a straight line]
- $|X - Y|$ is a metric, i.e., modulus distance between 2 points.

Clustering (Unsupervised Classification):→

Problem:- We are given $S = \{x_1, \ldots, x_n\}$. We need to divide the data set into K groups (or, clusters) so that each cluster is "homogeneous" and two different clusters are "heterogeneous".

- In other words, finding natural groups in data set.

Example 1:



$K = 2$.

Example 2:

Difficulties:- Let us assume the given data set $S = \{\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_n\} \subset \mathbb{R}^M$.

- No. of clusters K may not be known.
- Choice of Similarity / dissimilarity measure.
- Algorithms.

Example:- Satelite images land cover types. $K = ?$ Even river water and pond water are different.

Assumption:- K is known.

General Steps:- (i) Define a 'similarity' measure or a 'dissimilarity' measure between points.

(ii) A criterion is to be defined which expresses the meaning of "homogeneity" or "heterogeneity".

(iii) An algorithm is to be formed for clustering.

Q. Given a data set is clustering unique?

Ans:- Given a data set you can get many meaningful clusterings. Like in a pack of cards, Clustering can be: Black & Reds, Club - heart - diamond - spades, etc.

If similarity & dissimilarity measure changes, the clustering changes.

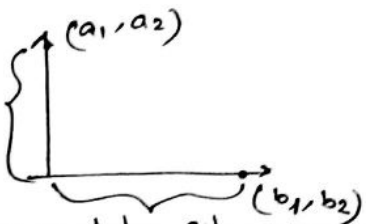
Example of Dissimilarity Measures:-

$$\underset{\sim}{a}' = (a_1, a_2, \ldots, a_M)$$
$$\underset{\sim}{b}' = (b_1, b_2, \ldots, b_M)$$

$$d_p(\underset{\sim}{a}, \underset{\sim}{b}) = \left(\sum_{i=1}^{M} |a_i - b_i|^p \right)^{1/p} ; \; p \geq 1.$$

$p = 2 \longrightarrow$ euclidean distance.

$p = 1 \longrightarrow$ city block distance



There are uncountably many metrics. Given a data set, you need to choose the metric.

Example of Similarity Measure :-

$$\underset{\sim}{a}' = (a_1, a_2, \ldots, a_m)$$
$$\underset{\sim}{b}' = (b_1, b_2, \ldots, b_m)$$

$$s(\underset{\sim}{a}, \underset{\sim}{b}) = \frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2} \sqrt{\sum_{i=1}^{m} y_i^2}}$$

Other such measures are also possible.

* This is like cosine(θ) or correlation coefficient measure.

A typical example:-

It depends on the problem at hand to define similarity and dissimilarity measures.

Example:

Subject	Person A	Person B
1	90/100	100/100
2	90/100	100/100
3	90/100	100/100
4	90/100	100/100
5	90/100	45/100

→ You need to choose who is better student among A and B?

→ Which distance measure will you use?

$d_1 = 50$, $d_2 = 55$; d_1 = city block distance $= 5(100-90)$.

Now, if the subjects are usual subjects, we can call A is better, but if subjects are learning Ragas in classical music, then B can be better (where perfection matters).

Criterion function:- An example of a criterion function is given below:

$S = \{ \underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_n \} \subseteq \mathbb{R}^m$.

Number of clusters $= K$ (known).

Let $d(\underset{\sim}{x}, \underset{\sim}{y})$ denote Euclidean distance between $\underset{\sim}{x}$ and $\underset{\sim}{y}$.

Let $P(A_1, \ldots, A_K)$ denote a partition of S into K-subsets.

i.e.,
(i) $A_i \neq \phi \ \forall \ i = 1, 2, \ldots, K$
(ii) $A_i \cap A_j = \phi$ if $i \neq j$.
(iii) $\bigcup_{i=1}^{K} A_i = S$.

— This criterion is known as MINIMUM WITHIN CLUSTER DISTANCE CRITERION.

▨ Let us assume we are finding dissimilarity using Euclidean distance.

Let us define $L(P(A_1, \ldots, A_K)) = \sum_{i=1}^{K} \sum_{\underset{\sim}{x} \in A_i} d^2(\underset{\sim}{x}, y_i)$; where y_i is mean of A_i; $i = 1(1)K$.

Find $(A_1^0, A_2^0, \ldots, A_K^0)$ such that

$$L(P(A_1^0, A_2^0, \ldots, A_K^0)) \leq L(P(A_1, A_2, \ldots, A_K)) \ \forall \ P(A_1, A_2, \ldots, A_K).$$

L → loss function; P → Partition

Remarks:-

(i) The criterion essentially provides convex shaped clusters.

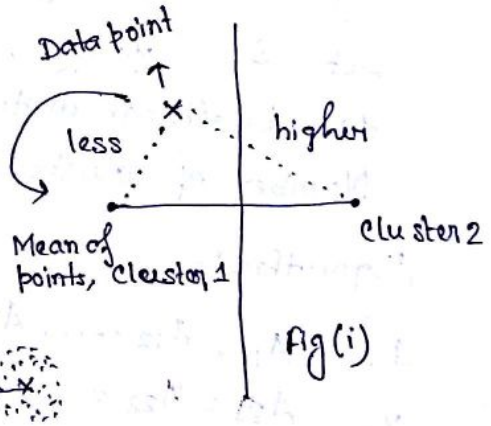


Fig (i)

(ii) It need not provide non-convex shaped clusters.

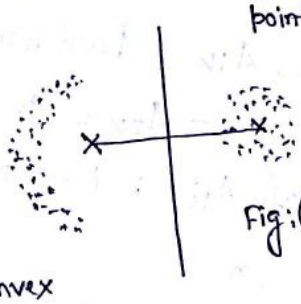


Fig: (i)

(iii) Suppose the clusters are convex shaped, then do we get those clusters by using the above criterion?

Fig (ii)

Ans. NO.

} convex shape

It is not necessarily true that convex clusters existing in the data set, they are always obtained from the above criterion.

NOTE:- Given any criterion, you have to see it's properties first.

(iv) Given n and k, $2 \le k < n$, find the number of partitions having k subsets

$$P = \frac{k^n - \binom{k}{1}(k-1)^n + \binom{k}{2}(k-2)^n \cdots (-1)^{k-1}\binom{k}{k-1}}{k!}$$

For $k = 2$, $P = \frac{2^n - 2}{2!}$

For $k = n$, $P = 1$.

[Note that $n! = n^n - \binom{n}{1}(n-1)^n + \binom{n}{2}(n-2)^n \cdots (-1)^{n-1}\binom{n}{n-1}$]

(v) One of the algorithm which tries to provide clusters having minimum within cluster distance is <u>K-Means algorithm</u>. This is a place where you can find a new algorithm to serve this purpose as a doctoral candidate.

<u>K-Means Clustering Algorithm:-</u> Several versions of K-Means algorithm are available in literature. One version of the algorithm by FORGY (1965) is given below:

Let $S = \{\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_n\} \subset \mathbb{R}^M$ be the given set.

d: Euclidean distance for dissimilarity measure.

Number of clusters $= K$ (known).

Algorithm:-

1. $A_{11}, A_{12}, \ldots, A_{1K}$ Partition of S into K subsets.
2. $A_{21} = A_{22} = \ldots = A_{2K} = \phi$
3. $y_i =$ mean of A_{1i} ; $i = 1, 2, \ldots, K$.
4. For $j = 1, 2, \ldots, n$
   Put x_j in A_{2i} if $d(x_j, y_i) < d(x_j, y_{i_1})$, $i_1 \neq i$.
5. If $A_{1i} = A_{2i}$ for all $i = 1, 2, \ldots, K$ then stop ow, Rename A_{2i} as A_{1i} $\forall$ $i = 1, 2, \ldots, K$ and go to step 2.

After FORGY, in FIFTH BERKELEY SYMPOSIUM, MAC QUEEN published a paper (1967) "Some Methods for Classification and Analysis of Multivariate Observations" which shows some modification to the above algorithm.

Remarks:-

1. The algorith 'USUALLY' converges.
2. In real life applications, many times the maximum number of clusters is decided by the user.
3. Many researchers start the algorithm with an initial partition and calculate the mean, etc. Two different sets of initial seed points may result in two different clusters.
4. The depth of the obtained clusters satisfy the proportion mentioned for Minimum within cluster distance (MWCD) criterion.
5. Medoid: Many definitions are available for this. One such definition is given below:

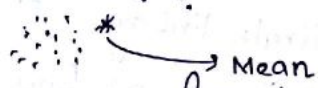
$A = \{\underset{\sim}{x}_1, \dots, \underset{\sim}{x}_n\} \subseteq \mathbb{R}^m$; d = distance function.

$a_{x_i} = \sum_{j=1}^{n} d(x_i, x_j)$; $i = 1, 2, \dots, n$.

[Medoid ≒ Median]

Call $x_0 \in A$ as medoid of A if $a_{x_0} = \min_i a_{x_i}$ $\forall i$. Note that, x_0 is not unique and $i = 1, 2, \dots, n$.

6. Split and merge techniques are also available.
7. This method assumes no. of clusters to be known & it has a problem with non-convex distance. There are few drawbacks of K-Means algorithm. Another problem with this method is — any method which is based on mean is very much susceptible to outliers. Extreme values may suffer the clustering method.



One method is to remove the outliers after studying it, then doing clustering. Question comes is it always good to remove outliers?

Answer:- Diameter method, Variance method, Split & Merge Algorithm

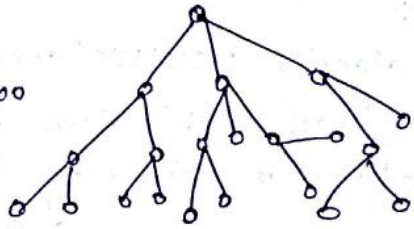
→ Now we will discuss about non-convex clusters. (Next Page).

8. Another way of defining MEDOID:- $S = \{\underset{\sim}{x}_1, \dots, \underset{\sim}{x}_n\} \subseteq \mathbb{R}^m$

$\bar{\underset{\sim}{x}} = \frac{1}{n} \sum_{i=1}^{n} x_i$. Let $\underset{\sim}{x}_i \in S$ be such that $d(\underset{\sim}{x}_i, \bar{\underset{\sim}{x}}) \leq d(\underset{\sim}{x}, \bar{\underset{\sim}{x}})$ $\forall \underset{\sim}{x} \in S$

Then call x_i to be the MEDOID of S.

