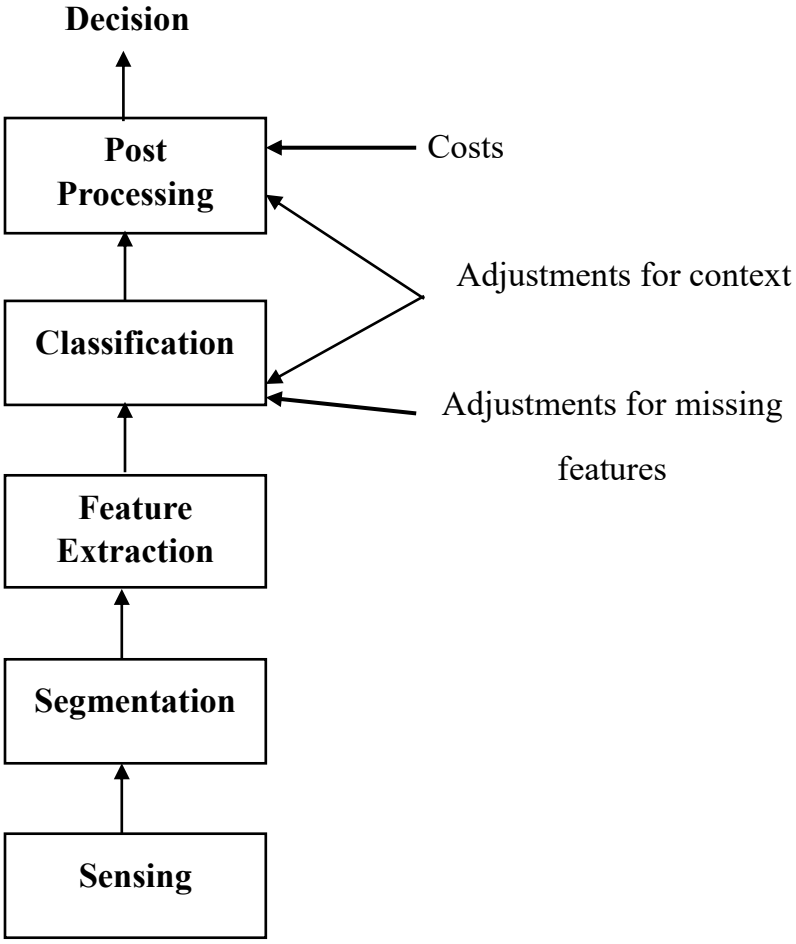
UNIT 1

PATTERN RECOGNITION & APPLICATIONS

1) TYPICAL PATTERN RECOGNITION SYSTEM:



- **SENSING**

  The input to a pattern recognition system is often some kind of a transducer ,such as camera or a microphone. The difficulty of the problem may well depend on the characteristics and limitations of the transducer it's bandwidth, resolution , sensitivity, distortion , signal to noise ratio, latency etc.

  So in practice the design of sensors for pattern recognition is beyond the scope. In this phase, the PR system converts input data to analogous data.

- **SEGMENTATION**

  This phase ensures that the sensed objects are isolated. In this stage the system groups the input data that are used to prepare the sets for future analysis. In practice the fish would often be abutting/overlapping . Our system would have to classify and next begins. The individual patterns have to be segmented. So if we have already recognised the fish then it would be easy to segment their images.

- **FEATURE EXTRACTION**

  It deals with the characterization of an object so that it can be recognize it easily by measurement. Those objects whose values are very similar for the object that are considered to be in the same category while those whose values are quite different for the objects are placed in different categories.

  Feature extraction is approach of determine in the features to be used for learning however for the classification task of hand it is necessary to extract the features to be used. It may involve carrying out some arithmetic operations on the features like linear combinations of the features are finding the value of the function. Feature selection is the process of discording or removing some of the features at the patterns and using only some of the features. In the end, the reduction of the data helps to build the model with less machine effort and also it increases the speed of learning.

- **CLASSIFICATION**

  It is the task of assigning a class label to and input pattern the class level indicates one of the given set of classes the classification is carried out with the help of a model after using a learning process according to the type of learning used there is two types of classification.

  1. Using supervised learning

2. other one is using an Unsupervised learning

Supervised learning makes of a set of examples which is already have the class labels assign to them.

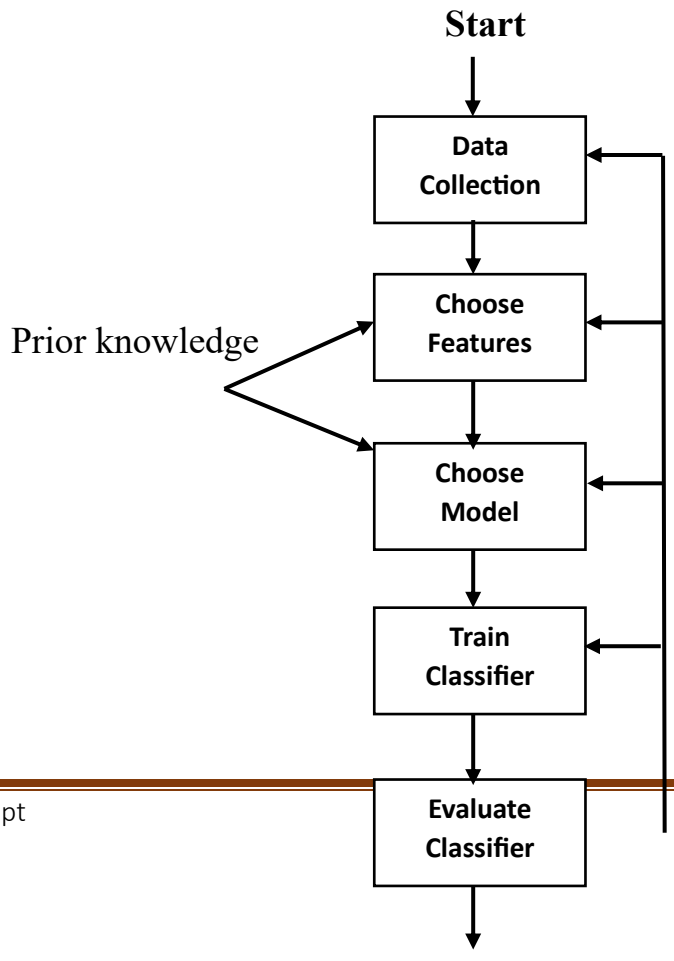
Unsupervised learning attempts to fight inheritance structure of the data and semi supervised learning makes use of a small number of labeled data makes use to learn classifier.

- **POST PROCESSING**

It deals with action decision making by using the output of the classifier.

Action such as minimum error rate classification to minimize the total expected cost. Here in this phase further considerations are made before a decision is made.

Design Cycle:



End

- **Data collection**

  Pattern recognition system is a data analysis method that uses machine learning algorithms automatically recognise pattern and regularities in data.

  We need to collect both training and testing data . This data can be anything from text to images or images to sound or other desirable quality. It is important that the data sources are available are trustworthy and well built. So, the data collected will be highest possible quality and collected data should be accurate in order to enable better decision making.

- **choose features**

  The choice of the distinguishing features is a critical design step and depends on the characteristics of the problem domain. Incorporating prior knowledge can be form more suitable and difficult in some applications of the knowledge ultimately derives from information about the production of the patterns as we saw in analysis by synthesis.

  In selecting or designing features. we would like to find features that are simple to extract invariant to irrelevant transformation in sensitive to noise and useful for discriminating patterns in different ways.

- **Choosing model**

  In general model selection depends on the nature of the data the sample size and the intended applications of the results model selection is necessary formation learning because it helps to determine the most appropriate model to solve a specific problem based on various criteria it helps to ensure that the model performs best and can generalize well to new data which is an essential for Real world applications it help prevent war fitting which happens on more Complex models when the model fixed the training data Set to closely.

selecting a model can save computing resources and time by eliminating candidate models that perform poorly and assist in improving the final performance.

Example: - Artificial neural networks model will give best performance.

- **Train Classifier**

  In general the process of using data to determine the classify is referred to as training the classify training is a process of fitting a model in into the data this selecting a model choosing appropriate hyper parameters and optimizing the models parameters to maximize the loss function different types of training protocols and stochastic batch and online.

- **Extract Classifier**

  Evaluation is important both to measure the performance of the system and to identify the need for improvements in its components. This allows us to estimate the generalization performance of the model and to compare the performance of different models.

2) Pattern and Feature Extraction:

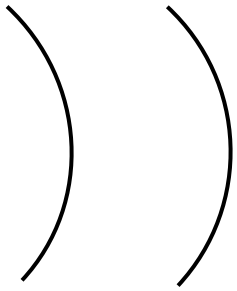
Pattern: -

Any object that you see in this world that forms a pattern.Suppose for example if I take an image of car it becomes a pattern. if I take an image of cow it also gives the pattern similarly when we talk about the speech signals if I say a word hello it is recommended using a microphone then if you see the output microphone is the object you see the output of microphone which is also a pattern if you say another word it gives another pattern so the pattern is something which describes what you see in the world what you hear in the world and what you

sense in the world so the job of pattern recognition is a machine should be able to understand what we are seeing around us or what we are speaking and we want to enable emission to that to do this each of the patterns has to be represented in such a way of form which the mission can understand so when we speak something or see something these has to be represented by something called features when these features are descriptive or added in a particular form of a vector which is known as feature vector.

Feature extraction

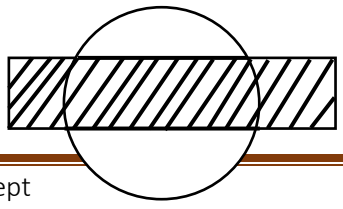
The main problem in pattern recognition is how we recognize these patterns or similar or dissimilar

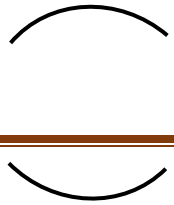


These two arks are same or part of a circle if you do not consider the invariance then we will have to parameters center of the circle and radius of the circle so if you change the place of the circle then the center of the circle will change but radius remains the same that is why we have various features like translation in variant features rotation invariant features in order to classify these two orcs' or similar we need to calculate the radius and center of the circle.

So, if R1=R2(the orcs are same (or) similar of the circle).

So, if we have to find out if the error of measurement is more we have to find out the measure of similarity which is nothing but the difference between R1 and R2 if the difference is small and negligible assume that the two arcs are same.

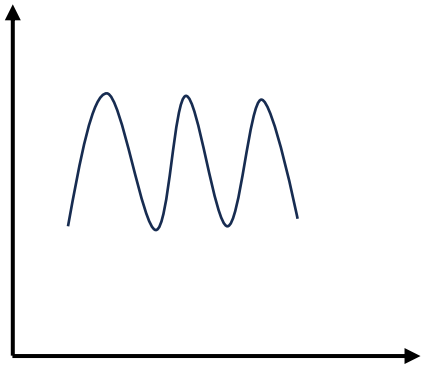




R1=R2

C1=C2

Two arcs are part of the same circle but discontinuous.



Whenever we find the similarities between two patterns is high and dissimilarities is low in this case taking one of the patterns as reference or as safe model which is there in our knowledge base.

Then say the second pattern is recognized or associated with the first one.

To generate the more patterns, we have different kinds of learning one is supervisor learning and another one is an unsupervised learning.

Supervised: -

Take a set of known patterns for which we have to find out the feature vectors and for those speech of vectors form a representative features vector which represents the same class of values if makes use of known patterns for training and learning.

Unsupervised: -

To generate feature vectors for each one of these patterns and form mixture of a feature vector process the mixture of feature vectors and by processing those features vectors we have to partition these set of feature vectors into a subset of vectors.

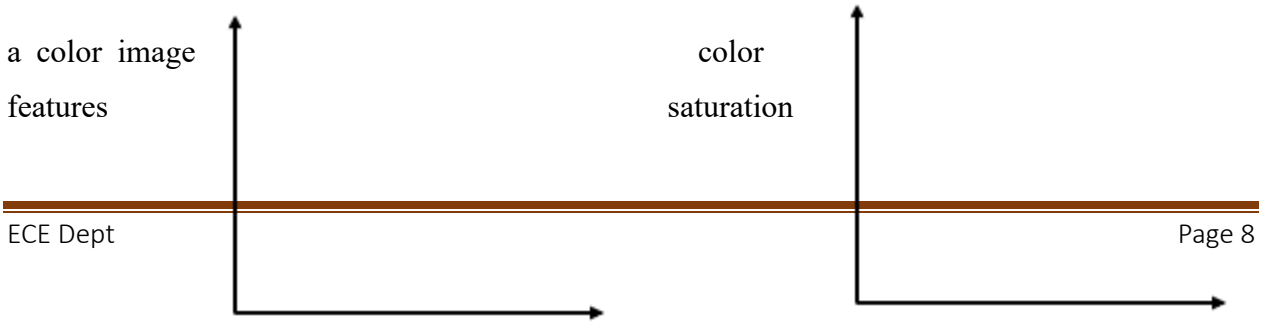
Future vector is one of the set or similar and another similar feature vectors will be another set.

There are two different types of feature descriptive

1. Shake based feature descriptive

2. Region based feature descriptive

The shape is based on feature descriptive describes what is the shape of object like circle square rectangle extra the region-based feature descriptive describe the boundary features for a grade level image features will be intensity and brightness if intensity is high image will be bright it is low of the image will be dark for



gives the purity of the color.

3) Training and Learning of Pattern Recognition System: -

Pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms.

Learning is a phenomenon through which a system gets trained and becomes adaptable to give results in an accurate manner. Learning is the most important phase as to how well the system performs on the data provided to the system depends on which algorithms are used on the data. The entire dataset is divided into two categories, one which is used in training the model i.e. Training set, and the other that is used in testing the model after training, i.e. Testing set.

- **Training set:**

  The training set is used to build a model. It consists of the set of images that are used to train the system. Training rules and algorithms are used to give relevant information on how to associate input data with output decisions. The system is trained by applying these algorithms to the dataset, all the relevant information is extracted from the data, and results are obtained. Generally, 80% of the data of the dataset is taken for training data.

- **Testing set:**

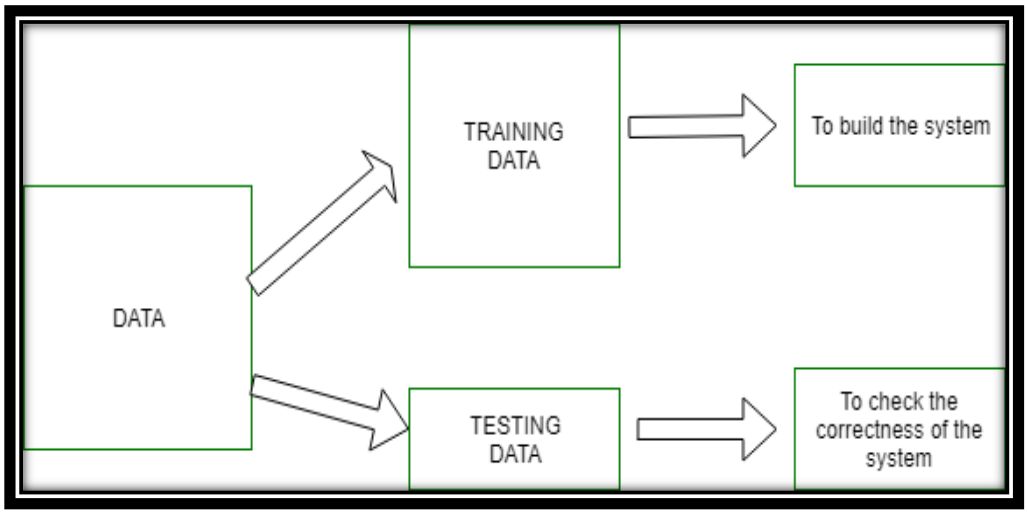
  Testing data is used to test the system. It is the set of data that is used to verify whether the system is producing the correct output after being trained or not. Generally, 20% of the data of the dataset is used for testing. Testing data is used to measure the accuracy of the system. For example, a system that identifies which category a particular flower belongs to is able to identify seven categories of flowers correctly out of ten and the rest of others wrong, then the accuracy is 70 %

Real-time Examples and Explanations:

- A pattern is a physical object or an abstract notion. While talking about the classes of animals, a description of an animal would be a pattern. While talking about various types of balls, then a description of a ball is a pattern. In the case balls considered as pattern,

- The classes could be football, cricket ball, table tennis ball, etc. Given a new pattern, the class of the pattern is to be determined. The choice of attributes and representation of patterns is a very important step in pattern classification. A good representation is one that makes use of discriminating attributes and also reduces the computational burden in pattern classification.
- An obvious representation of a pattern will be a vector. Each element of the vector can represent one attribute of the pattern. The first element of the vector will contain the value of the first attribute for the pattern being considered.

Example: While representing spherical objects, (25, 1) may be represented as a spherical object with 25 units of weight and 1 unit diameter. The class label can form a part of the vector. If spherical objects belong to class 1, the vector would be (25, 1,1), where the first element represents the weight of the object, the second element, the diameter of the object and the third element represents the class of the object.



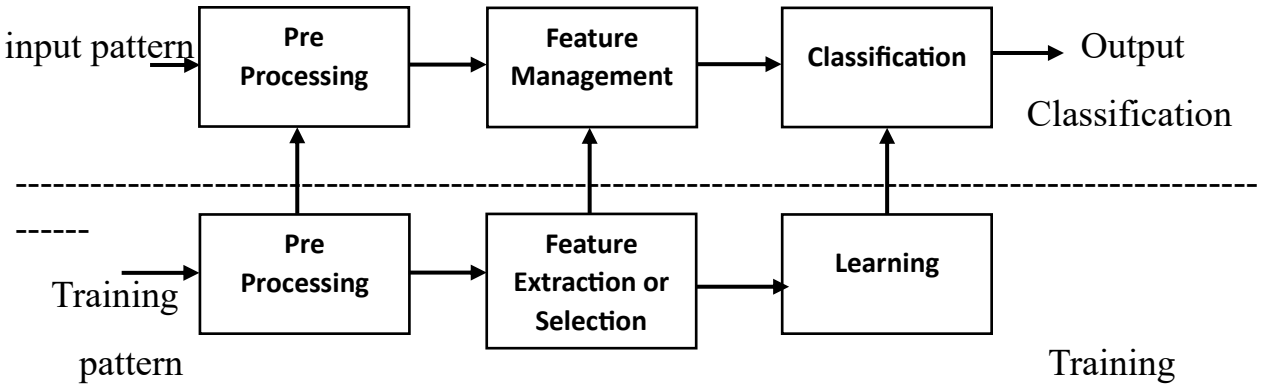
4) Different Types of Pattern Recognition System:-

There are three main types of pattern recognition, dependent on the mechanism used for classifying the input data.

Those types are:

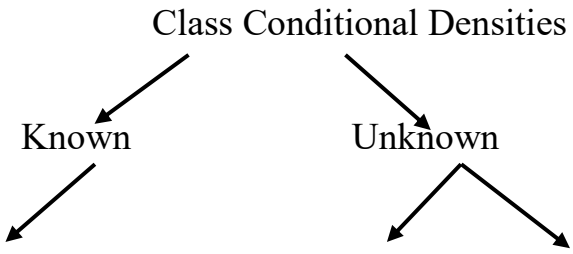
- Statistical
- Structural (or syntactic)
- Neural.