Heararchical Algorithm:-



These algorithms are basically of two types:

(i) Agglomerative Algorithm:-

Let $S = \{\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_n\} \subset \mathbb{R}^m$ be given & d be the dissimilarity measure. Algorithm is given below:

1. We have n clusters $C_1 = \{\underset{\sim}{x}_1\}$, $C_2 = \{\underset{\sim}{x}_2\}, \ldots, C_n = \{x_n\}$
2. Clusters at level i — $C_1, C_2, \ldots, C_{n-i+1}$
   Merge two clusters C_i, C_j if $D(C_i, C_j) < D(C_{i_1}, C_{j_1}), \forall i_1, j_1$
   (one cluster is reduced)
   Rename the clusters as $C_1, C_2, \ldots, C_{n-i}$
3. Repeat step 2 till the required no. of clusters is obtained.
4. Stop the method when the number of obtained clusters is K.

Q. How to define D?

Ans:-
- $D(A, B) = \min\limits_{\substack{x \in A \\ y \in B}} (d(\underset{\sim}{x}, \underset{\sim}{y})) \longrightarrow$ Single linkage
- $D(A, B) = \max\limits_{\substack{x \in A \\ y \in B}} (d(\underset{\sim}{x}, \underset{\sim}{y})) \longrightarrow$ Complete linkage

  [D is not metric bcoz $D(0) \not\Rightarrow \underset{\sim}{x} \neq \underset{\sim}{y}$ in both these cases.]
- Several other such 'D's can be considered.
- Single linkage provides non-convex clustering generally.
- D is a set here, this is not a metric. Hausdorff distance between sets is a metric though.

→ So, in Agglomerative method, ititially we assume n-clusters, and in each iteration we merge two clusters in this method.

(ii) Diversive Method:-

Here we assume initially that we have a single cluster. In every iteration, one of the existing cluster is chosen, and it is divided into two parts.

- Single Linkage is easy to understand and has some good mathematical property. Single linkage minimizes the maximum dissimilarity, kind of optimistic way of looking at life.
- Complete Linkage maximizes the minimum dissimilarity, kind of pessimistic view.

Remarks:-

1. If $S \subseteq \mathbb{R}^m$ and d denotes Euclidean distance, then $D(A,B)$ in both the cases are not metric.

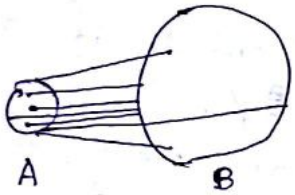
2. Let $\mathcal{C}$ = Set of all non-empty compact subset of $\mathbb{R}^m$.
   d = Euclidean distance

$$D : \mathcal{C} \times \mathcal{C} \to [0, \infty)$$

Define $d(\underset{\sim}{x}, A) = \inf_{\underset{\sim}{y} \in A} d(\underset{\sim}{x}, \underset{\sim}{y}) \quad \forall\, x \in \mathbb{R}^m,\ A \in \mathcal{C}$

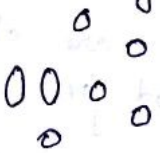
$$D(A,B) = \text{Max} \left\{ \sup_{\underset{\sim}{x} \in B} d(\underset{\sim}{x}, A),\ \sup_{\underset{\sim}{y} \in A} d(\underset{\sim}{y}, B) \right. \quad \forall\, A, B \in \mathcal{C}$$

[Note: If you are dealing with finite set, Sup is same as maximum
Inf is same as minimum.]



Then D is a metric on $\mathcal{C}$. D is known as <u>HAUSDORFF distance</u>

3. Single Linkage algorithm says you find the dissimilarities between every pair of clusters. The one for which the dissimilarity is minimum, you will merge those two clusters.



4. Complete linkage says between two clusters what can be the maximum amount of dissimilarity and you are minimizing it.

5. One can use other choice of D in the above method, example:

$$D(A,B) = \frac{1}{|A||B|} \sum_{\underset{\sim}{x} \in A} \sum_{\underset{\sim}{x} \in B} d(\underset{\sim}{x}, \underset{\sim}{y})$$

6. Note that single linkage is susceptable to noise.

Assignment IV:- Apply k-means algorithm on satelite image data for $k = 3, 4, 5, 6$ and 7.

Papers to read:- Biometrika (1973)
"Admissible Clustering Procedures" by L. Fisher and J. Van Ness.

DBScan :- Easy, Most popular clustering method.
"JASA" (1973): "Probabilistic theory of Cluster analysis" by R.F. Jing.

Let $S = \{\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_n\} \subseteq \mathbb{R}^n$; $d(\underset{\sim}{x}, \underset{\sim}{y})$ denotes dissimilarity between $\underset{\sim}{x}$ and $\underset{\sim}{y}$.

This algorithm automatically choose the number of clusters. See below-

(i) Choose values for two constants $r > 0$ and +ve integer θ.

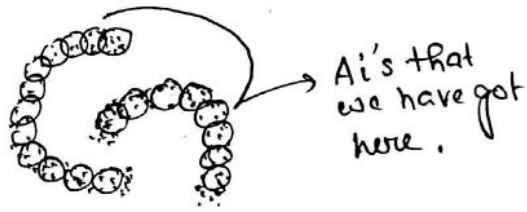
(ii) Let $A_i = \{ \underset{\sim}{y} \in S : d(\underset{\sim}{x}_i, \underset{\sim}{y}) \leq r \}$; $i = 1, 2, \ldots, n$.

(iii) Let $a_i = |A_i|$; $i = 1, 2, \ldots, n$.

(iv) Merge two clusters A_i and A_j if $a_i > \theta$ and $a_j > \theta$ and $A_i \cap A_j \neq \phi$.

(v) Repeat step (iv) till no more merges take place.

Q. How to choose values of r and θ?



- If r and θ are chosen properly, this method gives good result usually.
- If $r = 2$, you will get single linkage method.

General comments on Clustering:-

— More theory needs to be developed in the case of many methods.

— As $n \to \infty$, single linkage gives better result, where n is the number of points.

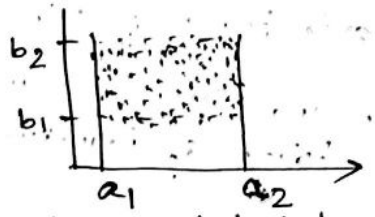
- **Examples of uses or Application of Pattern Recognition :-**

Suppose you are going to cross a road. Speed of the car coming from left/right is not known, that's why an illiterate person/child can cross the road. Now you want to write a program for crossing the road? The input is not precise and the logic is not precise; even then you can cross the road...

Human being can make judgement in these situations. To model this imprecisions, ambiguity and to make a system so that it can work, we need Pattern recognition.

Note that, the output need not to be unique.

- **When to do Clustering?**



So, it's uniformly distributed data which has no clusters. So, if you have a data uniformly distributed, you can't have any clusters.

- So, before applying clustering, check whether there are clusters or not. One way of checking it whether data follows uniform distribution or not.

[Ref: Algorithm for Clustering Data by Jain & Dubes]

- If the data is unimodal,  then we have only one cluster, then no need of clustering.

-  If the data is something like this (non-uniform, more than one mode) you can use clustering.

FEATURE SELECTION (Variable selection)

- Let us consider an example of heights of Punjabi and south indian communities. One of the distinguishing features which seperates these two communities is heights. It's based on our observation. Height, weight are some of the features to distinguish between Punjabi and South Indian. But this is based on our assumption.

— How does the data will tell us that these features are really distinguishing features? We need to write an algorithm for it.

— The feature selection method should automatically feature height as a distinguishing feature between these two communities.

- People living in hilly regions usually have shorter noses than nests. So, the length of the nose is a distinguishing feature. This is another problem of feature selection.

USES:-

1. Reduction in computational complexity.
2. Redundant feature act as a noise. So, feature selection can be looked upon as a noise removal step. ← Supervised case
3. Feature selection provides insight into the classification problem. ← Unsupervised case

Let $S = \{X_1, X_2, \ldots, X_M\}$ be the set of features (or, variables) under consideration. We are supposed to select m features, $m < N$, from S.

Let us assume that m is known. (Usually based on some constraints or computationally problem, m is partially known, but in real life there can be situation where information about m is not known at all)

- Example of Redundant feature:- 1. $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$; $x_2 = 2x_1 - 1$

x_1	x_2
1	1
3	5
−10	−21

So, we can remove one of the feature.

2. (Non-linear relationship case)

$$X_2 = 2X_1^3 - 10X_1^2 + 5X_1 - 7$$

$$X_1 = ?$$

Unique both way relationship may not always exist.
Here we can say X_2 is redundant, but not X_1.
So, ambiguity exists where there is a non-linear relationship.

[Numerical analysis concepts say that there is always a polynomial relationship between two variables (data set is given) of degree $n-1$.]

- Correlation coefficient is not a very good measure of relationship.

$X_1 \in \{-1, 0, 1\}$
$X_2 \in \{0, 1\}$

Y \ X	-1	0	1	
0	0	1/2	0	1/2
1	1/4	0	1/4	1/2
	1/4	1/2	1/4	1

$P(Y=0) = 1/2 = P(Y=1)$

$P(X=0) = 1/2$; $P(X=-1) = P(X=1) = 1/4$.

Here $r_{XY} = 0$ but X and Y are dependent, i.e. $Y = X^2$.

— How do we know two features are related? Measure of it? This problem is also there in Feature Selection.

— This is a research issue. There are many papers on measuring the relationship between two features. The problem is still relevant.

<u>Basic Steps of Feature Selection:-</u> 1. Objective function J which attaches a value to every subset of features is to be defined. 2. Algorithms for feature selection are to be formulated.

<u>Algorithm:-</u> Let $S = \{X_1, \ldots, X_M\}$ be set of features.
Selecting m features (m is known) where $m < M$ from S.

Let $M = 100$, $m = 10$. Total number of possible subsets $= \binom{100}{10} > 10^{12}$

- Are we in a position to go through all these subsets to find optimal one? (<u>ANS</u>: NO)
- Suppose I don't search the whole space, can I guarantee I will get the optimal one? (<u>ANS</u>: NO)

— So, there exists no feature selection algorithm which provides optimal feature subset for "<u>any</u>" criterion function without doing exhaustive search.