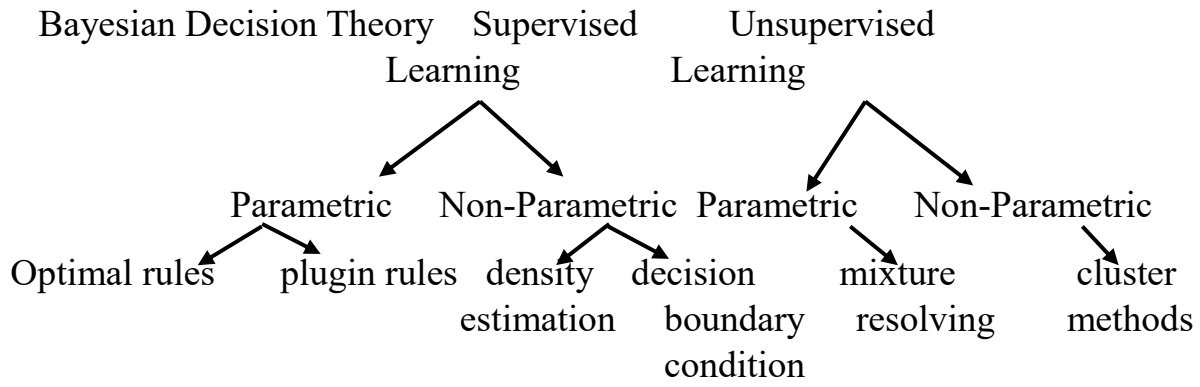
Statistical Approach: -



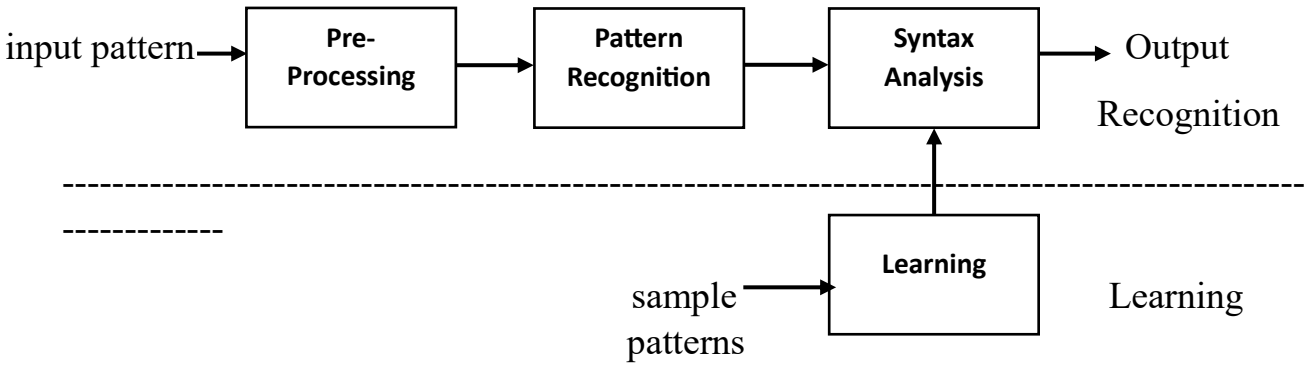
This approach is based on statistical decision theory. Pattern recognizer extracts quantitative features from the data along with the multiple samples and compares those features. However, it does not touch upon how those features are related to each other.The basic components of statistical pattern recognition systems are

- Input of data: - Large amounts of data enter the system through different sensors.
- Preprocessing: - At this stage, the system groups the input data to prepare the sets for future analysis.
- Feature selection (extraction): - The system searches for and determines the distinguishing traits of the prepared sets of data.
- Classification. Based on the features detected in the previous step, data is assigned a class (or cluster), or predicted values are calculated (in the case of regression algorithms).
- Learning: - Learning in the context of a pattern recognition system is defined as the process that allows it to scope with real and ambiguous data.
- Training: - Training is the process through which the model learns or recognizes the patterns in the given data for making suitable prediction





Structural Approach: -



This approach is closer to how human perception works. It extracts morphological features from one data sample and checks how those are connected and related.

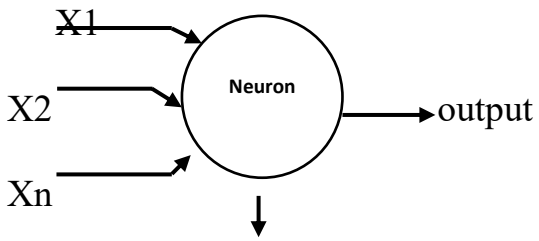
The basic components in structural approach are:

- Input of data: - Large amounts of data enter the system through different sensors.
- Preprocessing: - At this stage, the system groups the input data to prepare the sets for future analysis.
- Syntax Analysis or Parsing is the second phase, i.e. after lexical analysis. It checks the syntactical structure of the given input, i.e. whether the given input is in the correct syntax.

- Grammar inference is the task of learning grammars or languages from training data. It is a type of inductive inference, the name given to learning techniques that try to guess general rules.
- Classification/Recognition: - Based on the features detected in the previous step, data is assigned a class (or cluster), or predicted values are calculated (in the case of regression algorithms).
- Learning: - Learning in the context of a pattern recognition system is defined as the process that allows it to scope with real and ambiguous data.

Neural Network Approach: -

In this approach, artificial neural networks are utilized. Compared to the ones mentioned above, it allows more flexibility in learning and is the closest to natural intelligence.



The neural approach applies biological concepts to machines to recognize patterns the outcome of this effort is invention of artificial neural networks in neural network is an information processing system it consists of massive simple processing units with a high degree of interconnection between each unit.The design and function of neural networks simulate some functionality of biological brains and neural systems.

Normally only feed forward networks are used for pattern recognition feed forward means that there is no feedback to the input similar to the where that human beings learn from

mistakes neural networks could also learn from their mistakes by giving feedback to the input patterns.

Neural networks can tolerate noise and if trained properly will respond correctly for many inputs and one output.

The neuron has two modes of operation training mode using mode during training the network is train to associate outputs with input patterns during using mode when the network is used it identify the input pattern and tries to give output with the help of associated output pattern.

The power of neural networks comes to life when a pattern that has no output associated with it is given as input in this case the network gives the output that corresponds to a dot input pattern that is least different from the given pattern.

5) Discriminant Functions:

A discriminant function that is a linear combination of the components of X can be written as

$$g(X)=w^TX+w_0 \text{ -----}\rightarrow(1)$$

Where w is the weight vector w0 is the bias or threshold weight

The linear discriminant functions can be divided into three cases

1. two category case
2. multi category case
3. general case

Multi category case:

There are many ways to represent pattern classifiers one of the most useful is in terms of a set of discriminant functions

gi(X), where i=1,2,3, -------c

The classifier is said to assign a feature vector X to class with if gi(x) is greater than gj(x) for all I is not belongs to i

gi(X)>gj(X) for all i! =j---→(2)

Thus the classifier is viewed as a network or machine that computes C discriminant functions and selects the category corresponding to the largest discriminant

The Baye's classifier is easily and naturally represented in this way for the general case with the risk we can let

gi(x) = -R(αi/x)

because the maximum discriminant function will then correspond to the minimum conditional risk.

For the minimum rate case we can simplified things further by taking gi(X) = probability of (wi/X)

gi(X)=P(wi/X)

So that the maximum discriminant function corresponds to the maximum posterior probability

gi(X)=P(X/wi) (P(wi))/$\sum_{j=1}^{c} P\left(\frac{X}{wj}\right) P(wj)$----→(3)

gi(X)=P(X/wi)P(wi)-----→(4)

gi(X)=ln P(X/wi) + ln P(wi) ------→(5)

The effect of any decision don't is to divide the features space into C decision regions R1,R2,.....Rc

A linear machine divides the future space into C decision regions with gi(x) being the largest discriminate if x is in region R1

If Ri and Rj are continuous the boundary between them is a portion of the hyper plane hij defined by

$$(wi-wj)^{T}X+(wi_{0}-wj_{0})=0$$

It is easy to show that the decision regions for a linear machine can or converts and this restriction surely limits the flexibility and accuracy of the classifier.

Two category case:

While the two - category case is just special incidence of the multi category case indeed classifier that places pattern in one of only two categories has a special name dichotomyzer a classifier for more than two categories is called a polycotomyzer.

In this two-dimensional two category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas and the decision region are two is not simply connected

Instead of using two discriminant functions g1 and g2 and assigning x2w1 if g1>g2 it is more common to define a single discriminant function.

$$g(x)=g1(x)-g2(x)\text{--------}(6)$$

Thus, a dichotomyzer can be viewed as a machine that computes a single discriminant function g(x) and classifies x according to the algebraic sign of the result of the various forms in which the minimum error rate discriminant function can be written as

$$g(X)=P(w1/X)-P(w2/X)\ \text{------}\rightarrow(7)$$

$$g(X)=\ln [P(X/w1)/P(X/w2)]+\ln [P(w1)/P(w2)] \text{ --}\rightarrow(8)$$

The hyperplane h divides the features space into two half spaces R1 for w1 and R2 for w2. Because g(x)>0 if x is in R1 w points to R1. Sometimes, any x in R1 is in the positive side of h and any x in R2 is in the negative side

The discriminant function g(x) gives an algebraic measure of the distance from x to the hyperplane expressed as

$$X=X_P+rw/||w||$$

Where, X_P is a normal projection of X onto h. r it is the desire algebraic distance

$$g(X_P)=0$$

$$g(X)=w^tX+w_0$$

$$=r||w||$$

$$r=g(X)/||w||$$

The distance from origin to h is given by, w0/||w||

if w0>0, the origin is on the positive side of edge if w0<0, then the origin is on negative side.

UNIT- II

STATISTICAL PATTERN RECOGNITION

1. Parametric Estimation & Supervised learning
2. maximum likelihood estimation -
3. Bayesian parameter estimation
4. Non-parametric approaches - parzen window
5. K-NN Estimation
6. un-supervised learning - clustering concepts.

---

1. parametric Estimation & Supervised learning;

parametric Estimation:-

* parametric Estimating, a more accurate technique for estimating cost & duration, uses the relationship blw variables to calculates the cost or duration.
* Essentially, a parametric estimate is determined by identifying the unit cost or duration & the no. of units reqiuried for the project or activity.
* To obtain the parametric value used for Estimation, you can look at:
  - → previous internal projects.
  - → previous External "
  - → industry publications

* An optimal classifier is designed with known prior probabilities $P(\omega_i)$ & class conditional densities $p(x|\omega_i)$.

* we require learning of the class conditional probabilities, density functions & prior probabilities for designing a classifier based on Bayesian decision theory.

* The problem should be solved by using the samples to estimate the unknown probabilities

* In supervised learning, mode, the estimation of the prior probabilities is not difficult.

* following all the procedures which are used to solve the problem of parameter estimation:

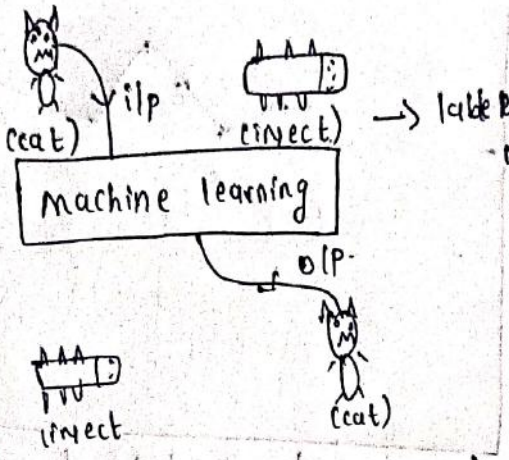
(a) maximum likelihood estimation.

(b) Bayesian estimation.

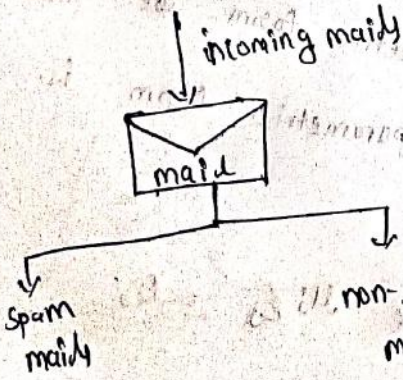
Supervised Learning:-

Before going into this topic, the main topic is to define the. Supervised learning is in machine learning, so, here machine learning. is defined as the branch of computer science which automatically accepts the data.

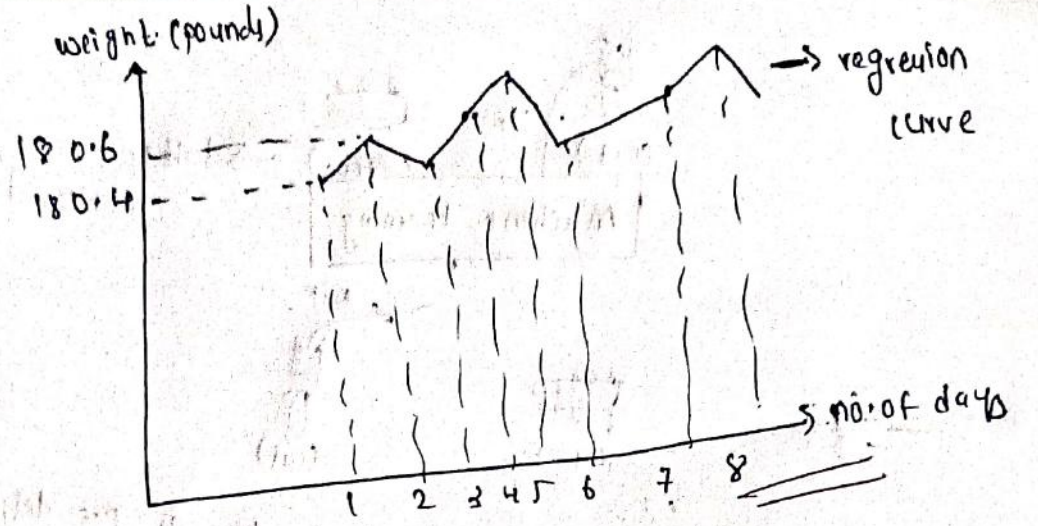
* In this type of learning we will train the machine based on "labelled data"



* In this type ilp is labelled & olp is pre defined.
* The type of problem domain that we can solve using the supervised learning is.
  * classification
  * Regression.
* A real time eg:- for classification is nothing but training a machine to classify the incoming mails into spam & nonspam mails automatically.



* A read time Example for regreuion is nothing but continuous track of weights of a peruon day to day which continuesly changes. it is graphically re presented as.



2. maximum likelihood Estimation:-

* MLE methods have a no. of attractive attributes.
* This MLE have involved the no. of Bayesian techniques (or) other methods.
* Suppose we have got the c no. of classes so, that we have c datasets D_1 D_2 D_3 ---- D_j. have drawn independently to the probability. $p(x|w_j)$ → assume known parametric form.
* for this known parametric form we assume if it is normal (or) Gaussian distribution parametric form, is. $N(\mu_j, \Sigma_j)$ → Co-variance matrix.
  ↓ mean vector

where $\mu_j \Sigma_j = \theta_j$

$p(x|w_j) = p(x|w_j, \theta_j)$

$\theta = \theta_1, \theta_2 \ldots \theta_j$

* we information from training samples in set D_j to have good Estimate parameter of θ_j.

for $i \neq j$

Samples in D_i doesn't provide any information of θ_j

$p(x|\theta) \rightarrow D$

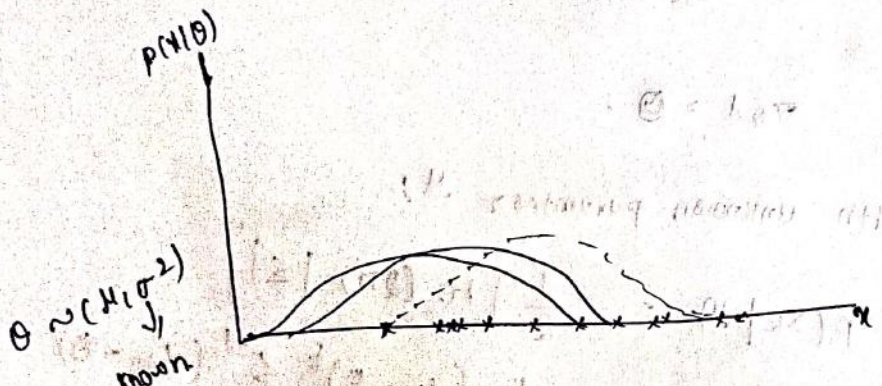
if D contains no. of samples. i.e. n. no. of samples, so I can write as

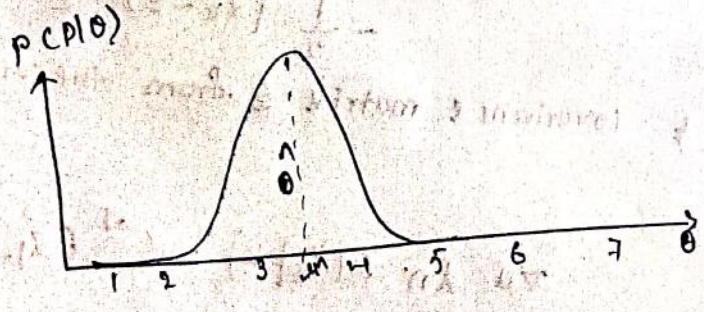
$$p(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$$

Here the samples are drawn independently.

$p(D|\theta)$ = likelihood value of θ.

$\theta = \hat{\theta}$ which maximises the $p(D|\theta)$





$$l(\theta) = \ln p(D|\theta)$$

$$= \sum_{k=1}^{n} \ln p(x_k|\theta)$$

∇_θ = radient with θ

$\theta = (\theta_1, \theta_2, \cdots \theta_p)^t$ denote the p-component.

$$\nabla_\theta = \begin{bmatrix} \frac{\partial}{\partial \theta_1} \\ \vdots \\ \frac{\partial}{\partial \theta_p} \end{bmatrix}$$

$$\nabla_\theta l = \sum_{k=1}^{n} \nabla_\theta \ln p(x_k|\theta).$$

$$\nabla_\theta l = 0.$$

Gaussian. Case. with unknown paramteas μ:-

$$\ln p(x_k|\mu) = -\frac{1}{2}\left[\ln(2\pi)|\Sigma|\right] - \frac{1}{2}(x_k-\theta)^t \Sigma^{-1}(x_k-\theta)$$

* with mean μ & covarivence matrik Σ. from the case mean is unknown.

$$\nabla_\mu \ln p(x_k|\mu) = \Sigma^{-1}(x_k-\mu).$$

multiplie with Σ & rearranging

$$\sum_{k=1}^{n} \Sigma^{-1}(x_k-\hat{\mu}) = 0$$

$$\sum_{k=1}^{n} \Sigma^{-1}(x_k - \hat{\mu}) = 0$$

multiplie with Σ

$$\sum_{k=1}^{n} \hat{\mu} = \sum_{k=1}^{n} x_k$$

$$n \cdot \hat{\mu} = \sum_{k=1}^{n} x_k$$

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^{n} x_k$$

* In the multivariant case. neither the mean μ & covariance matrix Σ is known

$\theta \approx (\mu_1, \Sigma)$ $\Rightarrow$ $\theta_1 = \mu$

$\theta_2 = \Sigma$

$$\ln p(x_k|\theta) = -\frac{1}{2} \ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

and the derivative is

$$\nabla_\theta \ln p(x_k|\theta) = \begin{bmatrix} \frac{1}{\theta_2}(x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

Applying the log-likelihood.

$$\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0$$

$$-\sum_{k=1}^{n} \frac{1}{2\hat{\theta}_2} + \frac{(x_k - \hat{\theta}_1)^2}{2\hat{\theta}_2^2} = 0$$

$\hat{\theta}_1$ & $\hat{\theta}_2$ are the MLE for θ_1 & θ_2

$$\hat{\theta}_1 = \frac{1}{n} \sum_{k=1}^{n} x_k \;,\; \theta_2 = \frac{1}{n} \sum_{k=1}^{n} (x_k - \hat{\theta}_1)^2$$

μ σ^2

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n} (x_k - \hat{\mu})^2$$

3. Bayesian parameter Estimation :-

* In this the Bayesian technique to calculate the a posteriori density $p(\theta/D)$ & the derived probability density $p(x/D)$

→ univarient case $p(\mu/D)$

→ " " $p(x/D)$

→ multivarient case. $p(x/\mu) \sim N(\mu, \Sigma)$

the univarient case $p(\mu/D)$ ⟹ μ unknown

$$p(x/\mu) \sim N(\mu, \sigma^2)$$

$p(\mu)$ = prior density then

$$p(\mu) \sim N(\mu_0, \sigma_0^2).$$

μ_0, σ_0 = known.

using Bayes formula.

$$p(\mu/D) = \frac{p(D/\mu)\, p(\mu)}{\int p(D/\mu)\, p(\mu)\, d\mu}$$

$$= \alpha \prod_{k=1}^{n} p(x_k/\mu)\, p(\mu)$$

$$p(\mu/D) = \alpha \prod_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right]$$

$p(x/\mu)$ $\qquad$ $p(\mu)$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu - x_k}{\sigma}\right)^2 + \left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right)\right] \quad (1)$$

$$= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n}x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$

$$p(\mu|D) = \frac{1}{\sqrt{2\pi}\sigma_n}\exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]$$

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}$$

$$\frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu_n} + \frac{\mu_0}{\sigma_0^2}$$

$\hat{\mu_n}$ = Sample mean

$$\hat{\mu_n} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\right)\hat{\mu_n} + \frac{\sigma^2}{n\sigma_0^2+\sigma^2}\mu_0$$

$$\sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2+\sigma^2}$$

The univariate case: $p(x|D)$

$$p(x|D) = \int p(x|\mu)\, p(\mu|D)\, d\mu$$

$$= \int \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n}\exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right]\cdot f(\sigma,\sigma_n).$$

$$f(\sigma, \sigma_n) = \int \exp\left[-\frac{1}{2}\frac{\sigma^2+\sigma_n^2}{\sigma^2\sigma_n^2}\left(\mu - \frac{\sigma_n^2 x + \sigma^2 \mu_n}{\sigma^2+\sigma_n^2}\right)^2\right] d\mu$$

$$p(x|D) \sim N(\mu_n, \sigma^2+\sigma_n^2)$$

The multivariate case:-

* The multivariate case, in which Σ is known but μ is not is a direct generalization of the univariate case.

$$p(x|\mu) \sim N(\mu, \Sigma) \quad \& \quad p(\mu) \sim N(\mu_0, \Sigma_0)$$

Σ, Σ_0, μ_0 = known.

$$p(\mu|D) = \alpha \prod_{k=1}^{n} p(x_k|\mu)\, p(\mu)$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\mu^t(n\Sigma^{-1}+\Sigma_0^{-1})\mu - 2\mu^t\left(\Sigma^{-1}\sum_{k=1}^{n} x_k + \Sigma_0^{-1}\mu_0\right)\right)\right]$$

$$p(\mu|D) = \alpha'' \exp\left[-\frac{1}{2}(\mu-\mu_n)^t \Sigma_n^{-1}(\mu-\mu_n)\right]$$

$$\Sigma_n^{-1} = n\Sigma^{-1} + \Sigma_0^{-1}$$

& $\Sigma_n^{-1}\mu_n = n\Sigma^{-1}\hat{\mu}_n + \Sigma_0^{-1}\mu_0$

$\hat{\mu}_n$ = sample mean.

$$\hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k.$$

from $(A^{-1}+B^{-1}) = A(A+B)^{-1}B$

$= B(A+B)^{-1}A$.

$$\therefore \mu_n = \Sigma_0\left(\Sigma_0 + \frac{1}{n}\Sigma\right)^{-1}\hat{\mu_n} + \frac{1}{n}\Sigma\left(\Sigma_0 + \frac{1}{n}\Sigma\right)^{-1}\mu_0$$

$\hat{\mu_n}$ & μ_0 = linear combination.

$$\Sigma_n = \Sigma_0\left(\Sigma_0 + \frac{1}{n}\Sigma\right)^{-1}\frac{1}{n}\Sigma.$$

$$p(x|D) \sim N(\mu_0, \Sigma + \Sigma_n)$$

By applying integeration.

$$p(x|D) = \int p(x|\mu)\cdot p(\mu|D)\, d\mu.$$

the sum of 2 mutually independent random variables,

$$\therefore p(x|D) \sim N(\mu_n, \Sigma + \Sigma_n).$$

4. Non-parametric approaches :-

* They can be used with arbitrary distributions & without the assumptions that the underlying density function.

* It is a non-parameteric estimation technique which is used to determine the probability density with dataset.

$$P = \int_R p(x)\,dx \quad \text{(density)}$$

$$P_k = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\frac{n}{k} = \frac{n!}{k!(n-k)!}$$

non-parametric approaches

Methods that are based on estimating the density function $p(x|\omega_j)$ from sample space. patterns

Eg :- parzen window.

Methods that are based on estimating the posterior probability $p(\omega_j|x)$

Eg :- K-NN estimation.

Parzen window :-

* parzen window is considered to be a claurification technique. used for non-parametric estimation.

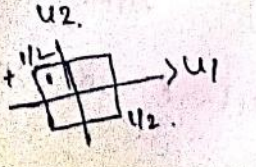
* Parzen-window approach to estimate densities assume that the Region R_n is a D-Dimensional hyper cube.

$$V_n = h_n^d$$

h_n = length of the edge of R_n.

* let $\phi(u)$ be a window function of the form

$$\phi(u) = \begin{cases} 1 & |u_j| \le 1/2 \\ 0 & \text{otherwise} \end{cases} \quad ; \; j = 1, 2, \ldots d$$



* $\varphi(u)$ is a hypercube, & $\varphi\left(\frac{(x-x_i)}{h_n}\right)$ is equal to unity, if x_i falls within a hypercube of volume V_n centered at x & equal to zero otherwise.

* The no. of samples in the hypercube is:

$$K_n = \sum_{i=1}^{n} \varphi\left(\frac{x-x_i}{h_n}\right).$$

Substitute K_n in $P_n(x)$ Then.

$$P_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{x-x_i}{h_n}\right) \qquad P_n(x) = \frac{k_n/n}{V_n}$$

* $P_n(x)$ estimates $p(x)$ as an average of functions of x & the samples x_i; $i = 1, 2, \ldots, n$, These functions φ can be general.

$$\varphi(x) \geq 0$$

&

$$\int \varphi(u)\, du = 1.$$

* Let us examine the effect that the window width h_n has on $P_n(x)$. if we define function $\delta_n(x)$ by

$$\delta_n(x) = \frac{1}{V_n} \varphi\left(\frac{x}{h_n}\right)$$

then $$P_n(x) = \frac{1}{n} \sum_{i=1}^{n} \delta_n(x - x_i)$$

* if $\delta_n(x)$ is very large, then δ_n is small, x is far from the x_i before $\delta_n(x-x_i)$ changes from $\delta_n(0)$.

$$\int \delta_n(x-x_i)\, dx = \int \frac{1}{V_n} \varphi\left(\frac{x-x_i}{h_n}\right) dx = \int \varphi(u)\, du = 1.$$

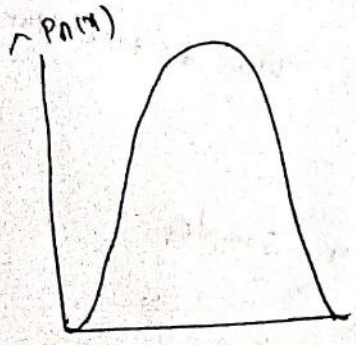
* If h_n approaches zero

$\delta_n(x-x_i)$ approches delta functions.

$p_n(x)$ " superposition delta functions.

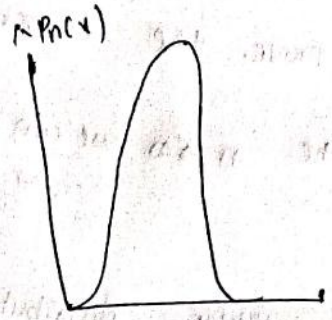
Then the samples are.

$h = 1$



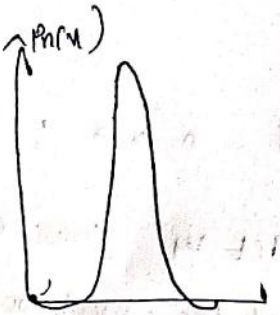
①

$h = 0.5$



②

$h = 0.2$



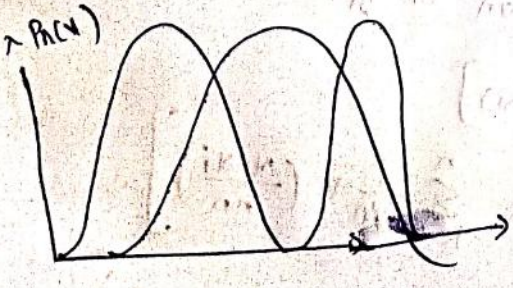
③

In this δ_n is normalized.

* If h_n is very large.

$p_n(x)$ is the superposition of functions & is a smooth "out-of-focus" Estimate of $p(x)$.

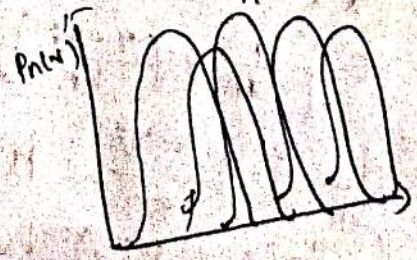
$h_n = 1$



* If h_n is very small,

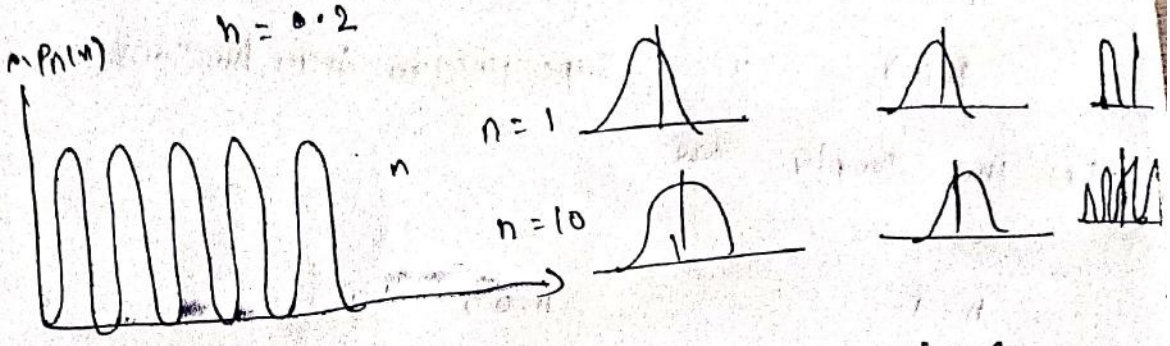
$p_n(x)$ is the superposition. & it Estimate noise of $p(x)$.

$h = 0.5$



* If h_n approaches zeros

$P_n(x)$ is a superposition of delta functions.



If $n \to \infty$, we can prove the convergence. mean & variance.

let us consider that the $n \to \infty$ we can get the two distributions of the likelihood.

$P_n(x)$ = probability distribution with n samples.

$$\lim_{n \to \infty} \bar{P}_n(x) = p(x)$$

$$\lim_{n \to \infty} \sigma^2_{n(x)} = 0.$$

Convergence of the mean:-

consider $\bar{P}_n(x)$, the mean of $P_n(x)$

$$\bar{P}_n(x) = E[P_n(x)]$$

$$= E\left[\frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \psi\left(\frac{x - x_i}{h_n}\right)\right]$$

$$= \frac{1}{n} \sum_{i=1} E\left[\frac{1}{V_n} \psi\left(\frac{x - x_i}{h_n}\right)\right]$$

All x_i ave's iid (independent & identical distributions)

$$= \frac{1}{n} \times n \, E\left[\frac{1}{V_n} \psi\left(\frac{x - x_i}{h_n}\right)\right]$$

$$= \int_{-\infty}^{\infty} \frac{1}{V_n} \psi\left(\frac{x - v}{h_n}\right) p(v) \, dv.$$

$$= \int \delta_n(x - v) \, p(v) \, dv.$$

Convergence of the variance:-

$$\sigma_n^2(x) = \text{Var}[P_n(x)]$$

$$= \text{Var}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{v_n}\varphi\left(\frac{x-x_i}{h_n}\right)\right]$$

$$= \sum_{i=1}^{n} \text{var}\left[\frac{1}{n}\frac{1}{v_n}\varphi\left(\frac{x-x_i}{h_n}\right)\right]$$

$$= n \times \left[E\left(\frac{1}{n}\frac{1}{v_n}\varphi\left(\frac{x-x_i}{h_n}\right)\right)^2\right] - E\left[\frac{1}{n}\frac{1}{v_n}\varphi\left(\frac{x-x_i}{h_n}\right)\right]^2 \qquad \text{var}(x) = E(x^2) - E(x)^2$$

$$= n \times \frac{1}{n^2 v_n} \cdot E\left[\frac{1}{v_n}\varphi^2\left(\frac{x-x_i}{h_n}\right)\right] - \left[\bar{P_n}(x)\right]^2$$

$$\sigma_n^2 \le \frac{1}{nv_n}\int \frac{1}{v_n}\varphi^2\left(\frac{x-x_i}{h_n}\right)p(x_i)\,dx_i$$

$$x_i = v$$

$$\sigma_n^2 \le \frac{1}{nv_n}\int \frac{1}{v_n}\varphi^2\left(\frac{x-v}{h_n}\right)p(v)\,dv$$

$$\le \frac{1}{nv_n}\int \frac{1}{v_n}\left[\varphi\left(\frac{x-v}{h_n}\right)\varphi\left(\frac{x-v}{h_n}\right)\right]p(v)\,dv.$$

$$\le \frac{1}{nv_n}\,\phi_{max}\left(\frac{x-v}{h_n}\right)\int \frac{1}{v_n}\phi\left(\frac{x-v}{h_n}\right)p(v)\,dv.$$

$$\le \frac{1}{nv_n}\,\phi_{max}\,\frac{(x-v)}{h_n}\,\bar{P_n}(x)$$

$$\boxed{\sigma_n^2(x) \le \frac{1}{nv_n}\,\phi_{max}\,\frac{(x-v)}{h_n}\,\bar{P_n}(x)}$$

K-NN Estimations:

* K-Nearest Neighbor is another method of non-parameter estimation of ~~est~~ clasification other than parzen window.

* K-Nearest Neighbor is also known as K-NN estimation is one of the best supervised statistical learning technique for performing non-parametric clasification.

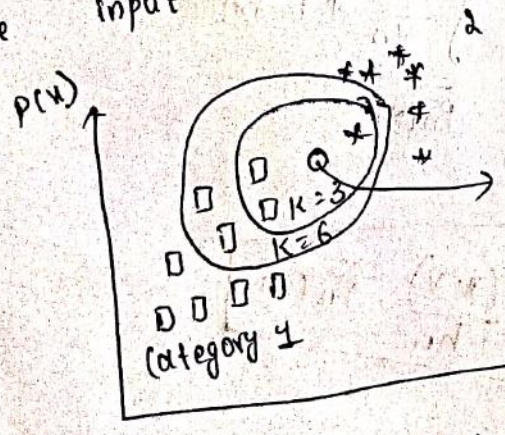
* To Estimate $P(x)$ from n samples, we can center a volume about x and let it grow until it captures k_n samples,

$P_n(x) = \frac{k_n/n}{V_n}$ $\Rightarrow$ k_n = some function of n.

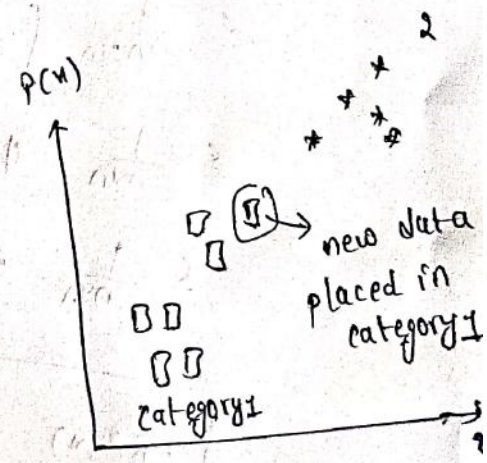
* These samples are called the k-nearest neighbors of x.

* The value of k is very crucial in the KNN estimation to define the number of neighbors in the algorithm.

* The value of K in the k-NN should be chosen based on the input data.