* Whatever optimum algorithm we can find in literature, optimal means you are able to get optimal set of features without doing exhaustive search, there they are using the properties of the criterion function. *

— Let $P(S)$ denote power set of S; $P(S)$ has then 2^M elements.
Let $A_m = \{B : B \subseteq S,\ B \text{ has } m \text{ elements}\}$
Let $J : P(S) \to (-\infty, \infty)$;
Let us assume J is known as objective function (criterion function) for feature selection.
We need to find $B_0 \in A_m$ such that $J(B_0)$ is optimal.
W.L.G. Let us assume that J is to be maximized, i.e.,

$$J(B_0) \geq J(B) \ \forall\ B \in A_m.$$

Cardinality of $A_m = \binom{M}{m}$.

For minimization problem: $J(B_0) \leq J(B) \ \forall\ B \in A_m$.

[Example:- We know that Bayes decision rule which minimize the probability of misclassification. Now we are supposed to find m number of features for which the prob. of misclassification is minimized.

Let us take $M = 10$, we have total 10 features.
Also choose $m = 2$, so, ${}^{10}C_2$ subsets we can have.
Considering one subset $\{X_1, X_2\}$ as feature, then we can find the value of Prob. of misclassification.
Like that for ${}^{10}C_2 = 45$ such subsets we can get the value of the minimum probability of misclassification.
Now within all these subsets which one has minimum of minimum prob. of misclassification is the J.]

[Minimum prob. of misclassification means:
for X_1 & X_2 : two features, You can have many decision rules,
let $X_1 + X_2 \leq 1$ or $X_1 + X_2 > 1$ or etc.
Now, for which one the prob. of misclassification is minimum, consider that one.]

- Initially, we are assuming J is given, and trying to find out the algorithm.

— If we know some properties of the function J, then many times we can find out an algorithm which provide you the optimal feature subset without doing the exhaustive search.

[Paper:- TM Cover, Campenhout (1973), IEEE Transaction Inf. Theory.
" The two best features need not to be the best two features".
— Consider four features: X_1, X_2, X_3, X_4. J is the criterion function

$$J(\{X_1\}) > J(\{X_2\}) > J(\{X_3\}) > J(\{X_4\})$$

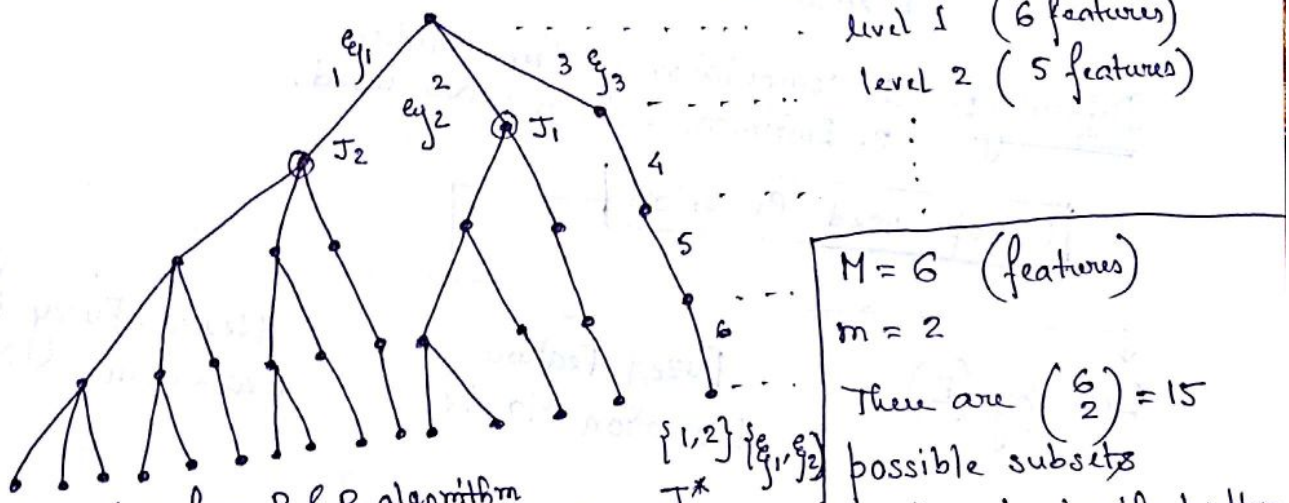
$$J(\{X_3, X_4\}) \geq J(\{X_i, X_j\}) \ \forall\ i \neq j.$$]

There are some algorithms which provide best set of features without doing exhaustive search for some criterion functions. One such algorithm is "BRANCH & BOUND ALGORITHM" by Narendran and Fuknaga (IEEE Paper / Fuknaga Book), for feature selection. This algorithm assumes the following:

$S = \{X_1, X_2, \ldots, X_M\}$

$J(A) \le J(B)$ if $A \subseteq B \subseteq S$.

B & B Algorithm application is difficult to use when M is large & assumption is not satisfied.



M = 6 (features)
m = 2
There are $\binom{6}{2} = 15$ possible subsets
= # nodes in the bottom layer.

Fig:- Solution tree for B & B algorithm feature selection method for M=6, m=2.

$J_1 < J^*$ and $J_2 < J^*$; then $J^*\{1,2\}$ is the best feature subset.

Q. How to construct the tree? How does one decide the number of branches of a particular node?

Ans:- Number of levels = $(M - m + 1)$.

(Number of branches from a node) + (No. of features to be preserved at that node) = $m+1$.

* The tree is constructed from right to left.*

Choose $e_{y_1}, e_{y_2}, e_{y_3}$ (3 features) randomly from S.

$$J(S - \{e_{y_1}\}) \le J(S - \{e_{y_2}\}) \le J(S - \{e_{y_3}\})$$

Homework:- Write the solution tree for Branch and Bound feature selection scheme for the following two cases:

(i) M = 7, m = 2 ; (ii) M = 7, m = 3.

Objective Function for Feature Selection

Supervised Criterion
- Interclass distance Measures
- Probabilistic Separability Measures
- Information Theoretic measures

Difficulty:- 1. Computation of Probabilities.
2. Empirical estimates are used.

Unsupervised Criterion
- Entropy (E)
- Fuzzy Feature Evaluation Matrix
- Neuro - Fuzzy Feature Evaluation Matrix

Algorithms for Feature Selection
- → Sequential Forward Selection
- → Sequential Backward Selection
- → (l, r) algorithm
- → Feature Ranking
- → Branch & Bound Algorithm.

- **Reference Book:-** Devijver and Kittler, Pattern Recognition: A Statistical Approach, Englewood Cliffs, 1982.

$\begin{pmatrix} 160cm \\ 65kg \end{pmatrix} \begin{pmatrix} 158cm \\ 64kg \end{pmatrix} \quad \begin{pmatrix} 1.60m \\ 65000gm \end{pmatrix} \begin{pmatrix} 1.58m \\ 64000gm \end{pmatrix}$

$$\underset{\sim}{X}'A\underset{\sim}{X} = \left((x_1-y_1)(x_2-y_2)\right) A_{2\times 2} \begin{pmatrix} (x_1-y_1) \\ (x_2-y_2) \end{pmatrix}$$

A is a p.d. matrix. $\sqrt{\underset{\sim}{X}'A\underset{\sim}{X}}$ satisfies all the properties of a metric.

Some Algorithm for feature Selection:—

Let $S = \{x_1, \ldots, x_M\}$ be given set of features. We need to choose m features from S where m is known, the criterion function J is also need to be maximised.

1. **Feature Ranking**:— Let $\xi_1, \xi_2, \ldots, \xi_M \in S$ be such that

$$J(\{\xi_i\}) \geq J(\{\xi_{i+1}\}) \quad \forall\, i = 1, 2, \ldots, M-1.$$

Then take $\{\xi_1, \ldots, \xi_m\}$ as the reduced set.

2. **Sequential Forward Selection**:— Let $A_0 = \phi$

Let A_K denotes the selected K features.

Let $\xi_0 \in S - A_K$ be such that,

$$J(A_K \cup \{\xi_0\}) \geq J(A_K \cup \{\xi\}) \quad \forall\, \xi \in S - A_K.$$

Then $A_{K+1} = A_K \cup \{\xi_0\}$.

Run the loop till m features are selected where we start from A_0.

3. **Sequential Backward Selection**:—

Let $\bar{A}_0 = S$

Let $\bar{A}_K$ denote the set containing $(M-K)$ features.

Let $\xi_0 \in \bar{A}_K$ be such that

$$J(\bar{A}_K - \{\xi_0\}) \geq J(\bar{A}_K - \{\xi\}) \quad \forall\, \xi \in \bar{A}_K.$$

Then $\bar{A}_{K+1} = \bar{A}_K - \{\xi_0\}$.

Start from $\bar{A}_0$ and stop at $\bar{A}_{M-m}$.

(4) <u>Generalised Sequential Forward Selection Algorithm (GFS(r))</u>:

Let r is the integer. It is taken in such a way that $\frac{m}{r}$ is also an integer.

Let $A_0 = \phi$

Let A_K denote the set of K selected features.

Let $\{\xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_r}\} \subseteq S - A_K$ be such that

$$J(A_K \cup \{\xi_{j_1}, \xi_{j_2}, \ldots \xi_{j_r}\}) \geq J(A_K \cup \{\xi_{j_1}, \ldots, \xi_{j_r}\})$$

$$\forall \{\xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_r}\} \subseteq S - A_K.$$

Then, $A_{K+r} = A_K \cup \{\xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_r}\}$.

Start with A_0 and run loop $\frac{m}{r}$ times.

<u>Choice of Criterion Function</u>:- Let $S = \{X_1, \ldots, X_M\}$ be the given set of features, m is known.

We need to choose m variables from S, $m < M$, m is known,

$$A_m = \{B : B \subseteq S, B \text{ has } m \text{ elements}\}$$

1. Let there be 'c' classes. Choose those m features for which the misclassification probability corresponding to <u>Bayes decision rule</u> is minimum.

2. Let the number of classes be 2. We know P_1, P_2 and p_1, p_2.

We need to define a function g on p_1 and p_2 $\ni$

$$g(p_1, p_2) = h_2\left(\int h_1(p_1, p_2)\right) \text{ in the following way:}$$

(a) g is some "minimum value" if $p_1(\underset{\sim}{x}) = p_2(\underset{\sim}{x}) \; \forall \underset{\sim}{x}$.

(b) g is some "maximum value" if $p_1(\underset{\sim}{x}) > 0 \Rightarrow p_2(\underset{\sim}{x}) = 0$, <u>or</u>, $p_2(\underset{\sim}{x}) > 0 \Rightarrow p_1(\underset{\sim}{x}) = 0$.