Here I'm giving K=3



Clasification with K-NN:

Given a no. of claues. $w_s \in \{w_1, w_2, \ldots w_j\}$ & the set of training samples, where each sample belongs to a single class, Suppose that we place a volume. V around x an unknown parameters. x & capture k-samples

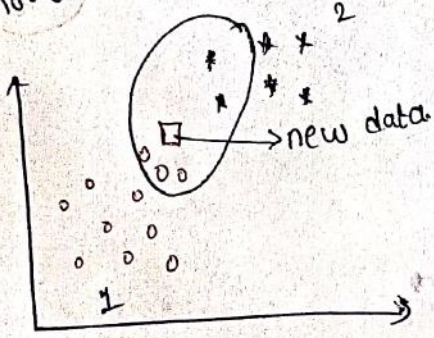
$$\therefore P_n(x, w_j) = \frac{k_i/n}{V}$$

Then the posteriori probability of an unknown sample x belonging to a specific class j, Then

$$P_n(w_j/x) = \frac{P_n(x, w_j)}{\sum_{j=1}^{c} P_n(x, w_j)} = \frac{k_i}{k}$$

Steps to carried :-

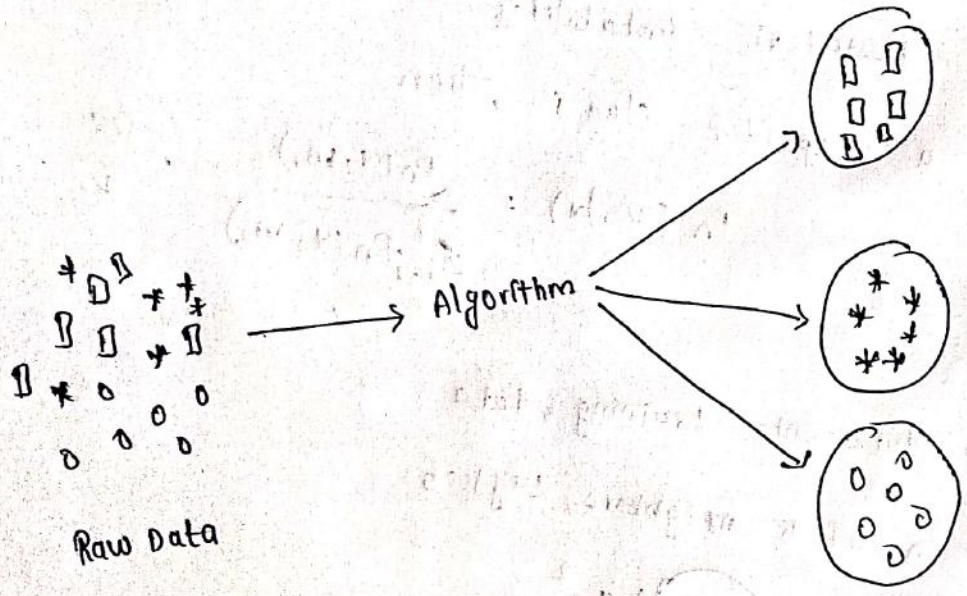
1. Divide the data into training & data.
2. Choose the no. of K neighbors, say k=5.



Un-supervised learning - clustering concepts :-

Clustering concepts :-

* It is basically a unsupervised learning method.
* clustering is the task of dividing the population or data points into a no. of groups. It is basically a collection of objects on the basis of similarity and dissimilarity b/w them



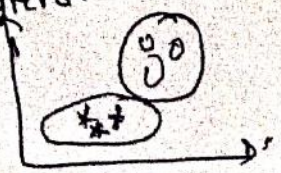
Raw Data

Types of clustering :-

1) K-means clustering.
2) Hierarchical "
3) Fuzzy "
4) Criterion function clustering.

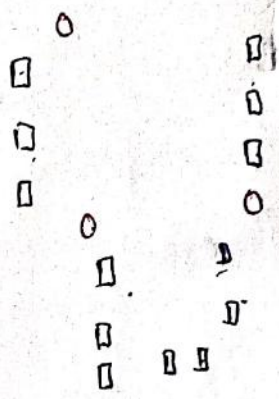
K-means clustering :-

The K-means clustering algorithms. algorithm is one of the most popular it classifies the dataset by dividing the samples into different clusters of equal variances.



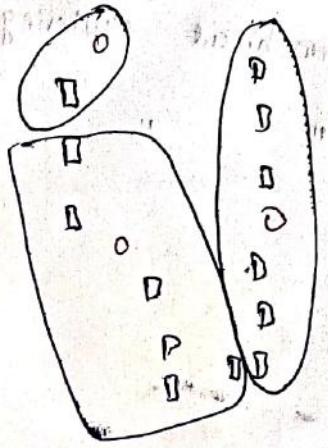
The working steps of this algorithms.

Step 1): Choose the no. of k (incase k=3) of clusters.

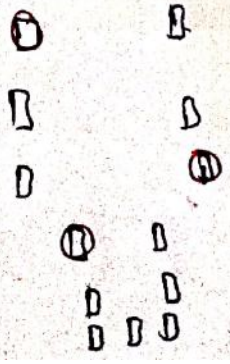


Step 2:- Select at random k points, the centroids.

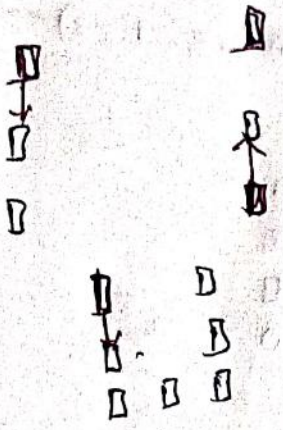
Steps :- Assign each data point to the closet centroid based on euclidance.



Step 4:- compute & place the new centroid of each clusters.
The centroid of each of the k clusters become new mean.
(shifting the circles)



Step 5:- Reassign each data point to the new closest centroid.

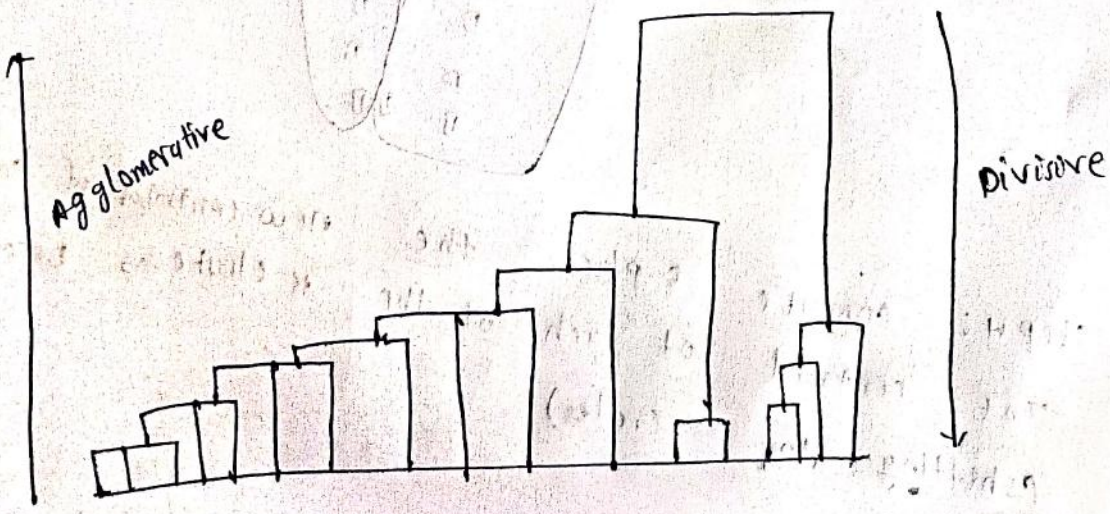


Hierarchical clustering:-

In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a dendrogram.

* There are two basic - distinctions of this algorithm

1. Agglomerative Hierarchical clustering. → follows a bottom-up
2. Divisive " " → follows a up-bottom

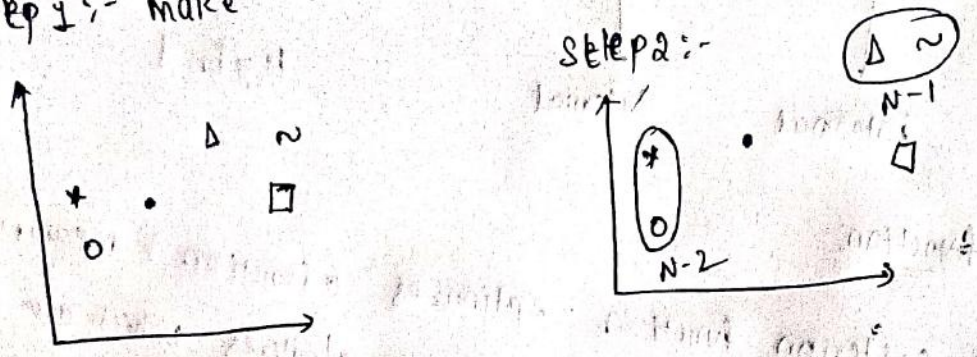


Agglomerative Hierarchical clustering :-

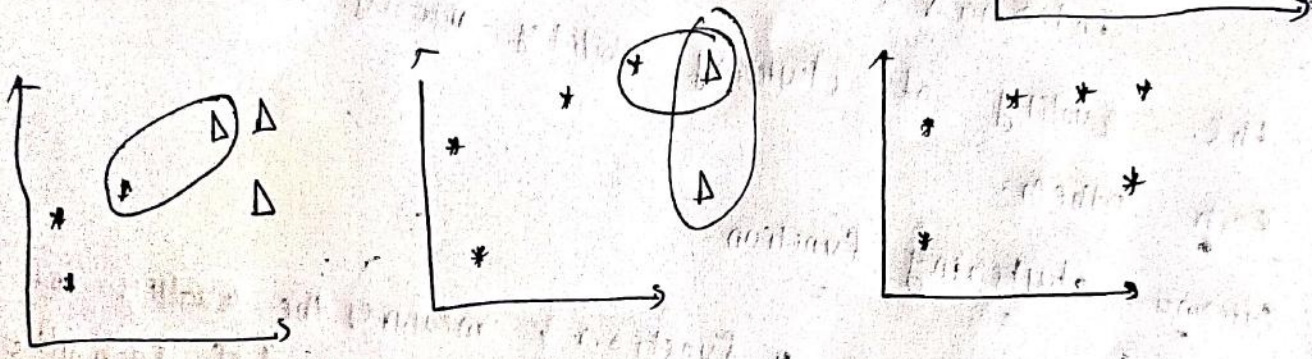
Each observation is initially considered as a cluster of its own and similar clusters are successively merged until there is but one single big root.

Step 1 :- make each data point a single point cluster.

Step 2 :-



Step 3 :-
repeat step 2 :-



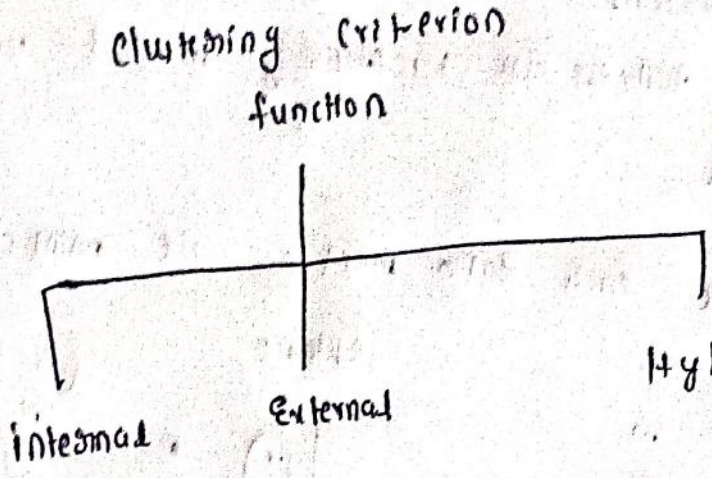
Fuzzy clustering :

Fuzzy clustering is a type of soft method in which a data object may belong to more than one group or cluster. Each dataset has a set of membership coefficients.

Criterion function clustering :-

* To measure the quality of clustering ability of any dataset, is used.
* consider a set $B = \{x_1, x_2, \ldots x_n\}$ containing 'n' samples, that is exactly in to the subsets i.e; $B_1\ B_2 \ldots B_n$;

* Sample inside the cluster will be similar to each other and dissimilar to samples in other clusters.



Internal clustering function:-

Internal criterion function optimizes a function & measure the quality of clustering ability various clusters which are d/f each other.

External clustering function:-

It optimizes a function & measures the quality of clustering ability of various clusters which are d/f each other.

Hybrid clustering:-

This function is used to ability of simultaneously optimize multiple individual functions.

UNIT-3
SYNTACTIC PATTERN RECOGNITION

Introduction

Syntactic pattern recognition or structural pattern recognition is a form of pattern recognition, in which each object can be represented by a variable-cardinality set of symbolic, nominal features. This allows for representing pattern structures, taking into account more complex interrelationships between attributes than is possible in the case of flat, numerical feature vectors of fixed dimensionality, that are used in statistical classification.

Syntactic pattern recognition can be used instead of statistical pattern recognition if there is clear structure in the patterns. One way to present such structure is by means of a strings of symbols from a formal language. In this case the differences in the structures of the classes are encoded as different grammars.

An example of this would be diagnosis of the heart with ECG measurements. ECG waveforms can be approximated with diagonal and vertical line segments. If normal and unhealthy waveforms can be described as formal grammars, measured ECG signal can be classified as healthy or unhealthy by first describing it in term of the basic line segments and then trying to parse the descriptions according to the grammars. Another example is tessellation of tiling patterns.

A second way to represent relations is graphs, where nodes are connected if corresponding sub patterns are related. An item can be labelled as belonging to a class if its graph representation is isomorphic with prototype graphs of the class.

Typically, patterns are constructed from simpler sub patterns in a hierarchical fashion. This helps in dividing the recognition task into easier subtask of first identifying sub patterns and only then the actual patterns.

Structural methods provide descriptions of items, which may be useful in their own right. For example, syntactic pattern recognition can be used to find out what objects are present in an image. Furthermore, structural methods are strong in finding a correspondence mapping between two images of an object. Under natural conditions, corresponding features will be in different positions and/or may be occluded in the two images, due to camera-attitude and perspective, as in face recognition. Structural pattern recognition assumes that pattern structure is quantifiable and extractable so that structural similarity of patterns can be assessed. I Typically, these approaches formulate hierarchical descriptions of complex patterns built up from simpler primitive elements.

1) Grammar Based Approaches:

Construct a description grammars for each clay, of objects using either hand analysis of syntactic descriptors or automated inference. Generate a set of rules, called production rules. These rules governs interconnection between the said primitives Thus, they form grammar. The set of words is called a formed language and is described by a grammar. The grammar is a mathematical model of a generator of syntactically Connect words.

Grammar may be defined as H-Tuple

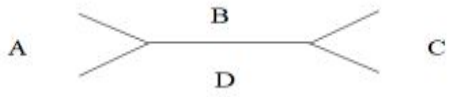
i.e, G = [Vn,Vt,P,S]

Vt= the finite set of terminal symbols or alphabets are Constants that represents the final substitution Phase of the production of sentences.

Vn = The finite set of non-terminals symbols or Variably that represent the intermediate construction.

S = The grammar axiom of the Start symbol.

P = The finite set ? production rules defining how Symbols can be Combined to form the sentences.



Here A, B, C= Terminals

v_n= The finite set of non-terminal symbols

p= The finite set of production rules

s= The grammar axiom or start the symbol.

In many cases, statistical pattern recognition does not offer good performance because statistical features do not (and cannot) represent sufficient information that is needed. Statistical pattern recognition attempts to classify patterns based on a set of extracted features and an underlying statistical model for the generation of these patterns.

Grammatical Methods

Grammars provide detailed models that underlie the generation of the sequence of characters in strings. I For example, strings representing telephone numbers conform to a strict structure. I Similarly, optical character recognition systems that recognize and interpret mathematical equations can use rules that constrain the arrangement of the symbols. I In pattern recognition, we are given a sentence (a string generated by a set of rules) and a grammar (the set of rules), and seek to determine whether the sentence was generated by this grammar.

Formally, a grammar consists of four components:

Symbols: Every sentence consists of a string of characters (or primitive symbols, terminal symbols) from an alphabet.

Variables: These are called the nonterminal symbols (or intermediate symbols, internal symbols).

Root symbol: It is a special variable, the source from which all sequences are derived.

Productions: The set of production rules (or rewrite rules) specify how to transform a set of variables and symbols into other variables and symbols.

For example, if A is a variable and c a terminal symbol, the rewrite rule cA → cc means that any time the segment cA appears in a string, it can be replaced by cc.

The language L(G) generated by a grammar G is the set of all strings (possibly infinite in number) that can be generated by G.

Types of Grammar-based Approaches

There are 3 types of approaches are:

Regular grammars: These grammars define patterns that can be recognized by finite-state machines, such as regular expressions.

Context-free grammars: These grammars allow for more complex patterns that involve hierarchical structures and recursion.

Stochastic grammars: These grammars introduce probabilistic elements to model patterns with uncertainties or variations.

2) Elements Of Formal Grammar

In formal language theory, a grammar (when the context is not given, often called a formal grammar for clarity) describes how to form strings from a language's alphabet that are valid according to the language's syntax. A grammar does not describe the meaning of the strings or what can be done with them in whatever context—only their form. A formal grammar is defined as a set of production rules for such strings in a formal language.

Formal language theory, the discipline that studies formal grammars and languages, is a branch of applied mathematics. Its applications are found in theoretical computer science, theoretical linguistics, formal semantics, mathematical logic, and other areas.

A formal grammar is a set of rules for rewriting strings, along with a "start symbol" from which rewriting starts. Therefore, a grammar is usually thought of as a language generator. However, it can also sometimes be used as the basis for a "recognizer"—a function in computing that determines whether a given string belongs to the language or is grammatically incorrect. To describe such recognizers, formal language theory uses separate formalisms, known as automata theory. One of the interesting results of automata theory is that it is not possible to design a recognizer for certain formal languages.[1] Parsing is the process of recognizing an utterance (a string in natural languages) by breaking it down to a set of symbols and analysing each one against the grammar of the language. Most languages have the meanings of their utterances structured according to their syntax—a practice known as compositional semantics. As a result, the first step to describing the meaning of an utterance in language is to break it down part by part and look at its analysed form (known as its parse tree in computer science, and as its deep structure in generative grammar).

- The formal grammar can be used to describe and recognise patterns in natural language.
- Formal grammars are used in computational to parse and analyse literature of sentences. They can be used to identify patterns in language and generate new patterns that follows the same rules.
- Formal grammar is used to mostly in the syntactic analysis phase particularly during the compilation.
- Formal grammar Plays crucial role in structural pattern recognition.
- It provides a framework for defining and analysing the syntactic structure of patterns.
- By using formal grammar we can establish rules and constraints for pattern recognition algorithms.
- Formal grammar consists of a set of symbols, rules and constraints.
- Symbols represent the elements of a pattern such as shapes or features.
- Rules define relationships and combinations of symbols within a pattern.

Types of Formal Grammar

There are different types of formal grammar, including generative grammar, transformational grammar, and dependency grammar.

- Generative grammar focuses on how sentences are generated.
- Transformational grammar studies the transformations that can occur between different sentence structures.

Regular Grammar:

It defines simple pattern with regular structures, often used for simple recognition tasks.

Context-free grammar:

Allows more complex structures and recursive patterns, widely used in natural language processing.

Context-sensitive grammar:

Provides flexibility for capturing intricate relationships between pattern elements.

Approaches

- Generative Grammar
- Recognition Grammar

Generative Grammar:

It defines a new rules for generating valid patterns from a given grammar

Recognition Grammar:

Focuses on recognizing & classifying patterns based on a set of predefined rules.

Components of Formal Grammar

- Formal grammar consists of three main components: syntax, morphology, and phonology.
- Syntax refers to the rules governing the structure and arrangement of words in a sentence.

- Morphology deals with the internal structure of words and how they can be modified or combined.

Syntax in Formal Grammar

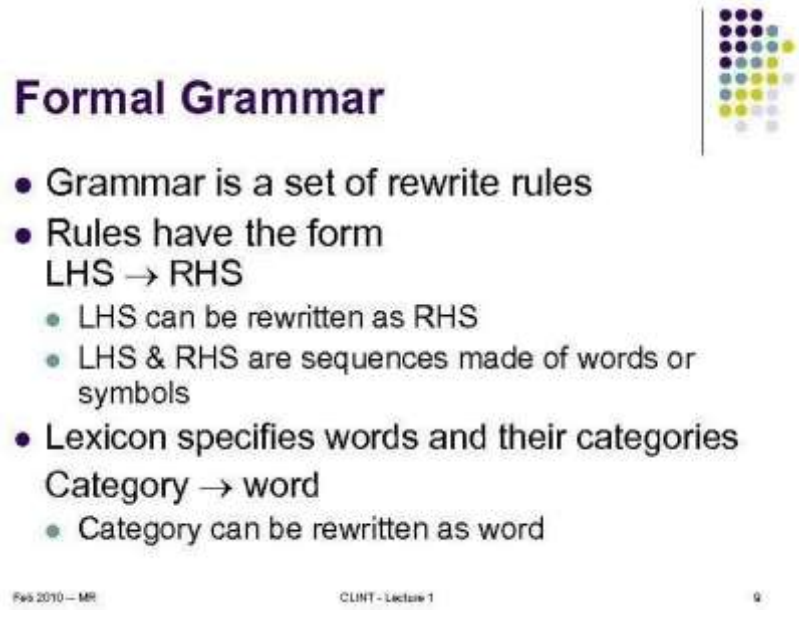
- Syntax defines the order and arrangement of words to create meaningful sentences.
- It includes rules for word order, sentence structure, and the formation of phrases.
- Syntax helps determine the grammaticality and meaning of a sentence.

Morphology in Formal Grammar

- Morphology focuses on the internal structure and formation of words.
- It includes rules for word formation, such as adding prefixes or suffixes.
- Morphology also studies the relationship between words and their inflectional forms.

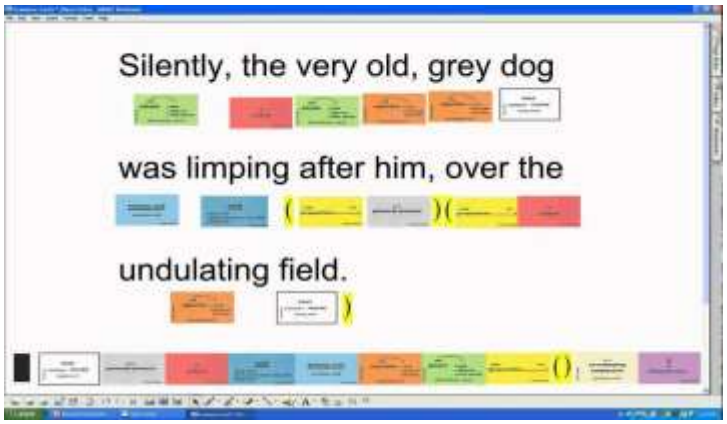
Formal Languages

- Formal grammar is closely related to formal languages.
- Formal languages are sets of strings defined by a specific grammar.
- They can be used to represent natural languages, programming languages, and mathematical expressions.



Parsing in Formal Grammar

- Parsing is the process of analysing a sentence according to the rules of a formal grammar.
- It involves breaking down the sentence into its constituent parts and determining their relationships.
- Parsing helps identify the syntactic structure and grammatical correctness of a sentence.

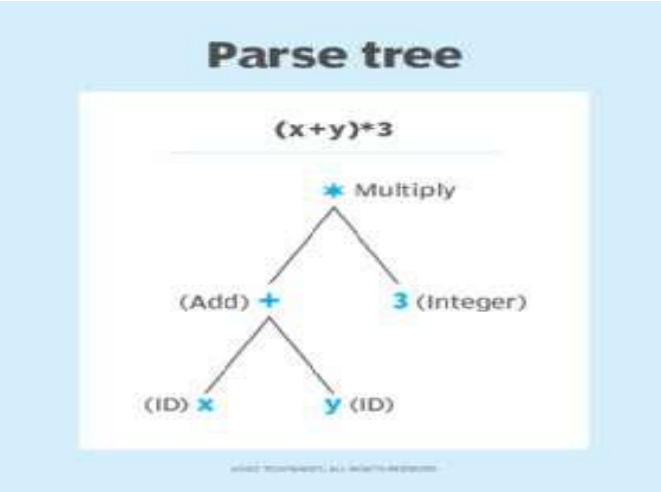


3) Parsing concept-Parsing Algorithm

Parsing is performed at the syntax analysis phase where a stream of tokens is taken as input from the lexical analyzer and the parser produces the parser tree for the tokens while checking the stream of tokens against the syntax errors.

Parsing is a grammatical exercise that involves breaking down a text into its component parts of speech with an explanation of the form, function, and syntactic relationship of each part so that the text can be understood.

The term "parsing" comes from the Latin pars for "part (of speech).“

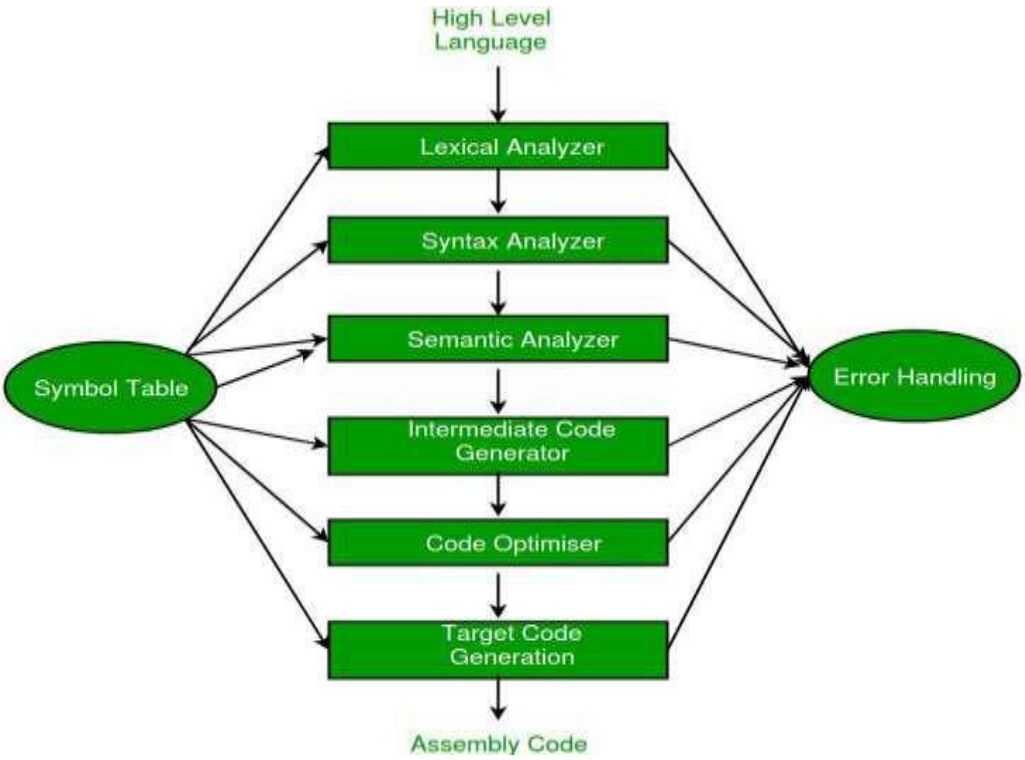


Parsing splits the input string at the matching text. It puts data up to the start of the match in one variable and data starting after the match in the next variable.

Parsing is a topic proper of Linguistics that can be considered as a basic step (syntax analysis) of the Syntactic Pattern Recognition procedure. Considering a possible application of such technique to the automatic interpretation of imaged shapes, preliminary tests have been carried out onto simple geometric forms.

Parsing Algorithm

- Parsing algorithm is a process used in computer science to analyze a sequence of symbols and determine their grammatical structure.
- It is commonly used in programming languages, natural language processing, and compilers.
- The goal of a parsing algorithm is to transform the input into a more structured representation, such as an abstract syntax tree or parse tree.



Is the pattern syntactically well formed in the context of one or more prespecified grammars.

Parser = syntactic analyser

Parsers are usually associated with grammar types. The more restrictive the grammar type, the simpler parser can be used.

A Æ BC where A,B,C ∈ VN A Æ a

where A ∈ VN, a ∈ VT

- Parser may have hierarchical structure to be more efficient
- Decomposition in subparts

Example

G1 ={VT, VN, P, S}

VT = {the, program, crashes, computer}

VN = {SENTENCE,ADJ,NP,VP,NOUN,VERB}

P = { SENTENCE Æ NP + VP, NP Æ ADJ + NOUN VP Æ VERB + NP NOUN Æ computer|program VERB Æ crashes ADJ Æ the }

S = SENTENCE

Generation using grammars

A. the program crashes the computer

B. the program crashes the program

C. the computer crashes the program

D. the computer crashes the computer

Types of Parsing Algorithms

Top-down parsing:

This algorithm starts with the highest-level rule of the grammar and attempts to match the input against it. It recursively expands the rule until the input is parsed.

Bottom-up parsing:

This algorithm starts with the input and attempts to reduce it to the highest-level rule of the grammar. It uses a stack to keep track of the parse and applies reduction rules to build the parse tree.

Left-to-right parsing:

This algorithm scans the input from left to right and constructs the

Applications of Parsing Algorithms

Compilers:

Parsing is a crucial step in the compilation process. It transforms the source code into a structured representation that can be further processed and optimized.

Natural language processing:

Parsing algorithms are used to analyze and understand human language. They can be used for tasks such as part-of-speech tagging, syntactic parsing, and semantic analysis.

Data extraction:

Parsing algorithms can be used to extract structured data from unstructured or semi-structured sources, such as web pages or log files.

Transition networks in Parsing

- A transition network is a finite state automaton that is used to represent a part of a grammar.
- A transition network parser uses a number of these transition networks to represent its entire grammar.
- Each network represents one non-terminal symbol in the grammar.

A transition network is a method of parsing which represents the grammar as a set of a finite state machine (FSM)

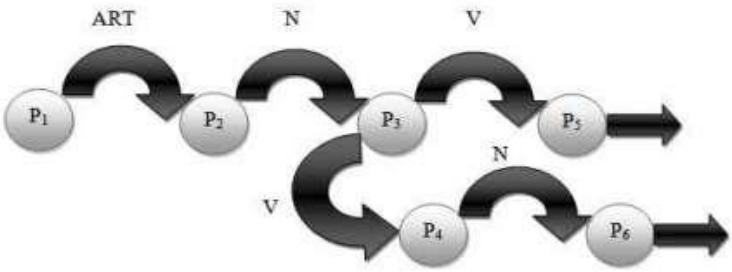


Figure Non-Deterministic Parsing Network

- The arc represents the rule or some conditions upon which the transition is made from one state to another state.

For example, a transition network is used to recognize a sentence consisting of an article, a noun, an auxiliary, a verb, an article,

a noun would be represented by the transition network as follows.

- The transition from N_1 to N_2 will be made if an article is the first input symbol.
- If successful, state N_2 is entered.

- The transition from N_2 to N_3 can be made if a noun is found next. If successful, state N_3 is entered.
- The transition from N_3 to N_4 can be made if an auxiliary is found and so on.

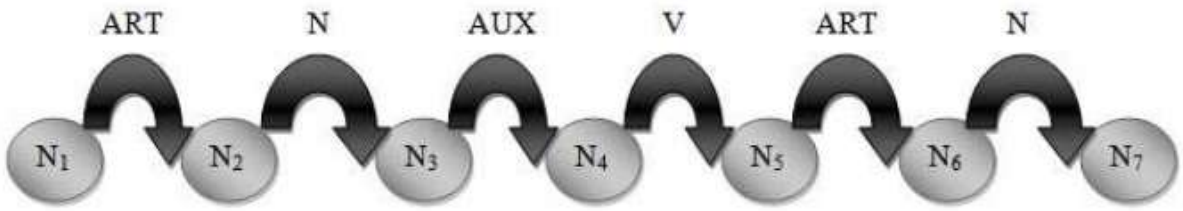


Figure Transition Network

Types of Transition Networks:

1. Augmented Transition Networks (ATNs)
2. Recursive Transition Networks (RTNs)

Augmented Transition Networks (ATNs)

An augmented transition network or ATN is a type of graph theoretic structure used in the operational definition of formal languages, used especially in parsing relatively complex natural languages, and having wide application in artificial intelligence. An ATN can, theoretically, analyze the structure of any sentence, however complicated.

An ATN is a modified transition network. It is an extension of RTN. The ATN uses a top down parsing procedure to gather various types of information to be later used for understanding system. It produces the data structure suitable for further processing and capable of storing semantic details. An augmented transition network (ATN) is a recursive transition network that can perform tests and take actions during arc transitions. An ATN uses a set of registers to store information. A set of actions is defined for each arc and the actions can look at and modify the registers. An arc may have a test associated with it. The arc is traversed (and its action) is taken only if the test succeeds. When a lexical arc is traversed, it is put in a special variable (*) that keeps track of the current word. The ATN was first used in LUNAR system. In ATN, the arc can have a further arbitrary test and an arbitrary action. The structure of ATN is illustrated in figure. Like RTN, the structure of ATN is also consisting of the substructures of S, NP and PP.

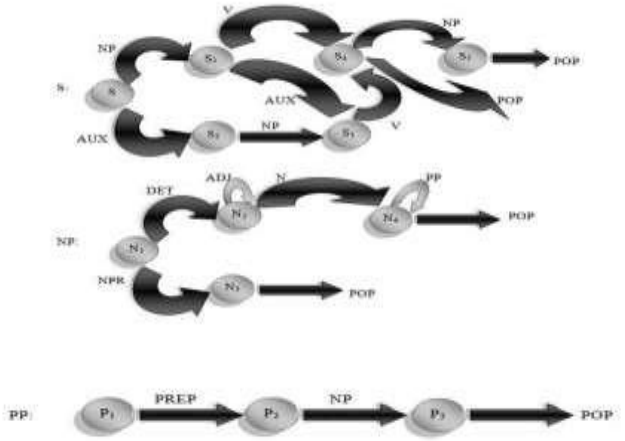


Figure ATN Structure

The ATN collects the sentence features for further analysis. The additional features that can be captured by the ATN are; subject NP, the object NP, the subject verb agreement, the declarative or interrogative mood, tense and so on. So we can conclude that ATN requires some more analysis steps compared to that of RTN. If these extra analysis tests are not performed, then there must some ambiguity in ATN. The ATN represents sentence structure by using a slot filter representation, which reflects more of the functional role of phrases in a sentence. For example, one noun phrase may be identified as "subject" (SUBJ) and another as the "object" of the verb. Wit hin noun phrases, parsing will also identify the determiner structure, adjectives, the noun etc. For the sentence "Ram ate an apple", we can represent as in figure. ATN was developed by Wiliam Woods in 1970. The ATN method of parsing sentences integrates many concepts from Chomsky's (1957) formal grammar theory with a matching process resembling a dynamic semantic network.

2) Recursive Transition Networks (RTNs):

RTN is a recursive transition network that permits arc labels to refer to other networks and they, in turn, may refer back to the referring network rather than just permitting word categories used previously

RTNs are considered as development for finite state automata with some essential conditions to take the recursive complexion for some definitions in consideration. A recursive transition network consists of nodes (states) and labeled arcs (transitions). It permits arc labels to refer to other networks and they in turn may refer back to the referring network rather than just permitting word categories. It is a modified version of transition network. It allows arc labels that refer to other networks rather than word category. A recursive transition network can have 5 types of arcs (Allen's, JM's) like

1) **CAT:** Current word must belong to category.

2) **WORD:** Current word must match label exactly.

3) **PUSH:** Named network must be successfully traversed.

4) **JUMP:** Can always be traversed.

5) POP: Can always be traversed and indicates that input string has been accepted by the network. In RTN, one state is specified as a start state. A string is accepted by an RTN if a POP arc is reached and all the input has been consumed. Let us consider a sentence "The stone was dark black".

The RTN structure is given in figure

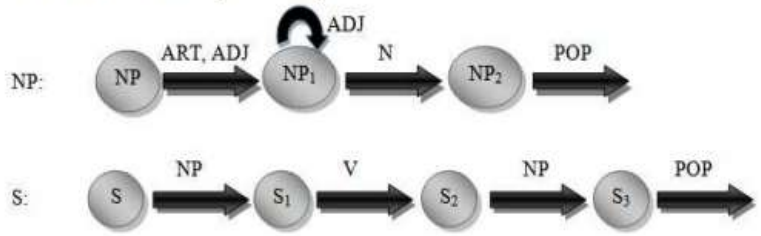


Figure RTN Structure

So we can parse the sentence through the RTN structure as follows.

Also there is an another structure of RTN is described by William Woods (1970) is illustrated in figure. He described the total RTN structure into three parts like sentence (S), Noun Phrase (NP), Preposition Phrase (PP).

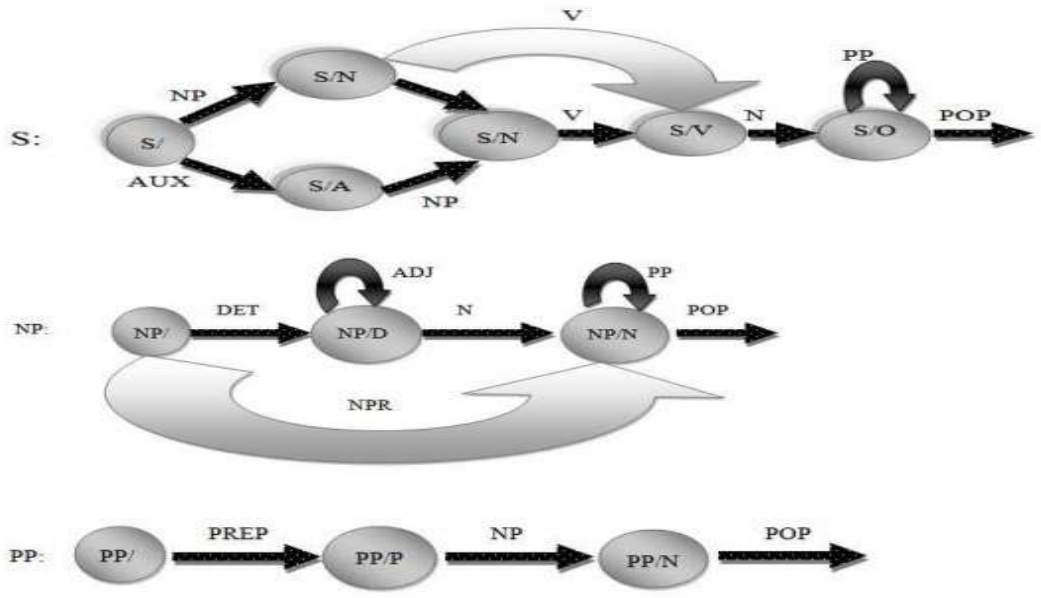


Figure RTN Structure

4) Higher Dimensional Grammars

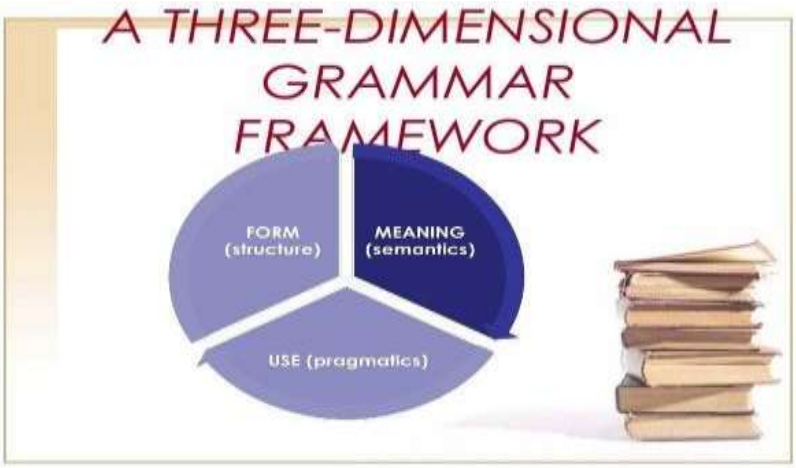
Graphical alternatives for structural representations are natural extensions of higher dimensional grammars because graphs are valuable tools for representing relational information.