(c) g takes values in between the maximum & minimum value otherwise.

[Reference Book:- Devijver and Kittler, Pattern Recognition: A Statistical Approach, Englewood Cliffs, 1982.]

Examples of Probabilistic separability Measures:-

(i) Bhattacharyya Distance: (By Anil Bhattacharyya)

$$J_B(p_1, p_2) = -\log \int \sqrt{p_1(\underset{\sim}{x}) p_2(\underset{\sim}{x})}\, d\underset{\sim}{x}$$

When $p_1 = p_2$; $J_B(p_1, p_2) = -\log \int p_1(\underset{\sim}{x})\, d\underset{\sim}{x} = -\log 1 = 0$.

(ii) By Chernoff:- [Also read Chernoff's faces]

$$J_C(p_1, p_2) = -\log \int p_1^{s}(\underset{\sim}{x})\, p_2^{1-s}(\underset{\sim}{x})\, d\underset{\sim}{x} \quad \text{for some } s,\ 0 < s < 1.$$

(iii) Jeffries - Matusita Distance:-

$$J_M(p_1, p_2) = \int \left(\sqrt{p_1(\underset{\sim}{x})} - \sqrt{p_2(\underset{\sim}{x})}\right)^2 d\underset{\sim}{x} = 2\left(1 - e^{-J_B}\right)$$

→ There are many such measures of separability for classification.

⊙ One need to find the values of J for each of $\binom{M}{m}$ subsets of S. Choose that subset for which the J value is maximum.

If the number of classes is K with prior probabilities $P_1, P_2, \ldots, P_K$ and the class pdfs $p_1, p_2, \ldots, p_K$.

Let the mixture probability p be $p(\underset{\sim}{x}) = \sum_{i=1}^{K} P_i p_i(\underset{\sim}{x})$

Let $I_B = \sum_{i=1}^{K} P_i \left(-\log \sqrt{p_i(\underset{\sim}{x})\, p(\underset{\sim}{x})}\, d\underset{\sim}{x}\right)$.

Similarly, we can define I_C and I_M.

⊙ Suppose the given data points are $\underset{\sim}{x}_{ij}$; $j = 1, 2, \ldots, n_i$, $i = 1, 2, \ldots, K$.

x_{ij} denotes the jth data point in the ith class.

Let P_i denotes the prior probability of the ith class.

Let B be a subset containing m variables.

Let $\delta(\underset{\sim}{x}_{i_1 j_1 B}, \underset{\sim}{x}_{i_2 j_2 B})$ denotes the distance between $x_{i_1 j_1 B}$, $x_{i_2 j_2 B}$.

class point variable

$$J(B) = \frac{1}{2} \sum_{i_1=1}^{K} \sum_{i_2=1}^{K} P_{i_1} P_{i_2} \frac{1}{n_{i_1} n_{i_2}} \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \delta(x_{i_1 j_1 B}, x_{i_2 j_2 B}).$$

Suppose $\delta(\underset{\sim}{x}_{i_1 j_1 B}, \underset{\sim}{x}_{i_2 j_2 B}) = (\underset{\sim}{x}_{i_1 j_1 B} - \underset{\sim}{x}_{i_2 j_2 B})' (\underset{\sim}{x}_{i_1 j_1 B} - \underset{\sim}{x}_{i_2 j_2 B})$

Let $\bar{\underset{\sim}{x}}_{iB} = \frac{1}{n_i} \sum_{j=1}^{n_i} \underset{\sim}{x}_{ijB}$, $\bar{\underset{\sim}{x}}_{B} = \sum_{i=1}^{K} P_i \bar{\underset{\sim}{x}}_{iB}$

$$J(B) = \underbrace{\sum_{i=1}^{K} P_i \cdot \frac{1}{n_i} \sum_{j=1}^{n_i} (\underset{\sim}{x}_{ijB} - \bar{x}_{iB})' (\underset{\sim}{x}_{ijB} - \bar{x}_{iB})}_{\text{Within Distance}} + \underbrace{\sum_{i=1}^{K} P_i (\bar{\underset{\sim}{x}}_{iB} - \bar{\underset{\sim}{x}}_{B})' (\bar{\underset{\sim}{x}}_{iB} - \bar{\underset{\sim}{x}}_{B})}_{\text{Between Distance}}$$

We can have another criterion function as J_1, where

$$J_1(B) = \frac{\sum P_i (\bar{x}_{iB} - \bar{x}_{B})' (\bar{\underset{\sim}{x}}_{iB} - \bar{\underset{\sim}{x}}_{B})}{\sum P_i \frac{1}{n_i} \sum_{j=1}^{n_i} (x_{ijB} - \bar{x}_{iB})' (x_{ijB} - \bar{x}_{iB})}$$

Maximize J_1 over B.

Comparison between performance of classifiers:-

1. **Leave-one-out method:-** $A = \{\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n\} \subseteq \mathbb{R}^N$

Let θ_i denotes the class (label) of $\underset{\sim}{x}_i$, $i = 1(1)n$.

Number of classes $= c$

$\theta_i \in \{1, 2, \ldots, c\}$ $\forall\, i = 1(1)n$.

Sum = 0

For $i = 1, 2, \ldots, n$

$B_i = \{\underset{\sim}{x}_i\}$,

$E_i = A - B_i$

Take E_i as training set, develop the classifier.

Check if $\underset{\sim}{x}_i$ is classified correctly using the classifier.

If the classification is wrong then Sum = Sum + 1.

End for.

2. **K-fold Cross Validation:-** $A = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^N$

θ_i denotes the class of $\underset{\sim}{x}_i$

No. of classes is c.

$\theta_i \in \{1, 2, \ldots, c\}$ $\forall\, i = 1(1)n$.

Let $A_2, A_2, \ldots, A_K$ be such that (i) $A_i \neq \phi$ $\forall\, i = 1, 2, \ldots, K$

(ii) $A_i \cap A_j = \phi$ if $i \neq j$

(iii) $\bigcup_{i=1}^{K} A_i = A$

(iv) sizes of A_i's are more or less same.

For $i = 1, 2, \ldots, K$

$B_i = A - A_i$

Take B_i as training set, and A_i as test set.

Find the misclassificate rate.

End for.

K – misclassification rates:
mean and standard deviation of these K-rates.

Remark:-

1. Here, we are trying to compare different classification schemes.

2. Leave-one-out method is an exhaustive method. At a time we are leaving out one point and you are doing that for every point in the data set. But in K-fold cross validation people might be interest in knowing how $A_1, A_2, \ldots, A_K$ are chosen.

3. If n is large, leave-one-out gives bad result. In those cases K-fold cross validation is useful.

Data Condensation:—

- Data sets often contain redundant data.
- Replace a large dataset by a small subset of representative patterns.
- Performance of Pattern Recognition algorithms when trained on the reduced set should be comparable to that obtained when trained on the entire data set.

Methods:—

- Statistical Sampling:
  - Random sampling (WOR, WR) (poor result for noisy data and sparse sampling ratios)
  - Stratified sampling (weightage to weak classes)

Ques:- What is the size of the reduced data set?

→ A way to do it is clustering. Like K-means clustering. So, it's one way to handle data.

Astrahan's Method:- (1970)

1. Select two radii d_1 and d_2.
2. For every point in the data set, find the number of other points lying within d_1 distance of it.
3. Find the point having the highest number of point in its d_1 neighbourhood.
4. Retain the above point in the reduced set.
5. Discard all points from the dataset lying within a distance d_2 from the selected point. Repeat till the dataset is exhausted.

Question:- How to choose d_1 and d_2?

Note:- Retains noise points in the reduced sets.
To achieve noise tolerance Aha (1991) suggested a modified version, the IB3 (Instance Based Learning algorithm).

Learning Vector Quantization Method:-

Use a set of codebook vectors to obtain a reduced representation of the data, such that squared quantization error is minimized.

LVQ1: m_c : a codebook vector, x : a data point

1. Choose a random set of codebook vectors.
   For each x in the dataset —
   Assign it to the class of closest codebook vector.
2. Update the codebook vectors, as:

$$m_c(t+1) = m_c(t) + \alpha(t)\,[x(t) - m_c(t)] \text{ if } x \text{ is correctly classified.}$$

$$m_c(t+1) = m_c(t) - \alpha(t)\,[x(t) - m_c(t)] \text{ if } x \text{ is misclassified.}$$

$\alpha(t)$: learning rate, critical for convergence.

— More sophisticated versions of LVQ exist.

Reference:- T. Kohonen, The self-organising Map, Proc. IEEE, Vol 78, 1990, pp 1464-1480.

Density Based Multiscale Condensation:- [IEEE TPAMI 24 (6), 2002]

1. Select an integer K.
2. For every point x_i in the dataset, find its distance to the Kth nearest neighbour, denote it by (r_i).
3. Select the point having lowest value of r_i.
4. Remove all the points lying within $2r_i$ of a selected point.
5. Repeat steps 2-4 till the dataset is exhausted.

Remarks:-

1. Provides detailed representation of denser regions of feature space and lenient representation of sparser regions (Multi-resolution representation).
2. Based on K-nearest neighbor nonparametric density estimation.
3. Different 'scales' of detail achieved by varying value of K.
4. Does not require choice of radii d_1, d_2 as in Astrahan's method.

Evaluation Criteria:- Goodness of reduced set is measured by the difference of nonparametric density estimates obtained using the original dataset and the reduced set.

If $g_1(x)$ and $g_2(x)$ are the estimates, error J:

$$J = \frac{1}{N}\sum_{i=1}^{N} D\left(g_1(x), g_2(x)\right) \quad ; \quad N: \text{Number of data points}$$

Distance D between two distributions:

$$D\left(g_1(x), g_2(x)\right) = \left| \ln \frac{g_1(x)}{g_2(x)} \right| \quad : \text{Log-likelihood ratio (LLR)}$$

$$D\left(g_1(x), g_2(x)\right) = \left| g_2(x) \ln \frac{g_1(x)}{g_2(x)} \right| \quad : \text{Kullback-Liebler information number (KLI)}$$

Satelite Image: (IRS image of Kolkata)

Number of samples: 262144

Number of features: 4 (spectral bands 0-255)

Task: clustering

Result: Density Estimation:-

Forest Covertype:

Method	CR%	LLR	KLI
Multiscale ($K=157$)	0.1	0.82	2.71
Astrahan	0.1	2.0	4.7
Random Sampling	0.1	3.8	7.0

CR: condensation ratio (condensed/actual);
LLR: log-likelihood ratio;
KLI: Kullback-Liebler information number.