A graph G = {N, R} is an ordered pair represented using:

a set of nodes (vertices), N, I a set of edges (arcs), $R \subseteq N \times N$.

A subgraph of G is itself a graph Gs = {Ns, Rs} where $Ns \subseteq N$ and Rs consists of edges in R that connect only the nodes in Ns.

- Higher dimensional grammars are a theoretical framework that extends traditional grammar models.
- They allow for the representation and manipulation of complex linguistic structures.
- Higher dimensional grammars provide a multidimensional approach to language analysis and generation.



- Higher that Dimensional Extend traditional grimaces are a theoretical framework grammar models.
- They allow for the representation and manipulation of Complex lingulitic structures.
- Higher dimensional grammars provide. a multidimensional approach to language analysts and generation.

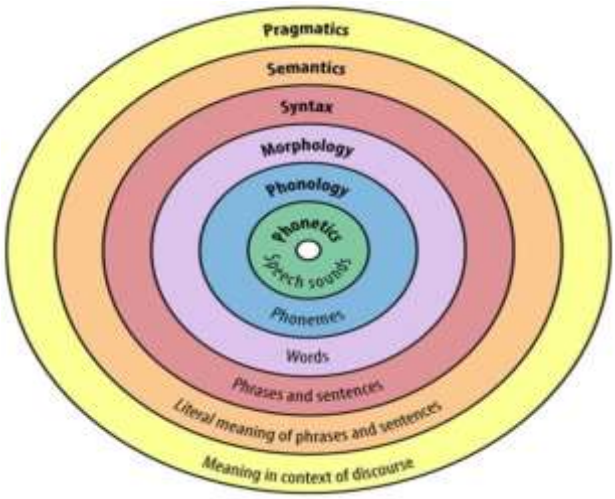
A 3D Grammar frame work is divided in to 3 types:

- Form (structure)
- USG(pragmatics)
- Meaning (semantic)

The in highest dimensional Grammars is to syntactic expressed by hyper graphs with linguistic units and edgy representing relationships. Representing it allow for the Simultaneous repro enation of different Syntactic features. Hypes graph enable the manipulation and generation of Complex Syntactic.

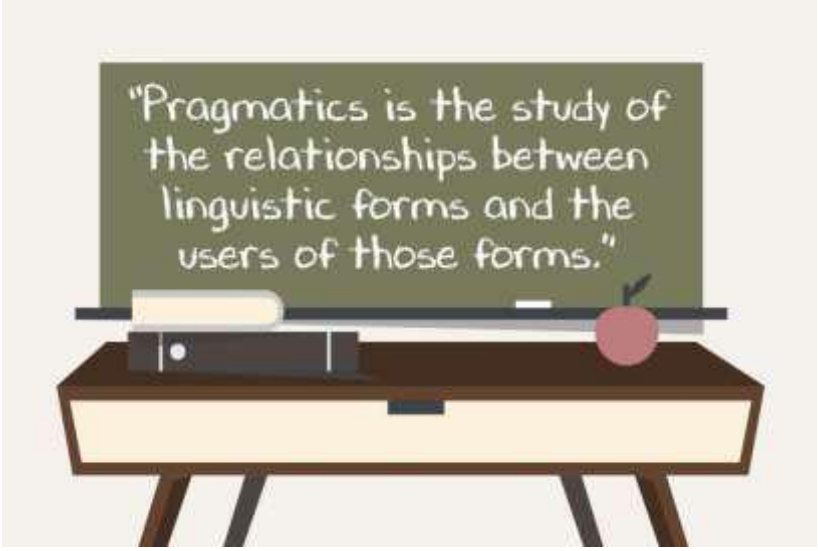
Syntax in Higher Dimensional Grammars

- Syntactic structures are represented as hypergraphs with nodes representing linguistic units and edges representing relationships.
- Multiple dimensions allow for the simultaneous representation of different syntactic features.
- Hyper graph transformations enable the manipulation and generation of complex syntactic structures.



Pragmatics in Higher Dimensional Grammars

- Higher dimensional grammars consider pragmatic aspects of language, such as context and speaker intentions.
- Pragmatic information is incorporated into the hypergraph representation to capture situational meaning.
- The multidimensional nature of higher dimensional grammars allows for the modeling of complex pragmatic interactions.



Applications of Higher Dimensional Grammars

Natural Language Processing:

Higher dimensional grammars offer a powerful framework for natural language understanding and generation.

Machine Translation:

The multidimensionality of higher dimensional grammars can improve the accuracy of machine translation systems.

Linguistic Analysis:

Higher dimensional grammars provide a versatile tool for analyzing and modeling linguistic structures.

Limitations of Higher Dimensional Grammars

- **Complexity:**

  Higher dimensional grammars can become complex, requiring sophisticated algorithms for analysis and generation.

- **Data Requirements:**

  Adequate training data is essential for effective utilization of higher dimensional grammars.

- **Computational Efficiency:**

  The multidimensional nature of higher dimensional grammars may pose computational challenges.

5) STOCHASTIC GRAMMARS

The grammar is realized as a language model. Allowed sentences are stored in a database together with the frequency uses stochastic, probabilistic and statistical methods, especially to resolve difficulties that arise because longer sentences are highly ambiguous when processed with realistic grammars, yielding thousands or millions of possible analyses. Methods for disambiguation often involve the use of corpora and Markov models. "A probabilistic model consists of a non-probabilistic model plus some numerical quantities; it is not true that probabilistic models are inherently simpler or less structural than non-probabilistic models.

Formal grammars assumed that languages generated by two grammars were disjoint, however it is rarely the case. It was not presented how to incorporate a priori information about likelihood of classes Stochastic grammar is a four-tuple

Gs = {VT, VN, Ps, Ss} production rules are of the form:

Thus, several rules with the same left side can be present in the stochastic grammar Sum of all probabilities for such rules = 1 If ≠ 1, the grammar is called fuzzy.

- A stochastic version of the Programmed Grammar is proposed as a powerful and convenient formalism for syntactic pattern recognition.
- An algorithm for parsing strings generated by Stochastic Context-Free Programmed Grammars is described and an example is presented of one such grammar which generates “noisy” squares.
- Assumptions of the formal grammar used in SyntPR

  Languages are disjoint

  No errors in the sentences produced by the grammar
- In practice the assumptions are faulty

  Errors in the primitive extraction process

  Noise or pattern deformation in descriptions

Definition

$G_s = \{V_N, V_T, P_s, S_s\}$

P_s is a set of Stochastic Productions

Each production is of form

- a_i -> b_j with probability p_{ij}

Derivations in Stochastic Language

Derivations of sentence from S_s to x

Labels $t_{k-1,k}$ where k=1 to n are given to each production such as β_{k-1} to β_k

Every production will have a probability pi

Unconditional Probability is given by

- $P(t_{0,1}$ ‘n’ $t_{1,2}$ ‘n’ … ‘n’ $t_{n-1,n}) = P(t_{0,1}).P(t_{1,2}) \dots P(t_{n-1,n})$

$P(t_{0,1},t_{1,2},\dots,t_{n-1,n}) = \Pi_{q=1 \text{ to } n} P(t_{q-1,q})$

This uses the assumption that every

- production is in d Proper Stochastic Grammar
    - Elements of Ps is of form
    - $A_i \rightarrow \beta_i$ with probability p_{ij}
    - Where $A_i \in V_N$, $\beta_i \in (V_N \cup V_T)^+$
    - $\Sigma_{k=1 \text{ to } ni} p_{ik} = 1$ (Sum of all the probabilities of each production in the Grammar is equal to 1)

Characteristic Grammar:

Remove the probability measure from the Stochastic grammar

Stochastic Languages:

$L(G_s)=\{(x, p(x))|x \in V^+, S$

derives x with probability p_j, j = 1 to k, $p(x) = \Sigma_{j=1 \text{ to } k} p_j\}$

Where p_j is the probability to parse a string x from S_S and p(x) is the total probability of deriving various strings (Say k number of strings) using the grammar. Independent of the previous one applied.

For example, x is 'abc' and productions of a grammar are

$S \rightarrow aA$ with p_1; $A \rightarrow bC$ with p_2

$B \rightarrow dC$ with p_3; $C \rightarrow eD$ with p_4

$B \rightarrow c$ with p_5; $B \rightarrow f$ with p_6

$B \rightarrow g$ with p_7; $C \rightarrow c$ with p_8

$C \rightarrow f$ with p_9; $C \rightarrow g$ with p_{10}

$D \rightarrow c$ with p_{11}; $D \rightarrow f$ with p_{12}

$D \rightarrow g$ with p_{13}

Then to get x we have S->aA->abC->abc.

Here the probability to get abc is $p(abc)=p_1.p_2.p_8$

$p_1+p_2+\ldots+p_{13} = 1$ if the given grammar is Proper Stochastic Grammar

6) Graphical Approaches

- A graph, G is represented using a set of nodes (vertices), h and edge set (arcs), R as G={h,R} where R is a subset
    - Fn Synt PR applications nodes represent the pattern primitives whereas t edges give the structural information
    - A sub graph of G is itself a graph Gs={hs,Rs} where hs & Rs are subsets of h & R, respectively
- A graph is coveted if there is a path between all pairs of its nodes
- A graph is complete if there is an edge between all pairs of its nodes
- A directed graph (digraph) is defined similar to a graph except the pair Ja,bl, which is an element of R, is defined as an edge.
- A relation from set A to set B is a subset of AxB
    - e.g. Relation “lies on" : R={(rug, floor), (chair, rug),(person, cihra)}
    - note that relations has a direction: (floor, rug) not element Rof
    - Usual notation using functions f:➔A B, b=f(a) (function==relation)
    - Relations can be higher dimensional : for (A((xB)xC)xD) □ (a,b,c,d)
- Graph represents one particular relation graphically by using an arrow to show this relation between the elements using a directed graph
- Semantic vet is a relational graph shows all the relations between its nodes using some labels
- Tree is a finite a finite cyclic (containing no closed loops or paths or cycles) digraph

- Graphical approaches are visual tools used to represent information, data, or concepts.
- They provide a clear and concise way to communicate complex ideas.
- Graphical approaches are widely used in various fields such as business, science, education, and marketing.

Types of Graphical Approaches:

Bar Graphs: Used to compare different categories or groups of data.

Line Graphs: Display trends and changes over time.

Pie Charts: Illustrate proportions and percentages of a whole.

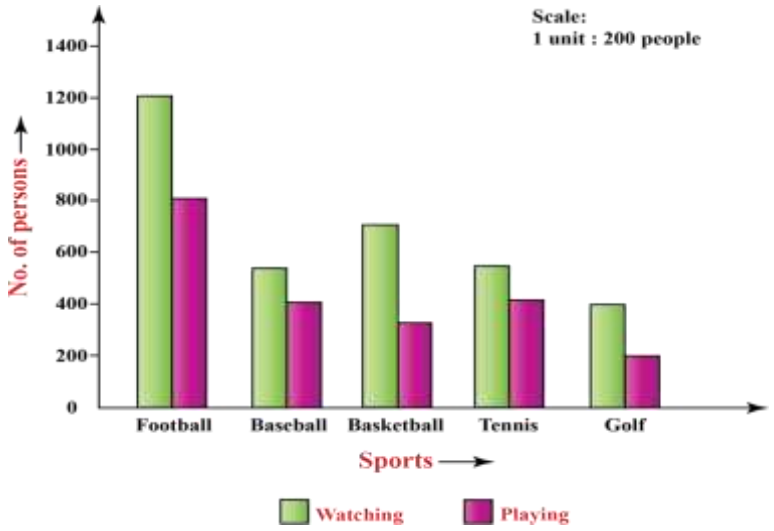
Examples of Graphical Approaches

Bar Graph Example: Comparing sales performance of different products in a given year.

Line Graph Example: Showing the population

growth of a city over a decade.

Pie Chart Example: Displaying the percentage distribution of students across different majors.



Applications of Graphical Approaches

Business: Visualizing sales data, market trends, and financial performance.

Education: Presenting student performance, attendance, and engagement metrics.

Science: Representing experimental results, data analysis, and scientific concepts.

Graph Isomorphism

- Consider two graphs, G1={h1,R1} and G2={h2,R2}
- A homomorphism /rom l l to l2 is a function f from h1 to h2
- $(v1 , w1) \hat{I} R1 \ Þ [f (v1), f (w1)] \hat{I} R2$
- An isomorphism /rom ll to l2 is a function f from h1 to h2 where f is required to be 1:1 and onto
- $(v1, w1) \hat{I} R1 \ \hat{U} [f (v1), f (w1)] \hat{I} R2$
- Isomorphism simply states that relabeling of nodes yields the same graph structure
- Unfortunately, determining graph isomorphism can be computationally expensive

- Given two graphs, G1={h1,R1} and G2={h2,R2} each with p nodes, an easy method to check isomorphism :

1) Label the nodes of each graph with labels 1,2,…,p

2) Form the adjacency graph matrices, 1Wand W2 , for two graphs

3) Ff W1=W 2 , then G1and G2 are isomorphic

4) Ff W1 is not equal to W2 , consider all p! possible labelling son G2

- There are some invariant properties which are preserved under graph isomorphism :
  - number of nodes,
  - number of arcs,
  - in-degree of a vertex,
  - out-degree of a vertex,
  - closed path of length l
  - An alternative method is to find G1and G2 unisomorphic by finding at least one property (e.g. equal number of vertices) that all isomorphic graphs must share but 1Gand G2 do not.
- G1and G2 are called sub isomorphic if a sub graph of G1 is isomorphic to a sub graph of G2

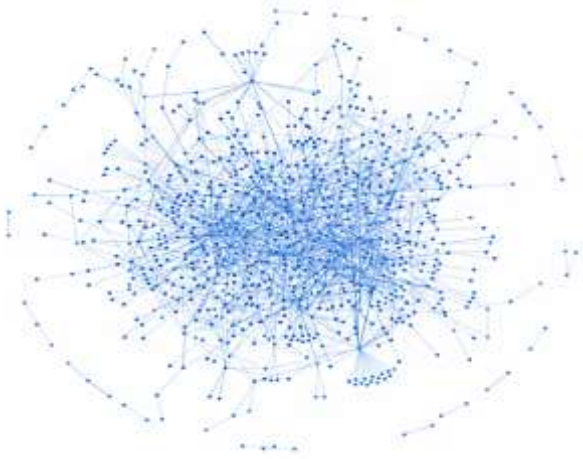
Graph Attributed

- Graphs play a crucial role in pattern recognition, allowing us to represent and analyze complex data structures.
- Graphs provide a powerful framework for capturing relationships and interactions between objects in a pattern.
- By leveraging graph attributes, we can extract meaningful patterns and make accurate predictions.

Types of Graphs Used in Pattern Recognition

- Directed graphs: Represent relationships with directional edges, allowing for capturing cause-and-effect relationships in patterns.
- Undirected graphs: Capture symmetric relationships between objects in a pattern, enabling the identification of clusters and groups.

- Weighted graphs: Assign numerical values to edges or nodes, allowing for the quantification of relationships and importance in a pattern.



Graph Attributes in Pattern Recognition

Node attributes:

Assigning characteristics to individual nodes, such as color, size, or shape, can help in distinguishing patterns or identifying key elements.

Edge attributes:

Adding attributes to edges, such as weight or distance, can provide important information about the strength or similarity of relationships in a pattern.

Graph topology:

Analyzing the overall structure of a graph, including the arrangement of nodes and edges, can reveal important patterns and motifs.

- The representation of pattern structure can be improved by includ ing numerical/symbolic attributes of pattern prim itivensai graph
- An attributed graph, Gi, is a 3-tuple : Gi={hi,Pi,Ri} where
    - hi: a set of nodes
    - Pi : a set of properties of these nodes
    - Ri : a set of relations between nodes

- Measuring the transformation difference between graphs
- Fn order to transform graphG_i to graph G_j, a similarity measure $D(G_i, G_j)$ must be defined with these properties :
    - $D(G_i, G_i) = O$
    - $D(G_i, G_j) > O$ for i not equal to j
    - $D(G_i, G_j) = D(G_j, G_i)$ (equal to cos ts for insertion/deletion)
    - $D(G_i, G_j) \leq D(G_i, G_k) + D(G_k, G_j)$ (triangle inequality)
    - Distance measure can be defined as

$D = \min\{Ds (x)\}$

where Ds (x) = wnicni + wndcnd + weicei + wedced + wncn (x)

w : weight for corresponding transformation ni : node insert, nd : node delete

c : cost for corresponding transformation ei : edge insert, ed : edge delete

cn (x) = å fn (pi , qj)

where fn (pi , q j) similarity measure between pi of Gi and qj of Gj

x : denotes a node mapping (configuration) between tw o graphs

Applications of Graphs in Pattern Recognition

Image recognition: Graphs can be used to represent and analyze image data, capturing spatial relationships between pixels or objects within an image.

Social network analysis: Graphs enable the study of social interactions, identifying communities, influential individuals, and patterns of information flow.

Bioinformatics: Graph-based approaches are used to analyze biological networks, such as protein-protein interactions, gene regulatory networks, and metabolic pathways.

UNIT-4

PATTERN PRE-PROCESSING & FEATURE SELECTION

Introduction

A feature is an attribute that has an impact on a problem or is useful for the problem, and choosing the important features for the model is known as feature selection. Each machine learning process depends on feature engineering, which mainly contains two processes; which are Feature Selection and Feature Extraction. Although feature selection and extraction processes may have the same objective, both are completely different from each other. The main difference between them is that feature selection is about selecting the subset of the original feature set, whereas feature extraction creates new features. Feature selection is a way of reducing the input variable for the model by using only relevant data in order to reduce over fitting in the model.

So, we can define feature Selection as, "It is a process of automatically or manually selecting the subset of most appropriate and relevant features to be used in model building." Feature selection is performed by either including the important features or excluding the irrelevant features in the dataset without changing them.

Need for Feature Selection

Before implementing any technique, it is really important to understand, need for the technique and so for the Feature Selection. As we know, in machine learning, it is necessary to provide a pre-processed and good input dataset in order to get better outcomes. We collect a huge amount of data to train our model and help it to learn better. Generally, the dataset consists of noisy data, irrelevant data, and some part of useful data. Moreover, the huge amount of data also slows down the training process of the model, and with noise and irrelevant data, the model may not predict and perform well. So, it is very necessary to remove such noises and less-important data from the dataset and to do this, and Feature selection techniques are used.

1) Distance measures in pattern recognition

A distance measure is an objective score that summarizes the relative difference between two objects in a problem domain.

Most commonly, the two objects are rows of data that describe a subject (such as a person, car, or house), or an event (such as a purchase, a claim, or a diagnosis).

Perhaps the most likely way you will encounter distance measures is when you are using a specific machine learning algorithm that uses distance measures at its core. The most famous algorithm of this type is the k-nearest neighbors algorithm, or KNN for short.
In the KNN algorithm, a classification or regression prediction is made for new examples by calculating the distance between the new example (row) and all examples (rows) in the training dataset. The k examples in the training dataset with the smallest distance are then selected and a prediction is made by averaging the outcome (mode of the class label or mean of the real value for regression).

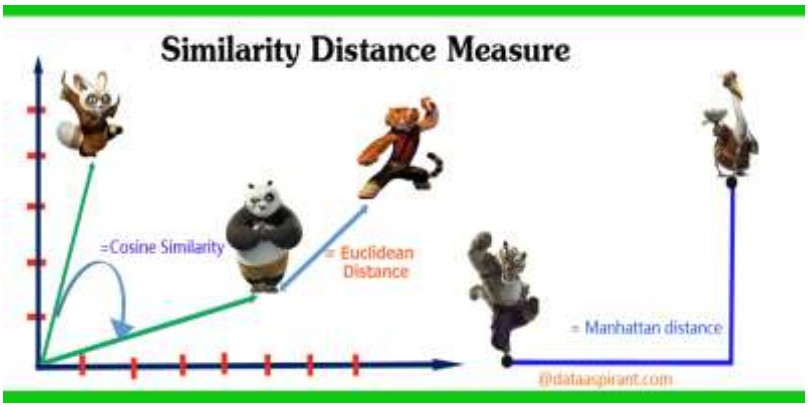
The distance measure is one of the proximity measures to classify an unknown pattern to its nearest class. In other words, it finds the dissimilarity between patterns.

The distance function could be metric or non-metric.

A similarity function holds properties like:

1. Positive Reflexivity:- $f(x,x)=0$
2. Symmetry:- $f(x,y)=f(y,x)$
3. Triangular Inequality $f(x,y)\leq f$
4. Triangular Inequality $f(x,y)\leq f(x,z)+f(z,y)$

Any similarity function that obeys all the above conditions is said to be ***metric*** & others that do not obey either the triangular inequality or symmetry are said to be ***non-metric***.



Metric Measures are of the following types:-

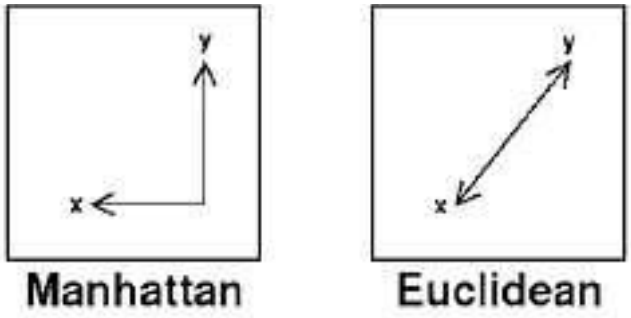
Minkowski Distance:-

$$D = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

when p=1, it is called Manhattan distance.

when p=2, it is called Euclidean distance.

To understand it better consider this image.



Mahalanobis Distance:-

$$D^2 = (x - m)^T \cdot C^{-1} \cdot (x - m)$$

where,

- D^2 is the square of the Mahalanobis distance.
- x is the vector of the observation (row in a dataset),
- m is the vector of mean values of independent variables (mean of each column),
- C^(-1) is the inverse covariance matrix of independent variables.

(x — m) is essentially the distance of the vector from the mean. We then divide this by the covariance matrix.

If the variables in your dataset are strongly correlated, then, the covariance will be high. Dividing by a large covariance will effectively reduce the distance.

Weighted Euclidean Distance:-

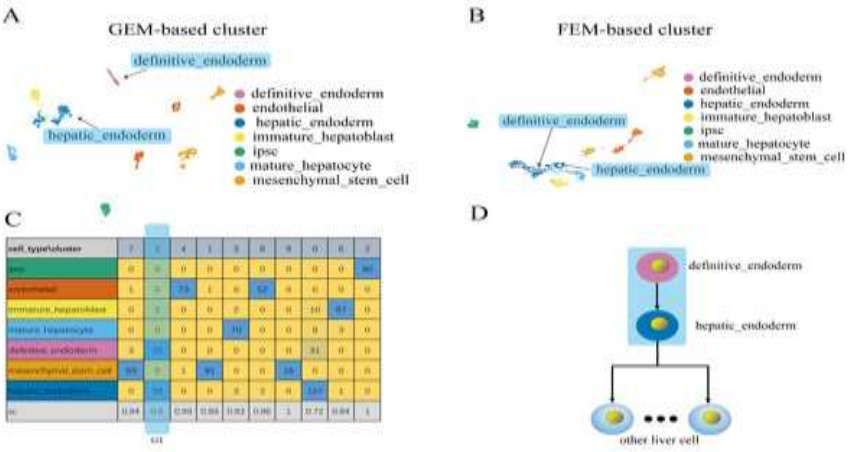
$$d_{WE}(x, y) = \left(\sum_{k=1}^{p} w_k (x_k - y_k)^2 \right)^{\frac{1}{2}}$$

Here W is the weight corresponding to the distance between point x&y.

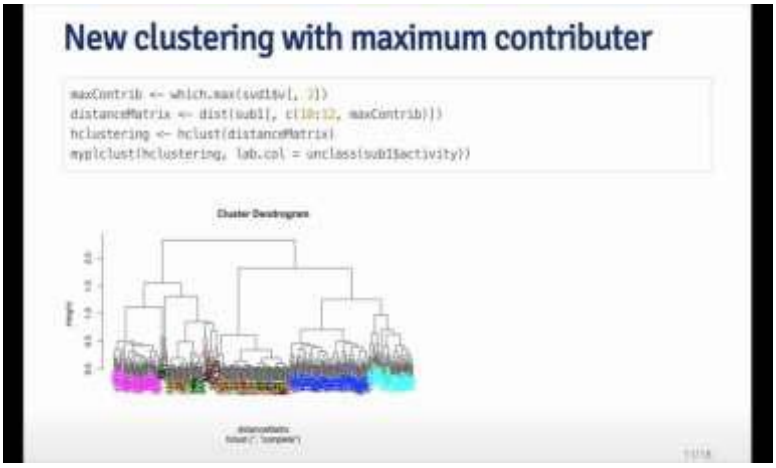
2) Clustering Transformations and feature ordering

Clustering Transformations

- Clustering transformation helps to identify patterns and similarities in the data by grouping similar data points together.
- It is commonly used for exploratory data analysis and feature engineering.
- Clustering algorithms such as K-means, hierarchical clustering, and DBSCAN are frequently employed for performing clustering transformation



- Clustering transformation is essential for exploratory data analysis.
- It helps in identifying outliers and anomalies in the dataset.
- Clustering transformation aids in feature engineering and dimensionality reduction



Types of Clustering Algorithms

K-means clustering:

Divides the data into k clusters based on the mean distance.

Hierarchical clustering:

Forms a hierarchy of clusters by recursively merging or splitting them.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

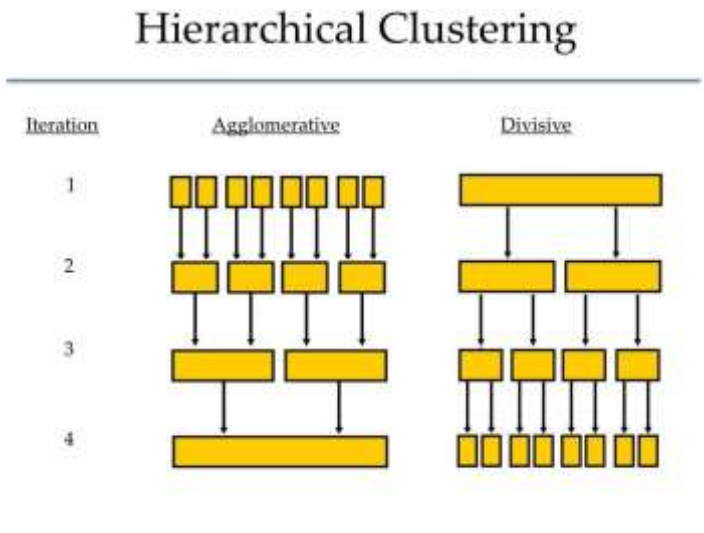
Identifies clusters based on density and connectivity.

K-means Clustering

- K-means clustering aims to minimize the sum of squared distances between data points and their cluster centroid.
- It requires specifying the number of clusters (k) in advance.
- The algorithm iteratively assigns data points to clusters based on their distances to the cluster centroids and updates the centroids accordingly.

Hierarchical Clustering

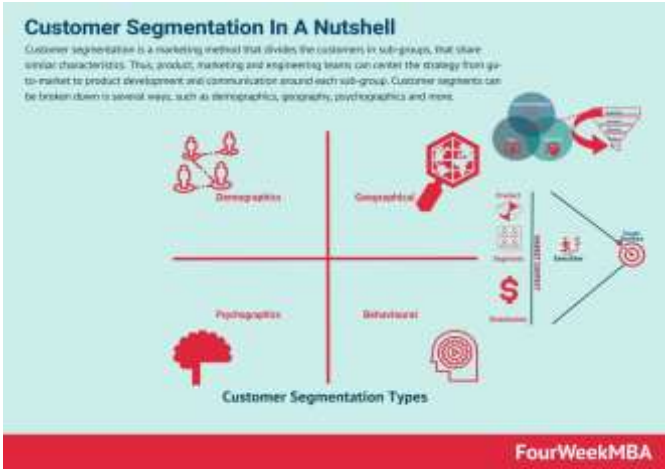
- Hierarchical clustering can be agglomerative (bottom-up) or divisive (top-down).
- Agglomerative clustering starts with each data point as a separate cluster and merges them until a desired number of clusters is obtained.
- Divisive clustering starts with the entire dataset as a single cluster and recursively splits it into smaller clusters.



Applications of Clustering Transformation

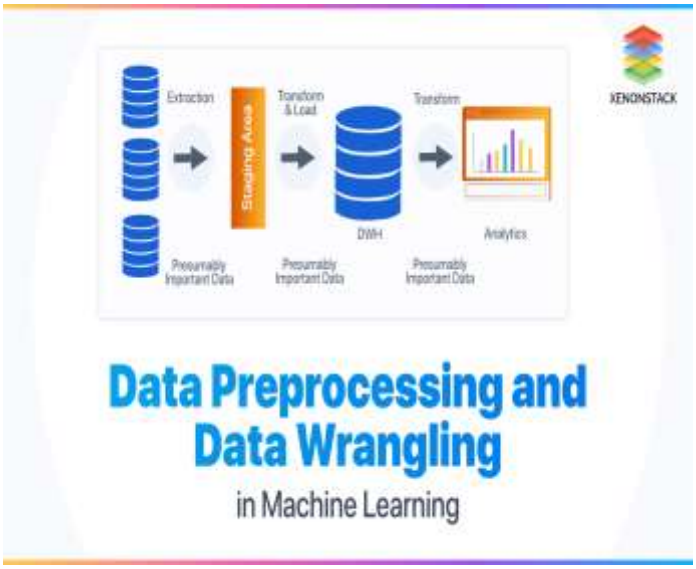
- Customer segmentation: Clustering helps in identifying groups of customers with similar characteristics for targeted marketing campaigns.
- Image segmentation: Clustering aids in dividing an image into regions based on similarities in color or texture.

- Anomaly detection: Clustering can identify data points that deviate significantly from the norm, indicating potential anomalies or fraud.



Feature Ordering

- Feature ordering involves arranging the features of a dataset in a specific order to optimize the performance of a machine learning model.
- It helps in reducing the complexity of the model and improving its interpretability.
- Feature ordering can be done based on domain knowledge, statistical techniques, or using machine learning algorithms such as random forests.

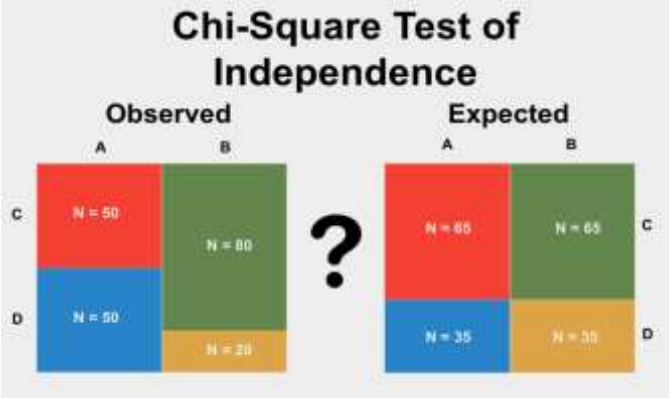


Importance of Feature Ordering

- Feature ordering helps in identifying the most important features for prediction or classification tasks.
- It reduces the computational complexity of the model by eliminating irrelevant or redundant features.
- Proper feature ordering can enhance model accuracy, speed up training and inference time, and improve model interpretability.

Techniques for Feature Ordering

- Statistical techniques such as correlation analysis, mutual information, and chi-square tests can be used for feature ordering.
- Machine learning algorithms like random forests, gradient boosting, and genetic algorithms can also be employed for feature ordering.
- Domain knowledge and expert judgment play a crucial role in determining the appropriate feature ordering techniques for a specific problem.



Feature Ordering Process

The feature ordering process involves the following steps:

Data preprocessing: Handling missing values, encoding categorical variables, and scaling numerical features.

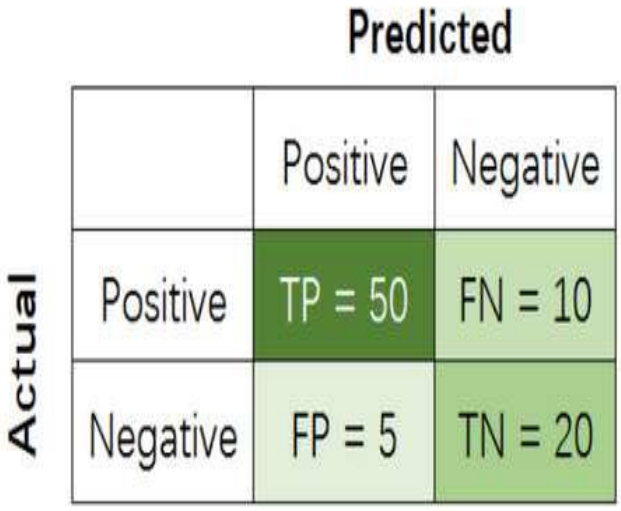
Feature selection: Identifying relevant features using statistical techniques or embedded methods.

Feature ranking: Ranking features based on their importance or relevance to the target variable.

Feature ordering: Arranging features in a specific order based on their ranking or importance.

Evaluating Feature Ordering

- The impact of feature ordering on model performance can be evaluated using metrics such as accuracy, precision, recall, and F1 score.
- Cross-validation techniques can be used to assess the robustness of feature ordering methods.
- Comparing different feature ordering techniques and their impact on model performance can provide insights into the effectiveness of each approach.

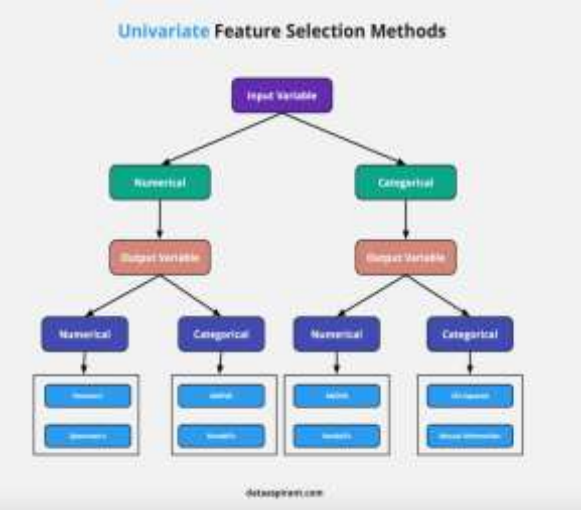


3) Clustering In Feature Selection through entropy minimization

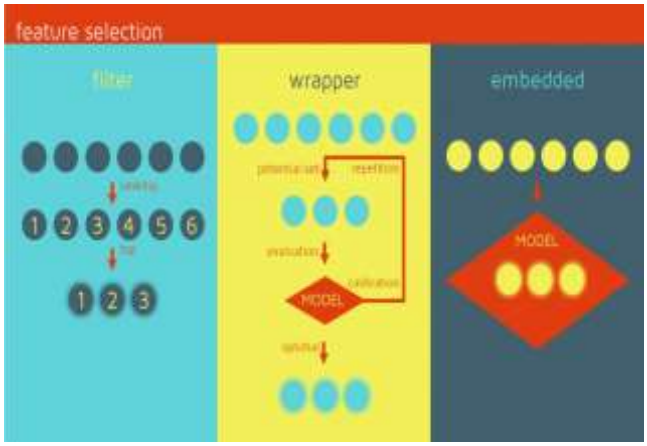
- Clustering in feature selection through entropy minimization.
- Importance of feature selection in machine learning.
- Entropy as a measure of uncertainty in data.

- The process of selecting relevant features from a dataset.
- Reduces dimensionality and improves model performance.
- Various methods for feature selection: filter, wrapper, and embedded approaches.

4) Binary Feature Selection



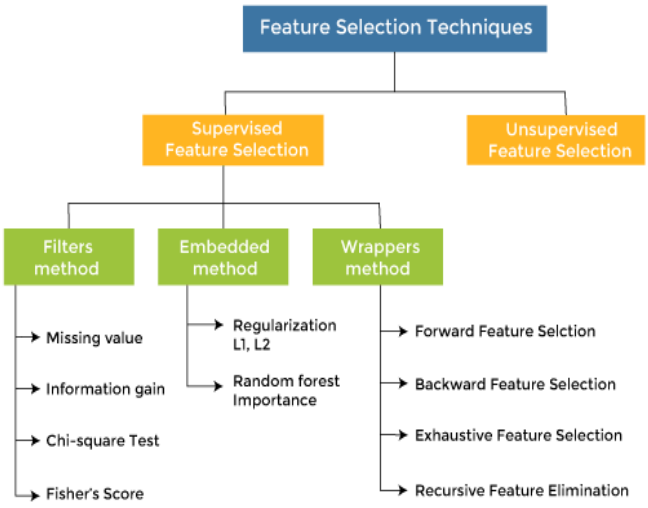
- Binary feature selection is a technique used in machine learning to choose the most relevant features for a binary classification problem.
- It aims to improve model performance by reducing the dimensionality of the input data.
- This process involves selecting a subset of features that have the most impact on the target variable.
- Feature selection helps in reducing over fitting by removing irrelevant or redundant features.
- It improves model interpretability by focusing on the most important features.
- By reducing the number of features, it also reduces the computational complexity and training time.