Unsupervised Feature Selection using feature similarity:-

Mitra, Murthy and Pal, IEEE TPAMI, 24(3): 301-312, 2002.

(This is also called clustering of features)

One way is correlation coefficient which measure dissimilarity. But a problem with this is if θ shifts, r_{xy} will change.

Contribution two fold:

a. Feature Similarity Measure: Maximal Information Compression Index λ_2.

$$\lambda_2(F_1, F_2) = \text{minimum eigenvalue of } Cov(F_1, F_2).$$

Properties:- • If F_1 and F_2 are linearly related $\lambda_2 = 0$.
• Measures the error in terms of second order statistics under maximal information compression.

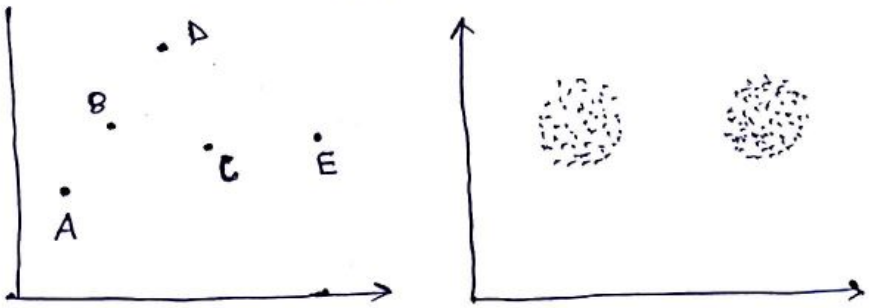
Advantages:- • Symmetric: $\lambda_2(F_1, F_2) = \lambda_2(F_2, F_1)$
• Invariant to rotation of the scatter plot.

[Read Kohonen's self organising map which explains reducing data dimensions preserving the topology of the dataset]

↓

Data Visualisation techniques (another example: Chernoff's faces)

Visualization and Aggregation :-



If the data is in $\mathbb{R}^2$, we can easily visualise the data well. Dimensionality reduction (like, Principal Component Analysis) is one way to handle data of higher dimension. But preserving the topology of the data set may not hold always.

* CHERNOFF FACES
* KOHONEN'S MAP

Ensemble Classifiers :- We have been discussing about which classifier is better and which one is not better on data sets. Now, can we combine two or more classifier and construct a new classifier?

— There is many ways it can be done.

— One way is : " BAGGING " , another way is " BOOSTING ".

— Bagging [Brieman 1996] is a bootstrap ensemble method that creates individuals for its ensemble by training each classifier on a random redistribution of the training set. Each classifier's training set is generated by random sampling with replacement.

— Boosting [Freund Schapire 1996] encompasses a family of methods. The training set used for each member of the series is chosen based on the performance of the earlier classifier(s). In boosting, examples that are incorrectly predicted by previous classifiers in the series are chosen more often than examples that were correctly predicted. Thus Boosting attempts to produce new classifiers that are better able to predict examples for which the current ensemble's performance is poor.

Neural Networking

Perceptron Learning Algorithm:- Let $(\underset{\sim}{x}_i, \theta_i); i=1,2,\ldots,n$ be given where $x_i \in \mathbb{R}^M \ \forall\ i=1(1)n$ and θ_i denotes the label of $\underset{\sim}{x}_i$. Let us assume that the number of classes is 2.

So, $\theta_i \in \{1,2\} \ \forall\ i=1,2,\ldots,n$.

Assumption:- $\exists\ \underset{\sim}{a} \in \mathbb{R}^M$ and $a_{M+1} \in \mathbb{R} \ni$

$$a'\underset{\sim}{x}_i + a_{M+1} > 0 \quad \forall\ i \text{ for which } \theta_i = 1$$
$$< 0 \quad \forall\ i \text{ for which } \theta_i = 2$$

Question:- How does one get one of the seperating hyperplanes?

Ans:- Let $A_1 = \{x_i : \theta_i = 1\} \neq \phi$

$A_2 = \{x_i : \theta_i = 2\} \neq \phi$

Check: $\text{Conv}(A_1) \cap \text{Conv}(A_2) = \phi \Leftrightarrow$ the assumption holds.

Let $\underset{\sim}{x}_i' = (x_{i1}, x_{i2}, \ldots, x_{iM}) \ \forall\ i=1,2,\ldots,n$

$\underset{\sim}{y}_i' = (x_{i1}, x_{i2}, \ldots, x_{iM}, 1) \ \forall\ i=1,2,\ldots,n$

Assumption can be reformulated as $\exists\ a \in \mathbb{R}^{M+1} \ni$

$$\underset{\sim}{a}'\underset{\sim}{y}_i > 0 \quad \forall\ i \text{ for which } \theta_i = 1$$
$$< 0 \quad \forall\ i \text{ for which } \theta_i = 2$$

Let $\underset{\sim}{y}_{Kn+l} = \underset{\sim}{y}_l \quad \forall\ l = 1,2,\ldots,n-1$ and $K=1,2,\ldots$

$\underset{\sim}{y}_{Kn} = \underset{\sim}{y}_n \quad \forall\ K = 2,3,4,\ldots$

$\theta_{Kn+l} = \theta_l \quad \forall\ l=1,2,\ldots,n-1$ and $K=1,2,\ldots$

$\theta_{Kn} = \theta_n \quad \forall\ K = 2,3,4,\ldots$

Let $\lambda_1, \lambda_2, \ldots$ be positive constants.

Let $\underset{\sim}{\omega}_1 \in \mathbb{R}^{M+1}$; let, $\underset{\sim}{\omega}_{K+1} = \underset{\sim}{\omega}_K + \lambda_K \underset{\sim}{y}_K$ if $\underset{\sim}{\omega}_K' y_K \leq 0$ & $\theta_K = 1$.

$= \underset{\sim}{\omega}_K - \lambda_K \underset{\sim}{y}_K$ if $\underset{\sim}{\omega}_K' y_K \geq 0$ & $\theta_K = 2$

$= \underset{\sim}{\omega}_K$ otherwise.

Let, $\underset{\sim}{\omega}_K' \underset{\sim}{y}_K \leq 0$ and $\theta_K = 1$.

then, $\underset{\sim}{\omega}_{K+1} = \underset{\sim}{\omega}_K + \lambda_K \underset{\sim}{y}_K$

$$\underset{\sim}{\omega}_{K+1}' \underset{\sim}{y}_K = \omega_K' y_K + \lambda_K' \underset{\sim}{y}_K' \underset{\sim}{y}_K$$

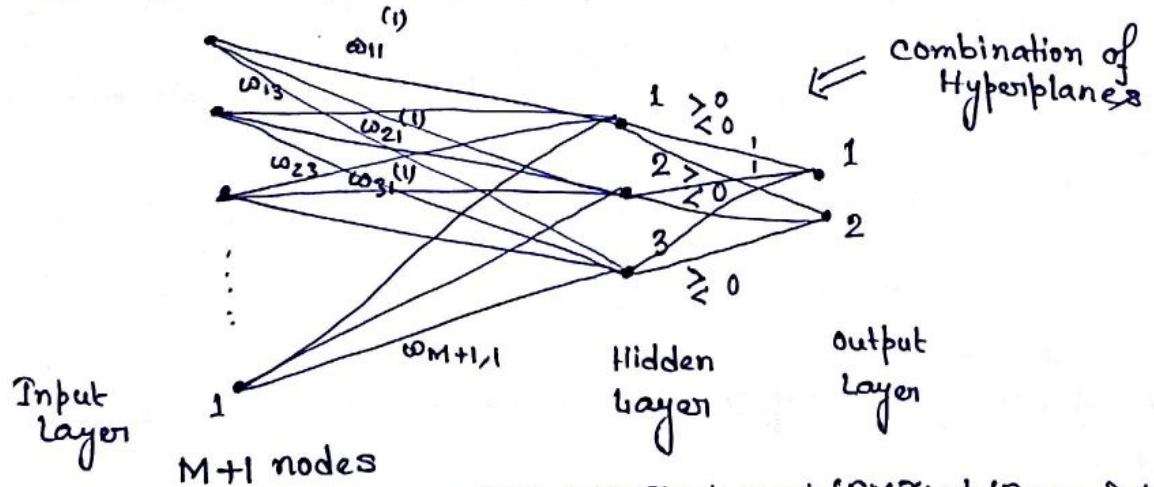
<u>Theorem:-</u> (<u>Perceptron Convergence Theorem</u>)

For any $\lambda > 0$ and for any $\underset{\sim}{\omega}_1 \neq \underset{\sim}{0}$. If $\lambda_1 = \lambda_2 = \cdots = \lambda$ then the algorithm converges, i.e., $\exists$ $\underset{\sim}{\omega}_0 \in \mathbb{R}^{M+1}$ such that

$$\lim_{K \to \infty} \underset{\sim}{\omega}_K = \underset{\sim}{\omega}_0 .$$

— λ is called "learning rate".

Multilayer Perceptron (MLP):-



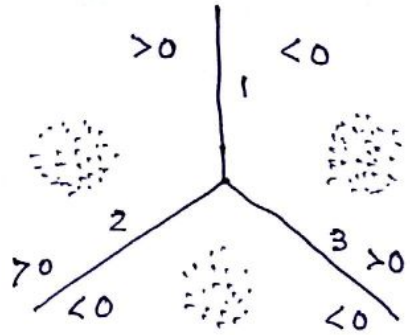
$\omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_M x_M + \omega_{M+1} > 0$ 1

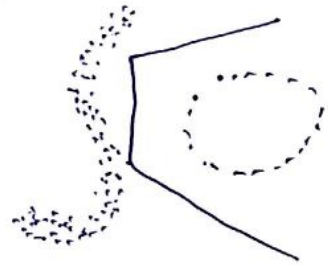
< 0 2

Number of nodes in the input layer $= M+1 = I$

Number of nodes in the output layer $= 2 = L$.

Number of nodes in the hidden layer $= K$.