Feature Selection Techniques

There are mainly two types of Feature Selection techniques, which are:

- **Supervised Feature Selection technique**
  Supervised Feature selection techniques consider the target variable and can be used for the labelled dataset.
- **Unsupervised Feature Selection technique**
  Unsupervised Feature selection techniques ignore the target variable and can be used for the unlabelled dataset.



Filter methods:

These methods select features based on statistical measures like correlation or mutual information with the target variable.

Some common techniques of Filter methods are as follows:

- Information Gain
- Chi-square Test
- Fisher's Score
- Missing Value Ratio

Information Gain: Information gain determines the reduction in entropy while transforming the dataset. It can be used as a feature selection technique by calculating the information gain of each variable with respect to the target variable.

Chi-square Test: Chi-square test is a technique to determine the relationship between the categorical variables. The chi-square value is calculated between each feature and the target variable, and the desired number of features with the best chi-square value is selected.

Fisher's Score:

Fisher's score is one of the popular supervised technique of features selection. It returns the rank of the variable on the fisher's criteria in descending order. Then we can select the variables with a large fisher's score.

Missing Value Ratio:

The value of the missing value ratio can be used for evaluating the feature set against the threshold value. The formula for obtaining the missing value ratio is the number of missing values in each column divided by the total number of observations. The variable is having more than the threshold value can be dropped.

Wrapper methods:

These methods evaluate the performance of a model with different subsets of features, usually using a search algorithm.

Some techniques of wrapper methods are:

- **Forward selection** - Forward selection is an iterative process, which begins with an empty set of features. After each iteration, it keeps adding on a feature and evaluates the performance to check whether it is improving the performance or not. The process

continues until the addition of a new variable/feature does not improve the performance of the model.

- **Backward elimination** - Backward elimination is also an iterative approach, but it is the opposite of forward selection. This technique begins the process by considering all the features and removes the least significant feature. This elimination process continues until removing the features does not improve the performance of the model.
- **Exhaustive Feature Selection-** Exhaustive feature selection is one of the best feature selection methods, which evaluates each feature set as brute-force. It means this method tries & make each possible combination of features and return the best performing feature set.
- **Recursive Feature Elimination-**
  Recursive feature elimination is a recursive greedy optimization approach, where features are selected by recursively taking a smaller and smaller subset of features. Now, an estimator is trained with each set of features, and the importance of each feature is determined using coef_attribute or through a feature_importances_attribute.

Embedded methods:

These methods incorporate feature selection as part of the model training process, like L1 regularization in logistic regression.

These methods are also iterative, which evaluates each iteration, and optimally finds the most important features that contribute the most to training in a particular iteration. Some techniques of embedded methods are:

- **Regularization**- Regularization adds a penalty term to different parameters of the machine learning model for avoiding overfitting in the model. This penalty term is added to the coefficients; hence it shrinks some coefficients to zero. Those features with zero coefficients can be removed from the dataset. The types of regularization techniques are L1 Regularization (Lasso Regularization) or Elastic Nets (L1 and L2 regularization).
- **Random Forest Importance** - Different tree-based methods of feature selection help us with feature importance to provide a way of selecting features. Here, feature importance specifies which feature has more importance in model building or has a

great impact on the target variable. Random Forest is such a tree-based method, which is a type of bagging algorithm that aggregates a different number of decision trees. It automatically ranks the nodes by their performance or decrease in the impurity (Gini impurity) over all the trees. Nodes are arranged as per the impurity values, and thus it allows to pruning of trees below a specific node. The remaining nodes create a subset of the most important features.

UNIT- 5

Applications of pattern Recognition

1) Applications of pattern Recognition:

The various applications of pattern recognition are:

1. Machine Vision: A machine vision system captures images Via a camera and analyzer them to produce descriptions of images = d objects.

- For example, during inspection in manufacturing industry, when the manufactured objects are passed through the camera, the Images have to be analyzed online.

2. Computer Aided Diagnosis: CAD helps to assist doctors in making diagnostic decision. Computer assisted diagnosis has been applied in medical field such as X - rays, ECG's ultrasound images etc.

3. Speech Recognition: This process recognizes the spoken information. In this the Software in built around a pattern recognition system which recognizes the spoken text and translates it into ASCII characters which are shown on the Screen. In this we can also identify the identity of Speaker.

4. Character Recognition: This application recognizes both letter and number. In this the optically scanned image is provided as input and alphanumeric characters are generated as olp . It's major implication is in automation and information handling. It is also used in page readers, Zip Code, license plate etc.

5.Manufacturing: In this the 3-D images such as Structured light, laser, stereo etc., is provided as input and as a result we can identify the objects.

6. Fingerprint Identification: In this the ilp image is obtained from fingerprint Sensors and by this technique various fingerprint classes are obtained and we can identify the owner of the fingerprint.

7. Industrial Automation: In this we provide the intensity or the ranges of images of the product and by this the defective / non - defective product is identified.

8. Seismic Activity Analysis: when observing how earthquakes and other natural calamities disturb the Earth's Crust, pattern recognition is an effective tool to study such earthly parameters. Researchers can study the Seismic records & identify recurring patterns to develop disaster - resilient models that can mitigate Seismic effects on time.

9. Robotics: Now-a-days, the detection of radioactive material is performed by robots. These machines use pattern recognition to complete the task. In this case, the robot's Camera Captures images of a mine, Extracts the discriminative features and uses classification algorithms to segregate images into dangerous/non - dangerous based on the detected features.

10. Cyber Security: organizational networks Can use pattern Recognition - based security systems that detect activity trends and respond to changing user behaviour to block potential hackers. If cybersecurity teams have instant access to malware patterns, they can take appropriate action before an attack / threat hits the network.

2) Formal Language Theory:

Language:

Language is an organized system of signs, symbols and rules.

There are two types of languages:

1. Natural language.
2. Formal language.

- Natural language may refer either to the specifically human capacity for acquiring and using Complex systems of Communication or to a specific instance of such a System of Complex Communication.
- Natural language can be band on visual rather than auditive stimuli, for example in sign languages and written language.
- Natural language is Created Spontaneous and evolutionary. It is hand to define this language
- Formal language is a set of words i.e . , finite Strings of letters , Symbols or tokens . In Computer science they are used for the precise definition of data formats and the syntax of programming languages.

Elements of language:

- **Syntax:** Syntax is the study of the principles and rules for Constructing sentences in natural languages.

- **Semantics:** It is the study of meaning. It focuses on the relation b / w signifiers, such as woods, phrases, signs and symbols and what they stand for Linguistic Semantics is the study of meaning that is used by humans to express themselves through languages.

- **Pragmatics:** It is a subfield of linguistics which studies the ways in which context Contributes to meaning.
- It studies how the transmission of depends not only on the linguistic knowledge of the Speaker & listeners, but also on the Context of the utterance, knowledge about the status of those involved, the inferred intent of the Speaker.
- The formal language is a set of strings over a finite alphabet. " Formal Language theory " is the study of formal languages, or often more accurately the study of families of formal languages.
- Formal language theory is concerned with the purely Syntactical aspects, rather than a Semantics or meaning of the strings.
- It is closely related to automata theory, which deals with formally defined machines that accept formal Languages.
- A formal machine takes strings of symbols as ilp and olp's either " yes " or " no ".
- A machine is said to accept a language if it says " yes " to all and only those strings that are in the language.
- Formal languages can be grouped into a series of successively large classes known as the " chomsky hierarchy ".
- Most of the classes can be categorized in two ways:
- By the types of rules that can be used to generate the set of strings.
- By the type of formal machine that is capable recognizing the language.
- As we have seen regular languages are defined by using Concatenation, alternation and recursion and are recognized by a scanner.
- Content - free languages are a proper Superset of the regular languages. They are defined by using Concatenation, alternation and recursion and are recognized by a parser
- A Scanner is a Concrete realization of a finite automaton, a type of formal machine.
- A parser is a Concrete realization of a push - down automaton.
- An automation recognizes a language
- A Grammar generates a language.
- For " good classes of grammars, it is possible to build an automation, MG_1 from the grammar, G in the class, so that MG recognizes the language L (G) generated by the grammar G.

However, grammars are non - deterministic in nature.

Types of grammars:

1. Regular grammars
2. Context - free grammars

Alphabets:

Language is a set of strings. A string is a finite sequence of letters from some alphabet.

- An alphabet Σ is any finite set.

  Σ= {a1, - - - - , ak}.

  ai are called the Symbols of the alphabet.

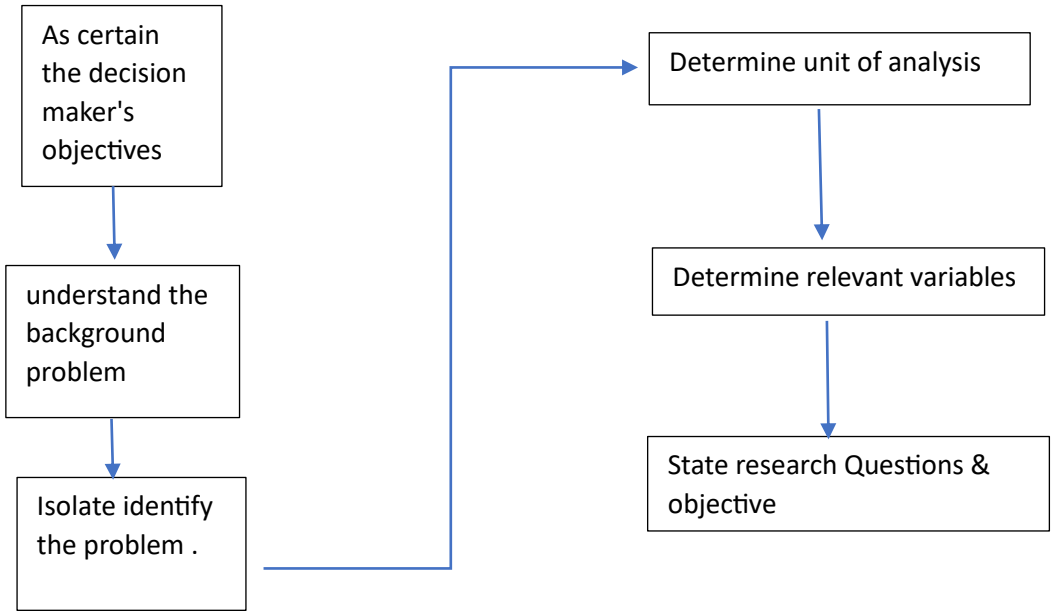
3) Formulation of Syntactic pattern recognition problem :

- Syntactic pattern Recognition involves the identification of Specific Patterns or structures with in a given input.
- Formulating Syntactic pattern recognition problems requries defining the problem, Specifying the input, & determining the desired output.
- The Famulation stage is crucial in designing effective pattern Recognition System.
- Syntactic pattern Recognition involves analyzing and Recognizing patterns in data used grammer - based approaches
- These problems arise in various fields such as natural language processing, Speech Recognition, image processing.
- The Formulation of Syntactic pattern recognition problems involves defining the problem, selecting appropriate data recognition and designing the recognition algorithm.

Problem Definition:

- The problem defined by Specifying the input data, desired output, and the relationship between them.
- The problem definition stage involves understanding the Objective of the pattern recognition task.
- It includes determining the type of patterns to be recognized, Such as text, images or Sequences.
- The problem definition also Includes identifying any constraints or requirements for the recognition talk.

The process of problem definition.



In this it is divided into a type of specifications.

1) Input Specification

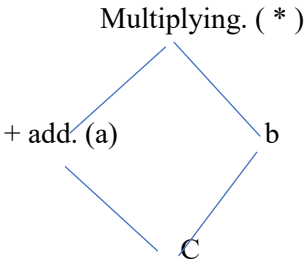
2) output

Input Specification:

- Specifying the input is essential for formulating syntactic pattern recognition problems.
- The Input can vary depending on the type of pattern being recognized, Such as textual data, images, or time series.

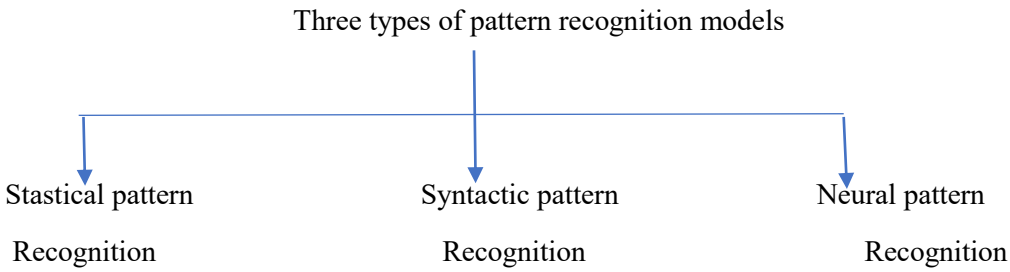
- It is important to consider the size, format, and quality of the input data during the formulation stage.

Eg:-(a + b) * c.



Output Specification

- Defining the desired output is crucial for Evaluating the performance pattern of the recognition system.
- The output can be binary (Eg .. presence or absence of a pattern) or Categorical (Eg.. Classification into different layers).
- The desired output Specification helps in determining the Evaluation metrics for the. recognition system.



Pattern Representation:

- Pattern Representation involves converting the input data into a suitable format for recognition.
- It Includes selecting appropriate features or descriptors that captures the relevant information about the pattern.
- The Choice of pattern representation affects the accuracy and efficiency of the recognition System.

4) Syntactic pattern description:

- Syntactic pattern description is a method and to analyze and describe structure of sentence.
- It focuses on the arrangement and relationship of words within a sentence.
- This technique helps to identify and understand the grammatical Structure of a Sentence
- Syntactic pattern description is a technique used in computational linguistics.
- It involves describing the Structure and arrangement of words in a Sentence or phrase.
- This technique helps to understanding and processing natural language.
- Syntactic patterns consist of a combination of words, phrases and clauses.
- Pattern can include the subject, verb, object----
- These Components work together to form a complete sentence and convey meaning.

This can be divided into 3 types :-

1) Subject - Verb (SV) pattern

2) Subject - verb - object (SVO) pattern

3) Subject - verb - Complement (SVC) pattern

1) Subject - verb (SV) pattern: A basic Structure where a subsect performs an action or state described by the verb.

2) Subject -verb - object (SVO) pattern: A common pattern where a Subject performs an action on an object.

3) Subsect - Verb - Complement (svc) pattern: A pattern where a Subject is linked to a complement that describes or rename it

- This is to analyze a sentence using Syntactic patterns breaks it down into its constituent ports.
- Identify the Subject, verb, object in the sentence.
- Determine the order and relationship of these components within the sentence.

Eg: sentence:- " The cat chased the mouse " .

Syntactic pattern:- SVO (subject - verb- object)

Subject :- The cat

Verb :- chased

Object :- Mouse

5) Recognition Grammars :

- Grammars can be used to create a definition of the Structure of Each pattern class.
- Recognition grammars are a type of formal Grammar used in computer Science and linguistics.
- They are a tool for Modelling and analyzing
- Structure of languages or patterns in data
- Recognition grammars can be used for tasks such as parsing, pattern matching& language generation

Type 0 :- Recognized by Tung machine

Type 1 :- Accepted by linear bound Automata

Type 2 :- Accepted by push Down Automata

Type 3. :- Accepted by finite Automata .

The types of Recognition Grammars are several types of including regular Grammar, content - free grammars & context - sensitive Grammars.

- So, Here the. Regular Grammar are the simplest type, where production rules are of the from A to aB or A to a, and can be recognized by a finite machine.
- Content free grammars have production rules of the form A -alpha, where alpha can be can be c combination of terminals and non - terminals

Parsing with Recognition Grammars :-

- Parsing is the process of analyzing a sequence of Symbols according to the rules of a grammar
- Recognition grammars can be used for top to down or bottom to up parsing
- Top - down parsing Starts from the start symbol and fries to derive the inpu Sequenc by Expanding non - terminals .

Applications :

- It has various applications in computer science and intelligence
- In computer science they are used for programming languages compilers, neutral language processing
- In linguistics, they are used for analysing the structure of natural languages, general sentence.

6) Automata as pattern recognition

- Automata have been successfully applied in various fields, including bio-metric pattern recognition.
- Bio-metric data refers to unique physical or behavioural characteristics of individuals.
- Automata models can help recognize and authenticate bio-metric patterns accurately and efficiently.
- Automata can recognize various bio-metric patterns, such as fingerprints, iris scans, facial features, voiceprints, and gait analysis.
- Each bio-metric pattern has distinctive characteristics that can be captured and analysed using automata models.
- Automata-based recognition techniques provide high accuracy and reliability in identifying individuals based on their bio-metric patterns.
- Automata models are designed to capture the unique features of bio- metric patterns.
- These models use algorithms to analyse and compare the input bio-metric pattern with a pre-defined set of patterns.
- The automata model assigns a similarity score to determine the degree of match between the input pattern and the reference patterns. Automata-based recognition techniques offer robustness against noise and variations in bio-metric patterns.
- These techniques can handle large-scale bio-metric databases efficiently, enabling quick identification and authentication.
- Automata models provide high accuracy and reliability, reducing the chances of false positives or false negatives in bio-metric recognition.

7) Application on pattern recognition technique in bio-metric

- Pattern recognition techniques have revolutionized the field of bio-metrics by providing accurate and efficient methods for identifying individuals based on their unique physiological or behavioural characteristics.
- These techniques have found applications in various fields, including security, healthcare, access control, and forensic science.
- By leveraging advanced algorithms and machine learning, pattern recognition techniques have significantly improved the accuracy and reliability of bio-metric systems.

Facial recognition

- Facial recognition is one of the most widely used pattern recognition techniques in bio-metric applications.
- It analyses unique facial features, such as the distance between the eyes, shape of the nose, and contours of the face, to identify individuals.
- Facial recognition is employed in security systems, law enforcement, and access control, enabling quick and contactless identification.

FINGERPRINT RECOGNITION

- Fingerprint recognition is another popular pattern recognition technique in bio-metrics.
- It analyses the unique patterns, ridges, and valleys present in a person's fingerprints to establish their identity.
- Fingerprint recognition is widely used in law enforcement, border control, and mobile devices for secure authentication.

IRIS RECOGNITION

- Iris recognition is a pattern recognition technique that analyses the unique patterns in the cooler part of the eye, known as the iris.
- This technique is highly reliable, as the iris pattern remains stable throughout a person's lifetime.
- Iris recognition is used in high-security applications, such as national identification systems, airports, and secure facilities.

VOICE RECOGNITION

- Voice recognition is a pattern recognition technique that analyses the unique characteristics of an individual's voice, including pitch, tone, and vocal tract shape.
- It is used for speaker identification and verification in applications such as call centres, voice assistants, and authentication systems.
- Voice recognition can also be used for forensic analysis, helping to identify individuals based on recorded audio evidence.

HANDWRITING RECOGNITION

- Handwriting recognition is a pattern recognition technique that analyses the unique characteristics of an individual's handwriting, such as stroke patterns and letter formations.
- It is used in applications such as signature verification, document authentication, and forensic analysis.
- Handwriting recognition can help detect forgery and provide evidence in legal proceedings.