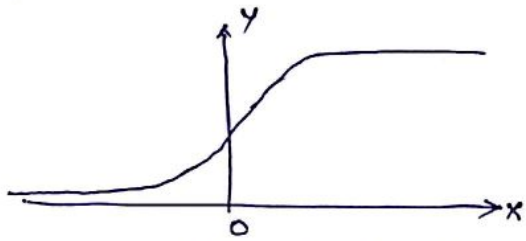
3 hyperplanes

Transfer function:-

$$\frac{1}{1+e^{-x}} \longleftarrow \text{sigmoid function}$$

$$\frac{1}{1+e^{-ax}} \quad ; a>0$$

$$\tanh x = \frac{e^{x}+e^{-x}}{e^{x}+e^{-x}}$$



$w_{ij}^{(1)}$: connection weight of the edge joining the ith node in the input layer to the jth node with the hidden layer.

$i = 1, 2, \ldots, I$

$j = 1, 2, \ldots, K.$

$w_{ij}^{(2)}$: connection weight joining i^{th} node in the hidden layer to the jth node in the output layer.

Let us consider the transfer function as $\frac{1}{1+e^{-x}}$.

Let $\underset{\sim}{x}' = (x_1, \ldots, x_M)$ be an input vector; $x_{M+1} = 1$.

Input for the jth node in the hidden layer is $\sum_{i=1}^{M+1} w_{ij}^{(1)} x_i$; $j = 1, 2, \ldots, K.$

Now, output for the jth node in the hidden layer is

$$y_j = \frac{1}{1+e^{-\sum_{1}^{M+1} w_{ij}^{(1)} x_i}} \; ; \; j = 1, 2, \ldots, K.$$

Input for the jth node in the output layer is $\sum_{i=1}^{K} w_{ij}^{(2)} y_i$; $j = 1, 2, \ldots, L$

Output for the jth node in the output layer is $= z_j$;

$$z_{ij} = \frac{1}{1+e^{-\sum w_{ij}^{(2)} y_i}} \; ; \; j = 1, 2, \ldots, L.$$

Let the target value corresponding to the point x for the jth node in the output layer is t_j.

Error corresponding to the point $\underline{x}$ as $\frac{1}{L}\sum_{j=1}^{L}(t_j - z_j)^2$.

Total error for the training data set is $\frac{1}{n}\sum_{x}\frac{1}{L}\sum_{j=1}^{L}(t_{jx} - z_{jx})^2$

Gradient Descent Technique:-

Let $f: \mathbb{R} \to \mathbb{R}$ be differentiable everywhere. We need to find $a \in \mathbb{R}$ such that $f(a) \leq f(x)\ \forall\ x \in \mathbb{R}$.

Let $x_0 \in \mathbb{R}$ and $x_{n+1} = x_n - \lambda f'(x_n)$ where $\lambda > 0$, $n = 0, 1, 2, \ldots$

<u>Remark:-</u> Let x_0 be such that $f'(x_0) = 0$, then $x_n = x_0\ \forall\ n$ and hence $\lim_{n\to\infty} x_n = x_0$.

- Note that x_0 can be local maxima or minima or a point of inflexion.
- Let $f(x) = x^2$

$x_{n+1} = x_n - \lambda f'(x_n) = x_n - 2\lambda x_n = x_n(1-2\lambda) = x_0(1-2\lambda)^{n+1}$.

i.e., $x_n = x_0(1-2\lambda)^n$.

(1) $0 < \lambda < \frac{1}{2} \Rightarrow \lim_{n\to\infty} x_n = 0$ for every $x_0 \in \mathbb{R}$.

(2) $\lambda = \frac{1}{2}$, $\frac{1}{2} < \lambda < 1$, $\lim_{n\to\infty} x_n = 0\ \forall\ x_0 \in \mathbb{R}$.

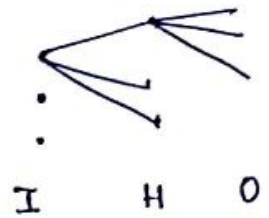
(3) $\lambda = 1$, $\lambda > 1$, $\lim_{n\to\infty} x_n$ does not exist for any $x_0 \in \mathbb{R}$.

- For $f(x) = 3200x^2$, λ has to be very small for gradient descent to converge.
- The whole method depends on how we choose λ.

— Feed forward Neural Network / Back Propagation of Error:

$f : \mathbb{R}^K \to \mathbb{R}$

Let $\underset{\sim}{y_0} \in \mathbb{R}^K$; $y_n' = (y_{n1}, \dots, y_{nK})$



$$y_{n+1} = y_n - \lambda \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_K} \end{pmatrix}_{(y_{n1}, y_{n2}, \dots, y_{nK})}$$

$$\text{Error } (E) = \sum_{j=1}^{L} (t_j - z_j)^2$$

$$\frac{\partial E}{\partial \omega_{i_0 j_0}^{(2)}} = \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial \omega_{i_0 j_0}^{(2)}} = -2(t_{j_0} - z_{j_0}) \cdot \frac{\partial z_j}{\partial \omega_{i_0 j_0}^{(2)}} ;$$

where $$\frac{\partial z_j}{\partial \omega_{i_0 j_0}^{(2)}} = \frac{\partial \left(\frac{1}{1 + e^{-(\Sigma \cdots)}} \right)}{\partial \omega_{i_0 j_0}^{(2)}}$$

$$= \frac{\partial \left(\frac{1}{1 + e^{-(\Sigma \cdots)}} \right)}{\partial (\Sigma \cdots)} \cdot \underbrace{\frac{\partial (\Sigma \cdots)}{\partial \omega_{i_0 j_0}^{(2)}}}_{y_{i_0}}$$

Read *Online Learning* & * Batch mode Learning*.

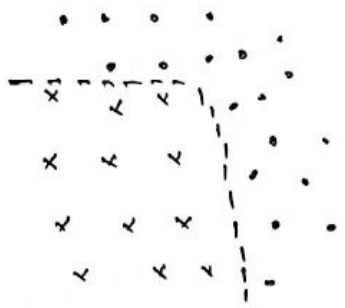
<u>Assignment</u>:- Use online learning & Batch mode learning in Satelite Image data set.

Radial Basis Function Networks:- (Cover, 1965)

"Non Linear transformation to a high dimensional space".

$\downarrow$

points become linearly seperable by a higher probability.



n such points $\rightarrow$ deterministically map to n vertices of $(n-1)$ simplex.

$$\vec{x} = [x_1 \; x_2 \; \cdots \; x_{m_0}] \in \mathbb{R}^{m_0}.$$

N points $\begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_N \end{pmatrix}$

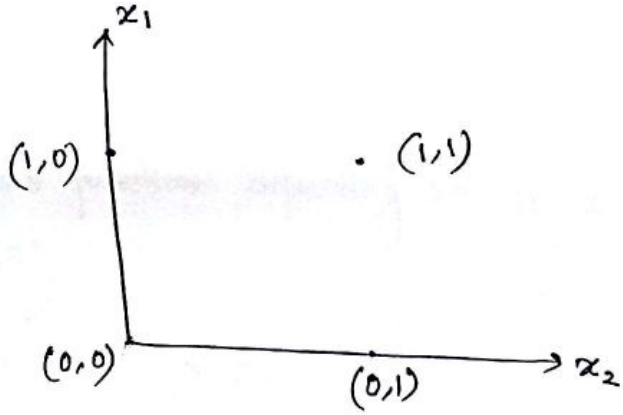
$$\phi(\vec{x}) = [\underbrace{\phi_1(\vec{x}), \phi_2(\vec{x}), \phi_3(\vec{x}), \ldots, \phi_{m_1}(\vec{x})}_{\text{hidden functions}}]$$

; m_1 is usually larger than m_0.

XOR Problem:-

$(0,0)$ and $(1,1) \rightarrow$ Class Ω_1

$(1,0)$ and $(0,1) \rightarrow$ Class Ω_2



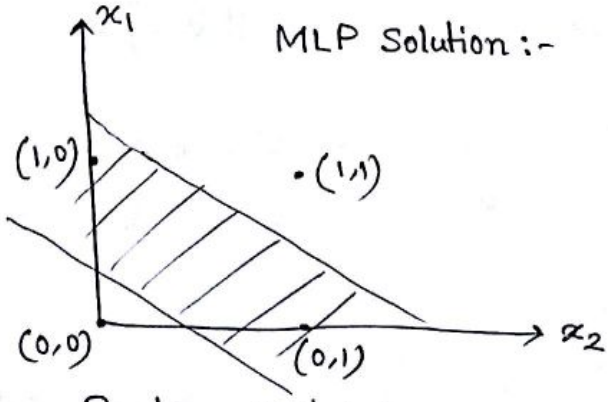
$$\phi_1(\vec{x}) = \exp\left(-\|\vec{x} - \vec{t}_1\|^2\right)$$

$$\phi_2(\vec{x}) = \exp\left(-\|\vec{x} - \vec{t}_2\|^2\right)$$

$$\vdots$$

$$\phi(\vec{x}) \longrightarrow \phi\left(\underbrace{\|\vec{x} - \vec{t}\|}_{\text{radial}}\right)$$

MLP Solution:-



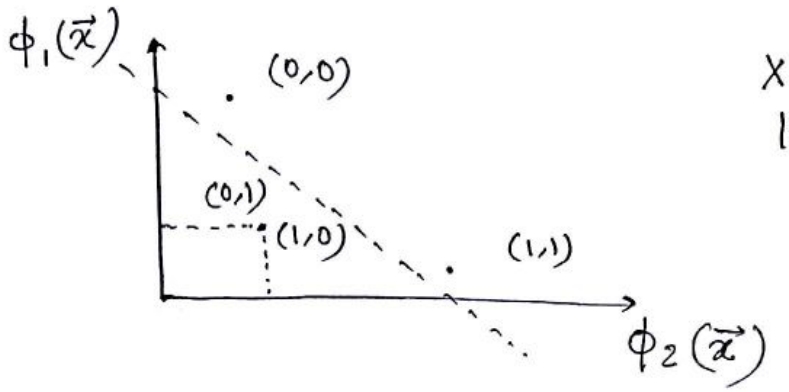
Receptor vectors

$$\vec{x}_1 \longrightarrow \phi(\vec{x}_1) \qquad t_1 = (1,1)'$$

$$\vec{x}_2 \longrightarrow \phi(\vec{x}_2) \qquad t_2 = (0,0)'$$

i/p pattern	First Hidden function	2nd Hidden function
(1,1)	1	$e^{-2} = 0.1353$
(0,1)	$e^{-1} = 0.3678$	0.3678
(0,0)	$e^{-2} = 0.1353$	1
(1,0)	0.3678	0.3678

(1,1) becomes (1, 0.1353)



XOR problem becomes linearly separable.

Each $(x_n, y_n) \in D$ influences the $h(\vec{x})$ (hypothesis function) according to some radial basis functions depending on $\underbrace{\|\vec{x} - \vec{x}_n\|}_{\text{radial}}$

$$h(\vec{x}) = \exp\left(-\gamma \|\vec{x} - \vec{x}_n\|^2\right)$$

Sum up influences from all the points

$$h(\vec{x}_m) = \sum_{n=1}^{N} w_n \exp\left(-\gamma \|\vec{x}_m - \vec{x}_n\|^2\right) = y_m$$

$$\underbrace{\begin{bmatrix} \exp(-\gamma\|\vec{x}_1 - \vec{x}_1\|^2) & \cdots & \exp(-\gamma\|\vec{x}_1 - \vec{x}_N\|^2) \\ \exp(-\gamma\|\vec{x}_2 - \vec{x}_1\|^2) & \cdots & \exp(-\gamma\|\vec{x}_2 - \vec{x}_N\|^2) \\ \vdots & & \vdots \\ \exp(-\gamma\|\vec{x}_N - \vec{x}_1\|^2) & \cdots & \exp(-\gamma\|\vec{x}_N - \vec{x}_N\|^2) \end{bmatrix}}_{\phi} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{y}$$

$\Rightarrow W = \phi^{-1} y \rightarrow$ leads to exact interpolation.

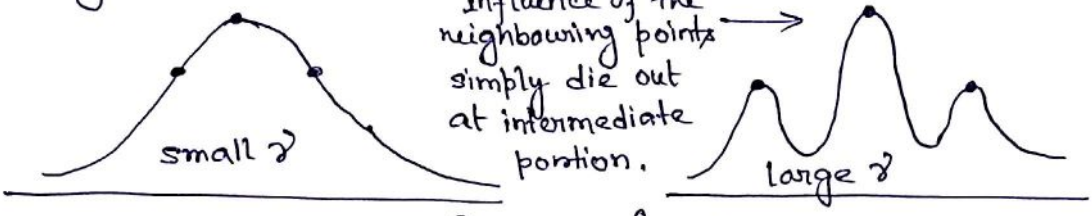
solution is only possible when ϕ is non-singular.

For $\phi_i = \exp\left(-\gamma \| \vec{x} - \vec{x}_i \|^2\right)$ is always non-singular.

$$h(\vec{x}) = \sum_{n=1}^{N} w_n \exp\left(-\gamma \| \vec{x} - \vec{x}_n \|^2\right).$$

Small $\gamma \Rightarrow$ the Gaussian becomes flatter/wider.

Large $\gamma \Rightarrow$ " " " sharper.



Extend RBF model for classification:-

$$h(\vec{x}) = \text{sign}\left(\sum_{n=1}^{N} w_n \exp\left(-\gamma \| \vec{x} - \vec{x}_n \|^2\right)\right)$$

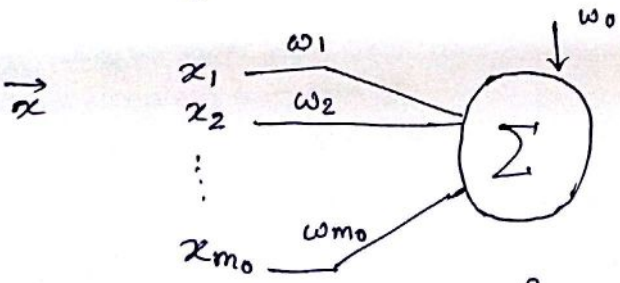
↳ sign as +1 (yes) or −1 (no)

↳ Binary classification.

$$S = \sum_{n=1}^{N} w_n \exp\left(-\gamma \| \vec{x} - \vec{x}_n \|^2\right)$$

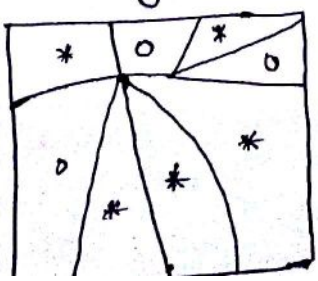
Minimise $\left(h(\vec{x}) - y\right)^2$ on D, $y = \pm 1$.

$h(\vec{x}) = \text{sign}(S)$.



Relationship with NN classifiers:-

— Adopt the y-value of the nearest representative in the training set > 1-NN.



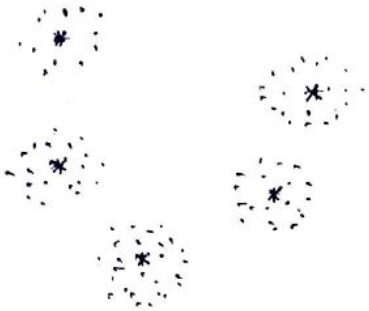
— Essential technique for smoothing the boundaries ⇒ use K-NN.

K NN equivalent basis function should have gradual roll off or delay ⇒ Gaussian (most likely)

RBF with K centers:-

N parameters: $\omega_0, \omega_1, \omega_2, \ldots, \omega_N$ based on N data points.

Using $K (\leq n)$ centres or recepters for our radial basis functions.
$\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_K$ instead of $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N$.



How to choose the k-centres? → without consulting the y_ns.

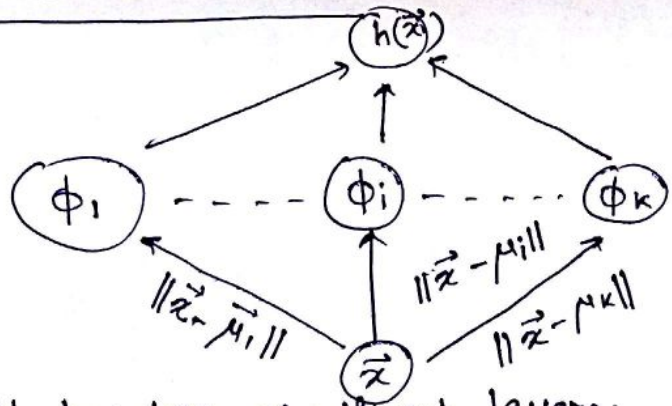
↳ direct K-means clustering algorithm usually employing the lyod's heuristics.

Choose the correct weights so that —

$$\sum_{K=1}^{K} \omega_K \exp\left(-\gamma \| \vec{x}_n - \vec{\mu}_n \|^2\right) \approx y_n$$

N data points, K centres.

Towards RBF Network:-



The features $\exp\left(-\gamma \|\vec{x} - \vec{\mu}_K\|^2\right)$ is non linear transformation in D.

RBFN has two significant layers:

1. A hidden layer of radial non-linear kernels.
   — perform a non-linear mapping to make patterns more linearly seperable.
2. An output layer of linear neurons to perform linear regression to predict the desired targets.

VC Dimension :- (Vapnic-Chernenkis Dimension)

$\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$

Assign '+' & '−' to them $\Rightarrow$ 2^n possible training sets can be made.

A ~~train~~ learning machine $\longrightarrow$ $f(\vec{x}, \vec{\omega}) = y \rightarrow$ the class label.

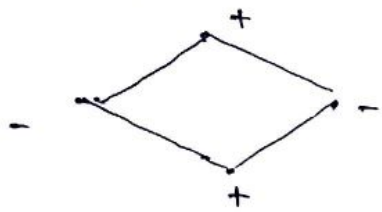
$f(\vec{x}, \vec{\omega}) = \text{sign}(\vec{\omega}^T \vec{x})$ $\qquad \vec{x} \in \mathbb{R}^d$,

$\vec{\omega}$: parameter vector.

$\text{sign}(\omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_0)$ if +ve one class and −ve others.

The VC dimension for a model f is h if $\exists$ at least one set of h points that can be shattered by f but no possible set of $K+1$ points that can be shattered by f.

4 points can't be shattered by set of straight lines.



VC dimension of linear machine is 3,
and VC dimension of sine curve is ∞.

If you have a d-dimensional hyperplane, it's VC dimension is $d+1$.

Disadvantages of VC dimension being high:-

1. No. of trainning samples required to train (PAC bound)

$$m \geq \frac{1}{\epsilon}\left(4\log_2(2/\delta) + 8VC\log_2(13/\epsilon)\right).$$

2. Test error increases:

E_{test} : Test error

E_{train} : training error

$$E_{test} \geq E_{train} + \sqrt{\frac{VC\left(\ln\left(\frac{2m}{VC}\right)+1\right) + \ln\left(\frac{4}{\delta}\right)}{m}}.$$

Support Vector Machine (SVM):-

Statistical learning theory or SVM was invented by Vapnik and Chervonenkis in 1963. When we design a classifier, we have a training set by using it the classifier is designed as tested using test set. So, if the performance of the classifier in test set is fine then we say the classifier is fine. But How do we say it has generalisation property?

Support Vector machine is a by product of "Statistical Learning Theory" (a book by Vapnik).

Statistical Learning Theory:-

- $(\underset{\sim}{x}_i, \theta_i)$: $\forall i = 1, 2, \ldots, n$. given $\underset{\sim}{x}_i \in \mathbb{R}^N$; $\theta_i \in \{-1, 1\}$
- $P(\underset{\sim}{x}, \theta)$ probability distribution on the data.
- P is unknown
- Let $(\underset{\sim}{x}_i, \theta_i)$: be the i.i.d. from $P(\underset{\sim}{x}, \theta)$
- Suppose we have a machine whose task is to learn the mapping $x_i \longrightarrow \theta_i$. Finding $f : \{x_1, \ldots, x_n\} \to \{-1, 1\} \ni f(\underset{\sim}{x}_i) = \theta_i \ \forall i$.

Note that while learning weights in a neural network model for classification with a given architecture, we are dealing with a set of functions.

Let us denote it with $\mathcal{F}$.

$\mathcal{F} = \{ f_{\underset{\sim}{\alpha}} : \underset{\sim}{\alpha}$ belong to some k dimensional space $\mathbb{R}^k \}$

We are trying to minimize $\sum_{i=1}^{n} | f_{\underset{\sim}{\alpha}}(\underset{\sim}{x}_i) - \theta_i |$ over all $\underset{\sim}{\alpha}$.

Let $P(\underset{\sim}{x}, \theta)$ denote the original prob. distribution

$(\underset{\sim}{x}_1, \theta_1), (\underset{\sim}{x}_2, \theta_2), \ldots, (\underset{\sim}{x}_n, \theta_n)$ are i.i.d. sample points from $P(\underset{\sim}{x}, \theta)$. Note that $P(\underset{\sim}{x}, \theta)$ is unknown.

Risk for a function $f_{\underset{\sim}{\alpha}}$ is given by

$$R(f_{\underset{\sim}{\alpha}}) = \frac{1}{2} \int | \theta - f_{\underset{\sim}{\alpha}}(\underset{\sim}{x}) | \, dP(x, \theta).$$

The emperical risk is given by

$$R_{emp}(f_{\underset{\sim}{\alpha}}) = \frac{1}{2}\sum_{i=1}^{n} \left| \theta_i - f(\underset{\sim}{\alpha}, \underset{\sim}{x}_i) \right|$$

[Vapnic (1995)] Let $0 < \eta < 1$,

$$R(f_{\underset{\sim}{\alpha}}) \leq R_{emp}(f_{\underset{\sim}{\alpha}}) + \sqrt{\frac{h\log\left(\frac{2n}{h}+1\right) - \log\left(\frac{\eta}{4}\right)}{n}}$$ is true with prob. $(1-\eta)$.

h is known as Vapnic Chervonenkis (VC) dimension of $\mathcal{F}$.

<u>Definition 1</u>:- A set of l points is said to be shattered by $\mathcal{F}$ if for every labelling of l points if $\exists$ a function $f \in \mathcal{F}$ which provides the labelling.

<u>Definition 2</u>: A set of functions $\mathcal{F}$ is said to have VC dimension h if

(i) $\exists$ a set of h points that can be shattered by $\mathcal{F}$.

(ii) No set of l points can be shattered by $\mathcal{F}$ where $l > h$.

<u>Remarks</u>:-

1. Note that every set of 3 points can't be shattered by $\mathcal{F}$.
   . . . ⟵ collinear points.
2. Note that, no set of 4 points can be shattered by $\mathcal{F}$.
3. In fact the VC dimension of $\mathcal{F}$ is 3.
4. If the space is $\mathbb{R}^M$, $M \geq 2$ and $\mathcal{F}$ denotes the set of hyperplanes then $h(\mathcal{F}) = M+1$.
5. We can get hold of examples where the VC dimension $(\mathcal{F})$ is ∞ but $\mathcal{F}$ can't shatter a 4 point set.

SVM: Let (x_i, θ_i); $i = 1, 2, \ldots, n$ be given where $\underset{\sim}{x}_i \in \mathbb{R}^M \; \forall \; i$. $\theta_i \in \{-1, 1\} \; \forall \; i$. θ_i denotes the label of $\underset{\sim}{x}_i$ for each i.

Let us assume that $\exists$ a hyperplane that seperates the positive points from the negative points.

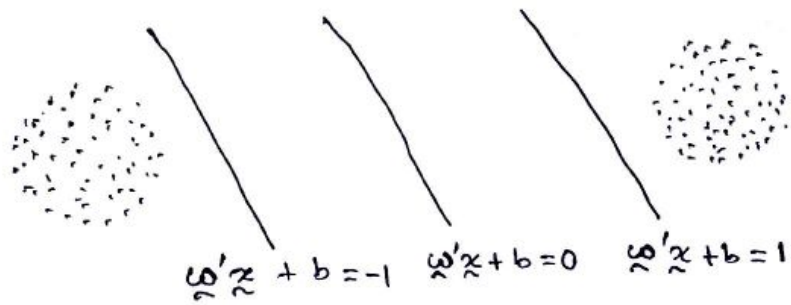
i.e., $\exists \; \underset{\sim}{w} \in \mathbb{R}^M$ and $b \in \mathbb{R} \ni$

$$\underset{\sim}{w}' \underset{\sim}{x}_i + b > 0 \quad \forall \; i \text{ for which } \theta_i = 1$$
$$< 0 \quad \forall \; i \text{ for which } \theta_i = -1$$

We can adjust $\underset{\sim}{w}$ and b in such a way that

$$\left.\begin{array}{ll} \underset{\sim}{w}' \underset{\sim}{x}_i + b \geq 1 & \text{for which } \theta_i = 1 \\ \leq -1 & \text{for which } \theta_i = -1 \end{array}\right\} \text{(I)}$$

(I) can be written as $\theta_i \left(\underset{\sim}{w}' \underset{\sim}{x}_i + b \right) - 1 \geq 0 \; \forall \; i$.



We like to maximize the distance between $\underset{\sim}{w}'\underset{\sim}{x} + b = 0$ and the "positive" data set, and $\underset{\sim}{w}'\underset{\sim}{x} + b = 0$ and the negative data set. In other words, we want to minimize $\dfrac{2}{\| \underset{\sim}{w} \|}$.

$$d_- = d_+ = \frac{1}{\| \underset{\sim}{w} \|}.$$

Then the problem boils down to

Minimize $\frac{1}{2} \| \underset{\sim}{w} \|^2$ subject to $\theta_i \left(\underset{\sim}{w}' \underset{\sim}{x}_i + b \right) - 1 \geq 0 \; \forall \; i$.

Those points from the data which are falling on $\underset{\sim}{w}' \underset{\sim}{x} + b = +1$ and $\underset{\sim}{w}' \underset{\sim}{x} + b = -1$ are called support vectors.

The optimization problem is a QP Problem.

Principal Component Analysis:-

$$\underset{D\times D}{Cov(\underset{\sim}{X})} = \begin{pmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) \ldots\ldots & Cov(X_1, X_D) \\ Cov(X_2, X_1) & Cov(X_2, X_2) \ldots\ldots & Cov(X_2, X_D) \\ \vdots & \vdots & \vdots \\ Cov(X_D, X_1) & Cov(X_D, X_2) \ldots & Cov(X_D, X_D) \end{pmatrix}$$

$\underset{\sim}{X} = (X_1, X_2, \ldots, X_D)'$. $\underset{\sim}{\lambda} = (\lambda_1, \lambda_2 \ldots, \lambda_d)'$: eigen values

$Var(\underset{\sim}{a}_i' \underset{\sim}{X}) = \lambda_i$

$\boxed{d < D}$

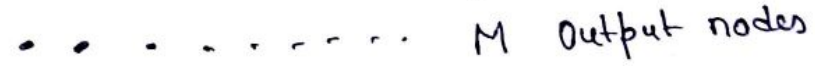
$$\begin{array}{c} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_d \end{array} \begin{pmatrix} \underset{\sim}{a}_1' \underset{\sim}{X} \\ \vdots \\ a_d \underset{\sim}{X} \end{pmatrix} \leftarrow d \text{ principal components}$$

$$Trace(\Sigma) = \sum_{i=1}^{D} Var(X_i) = \sum_{i=1}^{D} \lambda_i$$

Principal Component Analysis using Neural Networking:-

PCA Networks:

$K < M$



K: PCs

Fuzzy C-Means Algorithm :-

Let Ω be a set. μ is known as membership function.
A is said to be a fuzzy subset of Ω if $\exists$ a function
$\mu : \Omega \to [0,1]$.

(Ω, μ) gives a fuzzy set.
Every (Ω, μ) can be called as a fuzzy set.
It is generally represented by μ_j where different j's denotes different membership function.
The union of two fuzzy sets μ_A and μ_B ; denoted by

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}.$$

And the intersection is denoted by $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$
And $\mu_{A'}(x) = 1 - \mu_A(x)$.

Fuzzy C-Partition:- Let $\{\underset{\sim}{x}_1, \dots, \underset{\sim}{x}_n\} \subseteq \mathbb{R}^M$.
Let C denote the no. of clusters and c is known.
Let $U = ((u_{ij}))_{n \times c}$ be a membership matrix.
u_{ij} denotes the membership value of the ith point belonging to j^{th} cluster $u_{ij} \in [0,1]$ $\forall$ i,j; $1 \le i \le n$, $1 \le j \le c$.
U denote a fuzzy c-partition of S if

(i) $\sum_{j=1}^{c} u_{ij} = 1$ $\forall$ $i = 1, 2, \dots, n$.

(ii) $\sum_{i=1}^{n} u_{ij} > 0$ $\forall$ $j = 1, 2, \dots, c$ (each set has to be non-empty)

Let $m > 1$, m is called an exponent.

Let $J_m(U) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \| \underset{\sim}{x}_i - \underset{\sim}{v}_j \|^2$ where $V_j = \dfrac{\sum_{i=1}^{n} u_{ij}^m \underset{\sim}{x}_i}{\sum_{i=1}^{n} u_{ij}^m}$; $j = 1, 2, \dots, c$.

$U = ((u_{ij}))_{n \times c}$ and $V = (\underset{\sim}{v}_1 \cdots \underset{\sim}{v}_c)_{m \times c}$.

Given m, c and s, we like to minimize $J_m(U)$ over U.

— This method is a generalisation of K-Means clustering.

FCM Algorithm:- We are given $S = \{x_1, \dots, x_m\} \subseteq \mathbb{R}^m$, c (no. of clusters) ≥ 2 and $m > 1$.

(i) We shall start with a $U_{n \times c}$ matrix. Let us write the U matrix as $U^{(0)}_{n \times c} = ((u_{ij}^{(0)}))_{n \times c}$; also let at the $(k-1)^{th}$ step, the U matrix is denoted by $U^{(k-1)}$.

(ii) Let $V_j^{(k)} = \left(\sum_{i=1}^{n} u_{ij}^{(k-1)^m} \underset{\sim}{x}_i\right) \Big/ \left(\sum_{i=1}^{n} u_{ij}^{(k-1)^m}\right)$; $j = 1, 2, \dots, c$; $k = 1$.

(iii) $$u_{ij}^{(k)} = \left(\sum_{l=1}^{c} \left(\frac{\| x_i - v_j^{(k)} \|}{\| x_i - v_l^{(k)} \|} \right)^{2/(m-1)} \right)^{-1}$$

(iv) Repeat (ii) and (iii) till the process converges.

Convergence of Fuzzy C-means algorithm:-

It has been shown that $\lim_{k \to \infty} V_j^{(k)}$ exists for every $j = 1, 2, \dots, c$ gives S, m and any starting fuzzy c-partition U of S, i.e., $\lim_{k \to \infty} U^{(k)}$ exists.

Deep Learning:- This technique is an extension of MLP. It has more hidden layers, so processing is done in depth. That's why it is called deep learning.